Deployment of a Plug-In Multi-Agent System for Traffic Signal Timing

Behnam Torabi, Rym Zalila-Wenkstern
University of Texas at Dallas
Richardson, Texas, USA
{behnam.torabi,rymw}@utdallas.edu

Robert Saylor, Patrick Ryan
Department of Transportation and Mobility
City of Richardson, Texas, USA
{robert.saylor,patrick.ryan}@cor.gov

ABSTRACT

In this paper we discuss the deployment of DALI, a multi-agent traffic signal timing system. Intersection controllers are augmented with agents which communicate with one another through direct links. The agents collaboratively adapt signal timings by considering the feedback of all agents affected by a change. DALI was deployed in the City of Richardson's Waterview Parkway corridor at three major intersections. The data collected for a three week period shows that on average, DALI reduced delay by 40.12%.

KEYWORDS

Deployed Multi-Agent Systems; Intelligent Transportation Systems; Traffic Signal Timing

ACM Reference Format:
Behnam Torabi, Rym Zalila-Wenkstern and Robert Saylor, Patrick Ryan. 2020. Deployment of a Plug-In Multi-Agent System for Traffic Signal Timing. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 9 pages.

1 INTRODUCTION

In the US, over 300,000 traffic signals punctuate the daily lives of hundreds of millions of people. When properly operated and maintained, traffic signals improve the safety and efficiency of traffic by reducing the frequency of accidents, facilitating the orderly movement of traffic flow along the streets.

Current traffic management systems have been used for decades [19, 20]. These systems consist of intersection controllers [1], i.e., devices that control the intersection's traffic signals; vehicle detectors; a communications network; and a central computer or a hierarchy of computers to manage the system. Traffic engineers interact with the intersection controllers through the central computer.

In traffic management systems, traffic engineers use traffic data to define signal timing plans, and communicate these plans to the intersection controllers. The controllers monitor the signal operations and change the timing plans by time-of-day or in response to external inputs (e.g., vehicle detection). A few systems provide adaptive signal timing mechanisms that allow controllers to adjust the signal timing parameters based on vehicle detector data [14, 22, 23]. Although results show that most adaptive systems improve traffic operations, only 4% of the total signalized intersections in the US have an adaptive control strategy [31]. This minimal implementation is due to the following: a) concerns for costs as adaptive systems require the installation and maintenance of a number of detectors at specific locations, on every road segment; b) concerns for the shortage of skilled staff to operate and maintain these systems; and c) reservations about the actual benefits [31].

From a research perspective, the application of the agent paradigm to traffic signal timing has been of interest to the MAS community for some time. Distribution, autonomy and collaboration are agent properties that are naturally suited for the traffic domain. Several signal timing approaches using various techniques have been discussed in the literature (e.g., game theory [4, 5], neural networks [6, 25], fuzzy logic [13, 30], reinforcement learning [3, 7, 11, 15, 21]). Most approaches simplify the signal timing problem and were validated on simple simulated traffic networks. Only a very few were validated on realistic simulated traffic network models [10, 11]. No published papers discuss the deployment of a collaborative multi-agent system for traffic signal timing.

In this paper, we discuss the deployment of DALI (Distributed Agent-based traffic LIghts), a collaborative multi-agent Traffic Signal Timing system (TST). In DALI, intersection controllers are augmented with software agents which collaboratively adapt signal timings by considering the feedback of all controller agents affected by a change. DALI agent's algorithms for congestion reduction and their evaluation in a simulated model were introduced in [26, 27]. In this paper we discuss how real-world deployment constraints have impacted DALI's architecture and the agent's algorithms. We present the agents' collaborative approach for *non-congested* scenarios and discuss how DALI was deployed on three major intersections in the City of Richardson, Texas. We present the evaluation results which show an average reduction in delay of 40.12%, and share the lessons learned in transitioning agents from a simulation environment to the real world.

2 RELATED WORK

Many traffic signal timing approaches have been proposed by manufacturers, traffic engineers and researchers. In this section, we restrict our discussion to TSTs that have been deployed. A discussion on research-oriented solutions for TSTs

---

[1] In this paper, the term "intersection controller" is used to refer to the traffic engineering concept of "traffic control."

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

is discussed in [28]. We classify these systems according to the decision making approach employed, namely *fully centralized*, *partially centralized* and *decentralized*.

Fully Centralized. Centralized TSTs are systems in which intersection controllers are fully controlled by one or several higher-level entities. The higher-level computer is responsible for making decisions about signal timings. Systems in this category include TRANSYT and SCOOT.

TRANSYT [20] is an off-line system. It uses historical data to calculate the network's performance index and then applies an optimization process to determine whether changes to the signal settings will improve the index. The main limitation of TRANSYT is the use of historical data which often results in timing plans that are out-of-date and ill-matched with the current traffic conditions.

SCOOT (Split, Cycle, and Offset Optimization Technique) [19] is an online, centralized TST. SCOOT was first deployed in the UK in the 80s and has been used worldwide in more than 200 locations [31]. Traffic data is collected in real-time through road sensors and processed every few seconds. The data is passed on to a central computer which predicts queue lengths. The predictions are passed onto an optimizer which determines the optimal timing. Optimizations take effect by incrementally updating a fixed-time plan. Both SCOOT and TRANSYT are responsive control systems with fully centralized control. As such, they are not fit to accommodate highly dynamic traffic patterns and changes in the traffic network.

Partially Decentralized. *Partially centralized* TST are systems in which the intersection controllers have full control over the definition and execution of the signal timing plans for their intersection, but the intersection controllers' coordination or control strategies are defined and monitored by one or more higher-level computers. TSTs in this category include SCATS, UTOPIA and RHODES.

SCATS (Sydney Coordinated Adaptive Traffic System) [23] was deployed in Australia in the late 70s. It has been widely used in countries such as the US, China, Singapore and Ireland [31]. SCATS is structured as a three-layered hierarchical system with a *control center* at the highest level, followed by *regional computers* in the next layer and *local intersection controllers* at the lowest layer. The central computer monitors the system performance whereas the regional computers execute area-based adaptive strategies. Local controllers can modify, within certain limits set by their regional master, their intersection signal settings in response to local traffic conditions. SCATS was primarily designed to respond to time-of-day and long-term variations in traffic. This is achieved by increasing the timings by a few seconds every cycle in response to changes in the traffic conditions. SCATS makes use of real-time measurements from the intersections' incoming roads only. As such SCATS does not perform well when unexpected traffic disruptions occur.

UTOPIA (Urban Traffic Optimization by Integrated Automation) [29] was developed by Mizar Automazione in Turin, Italy and has been used in several countries including Italy, Sweden, Norway and Finland. UTOPIA uses a two-level hierarchical structure. At the lower intersection level, controllers implements signal timings according to the local traffic conditions. The higher *area level* is responsible for setting the network control strategy (i.e., weights for all the elements, minimum and maximum length of each stage, and offsets). The central philosophy of UTOPIA is to provide absolute priority to public transport vehicles and improve the traffic flow for private vehicles, when possible.

RHODES [22] is another adaptive system that has been deployed in 4 locations in the US [31]. RHODES' architecture decomposes the control-prediction problem into three hierarchical levels: 1) intersection control; 2) network control, and 3) network loading. At the network control level, predictions of platoon flows are used to establish coordination constraints for each intersection in the network. At the highest level, the network loading level predicts the general travel demand over longer periods of time, typically one hour. RHODES requires complex intersection control equipment that is not readily available in the field.

Although partially centralized systems allow intersection controllers to have more decision-making responsibilities, network-level decisions are still made at higher levels.

Fully Decentralized. Decentralized TSTs are systems in which both decision making for signal timing plans and network-level coordination is given to the intersection controllers. A central computer may exist, but its responsibility is limited to traffic monitoring and data management.

PRODYN (Programmation Dynamique) [16] is one of the first attempts at developing a distributed TST. It was field tested in the early 1990s in the Zone Experimentale et Laboratoire de traffic de Toulouse, France [12]. In PRODYN, the basic optimization criterion is the minimization of delay which is achieved by a Bayesian estimation of queue lengths. Coordination between controllers is implicit (i.e., no direct communication between controllers) and achieved by sending information (i.e., output of the application of the optimal policy) to the direct downstream controllers. PRODYN's optimizations use dynamic programming which severely limits scalability [9].

OPAC (Optimized Policies for Adaptive Control) [14] was the first comprehensive strategy developed in the US for decentralized, adaptive TST. It was deployed in 4 locations [31]. OPAC has gone through several development cycles ranging from OPAC I [14] to OPAC-VFC [14]. OPAC's general control strategy features a dynamic optimization algorithm that calculates signal timings to minimize a performance function for the total intersection delay and stops. Similarly to PRODYN, OPAC is limited by the complexity of dynamic programming. Also, as for PRODYN, coordination between controllers is implicit and based on information sharing with direct neighbors. In both PRODYN and OPAC-VFC, controllers aim to find an optimal signal timing plan for their intersections using their direct neighbors' traffic data [14]. They do not make collaborative network-level decisions. It is well-known that an optimization at the intersection-level may lead to undesirable outcomes at the network-level.

In SURTRAC [24], the intersection control optimization is formulated as a scheduling problem where each intersection is considered as a single machine, and platoons of vehicles as "non-divisible jobs". Intersection controllers receive information about incoming vehicles from their direct neighbors and use forward dynamic programming to calculate near optimal schedules. Similarly to the approach followed by PRODYN and OPAC, in SURTRAC, the interactions between controllers are limited to exchange of traffic data about neighboring intersections, and scheduling is done in isolation, at the intersection level. In addition, from a deployment perspective, although it is mentioned that SURTRAC can be used with a variety of detectors (including basic inductive loops), the only deployed version discussed in the literature uses advanced video cameras. In addition, SURTRAC requires that advanced detectors be placed on the upstream end points of entry approaches.

In this paper, we discuss the deployment of a fully decentralized, collaborative multi-agent Traffic Signal Timing system (TST) that we call DALI (Distributed Agent-based traffic LIghts). DALI aims at turning existing TSTs with basic sensing mechanisms (e.g., inductive loops) into smart TSTs by plugging-in software agents into controllers. DALI's unique characteristics include:

(1) DALI agents continuously communicate with one another through direct links (i.e., explicit communication).
(2) A DALI agent's goal is to execute a timing strategy that a) improves the traffic flow at its intersection and b) does not create congestion at downstream intersections. As such, the decision-making for a signal-timing change is collaborative and involves the feedback of all controllers that may be impacted by the change.
(3) As a software solution, DALI does not require the installation of additional expensive hardware or additional sensors.

3 DEPLOYMENT CONSTRAINTS

In this section we discuss the deployment constraints that had to be considered during the development of DALI.

C1 **Infrastructure**. The current traffic infrastructure is to remain as-is. No additional equipment or hardware is to be acquired.

C2 **Controller**. DALI should work with a variety of controllers manufactured by different companies. These controllers should be programmable and IoT ready (i.e., can be connected to a server via the internet).

C3 **Communication**. Communication should be achieved through WiMAX or similar technologies. WiMAX is a wireless broadband communication technology that embodies the IEEE 802.16 family of standards and enables the delivery of wireless broadband access [1].

C4 **Traffic Engineering and Safety Standards**. Traffic engineering and safety standards should be implemented at the level of commercial systems.

C5 **Detector Type**. There should be no assumption on the availability of video cameras, radars or other advanced sensing devices. Vehicles are detected through basic inductive loops. An inductive loop is a coiled wire that is formed into a loop and installed under the surface of roadways. When a vehicle passes over the loop or is stopped within its area, a pulse is sent to the traffic signal controller signifying the passage or presence of a vehicle.

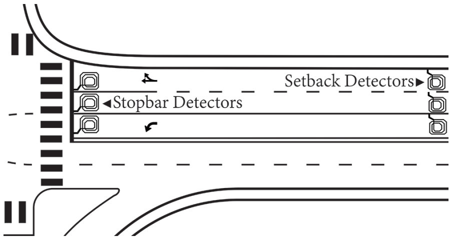

Figure 1: Induction Loops and Vehicle Sensing Area.

C6 **Number and Position of Detectors**. There is a limited number of detectors on each road segment. The inductive loops are either *stopbar detectors*, i.e., placed right behind the stop bar or *setback detectors*, i.e., placed hundreds of feet away (see Figure 1). Residential roads may not be equipped with detectors.

C7 **Agent Update**. In the initial deployment phase, agent behavior should be monitored and easily updated.

4 ARCHITECTURE

Several deployment constraints played a central role in the definition of the DALI architecture. (C1) imposed that the DALI solution be purely software. To fulfill (C2) we decided to design DALI agents as "plug-in" components that interact with their respective controllers through well-defined interfaces. (C3) required that agent-to-agent and agent-to-controller communications use wireless broadband technologies as opposed to wired technologies such as cable and DSL. To comply with (C4), instead of developing a traffic software from scratch, we decided to use the full capabilities and safety features offered by existing controllers. To this effect, we designed agents to serve as the "brains" of the controllers, and the controllers as plan executors. Therefore, the conventional pre-defined timing plan updates periodically "pushed" into the controllers by traffic engineers is replaced with continuous real-time plan commands sent by the agent. The execution of these plans uses the original commercial controller software which implements all traffic and safety requirements.

These design decisions led to the architecture given in Figure 2. The original centralized traffic control system which comprises a central management system connected wirelessly to a number of intersection controllers remains untouched. A controller consists of a web server and a black box component, i.e., proprietary software that implements traffic operations and safety features. This black box component also implements the controller's behavior which includes (i) detector data processing, (ii) timing plan execution and (iii) signal change. The controller's web server includes an interface that

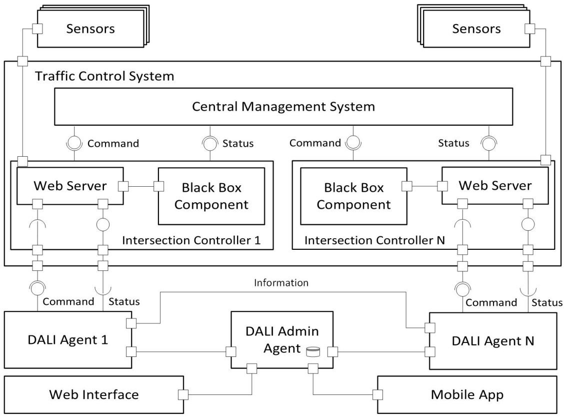

Figure 2: Architecture

is used by the central management system to send timing plans and receive intersection status details.

When operating in DALI mode, an agent interacts with its controller through the controller's web server to receive detector information (i.e., vehicle count) and send timing commands. In addition, the agent sends its status to the DALI Admin agent which stores the data in its database. DALI Admin continuously analyzes the stored and received data and notifies traffic engineers in case of unusual circumstances.

5 AGENT ALGORITHMS

We start this section by providing a few definitions of core traffic signal timing concepts. These definitions are based on the U.S. Department of Transportation Traffic Signal Timing Manual [18].

5.1 Traffic Signal Timing Concept Definitions

Phase: A controller timing unit associated with the control of one or more movements (i.e., through movement, right turn movement) at an intersection.
Interval: Duration of time during which the signal indications do not change. Examples of intervals include green, yellow and red intervals.
Cycle Length: This is the total time to complete one sequence of signalization around an intersection.
Split: It is the amount of time in a cycle length given to a particular phase. This time would include the green, yellow and red clearance time of a particular phase.
Offset: This is the time difference between the beginning of a cycle at one intersection and the beginning of a cycle at the adjacent coordinated intersections.

5.2 Impact of Deployment Constraints

An agent's decision making is mostly based on vehicle count and therefore, the limitations in the number and type of detectors imposed by constraints (C5) and (C6) add a level of complexity to the problem. As mentioned in Section 3, inductive loop detectors can only detect vehicles that cross the inductive loop area. Outside the area, vehicle positions are not known. In addition, given that detectors are only used at signalized intersections, vehicles traveling to/from residential or service entries are not detected. Therefore, in a real-world setting where inductive loops are used,

(1) we do not know with certainty which road segment a detected vehicle leaving an intersection will enter;
(2) when a vehicle enters a road segment, we do not know whether the vehicle will reach the next intersection or take a residential road;
(3) when a vehicle enters a road segment, we do not know which lane it will be on when it arrives at the next intersection;
(4) due to the detector location, it is not possible to get an accurate count of the queue length for each lane of a road segment.

Knowledge about the cases discussed above is critical for the execution of the agent's core algorithms and therefore we had to consider the following:
Definition of probabilities. We denote by C the set of intersection controller agents $\{c_1, .., c_n\}$ and by Rd the set of road segments $\{r_{1,2}, .., r_{m,n}\}$ between intersections. A road segment $r_{m,n}$ is defined in terms of attributes such as its set of lanes $\{ln_1..ln_q\}$. We denote by $\xi_{r_{m,n}.ln_w}$ the traffic flow on lane ln_w of road segment $r_{m,n}$. $\xi_{r_{m,n}.ln_w}$ is defined as the number of vehicles passing through the inductive loop area for lane ln_w per minute.
To compensate for the lack of accurate data for (1)-(3), we have to define two probabilities:
$p_1\left(r_{m,n}.ln_w, r_{n,p}\right)$, the probability that a vehicle exiting lane w in road segment $r_{m,n}$ enters road segment $r_{n,p}$, and $p_2\left(r_{m,n}, r_{m,n}.ln_w\right)$, the probability that a vehicle which enters road segment $r_{m,n}$, leaves it from lane w.
These probabilities are i) computed by the DALI Admin agent based on the high-resolution data stored in its database and ii) periodically passed on to the controller agents.
Estimation of Queue Lengths. Given the lack of advanced broad-view detectors (e.g., video cameras), it is not possible to get an accurate count of a vehicle queue for a specific road lane. Several cases need to be considered based on the detector type (i.e., *stopbar* or *setback*) and the status of the phase (i.e., green, red, yellow).

5.3 Collaboration to Optimize Split and Offset

5.3.1 Approach. In the DALI strategy, agents continuously communicate with one another to exchange traffic information. In order to illustrate collaboration in a non-congested scenario, we consider three consecutive unidirectional road segments $r_{v,s}$, $r_{s,n}$ and $r_{n,m}$ controlled respectively by agents c_s, c_n and c_m. By default, c_n exchanges two sets of information with c_m:

(1) The vehicles detected by road segment $r_{s,n}$'s detector, and for each vehicle, its estimated arrival time at road segment's $r_{n,m}$ stop bar. The vehicle's estimated arrival time is computed based on $p_1\left(r_{m,n}.ln_w, r_{n,s}\right)$, traffic flow rate ξ, and the distance between the two stop bars.

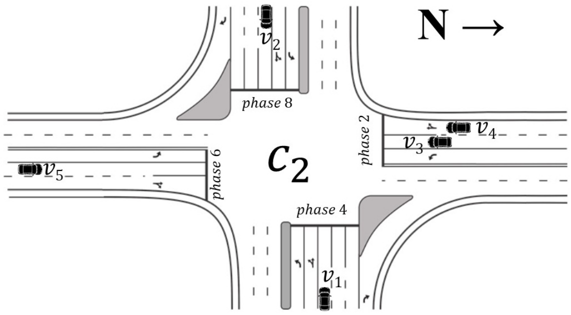

Figure 3: Overview of intersection status in case study.

(2) The anticipated vehicles to arrive at $r_{s,n}$'s stop bar from s and from any other intersection preceding s (up to a maximum estimated arrival time of four minutes), and the estimated arrival time of these vehicles at $r_{n,m}$'s stop bar based on c_n's current timing plan.

Taking into account the anticipated arrival time of vehicles coming from farther intersections is important for network topologies with short distances between intersections.

As the exchange of information takes place, agent c_n continuously defines possible timing plans based on ξ, the status of its phases, and timing constraints (e.g., minimum and maximum green, yellow and red clearance intervals). A timing plan includes a sequence of phase combinations as well as their *split* and *offset*. Given that the agent's goal is to find the values of the offset and split that minimize delay, for each timing plan, it computes the estimated delay using the phases' estimated queue lengths, estimated vehicle arrivals, and the value of probabilities p_1 and p_2. It then prioritizes the plans based on minimum delay, executes the plan with the highest priority, and re-start the process immediately.

5.3.2 Illustrative Example. To illustrate the definition and selection of timing plans by agents, we use the traffic scenario depicted in Figure 3. In this scenario, the intersection's phases four and eight are red, and phases 2 and 6 are green. We assume that the estimated vehicle arrival times communicated to c_2 by adjacent controllers for vehicles v_3, v_4, v_1, v_2 and v_5 to be 2s, 3s, 6s, 6s and 9s, respectively. Agent c_2 first defines possible timing plans. For the sake of simplicity, we assume that there are only two possible timing plans: a) plan pl_1: keep phases two and six green for nine seconds and then switch to phases four and eight; b) plan pl_2: keep phases two and six green for the next three seconds; then, switch to phases four and eight and keep them green for four seconds; finally switch back to phases two and six. Agent c_2 then computes the estimated delay for each timing plan. The delay for a plan is the sum of the estimated delay for vehicles. In their computations, we assume that the yellow interval is one second and there is no red clearance interval. For pl_1, given that v_1's estimated arrival is 6s, and phase four will become green after ten seconds, the estimated delay for v_1 is $10 - 6 = 4$ seconds. The estimated delay for v_2, v_3, v_4, and v_5 are 4s, 0s, 0s and 0s, respectively. Therefore, the estimated delay for plan pl_1 is eight seconds. With similar computations, the estimated delay for pl_2 is zero. Therefore, c_2 selects and executes pl_2.

5.4 Collaboration to Alleviate Congestion

As they plan to optimize split and offset, agents continuously evaluate their phases' congestion levels. For instance agent c_n, computes the congestion level $Cong$ for phase $ph_{n,k}$ as the sum of the traffic flows ξ of the lanes controlled by $ph_{n,k}$. If $Cong$ is above a given threshold a for a certain duration d, c_n determines that phase $ph_{n,k}$ is congested. It then deliberates and defines a new timing configuration to alleviate congestion at its intersection by adjusting the split of $ph_{n,k}$. It proceeds by determining the effects of executing this possible configuration on each neighboring intersection. These effects correspond to the additional traffic flow called κ that neighboring intersections need to account for in case the plan is to be executed. A request for evaluation including κ is then sent to each affected neighbor.

Upon receipt of κ, a receiving agent deliberates and, based on the status of its intersection, computes the effect of executing the new timing configuration on each outgoing road of its intersection. It computes and sends its own κ to its neighboring agents that are affected by the change. And the process iterates until it either reaches an intersection within the city boundaries for which κ is below threshold g (i.e., the effect is insignificant), or an exit junction at the city's boundaries. Then, each farthest affected agent determines its level of agreement Ψ with the effects of the timing configuration. Ψ, a real number, is computed based on the status of the intersection, the priority level of the intersection and κ.

Each agent sends its Ψ to the requesting agent which incorporates it in the computation of its own Ψ. The backward propagation continues until the initiating agent receives the Ψs from the immediate agents. The initiating agent then deliberates and decides whether to execute or ignore the plan based on the received values of Ψ. It proceeds by informing the agents of its final decision and, in case the plan is to be executed, requests that all affected agents update their timings accordingly. The detailed agent algorithms for congestion management is given in [26].

In order to make the collaborative process adaptive, an RL-based approach is used to dynamically assign values to thresholds a, d and g. Threshold a controls the agent's sensitivity to detecting congestion. The lower the values of a, the higher the likelihood for an agent to detect congestion. Threshold d controls the duration a phase needs to be flagged as congested in order to be considered as congested. Finally, g controls the collaboration scope. With lower values of g, a higher number of agents will be involved in the decision-making for a new timing plan. The detailed agent algorithms for adaptive assignment of threshold values is given in [27].

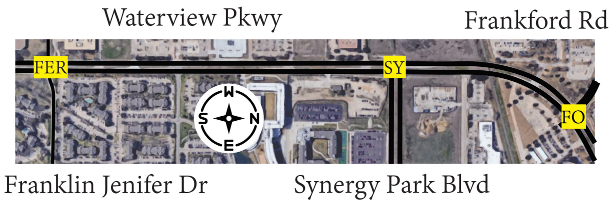

Figure 4: Overview of The Corridor.

6 EVALUATION

6.1 Deployment of DALI

6.1.1 The City of Richardson. The City of Richardson is located fifteen miles north of downtown Dallas and is part of the Dallas-Fort Worth Metroplex. The City's traffic control system offers SCATS-based functionalities. We refer to this system as SCATS-R. The intersection controllers are manufactured by Intelight [17]. An Intelight controller integrates a web server called MAXTIME which is accessed remotely via VPN. The communication protocol with the client side is achieved through html forms. A central traffic management center communicates with the controllers via a WiMAX wireless network, and vehicles are detected through inductive loops.

DALI agents were deployed on the Waterview corridor which includes three intersections (see Figure 4): Waterview and Frankford Rd (FO), Waterview and Synergy Park Blvd (SY), and Waterview and Franklin Jenifer Dr (FER).

6.1.2 Safety and Monitoring Constraints. When operating in SCATS-R mode, the Intelight controllers execute the timing plans programmed by the City of Richardson's traffic engineers. When in DALI mode, they operate as instructed by the agents. To enforce safety constraint (C4) (see Section 3), when a disruption occurs in DALI mode, the agents give back control to the Intelight controllers which resume their SCATS-R operations.

With respect to monitoring constraint (C7) (see Section 3), while it would have been possible to place the agents inside the Intelight controller cabinets, we decided to install them in the lab computers and use VPN to connect to the controllers. This approach allowed us to closely monitor all agents concurrently and adjust their behavior in a timely manner (see Figure 5).

6.1.3 Agent Implementation. The agents were implemented in JAVA. Each agent runs on a PC with 2 gigabytes of ram and 3.33 GHz clock and communicates with the other agents through a Broadcom gigabit ethernet net link with a minimum speed of 100 mbps. Each agent runs three threads to update its intersection status, one thread to execute its algorithm and one thread to communicate with other agents.

6.1.4 Agent-Controller Interaction Mechanism. In order for a DALI agent to instruct a controller to perform an action, it was necessary to:

a) Define a new *pattern* for the agents. A pattern specifies the timing configurations, i.e., operation mode, detector plan, sequence of phases, etc. The DALI pattern is associated with an *action* which is included in a *day plan*. Agents update the day plan every one minute.

Figure 5: DALI Agents Running in the Lab.

b) Define *virtual detectors*. These *virtual detectors* are used by the agents to simulate a detector state (*active* or *inactive*) and therefore trigger the execution of a signal change by the controller. In DALI mode, a controller operates in fully actuated mode. As such, it always gives green to a phase when a vehicle is detected for that phase and no vehicles are detected for the other phases. When an agent wants a signal to be green, it changes the status of the virtual detector to *active* and all other virtual detectors to *inactive*. This prompts the controller to give green to that phase. The same approach is followed for a combination of non-conflicting phases. From a deployment perspective, the virtual detectors controlled by the agents are connected to a spare Intelight controller's input point.

6.1.5 Executing DALI Agents. At initialization time, a DALI agent connects to its controller's web server through VPN and changes the controller's configuration so that the controller executes the DALI day plan and not its original plan. This is achieved by adding a rule to the controller's *schedule table*.

The agent then takes control and immediately requests an update on the status of the intersection. The Intelight controller updates a form called the *phase status form* which shows the state of the intersection's traffic lights (i.e., red, green or yellow) and detectors (i.e., active or inactive). This information is read by the agent. The time interval between the agent request and the controller update called *communication speed* has to be less than three hundred milliseconds. To release control to the Intelight controller, an agent only needs to remove the rule from the schedule table.

6.2 Experimental Results

Various measurements are commonly used in Texas to evaluate the effectiveness of a signal timing system. In this section, we discuss the evaluation of DALI with respect to *delay*, and *cost of the delay*. Delay is defined as the increment in a vehicle's travel time caused by traffic control devices, compared with the travel time if the vehicle was to maintain its

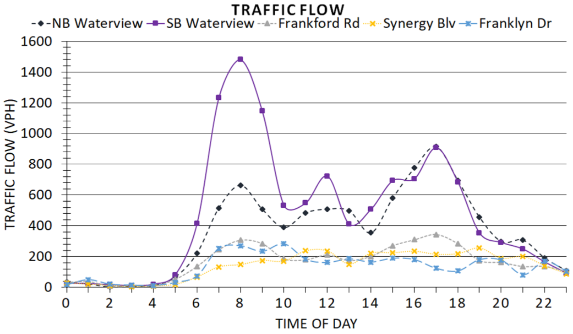

Figure 6: Traffic Flow Rate for Different Times of the Day.

Table 1: Distribution of data collection time

Dali - Duration of Control (Hours)

	Peak	Off-peak	Midnight	Total
Weekdays	38.49	67.37	36.16	142.02
Weekends	29.84	53.87	34.25	117.61
Total	67.97	121.24	70.42	259.63

SCATS-R - Duration of Control (Hours)

	Peak	Off-peak	Midnight	Total
Weekdays	54.65	93.09	40.64	189.20
Weekends	15.08	29.83	25.68	70.59
Total	69.73	123.74	66.32	259.80

expected speed in the absence of any control device [2]. The cost of delay is computed using data from [8].

An initial analysis of traffic on the Waterview corridor showed that overall, traffic variations occur between 7:00 to 9:00 and 16:00 to 20:00; 9:00 to 16:00 and 20:00 to 00:00; and 00:00 to 7:00 (see Figure 6).

The results presented in this section are based on data collected over a period of 520 hours as follows: we ran DALI and gathered data for 260 hours. Then we turned off DALI and gathered data for SCATS-R for the remaining 260 hours. We considered week days and weekends, and given the traffic variations discussed above, divided days into three time periods: peak hours (i.e., 7:00 to 9:00 and 16:00 to 20:00), off-peak hours (i.e., 9:00 to 16:00 and 20:00 to 00:00), and nighttime (i.e., 00:00 to 7:00). Table 1 shows the distribution of the 520 data collection hours.

6.2.1 Delay. Figure 7 shows the average reduction in delay for different time periods. We notice that there is a high reduction in delay for time periods *week day nighttime*, *weekend peak & off peak* and *weekend nighttime*. This is explained by the fact that, during nighttime (on weekdays and weekends), traffic is very slow and there are only very few requests for green signals at the intersections. In SCATS-R mode, once a vehicle is detected, the Intelight controllers either give green immediately or, if in the middle of a cycle, complete the sequence then give green. In DALI mode, given that the agents can predict the arrival of vehicles, they schedule green signals before the vehicles arrive at the intersection. On weekends, during the day (peak and off-peak), although traffic is slightly heavier than during nighttime, the same behavior occurs.

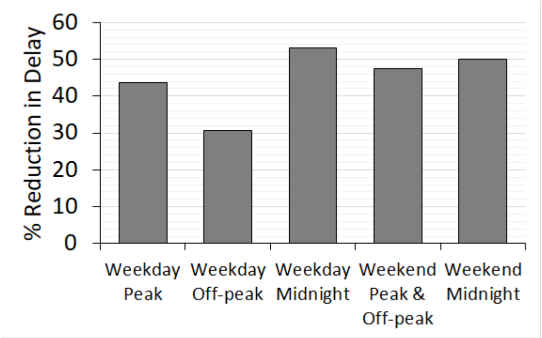

Figure 7: Delay Reduction.

Figure 7 shows that the lowest reduction in delay happens for *weekday off-peak* hours. The 12.81% difference between *weekday peak* hours and *weekday off-peak* hours is due to the fact that during peak hours, intersections are likely to be congested and the agent collaboration process is superior to the SCATS-R timing strategy. During off-peak hours, the number of vehicles although high does not necessarily lead to congestion. Therefore agents are unlikely to collaborate and their performance is not as superior.

6.2.2 Cost of Delay. Figure 8 shows the average delay (in seconds) and the corresponding average cost of delay (in US $) for each vehicle at each intersection for both SCATS-R and DALI. According to [8], in the Dallas area, the cost of delay for a driver is on average $31 per hour which translate into $0.0086 per second. For *weekdays peak* hours, in SCATS-R mode, the average delay is 43.2 seconds. In the same period, in DALI mode, the average delay is 24.38 seconds and therefore delay reduction is on average 43.2-24.38=18.8 seconds which translates into a savings of 18.8×$0.0086=$0.16 per vehicle per intersection. During that period, an average of 1,604 vehicles/hour passed through a Waterview intersection. Therefore the average savings for all the vehicles going through an intersection is $1,604 × $0.16 = $256.64/hour.

We computed the average delay for SCATS-R and DALI during their respective 260 hours of execution time. This was achieved by computing the difference between the arrival time and the departure time of each vehicle at each intersection. The value of the average delay for a vehicle at an intersection for SCATS-R is 27.94 seconds whereas for DALI it is 16.73 seconds. Therefore delay reduction is on average 27.94-16.73=11.21 seconds which translates into a savings of 11.21×$0.0086=$0.1 per vehicle per intersection. During the 260 hours of execution, an average of 725 vehicles/hour pass through one Waterview Pkwy intersection. Therefore the average savings for vehicles going through the three Waterview intersections over the 260 hours is 3×725×$0.1×260=$65,550 overall.

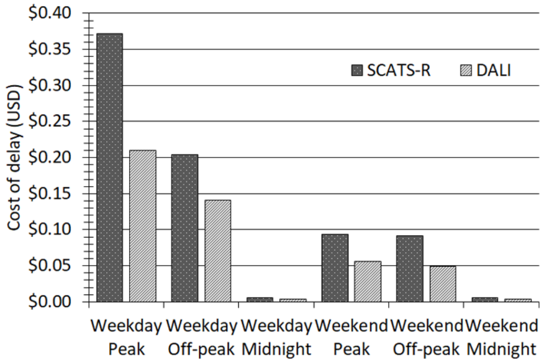

Figure 8: Average delay and cost of the delay for each vehicle at each intersection.

7 LESSONS IN TRANSITIONING FROM THE LAB TO THE REAL WORLD

The deployment of DALI is the result of a collaboration between university researchers and a municipality for the purpose of deploying and field-testing outcomes of research at a city level. The collaboration resulted in valuable lessons in three areas:

Stakeholder Consent.Transitioning from the lab to the real world required two years of interactions with various stakeholders. The deployment of DALI on three intersections in the Waterview corridor required that a) the DALI model and agent algorithms be validated by traffic engineers; b) a full replica of the City of Richardson's traffic network be created in a simulation model for review by the Director of the Transportation Department; c) extensive simulations be conducted based on historical data and presented to the City Manager; d) controllers in the field be integrated in the simulation model and validated by traffic engineers; and e) extensive testing be done in the hybrid simulation using real-time data with results presented to the City Council.

Learning Curve. Researchers often simplify the traffic signal timing problem and make unrealistic assumptions. Before developing solutions, it is critical for researchers to learn the theory of traffic signal timing and experiment with the development of actuated traffic signal timing plans. In addition, it is also necessary to learn about the technology used in the development of commercial intersection controllers.

Deployment Hurdles. Expect the unexpected during deployment. Although most unusual deployment issues were discussed during the initial development of DALI, many were discovered during deployment and required architecture and code updates. We mention the following:

Time Synchronization. In the simulated traffic environment, we assumed that all agents use the same clock time. After deployment, we realized that the controller's clock values were not synchronized. As a result, when an agent sends a vehicle detection time to its neighbor, the neighbor computes the estimated arrival time for that vehicle incorrectly. To address this issue, we added a *time-server* to the DALI Admin agent. When an agent detects a vehicle, it sends the detection time using the *time-server* time and not the controller time.

Detector Failures. We realized that when a detector fails, two things happen: the detector status (i.e., *active*, *inactive*) stays the same for a long period of time, or it changes randomly. We also realized that the Intelight controllers do not have a mechanism to address detector failures. We improved our agent algorithms to consider this case as follows: when an agent notices that a detector has failed, it replaces the actual detector data with an estimate of the data. This estimate is computed using prediction models and historical data.

Loss of Communication. Communications get interrupted. When an agent loses communication with its controller, the virtual detector that was given the status *active* by the agent remains in that status for as long as the communication is interrupted. This results in giving maximum green to the phase associated with the detector, and minimum green to all other phases. In order to avoid this situation, we decided that if an agent is disconnected from its controller for over two minutes, the controller is given control of the intersection.

Road Construction. While DALI was running, some road construction work was done at the Franklyn intersection. During construction, the topology of the intersection was changed. This required a re-definition of the intersection structure for the agent controlling the Franklyn intersection.

Rain. During heavy rainy days, the communication speed between agents and controllers decreased drastically. In addition, detectors did not work properly. Given that an agent's performance highly depends on the communication with its controller and the detection of vehicles, we decided that when the communication speed is below 300ms, an agent has to give control to the Intelight controllers.

8 CONCLUSION

In this paper we discussed the deployment of a collaborative multi-agent Traffic Signal Timing system (TST) that we call DALI on the Waterview Pky corridor, in the City of Richardson, Texas. Our goal was to transform the City of Richardson's SCATS-R system into a smart collaborative system without any change to the infrastructure. To this effect, we implemented DALI agents as add-ons to the Intelight intersection controllers used by the City. The execution of DALI over a period of three weeks resulted in a reduction of delay by 40.12%.

With the demonstrated success of DALI in this initial deployment, the City of Richardson has initiated a year-long upgrade of its communications and detection systems for additional deployment. In addition, the City of Dallas expressed interest in the deployment of DALI in the City's Smart Corridor.

REFERENCES

[1] Syed A Ahson and Mohammad Ilyas. 2018. *WiMAX: applications*. CRC press.

[2] Kevin N Balke and Curtis Herrick. 2004. *Potential measures of assessing signal timing performance using existing technologies*. Technical Report FHWA/TX-04/0-4422-1. Texas A&M Transportation Institute, College Station, Texas 77843-3135.

[3] Ana LC Bazzan, Denise de Oliveira, and Bruno C da Silva. 2010. Learning in groups of traffic signals. *Engineering Applications of Artificial Intelligence* 23, 4 (2010), 560–568.

[4] Khac-Hoai Nam Bui and Jason J Jung. 2018. Cooperative game-theoretic approach to traffic flow optimization for multiple intersections. *Computers & Electrical Engineering* 71 (2018), 1012–1024.

[5] Rodrigo G Castillo, Julio B Clempner, and Alexander S Poznyak. 2019. Solving traffic queues at controlled-signalized intersections in continuous-time Markov games. *Mathematics and Computers in Simulation* (2019).

[6] Kuei-Hsiang Chao, Ren-Hao Lee, and Meng-Hui Wang. 2008. An intelligent traffic light control based on extension neural network. In *Knowledge-based intelligent information and engineering systems*. Springer, 17–24.

[7] Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. 2019. Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control. *IEEE Transactions on Intelligent Transportation Systems* (2019).

[8] Graham Cookson and Bob Pishue. 2017. Inrix global traffic scorecard. *INRIX Research, February* (2017).

[9] Christina Diakaki, Markos Papageorgiou, and Kostas Aboudolas. 2002. A multivariable regulator approach to traffic-responsive network-wide signal control. *Control Engineering Practice* 10, 2 (2002), 183–195.

[10] Ivana Dusparic and Vinny Cahill. 2012. Autonomic multi-policy optimization in pervasive systems: Overview and evaluation. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 7, 1 (2012), 11.

[11] Samah El-Tantawy, Baher Abdulhai, and Hossam Abdelgawad. 2013. Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): methodology and large-scale application on downtown Toronto. *IEEE Transactions on Intelligent Transportation Systems* 14, 3 (2013), 1140–1150.

[12] JL Farges, I Khoudour, and JB Lesort. 1994. PRODYN: On site evaluation. In *Third International Conference on Road Traffic Control, 1990.* IET, 62–66.

[13] Maheen Firdous, Fasih Ud Din Iqbal, Nouman Ghafoor, Nauman Khalid Qureshi, and Noman Naseer. 2019. Traffic Light Control System for Four-Way Intersection and T-Crossing Using Fuzzy Logic. In *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*. IEEE, 178–182.

[14] Nathan H Gartner and Chronis Stamatiadis. 2019. Optimization and Control of Urban Traffic Networks. *Complex Dynamics of Traffic Management* (2019), 131–165.

[15] Hongwei Ge, Yumei Song, Chunguo Wu, Jiankang Ren, and Guozhen Tan. 2019. Cooperative Deep Q-Learning With Q-Value Transfer for Multi-Intersection Signal Control. *IEEE Access* 7 (2019), 40797–40809.

[16] Jean-Jacques Henry, Jean Loup Farges, and J Tuffal. 1983. The PRODYN real time traffic algorithm. *IFAC Proceedings Volumes* 16, 4 (1983), 305–310.

[17] Intelight ITS. [n. d.]. https://www.intelight-its.com/. ([n. d.]). Accessed November 2019.

[18] Peter Koonce, Lee Rodegerdts, Kevin Lee, Shaun Quayle, Scott Beaird, Cade Braud, Jim Bonneson, Phil Tarnoff, and Tom Urbanik. 2008. *Traffic signal timing manual*. Publication FHWA-HOP-08-024. Department of Transportation, 1200 New Jersey Ave, SE, Washington, DC 20590.

[19] UK's Transport Research Laboratory. [n. d.]. Split Cycle and Offset Optimisation Technique. https://trlsoftware.co.uk/products/traffic_control/scoot. ([n. d.]). Accessed November 2019.

[20] UK's Transport Research Laboratory. [n. d.]. Traffic Network and Isolated Intersection Study Tool. https://trlsoftware.co.uk/products/junction_signal_design/transyt. ([n. d.]). Accessed November 2019.

[21] Congcong Li, Fei Yan, Yiduo Zhou, Jia Wu, and Xiaomin Wang. 2018. A Regional Traffic Signal Control Strategy with Deep Reinforcement Learning. In *2018 37th Chinese Control Conference (CCC)*. IEEE, 7690–7695.

[22] Pitu Mirchandani and Larry Head. 2001. A real-time traffic signal control system: architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies* 9, 6 (December 2001), 415–432.

[23] Roads and Maritime Services. [n. d.]. SCATS: The benchmark in urban traffic control. http://www.scats.com.au/. ([n. d.]). Accessed November 2019.

[24] Stephen F Smith, Gregory J Barlow, Xiao-Feng Xie, and Zachary B Rubinstein. 2013. Smart urban signal networks: Initial application of the surtrac adaptive traffic signal control system. In *Twenty-Third International Conference on Automated Planning and Scheduling.*

[25] Dipti Srinivasan, Min Chee Choy, and Ruey Long Cheu. 2006. Neural networks for real-time traffic signal control. *IEEE Transactions on Intelligent Transportation Systems* 7, 3 (2006), 261–272.

[26] Behnam Torabi, Rym Z Wenkstern, and Robert Saylor. 2018. A Collaborative Agent-Based Traffic Signal System For Highly Dynamic Traffic Conditions. In *Proceedings of the 21st IEEE International Conference on Intelligent Transportation Systems (IEEE ITSC 2018)*. Maui, Hawaii, USA, 626–633.

[27] Behnam Torabi, Rym Z Wenkstern, and Robert Saylor. 2018. A Self-Adaptive Collaborative Multi-Agent based Traffic Signal Timing System. In *Proceedings of the 4th IEEE International Smart Cities Conference (ISC2 2018)*. Kansas City, Missouri, USA, 1–8.

[28] Behnam Torabi, Rym Z Wenkstern, and Robert Saylor. 2020. A collaborative agent-based traffic signal system for highly dynamic traffic conditions. *Autonomous Agents and Multi-Agent Systems* 34, 1 (2020), 1–24.

[29] Trafitek. [n. d.]. ATC - UTOPIA. http://www.trafitek.com/atc-utopia.php. ([n. d.]). Accessed November 2019.

[30] Kaouther Youcef-Toumi, Mounir Bouhedda, and Sofiane Tchokecth-Kebir. 2018. Smart Cooperative Control of an Intersection Group Based on Fuzzy Logic. In *2018 International Conference on Applied Smart Systems (ICASS)*. IEEE, 1–4.

[31] Yi Zhao and Zong Tian. 2012. An overview of the usage of adaptive signal control system in the United States of America. In *Applied Mechanics and Materials*, Vol. 178. Trans Tech Publ, 2591–2598.