Computing and Informatics, Vol. 41, 2022, 609–626, doi: 10.31577/cai_2022_2_609

TOWARDS A NETWORK OF DYNAMIC MESSAGE SIGNS FOR CONGESTION ALERTING

Nabil Sahli

German University of Technology in Oman
Muscat, Oman
e-mail: nabil.sahli@gutech.edu.om

Wassim Trojet, Zhehan Zhang

ESIGELEC, Rouen
Normandie University
France
e-mail: wassim.trojet@esigelec.fr, zhehan.zhang@groupe-esigelec.org

Nesrine Ouled Abdallah

German University of Technology in Oman
Muscat, Oman
e-mail: n.ouledabdallah@gutech.edu.om

Abstract. Traffic applications such as Google Traffic and Waze have been introduced to let users know about existing congestions on real time. However, this cannot help drivers who are not using these applications or not connected to Internet. Besides, it also suggests that drivers can interact with their smart phones while driving, which is illegal in most countries. The idea of this paper is to use dynamic road signs which can collect real-time data from traffic applications and alert drivers who are heading towards congestions. A proof-of-concept of the dynamic road sign has been developed.

Keywords: Dynamic message sign, traffic congestion, traffic applications

1 INTRODUCTION

Smart cities involve the integration of multiple Information and Communications Technology (ICT), big data and Internet of Things (IoT) solutions to manage infrastructure and assets. In the transport sector, this includes traffic lights, parking controls and traffic management systems. The latter systems mainly aim to ease congestion and improve the flow of vehicles around the network.

Most big cities worldwide are suffering from traffic jams as the number of vehicles is exceeding current infrastructures' capacities. Consequently, researchers are constantly looking for new ways to reduce congestion and then air pollution. In this context, our proposal aims at finding concrete solutions to overcome or at least to mitigate traffic congestion. We focus here on a particular problem: how to alert drivers about congested junctions/roads to prevent making traffic jams worse.

In particular, when a road or a roundabout is congested because of an incident or the usual rush hour, drivers who are heading towards that particular road/junction would certainly like to be notified so that they can change their routes while it is still possible. Such notification should reach the driver while he/she still has other options to reach his/her final destination without going through the congested road. While in some developed countries, main roads are equipped with dynamic road signs which display alerts sent by a central traffic management unit, drivers in most countries still rely on traffic announcements and warnings reported in radio channels. However, this does not represent a reliable and continual real-time alert source since these notifications are not detailed and occur on scheduled time only. More technophile drivers will check a Traffic Application (e.g. Waze or Google Traffic) on their smart phones while driving. Such traffic applications provide drivers with detailed and real-time data, enabling them to avoid bottlenecks. However, such solutions present two main drawbacks:

1. Only drivers equipped with the application and connected to Internet can benefit from the service;
2. Interacting with the mobile is a distraction by itself.

It is also illegal according to the traffic laws of most countries.

The idea in this paper is to use Dynamic Message Signs (DMSs) which are equipped with a digital screen that can display warning messages to drivers. Unlike current dynamic road signs which display alerts based on instructions received by a central traffic management system, we follow a more decentralized approach where each road section is controlled by a DMS. DMS collects real-time data about its road section from traffic applications (e.g. Google Traffic or Waze) and displays alerts whenever a congestion is detected in its road section.

A hardware proof-of-concept of the DMS has been developed. A software module has been added to allow DMSs to detect congestions. The entire prototype has been tested with real-time traffic data of a road section in Muscat, Oman.

The remaining of this paper is organized as follows. Section 2 gives a brief about related works while Section 3 presents our approach which uses Dynamic Message

Signs and real-time traffic data to alert drivers about ahead congestions. Section 4 describes our proof-of-concept and Section 5 presents our preliminary results and discuss them. Finally, Section 6 concludes the paper and announces our future works.

2 RELATED WORKS

The traditional automatic traffic management scheme (the three-color traffic signals) is still widely used worldwide. Although it organizes the traffic locally (at each junction), it is not capable of preventing congestions in the global network. It may even cause congestions if the timing of the traffic light is not adjusted to the traffic flows. Nowadays, along with the traditional traffic signals, authorities are adopting an Active Traffic Management (ATM) approach which consists on controlling the traffic based on real-time traffic conditions while combining real-time and predictive operational strategies [1]. To implement this approach, different tactics can be used. For instance, Adaptive Ramp Metering [2] uses traffic signals on ramps in order to dynamically control the flow of vehicles entering the road/highway. Dynamic Lane Use Control [1] is another tactic where individual lanes are dynamically opened/closed depending on the traffic. Other tactics include Dynamic Shoulder Lane, Queue Warning, and Dynamic Speed Limits [1].

Alternatively, authorities may use Demand Modification Tactic (DMT), which involves actions from the drivers. Indeed, DMT includes the dissemination of travel information to the public in the hope that drivers will take other routes or change their travel time [3]. To disseminate such information, authorities use mobile apps, portals, and Dynamic Message Signs (DMS). DMT may also include Dynamic Route Guidance [4, 5] which is a more proactive dissemination of traffic information. This tactic is mainly used for unexpected congestions (non recurrent traffic congestions) due, for instance, to bad weather or accidents. It consists on monitoring the traffic flows and when a congestion is detected, dynamic message signs are used to alert drivers and inform them about rerouting options.

Our proposal focuses on the Dynamic Route Guidance (DRT) tactic which mainly relies on Dynamic Message Signs (DMS) that can alert drivers about changing events and situations. DMSs are devices which display alternative messages to provide drivers with messages/alerts about the traffic, based on real-time data.

Researchers have been very active for the last two decades in studying their effectiveness and their impacts on drivers behavior [10]. For instance, Ahmed et al. [11] studied their performance when used to lower drivers speed in work zones in the United Arab Emirates. A deployment of DMSs in Sweden shows its effectiveness in reducing speed [12]. Chaurand et al. [13] investigated the efficiency of DMSs in running safety campaigns by displaying short messages about speed limits in highways.

Hassan et al. [14] studied the effectiveness of DMSs in convincing drivers to reduce their speed in reduced visibility conditions (e.g. due to weather conditions).

However, one of the current most popular usage of DMSs is still Route Guidance. Indeed, when used for route guidance, DMS would suggest alternative routes to drivers depending on the traffic conditions. The aim is to influence drivers route choice and convince them to avoid heading towards congested roads [15]. In order to redirect traffic flows toward less crowded routes, three main aspects should be addressed. First, travel time of selected route portions have to be calculated/estimated. Second, the traffic delay time should be displayed in the DMS in a way that drivers can easily see and understand. Indeed, the main challenge is to allow drivers, while driving, to correctly comprehend the displayed information within short exposure time (up to 6 seconds when driving highway speeds). For instance, showing travel time without proper indication may confuse drivers as they may not be able to see whether this travel time corresponds to the total travel time or to the delay time [16]. Using red color to highlight congested roads may also help drivers understand the situation much easier [16]. Third, displayed information should have an impact on drivers behavior, which means traffic information should be presented in a way that can persuade drivers to change their itinerary to avoid congested nodes.

Wardman et al. [17] proved that when showing delay time in minutes, drivers were more likely apt to divert. Shalloe et al. [18] demonstrated that using pictogram illustrating the nature of the delays (general congestion, crash or road works) can persuade drivers to change their routes in the UK. More recently, a study conducted in Qatar [19] showed that using more visual panels such as the Graphical Route Information Panel has a better influence on drivers to persuade them to divert.

Regarding the first aspect (calculating/estimating the travel time), a relatively scare literature can be found. For instance, Wang et al. [20] used traffic stream directions in order to estimate the travel time. Other research efforts proposed travel time estimation models based on traffic lights parameters, queue forming, types of vehicle detectors, or type of measurement data [21, 22, 23]. Ziolkowski and Dziejma [24] work is one of the most related to our present paper. The authors placed a VMS (or DMS) at each intersection which links a highly populated district to the city center. DMSs display estimated travel time to incoming drivers, which may allow drivers to divert in case of noticeable congestion. Travel time is calculated during real time at each road section and based on ANPR (automatic number plate recognition) cameras.

In this paper, we are focusing on the travel time calculation/estimation challenge. While most methods are centralized, we use a more distributed approach. Moreover, while most solutions rely on collecting data from sensors and cameras, we plan to use a more affordable solution which relies on real-time traffic data provided by reputed traffic applications.

3 PROPOSED APPROACH

3.1 Motivating Scenario and Hypotheses

To illustrate our approach, we consider the following hypotheses.

- Not all Dynamic Message Signs (DMSs) are within the Internet connection range.
- Adjacent DMSs can exchange data between each other (e.g. using Long Range (LoRa) communication).
- Authorities are concerned about the cost of DMSs and aim at minimizing expenses.

To address these three constraints, we consider two types of DMSs: Slave DMS (passive DMS that only displays alerts and is not connected to the Internet) and Master DMS (connected to the Internet and having processing capabilities).

For the sake of simplicity, we consider the following configuration. A main road has three exits. Few hundred meters before each exit, DMS is placed to alert drivers about potential ahead congestions. The Master DMS is placed between two Slave DMSs. If a traffic congestion occurs as depicted by Figure 1, the Master DMS will detect it by using the real-time traffic data that it gets from the cloud. Then, the Master DMS will have to do the following:

- Display an alert to warn drivers who are within the visual range of the master DMS.
- Send a notification to its Slave DMSs so that they can, in their turn, display an alert to the drivers.
- Periodically update the alerts based on newly received traffic data.

In the particular case shown by Figure 1 (the congestion is after Slave DMS2), Master DMS would display an alert "Junction3 +10 min", which means that drivers may face a delay of 10 minutes if they stay on the main road and head towards Junction3. A driver who would see this alert may then choose between ignoring the alert (e.g. the driver is fine with 10 minutes of delay) or taking one of the upcoming exits (before Junction3). Since the detected congestion is located after exit3, drivers who can see Slave DMS2 can still take exit3 to avoid the congested road. Drivers who are still near the Slave DMS1, can take one of the 3 exits. Consequently, the Master DMS will have to send the alert to both slaves DMS1 and DMS2. However, if the congestion was localized between exit2 and exit3, drivers near Slave DMS2 would not be able to avoid the congestion as no exit is available. Nevertheless, the alert will give them an indication on the expected delay.

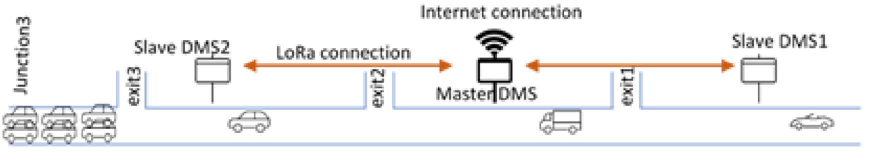


Figure 1. Motivating scenario

3.2 Graph Model

In order to organize the cooperation between DMSs, we have to model the network of DMSs and road exits. Both are important since DMS has to alert drivers before a given exit. We first consider a local network composed of one Master DMS and its Slaves. We model this network with a directed graph where nodes are exits and edges are DMSs. The graph of Figure 2 is interpreted as follows: a vehicle travelling on the road between exit E1 and exit E2 may be warned by DMS1 about an upcoming congestion. The vehicle can then take exit E2 to avoid the congestion.

Figure 2. Graph Model of exits and DMSs

In general, a Master DMS is controlling n slaves placed before the master and m slaves placed after the master. For the sake of simplicity, we take $n = m = 1$ as depicted in Figure 2. The Master DMS and its two slaves can alert vehicles which are travelling between both edge nodes ExitStart and ExitEnd (see Figure 2). However, a vehicle alerted by one of these 3 DMSs can either take exit1, exit2, exit3, or exitEnd, but not exit Start. Indeed, another DMS should be placed before exitStart to alert vehicles and "redirect" them to take exitStart. At the opposite, slave DMS2 can alert vehicles and "redirect" them to take exitEnd since no DMS is placed between exit3 and exitEnd. In other words, Master DMS has to monitor traffic between exit1 and exitEnd. We call this grouping (Master + both slaves + exit1 + exit2 + exit3 + exitEnd), a Cluster. The cluster of Figure 3 can then be represented by the graph (named CL-graph) of Figure 4. Although exitStart is represented in Figure 4, it actually belongs to another adjacent cluster.

Congestion between each two consecutive nodes of the CL-graph is measured by Traffic Delay (TD). The TD is often expressed in minutes and represents the average time delay that a vehicle may experience in comparison to the time it would spend under normal traffic conditions. Let $TD(E_i, E_j, t)$ represent the traffic delay that a vehicle may experience when travelling from E_i to E_j and measured at time t (as the traffic delay may change over time). In the CL-graph we thus add to each node E_i the value of $TD(E_i, E_j, t)$ where E_j is the successor node. At the edge, $TD(\text{exitEnd}, \text{adjCluster}, t)$ represents the delay after exitEnd (delay on the road section which comes after exitEnd and which is monitored by another Master DMS).

3.3 Congestion Monitoring

Each master DMS is in charge of monitoring traffic in its cluster. The CL-graph indicates to the Master which sections should be monitored and in which order. We

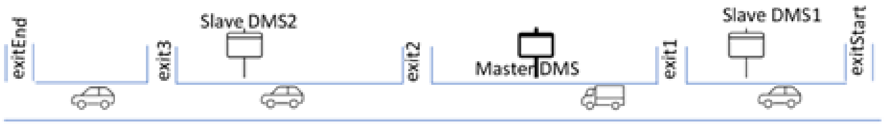


Figure 3. Sample cluster

Figure 4. Sample cluster graph (CL-graph)

suppose that the Master DMS can get, at any time, the Travel Time $TT(E_i, E_j, t)$ where E_i and E_j are two consecutive exits of the same cluster (managed by the master DMS) and t is the timestamp when the TT value was received. We also suppose that the Master DMS knows the normal travel time $TT\text{norm}(E_i, E_j)$ from E_i to E_j. In Section 4, we explain how the Master DMS can calculate the TTnorm. Consequently, Master DMS can deduce the traffic delay at each of its cluster's mode and each instant t, as follows.

$$TD(E_i, E_j, t) = TT(E_i, E_j, t) - TT\text{norm}(E_i, E_j). \tag{1}$$

3.4 Travel Delay Dissemination

Periodically (e.g. each 1 minute), Master DMS updates the travel delays associated to each node of its cluster. We suppose also that it gets the TD of the exitEnd (the edge of its cluster) from the adjacent cluster (see next subsection). To disseminate the travel delays among the DMS within its cluster, Master DMS runs the algorithms of Figure 5 that we explain in what follows.

Definitions. Let L be a list of triplets (E, td, dms) where E is the exit that precedes a road section on which there is a travel delay td. The parameter dms represents the dynamic message sign which would display the delay td to alert drivers heading to exit E. For instance, $(E_3, 10, \text{DMS2})$ means that DMS2 is displaying a Travel Delay of 10 min on the road portion after exit E_3, and drivers may thus quit the main road at exit E_3 to avoid the congestion. Predecessor(E) returns the exist preceding E which belongs to the same cluster. Successor(E) returns the exit after E which belongs to the same cluster. In Figure 4, Predecessor$(E_2) = E_1$ while Successor$(E_2) = E_3$. IncidentEdge(E) returns the DMS which leads to the exit E. For instance, in Figure 4, IncidentEdge$(E_2) =$ Master.

Travel Delay Dissemination Algorithm. The idea of the algorithm, depicted in Figure 5, is to browse the nodes of the C-graph starting from the end node and update the travel delay of each node based on the travel delays of successor nodes. The algorithm also calculates the total travel delay (cluster_travel_delay) of the entire cluster which has to be shared with the predecessor adjacent cluster.

```
E = exitEnd // we start browsing the graph from exitEnd backward
cluster_travel_delay = 0 // initialise the total travel delay of the entire cluster
While E != exitStart
        current_td=get_TD(E) // read the travel delay of the current node
        if E != exitEnd
                cluster_travel_delay += current_td
        current_dms= IncidentEdge(E) // get the DMS located before exit E
        addToList(L, E, 0, current_dms) // add triplet (E, 0, current_dms) to list L
        For each element (e, td, dms) of L // this includes the triplet we just added
                td += current_td
                send_msg(current_dms, e, td) // ask current_dms to display a travel delay td at exit e
        end For
        E = Predecessor(E) // move to the previous exit in the graph
End while
Send the cluster's travel delay (cluster_travel_delay) to adjacent cluster (at exitStart)
```

Figure 5. Algorithm run by Master DMS to disseminate travel delays within its cluster

If Master DMS runs the algorithm on the cluster of Figure 6, the output (messages sent to DMSs) of the while loop iterations would be as follows.

- Iteration 1 (E = exitEnd): send_msg(N/A, exitEnd, 5). As no DMS is directly associaed to this node, the message will not be sent to any DMS (first parameter is "N/A").
- Iteration 2 ($E = E_3$): send_msg(DMS2, exitEnd, 15); send_msg(DMS2, E_3, 10). This means that upon reception of these two messages, DMS2 will alert drivers that there is a 10min delay after exist E_3 and a 15 min delay after exitEnd.
- Iteration 3 ($E = E_2$): send_msg(Master, exitEnd, 16); send_msg(Master, E_3, 11); send_msg(Master, E_2, 1).
- Iteration 4 ($E = E_1$): send_msg(DMS1, exitEnd, 24); send_msg(DMS1, E_3, 19); send_msg(DMS1, E_2, 9); send_msg(DMS1, E_1, 8).

The cluster_travel_delay which will be shared with the adjacent cluster is 19, which means that a vehicle which would travel along the entire cluster, would be delayed by an average of 19 minutes comparing to the normal traffic (when no congestion).

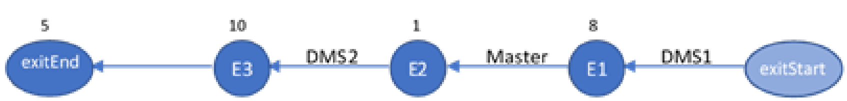


Figure 6. Example of cluster graph with numerical values of travel delays

3.5 One-Hop Inter-Cluster Collaboration

As mentioned in Section 3.4, the Master DMS starts calculating the travel delays within its cluster using the initial delay at the exitEnd node. The delay at exitEnd is actually the cluster_travel_delay of the exitStart of the adjacent cluster. As shown in Figure 7, each two adjacent clusters are sharing the same node. For instance, the exitEnd of Cluster B is actually the exitStart of Cluster C and the exitStart of B is the exitEnd of Cluster A. In other words, Cluster B uses cluster_travel_delay provided by Cluster C, calculates and disseminates the travel delays within its nodes and then passes its own cluster_travel_delay to Cluster A. As depicted in the algorithm of Figure 5, the cluster_travel_delay relies on the cumulative delays of the nodes of that cluster and does not include the delay of the next cluster (at exitEnd). This means that travel delay is at maximum transferred to one adjacent cluster.

In other words, Cluster A (see Figure 7) is only affected by the delay occurring inside Cluster B and not by the delay which may happen inside Cluster C. The hypothesis of considering a one-hop intercluster collaboration is actually justifiable. First, it simplifies the whole mechanism. Second, and since a cluster is spread over few kilometers, providing travel delays based on two clusters is already very helpful for the drivers. If more clusters are considered, for instance five, the overall travel delay (calculated by accumulating delays of 5 clusters) may not be relevant anymore by the time a vehicle reaches the second or third cluster as the situation may dramatically change over this relatively long period of time. Third, even if a big travel delay is occurring in Cluster C (see Figure 7), a vehicle starting at Cluster A will know about it once it reaches Cluster B and still can take one of the exits of Cluster B to avoid the congestion at Cluster C.

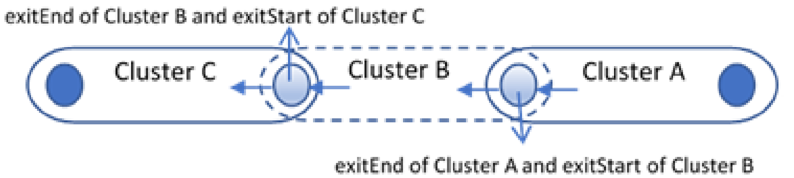
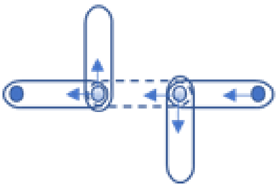


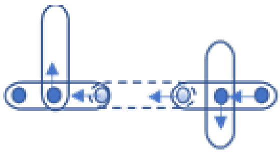
Figure 7. Sample of three adjacent clusters

In this paper, we address the simplest case of clusters pattern where all clusters are heading to the same direction and intersect at the edges (see Figure 7). Other road sections starting at the junctions are not represented by clusters. This may correspond to road sections which are not monitored by DMSs. This simple pattern (see Figure 7) actually corresponds to the scenario where a main long road/highway, which can get congested at any point, has to be monitored. When a traffic congestion is detected, drivers are then alerted. Drivers can then take any of the available exits to avoid being stuck but also in order not to make the congestion even worse.

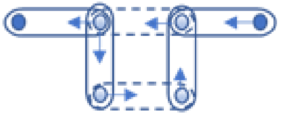
Our cluster-based approach can be extended to support more complex patterns such as cases where more than two clusters are intersecting at the edge (see Figure 8 a)), clusters are intersecting at a node different than the edge (see Figure 8 b)), or clusters are forming a cycle (see Figure 8 c)). These patterns will be discussed in a future publication.



a) More than 2 Clusters intersecting at the edges



b) Clusters intersecting at the non-edge nodes



c) Clusters forming a cycle

Figure 8.

4 PROOF OF CONCEPT

In this section, we first present the software module which is used by the Master DMS to collect real-time travel times and detect congestions. We then describe the hardware prototype of the DMS.

4.1 Real-Time Detection of Congestion

As shown in Figure 9, the Master DMS works as follows. First, the Main Processing Module requests from the Google Traffic API the travel time between two given nodes A and B at time t. A and B are represented by the spatial coordinates of their corresponding nodes. Let us suppose that Google API replies with a travel time of 10 minutes.

The Main Processing Module has then to interpret this time and decide whether it corresponds to a normal traffic or to a congestion. To do so, it relies on the Image Recognition Module (see Figure 9) which runs an automatic script to open a web page with Google Maps centered on the section delimited by A and B. Image Recognition Module captures the traffic data as a PNG image and stores it in a local database (not represented in Figure 9). This image is then analyzed by detecting the color of the pixels forming the section AB. Since Google Maps uses four colors (green, yellow, red, crimson) which indicate the traffic intensity, the script detects the number of pixels of each color along the section AB and calculates the level of congestion as a percentage (cong %). While green pixels have a weight of zero in the calculation of the cong %, red pixels have higher weight than yellow and crimson pixels have higher weight than all other pixels. Obviously, if all pixels are green, cong % is equal to 0 % (fluid traffic) and the collected travel time (here, 10 minutes) will correspond to the normal travel time TTnorm. If all pixels are crimson, the cong % is 100 % and then the 10 minutes would correspond to an "extreme congested traffic". If the section has a mixture of pixels from different colors, cong % will have a value between 0 % and 100 %. Once it gets the cong %(A, B, t), the Main Processing Module stores this cong % along with the corresponding travel time TT(A, B, t) in a local database. The stored entries are then used to identify the normal travel time TTnorm (corresponding to null cong %) and then to deduce the Travel Delay as defined by formula (1) (see Section 3.3). The Main Processing Module runs then the algorithm Travel Delay Dissemination (see Figure 5) in order to communicate alerts to the concerned Slave DMSs. This entire process is repeated periodically (ideally, each few minutes) and alerts are updated accordingly.

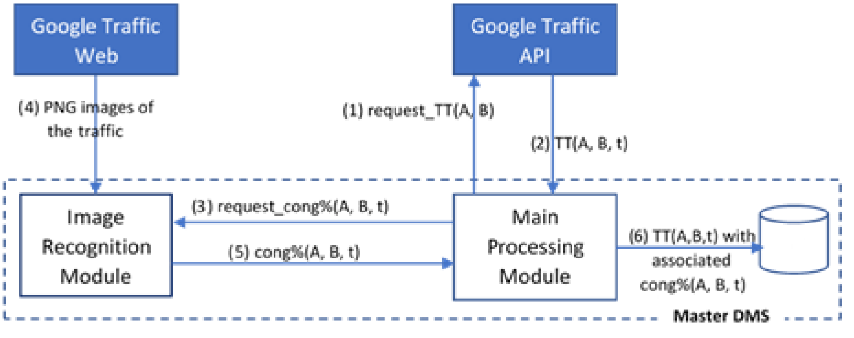


Figure 9. Architecture of the software prototype

4.2 Hardware Prototype

As a proof of concept, a small Cluster has been built (see Figure 11) including one Master DMS and one Slave DMS. Figure 10 depicts the general architecture of the prototype. A Raspberry Pi 4B is used as a processing unit for each DMS and communication between DMSs is controlled by a LoRa module. A screen is used to simulate the display panel of the real DMS. The Master DMS is connected via HTTP protocol to Google Maps server and more specifically to Google Traffic API. Master DMS is also pulling images from Google Maps (containing traffic colors of the concerned area) and storing them in an internal database. These images are used to help the Master DMS interpret the Travel Times collected from Google Traffic API, as explained in Section 4.1.

5 SIMULATION AND RESULTS

We applied our approach on a portion of one road (see Figure 12) in Muscat, Oman. This road section fits well with the cluster pattern presented in this paper. We consider a simple cluster which is not linked to any other cluster. The 2 km road

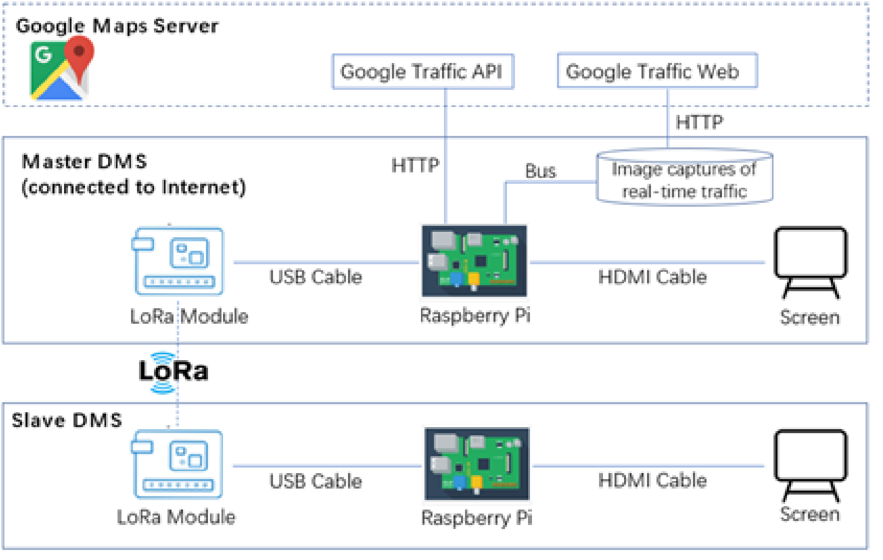


Figure 10. Architecture of the hardware prototype

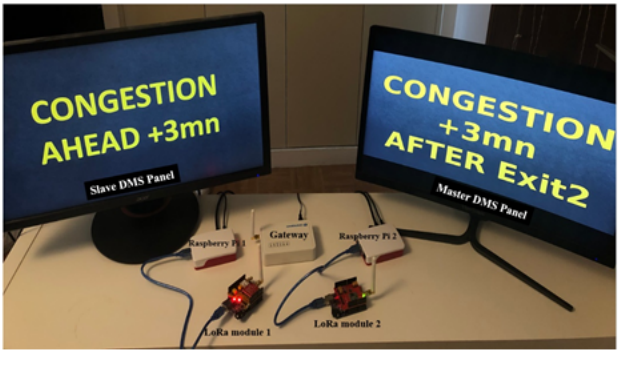


Figure 11. The hardware prototype

portion is part of a main road that crosses one of the busiest districts in Muscat. While most of the day, traffic is smooth, it can get congested at the roundabout near the point C (see Figure 12) at any time as the road ends at a main intersection. While congestions may cause long delays, the considered road section has exits which lead drivers to alternative roads. Knowing that a congestion is ahead, a driver may opt for taking one of these exits (e.g. exit 2 or 3 in Figure 12) to avoid the congested junction.

Figure 12. Studied road portion

To monitor the road section, a Master DMS was placed at point A and a Slave DMS at the point B (see Figure 12). In particular, the Master DMS was periodically checking the travel time between B and C as this section is prone to congestions.

We present here the results collected on 18th of May 2021 between 5.10 pm and 6.20 pm. Master DMS collected data each 1 minute, but for the sake of simplicity, we only present here data at key moments (see Figure 13). Initially, at 5.10 pm, the travel time collected from Google Traffic API is 2 minutes and the corresponding congestion percentage cong % is 0 % (entire section is green). Consequently, the TTnorm is set to 2 minutes and the Travel Delay TD(B, C, 5.10 pm) is set to zero. Slave DMS, upon reception of this null delay, displays "Fluid Traffic Ahead". Then at 5.29 pm, congestion starts and a one-minute delay is detected. The congestion reaches its pick at 5.55 pm where a delay TD(B, C, 5.55 pm) of 3 minutes is communicated to the Slave. Once alerted, the Slave DMS displays "Congestion +3 min after Exit2" while the Master will display "Congestion ahead +3 min" (see Figure 11). It is worth mentioning here that the calculated cong % is 25 %, which can be visually verified in Figure 13 as approximatively a quarter of the road section is congested. This gives an extra indication about how bad or good the traffic is. Although this information is currently stored in the database without being really used, we plan, in our future work, to utilize it in order to give a better estimation of the traffic.

Figure 13. Master DMS output over time

6 CONCLUSION AND FUTURE WORKS

In this paper, we propose the usage of Dynamic Message Signs (DMSs) which can collect real-time data from traffic applications and alert drivers who are heading towards congestions. Each group of DMSs (cluster) is controlled by the Master DMS which is in charge of detecting potential congestions within its cluster and disseminate the alerts among its Slave DMSs.

A hardware proof-of-concept of the DMS (both Master and Slave) has been developed. A software module has been added to allow DMSs to detect congestions. The entire prototype has been tested with real-time traffic data of a road section in Muscat, Oman. Our preliminary experiments show that the concept is working within one cluster. The solution is also affordable. Indeed, if the master DMS makes in average one call per minute to the traffic API, the cost will be around 4 $ per day (as per August 2021).

In the present paper, we address the simplest case of clusters pattern where all clusters are heading to the same direction and intersect at the edges. We are currently working on extending our cluster-based approach to support more complex patterns such as cases where more than two clusters are intersecting at the edge, clusters are intersecting at a non-edge node, or clusters are forming a cycle. Making clusters collaborate while taking into account all different patterns will allow the implementation of the idea on a wider traffic network.

Finally, and as mentioned in the Simulation section, we plan to utilize the stored data about congestion percentages and its corresponding travel delays in order to better assess the traffic. This data can also be analyzed to find congestion patterns.

REFERENCES

[1] Kuhn, B.—Balke, K.—Wood, N.: Active Traffic Management (ATM) Implementation and Operations Guide. Report No. FHWA-HOP-17-056, Federal Highway Administration (FHWA), Washington, DC, USA, 2018.

[2] Shaaban, K.—Khan, M. A.—Hamila, R.: Literature Review of Advancements in Adaptive Ramp Metering. Procedia Computer Sciences, Vol. 83, 2016, pp. 203–211, doi: 10.1016/j.procs.2016.04.117.

[3] Dowling, R. G.—Elias, A.: Active Traffic Management for Arterials. Transportation Research Board, 2013, doi: 10.17226/22537.

[4] Kim, K.—Kwon, M.—Park, J.—Eun, Y.: Dynamic Vehicular Route Guidance Using Traffic Prediction Information. Mobile Information Systems, Vol. 2016, 2016, Art. No. 3727865, doi: 10.1155/2016/3727865.

[5] Isa, N.—Mohamed, A.—Yusoff, M.: Implementation of Dynamic Traffic Routing for Traffic Congestion: A Review. In: Berry, M., Mohamed, A., Yap, B. (Eds.): Soft Computing in Data Science (SCDS 2015). Springer, Singapore, Communications in Computer and Information Science, Vol. 545, 2015, pp. 174–186, doi: 10.1007/978-981-287-936-3_17.

[6] Shi, W.—Wu, J.—Zhou, S.—Zhang, L.—Tang, Z.—Yin, Y.—Kuang, L.—Wu, Z.: Variable Message Sign and Dynamic Regional Traffic Guidance. IEEE Intelligent Transportation Systems Magazine, Vol. 1, 2009, No. 3, pp. 15–21, doi: 10.1109/MITS.2009.934642.

[7] Erke, A.—Sagberg, F.—Hagman, R.: Effects of Route Guidance Variable Message Signs (VMS) on Driver Behaviour. Transportation Research Part F: Traffic Psychology and Behaviour, Vol. 10, 2007, No. 6, 2007, pp. 447–457, doi: 10.1016/j.trf.2007.03.003.

[8] Esawey, M. E.—Walker, S.—Sowers, C.—Sengupta, J.: Safety Assessment of the Integration of Road Weather Information Systems and Variable Message Signs

in British Columbia. Transportation Research Record: Journal of the Transportation Research Board, Vol. 2673, 2019, pp. 305–313, doi: 10.1177/0361198119840335.

[9] Xu, W.—Zhao, X.—Chen, Y.—Bian, Y.—Li, H.: Research on the Relationship Between Dynamic Message Sign Control Strategies and Driving Safety in Freeway Work Zones. Journal of Advanced Transportation, Vol. 2018, 2018, Art. No. 9593084, doi: 10.1155/2018/9593084.

[10] Romero, F.—Gomez, J.—Rangel, T.—Jurado-Piña, R.—Vassallo, J. M.: The Influence of Variable Message Signs on En-Route Diversion Between a Toll Highway and a Free Competing Alternative. Transportation, Vol. 47, 2020, No. 4, pp. 1665–1687, doi: 10.1007/s11116-019-09976-8.

[11] Ahmed, K.—Al-Zoubi, K.—Siddiqui, M. A.—Anas, M.: Evaluation of the Effectiveness of Portable Variable Message Signs in Work Zones in United Arab Emirates. IET Intelligent Transport Systems, Vol. 10, 2016, No. 2, pp. 114–121, doi: 10.1049/iet-its.2015.0007.

[12] Karlberg, N. O.: Road, Weather, Action. Traffic Technology International, AutoIntermediates Limited, Surrey, UK, 2003, pp. 90–92.

[13] Chaurand, N.—Bossart, F.—Delhomme, P.: A Naturalistic Study of the Impact of Message Framing on Highway Speeding. Transportation Research Part F: Traffic Psychology and Behaviour, Vol. 35, 2015, pp. 37–44, doi: 10.1016/j.trf.2015.09.001.

[14] Hassan, H. M.—Abdel-Aty, M. A.—Choi, K.—Algadhi S. A.: Driver Behavior and Preferences for Changeable Message Signs and Variable Speed Limits in Reduced Visibility Conditions. Journal of Intelligent Transportation Systems, Vol. 16, 2012, pp. 132–146, doi: 10.1080/15472450.2012.691842.

[15] Spyropoulou, I.—Antoniou, C.: Determinants of Driver Response to Variable Message Sign Information in Athens. IET Intelligent Transport Systems, Vol. 9, 2015, pp. 453–466, doi: 10.1049/iet-its.2014.0053.

[16] Richards, A.—Mcdonald, M.—Fisher, G.—Brackstone, M.: Investigation of Driver Comprehension of Traffic Information on Graphical Congestion Display Panels Using a Driving Simulator. European Journal of Transport and Infrastructure Research, Vol. 4, 2004, No. 4, pp. 417–435.

[17] Wardman, M.—Bonsall, P. W.—Shires, J. D.: Stated Preference Analysis of Driver Route Choice Reaction to Variable Message Sign Information. ITS Working Paper 475, University of Leeds, 1996.

[18] Shalloe, S.—Sharples, S. C.—Burnett, G.—Crundall, D.—Meekums, R.—Morris, D.: Developing a Graphical Route Information Panel (GRIP) for Use on the UK Motorway Network. The First Steps. Transportation Research Part F: Traffic Psychology and Behaviour, Vol. 27, 2014, p. 133–149, doi: 10.1016/j.trf.2014.10.001.

[19] Reinolsmann, N.—Alhajyaseen, W.—Brijs, T.—Pirdavani, A.—Ross, V.—Hussain, Q.—Brijs, K.: Dynamic Travel Information Strategies in Advance Traveler Information Systems and Their Effect on Route Choices Along Highways. Procedia Computer Science, Vol. 170, 2020, pp. 289–296, doi: 10.1016/j.procs.2020.03.042.

[20] Wang, D.—Fu, F.—Luo, X.—Jin, S.—Ma, D.: Travel Time Estimation Method

for Urban Road Based on Traffic Stream Directions. Transportmetrica A Transport Science, Vol. 12, 2016, pp. 479–503, doi: 10.1080/23249935.2016.1151964.

[21] Elhenawy, M. H.—Chen, H.—Rakha, H. A.: Dynamic Travel Time Prediction Using Data Clustering and Genetic Programming. Transportation Research Part C: Emerging Technologies, Vol. 42, 2014, pp. 82–98, doi: 10.1016/j.trc.2014.02.016.

[22] Jamshidnejad, A.—De Schutter, B.: Estimation of the Generalised Average Traffic Speed Based on Microscopic Measurements. Transportmetrica A Transport Science, Vol. 11, 2015, pp. 525–546, doi: 10.1080/23249935.2015.1026957.

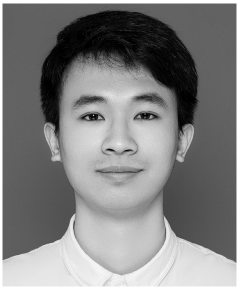
[23] Shang, H.—Huang, H.—Gao, Z.: Impacts of Variable Message Signs on Traffic Congestion. Science in China Series E: Technological Sciences, Vol. 52, 2009, Art. No. 477, doi: 10.1007/s11431-008-0229-z.

[24] Ziółkowski, R.—Dziejma, Z.: Investigations of the Dynamic Travel Time Information Impact on Drivers' Route Choice in an Urban Area – A Case Study Based on the City of Bialystok. Energies, Vol. 14, 2021, No. 6, Art. No. 1645, doi: 10.3390/en14061645.

[25] Zhao, W.—Ma, Z.—Yang, K.—Huang, H.—Monsuur, F.—Lee, J.: Impacts of Variable Message Signs on En-Route Route Choice Behavior. Transportation Research Part A: Policy and Practice, Vol. 139, 2020, pp. 335–349, doi: 10.1016/j.tra.2020.07.015.

Nabil Sahli holds his Engineer diploma (1999), his Master degree (2001), and his Ph.D. degree (2006) in computer science from the Laval University (Canada). He is currently Head of Department of Computer Science in the German University of Technology in Oman. He received several grants for research projects about agent-based simulation, e-government, sensor networks, trust models, and traffic management. His current research interests include multiagent systems, machine learning, and smart transportation.

Wassim Trojet received his Ph.D. and Master degree in computer science from the University of Aix-Marseille, Marseilles, France. His primary research focused on the integration of formal methods (mathematic based techniques) in discrete event simulation models in order to improve their credibility and consistency. This integration has contributed to the development of safer system design and more secure applications. He is currently interested in the use of machine learning and blockchain for improving trust models and context-aware security solutions. He is focusing on infrastructure-centric and collaborative IoT devices approaches. He is considering the human factor while developing these approaches. Several research programs have funded his research activities.



Zhehan Zhang received his Engineer diploma and Master degree in embedded systems from ESIGELEC, France in 2022. Currently he is Software Engineer of steering system in Bosch HUAYU Shanghai, China. His main research interests include wireless communication and deep learning.

Nesrine Ouled Abdallah is currently Senior Lecturer at the Computer Science Department at the German University of Technology in Oman (GUtech). She holds her Engineer Diploma (2007) and Master of Science degree (2010) in computer science from the University of Sfax in Tunisia. She also holds her Ph.D. degree (2017) in computer science from the University of Bordeaux in France and from the University of Sfax. She worked as Assistant Professor and researcher in Tunisia and in France. Her current research interests include distributed algorithms, AI and IoT.