ADAPTIVE TRAFFIC LIGHT CYCLE TIME CONTROLLER USING MICROCONTROLLERS AND CROWDSOURCE DATA OF GOOGLE APIs FOR DEVELOPING COUNTRIES

Sumit Mishra [1], Devanjan Bhattacharya [2, *], Ankit Gupta [3], Vaibhav Raj Singh [1]

[1] Dept. of Electronics and Communication Engg., Dr.A.P.J.Abdul Kalam Technical University, G.C.E.T., India - (sumitmishra209, vaibhavrajsingh9)@gmial.com
[2] Nova Information Management School, Universidade Nova de Lisboa, Lisbon, Portugal - dbhattacharya@novaims.unl.pt
[3] Asst. Prof. of Dept. Civil engg. Indian Institute of Technology (Banaras Hindu University), Varanasi, India - Ankit.civ@iitbhu.ac.in

Commission VI, WG VI/4

KEY WORDS: Traffic light controller, Real-time traffic signaling, congestion, ZigBee communication board, Google Traffic API, Agent-based traffic modeling.

ABSTRACT:

Controlling of traffic signals optimally helps in avoiding traffic jams as vehicle volume density changes on temporally short and spatially small scales. Nowadays, due to embedded system development with the rising standards of computational technology, condense electronics boards as well as software packages, system can be developed for controlling cycle time in real time. At present, the traffic control systems in India lack intelligence and act as an open-loop control system, with no feedback or sensing network, due to the high costs involved. This paper aims to improve the traffic control system by integrating different technologies to provide intelligent feedback to the existing network with congestion status adapting to the changing traffic density patterns. The system presented in this paper aims to sense real-time traffic congestion around the traffic light using Google API crowdsource data and hence avoids infrastructure cost of sensors. Subsequently, it manipulates the signal timing by triggering and conveying information to the timer control system. Generic information processing and communication hardware system designed in this paper has been tested and found to be functional for a pilot run in real time. Both simulation and hardware trials show the transmission of required information with an average time delay of 1.2 seconds that is comparatively very small considering cycle time.

1. INTRODUCTION

The unplanned growth of metro cities contributed to the problem of traffic congestion in many developing countries. Ever increasing number of vehicles cause major problems that are growing exponentially in metropolitan cities all around the world. Problems attributed to traffic congestion are pollution, wastage of work hours, stress due to traffic jams, and fuel cost. All these problems are interlinked and increase vehicle emissions. Therefore, it is necessary to continuously manage and control congestion. The major factor involved in controlling congestion includes increasing the road infrastructure or cutting down the vehicles count. However, in today's world of globalization the space available for infrastructure is hard to find, and reducing the rising vehicle numbers is proving impossible. Therefore, demand management could maximize vehicle throughput for bottleneck conditions.

Out of the many definitions for congestion, the oft stated one is, a travel time or delay in excess of the normally incurred time under light or free-flow travel conditions. Unacceptable congestion is the travel time or delay in excess of an agreed-upon norm. The agreed-upon norm may vary by type of transportation facility, travel mode, geographic location, and time of the day. Travel time estimate is a better way to map the congestion as well as being a common congestion measurement parameter in the research community. The variance in travel time on one hand and hence, the congestion itself, directly depends upon the control of traffic light signal time which on the other hand control the flow of traffic, and hence can manage congestion. To relieve traffic congestion under the growing pressure on existing road infrastructure using congestion adaptive traffic light control can lead to an efficient demand management.

Literature outline (He, Q. et al., 2014) the fact that different cycle times with different split times, for different lanes, are needed to increase vehicle serving rate for a given road infrastructure. Here, cycle time refer to the time to complete a set of stages (set of non-conflicting phases). Taking this fact into account, as also road-infrastructure knowledge-base, different cycle times with defined split times are calculated and deployed in accordance to congestion. Traffic control is an important component of Intelligent Transportation Systems (ITS) that aims at integrating advanced communications, information, and electronic technologies into transportation infrastructure to serve the related purpose. Controlling traffic lights plays a key role in increasing traffic throughput and reducing delay. When scheduling traffic lights, current traffic conditions should be considered as they can significantly affect the control scheme (Zhou et al., 2013; Wiering, et al., 2004). The situation of metro cities in India gets so worse that, traditional traffic control system (TCS) are inefficient due to randomness in the traffic density pattern throughout the day.

* Corresponding author

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-4-W7-83-2018 | © Authors 2018. CC BY 4.0 License.

Although road traffic authorities in India try to mitigate the problem by switching different fixed time traffic signal timing on basis of the time-of-day plan, this is ineffective as traffic is generally random for the larger time period, which is taken to be on an hourly basis. Traffic congestion being a stochastic process can be mapped more accurately for the small time- period, therefore, managing traffic light will be better if time- period considered for congestion checking is small. Also, the traffic congestion build-up curve states that build-up of congestion is a short extent temporal and spatial process but once built-up it takes hours to disperse. This leads to an irregular traffic delay during transit in the urban areas (Sen, Rijurekha, et al., 2013). Due to this, not only the vehicles get trapped in the unnecessary traffic jam but also some time in the regular condition they have to wait for a long time-span even if the traffic density is very less. This problem can be resolved if the traffic signal timer (TST) of the controller can be programmed to be manipulated with the continuously varying traffic density.

For preventing the traffic congestion by ensuring demand management needs real-time traffic congestion status, in cost-effective way. Using this traffic status, cycle time control methodology can be easily deployed to control the traffic light smartly. Various methodologies can be leveraged to get the real-time congestion data, some of these methods are comparatively reliable but are more economically challenging, being resource-heavy, maintenance hungry and infrastructure intensive. Since a developing nation need not necessarily invest in high-level computing infrastructure so a simpler method is described in the present research for real-time congestion resolution. The technique involves the use of application program interfaces (APIs) provided by major companies in the field, like Google or Microsoft, for real-time traffic status of traffic congestion. Thus, for optimizing congestion it eliminates the need for initial infrastructure deployment for tracking vehicles and thus cuts down the cost involved with the distribution of the sensor network. Apart from the low-cost sensing, this work also aims at low-cost practical deployment of the proposed scheme with a light-weight application program (software) upgrade without the need of new personalized full-scale hardware change. Hence, for sending the API sensed congestion deployment time to traffic light master, ‘ZigBee’ communication boards are made and tested that can easily integrate into existing traffic light infrastructure.

2. TRAFFIC CONGESTION COLLAPSE AND CHALLENGES

Theory of traffic flow explains that traffic congestion sets in quickly and takes a long time to dissipate once the network operates above critical point for very short time viz. 5 minutes only (Elefteriadou, L., 2014; Knoop & Daamen, 2017). Traffic flow curve shows that the operational flow rate varies with traffic density. For low density, the rate increases linearly with traffic density and peaks at a maximum flow rate. After a peak, flow degrades exponentially with the increasing density to saturate with maximum “road vehicles contain capacity”.

Consider the case when the road link is operating at maximum flow for which they are designed, if a short burst of traffic enters the link and temporarily pushes the traffic density above optimized value, then the flow rate will drop below maximum. This decreased exit rate will further increase traffic density of road link. This domino effect leads to the exit rate decaying rapidly that leads to congestion collapse and jam if no provision is taken to minimize the congestion. This is the reason that once the congestion occurs, the more the time it will take to disperse. The more is the intensity of traffic burst the more will be the time needed to decongest the area. So, to avoid the jamming condition, the input rate, on one hand, is reduced by adjusting the capacity of the cycle time movement's split, in accordance with real-time congestion status, which on the other hand cut short the incoming vehicle density (Jain et al., 2012).

These jamming problems are more intense in developing countries due to unplanned cities growth. However, to set-up extensive hardware-based traffic light real-time controllers and congestion sensing infrastructure calls for significant amount of investment which can scale-up with the increasing traffic. Also, in developing countries, various other hurdles are often encountered before the funds reach such demanding initiatives. Therefore, for practical deployment in developing countries scenarios, a system with low investment is a sensible endeavour.

3. REAL-TIME TRAFFIC CONGESTION ACKNOWLEDGMENT TECHNIQUES

3.1 Wireless Sensor Network (WSN)

Sensor networks are the key to gathering the information needed by smart environment whenever needed. WSNs can be used to know the real-time traffic density around the area and help traffic light timing control system and as well as the driver to take several decisions in order to optimize arrival time and avoid traffic congestion. The sensor could be LIDAR or RADAR based sensor connected wirelessly or, induction loop and pulmonic tubes based infrastructure sending data wirelessly to a center. Sometimes wiring is also used to communicate with center. The wireless sensors with integrated sensing, computing, and wireless communication capability make them easy to install but cost factor comes in the play when deployed for large area. Currently, collecting traffic data for traffic planning and management is mostly achieved by wired sensors in developed nations around the globe. However, installation of wired sensors and its maintenance cost limits large-scale deployment for monitoring real-time data that can be used to avoid the traffic congestion and to look after the implementation of traffic rule (Pascale, A., et al., 2012).

3.2 Video Surveillance and Camera Feeds

Yet another way to monitor the traffic-related problems is video surveillance cameras (Jain et al., 2012). Much research has been done to sense the traffic congestion through video surveillance cameras using low quality as well as high-quality video feeds for both night and day. There are also some works leads to merging information sensed from WSN as well as video surveillance camera to give the more intelligent system to set focus when congestion predicted (Collotta et al., 2014). Many adaptive and fuzzy logic controller are based on surveillance cameras information collection. The major limitation of this approach is the wired communication infrastructure of surveillance camera involves considerable maintenance costs, high installation cost and reduces the architecture scalability.

3.3 Infrastructure Free Vehicular Network

Rather than hardware sensing technique, infrastructure-free vehicular networks are reported, with very low adoption of vehicle to vehicle (V2V) technology, at this time (Lu, N. et al., 2014; Hadded, Mohamed et al., 2014). Many works leverage

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-4-W7-83-2018 | © Authors 2018. CC BY 4.0 License.

Vehicular Ad-Hoc NETwork (VANET) for various applications (Al-Sultan, S. et al., 2014). Also, several algorithms have been developed to establish communication between vehicles using VANET. Some algorithms used to cache data of congested regions and pass the data to another vehicle are listed in Lakas, A., & Shaqfa, M., 2011. Due to non-prevalence of such vehicles, these algorithms might fail to work sometimes, however to some extent they have addressed the problems related to the data loss (Su, Hang, and Xi Zhang., 2007). Multichannel communications scheme to support audio/video and other data are realized using Dedicated Short Range Communications (DSRC) for V2V applications. However, as mentoned, V2V is not so popular due to the high technology needed as well as sparse crowdsource data collection and roads with limited V2V enabled vehicles, which may limit the application scenarios of the solution.

3.4 Application Program Interface (API) Acknowledgment

The vehicular topology is highly dynamic which raises some challenges. The main challenges are the high mobility of nodes, intermittent links and stringent latency requirements. So, instead of applying VANET to forward the sensed and calculated data, the sensed data using GPS and many other sensors in the mobile device are uploaded to the central server where analysis and calculation are done. After, that these processed data are made available to any client device or application in the form of APIs. Due to the advent of big data, machine learning, mobile computing and more technologies, these APIs are gaining many application usages (Ning, Zhaolong, et al., 2017).

For retrieving estimated time to travel / arrival, there is an API called Google Maps Distance Matrix API (Google Developers., 2017) that provides a matrix of origins and destinations for travel distance and time. For calculating the ETA, 'best_guess', traffic model of the API can be used. Which specifies the assumptions to use when calculating time in traffic. This setting affects the value returned in the 'duration_in_traffic' field, in the response which contains the best estimate of travel time, through historical traffic conditions and live traffic. Live traffic becomes more important in case the 'departure_time' is close to now. Predictive travel time uses historical time-of-day and day-of-week traffic data to estimate travel times at a future date. This makes it easier to predict how long it will take to get somewhere and therefore, is a promising parameter to measure congestion.

4. METHODOLOGY AND SYSTEM ARCHITECTURE

The proposed system adapts the traffic signal controller according to the estimated traffic density in near real time. After sensing the congestion and mapping it in different congestion status level, the system sends the knowledge-based data of different cycle times available for 830 intersections all over Delhi (India) on (Delhitrafficpolice.nic.in., 2017). Along with split cycle time data available for Delhi and Bangalore intersections, for congestion management, this whole system works on the problem by sending optimized cycle time to traffic control system. Therefore, it is an endeavor to tackle the congestion by adjusting demand management as it can't be eliminated because the resources, i.e. road infrastructure is limited.

The proposed method performs the congestion status check for the point of deployment i.e. from the center point of road intersection up to next consecutive traffic light in all the directions as shown in Fig.1. However, the congestion building curve (Jain, V. et al., (2012) suggests that for preventing choking, congestion must not accumulate beyond a defined level for a given infrastructure. Therefore, in Google API request, time input of 'now' is provided (the actual time of request). So that it takes care of congestion from origin to destination from 'now' to up till the destination. Data resolution provided by Google API vary for different places by a reasonable amount, say in less than a minute, and so can be considered as real time.

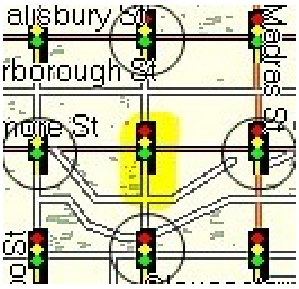


Fig.1 Highlighted traffic light under consideration is origin and Circled lights are destination

The request is-
[https://maps.googleapis.com/maps/api/distancematrix/json?origins=
latitude,longitude&destinations=latitude,longitude|latitude,longitude|latitude,longitude|latitude,longitude&departure_time=now
&key=YOUR_API_KEY].

Congestion Value (CV)	Congestion status	Type
CV ≤ (Max. hourly congestion value of last week + 2*Min. hourly congestion value of last week)/3	1	Minimum congestion
(Max. hourly congestion value of last week + 2*Min. hourly congestion value of last week)/3 < CV ≤ (2*Max. hourly congestion value of last week + Min. hourly congestion value of last week)/3	2	Medium Congestion
(2*Max. hourly congestion value of last week + Min. hourly congestion value of last week)/3 < CV	3	High Congestion

Table 1. Congestion Status Lookup Table

The Google API responds through a JSON (data type) string to this query by considering real time as well as near future traffic data grabbed by their ML and AI algorithms using extrapolating previous records. The request can be made as many times as a user likes for a paid account, however, for testing purposes a reasonable number of hits are allowed. Hence, for testing, API response is called in every two minutes which is comparable to smallest cycle time. Afterward, the status calculating algorithm defines the congestion status in three different quantized level according to the processed data grabbed by Google distance matrix API. The Google API response data provides maximum average time to travel between the origin and destination considering the congestion on the way. For a fair quantization and partitioning, three levels (1, 2 and 3) of traffic congestion status has been chosen as no or low congestion, medium congestion, and high congestion, respectively. As the different knowledge-based data of each intersection for Delhi and Bangalore is divided in 3 or 4 different cycle times according to congestion hours, therefore, 3 levels of congestion are adopted in order to avoid any quantization error. This API fetching and congestion level calculation is done in authorized traffic server centrally and the calculated cycle time is sent to local unit of the particular site for deployment.

Google API doesn't publish its live congestion status data explicitly in digital form but mapping the response of estimated time to travel can be done. For this, some road Infrastructure related parameters have to be fed once by traffic personnel at the authorize traffic center. After feeding the parameters once, they get stored unless traffic personnel, themselves decide to change.

To map API-data to congestion level, a 3-way process is followed -

1. The origin is set at the central coordinates of crossing with traffic light infrastructure, and the destination points are set to next consecutive central coordinates of crossing with traffic light infrastructure, one in each direction.
2. All the Differences (Estimated times to arrival provided by Google API - Estimated time of arrival provided by road authority of each road lane) of each lane joining the intersection are added in accordance with the rationed weight factor, according to traditional traffic, of each road to calculate congestion value.
3. Comparison of calculated value is done as mentioned in table 1 with maximal and minimal congestion value of the same hour from the previous week, in order to find congestion level.

Status calculating algorithm intakes the weight factor of each road, in form of approximate percentage proportional to the usage of linked roads, as input by the road authority personnel. Multiply the weight factor with the difference of initially grabbed estimated time to travel and general time to travel (input by the road authority personnel) of each road. Submission is compared to give congestion status.

CV = $\sum$ [(Weight factor of lane 'n') * (Difference of estimated time for lane 'n')]

Where CV is congestion value, 'N' is total number of lane connected to intersection and 'n' is lane number.

The mapping so done is linear to maintain the wide applicability however for practical usage on some very low or very high congested road intersection and can also be customized according to the location. Sometimes it will become necessary to change this threshold manually in the server by traffic agent as Google API is unnecessarily showing high congestion due to either some accident; construction work or some transport vehicle intentionally choose to stand or drive at low speed on roadside instead the road is vacant and can handle high traffic rate. This scenario of driving at low speed or stand still is generally occurs near industrial parks/areas. After the congestion status is calculated cycle time is updated accordingly as mentioned in the plan-of-day database. The congestion value calculation is so done in order to take the average effect of the whole area instead of the fact that the timing could be completely different based on the congestion level in each outgoing direction at the intersection. For considering this fact congestion weight is taken into account (inputted by road authority officer/agent) of each incoming road at the intersection. Mapped of the incoming as well outgoing traffic congestion has to be done by averaging on each lane so congestion level then calculated will average out the different congestion levels on the different road to give average level for selecting cycle time out of 3 levels. However, the timing allotted for the different phase of different cycle time will be the same as mentioned in the knowledge-based database of each intersection calculated by the traditional model.

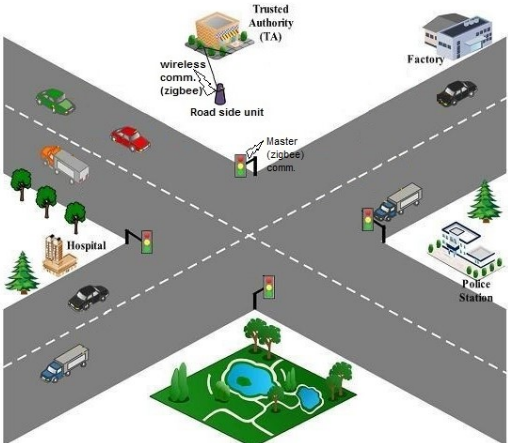


Fig.2 Purposed interconnected network

In order to update the cycle time, roadside unit (RSU) has to send the deployment time of red, green and amber for each slave of traffic control system, as shown in Fig.2. For traditional traffic light controller manufacturer in India or outside like- Envoys (India), SHENZHEN NOBLE OPTO CO. LTD (China), FORBIX SEMICON (India) etc., all provide traditional traffic light slave connected with one master panel. So, as these slaves are joined by the pre-existed master board in traditional traffic light infrastructure and therefore there is no need of full hardware change. However, traditionally, road authority personnel manually change the different cycle time status in sets of hours, which are not only based on crude manual congestion approximation but also involve human error due to lack of computations like delay in cycle time changing. So, for traditional method, there is no need of RSU but to implement the discussed system, RSU with centralized control is needed to communicate varying congestion status.

These RSUs should be simple connection module loaded with functionality to do communicating using the Internet with trusted authority to get deployment time in accordance to congestion. The traditional master board is just an embedded board loaded to communicate slave either wirelessly or through the wire. For traditional system generally, the master board has no functionality of internet connection. So, in order to communicate changed deployment time from RSU to master, wireless (ZigBee) enabled microcontroller board is manufactured. Central RSU unit is fitted with microcontroller programmed to send data using 'ZigBee'. Master is fitted with microcontroller programmed to receive using 'ZigBee'. After that master sends implementation time for each slave to them via traditional linkage, wired or wireless. The proposed methodology will absolutely save misuse of money as needed if, fully personalized traffic light machine has to be changed but here, upgrade to traditional traffic light machines is proposed so there will be no such excesses.

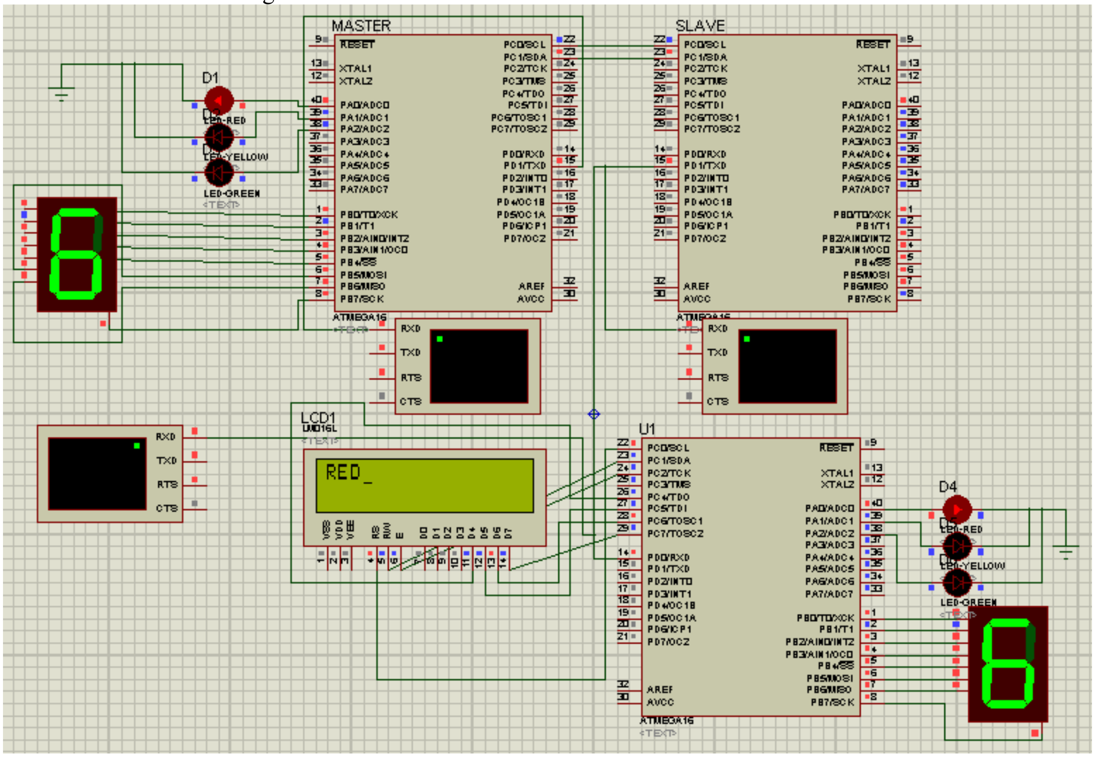
5. HARDWARE IMPLEMENTATION AND TESTING

'ZigBee' module with its protocol-set is standard to made personal small network, with the communication range of 20-100 meters in line of sight (Sparkfun.com., 2017)). Hence 'ZigBee' is opted for realizing the small communication network. For 'ZigBee' sender and receiver realization, first testing using 'Proteus Design Suite' is done. These 'ZigBee' sender and receiver boards communicate with RSU as well as master board using 'I^2C' (Inter-Integrated Circuit). 'I^2C' is one of the most common and basic embedded onboard connection protocol and, nowadays, almost all embedded devices are capable to use 'I^2C'. As confirmed by the local authority, RSU that connected to trusted authority via the internet is also 'I^2C' protocol enabled. So we simulated 3 boards, one for imitating the RSU, another 2 as 'ZigBee' transmitter and sender to communicate deployment time to master board.

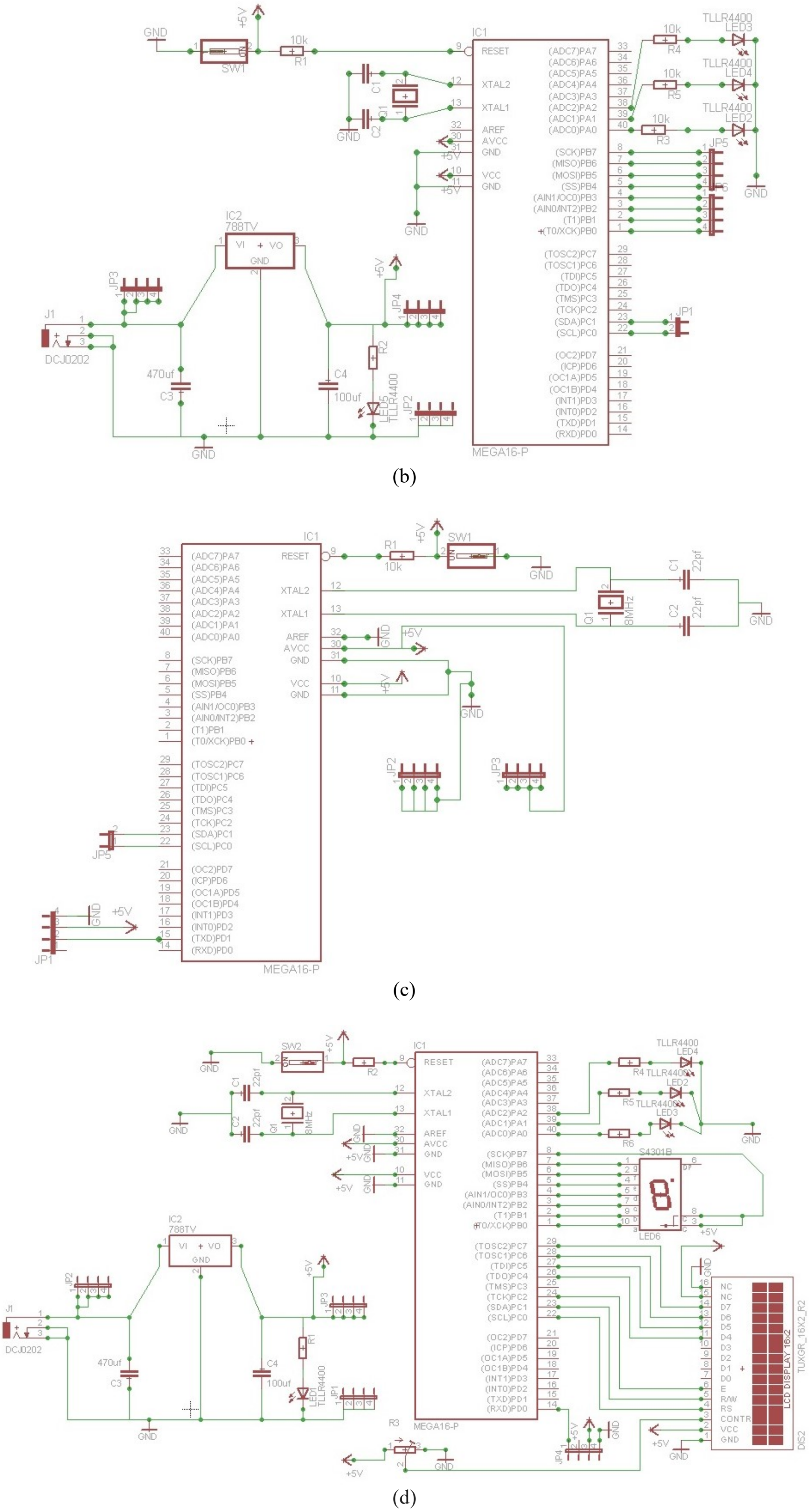
The algorithm so opted for testing is-

- Read data from board 1 (RSU).
- Transfer it to board 2 (Sender ZigBee board) using I^2C protocol.
- Board 2 transfer data to board 3 (Receiver ZigBee board) using ZigBee protocol-set.
- Receive data from board 2 using board 3.
- Display the status/data of RSU on 'LCD' screen using I^2C, i.e. pass it to Master if 'LCD' is assumed to be master.

The testing of software and hardware suit on simulation shows a delay of '1.2' second in between sending and receiving. The hardware implementation of both transmission and receiver board shows the same result. The delay is very small in comparison to any 'cycle time' and hence will not affect the system at all. Various figures of simulation and hardware are shown in Fig.3. The hardware used for PCB fabrication are - ATMEGA 16 Microcontroller, XBEE-S2 (ZigBee), Seven Segment Display, (16X2)- Liquid Crystal Display, Light Emitting Diodes – 3 (Green, Red, Yellow), 7805-IC, 8 MHz Crystal Oscillator, Capacitance (22pf, 100uf, 470uf), Resistance(10K ohms, 470 ohms), Momentary-NO-Switch, Jumpers, Female-to-Female Wires, Potentiometer. As ZigBee module communicate via I^2C with any board sender or receiver, hence for simulation purpose only, we eliminate ZigBee module and connected the different boards via 2 wire I^2C directly as presented in Fig.3 (a). The 'U1', 'SLAVE' and 'MASTER' in Fig3 (a) can be interpreted as sender, RSU and receiver board, respectively. For circuit-level understanding, schematic diagram of each board- RSU, sender and receiver board are presented in Fig.3 (b, c, and d), respectively.



(a)



(b)

(c)

(d)

Fig.3 (a) Proteus simulation (b) RSU board (imitation as done to verify our purpose only) (c) ZigBee sender board (d) ZigBee receiver board with LCD screen on-board

	RUN-1	Run-2	Run-3
MAX_Avg. ETA time (sec)	771	771	711
Avg_ETA real time (sec)	619.134	696.2759	689.6667
Avg. CV	-151.866	-74.7241	-81.3333
Level-1	9	18	10
Level-2	11	7	10
Level-3	10	4	10
Standard deviation for CV	21.1104	28.5559	33.1132
Median for CV	-148.5	-83	-76

Table 2. Congestion value and level calculation

For API data testing, data sampling for 3 hours a day, 1 hour in 8-9 AM, 1 hour at 2-3 PM and 1 hour at 8-9 PM, after every 2 minutes for 4-way intersection centered at 12.927586, 77.620986 (LAT, LONG) is done. Finally, it is repeated for next Monday, the samples collected in three software run are 30, 29, and 30 respectively. Afterward, complete analysis with congestion value calculations are done and is presented in table 2. In the morning time, even though there is less congestion as depicted by the average value for CV (Avg. CV) but the traffic condition fluctuates as depicted by the standard deviation. The congestion status calculated is in comparison to that single hour in which data is grabbed, therefore most of the time the congestion status is either 1 or 2. For the afternoon, averagely intersection is congested. Although, average congestion is more than that of the morning. In afternoon, the traffic value fluctuates more than the morning and hence the status values got scattered status for 1, 2 and 3, more inclined to 1 and 2. In evening the congestion is worst with most fluctuating congestion condition, hence for the evening, the status values are mostly scattered. This congestion value calculated is in respect to an hour and therefore give a good insight into temporal congestion values. The congestion value so grabbed for different hours, say for afternoon and evening are spread around relatively same median values for some instances and therefore there are cases that the congestion in afternoon is more than that of average evening value. Even though, traditionally evening congestion is taken to be worse. So, the adjustment in split time of cycle time is done in accordance with status value, avoid congestion forming on the short temporal basis.

6. CONCLUSION

This paper presents a method of determining junction level congestion values and using the congestion values determines the most appropriate signaling scheme at an intersection. The congestion values are determined based on journey time estimations obtained through requests to the Google Maps Distance Matrix API. These are used at a local server to determine the congestion value based on historical readings. The congestion value is mapped onto 3 status levels, and the most appropriate fixed time cycle length with defined split time for each lane is selected from a pre-existing database based on the traffic demand. Testing for congestion level on a given intersection proves that traffic congestion level is fluctuating for a given time period. The paper also presents a feasibility study for a hardware interface that supports wireless I2C communications and interfaces with existing infrastructure. The cycle time conveying hardware is tested and proves to be responsive with a minimal delay of 1.2 seconds which is comparatively very small than the cycle time. Hence, it is of great practical use, presenting a low-cost solution to the problem of intersection control in restrictive environments. The proposed control scheme readily lends itself to deployment in developing countries, adapting existing infrastructure rather than requiring an entirely new system. The idea would be equally applicable in any busy city with low infrastructure budgets and no roadside sensing capabilities.

Further, we are looking forward to work on an autonomous system to give feedback of upcoming traffic light status and countdown to the user, using the same technique. Also, as in distance matrix API, one has to provide coordinates of starting and destination points which is somewhat appropriate for small infrastructure but for large infrastructure with many links and intersections, it is not feasible so another API usage like JAVA map API can be leveraged out.

REFERENCES

Al-Sultan, S., Al-Doori, M. M., Al-Bayatti, A. H., & Zedan, H. (2014). A comprehensive survey on vehicular ad hoc network. Journal of network and computer applications, 37, 380-392.

Collotta, M., Pau, G., Scatà, G., & Campisi, T. (2014). A dynamic traffic light management system based on wireless sensor networks for the reduction of the red-light running phenomenon. *Transport and Telecommunication,15*(1), 1-11. doi:10.2478/ttj-2014-0001

Delhitrafficpolice.nic.in. (2017). Traffic Signals | Delhi Traffic Police. [online] Available at: https://delhitrafficpolice.nic.in/be-road-smart/signals/ [Accessed 9 Jul. 2017].

Elefteriadou, L. (2014). An introduction to traffic flow theory (Vol. 84). Chapter 3, New York, NY, USA: Springer.

Google Developers. (2017). Developer's Guide | Google Maps Distance Matrix API | Google Developers. [online] Available at: https://developers.google.com/maps/documentation/distance-matrix/intro [Accessed 12 Jul. 2017].

Hadded, M., Laouiti, A., Muhlethaler, P., & Saidane, L. A. (2016). An Infrastructure-Free Slot Assignment Algorithm for Reliable Broadcast of Periodic Messages in Vehicular Ad Hoc Networks. *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*. doi:10.1109/vtcfall.2016.7880903

He, Q., Head, K. L., & Ding, J. (2014). Multi-modal traffic signal control with priority, signal actuation and coordination. Transportation Research Part C: Emerging Technologies, 46, 65-82.

Jain, V., Sharma, A., & Subramanian, L. (2012). Road traffic congestion in the developing world. *Proceedings of the 2nd ACM Symposium on Computing for Development - ACM DEV 12*. doi:10.1145/2160601.2160616

Knoop, V. L., & Daamen, W. (2016). Automatic fitting procedure for the fundamental diagram. *Transportmetrica B: Transport Dynamics,5*(2), 129-144. doi:10.1080/21680566.2016.1256239

Lakas, A., & Shaqfa, M. (2011). Geocache: Sharing and Exchanging Road Traffic Information Using Peer-to-Peer Vehicular Communication. *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*. doi:10.1109/vetecs.2011.5956785

Lu, N., Cheng, N., Zhang, N., Shen, X., & Mark, J. W. (2014). Connected Vehicles: Solutions and Challenges. *IEEE Internet of Things Journal,1*(4), 289-299. doi:10.1109/jiot.2014.2327587

Ning, Z., Xia, F., Ullah, N., Kong, X., & Hu, X. (2017). Vehicular Social Networks: Enabling Smart Mobility. *IEEE Communications Magazine,55*(5), 16-55. doi:10.1109/mcom.2017.1600263

Pascale, A., Nicoli, M., Deflorio, F., Chiara, B. D., & Spagnolini, U. (2012). Wireless sensor networks for traffic management and road safety. *IET Intelligent Transport Systems,6*(1), 67. doi:10.1049/iet-its.2010.0129

Sen, R., Cross, A., Vashistha, A., Padmanabhan, V. N., Cutrell, E., & Thies, W. (2013). Accurate speed and density measurement for road traffic in India. *Proceedings of the 3rd ACM Symposium on Computing for Development - ACM DEV 13*. doi:10.1145/2442882.2442901

Sparkfun.com. (2017). XBee Buying Guide - SparkFun Electronics. [online] Available at: https://www.sparkfun.com/pages/xbee_guide [Accessed 8 Jun. 2017].

Su, H., & Zhang, X. (2007). Clustering-Based Multichannel MAC Protocols for QoS Provisionings Over Vehicular Ad Hoc Networks. *IEEE Transactions on Vehicular Technology,56*(6), 3309-3323. doi:10.1109/tvt.2007.907233

M. Wiering, et al. Intelligent Traffic Light Control. Technical Report UU-CS-2004-029, University Utrecht, 2004

Zhou, B., Cao, J., & Li, J. (2013). An adaptive traffic light control scheme and its implementation in WSN-based ITS. *International Journal on Smart Sensing and Intelligent Systems,6*(4), 1559-1581. doi:10.21307/ijssis-2017-604