Smart Traffic Systems Using Internet of Things (IoT)

Travis Warling
Division of Computer Science
University of Minnesota, Morris
Morris, Minnesota, USA 56267
warli010@morris.umn.edu

ABSTRACT

Traffic congestion is a major issue across the world. In India 600 billion Rupee (8,667,600,000 United State Dollars) is lost per year due to traffic congestion. Likewise, the United States wastes around 1.9 billion gallons or five days worth of fuel per year due to traffic congestion [5]. Traffic congestion is a problem that not only impacts the economies of every country that has the issue, but also every commuter on the road. With the ever increasing amount of cars entering the road ways every year, this problem will continue to increase. Through the use of Internet of Things (IoT), a smart traffic management system can be utilized to reduce traffic congestion by dynamically changing the green light times on the traffic lights in the area of traffic congestion. By having longer green light times in the direction of traffic congestion, and having shorter green light times in directions leading into congestion, the idea is that traffic congestion will be greatly reduced. In this paper we describe and compare three proposed smart traffic management systems. Each of the proposed systems require varying levels of human interaction once set up, thus making each system a viable option for areas with high levels of traffic congestion depending on the specific needs of the area.

Keywords

Internet of Things (IoT), Smart Traffic System

1. INTRODUCTION

With the increase of vehicles on the road every year, traffic congestion is becoming more and more of a pressing issue. In the United States alone, the average urban commuter is delayed due to traffic related congestion for 34 hours per year and the United States as a whole wastes 1.9 billion gallons of fuel per year [5]. In India, the average vehicle's speed on National Highway 44 between Delhi and Chennai is only 20 kmph (12.43 mph) even though the speed limit is 100 kmph (62.14 mph) [6]. With the ever increasing global population and thus the number of vehicles on the road, the global issue of traffic congestion is only going to become more of an issue. Current traffic management systems have fixed green light times for every traffic light, which does not do enough to mitigate traffic congestion and are quickly overwhelmed with large volumes of traffic. As technology continues to become more advanced, smart traffic systems using Internet of Things (IoT) may be an option to help mitigate traffic congestion. In this paper we compare three different implementations of smart traffic systems. Each system has a varying level of necessary human involvement once implemented as well as varying levels of technology needed. This allows areas with varying levels of infrastructure and available funds to be able implement a system that works best for the area while also still being able to reduce traffic congestion.

In Section 3, the proposed system suggests using a Raspberry Pi system on each traffic light to allow authorities to dynamically control green light times in the direction of traffic congestion. In Section 4, the proposed system suggests using RFID chips placed within the road surface to collect large amounts of data about traffic flow in every direction. Authorities are able to visualize where traffic congestion is taking place and allows them to adjust traffic lights accordingly. In Section 5, the proposed system suggests placing sensors in both the road surface and on cars. The sensors collect and send data to a cloud server where it is then analyzed to determine if traffic congestion is taking place. Once traffic congestion is detected, traffic lights within the area of the congestion are adjusted accordingly.

UMM CSci Senior Seminar Conference, April 2019 Morris, MN.

2. BACKGROUND

In order to understand how smart traffic systems work, it is important to understand the underlying concepts and technology required to implement these systems. Before we cover essential information related to smart traffic systems, it is important to note that each smart traffic system covered in this paper references how the system would be applied to a singular traffic light. However, a real life implementation would require implementing the system on multiple traffic lights located within the same general area. Having multiple traffic lights within the same general area fitted with a smart traffic system allows for a grid of traffic lights that can used in conjunction with each other to better reduce traffic congestion.

In this section, we first cover Internet of Things (IoT) and how it is used within smart traffic systems. IoT is important to every smart traffic system covered in this paper. We then cover different types of hardware used by the proposed smart traffic systems such as Radio-frequency identification (RFID), Raspberry Pi and General Purpose Input-Output (GPIO) board. Finally, we cover cloud servers and how they are used within smart traffic systems.

2.1 Internet of Things (IoT)

IoT is the basic concept that allows for smart traffic systems to exist. Simply put, IoT is the idea of enhancing devices by allowing them to connect to the Internet or with other devices [3]. This allows for the devices being enhanced to have more capabilities than they had without them, or to use information collected by the devices to make informed decisions remotely. By connecting devices to the Internet or to each other, real-time communication is possible. In terms of this paper, IoT allows the sensors used in sections 4 and 5 to communicate and send data to each other; IoT also allows traffic lights in Sections 3, 4, and 5 to dynamically change based on traffic congestion.

2.2 Raspberry Pi

Raspberry Pi is a system used in Section 3 to allow authorities to send signals to traffic lights to allow for dynamic changes in the timing of light cycles. The Raspberry Pi itself is located on each traffic light and allows authorities to access the trafic light remotely.

A Raspberry Pi is a single-board computer that uses the Linux operating system [8]. One of the reasons why Raspberry Pis are popular is because they cost less than other single board computers. In terms of a smart traffic system, Raspberry Pis are popular because they can be configured as a web server and allow for for easy installation and software updates just like any Linux machine [4].

2.3 General Purpose Input/Output (GPIO)

General Purpose Input/Output (GPIO) pins are a useful feature that comes with a Raspberry Pi. The pins are placed at the top edge of every Raspberry Pi. GPIO pins do not serve a specific purpose, thus allowing users to customize them to fit their needs.

GPIO pins are essentially switches. When a pin is set to high, it outputs 3.3 volts. Conversely when the pin is set to low, it does not produce any voltage. By utilizing multiple GPIO pins, users are able to program responses based on the number of pins that are sending voltage [7]. For example, we could use two GPIO pins to make an LED change color. We could program the system to make the LED green when both pins are sending a voltage, red when only one of the pins are sending voltage, and yellow when none of the pins are sending voltage. In terms of this paper, GPIO pins are used in Section 3 as a way to change how long the green light is on in a specific traffic light.

2.4 Radio-Frequency Identification (RFID)

RFID is a system that is used in Section 4 that allows data to be collected about traffic congestion. The data that is collected through the RFID system can be analyzed and traffic lights can be adjusted based on the results. This allows authorities to control traffic lights from remote locations.

RFID is a type of wireless communication that uses electromagnetic or electrostatic coupling to uniquely identify objects. RFID systems consist of a scanning antenna and transceiver, a transponder, and the RFID tag. Radio frequency waves are used to transmit signals that are picked up by the RFID tag. The RFID tag is activated and sends a wave back to the antenna. Finally, the wave sent back to the antenna is translated into data that can be used later[1].

In terms of this paper, Section 4 uses RFID to collect data about traffic congestion. Each car is fitted with an RFID tag that interacts with an antenna and transceiver placed within the road surface.

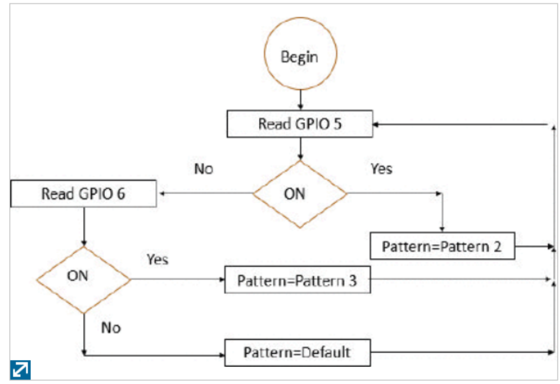

Figure 1: Flow chart of how each green light pattern is selected.

3. RASPBERRY PI BASED SYSTEM

The first proposed system of a smart traffic system suggests attaching a Raspberry Pi and a camera to every traffic light to allow for authorities to dynamically change green light times in the direction of the traffic congestion. This system is the least autonomous, and requires large amounts of human input in order for the system to work as intended. As such, this system doesn't automatically detect traffic congestion and requires authorities to know when traffic congestion is taking place before they can adjust green light times.

3.1 Raspberry Pi Implementation

The main centerpiece of the system is the use of a Raspberry Pi that is placed on every traffic light within a given intersection. The Raspberry Pi uses a wireless connection to transmit the camera feed talked about in section 3.2 to a predetermined destination. The Raspberry Pi also receives commands from authorities and uses the commands to change green light times. This allows authorities to view traffic congestion levels and change the green light duration on any of the traffic lights within a given intersection from a central location.

The Raspberry Pi is combined with the use of two GPIO pins to allow for three different green light durations. The number of GPIO pins being used is arbitrary and can be changed by authorities. When a signal is received by the Raspberry Pi from authorities, voltages are released by the GPIO pins. Figure 1 shows a flow chart of how each pattern is selected. If a voltage is being produced by the GPIO pin it is considered 'on' in the flow chart.

3.2 Adding Cameras to Traffic Lights

To allow for authorities to change traffic light times remotely, a camera is added to each traffic light. Each camera will use the Raspberry Pi's web server capabilities to take and send pictures to a folder over a period of time [4]. Authorities can access the images within the folder on a web server to observe the level of congestion in a given area [4]. The number of pictures taken and the length of time between pictures can be changed depending on what works best for the area the system is being placed in.

3.3 Practical Implementation of the System

Combining the use of a Raspberry Pi, GPIO pins and a camera on every traffic light within an intersection will allow for authorities to change the green light time in the direction of the traffic congestion to better allow for the vehicles in that direction to flow. For example, if there are large amounts of vehicles traveling north bound at a traditional four way intersection, authorities would be able to see this via the pictures taken from the camera placed on the north bound traffic light. The authorities would send a signal to the north bound traffic light telling it to change its green light time to another green light time that has been preprogrammed within the Raspberry Pi. The Raspberry Pi would receive ths signal and would convert the signal to a voltage sent through the GPIO pins. The traffic light would detect the voltage being transmitted and change its green light time to the time the voltages represent. From there the north bound traffic light would use the green light time the authorities told it to until the authorities sent another signal telling the traffic light to change.

3.4 Testing and Results

Unfortunately this system has not been formally tested yet, so there is no data to back up the idea that this would reduce traffic congestion. However, in relation to the other proposed systems in this paper that have been tested, it's not out of the realm of possibility that this system reduces traffic congestion on some level.

4. RFID BASED SYSTEM

The second proposed smart traffic system suggests utilizing RFID to monitor traffic congestion and to alert authorities when high levels of traffic congestion is detected [6]. RFID tags will interact with tranceivers within the road surface to share the tag's ID. This allows the system to collect information on how fast each car is travelling and the distance between cars. This system is more autonomous than that of the previously covered Raspberry Pi system in that traffic congestion is automatically detected. However, this system still requires human interaction as authorities are still required to change green light times of individual traffic lights within a given intersection. This system would also require installation and maintenance of RFID tags on every car driving on the road.

4.1 RFID Placement in Road Surfaces

The key aspect of this system is the implementation of RFID to collect data about traffic flow. The scanning antenna, transceiver and transponder would be placed within the surface of the road in every lane that traffic will be driving on. RFID tags would be placed on every vehicle to allow for data gathering to take place about how fast each vehicle is moving and how close vehicles are to each other. The system suggests placing sensors every 500 meters (1640.42 feet) to 1000 meters (3280.84 feet) as this will allow for accurate data to be collected [6]. In Figure 2 a basic diagram of how the system would work at one intersection is shown. In the figure, a RFID transmitter and antenna is placed within the centr of each lane of traffic and is actively sending out a signal. The vehicles driving on the road have a RFID tag within them and will communicate with the RFID transmitter and antenna as the vehicle drives over it.

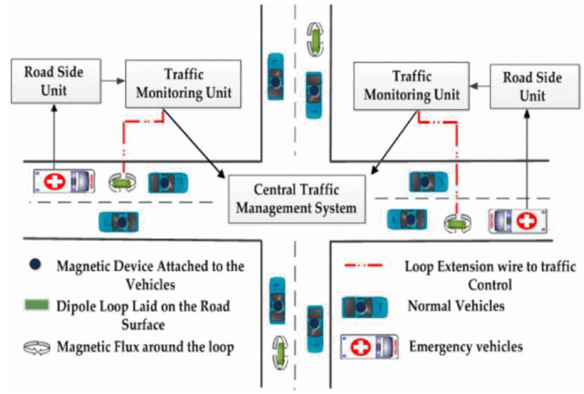

Figure 2: Placement of sensors within the road surface [6]

4.2 Calculating Traffic Congestion

Once data is collected, it is then sent to a central server to be analyzed for traffic congestion. As each vehicle n passes over the scanning antenna and transceiver, the position Y_n and speed veh_n is collected. When combining data from all of the vehicles in the system, we are able to determine the level of traffic congestion.

In order to determine the level of traffic congestion, we first pick a single car from the intersection and take its data. After taking a single car's data, we then take the data of the car directly in front of the car we are looking at, denoted $n + 1$. The main data that needs to be collected from the car directly in front is the exact time t that the car had driven over the antenna and transceiver. When subtracted from the time of car n, we are ably to know how long it has taken car n to reach the same spot that car $n + 1$ had been previously at. From there we are able to take the time between n and $n + 1$ as well as the speed veh_n of car n to determine the density den_z of vehicles in the direction the two cars were travelling in.

Once this data is gathered, the system first calculates the maximum speed at which car n can be traveling in order to not hit the car in front if they start decreasing their speed at a substantial rate. The maximum possible speed without crashing is denoted veh_{anti}. The equation takes into account the speed of the car in front veh_{n+1} as well as the distance between the two cars, denoted d_{n+1}.

$$veh_{\text{anti}} = min(veh_{n+1}, d_{n+1})$$

Once the minimum possible speed without crashing is calculated, the maximum or ideal distance D_n^{eff} between cars without causing unnecessary traffic congestion is calculated. In the equation, the distance between the two cars where they are not hitting each other even though they are extremely close, denoted $d_{/mboxsafe}$ and the distance car n is behind the car in front, denoted d_n, are used.

$$D_n^{\text{eff}} = max(d_{\text{safe}} - veh_{\text{anti}}) + d_n$$

Once the maximum speed car n and maximum or ideal distance between the cars is established, the gap g_n between the cars is considered. The system determines if the gap is increasing or decreasing using the following logic:

$$\begin{aligned} if(D_n(t+1) > d_n^{eff} - g_n AND g_n(t) > 0)\text{then}: \\ if(veh_n(t+1) > ven_n(t))\text{then}: \\ ven_n(t+1) \leftarrow veh_n(t) \\ g_n(t+1) \leftarrow max(d_n^{eff} - veh_n(t+1)Q) \end{aligned}$$

Finally, the general vehicle motion is calculated:

$$y_n(t+1) \leftarrow y_n(t) + veh_n(t+1)$$

Once all of the above equations are calculated, it is possible to form two yields: movement cycle (T_c) and aggregate movement thickness (T_d). The Movement Cycle is is the total time that it takes to complete one cycle of traffic light sequence. Aggregate movement thickness is the density of the traffic based on the calculations done in the above equations. Movement cycle and aggregate movement thickness are correlated through this equation where f is a function of the overall aggregate movement thickness:

$$T_c = f(T_d)$$

This equation states that the denser the traffic in a specific direction, the longer it should take the traffic light directing traffic in that direction to complete its sequence. At times where there is heavy traffic congestion, traffic lights in the direction of the congestion should be adjusted to have longer cycles to allow for the congested street to flow easier. Conversely if there is very little congestion, the traffic light in the direction of light traffic should be adjusted to have short cycles so that vehicles in other directions don't have to wait too long [6].

4.3 Implementation of a User Interface

Once traffic congestion is detected, it is useful to tell vehicles approaching the congested area to take a different route. By having drivers reroute before they reach the congested area, the traffic congestion should be mitigated quicker as fewer cars will be entering the area. In order to alert other drivers, the traffic congestion data is gathered and displayed on a variety of different interfaces [6]. These interfaces include mobile apps, internet browsers, and GPS systems. The data will be provided real-time to allow users to make accurate decisions.

4.4 Testing and Results

In order to test the proposed system, Rizwan et al. took data from a 12 km stretch of the German Autobahn A044 from November 4, 2010. The stretch of road consisted of a two lane highway with an entrance ramp and an exit ramp. Traffic entered the simulated highway via the entrance ramp at the same rate as that of the November 4, 2010 German Autobahn data. Vehicles also exited the highway at the same rate as the November 4, 2010 German Autobahn data, but this number was marginal enough that the number of vehicles leaving the highway didn't change the overall traffic density. Finally, the number of vehicles already travelling on the highway without entering via the entrance ramp had the same rate as the real life data as well [6].

Rizwan et al. used the data from November 4, 2010 to simulate how the system would have changed traffic flow. Rizwan et al. simulated placing RFID tags on each car, as well as placing transmitters and antennas within the road surface. The RFID system collected data and used the algorithm described in section 4.2 to determine the level of traffic congestion. Rizwan et al. simulated placing a traffic light on the entrance ramp and changed the legnth of the green light time based on the results of the algorithm. The results where then compared to the average travel time without the presence of a traffic light [6].

Rizwan et al. found that the proposed system reduced the average travel time by around 35 percent, going from an average travel time of 819 seconds, to 527 seconds. The proposed system also reduced the standard deviation from 200 seconds to 96 seconds [6]. This shows that the proposed system substantially helped reducing traffic congestion.

5. SENSOR AND SERVER BASED SYSTEM

The final proposed smart traffic system combines placing sensors in roads and cars with a cloud server and algorithm that will allow for real time traffic management. Unlike the previous two systems, this system is completely autonomous and doesn't require any human interaction once set up. This means that traffic light cycles will be changed automatically if traffic congestion is detected [2].

5.1 Placing Sensors on Roads and Cars

As with the system proposed in Section 4, Chong et al. also propose using sensors in roads and cars to detect traffic congestion. Within this system there are three ad-hoc sensor networks being used: an ad hoc on-road sensor network, an ad hoc vehicle sensor network, and an ad hoc hybrid sensor network [2]. The way each of these ad hoc sensor networks communicate can be seen in Figure 3. Ad hoc networks allow for components to connect without the use of a router or base station. This means that the individual sensors in the three sensor networks are able to connect to other sensors within the same network relatively quickly without having to connect to other systems. The on-road sensor network will consist of sensors placed within the road surface that communicate with each other. This sensor network would share data such as number of vehicles that are in the traffic lane as well as the time between vehicles. The on-vehicle sensor network will consist of sensors placed within every vehicle that communicate with each other. This sensor network would share data such as vehicle speed and distance between vehicles. Finally, the hybrid sensor network will consist of sensors placed within vehicles and within road surfaces that communicate with each other. This sensor network would share car IDs and road sensor IDs back and forth so the data can be compiled together once sent to the base station. The three networks are shown in Figure 3. When combined, the three sensor networks will gather enough data for the algorithm described in section 5.3 to determine the level of traffic congestion present [2].

5.2 Base Stations

Once data is gathered from the sensor networks, the data is automatically sent to a base station. There will be a base station placed at every intersection. Each base station will act as a wifi transmitter, and will transmit the data collected from the sensor network around the traffic light to a cloud server. Once the algorithm in the cloud server calculated the level of traffic congestion and determines how long the green light time should be for every traffic light in the intersection,

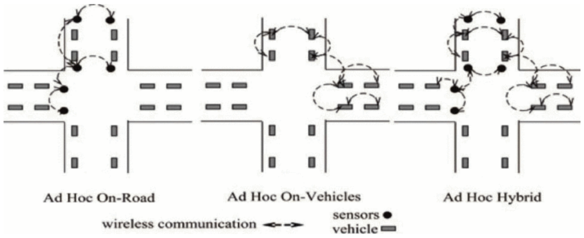

Figure 3: The three different types of ad-hoc sensor networks

the results are sent to the base station where the green light times are adjusted. [2].

5.3 Green Light Time Calculation Algorithm

Once data is transmitted to a cloud server, a green light time calculation algorithm will analyze the level of traffic congestion present and adjust the length of the green lights in the area of the congestion. The goal of the Green Light Time Calculation Algorithm is to determine if traffic congestion is present and adjust green light times in area of traffic congestion accordingly. In a given intersection, the equations that will be covered shortly will be used separately for every traffic light.

The green light time calculation algorithm will first calculate the current traffic volume (CTV) where n is the number of vehicles queued in a given direction and C is the current green light duration for a given traffic light. The equation takes into account what the traffic volume would be in a given direction over a 3600 second (one hour) period. The CTV shows how many cars, given the current traffic congestion, will be passing through the traffic light in a one hour time period.

$$CTV = \frac{(n \times 3600)}{C}$$

Using the CTV, the flow ratio (FR) can then be calculated where MFR is the maximum flow rate for the road in each direction. The FR shows the percent of traffic congestion present. A number closer to one shows higher levels of traffic congestion, while a number closer to zero shows lower levels of traffic congestion.

$$FR = \frac{CTV}{MFR}$$

Once these two ratios are calculated for every traffic light at a given intersection, the traffic lights are ordered and given priority. The traffic lights with the highest FR is deemed to have the highest priority as it has the highest amount of congestion compared to the other directions. The algorithm will then take calculate the effective green time (EGT) for each traffic light. The EGT shows how much of the current green light time for a given traffic light is actually used by cars driving through the intersection. If there is less traffic congestion present, there may be a large amount of green light time that isn't being used and is essentially dead time. Conversely, if there is a lot of traffic congestion, the majority of the green light time may be used.

$$EGT = FR \times C$$

After the EGT is calculated, the algorithm goes into a recursive function. The remaining three equations are calculated one traffic light at a time, starting with the traffic light with the highest priority.

Using EGT, the phase green time (PGT) is calculated where S is the lost time due to cars needing to accelerate from a stop after a red light, and Y is the length of the yellow light. The PGT is essentially the same calculation as EGT, but with outside factors taken into account.

$$PGT = EGT + S - Y$$

Now that we know the effective green time and the phase green time, it is possible to calculate the remaining green time RGT. This tells us how much time is lost due to outside factors.

$$RGT = EGT - PGT$$

The remaining green time is the total time that could be changed in each green light cycle. Finally, the remaining effective green time $REGT$ is calculated. CRT_{current} is the FR for the current traffic light being looked at in the recursive function. $CRT_{\text{remaining}}$ is the sum of all the FR of every traffic light with a lower priority than the traffic light being currently looked at. $CRT_{/mboxprevious}$ is the FR of the traffic light with a priority one above the current traffic light being looked at. If there is no traffic light with a higher priority, CRT_{previous} is zero. $REGT$ is the total amount of time that needs to be added or subtracted from the current green light time for a given traffic light in order to optimize traffic flow.

$$REGT = RGT \times \frac{CRT_{\text{current}}}{CRT_{\text{remaining}} - CRT_{\text{previous}}}$$

The PGT, RGT and $REGT$ is then calculated for all of the remaining directions. Once all of the traffic lights have a $REGT$, the results are returned to the base station. The base station then adjusts the green light times for every traffic light within the intersection using the $REGT$.

5.4 Monitoring Application

In order to ensure that the algorithm is working correctly and isn't producing troublesome results, a mobile interface will be designed to allow for authorities to monitor traffic light times [2]. This application will allow for authorities to zoom in on every traffic intersection to see the current green light time. If a problem is noticed, the authorities will be able to manually change the green light times using the application as well.

5.5 Testing and Results

To test the green light time calculation algorithm, Chong et al. used a MATLAB simulation software. A four-way intersection consisting of cars travelling north, south, east, and west with a maximum queue consisting of 75 vehicles. Only standard passenger cars with a length of 4.2 meters (13.78 feet) and a width of 1.75 metes (5.74 feet) were used in the simulation. As a fixed variable, the simulation required that every traffic light had a chance to go through their cycle in no more than 300 seconds. Each of the 4 traffic lights was set to have an initial green light time of 75 seconds. As the simulation ran, 0 to 50 vehicles would travel in a random direction. The simulation was run for 100 light cycles.

After the simulation was completed, it was found that the green light time algorithm greatly reduced the average queue length and the average wait time for an individual vehicle.

The results are shown in the table below.

Parameters	Dynamic Algorithm	Fixed Time Algorithm	Percentage of Improved
Average Queue Length (meters)	198	620	68.06%
Average Waiting Time (seconds)	16	48	66.67%

As seen in the above table, the average queue length was reduced by 68.09 percent and the average wait time was reduced by 66.67 percent. This shows that the proposed system and algorithm not only reduces the amount of traffic congestion in areas of congestion, but also reduces the average travel time for individual vehicles.

6. CONCLUSION

Smart traffic systems can be a helpful tool to mitigate traffic congestion across the world. In this paper we've looked at three different proposed smart traffic systems, with each have different levels of human assistance required in order to operate them. Each of the proposed systems has been shown to reduce traffic congestion and could greatly help many cities around the world if implemented.

Even though the results of each of the proposed systems looks promising, none of them have been implemented in the real world yet. The next step would either be to start implementing these systems in select areas so that further testing can be done in real world conditions or, to run more simulations on grids consisting of many different intersections to determine the effect the systems would have on a larger scale.

7. ACKNOWLEDGEMENTS

I would like to thank K.K. Lamberty, Elena Machkasova, and Scott Steffes for their help and guidance throughout this process. I would also like to thank my friends and family for their encouragement and support.

8. REFERENCES

[1] I. Agenda. Rfid (radio frequency identification). [Online; accessed 15-April-2019].

[2] H. F. Chong and D. W. K. Ng. Development of iot device for traffic management system. *IEEE Student Conference on Research and Development*, December 2017.

[3] Forbes. A simple explanation of 'the internet of things'. [Online; accessed 15-April-2019].

[4] S. Misbahuddin, J. A. Zubairi, A. Saggaf, S. Basumi, Jihad an dA-Wadany, and A. Al-Sofi. Iot based dynamic road traffic management for smart cities. *12th International Conference on High-capacity Optical Networks and Enabling/Emerging Technologies*, December 2015.

[5] Nationwide. Sitting fuming: traffic congestion statistics. [Online; accessed 4-March-2019].

[6] P. Rizwan, K. Suresh, and R. M. Babu. Real-time smart traffic management system for smart cities by using internet of things and big data. *International Conference on Emerging Technological Trends*, October 2016.

[7] T. Terms. Gpio. [Online; accessed 15-April-2019].

[8] Wikipedia. Raspberry pi. [Online; accessed 4-March-2019].