

An IoT Based Intelligent Traffic Management System and Its Implementation on Cupcarbon for Smart Cities

Oguz Aydiner¹, Adnan Kavak^{1,2}

¹Yeditepe Univ., Electrical and Electronics Eng. Dept., 34000 İstanbul, Turkey,
oguz.aydiner@std.yeditepe.edu.tr

²Kocaeli University, Computer Eng. Dept, 41380 İzmit, Turkey, akavak@kocaeli.edu.tr

Received Date : October 03, 2021 Accepted Date : October 25, 2021 Published Date : November 07, 2021

ABSTRACT

This paper proposes an internet of things (IoT) based intelligent traffic management system that can aid problematic traffic situations in smart cities by classifying congestions via sensory data, and then controlling traffic lights and creating alternate routes for incoming vehicles to the congested zones in order to relieve or avoid congestions completely. The proposed intelligent traffic management system consists of different subsystems such as Test Operation, Supervisory, Traffic Light, and Pathfinder subsystems. The system is represented by flowcharts with their explanations and its operation with some defined scenarios is validated with the CupCarbon simulation environment.

Key words : Cupcarbon, IOT, Smart City, Traffic Management, 5G

1. INTRODUCTION

The traffic congestion is one of the major problems in today's mega-cities. There are of course outliers to the congestion problems but some cities have it worse than others. As an example, Istanbul has the second most congested and sudden-stopping traffic in the world [1]. Istanbul's traffic congestions are very often, emissions are high, incidents are regular and road rages are a part of daily life. To address the traffic problem in the big and highly populated cities such as Istanbul, heavy infrastructure investments can be made which will cost a lot of taxpayer money, such as bridges and additional roads. However, these costly investments do not only cost a large amount of taxpayer money, they also come with additional costs such as taking up free space, environmental damage and prevent further infrastructure projects. Therefore, installing intelligent systems to manage traffic that can adjust themselves based on the vehicle density are beneficial in terms of the best use of resources [2].

The congestion in traffic is essentially because of high demand on traffic, which occurs in different times of the day with different contributing factors. Most of the congestions are caused by bottlenecks on the traffic [3]. They are mostly the reason why traffic even gets congested because the following reasons of congestions are secondary to bottlenecks. The poor signal timing or even worse, non-signaling causes an interesting phenomenon called "Phantom Traffic Jam" which occurs when the slightest tipping point of the traffic is reached when the road is heavily dense with vehicles, but still flowing nonetheless [4]. The traffic incidents play a similar role to the work zones as they too create bottlenecks and restrict the "airflow" of the roads so the traffic gets congested very quickly. Work zones usually creates bottlenecks on the roads since they need to close down that area of maintenance to the traffic and the five-lane road goes down to three, there is indeed going to be bottlenecks. Bad weather is also a cause for traffic congestion simply because it increases the demand for use of transportation via traffic.

Today's mega-cities which are prime contenders to become the future's smart cities are bound to have above mentioned traffic problems. Description of a framework for utilizing the various sources efficiently in the context of traffic management is addressed in [5]. One countermeasure of the traffic congestion problems of such cities is the design and installation of intelligent traffic systems that can be brought by the advancement of technology such as IOT networks enabled by fifth generation (5G) mobile networks [6]. In [7], Intelligent traffic control system to solve crossroad problems and parking space problems using wireless sensor network (WSN) to collect road data and available parking space in smart cities are given.

In this paper, as a solution to the congestion problem in smart cities, we propose a method for intelligent traffic management based on IOT sensor data. Our solution addresses a possible solution to traffic congestion problems. We perform simulations of the proposed system via CupCarbon IOT simulator.

2. INTELLIGENT TRAFFIC MANAGEMENT SYSTEM BASED ON IOT NETWORK

In this paper, we propose a traffic management system to operate in smart cities based on IOT sensor data. The proposed system consists of four subsystems namely Test Operation, Supervisory, Traffic Light, and Pathfinder subsystems. Below, we explain operation of each of these subsystems.

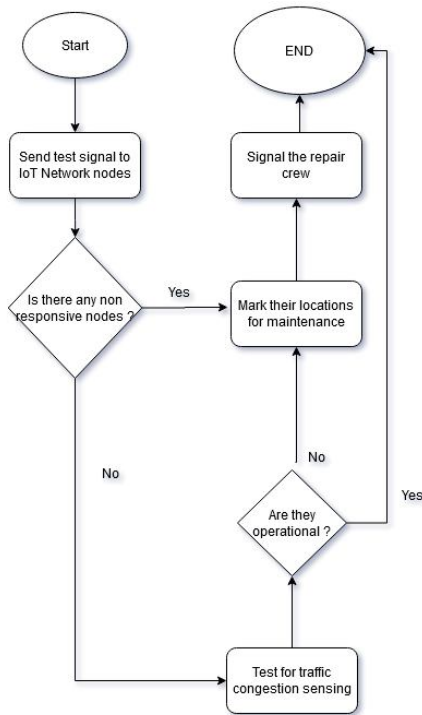


Figure 1: Test Operation Subsystem of the Proposed Traffic Management System

Figure 1 shows the “Test Operation Subsystem” of the proposed IOT Traffic Management System, which is good to have for a realistic system’s operations. It could save a lot of the hassle and time-waste of trying to detect the erroneous nodes in a system where there is not a safety measure built in. As we can see, it firstly orders the transmitter to send a test signal to all of the IOT sensor nodes to check which are responsive and which are not. The nonresponsive nodes are marked with their location data for the maintenance crew. The responsive nodes pass the first test and continue to the second step for testing their functionality, as in checking if they are doing what they are supposed to do. In the last step they are tested for their most essential function; congestion sensing. The test system sends a signal for their sensing and reporting capabilities and either confirms they are indeed operational or decides that the data they are reporting is erroneous and marks them for maintenance. If the test is passed by flying colors and there are neither errors in the operation of the nodes nor non-responsive nodes, the system ends operation and notifies the tester that the system is healthy.

The Supervisory Subsystem’s (SS) flowchart is given in Figure 2. Its purpose is to tie the whole Traffic Management System together with its subsystems. It starts off with sending a “situation report” signal and receive feedback. Classification of road congestion levels within the IoT Network area is firstly done by the Supervisory System to determine the situation of the traffic and decide whether to contact a relevant subsystem or end operation. If it decides to activate a subsystem, it then needs to decide which one to wake up. So, there has to be a classification about the road congestion levels.

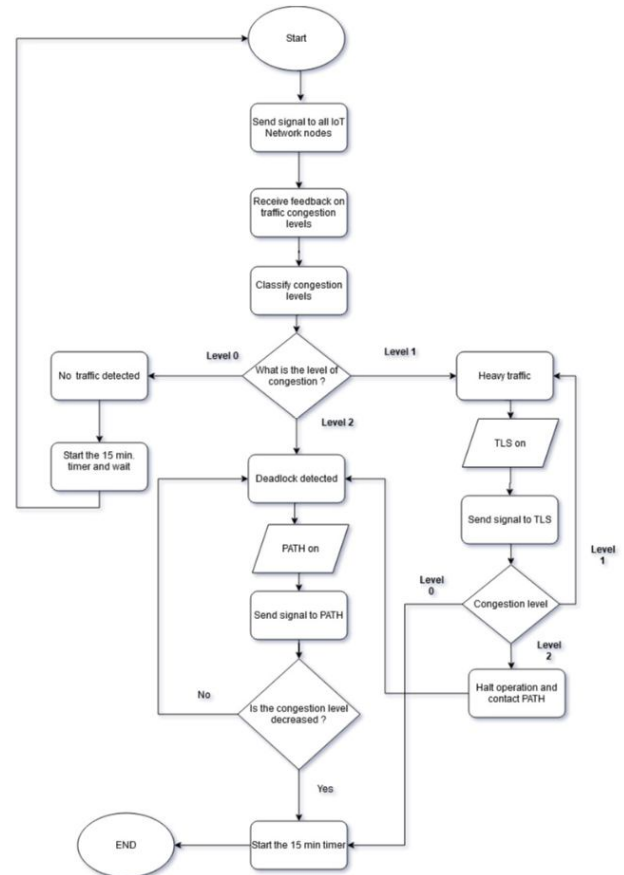


Figure 2: Supervisory Subsystem of the Proposed Traffic Management System

Level_0 is “no congested traffic” detected case, which means that the system detected no congestion as in the detected object count and their detection timeframes are as expected on the area where IoT network is deployed. So, it ends its operation and starts its timer for the duration of 15 minutes. Because it’s dealing with traffic and traffic never stops so the system shouldn’t stop either. That is why after setting its timer, it waits. Level_1 is “heavy traffic” detected case, which means that the vehicle count and their detection timeframes are between minimum and maximum parameters and the traffic is congested. So, the Supervisory Subsystem activates the Traffic Light Subsystem (TLS) on the road that are determined to be Level_1 congestion. The TLS operates as shown in Figure 3. Level_2 is “deadlock traffic” detected case, which means that the traffic is extremely congested. In

this situation, the IoT Network’s vehicle count is very high and the more deciding factor is, their detection timeframes are very high as well, meaning they are barely moving or stationary. So, if this happens to be the case, SS proceeds as shown in Figure 2 and activates PATH system to map alternative routes to incoming vehicles in order to decrease the congestion on the road that is already reached Level_2 congestion. After some iterations when the congestion level is decreased, the operating subsystems notify the Supervisory System right before ending its operation, and the system starts its timer for when it is over.

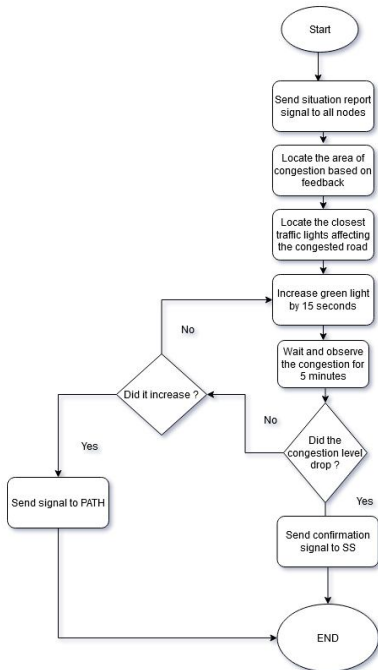


Figure 3: Traffic Light Subsystem of the Proposed Traffic Management System

The flowchart of the Traffic Light Subsystem (TLS) is shown in Figure 3. It first sends a signal like in Test Operation Subsystem to all nodes to receive a feedback about the current congestion levels of the region wherein the sensors are placed. When the feedback is received, the system locates the area of congestion, and shortly after that, it locates the traffic lights that effect those particular roads. Then the system sends a signal to increase the duration of green light by 15 seconds, which is a placeholder value and would change according to the circumstances. After increasing the green light duration, the system waits and observes the congestion levels by sending a signal again to the IoT network nodes at the end of the wait duration which is another placeholder value as 5 minutes. If the congestion level is dropped to Level_0 from Level_1, the system sends a confirmation signal to the SS and ends its operation. But if the congestion level is still the same, as in Level_1, the system goes back a few steps in the operation and increases the green light duration for another 15 seconds. However, if the congestion level is raised to Level_2 even if the system has increased the green light duration, then it means the TLS will not be enough to resolve

the current congestion. So, the system halts its operation, contacts Pathfinder (PATH) subsystem and ends all operations.

Figure 4 shows the flowchart of the Pathfinder (PATH) Subsystem in detail. Similar to other subsystems, PATH too sends a signal to all nodes of the IoT network to receive information about the current status of the road congestions from the sensors. Then the system classifies the received feedback from the sensors and sets a priority of Level_0 over Level_1 road when mapping alternative routes. This is done in order to both optimize the routes of incoming vehicles to their destinations and divert the traffic to more unused routes to help decrease the congestion of the deadlock road. Meaning a Level_0 road can potentially host more vehicles than a Level_1 road because it’s further away from becoming another deadlock. After the priority phase, the system reaches out to the driver’s GPS to request location. After receiving, it then notifies the driver about the upcoming congestion and requests a manual input regarding their destination.

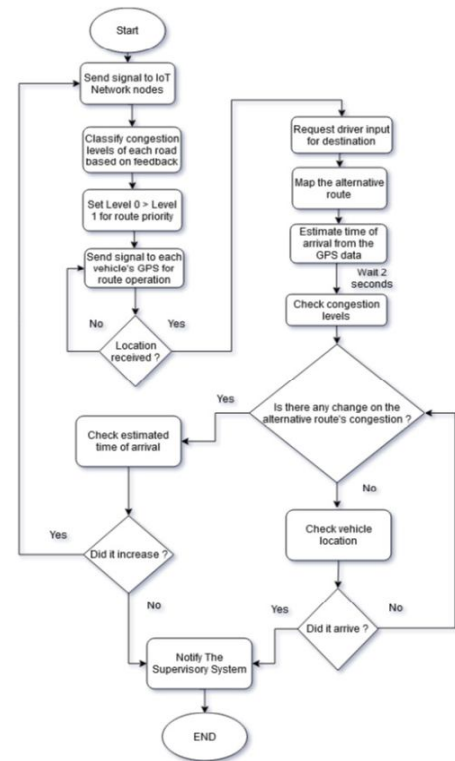


Figure 4: Pathfinder (PATH) Subsystem of the Proposed Traffic Management System

When the destination is set, the system maps an alternative route for the driver. Then constantly begins to check the alternative route’s congestion levels to see if there is any change in the congestion and if there is no change it then checks vehicle location to see if it arrived its destination. If the system senses a change of congestion in the alternative route, it then asks if this change is an unwanted change as in an increase in the congestion levels. If that is not the case and the change is for example, Level_1 to Level_0, then it does nothing since this case benefits everyone and ends its

operation after notifying the Supervisory Subsystem. However, if the change is in an unwanted direction, as in an increase in the congestion levels, then PATH goes back to the beginning of the operation to map a new route for the driver. The operation is only resolved when the congested road's levels have been reduced as there is no other means to deal with it better than the PATH system so the cycle repeats until when the levels are dropped.

3. CUPCARBON SIMULATIONS AND RESULTS

Proposed traffic management system is simulated using CupCarbon IOT Simulator [8],[9]. It is a communications simulation tool to visualize the cases created, specifically designed for IoT Networks and Smart Cities with built-in OpenStreetMap. It is very helpful to visually explain the workings of communications networks such as sensors and decision support system of the project. The program has two different environments for simulation. One is for events which include mobility like traffic and natural events. Second environment is for discrete events of the IoT Networks, which also encompasses the first environments. The platform is still being developed and the users have limited uses for their simulation scenarios due to the complex structure of the IoT Networks on the OpenStreetMap frame.

We have used several objects in CupCarbon simulations to accurately describe traffic congestions and aid the understanding of the system which is represented with flowcharts. Sensors are modules which has the purpose of detecting objects or changes within it's sensing radius. Sensors can be actually linked together by just placing them close enough and assigning the correct scripts with their roles as in receiver/sensor/transmitter to make them communicate. Base station's function is very similar to what they do in real life, which is transmitting and receiving signals and being a bridge between node-system communications. Base station locations need to be in the most optimal places to better communicate with the network. Mobile objects are to visualize moving vehicles / drones on the OpenStreetMap within the assigned routes. The mobile object also has a sensing radius similar to the sensor nodes on the simulation. This is because the object can be scripted to interact with the nodes it comes across to change its route. The basic principles of the CupCarbon simulations of proposed traffic management systems are as follows;

- The IOT Network refers to a network of sensors scattered around a designated area which communicate between themselves and the base stations for the management.
- The system never stops and when it comes to the end of the operation flow, it starts a timer for when it ends, the operation repeats.
- The system sends signal to all of the nodes of IoT Network and checks the congestion.
- The system classifies the traffic congestion into three levels as no traffic, heavy traffic and deadlock.

- Depending on the congestion levels, The system communicates with the related subsystems to resolve it.
- The two deciding factors of the congestion levels are *vehicle count* and *detection timeframes*. We chose to explain them without giving them numerical values as it would not be accurate and would differ heavily according to the IoT Network's placement.

3.1.IOT Node Communication in Intelligent Traffic Management System

For the demonstration of the operation of our proposed system, we have chosen the location as "Bostancı Bridge" in Istanbul, because the congestion problems are quite common in that region. It gets congested every rush hour and even on the weekends regularly. It was also quite handy for drawing routes as it is right above the D100 highway in Istanbul so that we can have more real estate for the sensor placements. The first simulation is designed to better understand how the communication between the IoT Network nodes occur and how do the systems send a check signal to nodes for them to check the traffic congestion levels and report back their gatherings. In Figure 5, the system is in its initialized state. The IoT Network is made up of 100 nodes which are scattered around the area of selection at random. The system is in its inactive state so we are not seeing any communication arrows popping out of them. The transmitter node is at the bottom-right of the network if you look closely, which will send a signal to the system and receive a feedback throughout the simulation.



Figure 5. A Screenshot of CupCarbon Simulation at Initial Condition for Node Communication in Intelligent Traffic Management. Total of 100 nodes are assumed to be deployed randomly around Bostancı Bridge in Istanbul

We can see from Figure 6 that the signal has been sent and it's being transmitted from one node to another within their radius in a wave-like formation. The closest nodes to the transmitter are not showing any arrow signs because they are already checked, given up their information and marked. We can see that transmitted signal is in the middle of the IoT network and progressing towards the outer edges. The nodes are rapidly checking and confirming each other's signals and relaying them to the unchecked nodes so on and so forth. Each node gets checked multiple times to see if it's marked and if

they are, they are left alone so that the signal wave can continue towards the outer edge. The nodes get marked once they have relayed their information to their corresponding neighbor node. When transmitted signal reaches the very edge nodes, they will signal the responsible node between them to relay a feedback signal to the transmitter. You can see the receiver script in Figure 7, which runs at every node in the simulation except the transmitter node which is located down below in Figure 6. In the node script, the process is easy to follow as nodes very quickly ask their neighboring nodes when they receive the signal if they are marked. If they are not marked it means they are not checked, meaning the signaling should progress towards them, so they get the information from their neighboring marked nodes and with adding their own, relay it to them, then the receiving node marks the neighboring node to keep them out of the checking loop and allowing the signal to progress like a wave towards the edges of the network.



Figure 6. A CupCarbon Simulation Screenshot After a While When Node Communication in Intelligent Traffic Management is Started.

```
simulation 0 100
getpos p
atget id id
set recA 0
loop
wait
read m
rdata $m rid type info info2
if(($type==A) && ($recA==0))
set recA 1
if($info==sid)
mark 1
println $p
data d $id B $p
send $d $rid
else
set prev $rid
data d $id A $info
send $d
end
end
if($type==B)
simulation 0 200
mark 1
data d $id B $info $info2
send $d $prev
end
```

Figure 7. The CupCarbon Script Running at the Individual Receiver Nodes for IOT Node Communication in Intelligent Traffic Management System

So as can be seen in Figure 8, the feedback message transmission has begun from the outermost responsible node. They are individual nodes following a straight line and not all of them at once, because that would be inefficient to have all nodes communicate back while we can have them report to their responsible nearest node and still have the same result but faster. We can see that the feedback signal is moments away from reaching the transmitter and the signaling operation is about to be finalized. You can see the transmitter script down below. As can be seen in Figure 9, basically the transmission occurs at the point where the “send” command is and then it waits until receiving the feedback from the nodes and checks if it matches its type, which it will, then marks it and ends the process.



Figure 8. A Screenshot of CupCarbon Simulation Showing Feedback Message Transmission Towards Transmitter Source Node for Node Communication in Intelligent Traffic Management System.

```
atget id id
data d $id A 24
send $d
loop
wait
read m
rdata $m rid type x y
if ($type==B)
mark 1
println $x $y
stop
end
```

Figure 9. The CupCarbon Script Running at the Individual Transmitter Nodes for Node Communication in Intelligent Traffic Management System

3.2. Route Changing with Location Signaling in Intelligent Traffic Management System

This simulation is combined with various features to better visualize the PATH subsystem in our proposed intelligent traffic management system. The operation will start with our mobile object interacting (representing a vehicle) with assigned roadside IOT sensors to receive “change route” commands and reaching its final destination. Upon reaching, it will actively signal its location to the IoT network established on the map.

In Figure 10, we can see that the simulation is in its initial phase, the mobile object is not moving and the system is not active but we can still see the pre-drawn routes for the mobile object to follow. It really is not possible in a CupCarbon simulation to literally draw alternative routes during the simulation progress and have that kind of processing capability but it's still quite alright to simulate a system like PATH as shown in the figure to give a vague sense of operation and actually explain the system in flowcharts.

Figure 11 shows that the simulation has begun, mobile object has quickly followed it's assigned marker route and interacted with the first route changing sensor, which signals it to follow the node m2 as in the other route in its radius. So, mobile changes its route and proceeds to follow m2. In Figure 12, you can see the script of the first sensor node. It simply reads a tag and sends a route change signal as m2 which is the name of the second route.

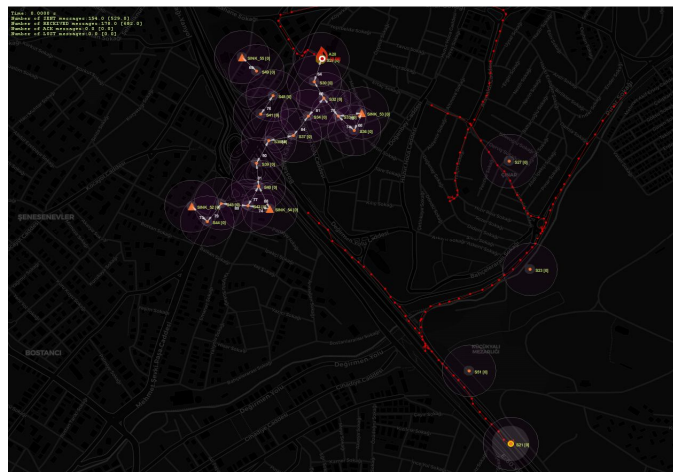


Figure 10. Initial Screenshot on CupCarbon for the Route Following and Changing with Location Signaling Simulation

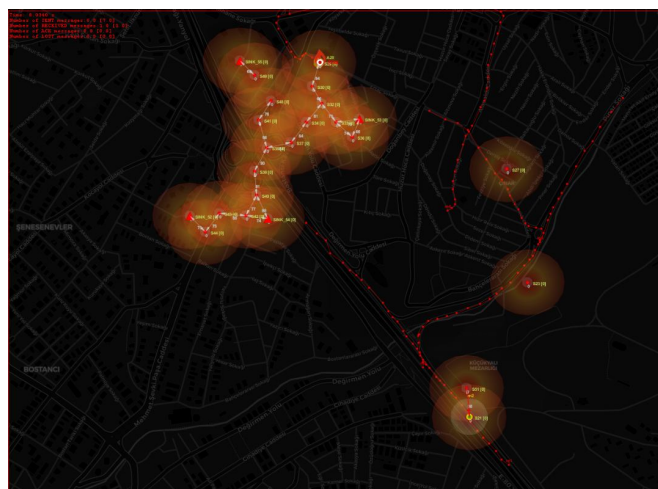


Figure 11. Second Screenshot on CupCarbon for the Route Following and Changing with Location Signaling Simulation

```

loop
wait
read x
if ($x==A)
send m2
end
    
```

Figure 12. The CupCarbon Simulation Script of the Sensor1 for Route Following and Changing with Location Signaling

In Figure 13, we can see that our mobile object (vehicle) has reached the second sensor for route changing and its script is almost identical to the first one, only difference being is the sent signal as in the name of the new route, which is m3 in this case, the name of the third marker route. The one minor difference is the tag name changes from A to B, which the sensor is checking, that is simply to avoid confusion as you will see in the script of the mobile object. The first sensor demands A tag, when receives, commences to send signal change route, the second sensor demands B tag, so on and so forth.

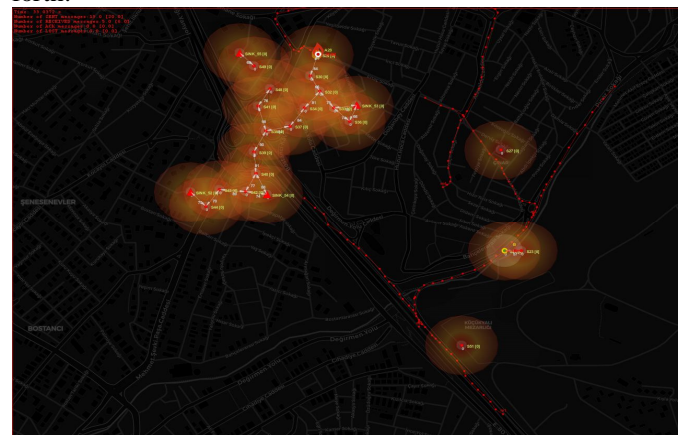


Figure 13. Third Screenshot on CupCarbon for the Route Following and Changing with Location Signaling Simulation

Figure 14 shows that the mobile object has reached its last stop regarding the route changer sensors, which is the third sensor. It will then send a signal to change its route from m3 to m4 which is the fourth and final route before arriving its destination.

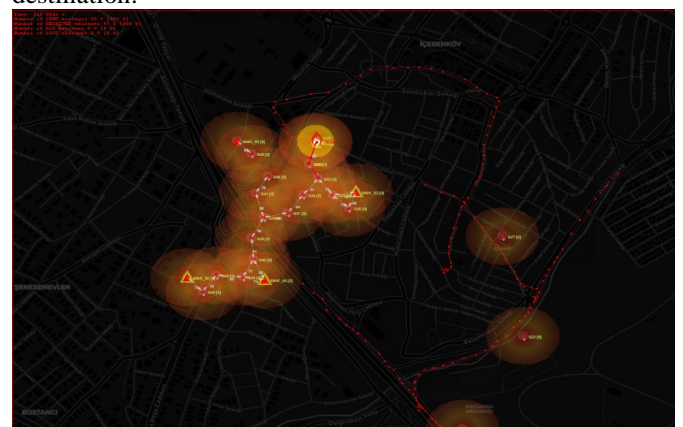


Figure 14. Fourth Screenshot on CupCarbon for the Route Following and Changing with Location Signaling Simulation

Figure 15 shows that the mobile has reached its final destination but it's not the end of the simulation. On the contrary, another interesting operation starts to take place, which is the location signaling from the end point to the IoT network for the Supervisory System to get out of the loop. When the mobile object has reached to the region shown by the fire sign at the end point, the sensors there are linked together and scripted to sense this particular occurrence and mark each other just like in the Node Communications simulation prior to this, only difference is that the signals are relayed by the base stations so they visually seem more orderly.



Figure 15. Fifth Screenshot of the Route Following and Changing with Location Signaling Simulation

```
//Router
atget id id
loop
wait
read rp
rdata $rp rid v
data p $id $v
send $p * $rid
```

Figure 16. The Script of the Sinks of the Route Following and Changing with Location Signaling Simulation

In Figure 16, the script of the sink which is a base station is given. They are linked to many sensors and control the flow of signals along the network to properly use the data when the events occur such as the marked vehicle reaching its destination point.

In Figure 17, the script of the main player of the simulation which is the mobile object is given. As explained priorly at the sensor scripts, you can see that the tag changes depending on the sensor. Which puts the mobile's operation in another "if" condition, allowing it to easily proceed to change its route depending on the sensor demanding the signal and incoming command.

```
simulation 80 200
atget id id
set x 0
loop
dreadsensor s
if($s==1)
simulation 100 200
data d $id A
send $d
delay 10000
end
if($x==0)
send A
end
if($x==1)
send B
end
if($x==2)
send C
end
wait 1000
read d
if($d!=\ )
if($x==2)
set x 3
print $d
route $d
end
if($x==1)
set x 2
print $d
route $d
end
if($x==0)
set x 1
print $d
route $d
end
end
delay 1000
```

Figure 20. The Script of the Mobile Object (Vehicle) for the Route Following and Changing with Location Signaling Simulation

4. CONCLUSION

In this paper, we have proposed a system for intelligent traffic management to overcome traffic congestion problems in smart cities. The system is composed of four subsystems: Test Operation Subsystem, Supervisory Subsystem, Traffic Light Subsystem, and Pathfinder (PATH) Subsystem subsystems. Algorithms of each of these subsystems have been given and simulation of this system with an emphasis on Route Following and Changing with Location Signaling scenario has been made on CupCarbon platform which is an IOT network simulation tool.

The systems that are designed here are raw and core concepts and there is vast amount room for improvement with possibilities for potential development in the future smart cities. For example, the IoT Network's sensors can be upgraded and reinforced with image processing which can enable the system to easily recognize the problems that are occurring and swiftly work on solutions. Or rather, the system can be combined with artificial intelligence which can be trained to manage the traffic flow entirely. These are all futuristic possibilities that have enormous potential but they can indeed be realized and they would solve a vast spectrum of problems.

REFERENCES

1. https://www.tomtom.com/en_gb/traffic-index/ranking/?population=MEGA,LARGE,SMALL [Accessed 10 2021].
2. H. Terzioglu, A. A. M. Harb and A. Durdu, **Design and Implementation of Real-Time System with Fuzzy Logic Method For Junction Traffic Signalization Control**, *European Journal of Science and Technology*, no. Special Issue, pp. 490-497, 2019.
3. A. Koźlak and D. Wach, **Causes of Traffic Congestion in Urban Areas. Case of Poland**, *SHS Web of Conferences*, vol. 57, 2018.
4. K. Goldmann and G. Sieg, **Economic Implications of Phantom Traffic Jams: Evidence From Traffic Experiments**, *Transportation Letter: The International Journal of Transportation Research*, vol. 12, no. 6, pp. 386-290, 2020.
5. A. Allström, J. Barceló, J. Ekström, E. Grumert, D. Gundlegård and C. Rydergren, **Traffic Management for Smart Cities**, *Designing, Developing, and Facilitating Smart Cities: Urban Design to IoT Solutions*, pp. 211-240, 2017.
6. L. Chettri and R. Bera, **UL-DMRS Based NB-IoT Uplink System and its Performance Analysis Toward 5G Machine Type Communications**, *International Journal of Emerging Trends in Engineering Research*, vol. 9, no.1, pp. 6-14, January 2021.
7. A. Hilmani, A. Maizate and L. Hassouni, **Automated Real-Time Intelligent Traffic Control System for Smart Cities Using Wireless Sensor Networks**, *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1-28, September 2020.
8. **CupCarbon IoT Simulator**, <http://cupcarbon.com>.
9. M. Akkurt and K. Küçük, **Simulation of Smart City Applications Based on IoT Technologies with CupCarbon**, *In Proc. 3rd International Conference on Computer Science and Engineering*, 2018, pp. 179-184.
- 10.