Intelligent Transportation System with Wireless Sensor Networks

Alperen Gündogan, Besar Lami, Cagatay Moroglu
Fakultät für Elektrotechnik und Informationstechnik, Technische Universität München
Wireless Sensor Networks Laboratory
München, Germany
gundoganalperen1@gmail.com, besarlami@gmail.com, moroglu58@gmail.com

***Abstract*—This project is designed to create an intelligent transportation system by exploiting low cost wireless sensor motes. The wireless network is constructed with the help of Destination-Sequenced Distance Vector(DSDV) routing algorithm to maintain multi-hop and loop-free communication. The network consists of Road Side Units(RSUs) and Mobile Units(MUs) which are equipped with IR-distance sensor and force sensor respectively. RSUs provide information about traffic condition on the road and maintain a communication channel between MU and traffic center in the case of emergency. Moreover, the network traffic and emergency conditions can be tracked by the GUI.**

***Index Terms*—DSDV, ITS, WSN, Zolertia, Contiki, traffic-monitoring,VehicleToInfrastructure, collision-detection, data-logger**

I. Introduction

The growth in the people mobility results in an increase in the number of cars on the road. The increasing traffic congestion in a city has negative social, environmental and economic impacts over the city life, like the congestion makes people more stressful which may cause mental issues and it increases the fuel consumption which leads to air pollution. In 2017, the people of Munich have spent the highest hours in congestion in Germany. The total cost of congestion in 2017 was over 2.9 billion Euro [1]. Besides these effects, increasing traffic volume increases the risk of car accidents, which are sometimes fatal. In 2017, 3180 people have died in Germany because of car accidents [2]. Early medical intervention is the most important way to reduce the casualties in car accidents.

The emerging technologies of Wireless Sensor Networks revolutionized the traffic management, control and response to an accident. WSN technologies make traffic monitoring easier, which can be used to distribute traffic to different roads to decrease traffic congestion. The decrease in traffic congestion brings many benefits to the country, city, and city life, such as the savings on money spent because of traffic congestion and air pollution can be reduced. Another benefit of traffic monitoring is that early medical intervention to car accidents can be provided, by understanding where the accident happened and steering the ambulance very quickly to the accident, resulting in many saved lives.

To provide a traffic monitoring environment, low-cost sensor motes with computing and wireless communication capabilities can be used. Real-time traffic data can be gathered in a traffic center with the help of the motes and WSN. This traffic center can intervene in emergencies.

Many similar works exist in the literature, provide wireless connectivity for vehicle-to-vehicle and vehicle-to-infrastructure communications. Two example funded by the European Union are SAFESPOT project [3] and COOPERS project [4]. Both make the road infrastructure intelligent enough to identify the dangerous situations and warn the drivers.

In this project, we designed an Intelligent Transportation System with Wireless Sensor Networks, by using Zolertia™ Re-Mote wireless motes which support IEEE Std. 802.15.4 specification for Low-Rate Wireless Personal Area Networks(LR-WPANSs) [5]. These motes are running Contiki OS which is an open source operating system for the Internet of Things [6]. Some of these motes are placed on the side of the road, which are called roadside units, RSUs. They divide the road to sectors and count the number of passing cars in front of them. In this way, it can be understood how many cars exist in each sector. This will give information about the traffic congestion in each sector. In our project, each car called Mobile Unit is equipped with a wireless mote, which can detect if a car is involved in an accident and send the collision data to everyone in the network.

The first thing we need to provide clear communication between nodes is the routing. Every node should know how to reach the other nodes when it is necessary. Even if the motes are not in range, the network should try to support the communication with the help of multi-hop feature. In this part, we have exploited the Destination-Sequenced Distance Vector(DSDV) routing protocol to provide multi-hop and low latency communication between motes which will be explained in the subsequent section.

The paper is organized as follows. Section II gives information about the routing algorithm we used in this project and its implementation. Section III presents the scenarios we covered in this project and the features of the network. Section IV presents the Graphical User Interface (GUI). The conclusion is presented in Section V.

II. Network description and routing technique

Our design mostly focuses on emergency cases i.e. a car had a crash, then the network should be able to broadcast the collision message to other vehicles in the vicinity to warn them for possible danger. Furthermore, the network provides an emergency channel between the collided car and the traffic center. The network consists of two parts, roadside units(RSUs) and mobile units(MUs).The RSU connected to the GUI is considered as the gateway. RSUs exploit DSDV routing algorithm to sustain updated routing table for each RSU. MUs only involve the RSUs' network when an accident occurs. All of the RSUs contain routing entries for each mote in the network, therefore collided car can rapidly obtain the most up-to-date routing entries for each destination in the network.

DSDV is a proactive(table-driven) routing algorithm which is developed by C. Perkins and P.Bhagwat in 1994 [7]. DSDV is based on Bellman-Ford algorithm with the improvement of sequence numbers to overcome the count-to-infinity problem. It provides a routing entity for all available destinations in the network and guarantees loop-free path. DSDV maintains only the best route for each destination address thus, the memory consumed by the routing table is diminished. Furthermore, DSDV outperforms the other routing algorithms in mobile ad hoc networks(MANETs) e.g. AODV, DSR, OLSR, etc. in terms of end-to-end delay and normalize routing load(NRL) [8]. That was our main motivation to use DSDV algorithm for this project because a crashed vehicle on the road should be able to talk as soon as possible with the traffic center in case of emergency. Reactive protocols require higher time for establishing this connection since the car first has to initiate the routing request and to wait until it gets a route reply for the discovery of the destination.

$$NRL = \frac{Number\ of\ routing\ packets\ transmitted}{Total\ number\ of\ packets\ received} \quad (1)$$

A. DSDV algorithm

Each mote broadcasts its entries in the routing table to the neighbours and exchanging of the routing entries converges to the optimal routing. The routing entries contain the following pieces of information:

- The destination's address:
- The next hop address.
- The number of hops to reach the destination.
- The sequence number of the entry.

The sequence number is originated by the destination motes and increased by two for periodic updates only by this mote.

1) Routing advertisements: Periodic routing table updates are exploited to keep the connection between the neighbor motes alive. Each mote broadcasts its routing table periodically e.g. 15 sec, by increasing the sequence number of its own entry, if the neighbor motes could not receive any higher sequence number entry from the destination for a predefined period of time, then they can assume that the destination mote is no longer in the network and mark the entry as a stale and advertise this stale entry to other motes. Moreover, the triggered advertisement is used if there is an update in the routing e.g. a new mote joins the network, route changes, or stale entry announcements. This mechanism facilitates the network to converge faster to changes and therefore decreases the latency in the unstable networks. The flowchart of the routing algorithm can be found in figure 1.

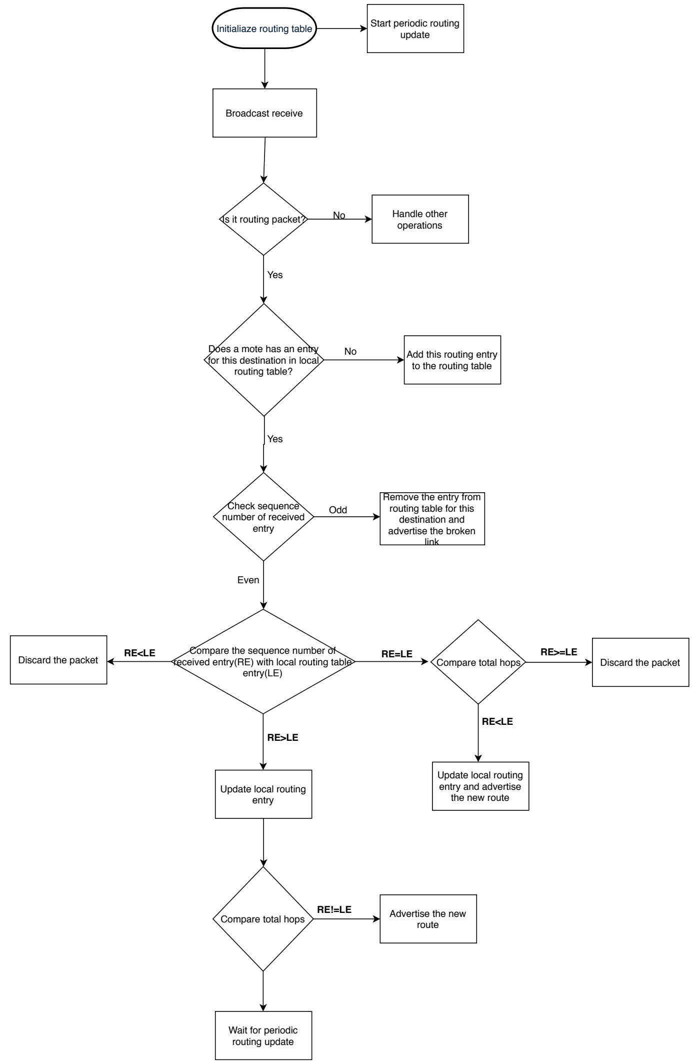


Fig. 1: DSDV routing flowchart

2) Update mechanism: The routing entries always converge to the higher sequence number for each of the available destinations. If the sequence number for the same destination is equal, then total hop count is considered as a metric to determine the shortest path to the destination. If a mote does not receive any routing entry with a higher sequence number for a destination, it increases its sequence number by one and advertises the broken link to the other motes without waiting for the periodic update event.

B. Failure-recovery mechanism

In the original paper, Perkins did not mention any solution in the case of restarting the motes such that the sequence number starts from zero. We circumvented this problem thanks to the periodic routing updates. If a mote receives a routing entry in which the destination address of the received entry is the mote, then it checks the sequence number of the packet. If the sequence number is higher than its own sequence number, then it synchronizes itself to its previous sequence number already in the network and starts increasing it. In other words, restarted motes always converge the latest sequence number for its destination.

III. SCENARIOS

In this project, we covered two main scenarios. First one is the sectoral localization. The road is divided into sectors, by using motes and distance sensors. The sectoral localization helps us to know the number of cars in each sector, therefore the traffic congestion in each sector. The second scenario is the accident detection, in which the traffic center is informed very quickly when an accident happened. An emergency channel is established between the collided MU and the traffic center, which allows for exchange of collision details.

A. Sectoral Localization

Motes are placed on the roadside at regular intervals and are equipped with IR Distance sensor. The sensor is used to detect on object on the road. State machine logic is applied to the sensing algorithm such that the sensor is only triggered once per object. In this way the RSU is able to know when an object is detected and then can inform the traffic center, using the implemented routing algorithm explained in Section II. Subsequently, the traffic center can calculate how many cars exist in each sector of the road.

B. Accident Detection

Each car on the road is equipped with a wireless mote integrated with a force sensor. When this car collides with another car, the force sensor located in front of the vehicle detects the force created. Then the mote sends a broadcast message to everyone. When an RSU receives this message, it applies a controlled rebroadcast mechanism to avoid broadcast storming while informing everyone in the network.

As soon as the car detects a collision, after broadcasting the collision message, this mobile unit will enable its routing algorithm to be involved on the network of RSUs. When the traffic center has an entry for the collided mobile unit in its routing table, it can create a communication channel to the passengers in the car. In this way, the traffic center can learn more details of the accident.

IV. GUI

The Graphical User Interface (GUI) of this project, as shown in figure 2, is implemented using Qt framework version 5. The purpose of the GUI is to be able to:

- connect to traffic center RSU via serial and print port activity
- visualize the routing table
- display road sectors between RSUs
- detect collision messages from MU and alert the operator
- show the collided MU in the routing graph
- send messages to the collided MU and display the reply
- show temperature and battery voltage data of the RSU
- send custom commands to connected RSU if needed

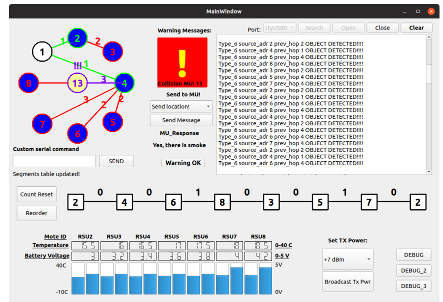


Fig. 2: GUI overview

A. Routing Visualization

The routing algorithm of the network is decentralized, therefore a dynamic GUI is necessary for the correct display of the routing table. The GUI uses circular nodes to represent RSUs and lines to represent a link between two RSUs. Using colors we distinguish between current connected RSU, 1-Hop, Multi-Hop and MU, as seen in figure 3

(a) Current RSU (b) One Hop (c) Multi Hop (d) Collided MU

Fig. 3: RSU node types

To better show the functionality of the interface, an example network is considered as shown in figure 4. The trees represent obstacle areas where a wireless connection is not possible.

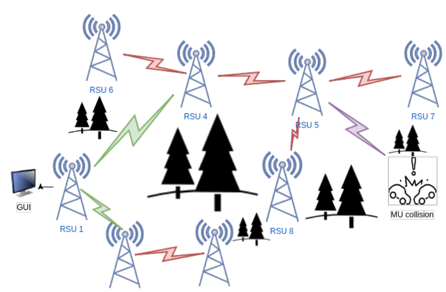
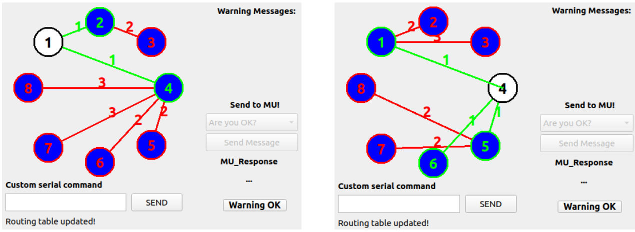


Fig. 4: Network example

The connected RSU will send the necessary information to construct the routing table via serial. Then the routing table is drawn on the screen and updated on the fly when new information is received. The routing graph can be observed in figure 5 for two of the nodes in the network, respectively RSU1 or traffic control and RSU4. The number over the lines represents the total hop to reach the respective node. In case of multi hop, only the next hop address is available in the routing table therefore more than two hops cannot be visualized explicitly.

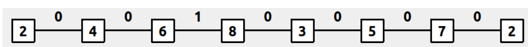


(a) Connected to RSU1 (b) Connected to RSU4

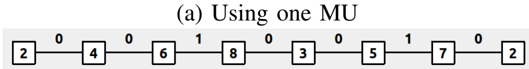
Fig. 5: Example network in GUI

B. Road sectors

In order to display traffic, sectors are used. A sector is defined as the part of the road between two RSUs. A sector holds a counter which represents the number of cars. The demo setup used has a circular type of road and therefore the first sector is the same as last one, as seen in figure 6.



(a) Using one MU

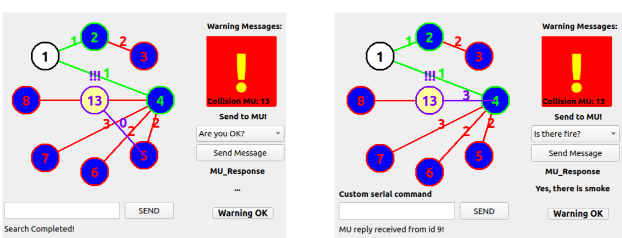


(b) Using two MUs

Fig. 6: Example display of sectors

C. Collision Visualization

The collision messages are processed by the traffic center node. When the traffic center receives a collision message, a big warning sign is flashed on the screen along with the MU node. Immediately, the "Send to MU" part becomes active. After the MU is fully connected to the network it will be displayed in the routing graph. When the accident is resolved, "Warning OK" button returns GUI to normal state. The initial collision response can be seen in figure 7:a.



(a) Collision received (b) Reply received

Fig. 7: Collision detected

D. Collision Handling

When a collision occurs, MU broadcasts a message which is routed by RSUs to the gateway. The gateway immediately displays the MU node in the routing graph and enables the "Send to MU" section. After the MU is fully connected to the network, the correct total hop is displayed in the routing graph. At this point, the operator would send direct messages to collided MU and wait for the response, as seen in figure 7:b. When the situation is resolved, the operator can clear the collision by pressing "Warning OK" button, which also informs the collided MU, which removes itself from the network, returning GUI and network to normal state.

E. Data-logger display

Along with the emergency situations, the GUI also serves as a data logger display. Temperature information and battery level of the RSU are send periodically to gateway. The received message will enable the corresponding display based on the ID. An example can be seen in figure 8.

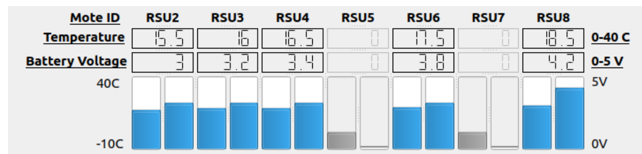


Fig. 8: Data logger elements

V. Conclusion

This project presents the implementation of the DSDV routing algorithm in the transportation system by using wireless sensor networks. An explanation of the routing algorithm, scenarios and components of the GUI are introduced. We have analyzed the network in emergency conditions e.g. car accident, to provide a communication with the collided vehicle and warn the other entities in vicinity. The GUI also can communicate with the collided car to learn the status of the passengers. Furthermore, RSUs provide an intuition to estimate traffic congestion on the road.

As a part of the future work, the RSUs can support buffering to decrease the packet loss. Furthermore, the routing algorithm can be improved for damping the fluctuations of the routing table [9].

References

[1] G. Cookson and P. B, "Inrix global traffic scorecard," *INRIX Research, Tech. Rep.*, 2017.
[2] B. für Straßenwesen (BASt) Federal Highway Research Institute, "Traffic and accident data - summary statistics - germany," Oct. 2018.
[3] "Safespot: 'integrated project of the european union's sixth framework programme of research'."
[4] "Coopers – cooperative networks for intelligent road safety: 'integrated project of the european union's sixth framework programme of research'," *Report on ITS Reference Architecture, Austria Tech*, 2008.
[5] Zolertia, "Zolertia resources," 2016.
[6] "Contiki: The open source os for the internet of things." Accessed: 2019-29-01.
[7] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," *SIGCOMM Comput. Commun. Rev.*, vol. 24, pp. 234–244, Oct. 1994.
[8] T. Bhatia and A. Verma, "Qos comparison of manet routing protocols," *International Journal of Computer Network and Information Security*, vol. 7, pp. 64–73, 08 2015.
[9] G. He, "Destination-sequenced distance vector (dsdv) protocol," *Networking Laboratory, Helsinki University of Technology*, pp. 1–9, 2002.