Prototyping of Urban Traffic-Light Control in IoT

Jérémy Petit‡, Rafik Zitouni‡ ⋆ and Laurent George†
ECE Paris‡, VEDECOM Institute⋆,
LIGM / ESIEE Paris†,
SIC Laboratory, 37 Quai de Grenelle, 75015 Paris, France
Email: jeremy.petit@ece.fr, rafik.zitouni@ece.fr, laurent.george@esiee.fr

***Abstract*—In this work, we propose a demonstration of Urban Traffic Light Control based on an IoT network (IoT-UTLC) for smart cities. We mocked up a real crossroad by integrating a 6LoWPAN Wireless Sensor Network (WSN) to control mini traffic light panels. The network's nodes are wireless sensors and actuators interacting with an IoT Cloud Platform. MQTT Quality of Service (QoS) protocol has been implemented to manage the priority levels of exchanged data between the Cloud and WSN. Our IoT-UTLC has been found functional after verification and validation using the UPPAAL model checker.**

Keywords—Internet of Things, Smart Cities, Wireless Sensor Networks, IoT Cloud Platform, 6LoWPAN, QoS, MQTT, UPPAAL

I. Introduction

The development of Wireless Sensor Networks (WSNs) and the Internet of Things (IoT) leads to new opportunities. Smart city's projects are rising, responding to the intent of interconnecting users to their environment. The aim of IoT is to make the Internet more immersive on everyday life by using Hardware and Software components to interconnect things through the Internet. In this smart city context, we prototyped an Urban Traffic Light Control based on an IoT network architecture (IoT-UTLC). Our traffic light system would be interactive and able to connect indirectly, via Internet, traffic lights and vehicles. Our objective is to propose and demonstrate a solution for actual traffic lights, which have a static behavior. By using IoT, traffic light control would be dynamic, bringing new services and becoming application scalable, *i.e.* not limited to a simple traffic light management system. For example, we can imagine noise or air pollution measurements through panels or road's sensors.

Solutions have been proposed to make Urban Traffic Light Control (UTLC) smart and dynamic. The first idea is to use a wired installation on every crossroad with cameras to detect vehicles [1]. It involves expensive means to implement it on various intersections and cannot be deployed throughout an entire city. Moreover, research works, such as in [2], deal with only a local control of traffic lights and vehicles at intersections. Intelligent solution needs a global interconnection between road's users and infrastructure. However, we propose a wireless sensor/actuator network remotely controlled via a dedicated QoS protocol for IoT. We also set up a Cloud Interface to collect sensed data and interact with the network.

Our solution has been developed with a simple use case in mind, which is the ability to interrupt the classic cycle of traffic lights to adjust the traffic flow for certain types of vehicles. For instance, emergency vehicles or public transportation services could use an IoT based system like IoT-UTLC to avoid congestion and "claim" a prioritized access. However, several technologies can coexist for communicating between infrastructure, *e.g.* Zigbee, LoRa, SigFox, ITS-G5. Consequently, the only thing that binds them together is the Internet, which is a natural mean for heterogeneous networks inter-communications. Thus, our prototype proposes an IoT Cloud Platform in order to collect information about the traffic lights' sensors and actuators. Thanks to the IPv6 over Low power Wireless Personal AreaNetwork (6LoWPAN) [3], our WSN would be energy-efficient and IPv6 accessible.

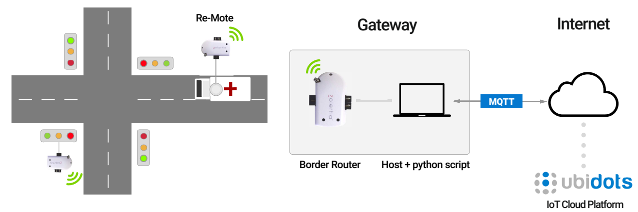

Fig. 1. Architecture of IoT-UTLC

For prototyping, students from ECE Paris engineering school [1] have built a demonstration maquette. They connect sensors, mini traffic lights panels and IEEE 802.15.4 transceivers via middleware establishing the connection to the IoT Cloud Platform. We improve the achievements of this project by integrating Message Queuing Telemetry Transport (MQTT), which is a light transport protocol, capable of doing Quality of Service (QoS). Mainly, we specified the levels of QoS managing a reliable network communication. When messages are sent without acknowledgement, our system defines the priority for crossing packets, which could have the highest level to guarantee an efficient and reliable communication.

II. Architecture and Prototype

Fig. 1 shows the architecture of our IoT-UTLC composed with three parts. From left to right, we have the WSN part with connected traffic lights' actuators, sensors and transceivers. The second part is the Gateway of the WSN ensured by the Border Router. The last part is the Ubidots

[1] Video of students' project: https://youtu.be/QottWpnE1zk

Fig. 2. Model of a crossroad for IoT-UTLC

Internet Cloud Plateform, which allows us to collect WSN information and trigger actions. Our prototype is a representation of a classical crossroad in Paris. We modeled our solution with a scale of 1:68 as shown in Fig. 2. To set up the 6LoWPAN network, we use Zolertia's Re-motes [2] coupled with Contiki OS [3]. This choice was motivated by long radio range and low power capabilities, *e.g.* two frequency bands 868-915 MHz and ISM 2.4 GHz. The Re-motes contain IEEE 802.15.4 transceivers with 6LoWPAN stack with easy Grove's input/output for sensors and actuators. A Re-mote could be driven by a computer and become a sink or a border router by being the gateway between the 6LoWPAN network and the computer.

In the second part of our architecture, a python script on the computer will run a MQTT client to connect our WSN to Ubidots. Note that MQTT insures the QoS and Publisher/Subscriber mechanism with a noteworthy topic organization. The QoS is a perfect tool to manage network resources by handling re-transmission and to guarantee the delivery of messages. The topics are strings used to filter messages and define a hierarchy of our data structure. They allow us to organize how to receive multiple data from sensors such as temperature, uptime, battery status and how to display them and obtain a real time glance of our system. In addition, the broker behaves as a server by filtering the messages and organize them in topics. It gets its messages from publishers and will send any modifications to entities, which would have subscribed to the updated topics. We used this mechanism with the middleware in order to publish messages to the broker and get from our main topic the new values using the subscriber functionality.

We have four types of programs running on Re-motes: Traffic lights, Sensors, Priority vehicles and a Border Router. Sensors will send periodically information to the IoT Cloud Platform with temperature, pressure or any relevant information that could be sensed. Traffic lights are sub-divided into two modes or roles: slaves and masters. It is necessary when all devices cooperate to avoid collision from devices on the same road. Masters would be the only one to request the middleware to change its light's color (or state) and slaves would simply change its state depending on the received packet. The working process could be described as: Master Traffic lights are sending periodically packets to request a change of state to the middleware which forwards them to Ubidots. By subscription, the middleware gets new states for the traffic lights and sends them accordingly to the master and slave traffic lights. It sends red states first and then green states to avoid unwanted behaviors.

To guarantee that our implementation behaves as a real system avoiding unexpected situations, we followed a rigor design of our system. In [4], a crossroad traffic light design has been proposed based on Petri Nets. In our work, we proposed a UPPAAL [5] model verifying formally the change of green, yellow and red states of our IoT-UTLC.

III. Findings and Arguments

Through this demonstration, we have proven the feasibility of our system. We used open-source software and cheaper devices and sensors; our system was found properly functional getting the relevant data and working cycles. We could interrupt the signal's cycle at any time and resume to it. Our demonstration would show a responsive system capable of managing future equipment of smart cities. In addition, through UPPAAL simulation, the obtained automata proved that our model worked without deadlock. It means that in our system, there is always a transition to get to the next state. This result proves that the system will not stop functioning during uptime.

Our ongoing work is to push our system to its limit and underline the benefits of MQTT. We want to see how the system responds in packet's management and distribution in stressed situations, such as increasing the number of sensed data and measure the packet's delay. We would stamp packets exchanged between Re-motes and our IoT Cloud Platform. Then we would be able to analyze the delays and data rate function of QoS levels.

IV. Conclusion

This abstract paper browsed quickly the architecture's elements and tools behind our IoT-UTLC prototyping. We explained how the IoT could be relevant to interconnect different network technologies. Getting this project feasible with a functional demonstration, we showed its usefulness for supporting more dynamic use cases in UTLC systems. As a further work, we will test the used protocols' limits. While we are still developing our project, other use cases could be defined, such as smart buildings and industrial IoT.

References

[1] "macq." [Online]. Available: https://www.macq.eu/page/smart-mobility-projects

[2] M. Tlig, O. Buffet, and O. Simonin, "Decentralized traffic management: A synchronization-based intersection control," in *2014 International Conference on Advanced Logistics and Transport (ICALT)*. IEEE, may 2014, pp. 109–114.

[3] A. Chalappuram, P. R. Sreesh, and J. M. George, "Development of 6LoWPAN in Embedded Wireless System," *Procedia Technology*, vol. 25, no. Raerest, pp. 513–519, 2016.

[4] Yi-Sheng Huang, Yi-Shun Weng, and MengChu Zhou, "Modular Design of Urban Traffic-Light Control Systems Based on Synchronized Timed Petri Nets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 530–539, apr 2014.

[5] A. David, K. G. Larsen, A. Legay, and M. Miku, "Uppaal SMC Tutorial *," vol. 269335, pp. 1–28, 2018.

[2] https://github.com/Zolertia/Resources/wiki/RE-Mote

[3] http://www.contiki-os.org/