XVI CEEL - ISSN 2178-8308
Universidade Federal de Uberlândia
05 a 09 de novembro de 2018

DATA COLLECTION IN IOT AND CONTIKI ENVIRONMENTS FOR SCADA APPLICATIONS

Gaëtan Michel[1], Remington Phelipe da Silva Correa*[1], Romain Alcesilas [1], Victor Acrani Ruivo [1], Renato Santos Carijo [1], Josué Silva de Morais [1] e Aniel Silva de Morais [1]
[1]FEELT – Universidade Federal de Uberlândia

Abstract - **The objective of this work is to propose an Internet of Things (IoT) architecture between a commonly used software for simulation of IoT devices in diverse environments and a SCADA data acquisition software responsible for the monitoring and control of the desired variables. For this, it was necessary the integration between both applications through a gateway responsible for translating the information obtained from the simulation software and forwarding them to a standard accepted by SCADA for monitoring and control.**

Keywords – Internet of Things – ScadaBR – Cooja – Contiki.

AQUISIÇÃO DE DADOS EM AMBIENTES IOT E CONTIKI PARA ACPLICAÇÕES EM SISTEMAS SCADA

Resumo - **O objetivo deste trabalho é propor uma arquitetura baseada em Internet das Coisas (IoT) entre um software comumente utilizado para simulação de dispositivos IoT em ambientes diversos e um software de aquisição de dados SCADA responsável pelo monitoramento e controle das variáveis desejadas. Para isso, foi necessário a integração entre ambas aplicações através de um gateway responsável pela tradução das informações obtidas do software de simulação e encaminhá-las para um padrão aceito pelo SCADA de monitoramento e controle.**

Palavras-Chave – Internet das Coisas – ScadaBR – Cooja – Contiki.

I. INTRODUCTION

The new trend of connecting all devices in an industrial environment with the aim of unifying all information in one place, to enable the best decisions to be made in real time with or without human intervention, will bring about a major change in the reality of industrial environments in the coming years [1]. The basis for this new revolution lies in the concept of the Internet of Things.

The Internet of Things (IoT) is based on connecting all devices, be they software or hardware, with the Internet forming a totally intelligent and automated network [1]. This allows all information to be shared between all the devices in this network, which enables devices to learn about all events occurring on the network, to make statistical forecasts and thus make decisions so that they are immune to any possible failure future [3], [1], [2]. As all devices are connected to the internet, there will be an increasing facility for an operator to remotely control the system [2].

With the growing integration of existing networks and systems with devices able to communicate with each other, there is a growing need for reliability and security of these devices. For this, the main tool available to designers is the simulators.

With the simulation tools it is possible to carry out several tests of hardware and software without the need of implementation. With these simulations it is also possible to identify problems and to ensure, based on statistical margins, satisfactory levels of safety and reliability.

That said, this work aims to propose an architecture between software simulator of IoT networks and a SCADA interface laterally used in existing systems.

The second section of this article will provide a description of the proposed structure as well as its main components.

After this, the technical concepts regarding the simulation and visualization software used will be explained and then the use of each of these will be demonstrated.

Finally, section V and VI presents the whole development process of this architecture, through validation tests of each of the parts that make up this system.

II. DESIGNED ARCHITECTURE

In this section it will be described the architecture of the project. The project is divided in 3 main parts, the first one is the IoT part including the nodes (on Contiki OS) in Cooja, the second part is ScadaBR called the human machine interface. The third and last part is about the transfer of data between the two other parts.

A. IoT – Contiki OS – Cooja simulator

This part of the architecture is composed of the several nodes running on Contiki OS in Cooja. There is a communication between the nodes and the router by using the

remingtonphelipe@gmail.com

Ipv6 protocol to send the temperature and luminosity values to the nodes.

B. ScadaBR – Human Machine Interface

The goal of this part is to provide the data from the node by using the opensource software ScadaBR. By using ScadaBR it was easy to show the evolution of the data like the light level from the nodes. But actually, there was a problem to fix about the communication. Indeed, on Cooja the nodes communicate by the Ipv6 protocol, but ScadaBR doesn't allow the capture of data from an Ipv6 source [4]. So, this is the goal of the next part to make possible the communication between Cooja and ScadaBR

C. Data Transfer

As describe in the previous section, there was a problem about the data transfer between Cooja and ScadaBR because of the different between the communication protocol. So the goal of this part is to allow the communication by using a router (exemplified on Figure 1.) and an script that can act as a gateway for both applications.

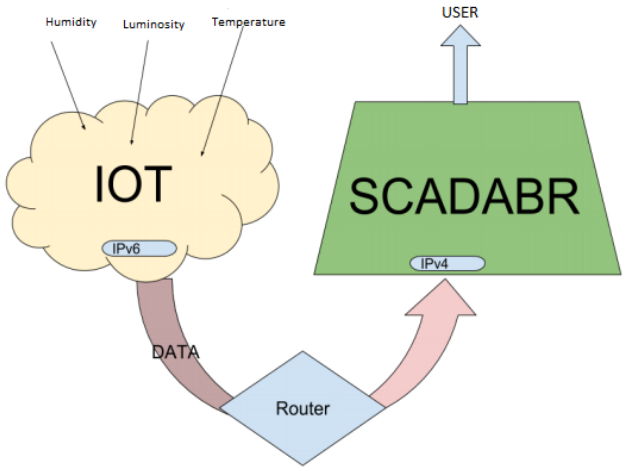


Fig. 1. Designed Project Architecture

III. TOOLS AND SOFTWARES USED

This section of this article is about the presentation of the main softwares that was used to develop the project.

A. Contiki OS

It's an operating system optimized for the use on connected objects, because of its low power consumption and its great performances because of the multi-threading. It's often used in the IoT like in the sensors wireless network. This project was built using the 2.7 version because of the great stability of this version.

B. Cooja simulator

It's a simulation platform about the connected objects called nodes or motes. This platform allows to simulate the sensors wireless network and a lot of other tools very useful, like the connection bridge between the virtual and physical IPv6 address.

C. ScadaBR

This is an open source SCADA software. It's very useful to manage the data from an information source, like the IoT for example. With this software it's possible to create an interface that make easy to view the data generated by the simulator software.

IV. CONTIKI OS USED ON COOJA

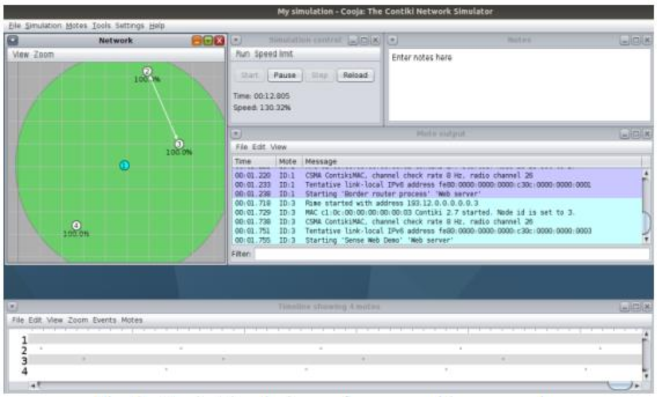
This section will introduce about how the Cooja simulator works and what was done to achieve the communication between nodes running on Contiki OS.

A. Cooja simulator

Before to test on physical support, it's important to test the performances of the system on a virtual support. To do this Cooja is very useful. Indeed, we can start the simulation and access to several tools like the mote output windows to analyze the messages exchanged between the nodes. We can have access to the time line of messages sent, we can see the range of the node in the network windows. The most useful tools for the next is the ability to connect our border router to a socket to make a connection between its virtual address and a physical address on my computer to become the nodes accessible from outside the virtual environment.

B. Nodes in Cooja

We can see on the Figure 2. that there are 4 nodes. These nodes are all the same: Z1 mote. It was used it because of the Fig. 1. Designed Project Architecture Fig. 2. Cooja Simulation software working on nodes 3 good compromise between the CPU resources and the power consumption. In the 4 implanted nodes a source code of rpl border router in the node number 1, this is this node which will communicate outside the virtual environment. The other nodes are running with a code of sky websense. All the code come from the Contiki example.



Time: 00:12.805
Speed: 130.32%

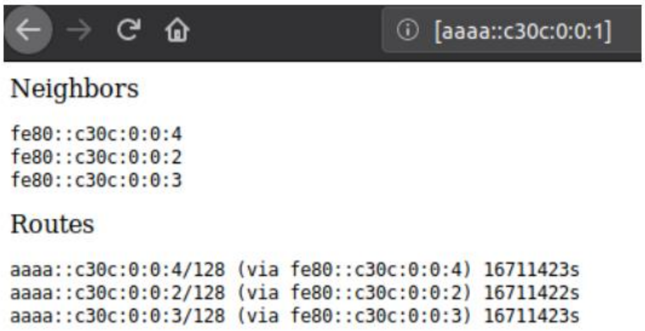
```
Time       Mote  Message
00:01.220  ID:1  CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26
00:01.233  ID:1  Tentative link-local IPv6 address fe80:0000:0000:0000:c30c:0000:0000:0001
00:01.238  ID:1  Starting 'Border router process' 'Web server'
00:01.718  ID:3  Rime started with address 193.12.0.0.0.0.0.3
00:01.729  ID:3  MAC c1:0c:00:00:00:00:00:03 Contiki 2.7 started. Node id is set to 3.
00:01.738  ID:3  CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26
00:01.751  ID:3  Tentative link-local IPv6 address fe80:0000:0000:0000:c30c:0000:0000:0003
00:01.755  ID:3  Starting 'Sense Web Demo' 'Web server'
```

Fig. 2. Cooja Simulation Software working on nodes

C. Accessibility to the node data

Like said before, Cooja allows the creation of a socket server and we can add a physical Ipv6 address to the rpl border router and then access to the data from every type of web browser on the computer. To achieve that it was used the tool *tunslip*. This tool creates a bridge between the virtual address and the physical address.

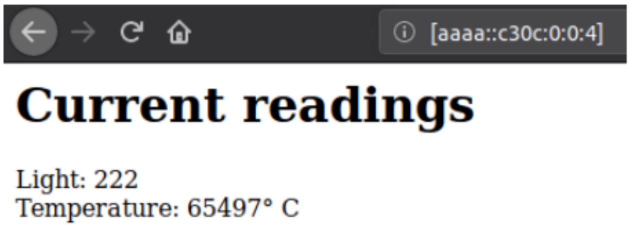
As shown by the Figure 3, the home page when we connect to the router, where we can find the address of the other nodes and some other information about the routes.



```
[aaaa::c30c:0:0:1]

Neighbors

fe80::c30c:0:0:4
fe80::c30c:0:0:2
fe80::c30c:0:0:3

Routes

aaaa::c30c:0:0:4/128 (via fe80::c30c:0:0:4) 16711423s
aaaa::c30c:0:0:2/128 (via fe80::c30c:0:0:2) 16711422s
aaaa::c30c:0:0:3/128 (via fe80::c30c:0:0:3) 16711423s
```

Fig. 3. Border Router

After that, a search was made to find the Ip of one of the other nodes to get the data from it. Indeed, we can observe the temperature and the light level on the Figure 4. The values were randomly generated just to validate the proposed architecture.



[aaaa::c30c:0:0:4]

Current readings

Light: 222
Temperature: 65497° C

Fig. 4. Simulated Node Data

V. DATA TRANSFER BETWEEN COOJA AND SCADABR

This section will detail how data transfer was performed between the Cooja and the SCADA. As previously shown, ScadaBR doesn't accept the data source from Ipv6.

A. Python Script

It was decided to write a Python script, because this language is very powerful to achieve the tasks like this. The script was only made on the shell, we just had to write the server Ipv6 address and precise the time of refresh, and the script run, it gets the data from the nodes and create a text file per node with the temperature value or the lightning level.

B. Python Script Ameliorations

The shell is not a pretty interface, and lots of people prefer to have a window with instructions. To improve the Python script, by add a configuration window. On the Figure 5 we can see the main menu of the application. As seen on the Figure 5 there are two other choices, one is about the information of this script and another about the configuration of the Ipv6 address of the router, the default one is aaaa::c30c:0:0:1.

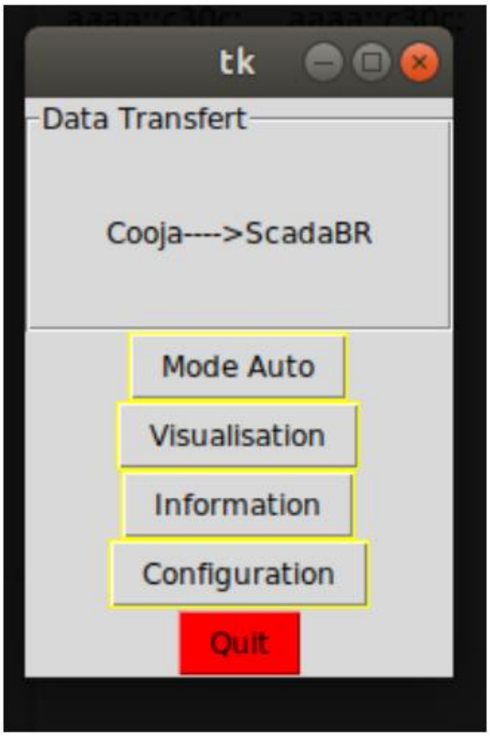


Fig. 5. Python Script Menu

VI. DATA MANAGEMENT BY SCADABR

This section will be presented the data management with ScadaBR. Indeed, we can display the node data on graphics to see the evolution in the time, or we can use a special picture for a type of data. So ScadaBR allows lots of different types of display.

These interfaces display the light level and the temperature from a node.

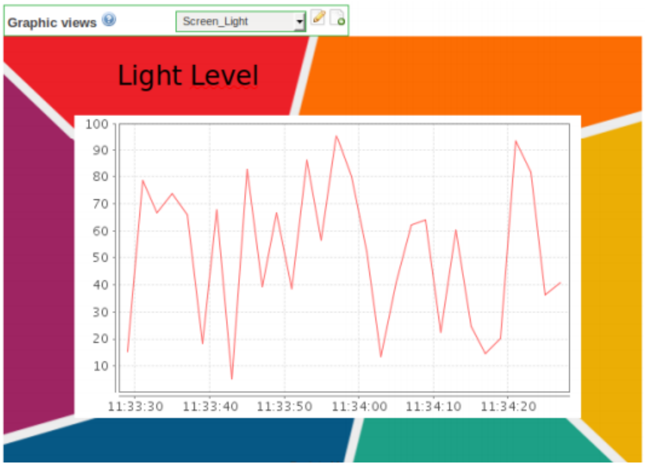
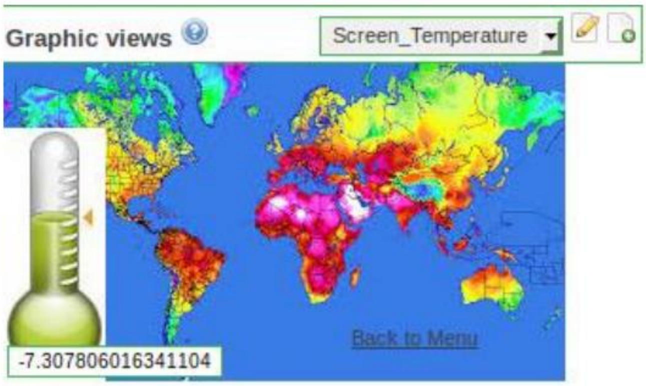


Fig. 6. Luminosity Level from a Node

On the figure 6 we can see the light level evolution from a node in Cooja, obviously the values from Cooja are randomly only for the purpose of validation.

On the figure 7 we can see another display of data from a node it's about the temperature in live, like we can see ScadaBR provide another type of display: a thermometer, with the exact value of temperature. ScadaBR allows to put some background to make the graphics view more pretty.



-7.307806016341104

Fig. 5. Temperature from a node on ScadaBR

VII. CONCLUSION

This article presents the use of an IoT simulation software and a traditional SCADA software to see data that travel in IoT networks.

At first, an architecture was proposed capable of performing the communication between the data simulator software and the SCADA capable of displaying the information in a visual way.

After that, all the tools used to develop the proposed system were detailed in order to raise compatibility problems of the protocols used to send information between the devices. With this, it was proposed the development of a script capable of acting as information gateway and translating them into a format common to the monitoring software.

Finally, validation tests were performed with the objective of generating concrete results on the proposed architecture, and validate the concept of communication between Ipv6 and Ipv4 environments;

VIII. REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions". Acedido em 23 de julho de 2018, em: http://www.sciencedirect.com/science/article/pii/S01677 39X1 3000241

[2] J. Morgan, " A simple explanation of 'Internet of Things' ". Acedido em 23 de julho de 2018, em: http://www.forbes.com/sites/jacobmorgan/2014/05/13/si mpleexplanation-internet-things-that-anyone-canunderstand/

[3] T. Paine, " Industrial Internet of Things and Communications at the Edge ",Kepware Tecchnologies, Acedido em 26 de julho de 2018, em: https://www.accenture.com/us-en/labsinsightindustrialinternet-of-things.aspx

[4] M. Daniel "Criação de Sistemas Supervisórios simples com ScadaBR e Arduino". Acedido em 01 de agosto de 2018, em: https://www.embarcados.com.br/supervisoriocom-scadabr/