

# Lab 3 - Tópicos Avançados em Redes de Computadores

Soares, Diogo\*  
Mota, Edjair

November 2019

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>O que deverá ser feito</b>	<b>3</b>
2.1	Equipe 1 - <i>End devices</i> + LoRa . . . . .	3
2.2	Equipe 2 - <i>Gateway</i> + LoRa + <i>server communication</i> . . . . .	4
2.3	Equipe 3 - <i>Broker</i> server + user application . . . . .	6
<b>3</b>	<b>O que deverá ser entregue</b>	<b>8</b>

---

\*para tirar dúvidas: diogosoaesm@ufam.edu.br ou me procure no GRCM

# 1 Introdução

À medida que a Internet das Coisas (IoT) avança, fica cada vez mais evidente que novos modelos de arquitetura de rede são necessários para comportar o modelo de negócio e dispositivos envolvidos, tais como sensores, roteadores e servidores. Um destes modelos de arquitetura de rede [7] é demonstrado na Figura 1, no qual podemos separar a comunicação em 3 etapas: a comunicação entre dispositivos (sensores) e *gateways*, usando comunicação LoRa, a comunicação entre *gateways* e servidores em nuvem ou locais, usando WiFi, Ethernet ou 3G/4G, e a etapa de disponibilização de acesso à informação para usuários e dispositivos finais através de PCs, *dashboards* de aplicação, celulares, etc.

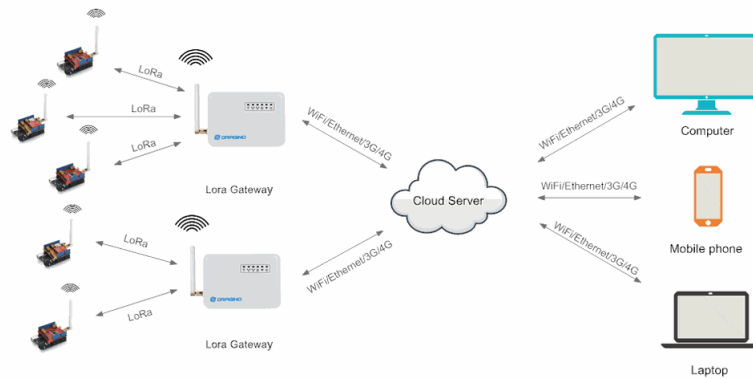


Figura 1: Etapas da comunicação com LoRa e sensores

Neste trabalho, focaremos na construção de uma arquitetura voltada para sensoriamento remoto com a utilização de comunicação LoRa entre dispositivos finais e *gateways* e comunicação infraestruturada entre *gateways* e servidores de aplicação até que a informação chegue ao usuário final que pode demandar: coleta de informações e realização de ações.

## 2 O que deverá ser feito

### 2.1 Equipe 1 - *End devices* + LoRa

A equipe 1 será incumbida de realizar as atividades relacionados aos dispositivos finais, que serão representados por dispositivos arduinos integrados a módulos LoRa.

Esta equipe terá à disposição dois arduínos, cuja suas respectivas funções são definidas a seguir:

1. **Sensor de coleta de dados:** o primeiro arduino ficará responsável por coletar as informações do sensor pH, semelhante ao utilizado no lab 2.
2. **Dispositivo atuador:** o segundo arduino será responsável por realizar alguma ação quando um usuário requisitar, também conhecido como dispositivo atuador. Com o intuito de não dificultar o experimento, é pedido que o arduino ligue ou desligue um led em uma *protoboard*.

Desse modo, sua equipe deverá implementar uma comunicação LoRa com o nó *gateway* que atenda os seguintes requisitos:

1. **Comunicação segura:** a comunicação deverá ser enviada criptografada para o nó *gateway*. Um exemplo de como isto pode ser feito pode ser encontrado na própria biblioteca Radiohead [1];
2. **Transmissão de informações:** a transmissão de informações do sensor de coleta de dados para o gateway deve ocorrer conforme é demonstrado no diagrama de sequência da Figura 2;

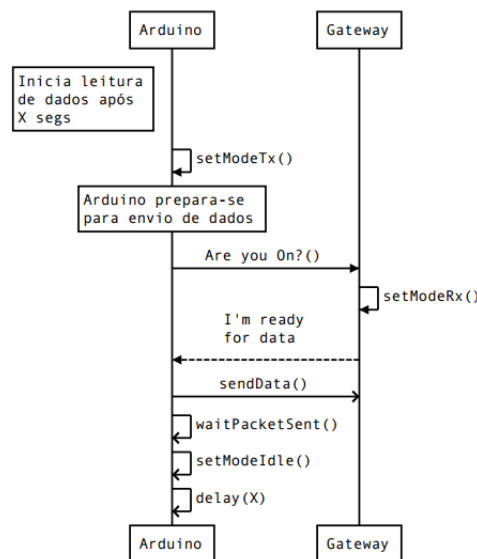


Figura 2: Diagrama de sequência do modo de transmissão de dados para o *gateway*

3. **Recepção de ações do usuário:** o dispositivo atuador deverá se comunicar com o nó gateway conforme o protocolo demonstrado pelo diagrama na Figura 3.

Tenha em mente que qualquer caso de fluxo de exceção, isto é, quando uma situação apresentada no diagrama retornar erro ou retornar um valor inesperado, este deverá ser reportado e tratado por sua equipe, e, no caso de exceções não tratadas, justificar no relatório final.

Além disso, quaisquer decisões de projeto estão livres, desde que sejam documentadas no relatório final. É necessário entregar os códigos utilizados/escritos através de *pull request* no repositório git<sup>1</sup> da disciplina e o relatório final do experimento.

<sup>1</sup><https://github.com/diogosm/TERC-2019>

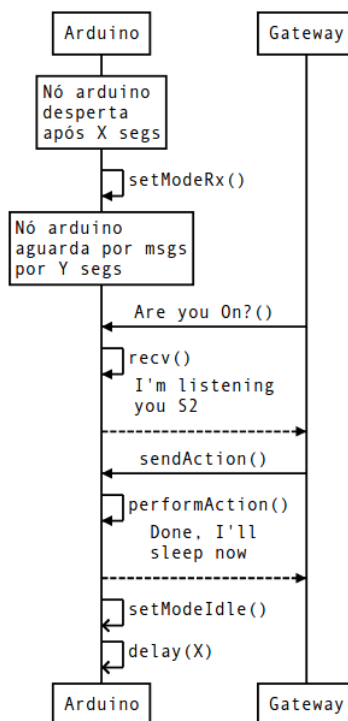


Figura 3: Diagrama de sequência do modo de recepcao de dados vindos do *gateway*

## 2.2 Equipe 2 - *Gateway + LoRa + server communication*

Esta equipe será responsável por montar a comunicação central da rede, através do nó gateway e seus dispositivos finais e do nó gateway com o servidor de aplicação. Para isto, utilizará um nó raspberry pi integrado a um módulo LoRa Dragino, com uma placa WiFi embutida e suporte a Ethernet.

Vocês deverão ser capazes de realizar comunicação com os dispositivos finais LoRa da Equipe 1 e com uma comunicação infraestruturada entre gateway e servidor de aplicação. A utilização da comunicação (WiFi ou Ethernet) é critério da equipe e a justificativa deve ser reportada no relatório final.

Sendo assim, é esperado que vocês desenvolvam um protocolo de comunicação para as seguintes situações de comunicação com o *gateway*:

1. **Comunicação LoRa:** o nó *gateway* deverá ser capaz de enviar ou receber dados dos dispositivos finais utilizando a modulação LoRa. Trabalhe em conjunto com a Equipe 1 e tenha como base que os métodos que precisam ser implementados são definidos pelos diagramas de sequência das Figuras 2 e 3.

Além disso, leve em consideração que todos os dados trafegados serão criptografados nas duas pontas. Um exemplo de como os dados criptografados podem ser tratados podem ser encontrados nas bibliotecas disponíveis na literatura em [2, 4]. Novamente, quaisquer decisões de projeto, problemas, dificuldades devem ser reportados no relatório final, contudo problemas e dificuldades sem a devida tentativa de solução serão considerados como justificativa para decremento de nota;

2. **Comunicação infraestruturada:** dado que, quaisquer dados trafegados pela rede LoRa é oriunda de uma requisição do usuário ou destinada para construção de uma base de dados, sua equipe deverá se comunicar com um servidor que será utilizado para aplicação. Para isto, você utilizará um *broker* MQTT [3, 5, 6, 8, 9], que será implementado pela Equipe 3. Desse modo, sua equipe será responsável por conectar-se ao *broker* MQTT e realizar as seguintes operações, correspondentes a cada tipo de dispositivo final e sua comunicação:

- **Coleta de dados usando o sensor de pH:** os dados do sensor chegarão via LoRa e serão encaminhados pela rede infraestruturada para o broker MQTT conforme Figura 4;

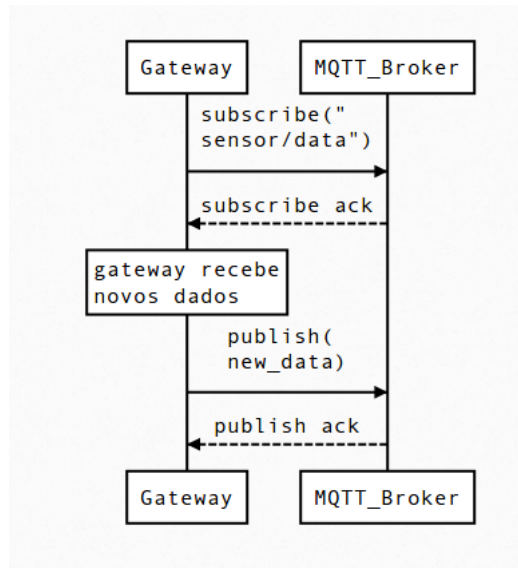


Figura 4: Diagrama de comunicação do gateway com o *broker* MQTT para envio de dados coletados pelo sensor

- **Passagem de ação para atuador:** o seu gateway deverá receber dados de *publish* do *broker* MQTT e, posteriormente, enviar para o atuador utilizando LoRa. Esta comunicação deve ocorrer do modo como é demonstrado na Figura 5.

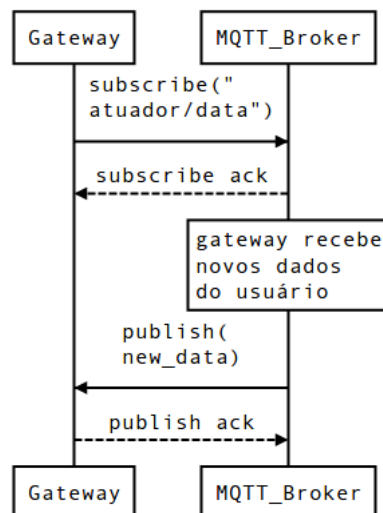


Figura 5: Diagrama de sequência do modo de recepcao de dados vindos do *gateway*

Tenha em mente que qualquer caso de fluxo de exceção, isto é, quando uma situação apresentada no diagrama retornar erro ou retornar um valor inesperado, este deverá ser reportado e tratado por sua equipe, e, no caso de exceções não tratadas, justificar no relatório final.

Além disso, quaisquer decisões de projeto estão livres, desde que sejam documentadas no relatório final. É necessário entregar os códigos utilizados/escritos através de *pull request* no repositório git<sup>2</sup> da disciplina e o relatório final do experimento.

<sup>2</sup><https://github.com/diogasm/TERC-2019>

## 2.3 Equipe 3 - *Broker* server + user application

Sua equipe será responsável pela construção, modelagem e *deploy* do *broker* MQTT [3, 5, 6, 8, 9]. Além disso, sua equipe terá que construir uma lógica que faça seu *broker* conversar com um servidor de aplicação de tal modo que o diagrama que representa o seu trabalho é apresentado na Figura 6.

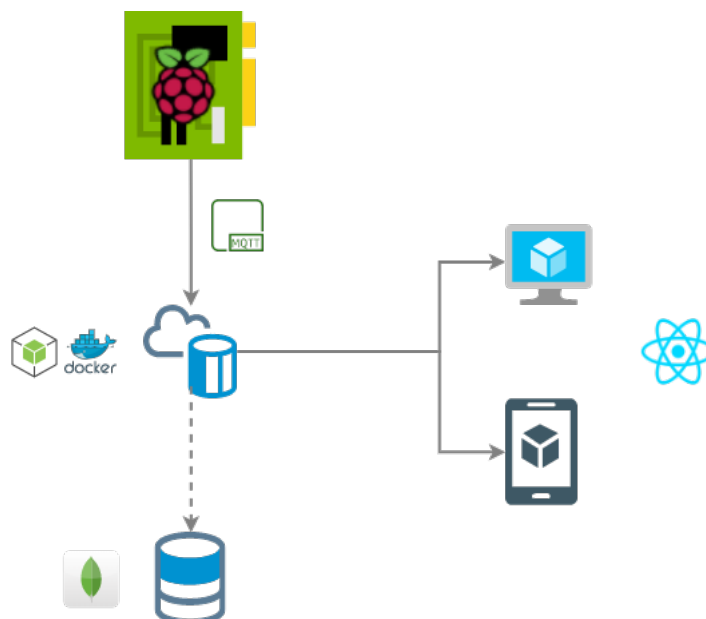
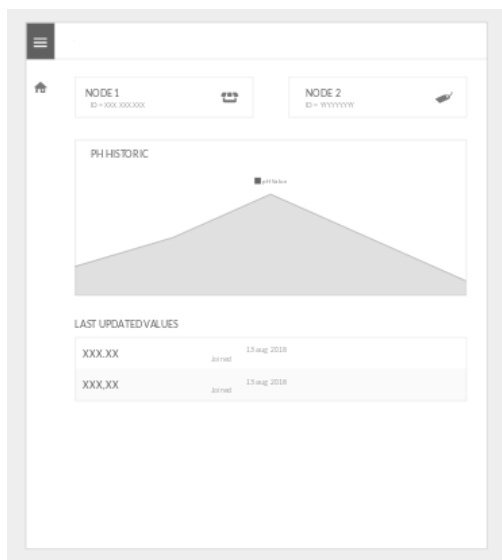


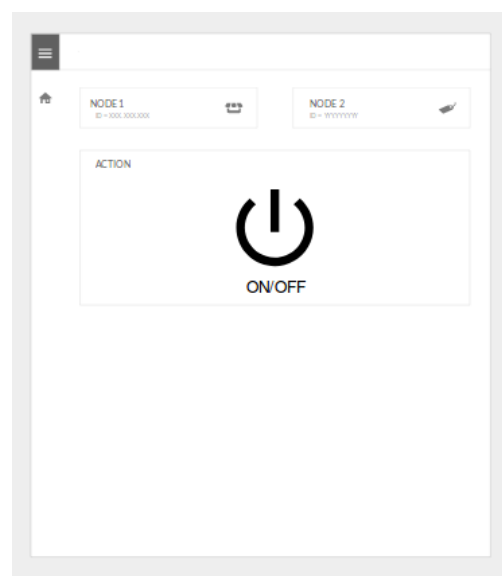
Figura 6: Diagrama opcional que representa o que deve ser entregue pela Equipe 3

Desse modo, pede-se que seja implementado:

1. **MQTT broker:** o nó *gateway* irá fazer *subscribe* no *broker*, que deverá ser o servidor que concentrará as requisições de usuário para os dispositivos atuadores e os dados coletados dos sensores. Implemente uma lógica que seja capaz de armazenar os dados coletados para gerenciamento de histórico e que garanta que cada requisição de usuário para nós atuadores tenha garantia caso o gateway esteja indisponível. Consulte a Figura 5, que demonstra a comunicação com o *gateway*, que será implementado pela Equipe 2.
2. **Database:** opcionalmente, utilize um banco de dados para armazenar os dados coletados pelo sensor como Firebase ou MongoDB. Dependendo da lógica que você implementará para o seu *broker*, este passo poderá ser opcional. Caso opte por um banco de dados, tenha em mente que ele será utilizado apenas com o sensor de dados.
3. **Aplicação pra usuário:** desenvolva uma aplicação voltada ao usuário final com 2 funções: visualizar dados do sensor de pH, incluindo o histórico e últimas coletas realizadas e alterar o comportamento do nó atuador (liga/desligar o LED). Fica a critério da sua equipe a linguagem de programação usada para desenvolver, porém, recomendamos o uso de react native, se possível. Um simples *mockup* de ambas as telas pode ser visualizado na Figura 7.



((a)) Tela 1 - captura de dados



((b)) Tela 2 - manipulando atuadores

Figura 7: Telas de usuário

### 3 O que deverá ser entregue

Ao final, gere um relatório, conforme o que é pedido para sua equipe. Todos os códigos utilizados deverão ser colocados em um repositório<sup>3</sup> git para correção. Lembre-se de realizar o *fork* do projeto, os *commits* necessários na pasta correspondente a de sua equipe (ex.: lab3/equipe1), e realizar o *pull request* para finalizar a entrega, junto com o *pdf* do seu relatório de experimentos. Descreva quaisquer dificuldades encontradas durante o experimento e lembre-se sempre de consultar qualquer referência antes de tirar qualquer dúvida.

A correção será baseado no resultado dos experimentos integrados entre os de cada equipe e na escrita do relatório apresentando a modelagem do que foi feito, detalhes de implementação/instalação/execução, desafios encontrados, metodologia de experimento, resultados e conclusões.

---

<sup>3</sup><https://github.com/diogosm/TERC-2019>



## Referências

- [1] (2019a). Lora encrypted client example using radiohead. [http://www.airspayce.com/mikem/arduino/ RadioHead/rf95\\_encrypted\\_client\\_8pde-example.html](http://www.airspayce.com/mikem/arduino/ RadioHead/rf95_encrypted_client_8pde-example.html). Accessed: 2019-12-01.
- [2] (2019b). Lora library for raspberry pi inspired by radiohead. <https://pypi.org/project/raspi-lora/>. Accessed: 2019-12-01.
- [3] (2019). Mqtt part 1 - understanding mqtt. <https://medium.com/@onur.dundar1/mqtt-part-i-understanding-mqtt-aade455baec9>. Accessed: 2019-12-01.
- [4] (2019). Radiohead main page. <https://www.airspayce.com/mikem/arduino/RadioHead/index.html>. Accessed: 2019-11-06.
- [5] Huang, A., Huang, M., Shao, Z., Zhang, X., Wu, D., and Cao, C. (2019). A practical marine wireless sensor network monitoring system based on lora and mqtt. *arXiv preprint arXiv:1906.09571*.
- [6] Mikhaylov, K., Moiz, A., Pouttu, A., Rapún, J. M. M., and Gascon, S. A. (2018). Lora wan for wind turbine monitoring: Prototype and practical deployment. In *2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 1–6. IEEE.
- [7] Raza, U., Kulkarni, P., and Sooriyabandara, M. (2017). Low power wide area networks: An overview. *IEEE Communications Surveys & Tutorials*, 19(2):855–873.
- [8] Spinsante, S., Ciattaglia, G., Del Campo, A., Perla, D., Pignini, D., Cancellieri, G., and Gambi, E. (2017). A lora enabled building automation architecture based on mqtt. In *2017 AEIT International Annual Conference*, pages 1–5. IEEE.
- [9] Standard, O. (2014). Mqtt version 3.1. 1. URL <http://docs.oasis-open.org/mqtt/mqtt/v3>, 1.