

Impacto de LoRaWAN no Desempenho de Plataformas de IoT baseadas em Nuvem e Névoa Computacional

Ivan Zyrianoff, Alexandre Heideker, Dener Ottolini, João Kleinschmidt,
Carlos Kamienski

Universidade Federal do ABC (UFABC)

{ivan.dimitry, alexandre.heideker, dener.silva, joao.kleinschmidt,
cak}@ufabc.edu.br

Abstract. *LoRaWAN is a new technology that has been consolidating as a key data communication component to send data in IoT-based systems, due to its ability to send data over long distances with low energy costs. However, literature considers only wireless aspects, disregarding its computational aspects and its integration with IoT platforms, as well as ignoring the deployment possibilities that involve cloud and fog computing. In order to understand the computational impacts of the LoRa architecture we performed a careful performance evaluation study in a complex IoT scenario, exploring cloud and fog computing scenarios and integrating with the IoT FIWARE platform. The results show that the LoRaWAN architecture is scalable, but it has impacts on system performance.*

Resumo. *LoRaWAN é uma nova tecnologia de comunicação de dados que está se consolidando como uma das principais formas de se enviar dados em sistemas de Internet das Coisas (IoT), devido a sua capacidade de envio de dados a longas distâncias e baixo gasto energético. No entanto, trabalhos na literatura consideram apenas o desempenho da comunicação sem fio, desconsiderando seus aspectos computacionais e sua integração com plataformas de IoT, além de ignorar as possibilidades de instalação em nuvem e névoa computacionais. A fim de compreender os impactos computacionais da arquitetura LoRa realizamos uma avaliação de desempenho criteriosa em um cenário complexo de IoT explorando cenários de computação em nuvem e névoa e integrando com a plataforma FIWARE de IoT. Os resultados mostram que a arquitetura LoRaWAN é escalável, mas possui impactos no desempenho do sistema.*

1. Introdução

Atualmente há um crescimento exponencial de dispositivos conectados à Internet e espera-se que bilhões de sensores sejam conectados na próxima década [Ericsson, 2016]. Isso tem permitido o surgimento de uma nova geração de aplicativos e serviços inteligentes para o benefício da sociedade em diversos setores, como o urbano e o da agricultura. Um dos desafios é como obter os dados dos sensores, que possuem requisitos como: baixa taxa de transferência, longo alcance, baixo atraso e baixo consumo de energia; que não são totalmente atendidos pelas tecnologias de rede atuais.

Para atender essa nova demanda foram introduzidas as redes sem fio de baixa potência (LPWAN) como um conjunto de novas tecnologias adequadas para IoT [Raza

et al. 2017]. Dentre as tecnologias LPWAN, a tecnologia *Long Range* (LoRa) está se consolidando como uma das principais formas de se enviar dados em sistemas de Internet das Coisas (IoT), devido a sua capacidade de envio de dados a longas distâncias e baixo gasto energético. LoRa define a camada física que fornece o enlace de comunicação de rádio enquanto LoRaWAN estabelece o protocolo de comunicação das camadas superiores (como acesso ao meio e segurança) e a arquitetura do sistema.

Apesar de estar despontando como uma das principais tecnologias existentes, os trabalhos na literatura consideram apenas o desempenho da comunicação sem fio do LoRaWAN, desconsiderando seus aspectos computacionais além da sua integração com plataformas de IoT. Para implementar a tecnologia LoRaWAN, no mínimo dois módulos de software são necessários: o LoRa Network Server e o LoRa App (Application) Server, além de seus respectivos bancos de dados para persistência de informações. Esses elementos possuem potencial para tornarem-se gargalos e inviabilizarem o uso da tecnologia em determinados cenários. Outras implementações usam a modulação LoRa, mas não usam toda a arquitetura de referência LoRaWAN, que inclui segurança, configuração e gerenciamento.

Esse trabalho visa preencher essa lacuna, analisando o impacto computacional dos módulos de software necessários para se implementar uma arquitetura LoRaWAN e sua integração com a plataforma de IoT FIWARE¹. Foram avaliadas diferentes configurações de uma comunicação entre sensores emulados e a nuvem computacional, envolvendo também o uso de névoa computacional. Para a emulação dos sensores se comunicando através de LoRaWAN foi necessário o desenvolvimento de um gerador capaz de implementar as opções de criptografia envolvendo a troca de chaves, acoplado à ferramenta SenSE [Zyrianoff, et. al. 2017], um gerador escalável de dados de sensores. A principal hipótese desse trabalho é que o uso da arquitetura LoRaWAN adiciona várias características importantes para o gerenciamento de redes IoT de larga escala e ao mesmo tempo adiciona uma sobrecarga computacional dentro de limites aceitáveis.

Os resultados obtidos esclarecem diversos pontos anteriormente nebulosos sobre a escalabilidade e o impacto computacional dos módulos de software da arquitetura LoRaWAN. A tecnologia é escalável e possui um bom desempenho. No entanto, há impactos no desempenho do sistema na presença desses módulos, em especial no tempo de processamento em cargas de trabalho baixas (aumento de aproximadamente 200 ms). Embora para a maioria das aplicações essa diferença no tempo de processamento não seja significativa, ela pode inviabilizar serviços que necessitem de processamento em tempo real com tempos de resposta muito baixos, como um serviço de semáforos inteligentes. No entanto, para a grande maioria das aplicações, vale a pena adicionar a arquitetura LoRaWAN à comunicação básica LoRa para obter as vantagens mencionadas de segurança, configuração e gerenciamento.

O restante do artigo está organizado como segue: na Seção 2 está presente o estado da arte, a seção 3 apresenta os trabalhos relacionados; na Seção 4 está descrita a nossa implementação da arquitetura LoRaWAN; a seção 5 trata da metodologia adotada no projeto enquanto os resultados são apresentados na seção 6 e discutidos na seção 7 e por fim são apresentadas as conclusões do trabalho.

¹ fiware.org

2. Estado da Arte

2.1. LPWAN e LoRaWAN

Nos últimos anos foram desenvolvidas várias tecnologias de comunicação sem fio para IoT conhecidas como LPWAN (*Low Power Wide Area Networks*) [Raza et al. 2017]. Estas redes possuem longo alcance de transmissão, baixo consumo de energia e baixa taxa de dados. Estas características tornam as redes LPWAN atrativas para aplicações de IoT que enviam poucas dezenas de bytes de dados a cada poucos minutos ou horas, como iluminação pública, monitoramento de poluição, *smart grids* e agricultura. Redes LPWAN possuem topologia estrela, em que vários sensores enviam dados diretamente a um concentrador, geralmente chamado de gateway, que possui conexão com a Internet. A tecnologia LoRa vem sendo bastante estudada na literatura e testada em aplicações práticas, já que a infraestrutura da rede pode ser instalada pelos próprios desenvolvedores das aplicações, ao contrário de outras tecnologias.

LoRa é um protocolo desenvolvido pela *Semtech Corporation* que opera nas faixas de frequência sub-GHz e permite atingir alcances de transmissão de até vários quilômetros com baixo consumo de energia. A camada física é chamada de LoRa e as camadas superiores formam o padrão LoRaWAN (padrão aberto desenvolvido pela LoRa Alliance) [Sornin et al. 2016].

2.3. Computação em Névoa

Computação em névoa (*fog computing*) é um paradigma relativamente novo, que visa lidar com desafios relacionados à enorme quantidade de dados que serão gerados com o aumento de dispositivos conectados à Internet [Bonomi, et. al. 2012]. A computação em névoa traz inteligência a dispositivos na borda da rede, solucionando sérios problemas como a diminuição da latência de sistemas de tempo real, diminuição do tráfego de dados entre a borda da rede e o núcleo, e diminui a carga de processamento na nuvem executando um processamento hierárquico dos dados.

2.4. Plataforma FIWARE

A plataforma FIWARE tem atraído a atenção de desenvolvedores de sistemas IoT no mundo inteiro por ser uma solução de código aberto financiada e incentivada pela Comissão Europeia. Ela é composta de uma série de componentes chamados de Generic Enablers (GE) que executam funções necessárias para uma variedade de aplicações, muitas voltadas para o uso de IoT em ambientes de sociedades inteligente.

O conceito do FIWARE implica em utilizá-lo para compor um sistema com uma arquitetura de micro-serviços, em que cada GE desempenha uma função e troca automaticamente mensagens estruturadas em um determinado padrão (OMA NGSI *standard*²) com outros GEs ou aplicações de terceiros por uma API REST bem definida. Dessa maneira, o FIWARE disponibiliza diversos desses blocos, os quais os desenvolvedores selecionam e os integram para criar o sistema que desempenhe as tarefas que necessitam. Entre os vários GEs disponíveis na plataforma FIWARE, alguns desempenham papel essencial para viabilizar aplicações inteligentes para IoT, como:

² openmobilealliance.org/release/NGSI

- Orion Context Broker³: Orion é um broker de informações de contexto baseado no padrão de comunicação *publish/subscribe*, sendo o principal GE da FIWARE. Ele oferece uma interface para clientes registrarem entidades de contexto e seus atributos, assim como produtores/consumidores dessas entidades. Na prática, o Orion é um grande distribuidor de dados que desacopla os transmissores dos receptores.
- IDAS e IoT Agent⁴: IDAS é uma implementação do GE chamado *Device Backend Management* que contém vários componentes IoT Agent que mapeiam dados vindos de dispositivos IoT no modelo de informação FIWARE NGSI. Componentes IoT Agent são conectados ao Orion para o qual enviam e do qual recebem dados.

O uso do FIWARE envolve a instalação de seus GEs numa infraestrutura apropriada para executá-los, que podem ser máquinas isoladas, nuvens públicas ou privadas usando diferentes controladores, como OpenStack⁵ ou OpenNebula.

3. Trabalhos Relacionados

Diversos trabalhos recentes fizeram testes experimentais e analisaram o desempenho de enlaces LoRa em diferentes cenários, como ambientes internos e urbanos [Ayele et al. 2017] [Magrin et al. 2017] [Georgiou & Raza 2017] [Petäjäjärvi et al. 2017] [Van den Abeele et al. 2017]. Estes trabalhos analisaram a influência dos parâmetros de camada física na taxa de entrega de pacotes, atraso e consumo de energia. Também há vários trabalhos na literatura que analisam o desempenho de redes LoRaWAN para verificar a escalabilidade e desempenho da rede com vários dispositivos na rede (nós sensores e gateways). Devido à dificuldade de implementar redes experimentais com dezenas ou centenas de dispositivos, estes trabalhos usam modelos analíticos e simuladores para a análise. Em geral, os trabalhos na literatura consideram apenas o desempenho da comunicação sem fio do LoRaWAN (dispositivos até o gateway), no entanto, a arquitetura LoRaWAN possui outros componentes, como o servidor de rede LoRa (*LoRa Network Server*) e aplicação LoRa (*LoRa App Server*), que não têm sido objeto de estudo na literatura em termos de desempenho e escalabilidade (podem agregar dados de dezenas de gateways, por exemplo).

Escalabilidade é uma característica fundamental para plataformas de IoT, já que visam gerenciar milhares de dispositivos enviando dados continuamente e o desempenho do FIWARE vem atraindo atenção da sua comunidade de usuários. Um estudo abrangente que propõe métricas qualitativas, quantitativas e avalia o desempenho de várias plataformas de IoT é apresentado por da Cruz et al. (2018). No entanto, eles adotaram uma abordagem genérica, sem considerar a integração com LoRaWAN e apenas uma única configuração de plataforma foi considerada.

Martínez et al. (2016) fornece uma descrição detalhada da arquitetura de um *testbed* da plataforma FIWARE configurada para o cenário de agricultura de precisão, no entanto, todos os aspectos de integração com sensores foram abstraídos, já que os

³ fiware-orion.readthedocs.io

⁴ catalogue-server.fiware.org/enablers/backend-device-management-idas

⁵ openstack.org

dados eram enviados diretamente para o FIWARE utilizando usando a API REST do Orion Context Broker. Nosso trabalho se difere dos demais citados já que ele considera aspectos de encriptação, empacotamento e envio dos dados pelo sensor com o respectivo impacto gerado na infraestrutura computacional, considerando uma plataforma de IoT e conceitos de computação em névoa.

Resultados de experimentos de avaliação de desempenho da plataforma FIWARE em cenários de agricultura foram apresentados em Zyrianoff et al. (2018) e Kamienski et al. (2019). No entanto, esses artigos não incluem a avaliação de componentes da arquitetura LoRaWAN, nem sua comparação com o uso da comunicação LoRa básica, como é mostrado aqui.

4. Implementação da Arquitetura LoRaWAN

LoRaWAN define o protocolo de acesso ao meio para que vários dispositivos possam se comunicar com um gateway que usa a modulação LoRa, assim como define mecanismos de segurança, como autenticação e criptografia. Os dados em um pacote podem variar até 255 bytes, atingindo taxas de dados de até 50 Kbps. LoRaWAN define também o mecanismo ADR (*Adaptive Data Rate*) em que o servidor LoRa pode alterar parâmetros de configuração dos dispositivos para otimizar a taxa de transmissão, tempo no ar e consumo de energia [Sornin et al. 2016].

A implementação típica de uma rede LoRaWAN é mostrada na Figura 1. O *end node* possui uma aplicação desenvolvida para transmitir dados com a tecnologia LoRa, usando uma camada de abstração de hardware (HAL) e o protocolo SPI (*Serial Peripheral Interface*) para comunicação do microcontrolador com os diversos componentes do sensor. A camada física gera o pacote LoRaWAN e transmite para o gateway, que encaminha os dados para o servidor de rede via protocolo MQTT – um protocolo típico de IoT. O servidor de rede LoRaWAN envia as informações a um servidor de aplicação LoRaWAN.

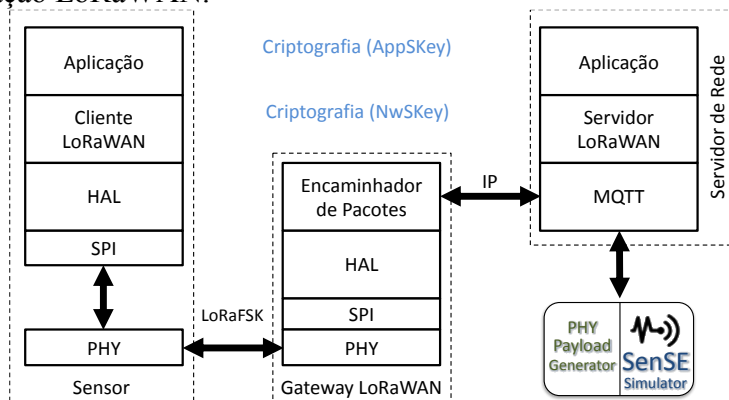


Figura 1. Arquitetura LoRaWAN e ponto de injeção dos dados gerados

LoRaWAN permite que pacotes sejam autenticados e encriptados com o algoritmo AES [Sornin et al. 2016]. A chave *Network Session Key (NwkSKey)* é conhecida pelo nó sensor e o servidor de rede LoRa, sendo usada para verificação de integridade das mensagens. A chave *Application Session Key (AppSKey)* deve ser conhecida pelo nó sensor e o servidor de aplicação LoRaWAN e é usada para encriptação e decriptação dos dados. Existem dois métodos para implementar estas chaves nos dispositivos LoRa:

ativação pelo ar (OTAA – *Over-the-Air Activation*) e ativação por personalização (ABP – *Activation by personalization*) [Sornin et al. 2016], utilizada nesse trabalho.

Para realizar a avaliação de desempenho da arquitetura LoRaWAN, é necessário gerar o tráfego de uma grande quantidade de dispositivos de IoT, enviados para um servidor de rede LoRa. Para este fim, foi utilizado o gerador de tráfego SenSE (*Sensor Simulating Environment*) e o programa PHYPayload para geração do *payload* de camada física. Como não existem geradores de *payload* para comunicação LoRaWAN disponíveis adequados à essa demanda, o programa PHYPayload foi desenvolvido especialmente para esse artigo, mas pode ser utilizado em outras avaliações por terceiros. PHYPayload utiliza as mesmas chaves de criptografia utilizadas pelo sensor LoRaWAN para criar um *payload* com a mesma estrutura e integridade de dados de um *payload* real.

A implementação da arquitetura LoRaWAN com seus servidores de rede e de aplicação pode ser feita de algumas formas diferentes: a) Software de código livre que pode ser instalado na nuvem ou na névoa de acordo com o cenário desejado, como o LoRa Server Project⁶; b) The Things Network⁷ (TTN), uma comunidade viabilizada por *crowdfunding* que implementa os servidores LoRaWAN e os disponibiliza através de um serviço web. No entanto, alguns serviços da TNN são pagos; c) Soluções proprietárias e pagas, como Loriot⁸; d) Desenvolvimento de nova implementação para LoRaWAN, que representaria um retrabalho, uma vez que existem outras opções disponíveis.

Neste trabalho foi utilizada a implementação disponibilizada pelo *Lora Server Project* (referenciado como módulos LoRaWAN ou loraserver no restante do trabalho), pois é de código aberto, gratuita e pode ser implementada em uma rede privada. Foram utilizados os seguintes componentes do LoRa Server Project: LoRa Server (servidor de rede) e LoRa App Server (servidor de aplicação). Para banco de dados, foi utilizado REDIS⁹ e Postgres¹⁰.

Nesse trabalho os *end-nodes* e *gateways* LoRa são abstraídos pela ferramenta SenSE com a adição do programa PHYPayload desenvolvido especialmente para essa finalidade. A abstração dos sensores e gateways não prejudica a comprovação da principal hipótese de pesquisa, uma vez que o objetivo é avaliar o comportamento e efeito dos componentes computacionais da arquitetura LoRaWAN numa implementação de IoT, e não os aspectos de modulação comunicação da camada física. Além disso, não existem dados disponíveis de milhares de sensores LoRaWAN para avaliar o desempenho e escalabilidade da solução.

5. Metodologia

O objetivo principal do trabalho é entender o impacto computacional da arquitetura de software do LoRaWAN, compará-la com o uso da comunicação LoRa básica (sem

⁶ loraserver.io

⁷ thethingsnetwork.org

⁸ loriot.io

⁹ redis.io

¹⁰ postgresql.org

LoRaWAN) e sua integração com plataformas de IoT. Para isso, foram executados experimentos com diferentes configurações da plataforma de IoT distribuídos na nuvem e na névoa e com LoRaWAN e LoRa básico – ou seja, com e sem os módulos do loraserver. A Figura 2 ilustra todas as configurações exploradas.

5.1. Cenário e Ambiente dos Experimentos

Um dos objetivos principais desse trabalho é entender o impacto computacional da arquitetura de software do LoRaWAN e sua integração com plataformas de IoT. Para isso avaliamos duas configurações diferentes: utilizando LoRaWAN (arquitetura de software loraserver - configurações A e B da Figura 2) e sem LoRaWAN (configurações C e D da Figura 2), na qual o gateway LoRa envia dados diretamente para a aplicação. A importância de tal comparação se justifica devido ao pouco tempo de existência da tecnologia e a de avaliações em grande escala. Muitas instalações menores dispensam o uso da arquitetura LoRaWAN considerando-a um *overhead* desnecessário para a escala da implementação.

O ambiente desses experimentos é baseado em cenários reais de sociedades inteligentes, modelados com base em pilotos do projeto SWAMP¹¹, uma colaboração entre instituições e empresas do Brasil e da Europa que visa desenvolver métodos e abordagens baseados em IoT para a irrigação de precisão na agricultura, com um foco experimental [Kamiński et. al. 2019]. Apesar dos cenários avaliados serem baseados em ambientes de agricultura inteligente, acreditamos que não possuam características muito específicas, tornando os resultados potencialmente adequados para ambientes urbanos e sistemas baseados em IoT de modo geral que usem LoRaWAN. É possível fazer essa afirmação uma vez que LoRaWAN e névoa são tecnologias adequadas para diferentes cenários de IoT. Além disso, a camada física, que poderia trazer diferenças, por exemplo entre cidades e campo, não está sendo avaliada.

Além da diferença de configuração envolvendo LoRa, também avaliamos duas formas diferentes de implantação da plataforma de IoT FIWARE: 1) Configurações A e C da Figura 2: toda a plataforma FIWARE está apenas na nuvem computacional, sendo que o apenas módulos do loraserver estão na névoa; 2) Configurações B e D da Figura 2: parte da plataforma está na névoa computacional, dessa maneira, dados podem ser consumidos localmente sem a necessidade de ter que se conectar através de uma WAN. A Figura 2 mostra o fluxo de dados entre os componentes de software avaliados e as diferentes formas para sua instalação em gateway, névoa e nuvem são apresentadas nas figuras subsequentes.

O *trade-off* entre ter a plataforma de IoT somente na nuvem ou distribuída entre a névoa e a nuvem é particularmente interessante de ser explorado. Por um lado, adicionamos mais módulos de software, que potencialmente podem comprometer o desempenho do sistema, todavia, esses módulos na névoa são importantes em cenários de agricultura inteligente, pois a conexão entre a fazenda e a internet é muito instável e dessa maneira serviços locais ainda podem funcionar normalmente em caso de desconexões com a nuvem. Em ambientes urbanos, o nó da névoa configurado desse modo diminui consideravelmente a latência do serviço, possibilitando aplicações em

¹¹ swamp-project.org/

tempo real. Muitos dos módulos de software apresentados na Figura 2 já foram apresentados nas seções anteriores, no entanto, alguns são novos. São eles:

- **Mosquitto:** Eclipse Mosquitto é um *broker open source* de mensagens MQTT. Ele é utilizado inclusive internamente pelo LoRa Server Project para comunicação entre os servidores e pela The Things Network para instalação de redes privadas LoRaWAN¹²;
- **FIWARE-based IoT Agent:** No presente momento, não há uma versão oficial do IoT Agent que suporte a comunicação LoRaWAN. Devido a isso, nossa equipe programou um novo IoT Agent, capaz de mapear as mensagens LoRaWAN enviadas pelo LoRa App Server até o Orion Context Broker. Esse módulo foi programado em JavaScript, executado pela *engine* do Node.js;
- **Consumer:** O Consumer é uma aplicação web Express.js que se inscreve no Orion e recebe dados dos sensores. Quando o Orion envia uma mensagem para a API do consumidor, um registro de data e hora é feito e é subtraído do *timestamp* de geração da mensagem.
- **SenSE:** um gerador de tráfego de sensores IoT em larga escala, capaz de abstrair dispositivos reais e modelar cenários complexos, como *smart farms* e *smart cities*. Embora os sensores sejam sintéticos, o tráfego é real [Zyrianoff, et. al. 2017].

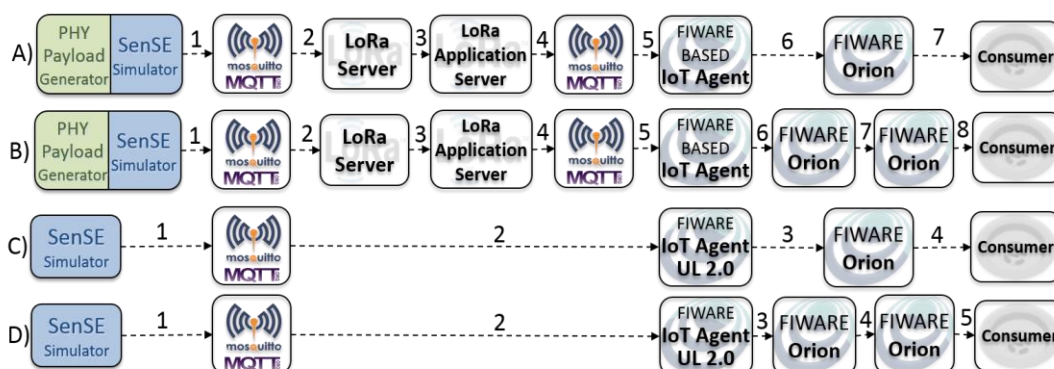


Figura 2. Diferentes fluxos de dados das configurações dos experimentos

Em todas as configurações, os módulos foram encapsulados em contêineres Docker. As Figuras 3, Figura 4, Figura 5 e Figura 6 mostram os cenários dos experimentos para cada configuração, em detalhes, respectivamente. Para emular um ambiente mais próximo do real, os experimentos foram feitos em um ambiente de nuvem privada (Openstack Queens), onde instanciou-se uma máquina virtual (VM) para o SenSE, outra representando o nó da névoa e uma última representando a nuvem computacional. Para simular a distância entre o nó da névoa e a nuvem, utilizou-se um emulador de rede, o WANem¹³. As VMs foram configuradas de acordo com os padrões de máquinas virtuais da Amazon AWS, sendo que para a nuvem foi usada uma máquina

¹² thethingsnetwork.org/article/setting-up-a-private-routing-environment

¹³ wanem.sourceforge.net/

t2.medium (4GB e 2vCPUs) e para representar o nó da névoa *t2.small* (2GB RAM e 1vCPU), já que a névoa possui recursos computacionais limitados. Devido a quantidade de módulos na névoa na configuração B foi necessário dividir entre dois nós de névoa.

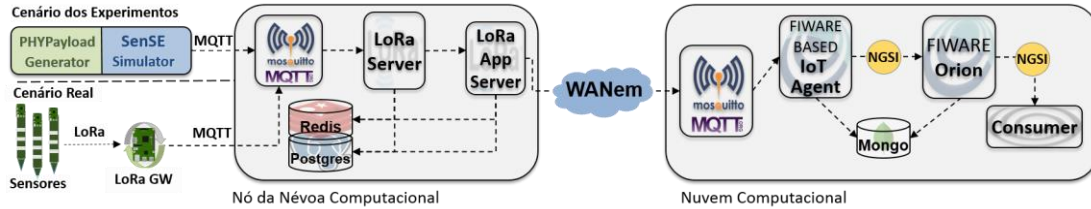


Figura 3. Cenário da configuração (a)

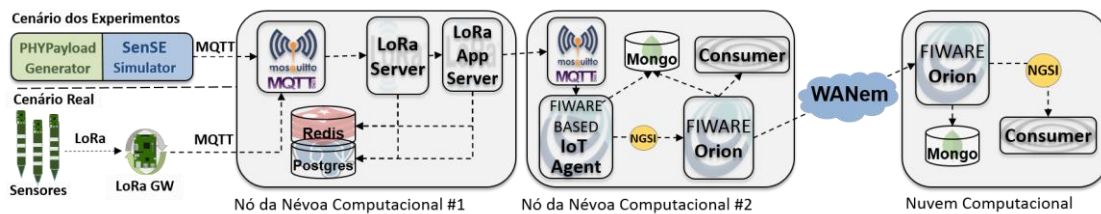


Figura 4. Cenário da configuração (b)

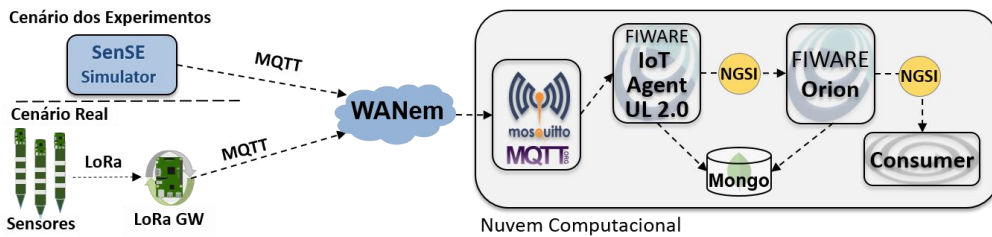


Figura 5. Cenário da configuração (c)

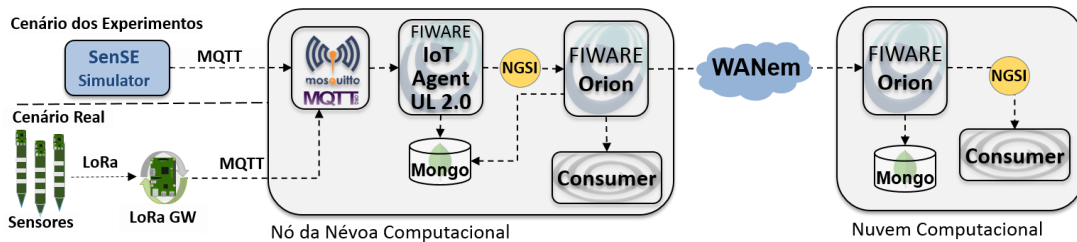


Figura 6. Cenário da configuração (d)

Em todos os experimentos, o SenSE emula os sensores e o LoRa gateway. Nos cenários utilizando a arquitetura de software LoRaWAN (Figura 3 e Figura 4), o SenSE utiliza o PHYpayload Generator para gerar os dados que enviará para o LoRa Server, que irá realizar a autenticação e o decriptação dos dados. Nos experimentos em que os dados são enviados diretamente para o FIWARE (Figura 5 e Figura 6), não é necessário gerar o PHYPayload. Ressalta-se mais uma vez que a motivação para cenários com e sem a arquitetura LoRaWAN é a comparação com implementações simples que usam apenas a comunicação LoRa na camada física entre o *end-node* (sensor) e o gateway, e o gateway envia dados diretamente para um *broker* MQTT, sem os recursos de controle de acesso ao meio, segurança e autenticação.

Para entender completamente o cenário de avaliação, é necessário compreender o modelo de dados das mensagens enviadas pelo SenSE em cada cenário. Para isso foi adotado o protocolo Ultralight 2.0 (UL) - um protocolo leve baseado em texto destinado a dispositivos e comunicações restritos onde a largura de banda e a memória do dispositivo podem ser escassos – para as mensagens enviadas diretamente para a arquitetura FIWARE. Nesse caso, cada sensor envia uma mensagem que consiste em sete valores ($t_1|v|t_2|v|t_3|v|m_1|v|m_2|v|m_3|v|c|v|t_s|v$, no protocolo UL, sendo que o v representam valores gerados aleatoriamente), simulando uma sonda de solo para agricultura [Zyrianoff, et. al. 2018], compondo um *payload* de 65 bytes. No caso do envio para o loraserver, é necessário enviar muitos outros dados com informações dos *end-nodes*, como modulação LoRa, gateway e segurança, então, apesar dos sensores simulados apenas enviarem um único valor, o *payload* é muito maior do que nos experimentos sem o loraserver, consistindo de 314 bytes.

5.2. Métricas

Dois conjuntos diferentes de métricas foram utilizadas nos experimentos:

- Tempo médio de processamento: o tempo médio decorrido desde que o dado do sensor é gerado até atingir o Consumer. Aqui avalia-se o tempo decorrido desde que os dados são gerados até que os mesmos estejam prontos para serem usados por outro aplicativo - como um *dashboard*, um módulo de *analytics* ou um tipo de *reasoner* que irá atuar em algum dispositivo.
- Métricas de sistema: foi coletado uso de CPU e memória para cada container Docker. Os dados são registrados com uma periodicidade de 5 segundos.

A análise da escalabilidade das diferentes configurações decorre diretamente do efeito do aumento da carga de trabalho (descrita a seguir) sobre o tempo médio de processamento e a completude dos experimentos.

5.3 Experimentos

Cada experimento consiste do SenSE simulando sensores e os enviando a um *broker* MQTT, que pode estar conectado a uma arquitetura LoRaWAN ou diretamente a uma plataforma de IoT. Para entender a escalabilidade do LoRaWAN, foram realizados experimentos com uma grande quantidade de sensores enviando dados simultaneamente, cada sensor envia uma mensagem a cada 10 minutos. Foram aplicadas em cada configuração (da Figura 6) quatro diferentes cargas de trabalhos: 1.000, 5.000, 10.000 e 15.000 sensores enviando dados, totalizando 16 experimentos, sendo que cada experimentos foi replicado 30 vezes. O intervalo de confiança assintótico calculado foi de 99%. As condições de rede emuladas pelo WANem nos experimentos foram obtidas replicando uma conexão de uma rede 4G até uma nuvem computacional, os parâmetros configurados foram de uma conexão de 10 Mbps com uma latência de 45 ms e 5 ms de *jitter*. O uso de 4G em ambientes rurais é comum, principalmente em casos onde a comunicação vai diretamente do campo (gateway LoRa) até a nuvem, sem passar pela sede da fazenda. A tecnologia 4G também é usado em cidades, quando sensores e gateways estão instalados em locais sem o auxílio de outras conexões com ou sem fio à Internet.

6. Resultados

Os resultados para o tempo médio de processamento em todas as configurações apresentadas na Figura 2, estão apresentadas na Figura 7. Em geral, os resultados mostram que utilizando ou não a arquitetura LoRaWAN o sistema permanece escalável e com bom desempenho, exceto nos experimentos com 15.000 sensores enviando dados continuamente. Sendo que apenas a configuração A não apresentou falhas quando submetida a essa carga de trabalho, as configurações C e D caíram durante esses experimentos e a configuração B cai logo após o início dos experimentos. O valor numérico máximo do eixo y foi definido como 600 para melhor visualização, no entanto, as configurações C e D apresentaram tempos de: $676,80 \pm 457,23\text{ms}$ e $17429.89 \pm 36165,70\text{ms}$, respectivamente.

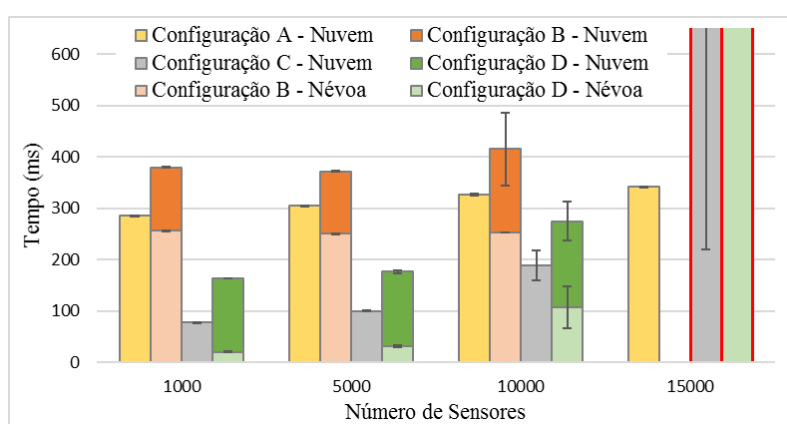


Figura 7. Tempo de processamento de cada configuração

Com relação a comparação dos experimentos utilizando o LoRaWAN (configuração A e B) e sem utilizá-lo (configuração C e D), observa-se que, nas cargas de trabalho baixas, a arquitetura LoRa adiciona cerca de 200ms ao processamento. Esse aumento no tempo de processamento era esperado, pois se adiciona módulos ao fluxo de dados que tratam questões de acesso ao meio, segurança e autorização - que demandam processamento.

Um resultado não intuitivo é a estabilidade e escalabilidade da configuração A, a única que não apresentou problemas na mais alta carga de trabalho. Isso acontece porque a arquitetura LoRaWAN controla melhor o fluxo de dados do que o IoT Agent UL. Nas configurações em carga alta em que os dados são enviados diretamente para IoT Agent UL em a plataforma cai, podemos observar o aumento exponencial do uso de recursos do IoT Agent UL na Figura 9. A plataforma também apresenta falhas na configuração B devido quantidade de módulos computacionais sendo executadas na névoa (que possui recursos computacionais limitados).

As configurações B e D são as únicas nas quais é possível consumir dados diretamente da névoa, possibilitando que aplicações processem parte dos dados antes da nuvem computacional e, portanto, responder a mudanças em tempo real. Na Figura 7 vemos que a maior parte do tempo processamento da configuração B está concentrado na névoa – devido aos módulos LoRaWAN – enquanto o oposto ocorre para a configuração D. Para determinadas aplicações em que um baixo tempo de resposta é essencial a arquitetura LoRaWAN pode não atender as restrições.

A Figura 8 mostra os resultados para recursos de sistema (uso de CPU e memória) para a configuração B. Por limitações de espaço, só iremos apresentar as métricas de sistema para essa configuração, mas não há grandes variações entre as configurações. O FIWARE Orion é o módulo que mais usa recursos do sistema, o que é esperado pois ele lida com diversas operações sobre os dados na forma mais verbosa do fluxo de dados (NGSI JSON). Não vemos crescimento do uso de memória de 10.000 para 15.000 sensores pois o fluxo de dados está limitado a taxa de serviço dos módulos LoRaWAN, sendo que o resto das mensagens são enfileiradas, dessa maneira, o uso desse recurso se mantém o mesmo.

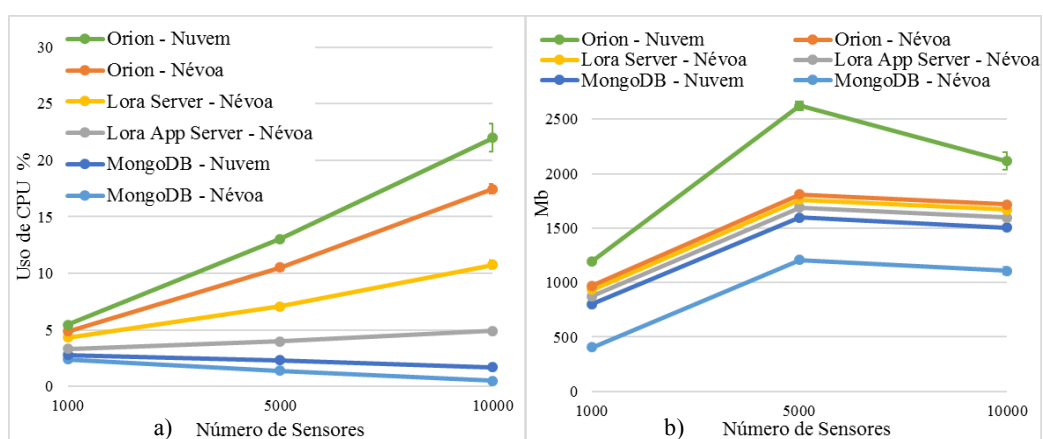


Figura 8. Recursos do sistema da configuração B; a) uso de CPU b) uso de memória

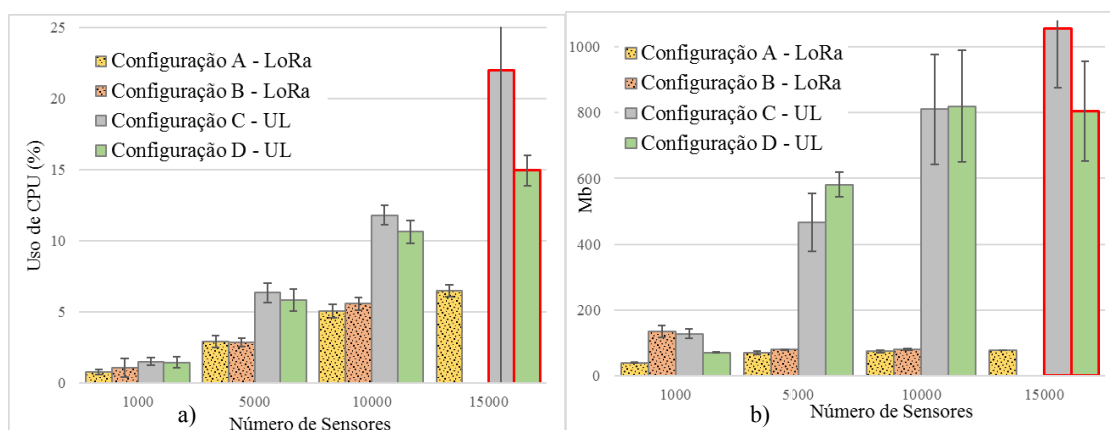


Figura 9. Uso de recursos de sistema do IoT Agent; a) uso de CPU; b) uso de memória

Um comparativo interessante é entre o IoT Agent LoRa (desenvolvido pelos autores) e o IoT Agent UL oficial do FIWARE. A Figura 9 mostra o uso de recursos computacionais desse módulo para cada configuração, podemos observar que o IoT Agent UL é menos escalável, aumentando o uso de recursos à medida que sobe o processamento, sendo que cai em cargas mais altas (15.000 sensores). Em contrapartida, o IoT Agent LoRa mantém o consumo de recursos estável para as cargas de trabalho apresentadas. Esse resultado é uma contribuição importante, pois mostra que a integração dos módulos LoRaWAN com o FIWARE foi realizada de maneira eficiente e pode ser utilizada em outros projetos.

7. Discussão

A partir da cuidadosa análise dos resultados dos experimentos observa-se que o loraserver se mostrou escalável, mantendo o tempo de processamento estável mesmo com o aumento da carga de trabalho. Existe um impacto computacional que a arquitetura LoRaWAN adiciona ao sistema, principalmente em cargas de trabalho baixas quando comparado a uma situação sem eles, um resultado esperado e intuitivo que pode significar que para algumas aplicações que necessitem de tempo real, a implantação da arquitetura LoRaWAN não seja viável. No entanto quando se trata do gerenciamento de uma rede com muitos sensores, é necessário possuir uma camada de controle de acesso ao meio, além de tratar de questões como segurança e autenticação.

Contra intuitivamente, a configuração mais escalável foi a com apenas módulos LoRaWAN executados na névoa e o FIWARE na nuvem. Isso é devido ao tratamento de fluxo de dados do LoRaWAN e a falta de escalabilidade do IoT Agent UL do FIWARE.

Os experimentos revelaram que o gargalo do sistema como um todo é o IoT Agent – um módulo da plataforma FIWARE de IoT – e não os módulos LoRaWAN, já que esse módulo é o primeiro que apresenta falhas quando submetido a uma carga de trabalho alta.

A integração dos módulos do loraserver com a plataforma de IoT foi mais trabalhosa do que foi previsto pois ainda não existem muitas aplicações que conseguem se comunicar com o LoRa Application Server de forma transparente. Tanto que para viabilizar os experimentos, foi necessário desenvolver um IoT Agent que conseguisse realizar essa integração, sendo a ponte entre os módulos LoRaWAN e o FIWARE. Os resultados mostram que o IoT Agent desenvolvido por nós possui melhor desempenho do que o IoT Agent UL do FIWARE, nas condições analisadas nos experimentos.

Há também a discussão na implementação de um nó de névoa e nas possíveis configurações e usos desse. Em alguns cenários, somente foram executados módulos referentes ao LoRa, e em outros, parte da plataforma de IoT para que os dados pudessem ser consumidos localmente. No primeiro caso, há o processamento hierárquico da informação, no entanto, os dados processados na névoa não são utilizados localmente, sendo necessário envia-los para nuvem para que possam ser consumidos por outras aplicações. Na segunda configuração, os dados podem ser usados por aplicações locais próximas de onde os dados são gerados, possibilitando aplicações tempo real e robustez do sistema a possíveis desconexões. Os resultados obtidos mostram que todas essas características podem ser acrescentadas sem grandes perdas no desempenho geral do sistema. A relevância desse trabalho está na avaliação de um cenário IoT complexo fim-a-fim, desde a geração do dado até ser consumido por alguma aplicação. Considerando aspectos de rede e sistema, além da introdução da névoa computacional e comparando esses resultados com diversas possíveis configurações.

9. Conclusão

Esse artigo investigou o impacto computacional no desempenho e escalabilidade da arquitetura LoRaWAN em cenários complexos de IoT e sua integração com uma plataforma de IoT (FIWARE) em diferentes configurações que exploram *trade-offs* entre a computação de névoa e a computação em nuvem. Os resultados obtidos mostram

que a arquitetura LoRaWAN possui impacto no sistema, entretanto é escalável e factível de ser implementada em ambientes alta carga de trabalho, tanto em nuvem quanto em névoa. Como trabalhos futuros deseja-se realizar experimentos explorando outras possíveis configurações além de comparar a arquitetura LoRaWAN com outras tecnologias LPWAN que necessitem de algum módulo de software.

8. Agradecimentos

Essa pesquisa foi parcialmente financiada pelo projeto SWAMP [Kamienksi et. al. 2018], uma colaboração entre Brasil e União Europeia.

Referências

- Ayele, E. D., Hakkenberg, C., Meijers, J. P., Zhang, K., Meratnia, N., and Havinga, P. J. M., (2017) "Performance Analysis of LoRa Radio for an Indoor IoT Applications," in Internet of Things for the Global Community (IoTGC).
- Bonomi, F., Milito, R., Zhu, J., Addepalli, S., (2012). "Fog Computing and its role in the Internet of Things", IEEE Workshop on Mobile Cloud Computing (MCC), pp. 13-16.
- da Cruz, M.A., Rodrigues, J.J., Sangaiah, A.K., Al-Muhtadi, J., Korotaev, V., (2018) "Performance Evaluation of IoT Middleware, Journal of Network and Computer Applications, 109, pp.53-65.
- Ericsson, "Cellular Networks for Massive IoT", White Paper, UEN 284 23-3278, (2016) [Online], www.ericsson.com/res/docs/whitepapers/wp_iot.pdf, acesso em 10/10/2018.
- Georgiou, O. and Raza, U. (2017). "Low Power Wide Area Network Analysis: Can LoRa Scale?" *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 162–165.
- Kamienski, C., Soininen, J.P., Taumberger, M., Dantas, R., Toscano, A., Cinotti, T.S., Maia, R.F. Torre Neto, A., "Smart Water Management Platform: IoT-Based Precision Irrigation for Agriculture", *Sensors*, 19, p.276, Janeiro 2019.
- Magrin, D., Centenaro, M. e Vangelista, L. (2017). "Performance evaluation of LoRa networks in a smart city scenario", in 2017 IEEE International Conference on Communications (ICC), p. 1–7.
- Martínez, R., Pastor, J.Á., Álvarez, B., Iborra, A., (2016) "A Testbed to Evaluate the FIWARE-based IoT Platform in the Domain of Precision Agriculture, *Sensors*, 16(11).
- Petäjäjärvi, J., Mikhaylov, K., Yasmin, R., Hämäläinen, M. e Iinatti, J. (2017). "Evaluation of LoRa LPWAN Technology for Indoor Remote Health and Wellbeing Monitoring", *Int J Wireless Inf Networks*, vol. 24, n° 2, p. 153–165.
- Raza, U., Kulkarni, P. e Sooriyabandara, M. (2017). "Low Power Wide Area Networks: An Overview", *IEEE Communications Surveys Tutorials*, vol. 19, n° 2, p. 855–873.
- Sornin, N., Luis, M., Eirich, T., Kramp, T. e Hersent, O. "LoRaWAN Specification v1.0.2," (2016). [Online]. Available: www.loraalliance.org/portals/0/specs/. Acesso em 04/12/2018.
- Van den Abeele, F., D., Haxhibeqiri, J., Moerman, I., Hoebeke, J. (2017). "Scalability Analysis of Large-Scale LoRaWan Networks in ns-3", *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2186-2198.
- Zyrianoff, I.; Borelli, F.; Kamienski, C., (2017), "SenSE – Sensor Simulation Environment: Uma ferramenta para geração de tráfego IoT em larga escala", SBRC 2017. Salão de Ferramentas.
- Zyrianoff, I.; Heideker, A.; Otollini, D; Kamienski, C., (2018), "Scalability of an Internet of Things Platform for Smart Waer Management for Agriculture", FRUCT 2018.