

Os desafios/problemas para interoperabilidade aplicada à Cidades Inteligentes

Victor Hugo Benatti Franco
Instituto de Informática - INF
Universidade Federal de Goiás
Goiânia, Goiás
benatti@discente.ufg.br

Rogério Rodrigues Rocha
Instituto de Informática - INF
Universidade Federal de Goiás
Goiânia, Goiás
rogerio.rocha@discente.ufg.br

Mauricio Gomes Rocha
Instituto de Informática - INF
Universidade Federal de Goiás
Goiânia, Goiás
mauricio.rocha@discente.ufg.br

Norton Almeida Pontes
Instituto de Informática - INF
Universidade Federal de Goiás
Goiânia, Goiás
nortonalmeyda@discente.ufg.br

Rafael Ribeiro Machado Parreira
Instituto de Informática - INF
Universidade Federal de Goiás
Goiânia, Goiás
rafael.parreira@discente.ufg.br

Resumo—A Internet das Coisas (IoT) enfrenta desafios de interoperabilidade devido à diversidade de protocolos. Este artigo propõe um padrão universal para integrar dispositivos IoT em Cidades Inteligentes, visando superar obstáculos e promover comunicação eficiente. Destacam-se protocolos como MQTT, CoAP, HTTP, AMQP e Wi-Fi, cada um com vantagens e limitações. Frameworks como Arduino, Raspberry Pi, IoTivity, AWS IoT, Microsoft Azure IoT, Google Cloud IoT e Eclipse IoT são explorados para padronização. A comparação revela que IoTivity é ideal para interação entre dispositivos, Azure IoT para escalabilidade na nuvem e Eclipse IoT para flexibilidade. A combinação personalizada de plataformas é recomendada com base nas necessidades específicas de uma Cidade Inteligente.

Palavras-chave— IoT, Cidades Inteligentes, interoperabilidade, padronização, comunicação entre sistemas

I. INTRODUÇÃO

A. Descrição do Problema:

A Internet das Coisas (IoT) está se tornando uma parte integrante de nossas vidas, conectando um extenso número de dispositivos ao redor do mundo. No entanto, à medida que a adoção da IoT aumenta, também surgem desafios significativos relacionados à comunicação entre esses dispositivos [1].

A interoperabilidade entre os dispositivos IoT é um problema persistente. Muitos desses dispositivos operam em silos, utilizando protocolos de comunicação diferentes, o que dificulta a criação de soluções abrangentes e eficazes. A falta de padronização compromete a capacidade de diferentes dispositivos se comunicarem de maneira eficiente, prejudicando a criação de ecossistemas verdadeiramente integrados [2].

Dessa forma, o artigo abordará os obstáculos significativos que as Cidades Inteligentes enfrentam em relação à interoperabilidade entre os dispositivos IoT.

B. Proposta de Solução

Com base nos desafios apresentados no artigo [2], uma sugestão para abordar a atual problemática é a concepção de um padrão de intercomunicação destinado a integrar os variados sistemas dos dispositivos IoT em uma Cidade Inteligente. A presença de diversos padrões, conforme

destacado no artigo [3], dificulta a interoperabilidade de dados. Portanto, para assegurar uma comunicação eficiente entre esses dispositivos, a adoção de um padrão universal emerge como uma medida eficaz contra o problema em questão. Para alcançar esse objetivo, propõe-se a exploração de frameworks que promovam a padronização e facilitem a interação com os diversos tipos de sistemas, possibilitando a troca de dados e informações. Essa abordagem viabilizará a criação de aplicações multiplataformas, acessíveis através das diversas arquiteturas de fabricantes.

II. FUNDAMENTOS TEÓRICOS

Com o conhecimento dos protocolos de comunicação mais usados, há ainda o desafio de solucionar o problema da interoperabilidade, que é um problema incessante [2], como todos os protocolos são bastante utilizados todos em seus propósitos específicos, há casos que irá conflitar os protocolos de comunicação, e para isso há uma possibilidade de solução, usando um bem maior, como por exemplo o avanço tecnológico como propósito a ser alcançado usando de soluções compartilhadas e experiências passadas como aprendizado para criar-se um termo de normas comuns.

A busca por uma interoperabilidade eficaz entre sistemas representa um dos desafios mais prementes no campo da tecnologia. A diversidade de plataformas, protocolos e padrões adotados por diferentes fabricantes dificulta a comunicação fluida entre dispositivos e sistemas heterogêneos. A ausência de uma padronização universal frequentemente resulta em silos de informação, onde a troca de dados torna-se complexa e limitada. Além disso, questões relacionadas à segurança, privacidade e a rápida evolução tecnológica também contribuem para a complexidade desse cenário.

Abordar os desafios na interoperabilidade requer esforços colaborativos, envolvendo padronização de protocolos, desenvolvimento de interfaces comuns e a implementação de práticas de segurança robustas. A superação desses obstáculos é crucial para promover ecossistemas tecnológicos mais integrados, capacitando a inovação e proporcionando benefícios significativos em diversas áreas, desde a Internet das Coisas até ambientes corporativos e sistemas de saúde.

Com base na problemática abordada, é necessário entender os principais protocolos de comunicação na IoT e frameworks para padronização já conhecidos para se ter uma base para determinar um padrão universal. Além disso, é preciso entender os desafios na interoperabilidade de sistemas dentro deste contexto.

A. Protocolos de Comunicação na IoT

Com base no contexto apresentado, para alcançar a interoperabilidade entre dispositivos IoT, é vital compreender os diferentes protocolos de comunicação utilizados. Temos como exemplos de protocolos de comunicação na IoT o MQTT, o CoAP, o HTTP, o AMQP e o Wi-fi.

O protocolo MQTT é caracterizado por ser leve e eficiente, tornando-se ideal para ambientes com largura de banda limitada. Adota o modelo de publicação/assinatura, permitindo comunicação assíncrona, e é projetado para redes instáveis, assegurando confiabilidade na entrega de mensagens. Dentre suas vantagens tem-se um baixo consumo de largura de banda, suporte a notificações push. Além disso, é adequado para dispositivos com recursos limitados, embora haja uma falta de mecanismos integrados de segurança, podendo depender de camadas adicionais.

O CoAP é projetado para dispositivos com restrições de recursos, apresentando eficiência e suporte a multicast. Utiliza o modelo de requisição/resposta, semelhante ao HTTP, e destaca-se por seu consumo eficiente de recursos, menor sobrecarga em comparação com o HTTP, sendo adequado para ambientes IoT com dispositivos limitados. Entretanto, o CoAP é menos difundido que o HTTP, o que pode limitar a interoperabilidade em alguns casos.

Amplamente utilizado na web, o HTTP é conhecido por seu modelo de requisição/resposta e pelo uso de URLs para identificação de recursos. Sua fácil integração com a infraestrutura web existente, suporte generalizado em dispositivos e plataformas, e compatibilidade com firewalls e proxies são vantagens notáveis. Como suas maiores limitações, o HTTP tem um maior overhead em comparação com protocolos mais leves como MQTT e CoAP e uma menor eficiência em ambientes com restrições severas de recursos.

O AMQP é um protocolo de mensagens orientado a filas, oferecendo suporte a filas de mensagens, trocas e roteamento flexível. Projetado para cenários de mensageria complexos, destaca-se pela flexibilidade na implementação de padrões avançados e suporte a cenários de mensageria assíncrona e distribuída. Entretanto, ele pode ser mais complexo do que protocolos como MQTT para casos de uso simples. Além disso, há um maior consumo de recursos em comparação com protocolos mais leves.

A tecnologia Wi-Fi, padrão IEEE 802.11, é amplamente utilizada para conectividade de rede local sem fio, oferecendo largura de banda relativamente alta em comparação com tecnologias mais antigas. Este modelo possui uma alta taxa de transferência de dados, além de uma ampla disponibilidade e compatibilidade. Em contrapartida, há também um consumo mais elevado de energia em comparação com tecnologias de baixo consumo, junto a um alcance limitado em comparação com tecnologias de longo alcance como LoRa.

B. Frameworks para a padronização na IoT

Seguindo a proposta de solução apresentada, para se alcançar a interoperabilidade entre dispositivos IoT é fundamental compreender os diferentes frameworks de desenvolvimento para IoT. Alguns dos principais são:

1) *Arduino: Plataforma de código aberto para prototipagem eletrônica. Suporte a uma ampla variedade de placas e módulos. Comunidade ativa e vasta documentação.*

2) *Raspberry Pi: Placa de computador de baixo custo. Pode ser utilizado como um controlador central em projetos IoT. Suporte a diversas linguagens de programação.*

3) *IoTivity: Framework open-source mantido pela OCF (Open Connectivity Foundation). Focado em padronizar a comunicação entre dispositivos IoT. Suporta diferentes protocolos de comunicação.*

4) *Node-RED: Plataforma de código aberto para programação visual. Facilita a criação de fluxos de dados IoT de maneira intuitiva. Suporte a uma variedade de dispositivos e protocolos.*

5) *Microsoft Azure IoT: Plataforma de IoT integrada ao ecossistema Azure. Oferece serviços para conectar, monitorar e gerenciar dispositivos IoT. Integração com ferramentas de análise de dados e machine learning.*

6) *Google Cloud IoT: Oferece serviços para conectar e gerenciar dispositivos IoT na nuvem. Integração com outros serviços do Google Cloud. Suporte a segurança e monitoramento.*

7) *Eclipse IoT: Conjunto de projetos de código aberto para desenvolvimento IoT. Inclui ferramentas, frameworks e protocolos. Foco na interoperabilidade e padronização.*

III. DESCRIÇÃO METODOLÓGICA

Diante da problemática apresentada e da proposta de solução que busca a padronização na intercomunicação dos dispositivos IoT em uma Cidade Inteligente, realizou-se a comparação entre os frameworks supracitados para se observar quais seus pontos fortes e sua limitações, principalmente, em relação a interoperabilidade.

A. Arduino e Raspberry Pi

- **Plataformas de Hardware:** Essas não são plataformas de software, mas placas de desenvolvimento de hardware. Elas podem alcançar interoperabilidade por meio de:
- **Protocolos padrão:** Ambas suportam protocolos comuns de comunicação, como Wi-Fi, Bluetooth e MQTT, permitindo a comunicação com outros dispositivos que utilizam os mesmos protocolos.
- **Bibliotecas e Frameworks:** Bibliotecas de código aberto como ArduinoJSON e bibliotecas MQTT para Raspberry Pi facilitam a troca de dados com diversos dispositivos.
- **Limitações:** Projetadas principalmente para o desenvolvimento de um único dispositivo, exigindo esforço adicional para cenários complexos de interoperabilidade.

B. IoTivity

- **Framework de Código Aberto:** Foca na interoperabilidade de dispositivo para dispositivo.

- Modelo de Dados Padronizado: Promove a descoberta de dispositivos e comunicação usando o protocolo CoAP leve e um modelo de dados comum, permitindo interação fácil entre dispositivos, independentemente do fornecedor ou plataforma.
- Adoção: Ainda está ganhando popularidade, exigindo dispositivos e bibliotecas compatíveis para implementação.

C. AWS IoT Core, Microsoft Azure IoT, Google Cloud IoT

- Plataformas de IoT Baseadas na Nuvem: Projetadas para gerenciar e conectar implantações de IoT em larga escala.
- APIs e Protocolos Proprietários: Conectam principalmente a dispositivos dentro de seus ecossistemas respectivos, limitando a interoperabilidade nativa com outras plataformas.
- Soluções de Interoperabilidade: Oferecem gateways e serviços de integração para conectar-se a dispositivos e plataformas externas, embora com aumento de complexidade e potencial bloqueio do fornecedor.

D. Some Common Mistakes

- Ecossistema de Projetos de Código Aberto: Fornece diversas ferramentas e frameworks para diversas necessidades de IoT, incluindo soluções de interoperabilidade como Eclipse Kura e Eclipse Leshan.
- Protocolos e APIs Padronizados: Promove padrões e protocolos abertos como MQTT e LwM2M, facilitando a comunicação com diversas plataformas e dispositivos.
- Complexidade: Requer escolher e integrar ferramentas adequadas do ecossistema Eclipse IoT, com diferentes níveis de suporte à interoperabilidade.

IV. CONCLUSÃO

Priorizando a interoperabilidade, e outras características essenciais para uma cidade inteligente como escalabilidade e flexibilidade, foram escolhidas as seguintes plataformas:

A. Comunicação forte de dispositivo para dispositivo

IoTivity: Possibilita interação perfeita entre dispositivos diversos, independentemente do fornecedor ou plataforma, sendo ideal para redes de sensores e aplicações distribuídas.

B. Gerenciamento baseado em nuvem e escalabilidade

Microsoft Azure IoT: Oferece capacidades robustas para implantações em larga escala, integra-se bem com os serviços da Microsoft e apresenta recursos de segurança sólidos.

C. Flexibilidade e integração de código aberto

Eclipse IoT: Utiliza várias ferramentas como Kura e Leshan para interoperabilidade, suporta padrões abertos, permitindo a conexão com diferentes plataformas e ecossistemas.

Com base nas necessidades é possível combinar diferentes opções:

IoTivity + Azure IoT: Aproveita a comunicação de dispositivo para dispositivo para processamento na borda e conecta perfeitamente à nuvem para gerenciamento centralizado.

Eclipse Kura + Google Cloud IoT: Utiliza o Eclipse Kura para interoperabilidade de código aberto e conecta-se ao Google Cloud IoT para monitoramento e análise em nuvem escaláveis.

Node-RED + AWS IoT Core: Utiliza o fluxo visual do Node-RED para processamento flexível de dados e conecta-se ao AWS IoT Core para gerenciamento de dispositivos e serviços em nuvem dentro do ecossistema da AWS.

Por fim, a chave é priorizar as necessidades e escolher plataformas que ofereçam o nível certo de interoperabilidade, escalabilidade e flexibilidade para a cidade inteligente

REFERÊNCIAS

- [1] D. L. Dantas, L. V. L. Filgueiras, and A. A. F. Brandão, "Portability and interoperability in IoT Platforms application layer portabilidade e interoperabilidade na camada de aplicação de plataformas de internet das coisas," in *Anais*, 2019.
- [2] S. Andrade and D. Luque, "Interoperabilidade de Sistemas aplicados às Cidades Inteligentes: Um Estudo de Mapeamento Sistemático", in *Anais do X Workshop de Computação Aplicada em Governo Eletrônico*, Niterói, 2022, pp. 97-108, doi: <https://doi.org/10.5753/wcge.2022.222970>.
- [3] D. V. A. Silveira et al., "Proposta de padronização de comunicação para dispositivos IoT," in *Congresso Brasileiro de Automática-CBA*, vol. 1, no. 1, 2019, doi: <https://doi.org/10.20906/CBA2022/511>.
- [4] Microsoft, Arquitetura do Connected Customer Service com o Hub IoT: <https://learn.microsoft.com/pt-br/dynamics365/customer-service/administer/cs-iot-connected-customer-service-architecture>.