Universidade Federal do Rio Grande do Norte – UFRN
Centro de Ensino Superior do Seridó – CERES
Departamento de Computação e Tecnologia – DCT
Bacharelado em Sistemas de Informação – BSI

Uma análise sobre as técnicas de Redes Tolerantes a Atrasos e Desconexões para mitigação de perda de informações na Internet das Coisas

Denys Nyckson Ferreira da Silva

Orientador: Prof. Ms. João Batista Borges Neto

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Sistemas de Informação como parte dos requisitos para obtenção do título de Bacharel em Sistemas de Informação.

Laboratório de Elementos do Processamento da Informação – LabEPI
Caicó, RN, 12 de dezembro de 2019

Universidade Federal do Rio Grande do Norte - UFRN
Sistema de Bibliotecas - SISBI
Catalogação de Publicação na Fonte. UFRN - Biblioteca Setorial Profª. Maria Lúcia da Costa Bezerra - CERES - Caicó

Silva, Denys Nyckson Ferreira da.
Uma análise sobre as técnicas de Redes Tolerantes a Atrasos e Desconexões para mitigação de perda de informações na Internet das Coisas. / Denys Nyckson ferreira da Silva. – Caicó, RN, 2019.
32 f.: il.

Orientador: Prof. Ms. João Batista Borges Neto.

Monografia (Graduação) – Universidade Federal do Rio Grande do Norte. Centro de Ensino Superior do Seridó - Campus Caicó. Departamento de Computação e Tecnologia. Curso de Bacharelado em Sistemas de Informação.

1. Redes Tolerantes a Atrasos e Interrupções. 2. Internet das Coisas. 3. Protocolo Bundle. I. Borges Neto, João Batista. II. Título.

RN/UF/BS-Caicó CDU 004.738.5:62-4

Elaborado por MAYANE PAULINO DE BRITO E SILVA - CRB-15/847

Uma análise sobre as técnicas de Redes Tolerantes a Atrasos e Desconexões para mitigação de perda de informações na Internet das Coisas

Denys Nyckson Ferreira da Silva

Monografia aprovada em 12 de dezembro de 2019 pela banca examinadora composta pelos seguintes membros:

Prof. Ms. João Batista Borges Neto (orientador) ..... DCT/UFRN

Prof. Ms. Luiz Paulo de Assis Barbosa ..... DCT/UFRN

Prof. Dr. João Paulo de Souza Medeiros ..... DCT/UFRN

Prof. Dr. Walber José Adriano Silva ..... DCT/UFRN

\No matter how bad life may be,
there is always something you can do,
and triumph. While there is life, there is hope."
Stephen Hawking

Agradecimentos

Agradeço primeiramente a Deus, que me deu saúde, física e mental, para investir nesse desafio. A minha mãe **Ediane Ferreira** e ao meu pai **Francisco José** que sempre me deram todo o suporte necessário durante minha vida. Sou extremamente grato ao senhor e a senhora que nunca mediram esforços para me ajudar em qualquer coisa que eu precisasse. A minha avó **Rita Fernandes** que também sempre me ajudou em tudo o que eu precisei. Em especial a minha vozinha **Elita Maria** que hoje está no céu, mas que sempre me deu forças para que continuasse minha jornada, obrigado por todos ensinamentos vó, e a toda a minha família. Amo vocês.

Ao meu orientador e amigo, professor **João Borges**, pelas palavras otimistas nos momentos mais intrincados, pelos longos períodos de orientação, sem nunca faltar com esforço para auxiliar nas resoluções dos problemas e principalmente pela confiança depositada em mim, a ele sempre serei grato. A todos os professores que tive nesse tempo de curso pelos ensinamentos, instruções durante todo o meu tempo nesta universidade e contribuições para minha formação.

À minha namorada **Larissa Adra**, pelo amor, pelo carinho, pelos incentivos, mesmo morando distante me deu total apoio. Obrigado por acreditar em mim, por enfrentar os problemas ao meu lado, pelos momentos inesquecíveis da mais plena felicidade. Te amo.

Sou grato ao professor **Luiz Paulo** pela oportunidade de iniciar trabalhos acadêmicos junto ao Laboratório de Elementos do Processamento da Informação (LabEPI), sediado Centro de Ensino Superior do Seridó da Universidade Federal do Rio Grande do Norte. Onde me propôs uma vasta visão de conhecimentos e soluções de problemas.

Por fim, obrigado a todos que, de alguma forma, contribuíram para que esse sonho se realizasse.

Resumo

Com o advento da Internet das Coisas, novos dispositivos, que fazem parte do nosso cotidiano, estão agora conectados à Internet. O uso crescente de sensores e equipamentos interligados através da Internet já é nossa realidade. Dispositivos esses que têm limitações energéticas, de processamento e de memória, gerando uma demanda por soluções eficientes, mas ainda compatíveis com a Internet. Com isso, certos protocolos foram criados para facilitar a comunicação de dispositivos limitados com a Internet, com o objetivo de não sobrecarregá-los. O MQTT (*Message Queue Telemetry Transport*) é um exemplo de protocolo voltado a dispositivos com restrições. No entanto, devido aos seus recursos limitados, esses dispositivos são muitas vezes incapazes de estar sempre conectados, tornando-se assim redes desafiadoras. Para resolver esse problema, este trabalho propõe uma integração entre as Redes Tolerantes a Atrasos e Interrupções (DTN, do inglês, *Delay Tolerant Networks*) com a Internet das Coisas. Em particular, através do Protocolo Bundle (BP) junto com o protocolo MQTT, como meio para contornar esses possíveis atrasos e desconexões perante a internet das coisas.

Palavras-chave: Redes Tolerantes a Atrasos e Interrupções; Internet das Coisas; Protocolo Bundle.

Abstract

With the advent of the Internet of Things, new devices, which are part of our daily lives, are now connected to the Internet. The increasing use of sensors and equipments connected to the Internet is already our reality. These devices have limited energy, processing and memory, which demand for efficient solutions, but still compatible with the Internet. As a result, protocols were created to facilitate the communication of limited devices with the Internet, in order not to overload them. MQTT (Message Queue Telemetry Transport) is an example of a protocol aimed at devices with restrictions. However, due to their limited resources, these devices are often unable to always be connected, thus becoming challenging networks. To solve this problem, this work proposes an integration between the Delay Tolerant Networks (DTN) with the Internet of Things. In particular through the Bundle Protocol (BP) together with the MQTT protocol, as a mean to overcome these possible delays and disconnections in the Internet of Things.

Keywords: Delay/disruption Tolerant Network; Internet of Things; Bundle Protocol.

Sumário

Lista de Figuras iii

Lista de Tabelas v

Glossário vii

1 Introdução 1
- 1.1 Tema e Delimitação do estudo . . . 1
- 1.2 Contextualização do Problema . . . 2
- 1.3 Objetivos . . . 3
  - 1.3.1 Objetivo Geral . . . 3
  - 1.3.2 Objetivos específicos . . . 3
- 1.4 Motivação e Justificativa do estudo . . . 3
- 1.5 Organização do trabalho . . . 3

2 Levantamento bibliográfico 5
- 2.1 Internet das coisas . . . 5
  - 2.1.1 Perspectiva histórica da IoT . . . 6
  - 2.1.2 Padrões e Estruturas da IoT . . . 6
  - 2.1.3 Arquitetura IoT . . . 7
  - 2.1.4 Protocolos da camada de aplicação para IoT . . . 8
- 2.2 Redes Tolerantes a Atrasos e Desconexões . . . 10
  - 2.2.1 Fundamentos e Características das DTNs . . . 10
  - 2.2.2 Arquitetura DTN . . . 10
  - 2.2.3 Tipos de Contatos . . . 12
- 2.3 Ambientes de Testes Reais, Emulação e Simulação . . . 13
  - 2.3.1 Cenários de Redes . . . 14
  - 2.3.2 Simulador ONE . . . 15
  - 2.3.3 Cooja . . . 15
  - 2.3.4 Ambientes de testes reais . . . 16

3 Desenvolvimento 19
- 3.1 Metodologia . . . 19
- 3.2 Modelo Proposto . . . 20
- 3.3 Experimentos . . . 21
  - 3.3.1 Ambiente . . . 21

4 Considerações Finais **27**
4.1 Resultados 27
4.2 Trabalhos futuros 28

Referências Bibliográficas **31**

Lista de Figuras

2.1 Arquitetura para IoT 8
2.2 Camadas do CoAP 9
2.3 Componentes do MQTT 9
2.4 Fases da operação do TCP. 11
2.5 Protocolo Bundle 12
2.6 Exemplo de contatos programados 13
2.7 Waypoint aleatório com 200 nós e alcance de rádio de 250m 15
2.8 Opções de visualização do cooja sendo utilizado 16

3.1 Exemplo convencional de conexão na IoT 20
3.2 Modelo proposto para amenizar a perda de dados nas conexões IoT 21
3.3 Inicializando conexão a plataforma IoT e causando uma desconexão. 22
3.4 Restabelecendo conexão com a plataforma IoT 23
3.5 Conexão entre a plataforma IoT com *Broker Bundle* 24
3.6 Restabelecendo conexão a plataforma IoT 25

Lista de Tabelas

3.1 Especificações do Raspberry Pi 2 modelo B 22
3.2 Especificações ESP8266 modelo ESP-01 22

Glossário

Acrônimos

ACK ..... *Acknowledgments*
BP ..... *Bundle Protocol*
CoAP ..... *Constrained Application Protocol*
CPU ..... *Central Processing Unit*
DTN ..... *Delay and Disruption Tolerant Networks*
HTTP ..... *Hypertext Transfer Protocol*
IETF ..... *Internet Engineering Task Force*
IoT ..... *Internet of Things*
IPv6 ..... *Internet Protocol version 6*
MQTT ..... *Message Queue Telemetry Transport*
ONE ..... *Opportunistic Network Environment*
PDU ..... *Protocol Data Units*
RFID ..... *Radio-Frequency IDenti cation*
RSSF ..... *Redes de Sensores Sem Fio*
TCP ..... *Transmission Control Protocol*
UDP ..... *User Datagram Protocol*
WSN ..... *Wireless Sensor Networks*

Capítulo 1

Introdução

\If knowledge can create problems,
it is not through ignorance that we can solve them."
Isaac Asimov

Este Capítulo está organizado da seguinte forma. Na Seção 1.1 está descrito o tema do referido trabalho, assim como a delimitação do mesmo. Logo após, na Seção 1.2 é apresentada a contextualização e o problema que se busca resolver. Em seguida, são definidos seus os objetivos gerais e específicos, juntamente com a motivação e justificativa, respectivamente nas seções 1.3 e 1.4. Por fim, na seção 1.5, está exposta a organização geral do trabalho.

1.1 Tema e Delimitação do estudo

A Internet das Coisas (do inglês, Internet of Things (IoT)) é um novo paradigma computacional que tem crescido substancialmente e que pode modificar a forma com que os seres humanos interagem com computadores (Koreshoff et al., 2013). A IoT está transformando o cotidiano das pessoas e provocando uma verdadeira revolução na área da computação e da comunicação (Atzori et al., 2010). Atualmente, mais e mais objetos inteligentes estão sendo incorporados e distribuídos na Internet. A gestão deles, o tráfego que geram, assim como o uso efetivo desses dados são desafios que desenvolvedores, pesquisadores e entusiastas da Internet das Coisas e tecnologias de redes de sensores sem fio estão enfrentando (Luzuriaga et al., 2017).

A IoT é composta por dispositivos necessariamente pequenos e de baixo consumo de energia e que são imprevisivelmente móveis. Sendo assim, estes fatores contribuem diretamente com diversos problemas nestas redes, dentre os quais atrasos e interrupções da informação são os principais, contribuindo para um ambiente em que a conectividade de rede não é garantida de ponta a ponta.

Desta forma, podemos citar alguns cenários nos quais o protocolo de comunicação da internet se torna pouco robusto, ocasiões como: comunicações sem fio, comunicações entre dispositivos móveis, comunicações entre dispositivos com restrições de energia, comunicações rurais, comunicações em campo de batalha, comunicações submarinas, comunicações interplanetárias etc. Estes cenários, considerados “desafiadores”, possuem em comum a dificuldade de manter uma comunicação fim-a-fim com baixa latência e pequena

perda de pacotes. Devido a estas características, as redes que consideram estes aspectos foram denominadas Redes Tolerantes a Atrasos e Desconexões (*Delay and Disruption Tolerant Networks* - DTNs). Apesar do termo DTN ser o mais utilizado na literatura, também podem ser encontradas outras terminologias, tais como: redes com conectividade eventual, redes móveis parcialmente conectadas, redes desconectadas, redes com conectividade transiente, redes incomuns, redes extremas e, mais recentemente, Redes com Desafios (Chen et al., 2006).

Sendo assim, este trabalho abordará como a aplicação de redes tolerantes a atrasos e interrupções podem nos auxiliar no problema de desconexões entre os dispositivos de Internet das Coisas.

Um exemplo desse problema é quando temos uma conexão ponto a ponto, para a qual ocasiona-se propositalmente uma desconexão entre os dispositivos, interrompendo assim sua comunicação. Desse modo, os dados enviados por esses dispositivos não podem ser dispersos ou perdidos durante a desconexão.

1.2 Contextualização do Problema

A recente evolução da Internet é caracterizada por um enorme aumento no número de dispositivos físicos na rede. A Internet está mudando gradualmente de uma rede de servidores e pessoas, conectadas através de seus terminais pessoais, para uma rede povoada de sensores, atuadores, receptores, etiquetas RFID (do inglês *Radio-Frequency IDenti cation*) e todos os novos dispositivos inteligentes que estão surgindo em um passo rápido. Suas características comuns são suas limitações, tais como: capacidade de computação e armazenamento; capacidade de interconexão de rede e suas restrições de energia. Esta rede emergente forma a chamada IoT (do inglês *Internet of Things*). Segundo Auzias et al. (2015) a diversidade dessas coisas tornam a rede mais heterogênea, tanto em termos de seus componentes e sub-redes, como até a extensão de protocolos já existentes e, algumas vezes, a criação de novos protocolos.

Assim, tendo em vista as características encontrados na IoT, a transmissão de seus dados é um dos principais desafios a ser considerado. Tal desafio não deve afetar a entrega da informação desde o nó sensor até o cliente, não podendo se perder ou ser descartada no meio do caminho. Assim, a hipótese levantada neste trabalho é a de que o problema citado pode ser evitado por meio da utilização de protocolos de aplicação como o protocolo de pacotes BP (do inglês *Bundle Protocol*).

O BP é visto dentro da camada de aplicação do protocolo DTN e tem como objetivo formar uma rede de sobreposição de armazenamento e transporte com a capacidade de lidar com conectividades intermitentes, incluindo casos onde o remetente e o destinatário não estejam presentes simultaneamente na rede, com isso armazenando os dados até a conexão ser estabelecida novamente.

Desta forma, espera-se que se utilizado juntamente com o MQTT (do inglês *Message Queue Telemetry Transport*), que é um protocolo de mensagens simples, projetado para dispositivos restritos com baixa largura de banda, o BP possa trazer uma solução adequada ao problema da entrega de mensagens na IoT, mesmo considerando seus desafios. Os quais serão descritos nos capítulos 2 e 3.

1.3 Objetivos

A seguir, estão apresentados os objetivos gerais e específicos deste trabalho.

1.3.1 Objetivo Geral

O presente trabalho tem como objetivo geral analisar a viabilidade da utilização das estratégias DTN para solucionar o problema da entrega de dados mesmo sob as conexões intermitentes observados no cenário de Internet das Coisas.

1.3.2 Objetivos específicos

Para que os principais objetivos deste trabalho possam ser alcançados, os seguintes objetivos específicos devem ser devidamente atendidos:

Caracterizar o problema de desconexões na IoT;

Analisar as principais estratégias DTN;

Propor uma das estratégias DTN para melhorar o desempenho da entrega de dados nos cenários da IoT.

1.4 Motivação e Justificativa do estudo

Novos desafios surgem enquanto são criadas novas aplicações para IoT. Os dados providos pelos objetos agora podem apresentar imperfeições (calibragem do sensor), inconsistências (fora de ordem, outliers) e serem de diferentes tipos (gerados por pessoas, sensores físicos, fusão de dados) (Santos et al., 2016).

Neste cenário, a possibilidade de novas aplicações é crescente, mas temos novos desafios de conectar à Internet objetos com restrições de processamento, memória, comunicação e energia (Ruiz et al., 2004).

A principal motivação para realizar este trabalho foi entregar uma uma solução viável para amenizar a perda de dados durante desconexões entre dispositivos. Para isso, pretende-se verificar se o protocolo MQTT implementado juntamente com DTN, especificamente por meio da integração do BP ao MQTT, oferece um desempenho adequado para os dispositivos de Internet das Coisas quando a conectividade entre eles tem pequenas ou longas desconexões. Espera-se, com isso, viabilizar um melhor funcionamento do protocolo de troca de mensagens perante os cenários desafiadores da IoT.

1.5 Organização do trabalho

Este trabalho é composto por, além deste Capítulo, outros que englobam assuntos os quais serão necessários para o seu desenvolvimento. No Capítulo 2 encontra-se todo o levantamento bibliográfico do trabalho em questão, no qual fornece a fundamentação necessária sobre Internet das Coisas (IoT) e redes DTN, mostrando como funciona sua estrutura e também propor uma de suas estratégias para resolver as desconexões.

No Capítulo 3 é mostrada a metodologia que irá ser utilizada para atender aos objetivos que foram citados anteriormente neste capítulo e o cronograma que deverá ser seguido.

Capítulo 2

Levantamento bibliográfico

\We can only see a short distance ahead,
but we can see plenty there that needs to be done."
Alan Mathison Turing

Este Capítulo está organizado da seguinte forma. Na Seção 2.1 é apresentada a Internet das Coisas, mostrando sua importância, suas características e os tipos de protocolos utilizados. Em seguida, na Seção 2.2, são mostrados os protocolos DTN, como se comportam e sua usabilidade, a qual é o objeto de estudo deste trabalho. Posteriormente, na Seção 2.3 são apresentados os ambientes de testes reais, emulação e simulação, próprios para DTN e dispositivos inteligentes.

2.1 Internet das coisas

A Internet das Coisas (Internet of Things, ou IoT) é um conceito que dispõe que a maioria dos dispositivos que utilizamos diariamente está conectada entre si e pela Internet. Uma revolução que envolve a conexão de nossos mundos físicos e digitais. O real significado de internet das coisas basicamente é pegar todas as coisas do mundo e conectá-las à Internet(Santos et al., 2016).

Quando algo está conectado à Internet, isso significa que ele pode enviar ou receber informações, ou ambas. Com a capacidade de enviar e/ou receber informações e ao mesmo tempo processá-las torna as coisas inteligentes.

Um dispositivo para ser inteligente não precisa ter super armazenamento ou um super processador dentro dele. Tudo que precisa ser feito é conectar-se ao servidor onde os dados estão sendo disponíveis ou processados. A IoT tem alterado aos poucos o conceito de redes de computadores, segundo Kurose e Ross (2012) "Redes de Computadores" começa a soar um tanto envelhecido devido à grande quantidade de equipamentos e tecnologias não tradicionais que são usadas na Internet.

Hoje, com o advento da IoT adquirimos novas habilidades junto à Internet, habilidades essas como conseguir controlar remotamente os objetos, permitindo também que os próprios objetos sejam acessados como provedores de serviços. Com essas novas habilidades dos dispositivos IoT, geram-se um grande número de oportunidades, tanto no âmbito acadêmico quanto no industrial. Cada vez mais, organizações de vários setores estão usando a IoT para operar com mais eficiência, tornando melhor suas tomadas de decisões e aumentando os valores de seus negócios.

2.1.1 Perspectiva histórica da IoT

Kevin Ashton comentou, em junho de 2009 (Ashton, 2009), que o termo Internet of Things foi primeiro utilizando em seu trabalho intitulado "*I made at Procter & Gamble*" em 1999. Na época, a IoT era associada ao uso da tecnologia (RFID). Kevin Ashton acreditava que a identificação por radiofrequência (RFID) era um pré-requisito para a Internet das Coisas. Ele concluiu que se todos os dispositivos fossem "endereçados", os computadores poderiam gerenciar, rastrear e detalhar.

Por volta de 2005, o termo bastante procurado (tanto pela academia quando indústria) e que apresenta relação com a IoT foi Redes de Sensores Sem Fio (RSSF) (do inglês *Wireless Sensor Networks* – WSN). Estas redes traziam avanços na automação residencial e industrial [Kelly et al. (2013), Da Xu et al. (2014)]. Nos anos seguintes (entre 2008 e 2010), o termo Internet das Coisas ganhou popularidade rapidamente. Isto se deve ao amadurecimento das RSSFs e ao crescimento das expectativas sobre a IoT.

2.1.2 Padrões e Estruturas da IoT

Conforme descrito por Santos et al. (2016), a IoT constitui-se pela combinação de diversas tecnologias, desde aquelas com o intuito de viabilizar a integração dos objetos no ambiente físico ao mundo virtual, quanto para prover a sua operação e comunicação entre si. Como exemplo desse ecossistema de padrões e tecnologias, existem várias soluções emergentes propostas para o cenário da IoT, alguns deles são:

Sistemas operacionais

- **ContikiOS** [i], um sistema operacional de código aberto que roda em microcontroladores de baixa potência e possibilita o desenvolvimento de aplicativos que fazem uso eficiente do hardware, além de fornecer comunicação sem fio de baixa potência padronizada para uma variedade de plataformas de hardware.
- **LiteOS**[ii], um sistema operacional semelhante ao Unix para redes de sensores sem fio. O sistema operacional também serve como uma plataforma de desenvolvimento para dispositivos inteligentes.

Comunicação

- **6LoWPAN** (IPv6 sobre redes de área pessoal sem fio de baixa potência), um padrão aberto definido pela *Internet Engineering Task Force* (IETF). O padrão 6LoWPAN permite que qualquer rádio de baixa potência se comunique com a Internet.
- **LoRaWAN** (Rede de área ampla), um protocolo para redes de longa distância, foi projetado para suportar grandes redes, como cidades inteligentes, com milhões de dispositivos de baixo consumo de energia.

Aplicações

- **CoAP**[iii] (*Constrained Application Protocol*), um protocolo de aplicação restrita, desenvolvido pela IETF que especifica como os dispositivos com restrição de computação de baixa potência podem operar na Internet.

[i] http://www.contiki-os.org/
[ii] https://www.huawei.com/liteos
[iii] https://coap.technology/

– **MQTT**[iv] (*Message Queue Telemetry Transport*), é um protocolo de mensagens leve para sensores e pequenos dispositivos móveis otimizado para redes TCP/IP não confiáveis ou de alta latência. O esquema de troca de mensagens é fundamentado no modelo *publish/subscribe*, extremamente simples e leve.

Mas detalhes sobre as principais tecnologias utilizadas neste trabalho serão apresentados nas próximas seções.

Ainda segundo Santos et al. (2016), também podemos destacar a arquitetura básica dos objetos inteligentes como sendo composta por quatro unidades: processamento/memória, comunicação, energia e sensores/atuadores. Analisemos a seguir uma percepção da arquitetura IoT e a interligação entre seus componentes:

Processamento/memória: É formada por uma memória interna para armazenamento de dados e programas, onde temos um microcontrolador e um conversor analógico-digital para receber sinais dos sensores. Sua *Central Processing Unit* (CPU) normalmente não apresentam um alto poder computacional, pois as características desejáveis para estas unidades são consumo reduzido de energia e ocupar o menor espaço possível.

Comunicação: São as diversas técnicas usadas para conectar os objetos inteligentes. Na qual desempenha um papel importante no consumo de energia dos objetos sendo, consequentemente, um fator crítico. Algumas das tecnologias usadas são WiFi, Bluetooth, IEEE 802.15.4 e RFID.

Energia: Normalmente a fonte de energia desses objetos inteligentes consiste de uma bateria (recarregável ou não) e um conversor AC-DC e tem a função de alimentar os componentes. Ainda assim, existem outras fontes de alimentação como energia elétrica, solar e mesmo a captura de energia do ambiente através de técnicas de conversão.

Sensores/atuadores: Encarregam-se do monitoramento do ambiente no qual o objeto se encontra. Os sensores podem capturar valores de grandezas físicas como temperatura, umidade, pressão e presença. Atualmente, existem literalmente centenas de sensores diferentes que são capazes de capturar essas grandezas. Atuadores são os dispositivos que produzem alguma ação, atendendo a comandos que podem ser manuais, elétricos ou mecânicos.

Após a apresentação dos padrões e estruturas da IoT, a próxima seção apresenta uma das arquiteturas propostas pela comunidade científica a fim de fornecer melhor funcionamento de soluções de IoT.

2.1.3 Arquitetura IoT

Para conectar bilhões de objetos inteligentes à Internet, deve-se ter uma arquitetura flexível. Na literatura, segundo Al-Fuqaha et al. (2015) temos uma variedade de propostas de arquiteturas sofisticadas, que se baseiam nas necessidades da academia e indústria.

Segundo Gusmao e Brito (2015), o modelo básico de arquitetura apresenta três camadas, conforme ilustrado na Figura 2.1. A primeira camada é a de percepção onde tem o

[iv] http://mqtt.org/

nível mais próximo ao ambiente, a qual identifica cada coisa em uma aplicação de IoT. Do mesmo modo que retrata os objetos físicos, os quais utilizam sensores para coletarem e processarem informações. A camada de rede, contém como função principal a transmissão dos dados obtidos pelos dispositivos da Camada de Percepção para a Camada de Aplicação. Na qual, a concepção das tecnologias de comunicação, serviços de gerenciamento, roteamento e identificação devem ser realizados. A terceira camada, de aplicação, tem como função o armazenamento, processamento e análise dos dados dos objetos inteligentes, a fim de extrair informações relevantes para o contexto que está sendo aplicada a solução.

Figura 2.1: Arquitetura para IoT

Fonte: Gusmao e Brito (2015).

2.1.4 Protocolos da camada de aplicação para IoT

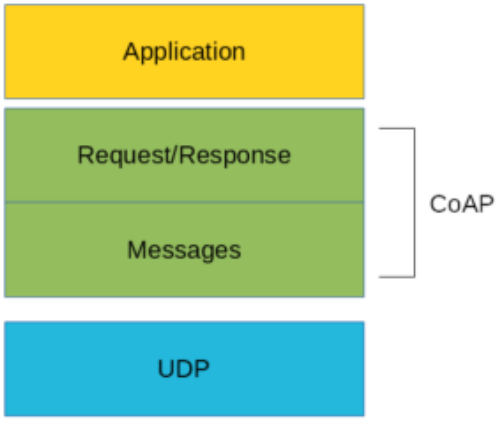
O protocolo de aplicação mais utilizado para prover serviços web é o *Hypertext Transfer Protocol* (HTTP), usado para acessar informações seguindo a estratégia requisição/resposta no paradigma cliente/servidor. Porém, ele possui muita complexidade computacional e tem um consumo de energia elevado para os dispositivos IoT. Com a finalidade de resolver esse problema, foram desenvolvidos dois protocolos da camada de aplicação especificamente para recuperar informações de dispositivos com baixo poder computacional: CoAP e MQTT.

O *Constrained Application Protocol* (CoAP) surgiu para suprir a necessidade em protocolos IoT e foi desenvolvido para interoperar com HTTP e com arquiteturas RESTful, por meio de simples *proxies*, tornando-se compatível com a Internet. Utiliza funcionalidades similares às do HTTP tais como: GET, POST, PUT, DELETE. Levando em conta que utiliza o *User Datagram Protocol* (UDP), o CoAP apresenta um menor consumo computacional e de energia.

Um exemplo da composição do protocolo CoAP pode ser visto em Azzola (2018), ilustrado na Figura 2.2, o CoAP apresenta duas camadas em sua arquitetura interna, permitindo que dispositivos de baixa potência possam interagir através de *RESTfull*. A primeira camada implementa os mecanismos de requisição/resposta. A segunda, detecta mensagens duplicadas e fornece comunicação confiável sobre o UDP.

O *Message Queue Telemetry Transport* (MQTT) é um protocolo projetado para dispo-

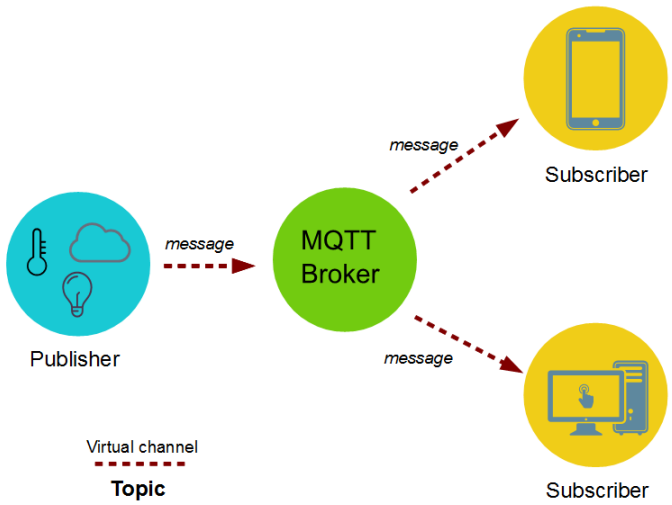
Figura 2.2: Camadas do CoAP



Fonte: Azzola (2018).

sitivos extremamente limitados e utiliza a estratégia de *publish/subscribe* para transferir mensagens, onde necessita de um *Broker* MQTT para gerenciar e direcionar as mensagens dentro da rede. O *Broker* é o elemento responsável por gerir as publicações e as subscrições do protocolo MQTT. Ele é como uma espécie de mediador entre as máquinas, capaz de fazer com que a comunicação de fato ocorra entre elas. Por utilizar o *Transmission Control Protocol* TCP, o MQTT nos permite uma maior confiabilidade.

Figura 2.3: Componentes do MQTT



Fonte: Mustafa (2018).

Um exemplo da operação deste protocolo pode ser visto em Mustafa (2018), como ilustrado na Figura 2.5. O MQTT consiste de três componentes básicos: o *subscriber*,

publisher e o *broker*. A Figura 2.3 mostra a ordem de operação do MQTT. Inicialmente dispositivos se registram (*subscribe*) a um *broker* para obter informações sobre dados específicos, para que o *broker* os avise sempre que publicadores (*publishers*) publicarem os dados de interesse. Os dispositivos inteligentes (*publishers*) transmitem informações para os *subscriber* através do *broker*.

2.2 Redes Tolerantes a Atrasos e Desconexões

As Redes Tolerantes a Atrasos e Desconexões (do inglês *Delay Tolerant Networks* - DTNs) são redes que proporcionam um ambiente que, através de determinados mecanismos e ferramentas sobre a camada física da rede, nos garantem qualidade na transferência de pacotes em ambientes com baixa conectividade, com grandes atrasos ou com desconexões frequentes (Oliveira, 2008).

As DTNs exploram o fato de muitas vezes não termos conexões diretas com o nosso destino. Além disso possuem um mecanismo de armazenamento persistente, que guarda as mensagens a serem enviadas e aguarda um momento em que a conectividade seja reestabelecida para enviar os chamados "*bundles*".

2.2.1 Fundamentos e Características das DTNs

As Redes Tolerantes a Atrasos e Desconexões, caracterizam uma arquitetura completamente inovadora na área das redes de computadores, pois permitem contornar as dificuldades encontradas em alguns cenários. Essa arquitetura aborda o conceito de redes conectadas ocasionalmente e que podem sofrer desconexões frequentes.

O desenvolvimento dessa nova arquitetura surgiu do projeto da NASA de redes Interplanetária, que tratava de comunicações através do espaço em ambientes de grandes atrasos. No entanto, o caso da Internet Interplanetária é um caso específico do que a definição de Redes Tolerantes a Atrasos e Desconexões pode solucionar. A arquitetura engloba os conceitos de redes ocasionalmente conectadas que podem sofrer desconexões frequentes e que podem ser compostas por mais de um protocolo.

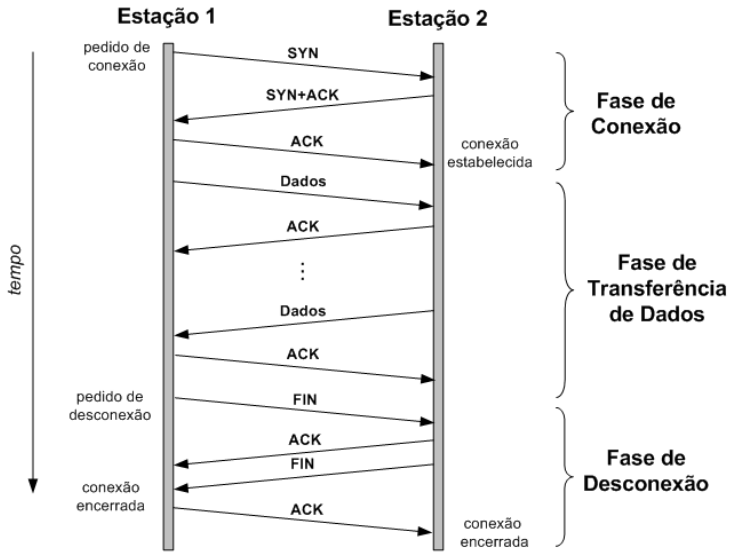
Conforme, POSTEL (1981) a principal causa da Internet convencional não apresentar um bom desempenho em redes com longos atrasos e frequentes desconexões está na operação do *Transmission Control Protocol* (TCP). Pois o TCP é um protocolo de transporte orientado a conexão que garante confiabilidade na entrega de dados fim-a-fim em cima de redes não confiáveis.

Com isso, de acordo com Al Hanbali et al. (2005) a operação do TCP pode ser dividida em três fases: estabelecimento de conexão, transferência de dados e desconexão. Um exemplo de seu funcionamento pode ser visto em Oliveira (2008), ilustrado na Figura 2.4, onde o estabelecimento de conexão se faz com a troca de três mensagens (*three-way handshake*). Em seguida, inicia-se a transferência de dados, onde a boa recepção dos dados pelo destino deve ser sinalizada por reconhecimentos positivos (*acknowledgments* - ACKs). O encerramento de uma conexão TCP se dá com a troca de quatro mensagens.

2.2.2 Arquitetura DTN

A arquitetura DTN fornece um método comum para interconectar *gateways* ou *proxies* diferentes que empregam roteamento da mensagens de armazenamento e encaminhamento

Figura 2.4: Fases da operação do TCP.



Fonte: Oliveira (2008).

para superar interrupções na comunicação. Mas a pergunta é por que uma arquitetura para redes tolerantes a atrasos?

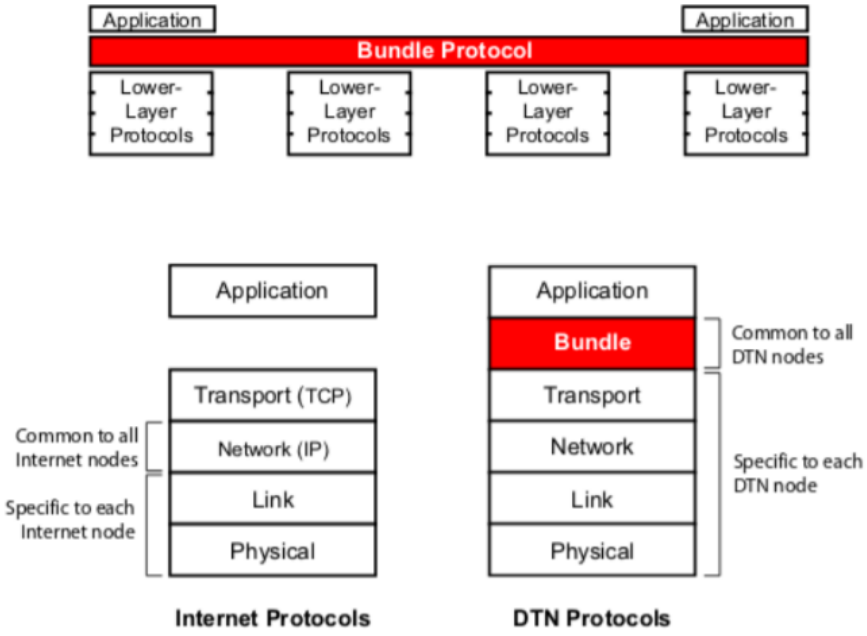
Os protocolos da Internet existentes não funcionam bem para alguns ambientes, devido a algumas premissas fundamentais construídas na arquitetura da Internet. Exemplos de tais ambientes são: as comunicações sem fio, as comunicações entre dispositivos móveis, as comunicações entre dispositivos que possuem restrições de energia, as comunicações interplanetárias, entre outro tipos de ambientes que torna a comunicação desafiadora. Devido a estas características, as redes que consideram estes aspectos foram denominadas Redes Tolerantes a Atrasos e Desconexões.

A arquitetura das redes DTN é constituída para amenizar a maioria das complicações presentes nos protocolos convencionais da Internet. Para isto, sua implementação consiste na adição de uma camada de agregação (*Bundle Layer*), situada acima da camada de transporte, e presente em todos os nós da rede DTN (Oliveira, 2008).

Nesta camada é implementado um protocolo de agregação (*Bundle Protocol*) que é especificado na RFC 5050 (Scott et al., 2007). A principal funcionalidade deste protocolo é a garantia da entrega das mensagens das aplicações aos seus destinos. As mensagens enviadas são transformadas por esta camada de agregação em uma ou mais unidades de dados, denominadas agregados (*bundles*), que são armazenados e encaminhados pelos nós DTN. Esta camada é responsável por lidar com as possíveis múltiplas cópias de um mesmo *bundle* que podem existir simultaneamente em diferentes nós da rede.

Um exemplo da operação do protocolo *bundle* pode ser visto em Warthman (2012), ilustrado na Figura 2.5, no qual o armazenamento está disponível em cada nó na rede, e isso se aplica a todos os nós da rede DTN por onde as mensagens trafegam.

Figura 2.5: Protocolo Bundle



Fonte: Adaptado de Warthman (2012)

2.2.3 Tipos de Contatos

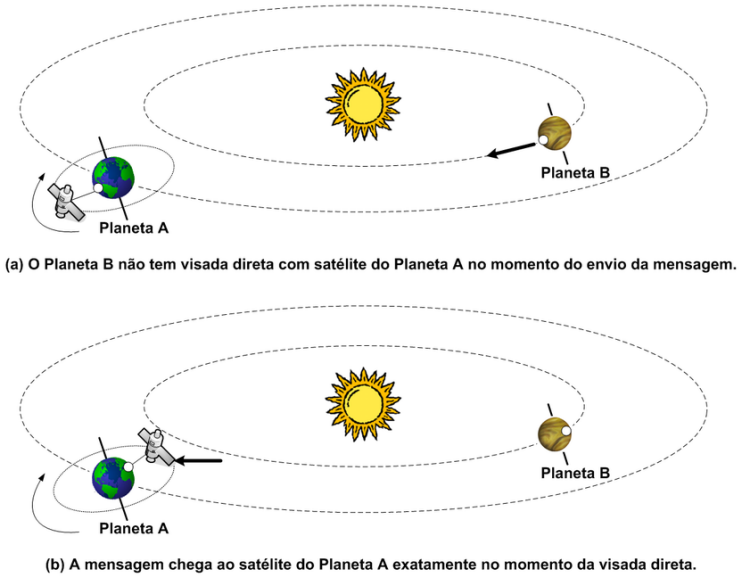
Os contatos geralmente se enquadram em uma de várias categorias, baseadas principalmente na previsibilidade de suas características de desempenho, e se alguma ação é necessária para os nós trocarem dados. Conforme Torgerson et al. (2007), até a presente data, os principais tipos de contatos a seguir são definidos como:

1. **Contatos Persistentes:** Os contatos persistentes estão sempre disponíveis, como uma conexão física entre dois nós. Não demandam qualquer ação para serem instanciados.

2. **Contatos Sob Demanda:** Os contatos sob demanda necessitam que alguma ação seja tomada para decorrerem. Uma conexão discada é um exemplo de contato sob demanda. Do ponto de vista do discador pode ser visto como um contato oportunista, abaixo, do ponto de vista da conexão discada do provedor de serviço o contato pode ser categorizado como oportunista.

3. **Contatos Agendados:** Os contatos agendados ou programados ocorrem em espaços de tempo previamente definidos e com duração conhecida. Há um acordo para estabelecer um contato em um horário específico, e por uma duração específica. Um exemplo de um contato programado é um *link* com um satélite em órbita terrestre. Uma lista de contatos de um nó com o satélite pode ser construída do horário de visualização do satélite, capacidades e latências. Atentamos para o exemplo da Figura 2.6.

Figura 2.6: Exemplo de contatos programados



Fonte: Oliveira (2008).

4. **Contatos Oportunistas:** Os contatos oportunistas não são agendados, ocorrem de forma inesperada e aleatória. Um exemplo bastante comum dos contatos oportunistas é o cenário no qual, temos alunos de um campus universitário enviando arquivos ou dados através do protocolo *Bluetooth*. Nesse momento os nós devem aproveitar as vantagens provenientes de contatos oportunistas para estabelecer uma comunicação com outros nós visando o encaminhamento de mensagens. Os contatos oportunistas ocorrem quando pares de alunos estão dentro do raio de alcance dos rádios e deixam de existir quando se afastam do raio de alcance, durando um quantidade indeterminada de tempo (ou seja, até que o *link* seja perdido ou terminado).

5. **Contatos Previsíveis:** Nos contatos previsíveis os nós DTN podem ter sua ocorrência prevista através de um histórico de contatos. Com entendimento do horário e da duração dos próximos contatos com base nos dados históricos de contatos realizados anteriormente ou fundamentados em outro tipo de informação. E dada a confiança suficiente em um contato previsto, as rotas podem ser escolhidas com base nessas informações. Mesmo com os dados históricos de contatos realizados anteriormente, deve-se, todavia, ressaltar que os contatos previsíveis apresentam um certo grau de incerteza.

2.3 Ambientes de Testes Reais, Emulação e Simulação

A proposta de novos protocolos ou técnicas precisa estar embasada em avaliações de desempenho para que seja comprovado que um determinado novo protocolo ou técnica é superior ao atual.

Tradicionalmente, a pesquisa em redes tem utilizado diversas ferramentas como ambientes de testes reais, simulações, análise matemática e mais recentemente, emulações de rede (Liu et al., 2007). Segundo Zimmermann et al. (2007), cada técnica possui um nível diferente de abstração das características presentes em redes reais e isto torna os resultados obtidos mais fáceis ou difíceis de serem transportados para a realidade.

Quanto mais parâmetros de entrada são considerados em uma análise de desempenho mais precisas podem ser as conclusões obtidas em relação ao mundo real. Além disso, segundo Jain (1991) existem diferentes custo-benefícios em relação a cada método, como o estágio em que este pode ser aplicado à pesquisa, o tempo requerido para obtenção dos resultados e o poder de convencimento dos mesmos.

Simulações e emulações são muito úteis durante a fase de avaliação de arquiteturas e protocolos para redes em geral. Em relação à intenção de avaliar a proposta do protocolo DTN junto ao protocolo CoAP, foram utilizados simuladores e emuladores. Esses ambientes de desenvolvimento permitem modelar uma rede de computadores arbitrária, especificando o comportamento dos elementos da rede, bem como os enlaces de comunicação. Esta seção apresenta os principais simuladores e emuladores de redes de computadores que possuem suporte para IoT e DTN.

A principal diferença entre **Simuladores** e **Emuladores**, é que simulação tem a ver com imitar o comportamento de um sistema sem necessariamente reproduzir seus componentes ou saber como ele funciona internamente na prática. Já a emulação tem a ver com a possibilidade de reconstruir um sistema a partir do entendimento do funcionamento do mesmo, de forma que o resultado seja bastante semelhante ao original.

2.3.1 Cenários de Redes

Simuladores de rede são as ferramentas geralmente escolhidas para estudo e avaliação de desempenho de redes devido à flexibilidade que elas oferecem na criação de cenários grandes, complexos e o controle durante a execução dos experimentos (Morgenroth et al., 2010).

Existem hoje alguns simuladores de rede conceituados como o *Opportunistic Network Environment*, ou **ONE**, é uma ferramenta *open source*, desenvolvida em Java, que simula de forma exemplar uma DTN. Implementando todos os mecanismos necessários ao funcionamento de uma Rede Tolerante a Atrasos.

Com relação aos cenários IoT os dispositivos inteligentes assim como os computadores, também utilizam sistemas operacionais. Entretanto, como os objetos inteligentes possuem recursos físicos limitados, os Sistemas Operacionais - (SO) para esses objetos devem ser muito menores e consumir menos recursos. Um dos principais Sistemas Operacionais para objetos inteligentes é o **ContikiOS**, leve para sistemas embarcados de rede, que fornece mecanismos para o desenvolvimento de softwares para IoT e mecanismos de comunicação que permitem a comunicação dos dispositivos inteligentes. O Contiki pode se comunicar diretamente com outros aplicativos baseados em IP e serviços Web.

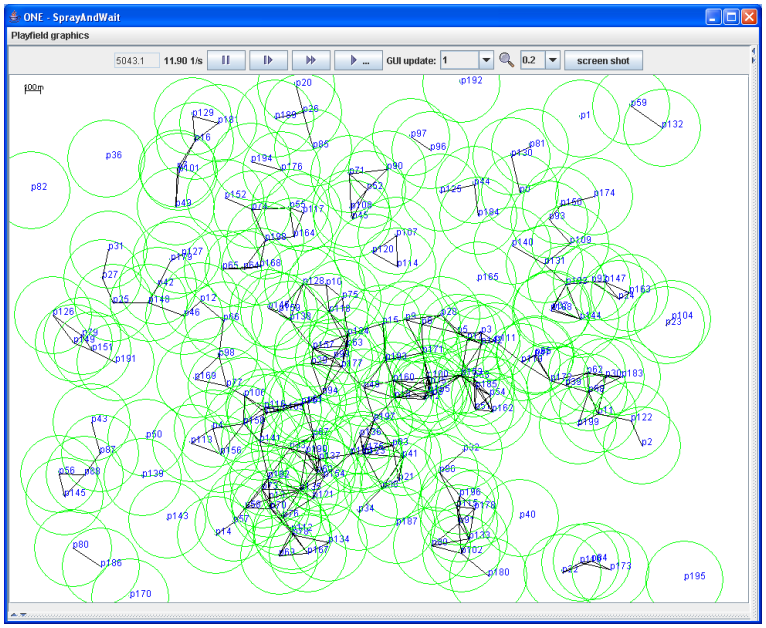
A execução dessas simulações ocorre por meio do **Cooja**, que é um emulador de aplicações do Sistema Operacional Contiki, desenvolvido na linguagem Java. As simulações no Cooja consistem em cenários onde cada nó possui um tipo, memória e um número de interfaces (Österlind, 2006). Um nó no Cooja é um sistema Contiki real compilado e executado. Esse sistema é controlado e analisado pelo Cooja, que possui uma interface para analisar e interagir com os nós, o que facilita o trabalho e a visualização da rede. Além disso, é possível criar cenários personalizados.

2.3.2 Simulador ONE

O ONE é um ambiente de simulação capaz de gerar movimento de nós usando diferentes modelos de movimento e roteamento de mensagens entre nós com vários algoritmos de roteamento DTN e tipos de emissor e receptor. Visualizando a mobilidade e a transmissão de mensagens em tempo real em sua interface gráfica com o usuário.

O ONE pode importar dados de mobilidade de rastreamentos do mundo real ou de outros geradores de mobilidade. Também pode produzir uma variedade de relatórios, da movimentação de nós à passagem de mensagens e estatísticas gerais (Keränen et al., 2009).

Figura 2.7: Waypoint aleatório com 200 nós e alcance de rádio de 250m



Fonte: Keränen et al. (2009).

Uma característica importante do ONE é sua capacidade de interagir com outros programas e fontes de dados. O simulador possui interfaces, por exemplo, para movimento de nó, conectividade e rastreamentos do roteamento de mensagens. Os resultados da simulação são coletados principalmente por meio de relatórios gerados e realizado pelos módulos de relatório durante a execução da simulação (Keränen et al., 2009).

2.3.3 Cooja

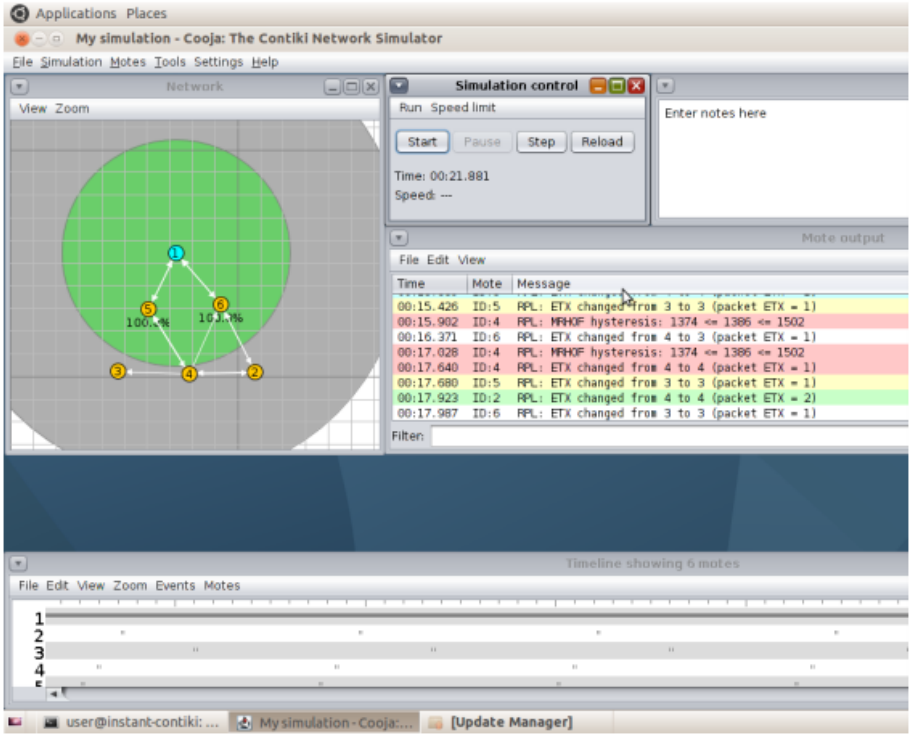
Cooja é o emulador de rede do Sistema Operacional Contiki. O Cooja permite simular redes grandes e pequenas de motes Contiki. Os Motes podem ser emulados no nível de hardware, que é mais lento, mas permite uma inspeção precisa do comportamento do sistema, ou em um nível menos detalhado, que é mais rápido e permite a simulação de redes maiores (ContikiOS, 2019).

O Cooja é uma ferramenta útil para o desenvolvimento do Contiki, pois permite aos desenvolvedores testar seus códigos e sistemas muito antes de executá-lo no hardware

do destino. Os desenvolvedores estabelecem regularmente novas simulações tanto para depurar o software quanto verificar o comportamento de seus sistemas.

O ContikiOS possui um sistema de compilação destinado a facilitar a execução do Contiki diretamente em hardware. O sistema de compilação foi projetado para ser o mesmo em diferentes plataformas de hardware, de modo que os comandos de compilação sejam familiares ao mudar de hardware. Um Mote Contiki simulado no Cooja é um sistema Contiki compilado e executado. O sistema é controlado e analisado pelo Cooja. Isso é realizado compilando o sistema Contiki para a plataforma nativa como uma biblioteca compartilhada e carregando a biblioteca em Java usando Java Native Interfaces (JNI). Várias bibliotecas diferentes do Contiki podem ser compiladas e carregadas na mesma simulação Cooja, representando diferentes tipos de nós de sensores (ContikiOS, 2019).

Figura 2.8: Opções de visualização do cooja sendo utilizado



Fonte: Thomson et al. (2016).

2.3.4 Ambientes de testes reais

Ambientes de testes reais geralmente proporcionam o mais alto grau de confiabilidade e convencimento dos dados obtidos quando possuem uma metodologia bem executada, pois geralmente é mais fácil acreditar em resultantes de um sistema implementado de forma muito parecida com as redes existentes na realidade. O fato de um ambiente real estar disponível, nos garante que, quando bem conduzida, a experimentação nos levará a resultados próximos dos observados no mundo real (Bittencourt e others, 2013).

Constantemente, para atingir o nível de confiabilidade desejado são necessárias diversas

iterações de coletas de dados além de que deve-se ter a preocupação constante com as adversidades impostas pelo ambiente e não previstas durante o projeto do experimento o que prejudica a reprodutibilidade dos resultados.

De acordo com, Iannone et al. (2006) outros problemas desta técnica ocorrem quando é necessário fazer análises de alternativas de configurações ou quando deseja-se investigar alguma tecnologia que depende de hardware completamente novo.

Além disso, a interrupção dos serviços à cada nova necessidade de reconfiguração dos equipamentos durante os testes na rede pode deixar frustrados os usuários que fazem uso desse ambiente de testes.

Com isso, um exemplo desses testes reais para implementar o *Bundle Protocol* junto ao MQTT seria, interligar dois microcontroladores em uma rede *End-to-end* fazendo com que se comuniquem, e logo em seguida cortamos sua conexão e testificaremos o protocolo DTN entrando em ação.

Capítulo 3

Desenvolvimento

\A big breakthrough solves a big problem, but there is always a hint of breakthrough in solving any problem. The Problem may be modest, but if it de es curiosity and calls into play the inventive faculties, those who solve it by their own will experience a sense of self-con dence and enjoy the triumph of discovery."
George Polya

A solução apresentada neste Capítulo utilizou os conceitos apresentados no Capítulo 2. Durante o desenvolvimento da aplicação, esses conceitos foram aplicados para que fosse possível obter resultados satisfatórios para o problema de atraso e desconexões na Internet das Coisas.

Este Capítulo está organizado da seguinte forma. Na Seção 3.1 encontra-se uma breve descrição do desenvolvimento do trabalho, destacando-se a metodologia que utilizada. Em seguida, na Seção 3.2, é detalhada a concepção dos experimentos propostos, bem como suas motivações e objetivos. Por fim, na Seção 3.3, os resultados dos experimentos realizados são apresentados e analisados.

3.1 Metodologia

O desenvolvimento deste trabalho se deu, inicialmente, com o estudo das principais questões e soluções presentes no cenário das redes tolerantes a atraso e desconexões. Em seguida, foram definidas estratégias de utilização das soluções de DTN para a IoT com o objetivo de evitar a perda de dados devido as conexões intermitentes que são características desses cenários de aplicação.

O cenário de aplicação de IoT abordado foi o dos Contatos Persistentes, com a intenção de aliviar as perdas de dados dos objetos inteligentes que estão sempre se comunicando. A estratégia adotada para a solução desse problema foi a de verificar se o protocolo MQTT com implementação do *Bundle Protocol* ofereceria um desempenho razoável quando a conectividade sofresse com a ocorrência de desconexões rápidas e duradouras.

Investigações dessa estratégia foram realizadas em diferentes cenários, via simulação, e os mais promissores foram escolhidos para realização de testes reais. Por fim, realizou-se a análise dos resultados obtidos neste trabalho, a qual permitiu avaliar a utilização dos protocolos BP e MQTT em conjunto.

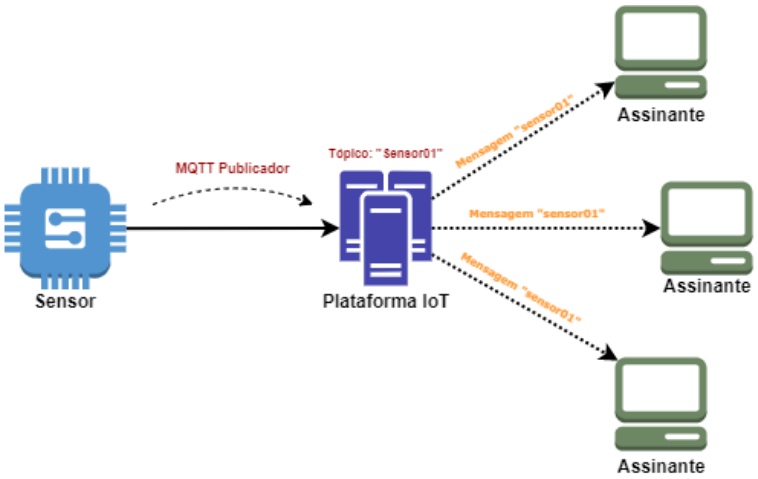
3.2 Modelo Proposto

A fim de desenvolver com êxito uma solução, é necessário olhar para o domínio do problema em maior detalhe. Ao mesmo tempo que um número de protocolos de comunicação da internet das coisas são discutidas na Seção 2.1.4, o MQTT foi escolhido para este projeto.

A implementação do modelo seguiu o caminho onde fosse possível gerar um produto final, isto é, um protótipo, para solucionar o problema de atraso e desconexões quando a realização dos experimentos fossem concluídos. Assim, para tornar a implementação mais simples de entender e usar, foi escolhida a linguagem Python[i] principalmente por sua simplicidade.

A fim de executar MQTT sobre o *Bundle Protocol* (BP), um número de possibilidades foram consideradas. A primeira dessas foi conectar um sensor a uma plataforma IoT onde encaminha-se mensagens via MQTT.

Figura 3.1: Exemplo convencional de conexão na IoT



Fonte: Elaborada pelo autor.

Na Figura 3.1 podemos ver como é concebida a conexão de dispositivos inteligentes para publicação de dados/mensagens em plataformas IoT. Um exemplo dessas plataformas seria o ThingSpeak[ii] uma plataforma de análise IoT que permite adicionar, visualizar e analisar fluxos de dados em tempo real, na nuvem, oriundos de dispositivos inteligentes.

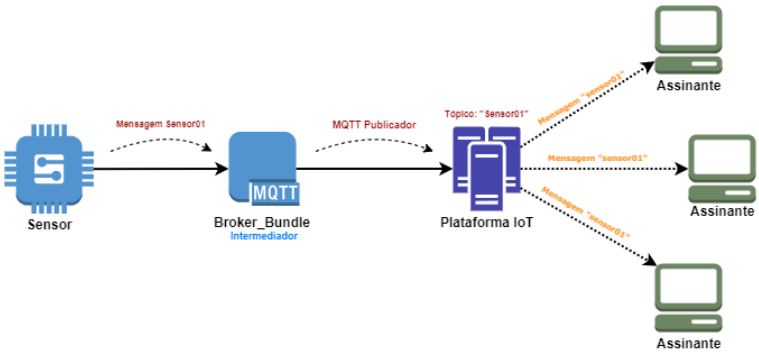
O padrão de *Publish/Subscribe* é uma alternativa para o modelo cliente-servidor tradicional. O cliente MQTT publica mensagens em uma servidor/plataforma IoT, para outros clientes que são assinantes. Cada mensagem tem um endereço, conhecido como um tópico. Os clientes podem subscrever em vários tópicos, tornando-se aptos a receber as mensagens publicadas em cada tópico em que estiverem inscritos.

Com base nessa abordagem tradicional, podemos perceber que ocorre alguns riscos de atraso ou desconexões durante a publicação da mensagem do sensor para plataforma IoT. Ainda que a ideia inicial, seja muito utilizada nos dias atuais, com o estudo desse trabalho

[i] https://www.python.org/
[ii] https://thingspeak.com/

foi pensado em uma solução para amenizar as perdas no cenário descrito anteriormente. Na Figura 3.2 está ilustrada a solução proposta para amenizar a perda de dados devido as desconexões na IoT. Observe que o *broker* é implementado externamente ao sensor, decisão motivada devido as limitações do sensor.

Figura 3.2: Modelo proposto para amenizar a perda de dados nas conexões IoT



Fonte: Elaborada pelo autor.

Um *Broker* MQTT é inserido na mesma rede que o sensor para gerenciar e direcionar as mensagens dentro da rede juntamente com o protocolo *Bundle*, criando uma fila em que armazena todos os dados transmitidos do sensor perante as falhas e desconexões. Com isso quando a conexão é restabelecida o *Broker* publica as mensagens da fila na plataforma IoT, permitindo que os dados não sejam descartados ou perdidos.

3.3 Experimentos

Nesta seção são descritos os materiais utilizados, designer experimental dos testes assim como o cenário DTN simulado. O objetivo dos experimentos foi avaliar o funcionamento e desempenho do *Broker* MQTT com a integração do *Bundle Protocol* em um cenário de desconexões, simulando seu uso operando em forma no qual guarda a mensagem em uma fila quando a conexão é perdida. O tipo de contato DTN adotado foram os Contatos Persistentes. Durante os experimentos foram elaborados dois cenários práticos que serão mostrados adiante.

3.3.1 Ambiente

Hardware

Um Raspberry Pi 2 modelo B (especificações na Tabela 3.1), ESP8266 modelo ESP-01 (especificações na Tabela 3.2) e um notebook foram utilizados nos experimentos. No notebook foi executado a função de cliente *subscribers* e também foi inicializado um serviço broker MQTT para simular a plataforma IoT.

Tabela 3.1: Especificações do Raspberry Pi 2 modelo B

Componente	Especificação
Processador	ARM Cortex A7 900MHz Quad-core
Disco	Micro SD 8GB
Memória	1GB
S.O.	Noobs Lite 2019-10
Ethernet	10/100 Mbps
Alimentação	Micro USB 5V/1.8A

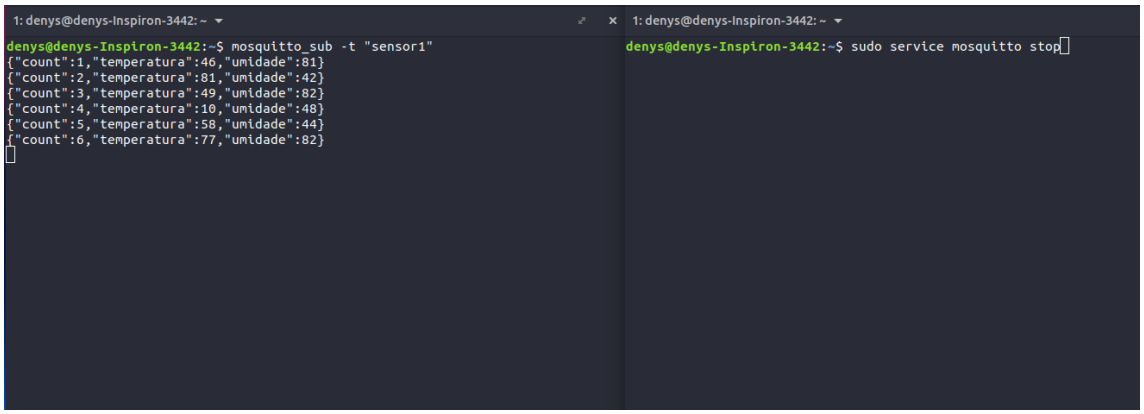
Tabela 3.2: Especificações ESP8266 modelo ESP-01

Componente	Especificação
Microprocessador	ARM de 32 bits
Memória	512KBytes
Wi-Fi	802.11 B/G/N
Alimentação	3.3 VDC

Design experimental

Observando o cenário da Figura 3.3 podemos perceber como funciona o modelo habitual. Na qual tem-se a comunicação direta do sensor com a plataforma IoT. Em que neste cenário é forçada uma desconexão do serviço da plataforma com o sensor que esta enviando dados para aquele tópico especifico. Na qual a Figura 3.4 é apresentado o caso onde a conexão é restabelecida e ao voltar a comunicação alguns dados medidos pelo sensor foram perdidos, dados esses que podemos perceber diretamente na figura onde tem-se um contador a cada dado medido pelo sensor.

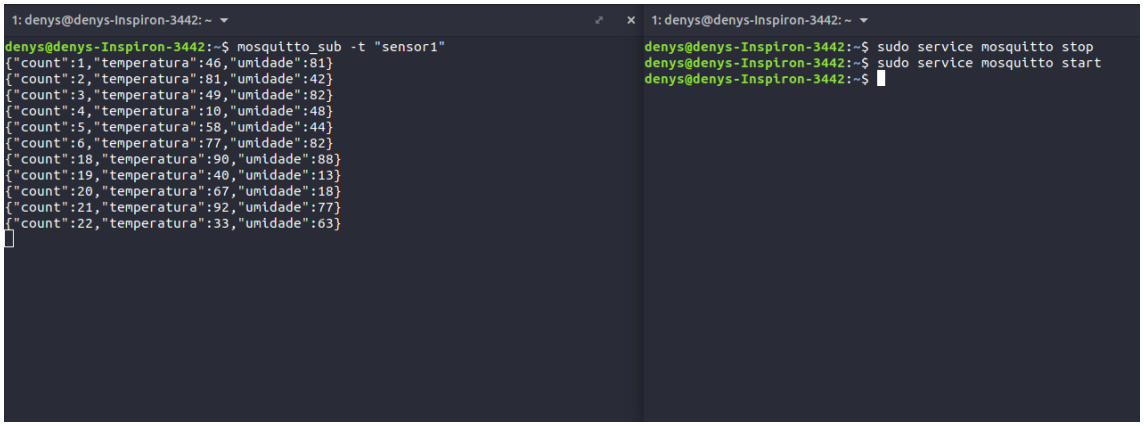
Figura 3.3: Inicializando conexão a plataforma IoT e causando uma desconexão.



Fonte: Elaborada pelo autor.

Com o propósito de amenizar essas perdas de dados enviadas pelo sensor a plataforma IoT, foi desenvolvido um código operando junto ao Broker Bundle MQTT na qual está

Figura 3.4: Restabelecendo conexão com a plataforma IoT



Fonte: Elaborada pelo autor.

inserido na mesma rede do sensor que envia os dados à plataforma. Executando em forma de armazenamento da mensagem em fila quando a conexão é perdida. Assim, esse recurso beneficia redes com pequenos atrasos.

algoritmo esperaMensagens()
1: *f*Variáveis globais.*g*
2: $novamsg$ = **false** *f*Indica se há nova mensagem.*g*
3: $fila$ = *fg*
4: **enquanto true faça**
5: **se** $novamsg$ = **true então**
6: $fila.push(msg)$
7: **fim se**
8: **fim enquanto**

Algoritmo 3.1: Algoritimo implementado no Broker Bundle para recepção e armazenamento das mensagens do sensor.

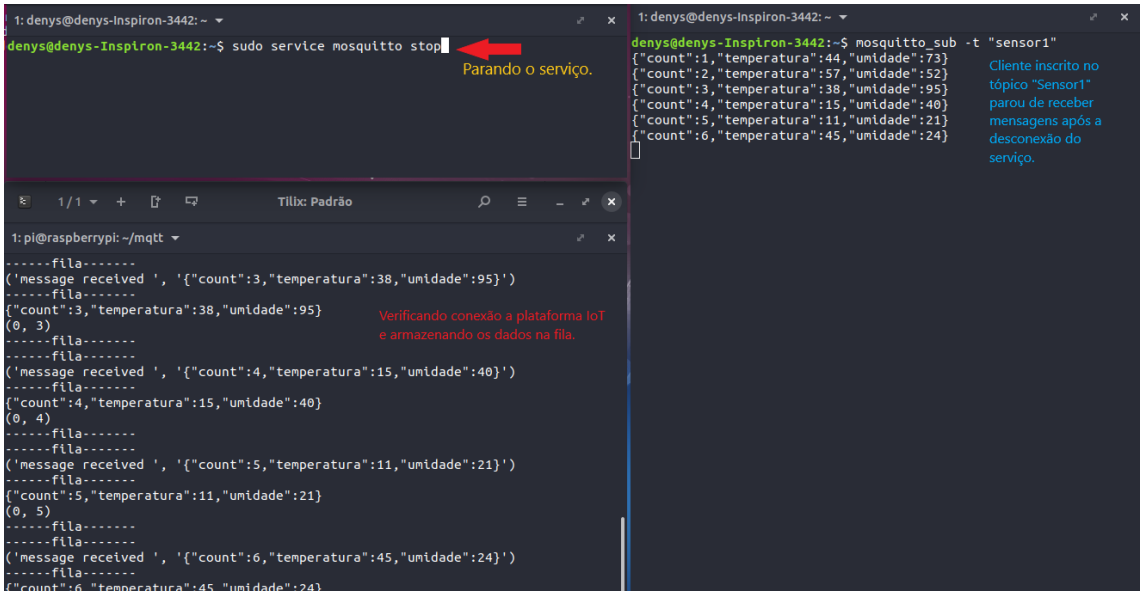
Com base nos Algoritimos 3.1 e 3.2, para recepção e envio das mensagens do sensor, respectivamente, foi desenvolvido um segundo cenário, ilustrado na Figura 3.2, na qual deu-se a solução desse trabalho. Desta forma, o experimento é iniciado como observado na Figura 3.5 onde o Broker está desempenhando o papel de *publisher* e *subscriber* ao mesmo tempo, onde recebe a mensagem do sensor e encaminha essa mensagem à plataforma IoT. Quando a conexão é perdida entre o Broker MQTT e a plataforma IoT é onde o agente bundle entra em ação, encaminhado todos os dados para uma fila no raspberry pi até que a conexão seja restabelecida novamente.

Dessa maneira, todos os dados armazenados na fila do broker bundle são encaminhados para plataforma IoT como mostra a Figura 3.6, logo nenhum dado foi disperso perante esse cenário. Com isso, mostra que a solução se torna viável para amenizar as perdas de dados no cenário proposto.

algoritmo publicaMensagens()
1: *f*Variáveis globais.*g*
2: $conexao$ = **true**
3: $fila$ = *fg*
4: **enquanto true faça**
5: **se** $conexao$ = **true então**
6: **enquanto** $jfilaj > 0$ **faça**
7: $publica(fila.pop())$
8: **fim enquanto**
9: **else**
10: *f*Inicia tentativa de reconexão.*g*
11: **fim se**
12: **fim enquanto**

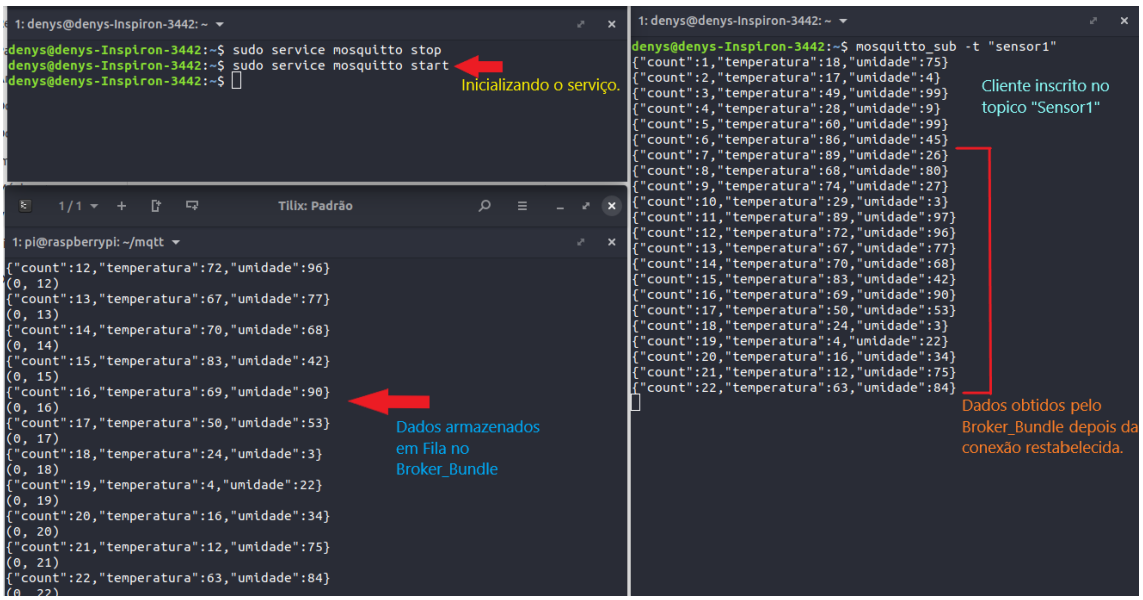
Algoritmo 3.2: Algoritimo implementado no Broker Bundle para envio das mensagens do Sensor à plataforma IoT.

Figura 3.5: Conexão entre a plataforma IoT com *Broker Bundle*



Fonte: Elaborada pelo autor.

Figura 3.6: Restabelecendo conexão a plataforma IoT



Fonte: Elaborada pelo autor.

Capítulo 4

Considerações Finais

\If we can really understand the problem,
the answer will come out of it,
because the answer is not separate from the problem."
Jiddu Krishnamurti

Todo o conhecimento apresentado durante o trabalho serviu para entender e desenvolver uma solução capaz de amenizar o problema da entrega de dados mesmo sob as conexões intermitentes no cenário de internet das coisas. Essa solução foi capaz de obter bons resultados e ao mesmo tempo servir como base para iniciantes que buscam compreender o problema em questão e as abordagens que podem ser utilizadas para resolvê-los.

Conclui-se que as atividades deste projeto e objetivos planejados antes e durante seu andamento foram desempenhados de forma satisfatória, mostrando a relevância do trabalho nas áreas tecnológicas envolvidas, das quais se destacam internet das coisas e redes tolerantes a atraso e desconexões.

Assim, este Capítulo se resume no que foi notado durante a execução dos experimentos, o que pode ser feito para melhorar os resultados obtidos e a conclusão que surgiu ao analisar resultados dos experimentos.

4.1 Resultados

No geral, o projeto foi bem sucedido. Os objetivos do projeto foram cumpridos, e seu desempenho foi analisado. O artefato desenvolvido neste trabalho pode ser usado como base para criação de uma solução mais complexa, auxiliando na elaboração de novos estudos sobre os temas e integrações com outras aplicações.

Os resultados obtidos evidenciaram que o *Bundle Protocol* (BP) desenvolvido junto ao broker MQTT mostrou-se bem adaptado para as mensagens utilizadas na coleta de resultados, o que reflete um equilíbrio obtido com o uso do Broker sendo um intermediário em meio a comunicação com a plataforma IoT.

Além disso, espera-se que, com as modificações citadas em Trabalhos Futuros, seção 4.2, o desempenho melhore de forma significativa. Por fim, com a estrutura que foi criada, pode-se, agora, concentrar-se no desenvolvimento de tarefas de mais alto nível para o refinamento dos algoritmos utilizados, para a adição de novas formulas e cenários IoT e em suas adaptações para situações específicas.

4.2 Trabalhos futuros

Para trabalhos futuros existem algumas tarefas que podem ser implementadas, seja para aperfeiçoamento da solução criada, para implementação de novas funcionalidades ou integração com outros protocolos. No que se refere ao aperfeiçoamento da solução pode-se realizar melhorias em sua análise real (protótipo), adicionando mais dispositivos inteligentes a rede, que favoreça guardar todos os dados por um maior tempo de custódia em cada nó DTN evitando assim uma maior perca dos dados.

Embora essa nova forma de comunicação tenha gerado grande interesse na comunidade de pesquisa, ainda esta em sua infância em termos de arquiteturas, algoritmos, protocolos, ferramentas, modelagem e padrões emergentes de comunicações.

Portanto, é muito promissor identificar aspectos para o desenvolvimento de padrões para DTNs. Isso exigiria considerar criteriosamente as várias atividades em uma DTN para essa finalidade. Desse modo, o esforço visaria remodelar as redes tolerantes a atrasos no contexto de internet das coisas.

Referências Bibliográficas

Al-Fuqaha, Ala, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari e Moussa Ayyash (2015), 'Internet of things: A survey on enabling technologies, protocols, and applications', *IEEE communications surveys & tutorials* **17**(4), 2347–2376.
(Citado na página 7)

Al Hanbali, Ahmad, Eitan Altman e Philippe Nain (2005), 'A survey of tcp over ad hoc networks.', *IEEE Communications Surveys and Tutorials* **7**(1-4), 22–36.
(Citado na página 10)

Ashton, Kevin (2009), 'That 'internet of things' thing', *RFID journal* **22**(7), 97–114.
(Citado na página 6)

Atzori, Luigi, Antonio Iera e Giacomo Morabito (2010), 'The internet of things: A survey', *Computer networks* **54**(15), 2787–2805.
(Citado na página 1)

Auzias, Maël, Yves Mahéo e Frédéric Raimbault (2015), Coap over bp for a delay-tolerant internet of things, *em* '2015 3rd International Conference on Future Internet of Things and Cloud', IEEE, pp. 118–123.
(Citado na página 2)

Azzola, Francesco (2018), 'Coap protocol: Step-by-step guide'. Acesso em: 13 out 2019.
URL: *https://dzone.com/articles/coap-protocol-step-by-step-guide*
(Citado nas páginas 8 e 9)

Bittencourt, Daniel da Costa et al. (2013), 'Avaliaçao de desempenho de redes tolerantes a atrasos e desconexoes utilizando a técnica de virtualizaçao'.
(Citado na página 16)

Chen, Ling-Jyh, Chen-Hung Yu, Tony Sun, Yung-Chih Chen e Hao-hua Chu (2006), A hybrid routing approach for opportunistic networks, *em* 'Proceedings of the 2006 SIGCOMM workshop on Challenged networks', ACM, pp. 213–220.
(Citado na página 2)

ContikiOS (2019), 'Get started with contiki'. Acesso em: 26 set 2019.
URL: *http://www.contiki-os.org/start.html*
(Citado nas páginas 15 e 16)

Da Xu, Li, Wu He e Shancang Li (2014), 'Internet of things in industries: A survey', *IEEE Transactions on industrial informatics* **10**(4), 2233–2243.
(Citado na página 6)

Gusmao, Gabriel e Parcilene Brito (2015), MODELO DE INTERNET DAS COISAS PARA O PARQUE ESTADUAL DO CANTÃO, Tese de doutorado.
(Citado nas páginas 7 e 8)

Iannone, Luigi, Konstantin Kabassanov e Serge Fdida (2006), The meshdvnet wireless mesh network test-bed, *em* 'Proceedings of the 1st International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization', WiNTECH '06, ACM, New York, NY, USA, pp. 107–108.
URL: *http://doi.acm.org/10.1145/1160987.1161014*
(Citado na página 17)

Jain, R. (1991), 'The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling'.
(Citado na página 14)

Kelly, Sean Dieter Tebje, Nagender Kumar Suryadevara e Subhas Chandra Mukhopadhyay (2013), 'Towards the implementation of iot for environmental condition monitoring in homes', *IEEE sensors journal* **13**(10), 3846–3853.
(Citado na página 6)

Keränen, Ari, Jorg Ött e Teemu Kärkkäinen (2009), Simulador ONE para avaliação de protocolo DTN, *em* ICST, ed., 'SIMUTools '09: Anais da 2ª Conferência Internacional sobre Ferramentas e Técnicas de Simulação', Nova York, NY, EUA.
(Citado na página 15)

Koreshoff, Treffyn Lynch, Toni Robertson e Tuck Wah Leong (2013), Internet of things: a review of literature and products, *em* 'Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration', ACM, pp. 335–344.
(Citado na página 1)

Kurose, James F e Keith W Ross (2012), 'Computer networking: A top down approach sixth edition'.
(Citado na página 5)

Liu, Jason, Scott Mann, Nathanael Van Vorst e Keith Hellman (2007), An open and scalable emulation infrastructure for large-scale real-time network simulations, *em* 'IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications', IEEE, pp. 2476–2480.
(Citado na página 14)

Luzuriaga, Jorge E, Marco Zennaro, Juan Carlos Cano, Carlos Calafate e Pietro Manzoni (2017), A disruption tolerant architecture based on mqtt for iot applications, *em* '2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)', IEEE, pp. 71–76.
(Citado na página 1)

Morgenroth, Johannes, Sebastian Schildt e Lars Wolf (2010), Hydra: virtualized distributed testbed for dtn simulations, *em* 'Proceedings of the fifth ACM international workshop on Wireless network testbeds, experimental evaluation and characterization',

ACM, pp. 71–78.
(Citado na página 14)

Mustafa, Amr (2018), 'Mqtt protocol'. Acesso em: 13 out 2019.
URL: *https://1sheeld.com/mqtt-protocol/*
(Citado na página 9)

Oliveira, Carina Teixeira de (2008), Uma Proposta de Roteamento Probabilístico para Redes Tolerantes a Atrasos e Desconexões, Tese de doutorado, UNIVERSIDADE FEDERAL DO RIO DE JANEIRO.
(Citado nas páginas 10, 11 e 13)

Österlind, Fredrik (2006), *A sensor network simulator for the Contiki OS*, Swedish Institute of Computer Science.
(Citado na página 14)

POSTEL, J (1981), 'Transmission control protocol (tcp). rfc 793', *Information Sciences Institute Web site.. Published http://www.ietf.org/rfc/rfc793.txt. Updated* .
(Citado na página 10)

Ruiz, Linnyer Beatrys, Luiz Henrique A Correia, Luiz Filipe M Vieira, Daniel F Macedo, Eduardo F Nakamura, Carlos MS Figueiredo, Marcos Augusto M Vieira, Eduardo Habib Bechelane, Daniel Camara, Antonio AF Loureiro et al. (2004), 'Arquiteturas para redes de sensores sem fio', *Tutorial of the simposio Brasileiro de Redes de Computadores e Sistemas Distribu dos(SBRC)* .
(Citado na página 3)

Santos, Bruno P, Lucas AM Silva, CSFS Celes, João B Borges, Bruna S Peres Neto, Marcos Augusto M Vieira, Luiz Filipe M Vieira, Olga N Goussevskaia e AA Loureiro (2016), 'Internet das coisas: da teoria à prática', *Anais do Simposio Brasileiro de Redes de Computadores e Sistemas Distribuidos (SBRC)* .
(Citado nas páginas 3, 5, 6 e 7)

Scott, K, S Burleigh et al. (2007), 'Rfc 5050: bundle protocol specification', *IRTF DTN Research Group* .
(Citado na página 11)

Thomson, Craig, Imed Romdhani, Ahmed Al-Dubai, Mamoun Qasem, Barraq Ghaleb e Wadhaj Isam (2016), Cooja Simulator Manual, Relatório técnico, Edinburgh Napier University. Acesso em: 26 set 2019.
URL: *https://www.napier.ac.uk/~/media/worktribe/output-299955/cooja-simulator-manual.pdf*
(Citado na página 16)

Torgerson, Leigh, Scott C. Burleigh, Howard Weiss, Adrian J. Hooke, Kevin Fall, Dr. Vinton G. Cerf, Keith Scott e Robert C. Durst (2007), 'Delay-Tolerant Networking Architecture', RFC 4838.
URL: *https://rfc-editor.org/rfc/rfc4838.txt*
(Citado na página 12)

Warthman (2012), 'Delay-and disruption-tolerant networks (dtns)', *A Tutorial. V.. 0, Interplanetary Internet Special Interest Group* pp. 5–9.
(Citado nas páginas 11 e 12)

Zimmermann, Alexander, Mesut Gunes, Martin Wenig, Ulrich Meis e Jan Ritzerfeld (2007), How to study wireless mesh networks: A hybrid testbed approach, *em* '21st International Conference on Advanced Information Networking and Applications (AINA'07)', IEEE, pp. 853–860.
(Citado na página 14)