

UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA
NÍVEL MESTRADO PROFISSIONAL

ARMANDO LEOPOLDO KELLER

INTERNET DAS COISAS APLICADA À INDÚSTRIA:
DISPOSITIVO PARA INTEROPERABILIDADE DE
REDES INDUSTRIAIS

SÃO LEOPOLDO

2016

Armando Leopoldo Keller

**Internet das Coisas aplicada à Indústria: Dispositivo para
Interoperabilidade de Redes Industriais**

Trabalho apresentado como requisito para a obtenção do título de Mestre, pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade do Vale do Rio dos Sinos – UNISINOS.

Orientador: Prof. Dr. Marcio R. da Silva

Coorientador: Prof. Me. Lúcio Renê Prade

São Leopoldo

2016

K29i

Keller, Armando Leopoldo.

Internet das coisas aplicada à indústria: dispositivo para interoperabilidade de redes industriais / Armando Leopoldo Keller – 2016.

54 f. : il. ; 30 cm.

Dissertação (mestrado) – Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Engenharia Elétrica, 2016.

“Orientador: Prof. Dr. Marcio R. da Silva.”

“Coorientador: Prof. Me. Lúcio Renê Prade.”

1. Internet. 2. Máquinas. 3. Redes elétricas inteligentes. 4. Geração distribuição de energia elétrica. I. Silva, Marcio R. da. II. Prade, Lúcio Renê. III. Título.

CDU 621.3

Dados Internacionais de Catalogação na Publicação (CIP)
(Bibliotecária: Raquel Herbcz França – CRB 10/1795)

Armando Leopoldo Keller

Internet das Coisas aplicada à Indústria: Dispositivo para Interoperabilidade de Redes Industriais

Trabalho apresentado como requisito para a obtenção do título de Mestre, pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade do Vale do Rio dos Sinos – UNISINOS.

Aprovado em 13 de Janeiro de 2017.

BANCA EXAMINADORA:

Prof. Dr. Paulo Ricardo Pereira da Silva –
UNISINOS
Avaliador

Prof. Dr. Rodrigo Marques de Figueiredo –
UNISINOS
Avaliador

Prof. Dr. João Batista Dias – UNISINOS
Avaliador Externo

Prof. Dr. Marcio R. da Silva (Orientador)
Prof. Me. Lúcio Renê Prade (Coorientador)

Visto e permitida a impressão
São Leopoldo

Prof. Dr. Eduardo Luis Rhod
Coordenador PPG em Engenharia Elétrica

RESUMO

O objetivo deste trabalho, é realizar um estudo de forma mais abrangente sobre o conceito de Internet das Coisas e seus principais protocolos. Explora-se especificamente o conceito de IoT (Internet of Things) aplicado em sistemas de automação. Para tanto é apresentada uma revisão bibliográfica sobre o assunto, explorando os diversos protocolos desenvolvidos para aplicações de IoT, caracterizando-os quanto a taxa de transmissão, eficiência, segurança e confiabilidade. Também é realizado um levantamento do cenário atual, quanto a aplicação de protocolos de IoT em sistemas de automação, sempre tendo em mente a confiabilidade do sistema. Percebe-se que um grande dificultador do uso destes tipos de protocolo em ambientes industriais é justamente a heterogeneidade das redes existentes. Diante deste problema, a proposta do trabalho é desenvolver um dispositivo que atue como *middleware* para a interligação de redes de automação distribuídas, no caso especificamente a rede Modbus RTU, fazendo com que esta interligação seja de forma transparente utilizando o protocolo de Internet das Coisas MQTT (*Message Queuing Telemetry Transport*). Este dispositivo é testado com equipamentos em um cenário real através de um estudo de caso, onde duas redes Modbus RTU de um sistema geograficamente distribuído de geração de energia solar fotovoltaica, são interligadas, permitindo a criação de uma planta virtual de geração de energia do inglês *virtual power plant (VPP)*. Com isso é possível tratar e gerenciar os sistemas distribuídos de geração como sendo uma única unidade geradora, facilitando o despacho. Para comprovar a eficiência e a confiabilidade do sistema, foram realizados testes onde o tempo entre as requisições e respostas foi medido, e através da sua distribuição foi obtido um tempo de 2,5 segundos para obter uma comunicação com baixa taxa de perda de mensagens. Estes testes comprovam o correto funcionamento do sistema proposto.

Palavras-chaves: Internet das Coisas. Máquina para Máquina. Internet Industrial das Coisas. MQTT.

ABSTRACT

The objective of this work is to develop a more comprehensive study on the concept of Internet of Things (IoT) and its main protocols, specifically exploring the concept of IoT applied in automation systems. A bibliographic review explores the diverse protocols developed for IoT applications, characterizing them as transmission rate, efficiency, safety and confiability. A survey of the current scenario about the application of IoT protocols in automation systems is presented, always having the system confiability in mind. The heterogeneity of the existent networks makes the use of this protocols a harder task. The proposal of this work is develop a device that acts as middleware for interlink distributed automation networks, in this case the Modbus RTU networks, in a transparent way using the internet of things procol MQTT (Message Queuing Telemetry Transport). This device is tested with equipments in a real scenario trough a case study, where two Modbus RTU networks of a geographically distributed solar photovoltaic power plant, is interlinked, allowing the criation of a VPP (Virtual Power Plant). This makes possible to manage the distributed power generator systems as a single generator unit, improving the electric energy dispatch. To prove the efficiency and confiability of the system, tests were made where the time between request and response was measured, and based on his distribution the time of 2.5 seconds was determined to have a low message loss communication. Those tests validate the proposed system and the achievement of the goals of the present work.

Key-words: Internet of Things (IoT). Machine to machine (M2M). Industrial Internet of Things (IIoT). Modbus.

LISTA DE FIGURAS

Figura 1 – Segmentação da Internet das Coisas.	17
Figura 2 – Funcionamento do modelo publicador/assinante.	18
Figura 3 – Tópicos MQTT.	19
Figura 4 – Níveis de QoS do MQTT.	20
Figura 5 – Histórico revolução industrial.	21
Figura 6 – <i>Frame</i> Modbus	23
Figura 7 – Tempo entre <i>frames</i> Modbus.	24
Figura 8 – Tansação Modbus sem erros.	24
Figura 9 – Tansação Modbus com erros.	25
Figura 10 – Retorno do código de função.	25
Figura 11 – Faixa de grupo de funções.	25
Figura 12 – Modo <i>unicast</i>	26
Figura 13 – Modo <i>broadcast</i>	27
Figura 14 – <i>Frame</i> Modbus TCP/IP.	27
Figura 15 – Conexão TCP/IP.	28
Figura 16 – Comparação das Redes Elétricas Tradicionais com as Redes Elétricas Inteligentes.	29
Figura 17 – Sistema de medição para NEM.	30
Figura 18 – Sistema proposto por Temiz et al. (2015)	31
Figura 19 – Requisição Modbus para leitura de registradores do EM1200.	34
Figura 20 – Dados lidos do EM1200.	34
Figura 21 – Dados lidos do EM1200 com troca de pacotes.	35
Figura 22 – Respostas para testes realizados com tecnologia 2G.	35
Figura 23 – Respostas para testes realizados com tecnologia 3G.	36
Figura 24 – Respostas para testes realizados com tecnologia 4G.	36
Figura 25 – Estrutura do <i>middleware</i> proposto por Thangavel et al. (2014).	36
Figura 26 – Atraso para rede cabeada	37
Figura 27 – Perda de pacotes para rede cabeada	38
Figura 28 – Atraso para rede celular	38
Figura 29 – Perda de pacotes para rede celular	39
Figura 30 – Etapas do desenvolvimento do trabalho	40
Figura 31 – Conversor EIA-485 para USB (<i>Universal Serial Bus</i>)	40
Figura 32 – Multimetro de grandezas elétricas ALTUS PH3100	41
Figura 33 – Multimetro de grandezas elétricas ABB Nexus II	42
Figura 34 – Aplicação do dispositivo para medição de tempo de resposta.	43
Figura 35 – Kit de desenvolvimento ST Nucleo 334R8	43

Figura 36 – Logica de funcionamento do microcontrolador.	44
Figura 37 – Conceito do <i>gateway</i> Modbus - MQTT	45
Figura 38 – Caminho dos pacotes.	46
Figura 39 – Algoritmo para conversão das mensagens Modbus para MQTT.	47
Figura 40 – Estrutura das mensagens do <i>gateway</i>	47
Figura 41 – Algoritmo para conversão das mensagens MQTT para Modbus.	48
Figura 42 – VPP utilizada para estudo de caso.	48
Figura 43 – Tempo de resposta do teste 1	49
Figura 44 – Tempo de resposta do teste 2	50
Figura 45 – Tempo de resposta do teste 3	50
Figura 46 – Tempo de resposta geral	51

LISTA DE TABELAS

Tabela 1 – Caracteres curinga.	19
Tabela 2 – Níveis de QoS do MQTT.	20
Tabela 3 – Resumo dos resultados dos testes de tempo de resposta.	49

LISTA DE QUADROS

Quadro 1 – Códigos de funções Modbus	26
Quadro 2 – Mapa de registradores do medidor de energia EM1200.	33
Quadro 3 – Atraso médio em relação a perda de sinal	37
Quadro 4 – Trabalhos relacionados.	39

LISTA DE ABREVIATURAS E SIGLAS

ADU	<i>Application Data Unit</i> (Unidade de Dados de Aplicação)
ANEEL	Agência Nacional de Energia Elétrica
ASCII	<i>American Standard Code for Information Interchange</i> (Código Americano Padrão para Troca de Informações)
CNPJ	Cadastro Nacional de Pessoa Jurídica
CoAP	<i>Constrained Application Protocol</i> (Protocolo de Aplicação Restrita)
CPF	Cadastro de Pessoa Física
CRC	<i>Cyclical Redundancy Check</i> (Verificação de Redundância Cíclica)
EMS	<i>Energy Management System</i> (Sistema de Gerenciamento de Energia)
EPA	<i>Ethernet to Plant Automation</i> (Ethernet para Automação de Instalações)
ERP	<i>Enterprise Resource Planning</i> (Planejamento de Recursos Empresariais)
FiT	<i>Feed-in-tariff</i> (Tarifa de Alimentação)
GPS	<i>Global Positioning System</i> (Sistema de Posicionamento Global)
HIoT	<i>Human Internet of Things</i> (Internet Humana das Coisas)
IHM	Interface Homem Máquina
IIoT	<i>Industrial Internet of Things</i> (Internet Industrial das Coisas)
IoT	<i>Internet of Things</i> (Internet das Coisas)
LRC	<i>Longitudinal Redundancy Check</i> (Verificação de Redundância Longitudinal)
M2M	<i>Machine to Machine</i> (Máquina para Máquina)
MBAP	<i>Modbus Application Protocol Header</i> (Cabeçalho do Protocolo de Aplicação)
MES	<i>Manufacturing Execution Systems</i> (Sistemas de Execução de Manufatura)
MQTT	<i>Message Queuing Telemetry Transport</i> (Enfileiramento de mensagens de transporte de telemetria)
NEM	<i>Net Energy Metering</i> (Medidor de Energia Líquida)

PCS	<i>Process Control Systems</i> (Sistema de Controle de Processos)
PDU	<i>Protocol Data Unit</i> (Unidade de Dados de Protocolo)
PRODIST	Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional
QoS	<i>Quality of Service</i> (Qualidade de Serviço)
REST	<i>Representational State Transfer</i> (Transferência de estado representacional)
RTU	<i>Remote Terminal Unit</i> (Unidade Terminal Remota)
SCADA	<i>Supervisory Control and Data Acquisition</i> (Controle de supervisão e aquisição de dados)
USB	<i>Universal Serial Bus</i> (Barramento Serial Universal)
VPP	<i>Virtual Power Plant</i> (Usina Virtual de Energia)

SUMÁRIO

1	INTRODUÇÃO	13
2	CONCEITOS BÁSICOS	14
2.1	Internet das Coisas	14
2.1.1	Definição	14
2.1.2	Exemplos de aplicações	15
2.1.3	Segmentação	16
2.1.4	Principais Protocolos	16
2.1.4.1	<i>REST</i>	16
2.1.4.2	<i>CoAP</i>	17
2.1.4.3	<i>MQTT</i>	18
2.2	Indústria 4.0	20
2.3	Redes de campo — <i>Fieldbus</i>	22
2.3.1	Modbus	22
2.3.1.1	<i>Modbus-RTU</i>	23
2.3.1.2	<i>Modbus-TCP/IP</i>	26
2.4	Geração distribuída e redes inteligentes	28
3	TRABALHOS RELACIONADOS	31
4	METODOLOGIA	40
4.1	Dispositivo para medição de tempo de resposta em interface EIA-485 . .	42
5	SISTEMA PROPOSTO	45
6	ANÁLISE DE RESULTADOS	49
	Conclusão	52
	REFERÊNCIAS	53

1 INTRODUÇÃO

A comunicação entre dispositivos vem sendo aprimorada ao longo dos últimos anos mudando o conceito de muitos sistemas como as redes de transmissão e distribuição de energia, que com uma comunicação eficiente, podem ser denominadas de redes elétricas inteligentes, muito conhecidas pelo seu nome em inglês *smart grids*. Além das redes de transmissão, dispositivos de uso pessoal e doméstico também estão sendo dotados de comunicação. Outra área de grande relevância econômica, onde a comunicação é de grande importância, são as redes industriais, a comunicação entre máquinas é denominada de M2M (*Machine to Machine*) que em português significa máquina para máquina, sendo um dos ramos da internet das coisas.

Com a grande gama de protocolos de comunicação existentes no mercado, é importante que existam dispositivos que realizem a conversão de dados para interoperabilidade dos dispositivos, estes dispositivos são chamados de *gateways*.

Neste trabalho é proposto o desenvolvimento de um *gateway* para comunicação entre equipamentos industriais que possuem comunicação Modbus, um dos protocolos mais utilizados nas indústrias, que estão instalados em diferentes localidades. Esta comunicação será dada utilizando um protocolo de comunicação desenvolvido especialmente para a internet das coisas, visando a comunicação entre diversos equipamentos, com uma baixa necessidade de processamento e transmissão de dados. Para analisar o desempenho desde dispositivo será realizado um estudo de caso para interligar equipamentos de dois sistemas fotovoltaicos situados em diferentes localidades, criando uma rede virtual.

O presente trabalho está dividido em 7 capítulos, esta introdução, que tem por objetivo apresentar e situar o trabalho, em seguida no capítulo de conceitos básicos são apresentados os principais conceitos necessários para o entendimento do trabalho. Após a apresentação destes conceitos, no capítulo de trabalhos relacionados são apresentados os trabalhos provenientes de uma pesquisa bibliográfica que se assemelham de alguma forma ao sistema proposto neste trabalho. Em metodologia são apresentadas as técnicas e ferramentas utilizadas na execução do trabalho, assim como o fluxo a ser seguido. O sistema é melhor detalhado no capítulo sistema proposto, onde é apresentado o objetivo do trabalho e suas delimitações. Os resultados obtidos são demonstrados e analisados no capítulo análise de resultados. Finalizando, o capítulo de conclusão apresenta as conclusões obtidas com a realização do presente trabalho.

2 CONCEITOS BÁSICOS

Neste capítulo são apresentados os conceitos básicos necessários para o completo entendimento do trabalho desenvolvido, contemplando a definição das principais áreas de estudo como redes de campo e Internet das Coisas, e também um aprofundamento nos protocolos utilizados.

2.1 Internet das Coisas

Nesta seção são apresentados alguns conceitos de Internet das Coisas, assim como exemplos de aplicação, os principais protocolos utilizados e como se difere da computação ubíqua.

2.1.1 Definição

O termo Internet das Coisas, também conhecido pela sua abreviação em língua inglesa IoT (*Internet of Things*) não possui uma definição clara e concreta, sendo muitas vezes utilizado como palavra do momento em pesquisas científicas, e estratégias de marketing e vendas (UCKELMANN; HARRISON; MICHAHELLES, 2011).

Entre as varias definições de Internet das Coisas existentes, tanto Zhou (2012) quanto McEwen e Cassimally (2013) resumem as definições de grandes empresas e grupos científicos, que demonstram os conceitos básicos sobre o que é Internet das Coisas.

Para Cassagras (apud ZHOU, 2012) a Internet das Coisas é definida como uma infraestrutura de rede global que interliga objetos físicos e virtuais explorando a captura de dados e capacidade de comunicação, sendo compatível com a internet atual. Esta estrutura tem como base para o desenvolvimento de serviços e aplicações, a capacidade de identificação de objetos específicos, o sensoriamento e conectividade. Tudo isto ocorrendo em alto nível, com grande interoperabilidade (ZHOU, 2012).

Para SAP (apud ZHOU, 2012), a internet das coisas criara um mundo onde os objetos físicos se integrarão sem problemas na rede de informações, e onde estes objetos podem se tornar participantes ativos em processos. Os serviços estarão disponíveis para se integrar com estes dispositivos inteligentes pela internet, podendo requisitar informações associadas a eles, e alterar seus estados, levando em conta a segurança e privacidade (ZHOU, 2012).

Já o EPoSS (apud ZHOU, 2012) conceitua IoT como uma rede composta por "coisas" ou "objetos" com identidade, personalidades virtuais operando em espaços inteligentes e se comunicando com usuários e com o ambiente (ZHOU, 2012).

E o CERP-IoT (apud ZHOU, 2012) define a internet das coisas como uma parte integrada da internet do futuro, que pode ser definida com uma infraestrutura de rede global

e com capacidade de auto-configuração baseada em padrões e protocolos de comunicação interoperáveis onde as "coisas" físicas e virtuais tenham identidades, atributos físicos (ZHOU, 2012).

Segundo McEwen e Cassimally (2013) a ideia de IoT é ter menos dispositivos com grande poder computacional como computadores portáteis, celulares e *tablets* e possuir uma quantidade maior de dispositivos de uso específico com poder computacional reduzido como os dispositivos *wearables*.

É importante também notar a diferença entre Internet das Coisas e a computação ubíqua, como demonstrado em McEwen e Cassimally (2013), onde é utilizado como exemplo de computação ubíqua um odorizador de ambientes que dispersa perfume quando acionado pelo sensor de presença. É um dispositivo inteligente de aplicação específica, porém não está conectado à Internet e também não possui um identificador único, logo não se encaixa nos conceitos de IoT, mas sim em computação ubíqua. Entretanto, o fato de simplesmente colocar um identificador único e conectar um objeto qualquer a rede mundial de computadores não é tornar este objeto um dispositivo de IoT, é necessário que exista um fluxo de informações relevantes, que representem as características deste objeto (MCEWEN; CASSIMALLY, 2013).

Diante destas definições de Internet das Coisas fica evidente alguns aspectos em comum, como a rede ser composta por dispositivos que se comuniquem através de protocolos de comunicação interoperantes, e o fato de cada dispositivo ter sua identificação única.

2.1.2 Exemplos de aplicações

A Internet das Coisas possui uma vasta gama de aplicações nas mais diversas áreas, comumente utilizada em dispositivos de aplicação extremamente específica que operam baseados em informações externas, ou que fornecem as informações para um sistema externo.

Um exemplo de dispositivo de IoT que atua baseado em informações para melhora do conforto do usuário final é o despertador inteligente, que ajusta o horário do alarme baseado nos horários do transporte público que o usuário utiliza. Assim como os *displays* disponíveis em paradas de ônibus e estações de trem informam o horário de chegada do transporte baseado em informações provenientes do sistema de GPS (*Global Positioning System*) (MCEWEN; CASSIMALLY, 2013).

No campo da saúde podem ser citados exemplos como o GlowCap ®, um dispositivo que lembra o usuário de tomar suas medicações no horário correto, e envia um email para o médico comunicando que o paciente está tomando a medicação ou não. Nesta área também se encaixam muitos dispositivos vestíveis, conhecidos pelo seu nome em língua inglesa "*wearable devices*", que são dispositivos, na grande maioria das vezes sensores, que são acoplados ao corpo ou nas roupas e calçados, como o Nike Plus ® que é acoplado aos tênis esportivos, tornando possível o monitoramento da atividade física do usuário (MCEWEN; CASSIMALLY, 2013).

Uma grande vantagem dos dispositivos de uso específico é a facilidade de utilização, como o *display* com os horários do transporte público, mesmo que possa existir um aplicativo para telefone que forneça esta informação, ela fica mais acessível em um *display* pois o usuário não precisa procurar o aplicativo no seu telefone. A comunicação do dispositivo com a Internet é de grande importância, como no caso dos dispositivos *wearable* que possuem restrições de tamanho e peso, podem realizar o armazenamento da informação em um servidor externo, e as informações podem ser visualizadas em telas de celulares, computadores e até mesmo televisores (MCEWEN; CASSIMALLY, 2013).

Como a classificação dos dispositivos quanto a Internet das Coisas é muito ampla, é necessário dividir o conceito em partes menores e bem definidas, como é apresentado na seção a seguir.

2.1.3 Segmentação

A internet das coisas ainda é um conceito relativamente novo e ainda estão sendo definidos conceitos, sobre o que ela realmente é. Juntamente com estes conceitos está sendo criada uma segmentação em sub-áreas para uma melhor definição.

A principal segmentação é definida por Teich (2013) onde pode-se dividir a internet das coisas em dois grupos o IIoT (*Industrial Internet of Things*), que é a IoT aplicada a indústria, e o HIIoT (*Human Internet of Things*) que é a internet das coisas para humanos. Para esta segmentação foi realizada uma análise comportamental dos dispositivos, como é apresentado na Figura 1. Esta análise divide as aplicações de IoT em dois eixos principais, o de controle e o de impacto, tendo o eixo de controle a experiência em um de seus extremos, e a segurança e saúde no outro. O eixo de controle tem a iteratividade em um dos seu extremos e autonomia no extremo oposto. Com esta divisão podem ser percebidos quatro grupos bem definidos que são, monitoramento, operações, obtenção e residencial. O grupo de monitoramento juntamente com operações formam a IIoT, caracterizada pelo controle autônomo. Já o grupo de obtenção e residencial forma o HIIoT, que é mais interativo e menos autônomo (TEICH, 2013).

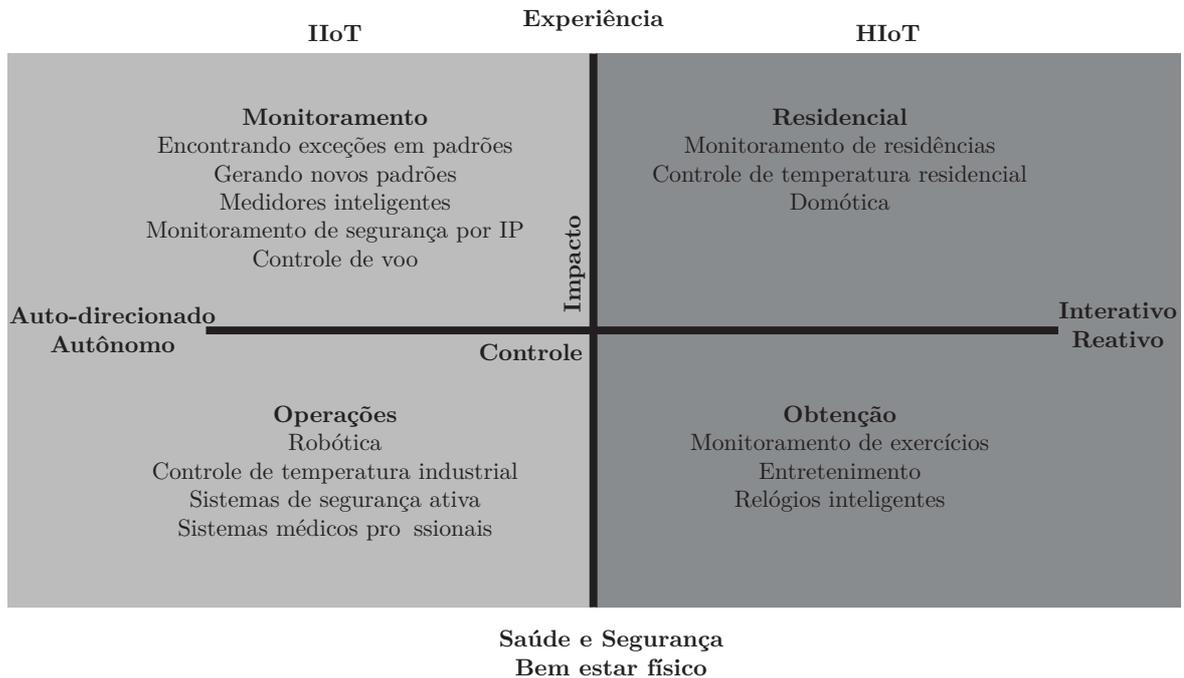
2.1.4 Principais Protocolos

Para a implementação da internet das coisas foram desenvolvidos alguns protocolos específicos como MQTT (*Message Queuing Telemetry Transport*) e CoAP (*Constrained Application Protocol*), assim outros protocolos já existentes foram explorados de maneira diferente como é o caso do HTTP que serviu de base para o estilo arquitetural REST (*Representational State Transfer*) (JAFFEY, 2014; RICHARDSON; RUBY, 2008).

2.1.4.1 REST

REST é um estilo arquitetural que utiliza recursos do protocolo HTTP para trabalhar com a representação de estados de recursos. Neste estilo arquitetural os recursos são mapeados e

Figura 1 – Segmentação da Internet das Coisas.



Fonte – Adaptado de Teich (2013).

através de seu endereço é possível obter o estado ou realizar alterações de contexto utilizando requisições HTTP como GET, PUT, UPDATE e DELETE (RICHARDSON; AMUNDSEN; RUBY, 2013).

Atualmente diversos serviços *online* possuem uma API (*Application Programming Interface*) com a qual o usuário, ou o desenvolvedor aplicações que utilizem o serviço, possam interagir com o serviço. Grande parte destas API's pode ser chamadas de "*RESTful*", ou seja, suportam o estilo REST (ALLAMARAJU, 2010).

O REST pode operar também sem o suporte do HTTP, como demonstra Richardson, Amundsen e Ruby (2013), para situações onde não existem recursos de *hardware* suficientes para implementar o HTTP como em sistemas embarcados. Esta versão é chamada de CoAP (RICHARDSON; AMUNDSEN; RUBY, 2013).

2.1.4.2 CoAP

CoAP é um protocolo de transferência de documentos como o HTTP, porém foi projetado para ser utilizado em dispositivos com recursos limitados, como sistemas embarcados e aplicações de Internet das Coisas. A similaridade do CoAP com o HTTP e REST dá-se pela utilização das requisições como GET, PUT, POST e DELETE. E ele foi projetado para operar em conjunto com o HTTP (JAFHEY, 2014).

Algumas das principais diferenças do HTTP para o COAP é que este opera com datagramas sem conexão, logo opera com UDP ao invés de TCP. O que diminui bastante o tamanho

dos *frames* enviados, possibilitando a implementação em dispositivos com poucos recursos de *hardware* (JAFFEY, 2014).

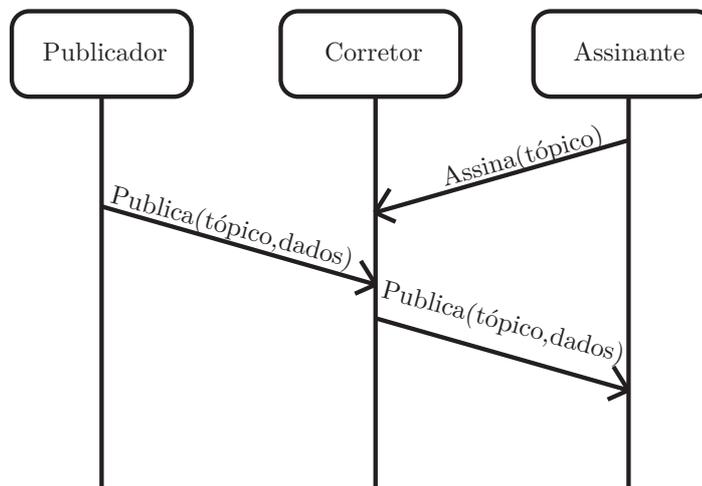
Sensores que utilizam CoAP são vistos como servidores, e devem aceitar requisições externas dos clientes, o que pode ser problemático em algumas redes que operam utilizando NAT (*Network Address Translation*) que realizam a conversão dos endereços, tornando necessária uma configuração avançada da rede para o redirecionamento de portas. (JAFFEY, 2014; TANENBAUM, 2003).

2.1.4.3 MQTT

O MQTT é um protocolo de transferência de mensagens, binário, simples e leve. Ele utiliza o modelo publicador/assinante muito conhecido pelo seu nome em língua inglesa *publisher/subscriber* e a sua abreviação *pub/sub*, estas assinaturas são gerenciadas pelo corretor, mais conhecido por *broker* (MESNIL, 2014).

Este protocolo trabalha com tópicos onde os clientes se inscrevem para receber as mensagens publicadas nestes tópicos pelos publicadores. Este processo está ilustrado na Figura 2 onde o assinante realiza a assinatura de um tópico, e quando o corretor recebe uma publicação do publicador, repassa esta aos assinantes que tenham realizado a assinatura do tópico da mensagem. (LAMPKIN et al., 2012).

Figura 2 – Funcionamento do modelo publicador/assinante.



Fonte – Elaborado pelo autor.

Estes tópicos podem ser chamados de áreas de interesse, e são *strings* codificadas em UTF-8 divididas em níveis, que são separados por uma barra, caractere "/", que é denominado como separador de nível de tópico. Esta estrutura fica mais clara com o exemplo apresentado na Figura 3 onde é definida uma estrutura hierárquica para representação de um endereço em seis níveis, onde o número é o nível mais baixo da hierarquia e o país é o mais alto (LAMPKIN et al., 2012; HIVEMQ, 2015).

Figura 3 – Tópicos MQTT.

Estrutura País/Estado/Cidade/Bairro/Rua/Número
Nível 1 Nível 2 Nível 3 Nível 4 Nível 5 Nível 6

[A] Brasil/RS/sao_leopoldo/cristo_rei/av_unisinos/500
 [B] Brasil/RS/sao_leopoldo/cristo_rei/av_unisinos/800
 [C] Brasil/RS/sao_leopoldo/centro/av_joao_correa/500

Fonte – Elaborado pelo autor.

Para dar mais flexibilidade à estrutura de tópicos, são utilizados caracteres curingas, chamados de *wildcards*, que são apresentados na Tabela 1. O caractere "+" tem a função de generalizar um nível dos tópicos e é utilizado em algum nível intermediário da estrutura. Já o caractere "#" é necessário quando deseja-se generalizar múltiplos níveis da estrutura, logo é utilizado no final da estrutura. O sistema reserva o caractere "\$" para uso interno do *broker*, tornando a estrutura disponível somente para leitura (HIVEMQ, 2015).

Tabela 1 – Caracteres curinga.

Símbolo	Nível	Descrição
+	um	Generaliza um nível intermediário
#	múltiplos	Generaliza diversos níveis
\$	um	Reservado para funções internas do <i>broker</i>

Fonte – Elaborada pelo autor.

Um exemplo de utilização do caractere curinga "+" nas estruturas apresentadas na Figura 3 seria assinar o tópico "Brasil/RS/sao_leopoldo/+/+/500" que assinaria os tópicos A e C pois o bairro e a rua foram generalizados, mas o número ficou especificado em 500. Para selecionar todos os endereços de um determinado bairro pode-se utilizar o *wildcard* "#", que ao assinar o tópico "Brasil/RS/sao_leopoldo/cristo_rei/#" assinaria todos os sub-tópicos abaixo do nível 4 que atendam os níveis mais altos da hierarquia, neste caso assinaria os tópicos A e B. Já para o caractere "\$" pode-se usar de exemplo o tópico "\$SYS/broker/clients/connected" que é reservado pelo *broker* e informa a quantidade de clientes conectados ao *broker*, sendo um tópico que permite somente leitura (HIVEMQ, 2015).

A entrega das mensagens do protocolo MQTT acontecem conforme o seu QoS (*Quality of Service*), em português qualidade de serviço. São definidos três níveis de QoS na especificação do protocolo, numerados de 0 a 2, e cada um possui uma função distinta como é apresentado na Tabela 2 (OASIS Open, 2014).

Dos níveis de QoS, o nível 0 é o mais baixo e simples, ele basicamente representa disparar a mensagem e esquecer, não sendo necessária a confirmação de recebimento pelo *broker*. Já o nível 1 garante que a mensagem é entregue pelo menos uma vez, mas pode ser entregue

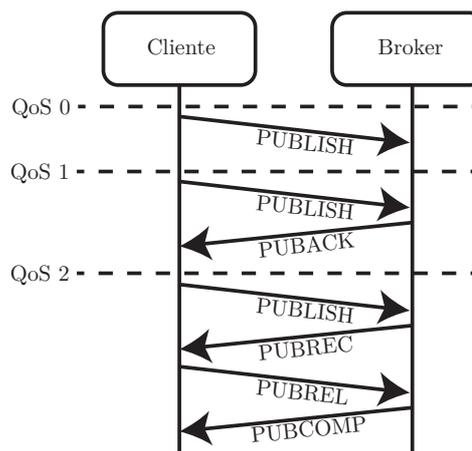
mais de uma vez caso ocorra algum atraso na transmissão. E o nível 2 garante que a mensagem seja entregue apenas uma vez, evitando que mensagens repetidas sejam enviadas para o *broker*. Os níveis 1 e 2 utilizam comandos de confirmação de resposta como PUBACK que contem o identificador da mensagem e confirma que foi entregue, são utilizados também os comandos PUBREC, PUBREL e PUBCOMP que também exercem a função de identificar os pacotes. A etapa de publicação para cada nível de QoS é apresentada na Figura 4.

Tabela 2 – Níveis de QoS do MQTT.

QoS	Significado
0	Disparar e esquecer
1	Pelo menos uma vez
2	Exatamente uma vez

Fonte – Elaborado pelo autor.

Figura 4 – Níveis de QoS do MQTT.



Fonte – Elaborado pelo autor.

A segurança, que é uma questão chave da Internet das Coisas segundo Zhang et al. (2014) é implementada no protocolo MQTT utilizando autenticação para a conexão e TLS (*Transport Layer Security*).

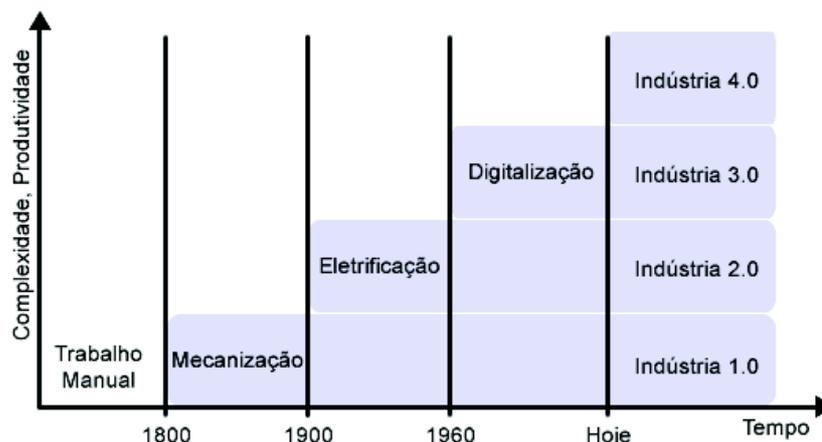
Com a Internet das Coisas, é possível obter informações sobre os ativos de uma empresa, além de outros dados que possam impactar no processo produtivo. Esta facilidade na obtenção dos dados, e aumento do volume de informações possibilita uma nova revolução industrial como é apresentado na próxima seção.

2.2 Indústria 4.0

O mundo já presenciou três grandes mudanças na indústria, chamadas de revoluções industriais, sendo a primeira no reino unido no final do século 18 com a mudança drástica de

uma economia agrária com a inserção de métodos de produção mecânicos. A segunda revolução ocorreu no início de século 20 com o nascimento das fábricas. A última revolução ocorreu na década de 1960 com o uso dos eletrônicos e tecnologia da informação, iniciando a automação industrial. Este histórico está representado na Figura 5 (MACDOUGALL, 2014).

Figura 5 – Histórico revolução industrial.



Fonte – Adaptado de Drath e Horch (2014).

Um grande diferencial da quarta revolução industrial é o fato dela ser prevista e projetada, ao contrário das outras que foram observadas depois de implementadas. Isso permite que as empresas se preparem e contribuam para o seu desenvolvimento (Hermann, Mario; Pentek, Tobias; Otto, 2015).

O conceito de indústria 4.0 muda o conceito de produção centralizada para produção descentralizada, e está fortemente ligado a sistemas embarcados e sistemas ciberfísicos. Estes sistemas ciberfísicos são as tecnologias que permitem interligar o mundo virtual com o mundo físico, como por exemplo a Internet das Coisas (MACDOUGALL, 2014).

A indústria 4.0 também está ligada ao conceito de fábricas inteligentes, onde os sistemas de produção, também chamados de PCS (*Process Control Systems*), estão ligados diretamente à sistemas de ERP (*Enterprise Resource Planning*) e MES (*Manufacturing Execution Systems*) (MACDOUGALL, 2014).

Atualmente os dados gerados pelas máquinas nas indústrias representam 15% dos dados totais da empresa, Greengard (2015) prevê um crescimento para 50% dos dados totais nos próximos 10 anos com o advento da Internet das Coisas. Estas informações provenientes de sensores inteligentes permitem conhecer a localização e estado de ativos em tempo real, tornando assim a tomada de decisões baseadas em sensores possível, uma vez que estes dados estejam ligados com o sistema de controle e automação (GREENGARD, 2015).

2.3 Redes de campo — *Fieldbus*

As redes de campo, muito conhecidas pelo seu nome em inglês *Fieldbus*, são conhecidas pelo seu grande impacto na história da automação industrial. A norma internacional IEC 61158, define conceitualmente uma rede de campo como um barramento para comunicação digital, serial e *multidrop* para dispositivos de instrumentação e controle industrial, como atuadores, transdutores e controladores, mas não limitados a estes (IEC - *International Electrotechnical Commission*, 2003; ZURAWSKI, 2005).

Esta definição atualmente não chega a ser completamente clara, segundo (ZURAWSKI, 2005), devido a grande evolução histórica das redes de campo, pois estas são utilizadas atualmente em todos os domínios da automação, indo de processos industriais em fábricas até a domótica e aviônica. Entretanto todas as redes de campo devem possuir algumas características em comum (ZURAWSKI, 2005):

- Flexibilidade e modularidade - Uma rede de campo deve poder ser expandida de maneira mais simples do que um sistema centralizado;
- Configurabilidade - As redes de campo devem permitir a parametrização dos dispositivos de campo;
- Manutenibilidade - Deve ser permitido o monitoramento, atualização e manutenção dos dispositivos de campo através da rede;
- Distribuição - As redes de campo devem permitir o processamento distribuído das informações.

Dentre os diversos tipos de redes de campo, destaca-se a rede Modbus que é a mais utilizada atualmente pelo fato de ser simples, de baixo custo e poder utilizar diversos meios físicos. Estas características fazem com que, apesar de o protocolo já possuir mais de 30 anos, os fabricantes continuem dando suporte em novos produtos. Dada a importância deste protocolo, ele é abordado mais detalhadamente na seção 2.3.1 (MCCONAHAV, 2011; ZURAWSKI, 2005).

2.3.1 Modbus

O Modbus é um protocolo de comunicação industrial, mais especificamente uma estrutura de envio de mensagens, que utiliza a arquitetura mestre/escravo e permite a comunicação entre dispositivos conectados em diferentes tipos de barramentos e redes, desde linhas seriais (EIA 232 e EIA 485), até redes TCP/IP (REYNDERS; MACKAY; WRIGHT, 2004; ZURAWSKI, 2005).

São implementadas pelo Modbus as camadas 7 e 2 (aplicação e enlace) do modelo OSI, regulamentadas pelas normas EN 1434-3 e IEC 870-5 respectivamente. Quanto a camada 1 (física) não há especificação (TANENBAUM, 2003; STEMMER, 2010).

O protocolo possui dois modos de operação, ASCII (*American Standard Code for Information Interchange*) e RTU (*Remote Terminal Unit*). Existem também derivações do Modbus como o Modbus Plus, que utiliza a passagem de ficha (*token passing*) para aumentar a velocidade, e o Modbus TCP/IP que utiliza o protocolo TCP/IP e *Ethernet* para transportar os dados (STEMMER, 2010).

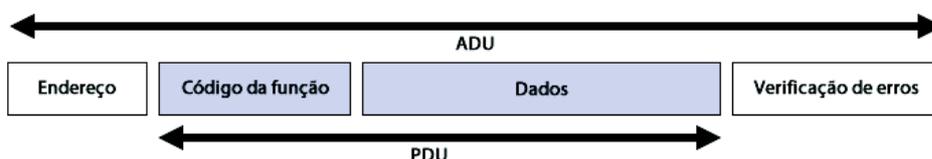
O modo de operação ASCII envia cada *byte* da mensagem como dois caracteres ASCII, utilizado quando deseja-se um tempo entre caracteres de até 1 segundo, o que torna a transmissão assíncrona. Os seus *bytes* são divididos em 1 *bit* de inicialização (*start bit*), 7 *bits* de dados (*data bits*), 1 *bit* de paridade que é opcional, e 1 *bit* de parada (*stop bit* caso for utilizada paridade ou 2 *bits* caso não for utilizada paridade. Para a verificação de erros, este modo utiliza a técnica LRC (*Longitudinal Redundancy Check*) (STEMMER, 2010).

Já no modo RTU cada *byte* da mensagem é enviado como dois caracteres hexadecimais de 4 *bits*, esta alta densidade de caracteres aumenta o fluxo de dados (*data throughput*) quando comparado com o modo ASCII, isso exige uma transmissão síncrona (fluxo de dados contínuo). Este modo utiliza o sistema de codificação binária de 8 *bits* hexadecimal, a codificação dos seus *bytes* difere-se do modo ASCII nos *bits* de dados pois utiliza 8 ao invés de 7, e no modo de verificação de erro utilizado é o CRC (*Cyclical Redundancy Check*). Neste trabalho será abordado mais aprofundadamente o modo RTU na próxima seção (STEMMER, 2010).

2.3.1.1 Modbus-RTU

Para entender o protocolo Modbus é importante entender o funcionamento do *frame*, que é apresentado na Figura 6, onde o ADU (*Application Data Unit*) representa todo o *frame* e o PDU (*Protocol Data Unit*) somente a parte dos dados. Basicamente ADU contém, o endereço do dispositivo escravo, os dados que devem ser entregues (PDU) e a verificação de erros para detectar falhas de integridade nos dados (STEMMER, 2010; Modbus Organization, 2012).

Figura 6 – *Frame* Modbus



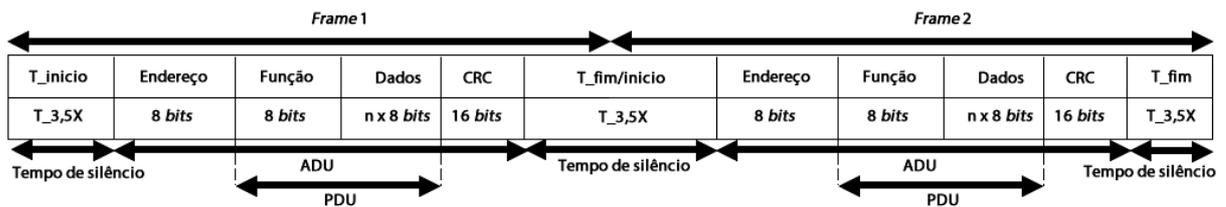
Fonte – Adaptado de Modbus Organization (2012).

O tamanho do PDU é limitado pelo tamanho máximo herdado da implementação em rede serial que é de 256 *bytes*, subtraindo 1 *byte* do endereço e 2 *bytes* da verificação de erro, o PDU fica com um tamanho máximo de 253 *bytes* (Modbus Organization, 2012).

A delimitação dos *frames* no modo RTU é dada por um tempo sem transmissão de dados, definido como 3,5 vezes o tempo de transmissão de um caractere (palavra de dados - 11

bits). Por exemplo, para uma comunicação com taxa de transmissão de 9600 kbps o tempo de transmissão de um caractere é de 1,15 ms, então o tempo de espera é de 4,025 ms. Este intervalo de silêncio deve ocorrer no início e no término da transmissão de cada *frame* como é apresentado na Figura 7 (STEMMER, 2010).

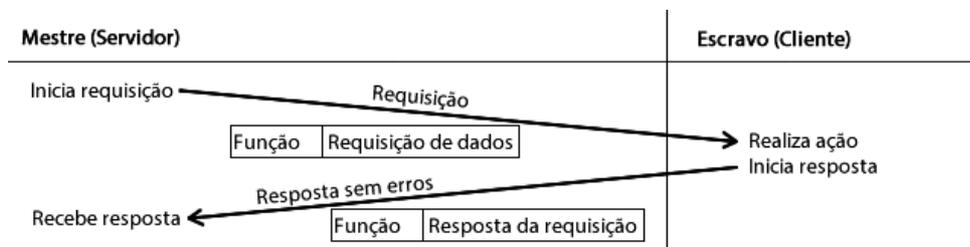
Figura 7 – Tempo entre *frames* Modbus.



Fonte – Elaborado pelo autor.

As requisições sempre partem do mestre, também chamado de servidor, e são processadas pelos escravos, conhecidos também como clientes. Um escravo só pode escrever no barramento se o mestre solicitou alguma informação deste escravo, determinada pelo campo do endereço do ADU. Uma vez que o escravo confirme que a mensagem é para ele, este executa o comando solicitado e responde a mensagem com o código da função e os dados da requisição como é apresentado na Figura 8 (STEMMER, 2010).

Figura 8 – Tansação Modbus sem erros.



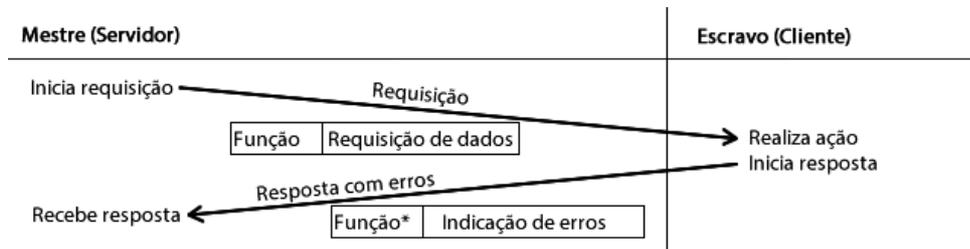
Fonte – Adaptado de Modbus Organization (2012).

Caso ocorra algum erro na requisição o cliente responde para o mestre que ocorreu um erro, indicando o código da função modificado e outras informações sobre os erros no campo de dados, como pode ser visto na Figura 9. Este código da função modificado possui o *bit* mais significativo em nível lógico 1, isto é reconhecido como indicação de erro (STEMMER, 2010).

Um exemplo disto pode ser visto na Figura 10 onde o código de função original é 00000011 binário, ou 0x03 hexadecimal, representado por "Função" nas figuras 8 e 9 que em caso de erro retorna 0x83 hexadecimal ou 10000011 binário que é representado por "Função*" na Figura 9 (STEMMER, 2010).

Os códigos de função assumem valores de 1 a 127 decimal, e podem ser separados em dois grupos, o de funções públicas e o de funções definidas pelo usuário. Estes grupos são separados em 5 blocos como é apresentado na Figura 11, onde os códigos de função contidos

Figura 9 – Tansação Modbus com erros.



Fonte – Adaptado de Modbus Organization (2012).

Figura 10 – Retorno do código de função.

Binário	Hexadecimal
0000 0011	0x03
➔ 1000 0011	0x83

Fonte – Elaborado pelo autor.

entre 1 a 64, 73 a 99 e 111 a 127 são reservados para funções públicas já definidas, enquanto os códigos de função contidos entre 65 a 72 e 100 a 110 podem ser definidos pelo usuário. Os principais códigos de funções são apresentados no Quadro 1 (Modbus Organization, 2012; STEMMER, 2010).

Figura 11 – Faixa de grupo de funções.

111..127	Funções públicas
100..110	Funções definidas pelo usuário
73..99	Funções públicas
65..72	Funções definidas pelo usuário
1..64	Funções públicas

Fonte – Adaptado de Modbus Organization (2012).

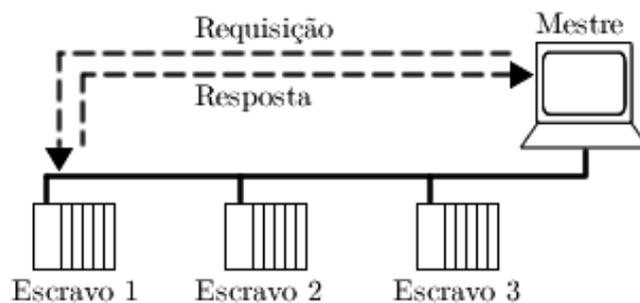
Utilizando linhas seriais existem duas opções de envio de mensagem, a *unicast* onde o mestre realiza uma requisição para algum escravo e somente este retorna uma resposta para o mestre enquanto os outros ignoram a mensagem, como está representado na Figura 12. E o modo *broadcast* onde o mestre envia uma requisição para o endereço 0 e todos os escravos recebem a mensagem e não retornam nenhum tipo de resposta. O *broadcast* só pode ser utilizado para funções de escrita e todos os escravos devem aceitar as requisições (Modbus Organization, 2006b).

Quadro 1 – Códigos de funções Modbus

		Código de funções				
		Código	Sub código	Hex		
Acesso de dados	Acesso ao bit	Entradas Discretas Físicas	Read Discrete Inputs	02		01
		Bobinas físicas ou bits internos	Read Coils	01		01
			Write Single Coil	05		05
			Write Multiple Coils	15		0F
	Acesso a 16 Bits	Registros de entrada físicos	Read Input Register	04		04
		Registros de saída física	Read Holding Register	03		03
			Write Single Register	06		06
			Write Multiple Registers	16		10
			Read/Write Multiple Registers	23		17
			Mask Write Register	22		16
			Read FIFO queue	24		18
	Acesso a arquivos	Read File record	20		14	
		Write File record	21		15	
	Diagnóstico	Read Exception status	07		07	
Diagnostic		08	00-18,20	08		
Get Com event Counter		11		0B		
Get Com Event Log		12		0C		
Report Server ID		17		11		
Read device identification		43	14	2B		
Outro	Encapsulated interface Transport	43	13,14	2B		

Fonte – Elaborado pelo autor.

Figura 12 – Modo *unicast*

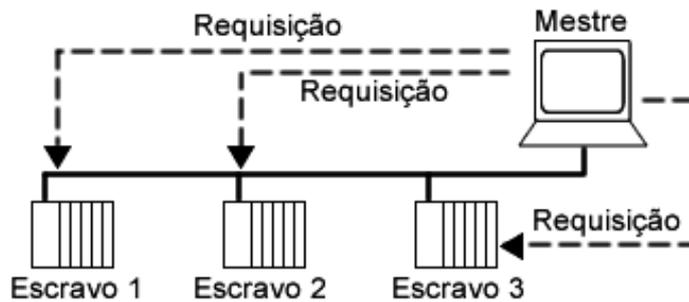


Fonte – Adaptado de Modbus Organization (2006b).

2.3.1.2 Modbus-TCP/IP

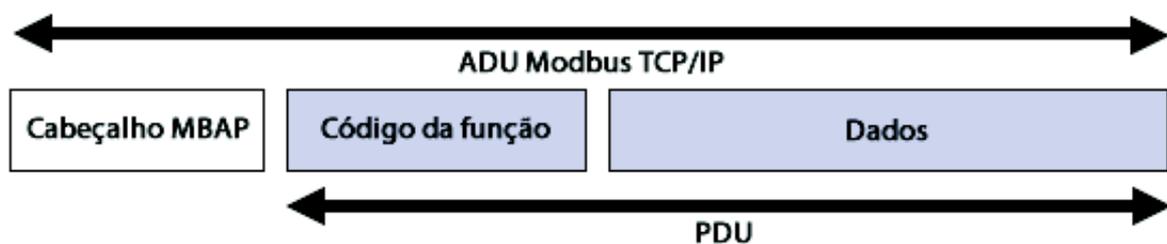
As mensagens Modbus podem ser transmitidas utilizando uma rede TCP/IP, para isto o *frame* Modbus é modificado conforme apresentado na Figura 14 onde existe alteração no ADU comparado com o *frame* do Modbus serial, porém o PDU continua o mesmo (MCCONAHAV, 2011; Modbus Organization, 2006a).

Figura 13 – Modo *broadcast*



Fonte – Adaptado de Modbus Organization (2006b).

Figura 14 – *Frame* Modbus TCP/IP.



Fonte – Adaptado de Modbus Organization (2006a).

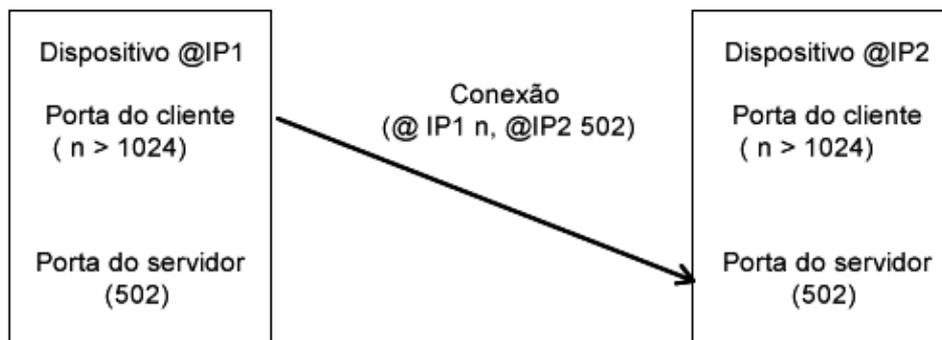
É adicionado ao frame o cabeçalho MBAP (*Modbus Application Protocol header*) que é utilizado para identificar o ADU do Modbus na rede TCP/IP. A verificação de erros por CRC é removida pois esta é implementada pela rede ethernet (Modbus Organization, 2006a).

O cabeçalho MBAP possui 7 bytes divididos em 4 campos: O de identificação de transação com 2 bytes responsável por identificar as transações, o identificador de protocolo com 2 bytes responsável por identificar o protocolo Modbus, o de comprimento também com 2 bytes que informa a quantidade de bytes restantes na mensagem, e o de identificação da unidade com 1 byte que é basicamente o endereço do escravo conectado na rede serial (Modbus Organization, 2006a).

Por ser uma comunicação por TCP é necessário estabelecer conexão entre os dispositivos através de sockets. Os servidores do Modbus TCP/IP utilizam a porta 502 para receber novas conexões, já os clientes devem utilizar portas diferentes e maiores do que 1024. Esta conexão está representada na Figura 15 (Modbus Organization, 2006a).

As redes de campo podem ser exploradas além do ambiente industrial, como por exemplo no sistema de transmissão e distribuição de energia elétrica, possibilitando o desenvolvimento de redes elétricas inteligentes, como é apresentado na seção a seguir.

Figura 15 – Conexão TCP/IP.



Fonte – Adaptado de Modbus Organization (2006a)

2.4 Geração distribuída e redes inteligentes

As redes de energia elétrica tradicionais contam somente com geração, transmissão, distribuição e consumo, com um fluxo unidirecional de energia da geração para o consumidor final como é apresentado na Figura 16a. Na Figura 16b pode ser visto o conceito das redes elétricas inteligentes, também conhecidas pelo seu nome em língua inglesa "*smart grid*", onde existem unidades de geração distribuída que utilizam energias renováveis como solar fotovoltaica e eólica. Neste caso existe o fluxo bidirecional da energia e informação (LOPES; FERNANDES; CHRISTINA, 2015).

A geração distribuída é um dos temas mais discutidos em relação a planejamento energético no mundo e apontada como o futuro da produção de energia elétrica segundo Freitas B. M. R. de; Hollanda (2015). Para INEE (2015) a geração distribuída inclui equipamentos para medição, controle, comando e fontes de geração de pequeno porte como geradores de emergência, co-geradores, painéis fotovoltaicos e pequenas centrais hidrelétricas. Uma das grandes vantagens da geração distribuída é a economia em investimentos em transmissão e redução nas perdas destes sistemas (INEE, 2015; FREITAS B. M. R. DE; HOLLANDA, 2015).

No Brasil a micro e minigeração distribuída são passíveis de implementação desde que estejam de acordo com a Resolução Normativa nº 482/2012 da ANEEL (Agência Nacional de Energia Elétrica) e do PRODIST (Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional) A micro e minigeração são diferenciadas pela potência, sendo a microgeração com potência instalada menor ou igual a 75 quilowatts(kW) e a minigeração com até 3 megawatts(MW) (ANEEL, 2014).

Uma das fontes de energia renovável que recebe grande destaque no Brasil, é a energia solar fotovoltaica, pois apesar do mercado brasileiro ainda não fabricar as células fotovoltaicas, realizando somente a montagem de módulos, possui a maior reserva mundial de quartzo de silício, matéria-prima para fabricação das células fotovoltaicas, em seu território (FREITAS B. M. R. DE; HOLLANDA, 2015).

Figura 16 – Comparação das Redes Elétricas Tradicionais com as Redes Elétricas Inteligentes.



(a) Redes Elétricas Tradicionais.



(b) Redes Elétricas Inteligentes.

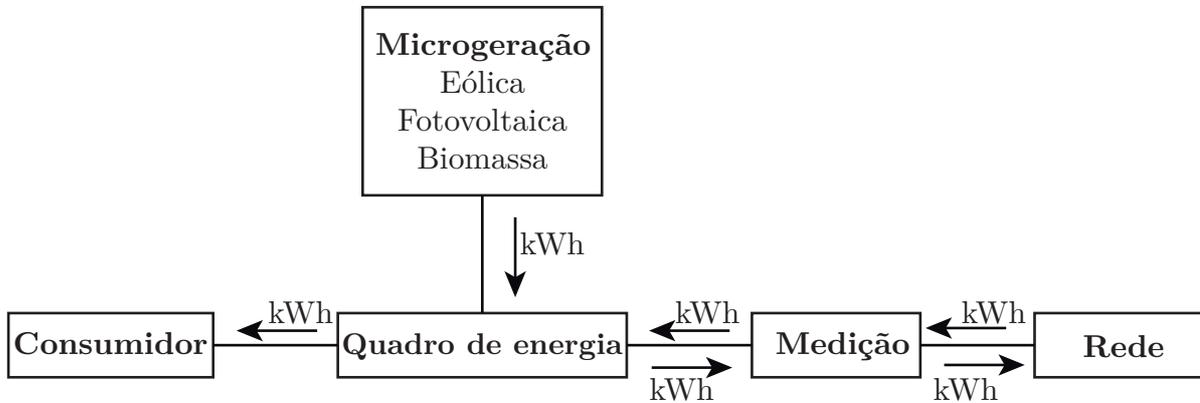
Fonte – Retirado de (LOPES; FERNANDES; CHRISTINA, 2015)

Destacam-se dois tipos de tarifação para a energia proveniente de mini e microgeração, sendo estes o FiT (*Feed-in-Tariff*) e o NEM (*Net Energy Metering*). No sistema FiT são estabelecidos contratos de longo prazo entre o cliente e a concessionária onde é estabelecido um valor para a energia injetada na rede, que normalmente é maior do que o da energia consumida, o que incentiva a geração distribuída. Já para os sistemas NEM trabalham com créditos de energia, logo não ocorre a comercialização da energia por parte do consumidor, este sistema foi o escolhido para o Brasil (FREITAS B. M. R. DE; HOLLANDA, 2015).

A medição utilizada pelo sistema NEM, também chamado de sistema de compensação de energia elétrica, é apresentada na Figura 17. A energia gerada pela microgeração atende o consumidor, caso a energia gerada exceda a energia demandada pela carga, a diferença entre a energia gerada e consumida é injetada na rede da concessionária gerando assim os créditos de energia. Caso a energia gerada não seja suficiente para atender a carga, a unidade consumidora absorve a energia da rede da concessionária. A fatura do cliente realiza um balanço energético, cobrando a diferença entre a energia consumida, os créditos de energia do mês anterior, e a

energia injetada na rede (ANEEL, 2014).

Figura 17 – Sistema de medição para NEM.



Fonte – Elaborado pelo autor.

O crédito de energia proveniente da geração distribuída pode ser compartilhado por unidades consumidoras diferentes, desde que estas sejam atendidas pela mesma concessionária e o titular esteja cadastrado sob o mesmo CPF (Cadastro de Pessoa Física) ou CNPJ (Cadastro de Pessoa Jurídica). Sendo que a prioridade do consumo dos créditos é para o local de geração dos mesmos (ANEEL, 2012).

Este compartilhamento de créditos de energia permite caracterizar uma VPP (*Virtual Power Plant* - Usina Virtual) que é um sistema heterogêneo composto de múltiplas fontes geradoras de energia, localizados em locais diferentes gerenciadas por um EMS (*Energy Management System*) que pode ser implementado com um sistema SCADA (*Supervisory Control and Data Acquisition*). Para que isto funcione é crucial que exista comunicação entre as plantas e o EMS (SABOORI; MOHAMMADI; TAGHE, 2011).

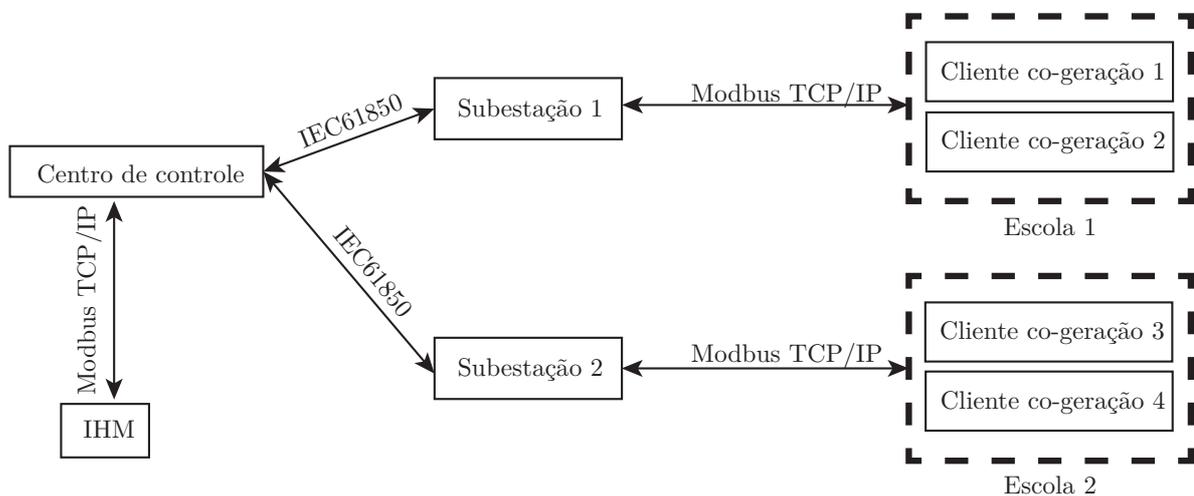
Vistos estes conceitos básicos é possível compreender com maior clareza os demais capítulos deste trabalho, começando pelos trabalhos relacionados apresentados no próximo capítulo.

3 TRABALHOS RELACIONADOS

Neste capítulo são apresentados os trabalhos que se aproximam do sistema proposto neste trabalho, os quais encontram-se em ordem de similaridade em relação ao estudo proposto.

É proposto por Temiz et al. (2015) um sistema de comunicação para o controle de micro usinas fotovoltaicas instaladas em duas escolas, utilizando os protocolos IEC61850 e Modbus TCP/IP como é apresentado na Figura 18. As escolas são alimentadas em média tensão, possuindo uma subestação transformadora abaixadora. Este sistema foi projetado para ser utilizado na Arábia Saudita onde não é permitido injetar energia na rede, logo o sistema tem como objetivo monitorar a geração fotovoltaica das escolas e desligar os inversores da rede caso estejam produzindo mais energia do que estão consumindo. A comunicação entre a IHM (Interface Homem Máquina) se comunica através do protocolo Modbus TCP/IP com o centro de controle, que utiliza o protocolo IEC61850 para se comunicar com as subestações, que por sua vez se comunicam com as unidades consumidoras através de Modbus TCP/IP novamente, uma vez que a maioria dos equipamentos de geração fotovoltaica suportam comunicação Modbus (TEMIZ et al., 2015).

Figura 18 – Sistema proposto por Temiz et al. (2015)



Fonte – Adaptado de Temiz et al. (2015).

Em Lampkin et al. (2012) é proposto como exemplo de aplicação do protocolo MQTT a criação de uma VPP onde a concessionária de energia possui comunicação com os clientes, recebendo dados sobre o consumo individual em tempo real, podendo assim prever o consumo ao longo do dia, e em alguns casos podendo controlar alguns dos equipamentos dos clientes para reduzir a carga do sistema quando necessário (LAMPKIN et al., 2012).

O desenvolvimento de *gateways* Modbus pode ser visto em Li, Zhang e Peng (2011),

onde foi proposto e implementado um *gateway* EPA (*Ethernet to Plant Automation*) para Modbus TCP/IP utilizando um microcontrolador da família ARM7. Para validação do *gateway* foram utilizados quatro sensores analógicos que se comunicam utilizando o protocolo EPA, e foi realizado o monitoramento por um computador que se comunicava utilizando o protocolo Modbus TCP/IP. A implementação de um *gateway* Modbus para fins industriais se demonstrou viável, além de poder ser utilizado em uma estrutura já existente, reduzindo o custo de novas instalações (LI; ZHANG; PENG, 2011).

Entretanto Gandhi e Adhyaru (2015) afirmam que a utilização do protocolo Modbus RTU através de meios de comunicação sem fio como Wi-Fi, Zigbee e GPRS pode ter efeitos negativos quanto a confiabilidade dos dados, pois os meios de transmissão sem fio enviam pacotes de dados menores do que o *frame*, e por possuírem diversos algoritmos de roteamento, podem entregar os pacotes em uma ordem diferente do que foram enviados. Para a demonstração destes efeitos negativos, os autores realizaram um estudo de caso baseado em redes elétricas inteligentes, onde utilizaram um computador executando o *software* Modbus Poll para emular o mestre da rede Modbus, *software* Modbus Slave para simular o escravo da rede, além do programa Hercules que foi utilizado para escrever dados hexadecimais na porta serial (GANDHI; ADHYARU, 2015).

O equipamento emulado por Gandhi e Adhyaru (2015) como escravo da rede foi um medidor de energia modelo EM1200 fabricado pela empresa Schneider Electric ®. Um medidor digital de energia, compacto e de fácil utilização com memória interna e com suporte ao protocolo Modbus através da interface EIA-485, onde o mapeamento dos endereços de uso específico do equipamento são apresentados no Quadro 2 (GANDHI; ADHYARU, 2015).

Nos testes realizados por Gandhi e Adhyaru (2015) inicialmente, foi solicitado o valor dos endereços 3005 e 3007 que são referentes a potência reativa na fase 1 e ao fator de potência da fase 1 respectivamente. A requisição pode ser vista na Figura 19, onde é feita a solicitação com o comando *Read Holding Registers* representada pelo byte "03", solicitando quatro valores a partir do endereço 3005. A resposta da requisição pode ser vista na linha 000418, que está representada de forma legível na Figura 20, onde identifica-se o a potência reativa na fase 1 como 5000 VAR e o fator de potência como 0,7.

Uma vez que as leituras estavam sendo feitas, Gandhi e Adhyaru (2015) simularam a troca de pacotes na resposta, como é apresentado na Figura 21 onde na linha 000424 está uma resposta com os pacotes trocados, forçados pelo autor. O mestre aceitou estes dados pois o autor modificou também os últimos dois *bytes* referentes a verificação de erro por CRC. Caso houvesse somente a troca dos pacotes, seria identificado o erro pela verificação do CRC, uma vez que a probabilidade da simples troca de pacotes por diferença de tempo de entrega devido aos algoritmos de roteamento resultar em um código de CRC adequado para a mensagem errada é muito pequena e quase improvável.

A velocidade de comunicação é um fator muito importante para a automação e para a Internet das Coisas, com o intuito de comparar os principais protocolos de IoT, Durkop, Czybik

Quadro 2 – Mapa de registradores do medidor de energia EM1200.

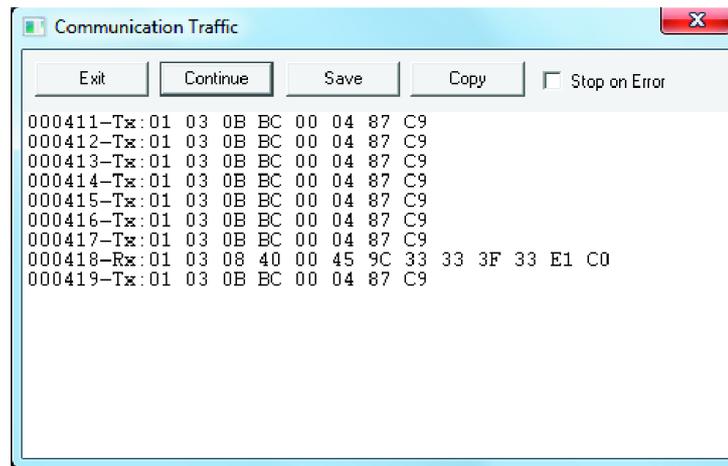
Parâmetro	Descrição	Endereço	Tipo de dados
Medição			
Medição - Potência			
W	Potência ativa, total	3051	Float *
W1	Potência ativa, fase 1	3003	Float *
W2	Potência ativa, fase 2	3019	Float *
W3	Potência ativa, fase 3	3035	Float *
VAR	Potência reativa, total	3053	Float *
VAR1	Potência reativa, fase 1	3005	Float *
VAR2	Potência reativa, fase 2	3021	Float *
VAR3	Potência reativa, fase 3	3037	Float *
VA	Potência aparente, total	3049	Float *
VA1	Potência aparente, fase 1	3001	Float *
VA2	Potência aparente, fase 2	3017	Float *
VA3	Potência aparente, fase 3	3033	Float *
Medição - Fator de potência			
PF	Fator de potência total	3055	Float *
PF1	Fator de potência, fase 1	3007	Float *
PF2	Fator de potência, fase 2	3023	Float *
PF3	Fator de potência, fase 3	3039	Float *
Energia			
FwdVAh	Energia aparente acumulada	3201	Float *
FwdWh	Energia ativa acumulada	3203	Float *
FwdVARh	Energia reativa acumulada	3205	Float *
FwdRun secs	Tempo de execução acumulado	3265	Unsigned long
FwdOn Sec	Tempo ligado	3267	Unsigned long
Intr	Numero de interrupções da tensão de entrada	3269	Unsigned long

Fonte – Adaptado de Gandhi e Adhyaru (2015)

e Jasperneite (2015) realizou testes em laboratório para verificar a performance dos principais protocolos de Internet das Coisas utilizando redes de celular. Em seus testes foram analisados os protocolos MQTT, CoAP e OPC-UA. Para estas análises foi utilizado o simulador de redes Anritsu MD8475A, capaz de simular redes com tecnologias 2G, 3G e 4G sem influências externas existentes nas redes reais (DURKOP; CZYBIK; JASPERNEITE, 2015).

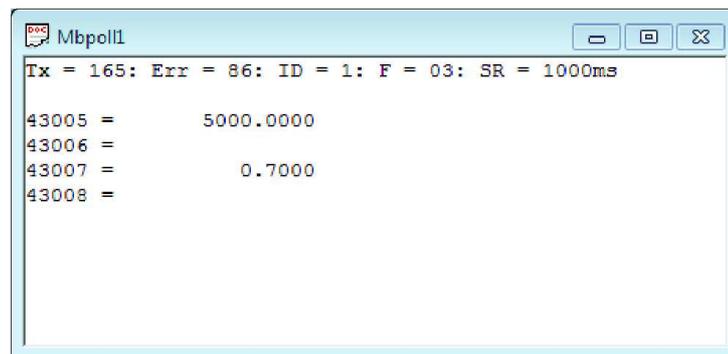
A metodologia empregada por Durkop, Czybik e Jasperneite (2015) para comparar a velocidade dos protocolos para a entrega contínua de mensagens, foi enviar dados com tamanho entre de 0 a 10000 *bytes* variando com um passo de 100 *bytes*, repetindo 100 vezes para cada passo. Em cada etapa o autor media o tempo entre o envio e confirmação de recebimento das mensagens. Os resultados podem ser vistos nas Figuras 22, 23 e 24. Foi observado que o protocolo CoAP possui uma performance inferior aos demais devido ao tamanho fixo dos pacotes a serem enviados. O MQTT obteve tempos similares ao OPC-UA, porém esta diferença é dada pelo tempo de conexão TCP uma vez que para cada pacote enviado foi realizada uma

Figura 19 – Requisição Modbus para leitura de registradores do EM1200.



Fonte – Retirado de Gandhi e Adhyaru (2015).

Figura 20 – Dados lidos do EM1200.



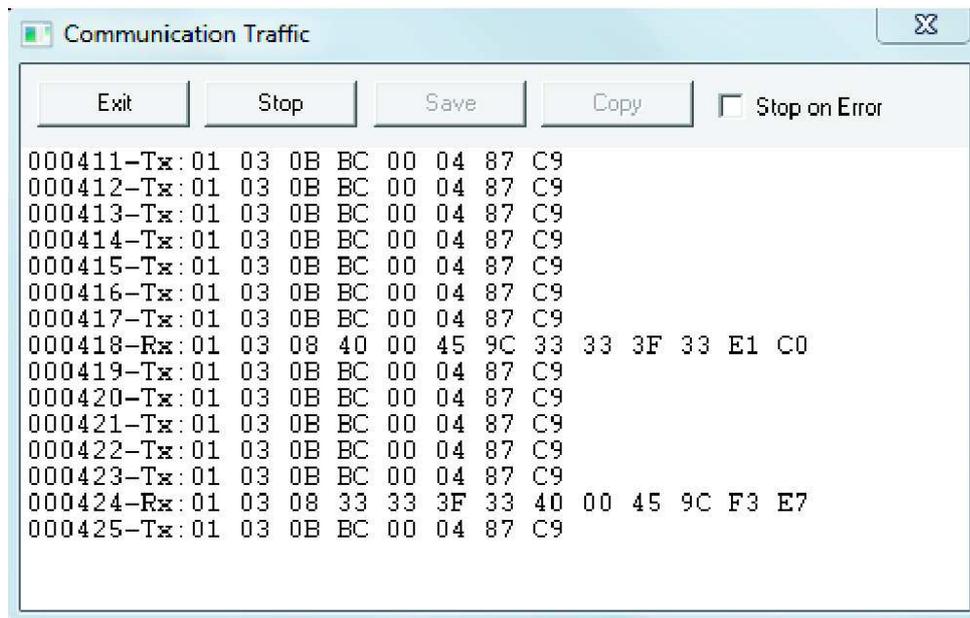
Fonte – Retirado de Gandhi e Adhyaru (2015).

nova conexão, o que não é uma boa prática segundo Lampkin et al. (2012). Com o melhor tempo ficou o protocolo OPC-UA devido a confirmação dos dados que é realizada somente ao final da entrega de todos os pacotes (DURKOP; CZYBIK; JASPERNEITE, 2015; LAMPKIN et al., 2012).

A comparação entre os protocolos CoAP e MQTT também pode ser vista em Thangavel et al. (2014) onde é proposto um *middleware* que se comunique por ambos protocolos, são realizados também um experimento prático e uma análise comparativa em relação a performance dos dois protocolos com simulação de perdas de sinal. A estrutura do *middleware* é apresentada na Figura 25. Para os testes foram utilizados dois computadores portáteis, uma placa de desenvolvimento modelo BeagleBoard-xM e um roteador. Em um dos computadores foram executados os servidores CoAP e MQTT, no outro foi executado o software Wanem para simular a perda de sinal na rede, já o *middleware* foi implementado na BeagleBoard-xM que também gerou dados simulando sensores (THANGAVEL et al., 2014).

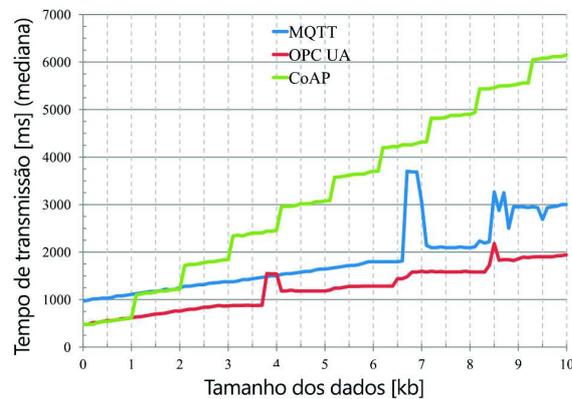
Para a mensuração da quantidade de *bytes* transmitidos e tempo de transmissão foi

Figura 21 – Dados lidos do EM1200 com troca de pacotes.



Fonte – Retirado de Gandhi e Adhyaru (2015).

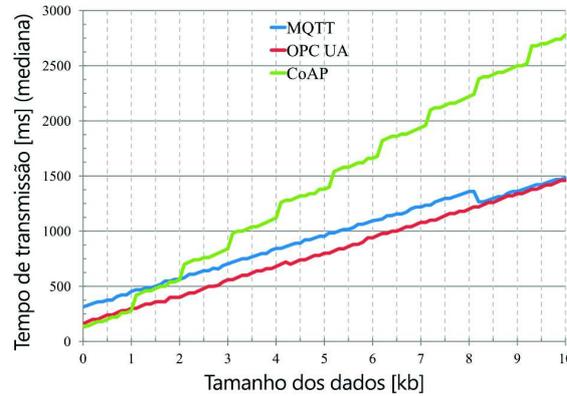
Figura 22 – Respostas para testes realizados com tecnologia 2G.



Fonte – Adaptado de Durkop, Czybik e Jasperneite (2015).

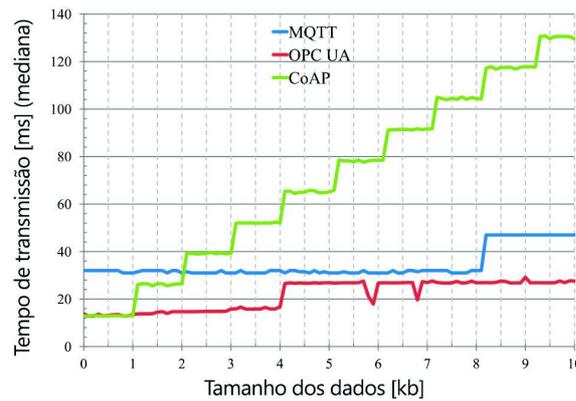
utilizado o *software* de análise de protocolos de rede Wireshark. O atraso medido entre os envios e recebimentos das mensagens estão resumidos no Quadro 3 onde percebe-se que o protocolo MQTT possui uma performance melhor do que o CoAP para perdas de sinal de até 20%, e uma performance inferior para perdas de 25%. Identificando esta variação de performance para diferentes perdas de sinal, o autor propõe que o *middleware* identifique as perdas de sinal e escolha automaticamente o protocolo que será utilizado para garantir uma melhor performance (THANGAVEL et al., 2014).

Figura 23 – Respostas para testes realizados com tecnologia 3G.



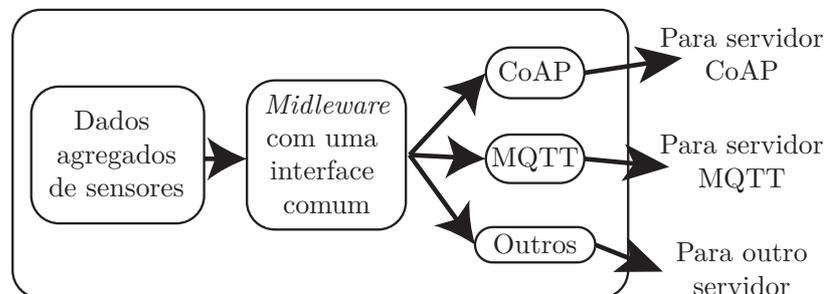
Fonte – Adaptado de Durkop, Czybik e Jasperneite (2015).

Figura 24 – Respostas para testes realizados com tecnologia 4G.



Fonte – Adaptado de Durkop, Czybik e Jasperneite (2015).

Figura 25 – Estrutura do *middleware* proposto por Thangavel et al. (2014).



Fonte – Adaptado de Thangavel et al. (2014).

O atraso nos pacotes do protocolo MQTT estão diretamente relacionados ao seu nível de QoS como é analisado em Lee et al. (2013) onde os autores realizam um experimento prático para determinar a correlação entre o nível de QoS e o tempo de atraso. Neste experimento foram

Quadro 3 – Atraso médio em relação a perda de sinal

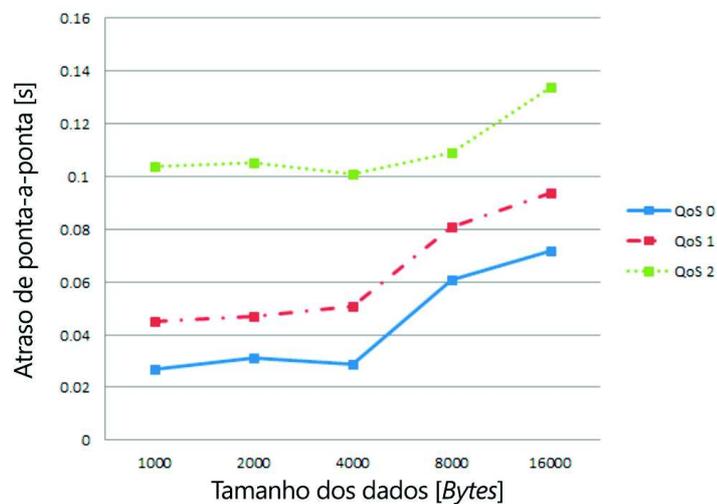
Protocolo	Taxa de perdas					
	0%	5%	10%	15%	20%	25%
MQTT	5,289 ms	22,609 ms	144,252 ms	450,860 ms	871,025 ms	10333,48 ms
CoAP	8,847 ms	552,006 ms	642,688 ms	1026,007 ms	1886,339 ms	2824,115 ms

Fonte – Adaptado de Thangavel et al. (2014).

utilizados dois computadores e dois celulares, sendo que os testes foram realizados em duas etapas, uma com rede cabeada, e outra utilizando a rede de celular com tecnologia 3G (LEE et al., 2013).

Na Figura 26 são apresentados os atrasos para os diferentes níveis de QoS em rede cabeada medidos por Lee et al. (2013). É possível verificar que quanto maior o nível de QoS, maior é o atraso de ponta a ponta, isto está diretamente relacionado com a quantidade de verificações de cada nível. A Figura 27 apresenta para esta mesma rede a perda de pacotes em relação ao tamanho do dados, onde percebe-se que com os dados até 8000 *bytes* a perda é muito baixa, e quando esta perda torna-se significativa o nível de QoS influencia, destacando o nível 2 que se manteve bem abaixo dos demais (LEE et al., 2013).

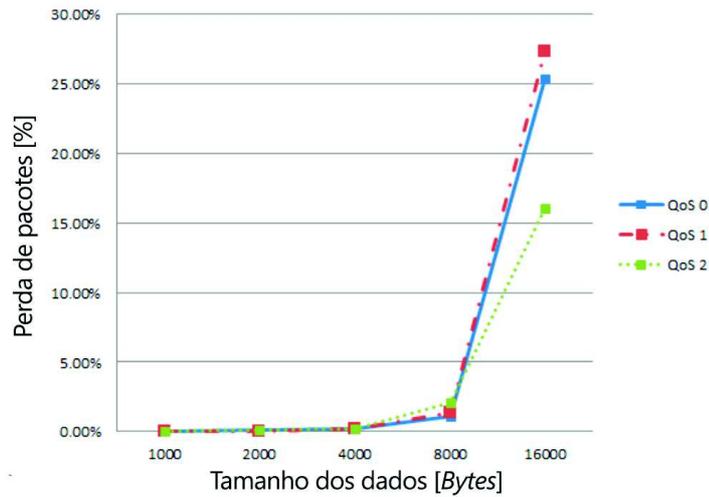
Figura 26 – Atraso para rede cabeada



Fonte – Adaptado de Lee et al. (2013).

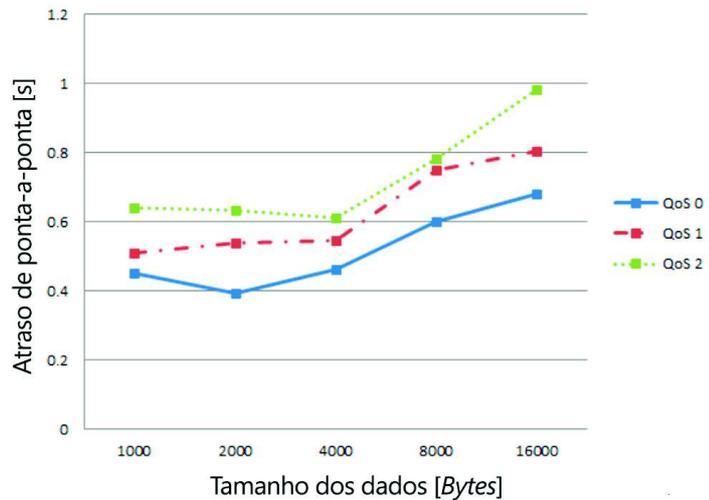
Para a rede celular foram realizados os mesmos testes, o atraso pode ser visto na Figura 28 onde percebe-se que o mesmo aumentou, em relação ao teste anterior, para todos os níveis de QoS porém a diferença entre eles diminuiu consideravelmente. Já a perda de pacotes, apresentada na Figura 29, caiu significativamente em relação ao teste anterior, é perceptível que o nível 2 teve muito pouca perda de pacotes, não chegando a 0,20% , ou seja, foram entregues mais de 99,8% dos pacotes (LEE et al., 2013).

Figura 27 – Perda de pacotes para rede cabeada



Fonte – Adaptado de Lee et al. (2013).

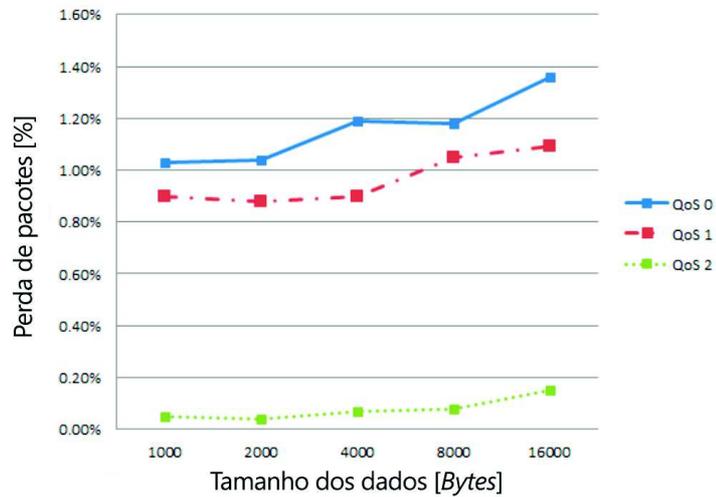
Figura 28 – Atraso para rede celular



Fonte – Adaptado de Lee et al. (2013).

A confluência das técnicas apresentadas nos trabalhos deste capítulo servem de motivação para o desenvolvimento do presente trabalho, uma vez que Temiz et al. (2015) utiliza o protocolo Modbus para a criação de uma VPP, e Lampkin et al. (2012) propõe a criação de VPP utilizando o protocolo MQTT que ainda não é suportado por muitos equipamentos. É importante dar atenção também a verificação de erros e a garantia de entrega dos pacotes como atenta Gandhi e Adhyaru (2015), assim como escolher corretamente o nível de QoS avaliando os limites de tempo de atraso de entrega e as perdas de pacotes aceitáveis. Esta relação entre os trabalhos estudados neste capítulo com o sistema proposto está resumida no Quadro 4. No capítulo seguinte é apresentada a metodologia escolhida para o desenvolvimento do trabalho.

Figura 29 – Perda de pacotes para rede celular



Fonte – Adaptado de Lee et al. (2013).

Quadro 4 – Trabalhos relacionados.

Autores	Trabalho realizado	Relação com o trabalho
Temiz et al. (2015)	VPP modbus usando IEC61850	Exemplo de aplicação para o estudo de caso
(LAMPKIN et al., 2012)	Proposta de usar MQTT para a criação de uma VPP	Definição do protocolo
Li, Zhang e Peng (2011)	Desenvolvimento de um <i>gateway</i> Modbus utilizando EPA	Motivação de aplicação
Gandhi e Adhyaru (2015)	Teste de perda de pacotes em transmissão sem fio	Definição da metodologia a ser utilizada
Durkop, Czybik e Jasperneite (2015)	Performance dos protocolos de IoT em redes de celular	
Thangavel et al. (2014)	Performance dos protocolos de IoT em <i>gateway</i> comum	
Lee et al. (2013)	Relação entre o atraso de pacotes e o nível de QoS do MQTT	

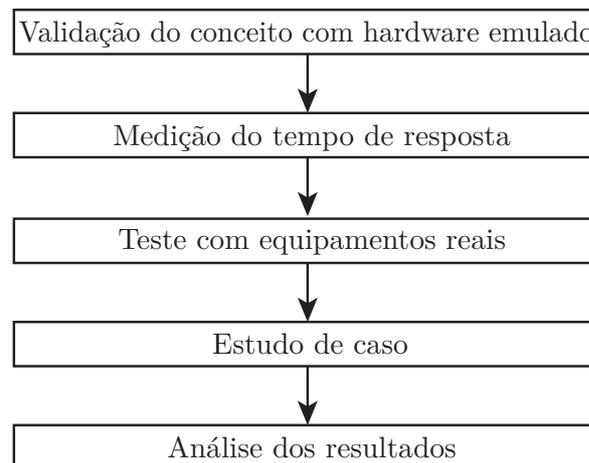
Fonte – Elaborada pelo autor.

4 METODOLOGIA

Neste capítulo são apresentadas as etapas e ferramentas que foram utilizadas no desenvolvimento deste trabalho e como o mesmo foi executado.

O desenvolvimento do trabalho foi dividido em 5 etapas como é apresentado na Figura 30, começando com a validação do conceito com *hardware* emulado, seguindo para a medição do tempo de resposta para então prosseguir para os testes com equipamentos reais. Feito isto, foi montado o estudo de caso onde foi desenvolvida uma VPP para a aplicação do sistema proposto em uma situação real. Este estudo de caso serve de base para a análise dos resultados para a validação do uso do sistema em casos reais.

Figura 30 – Etapas do desenvolvimento do trabalho



Fonte – Elaborado pelo autor.

Os testes foram executados em laboratório, utilizando computadores com sistema operacional Windows 8.1 ®, sendo conectados a internet utilizando rede sem fio IEEE 802.11. Para a comunicação da interface EIA-485 com os computadores foram utilizados conversores de interface como o apresentado na Figura 31.

Figura 31 – Conversor EIA-485 para USB (*Universal Serial Bus*)



Fonte – Elaborado pelo autor.

A implementação foi realizada utilizando a linguagem de programação Python, uma linguagem de *script* de alto nível interpretada, ideal para a prototipação de sistemas que apesar de ser nativa em sistemas operacionais baseados em Unix como o Linux, também pode ter o seu interpretador instalado em sistemas como Windows, permitindo assim o desenvolvimento multi-plataforma.

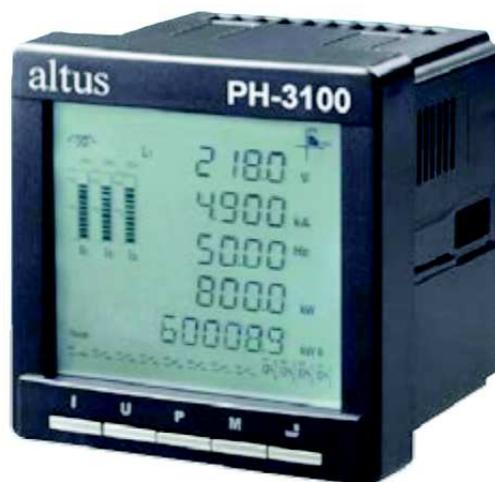
Para realizar a medição dos tempos de resposta, foi utilizado o software de análise de protocolos de rede wireshark, tal como foi realizado em Thangavel et al. (2014). Os testes foram realizados utilizando nível 2 de QoS uma vez que se trata de uma aplicação de automação e não deseja-se receber a mesma mensagem mais vezes do que foi enviada como pode ocorrer em QoS 0 e 1.

Os pacotes interceptados pelo *software* wireshark, foram filtrados utilizando um *script* escrito em Python utilizando a biblioteca PyShark. Este filtro analisa a origem dos pacotes e calcula o tempo de resposta de cada pacote, e se houve retransmissão de pacotes.

Como o sistema operacional não é um sistema de tempo real, podem ocorrer atrasos e até perdas na aquisição dos pacotes, então optou-se por interceptar os pacotes diretamente na interface EIA-485, para isto foi necessário desenvolver um dispositivo dedicado para a medição do tempo de respostas que é detalhado na seção 4.1 deste trabalho.

Com o conceito validado, foram executados testes com equipamentos reais com suporte a comunicação através do protocolo Modbus, como os multimedidores de grandezas elétricas modelo PH3100 da marca ALTUS apresentado na Figura 32 e modelo Nexus II da marca ABB apresentado na Figura 33, ambos possuem funções equivalentes ao EM1200 utilizado por Gandhi e Adhyaru (2015).

Figura 32 – Multimetro de grandezas elétricas ALTUS PH3100



Fonte – Adaptado de www.altus.com.br.

Figura 33 – Multimetro de grandezas elétricas ABB Nexus II



Fonte – Adaptado de www.abb.com.

O estudo de caso consiste na aplicação prática do sistema proposto em uma solução voltada para monitoramento e controle de redes elétricas inteligentes, onde foi desenvolvida uma VPP interligando duas unidades geradoras de energia solar fotovoltaica situadas em diferentes localidades. Este estudo de caso tem por objetivo fornecer informação sobre o funcionamento do *gateway* apresentado para aplicações reais.

Deste estudo de caso foi realizada uma análise quantitativa do tráfego de dados entre as duas redes, utilizando o dispositivo para medição do tempo de resposta, para a avaliação da performance do sistema proposto. O sistema utilizado neste trabalho é explicado no capítulo 5, explicitando os seus objetivos, delimitações e detalhes sobre a sua implementação.

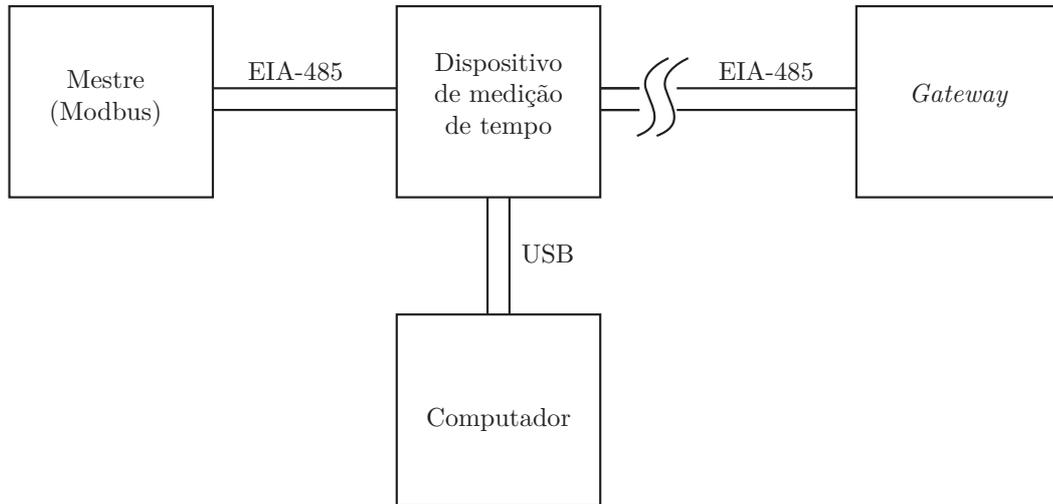
4.1 Dispositivo para medição de tempo de resposta em interface EIA-485

Para realizar uma medição mais exata do tempo de resposta do sistema, foi desenvolvido um dispositivo específico para interceptar os pacotes na interface EIA-485 e medir o tempo entre a requisição e a resposta, este deve ser instalado em série com o mestre da rede modbus e a rede EIA-485 conforme apresentado na Figura 34.

Optou-se por utilizar um microcontrolador pois o mesmo pode ser programado para uma aplicação dedicada, sem a necessidade de um sistema operacional, possuindo *hardware* para temporização, comunicação serial com disparo de funções por interrupção externa. Foi escolhido o microcontrolador STM32F334R8T6 da família ARM Cortex ® M4, presente no kit de desenvolvimento ST Nucleo-F334R8 apresentado na Figura 35.

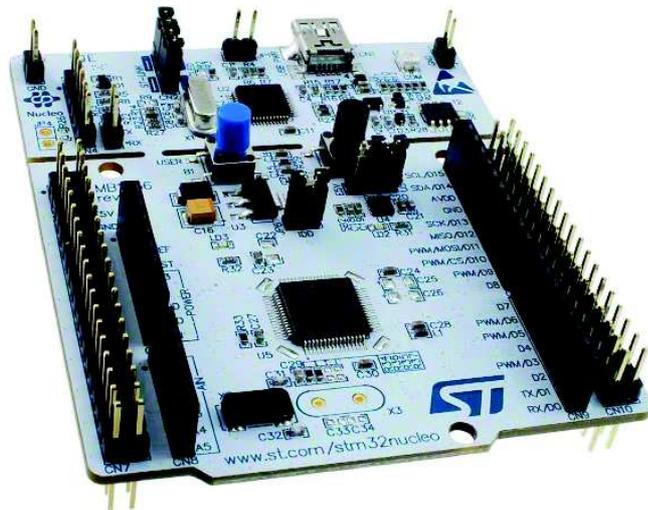
O dispositivo identifica o início das mensagens por meio de interrupções externas acionadas pelos pinos de comunicação do conversor de interface, sendo o pino TX para as mensagens

Figura 34 – Aplicação do dispositivo para medição de tempo de resposta.



Fonte – Elaborado pelo autor.

Figura 35 – Kit de desenvolvimento ST Nucleo 334R8

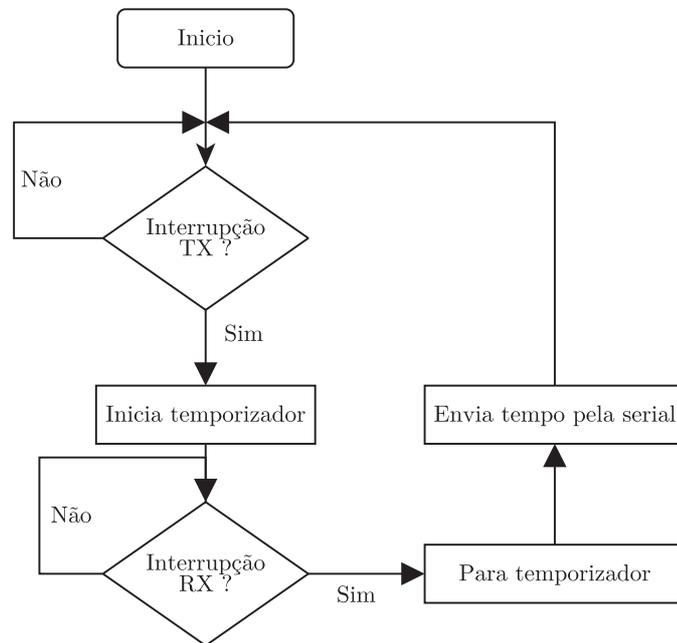


Fonte – Retirado de digikey.com.

do mestre para a rede, e o pino RX para as mensagens da rede para o mestre. Uma vez que seja detectada a requisição, mensagem do mestre para a rede, é iniciada a contagem do tempo. Quando for identificada a resposta, mensagem da rede para o mestre, a contagem do tempo é parada e seu valor é enviado para o computador através da interface serial, e o dispositivo volta a esperar uma requisição do mestre. Esta lógica de funcionamento é apresentada graficamente na

Figura 36.

Figura 36 – Logica de funcionamento do microcontrolador.



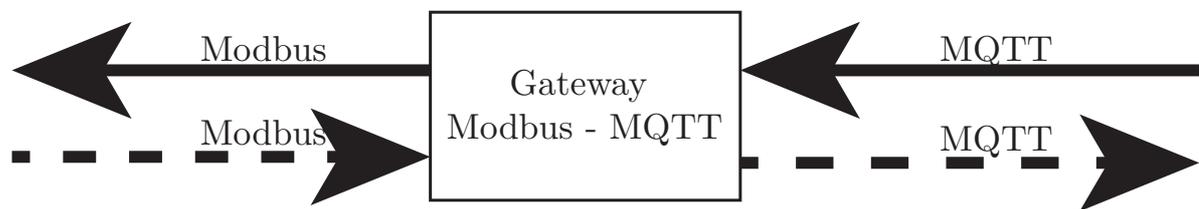
Fonte – Elaborado pelo autor.

Para o recebimento dos dados pelo computador, foi desenvolvido um *software* em Python que lê os dados recebidos pela interface serial, e armazena os valores das medições de tempo de forma ordenada em um arquivo de texto. Este arquivo de texto é importado para o *software* Excel [®] onde são analisados os tempos de resposta mínimos, médios e máximos assim como a sua distribuição, para que possa ser definido o tempo limite para o recebimento da resposta que garanta o recebimento da mensagem, e assim verificar a viabilidade da utilização do *gateway* em situações reais.

5 SISTEMA PROPOSTO

Neste trabalho é proposto o desenvolvimento de um *gateway* capaz de interligar redes de campo que utilizam o protocolo Modbus RTU através do protocolo MQTT, além da criação de uma VPP para estudo de caso quanto a sua aplicação em redes elétricas inteligentes. O conceito do *gateway* é apresentado na Figura 37 onde as setas contínuas representam a conversão do MQTT para Modbus, e as setas tracejadas representam a conversão Modbus para MQTT.

Figura 37 – Conceito do *gateway* Modbus - MQTT



Fonte – Elaborado pelo autor.

Para a validação do conceito, o *gateway* e o emulador de *hardware* foram implementados utilizando a linguagem de programação Python. Para isto foi utilizada a biblioteca *Paho Python Client* que implementa as funções necessárias de um cliente MQTT. Para que o *software* *wireshark* pudesse interceptar todos os pacotes MQTT para realizar a medição do tempo entre a requisição e a resposta, foi necessário que os dois clientes fossem executados no mesmo computador. Como *broker* MQTT foi utilizado o servidor Mosquitto rodando em um servidor dedicado hospedado na cidade de Nova Iorque, nos Estados Unidos da América, pela empresa Digital Ocean. Este servidor foi escolhido devido a grande distância entre os equipamentos e o servidor.

O caminho de rede entre o local do teste e o servidor é apresentado na Figura 38, apresentando uma distância aproximada de 13.550 quilômetros, logo entre a requisição e a resposta existe uma distância aproximada de 27.100 quilômetros, distâncias que não seriam viáveis utilizando apenas a interface EIA-485 que é limitada a um comprimento máximo de 1.200 metros.

Uma vez que o conceito foi validado, o *gateway* foi modificado, retirando a emulação do *hardware* e implementada a identificação das mensagens por interface serial utilizando a biblioteca da linguagem Python denominada *pyserial*, de modo que funcionasse com equipamentos reais. Para isto foi necessário também implementar a detecção de *frames* Modbus baseada em tempo.

O algoritmo para a conversão das mensagens Modbus para MQTT é apresentado em forma de fluxograma na Figura 39. A conversão identifica o endereço do escravo, o código de

Figura 38 – Caminho dos pacotes.



Fonte – Elaborado pelo autor.

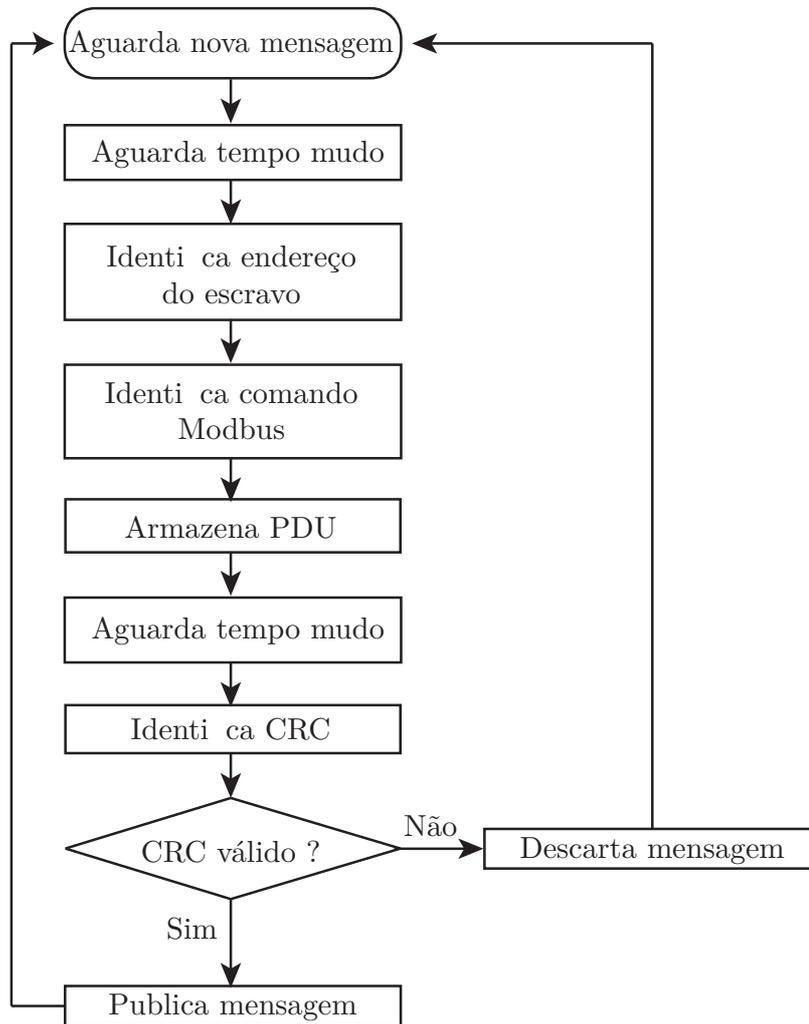
função da mensagem e o CRC, para enviar somente o bloco de dados e CRC caso a mensagem possuir o CRC correto.

A estrutura de tópicos MQTT é apresentada na Figura 40. Esta é dividida em três níveis para aproveitar a flexibilidade dos tópicos e caracteres curinga disponíveis no protocolo MQTT, podendo limitar o acesso à funções específicas ou monitorar diversos clientes ao mesmo tempo.

Os *gateways* podem ser configurados para repassar todas as mensagens ou somente as de alguns escravos que estejam conectados no barramento serial, não ocupando a rede com mensagens para escravos que estão em outra parte da rede.

A recomposição da mensagem é realizada utilizando o algoritmo apresentado na Fi-

Figura 39 – Algoritmo para conversão das mensagens Modbus para MQTT.



Fonte – Elaborado pelo autor.

Figura 40 – Estrutura das mensagens do *gateway*

Estrutura: Endereço/Função/Dados

Exemplo: 02/03/006B00037687

Fonte – Elaborado pelo autor.

Figura 41, onde, ao receber a publicação em um tópico o *gateway* forma o *frame* Modbus novamente a partir do tópico e dos dados recebidos, uma vez formada a mensagem esta é escrita na rede Modbus.

Com o sistema operando conforme o esperado no ambiente de virtualização, foi montado o estudo de caso, que consiste a utilização do sistema para implementação de uma VPP, que

Figura 41 – Algoritmo para conversão das mensagens MQTT para Modbus.

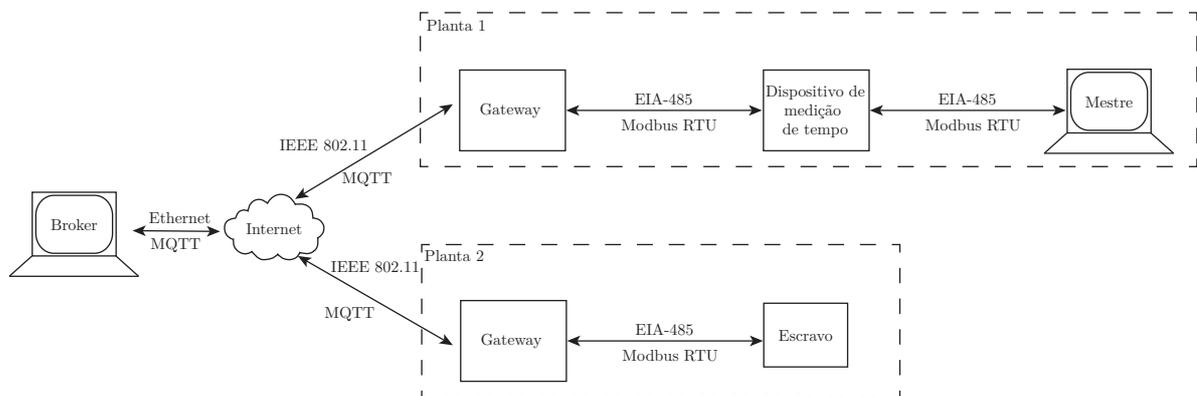


Fonte – Elaborado pelo autor.

interliga em termos de comunicação duas plantas de uma usina de geração de energia solar fotovoltaica situadas em diferentes localidades. O tempo de resposta é monitorado pelo dispositivo para medição de tempo de resposta, sendo os seus resultados registrados em um arquivo para análise posterior.

A VPP proposta tem a sua montagem esquematizada na Figura 42 onde tem-se duas redes com comunicação Modbus localizadas em locais diferentes que são interligadas pela internet através do protocolo MQTT, sendo esta conversão realizada pelo sistema proposto neste trabalho.

Figura 42 – VPP utilizada para estudo de caso.



Fonte – Elaborado pelo autor.

Na Planta 1, o Mestre é um sistema SCADA que solicita informações como tensão, corrente, potência e energia para o Escravo da Planta 2. Este escravo é o multimetro de grandezas elétricas, que é ligado na saída de um gerador de energia solar fotovoltaica com potência instalada de 1,56 kWp, com um inversor monofásico com potência de 1,5 kW modelo Galvo 1.5-1 da marca Fronius ®.

6 ANÁLISE DE RESULTADOS

Com a estrutura para o estudo de caso montada, conforme Figura 42 foram realizados três testes para medição do tempo de resposta, utilizando a mesma estrutura, tanto de *hardware* como de rede, porém em períodos diferentes do dia.

Destes testes, foram analisados os tempos mínimos, máximos, e médios entre as requisições e as respostas, que estão apresentados na Tabela 3, juntamente com a quantidade total de mensagens por teste. Pode-se perceber que os tempos de resposta variaram entre 330 e 4547 milissegundos, apresentando uma variação de 4217 milissegundos. Apesar desta variação, os tempos médios mantiveram-se entre 851 e 879 milissegundos, com uma variação de apenas 28 milissegundos.

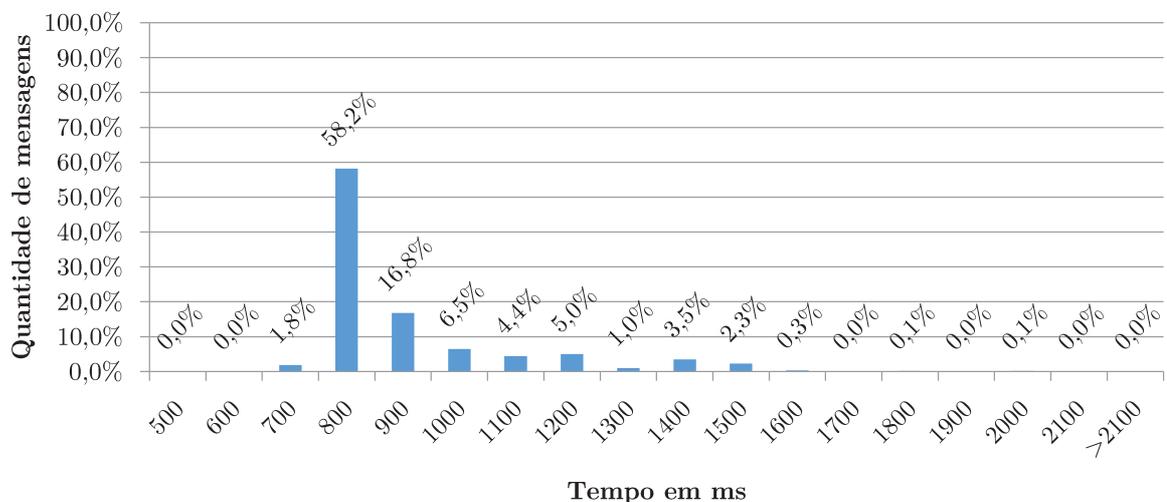
Tabela 3 – Resumo dos resultados dos testes de tempo de resposta.

Teste	Mensagens	Tempo mínimo [ms]	Tempo Máximo [ms]	Tempo médio [ms]
1	883	694	1942	851
2	2168	330	2269	876
3	4932	635	4547	879

Fonte – Elaborado pelo autor.

O primeiro teste tem a sua distribuição de tempo de resposta apresentada na Figura 43, onde é perceptível que a maioria dos tempos de resposta ficou entre 700 e 1500 milissegundos, sendo que mais da metade das respostas ocorreram em menos de um segundo.

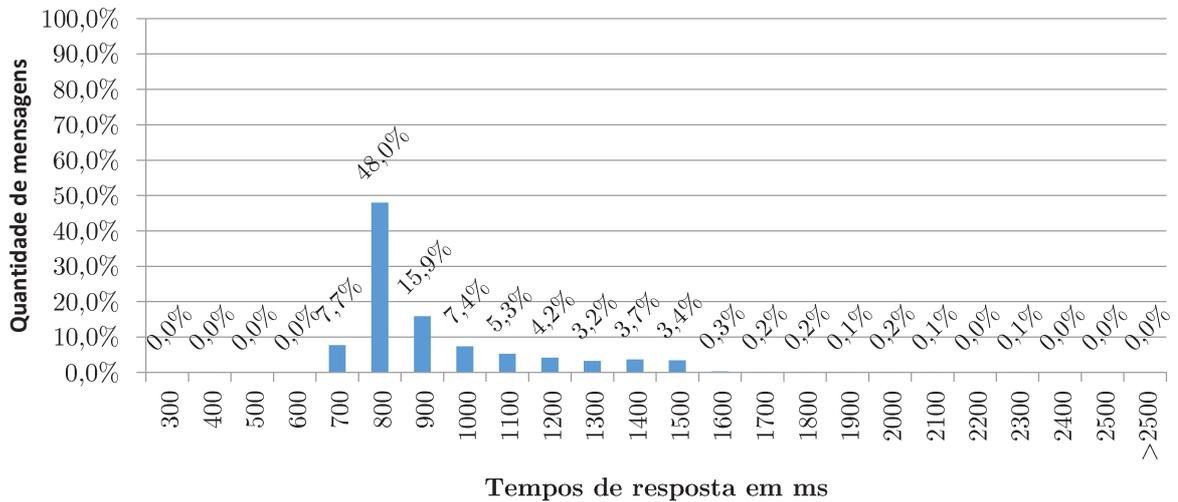
Figura 43 – Tempo de resposta do teste 1



Fonte – Elaborado pelo autor.

No segundo teste, com sua distribuição de tempo de resposta apresentada na Figura 44, os tempos de resposta com maior frequência situam-se entre 700 e 1500 milissegundos, com mais da metade das respostas em menos de um segundo.

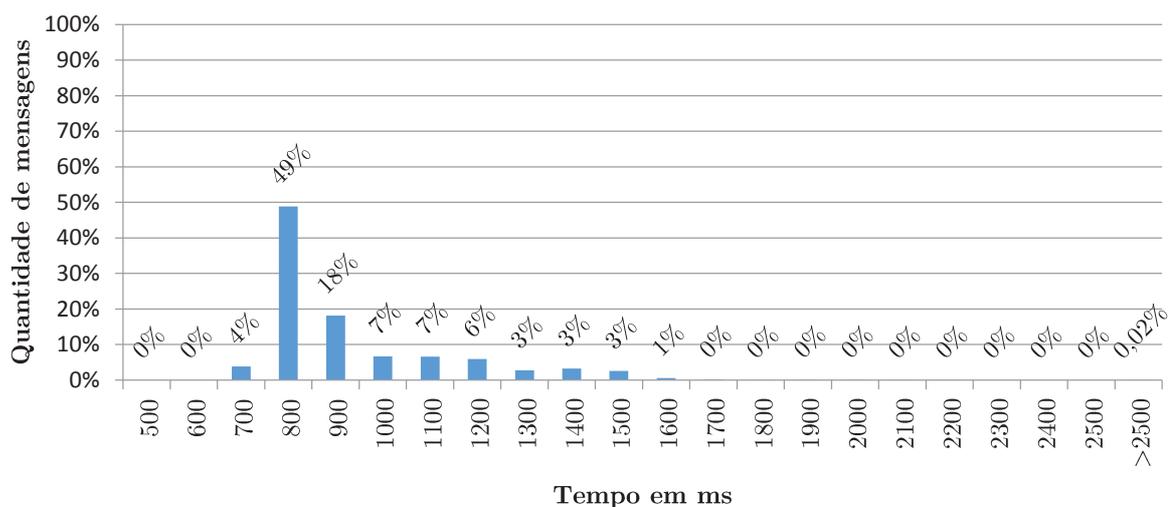
Figura 44 – Tempo de resposta do teste 2



Fonte – Elaborado pelo autor.

O terceiro e último teste tem a sua distribuição de tempos de resposta apresentada na Figura 45, onde a maioria dos tempos de resposta ficou entre 700 e 1600 milissegundos, com mais da metade abaixo de um segundo.

Figura 45 – Tempo de resposta do teste 3

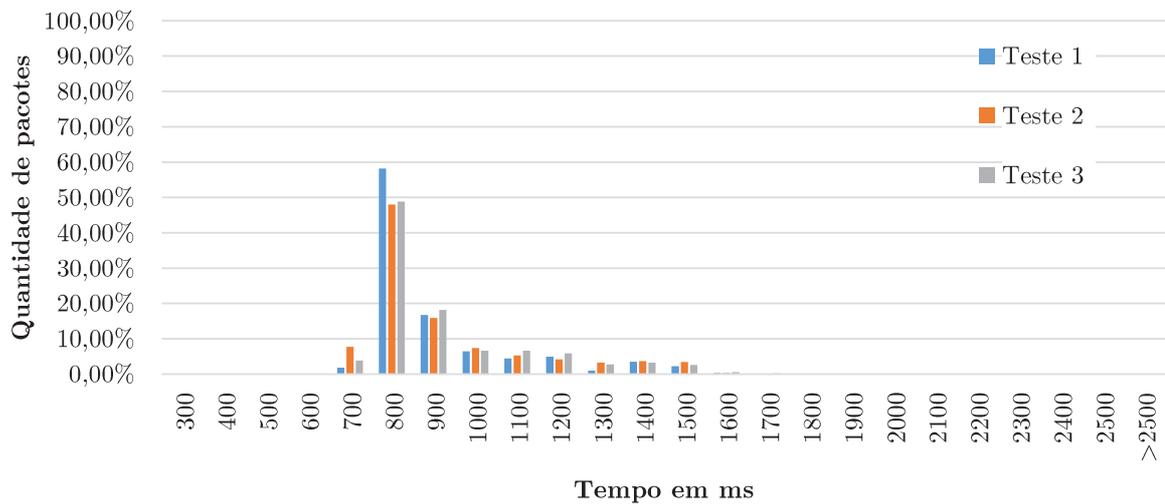


Fonte – Elaborado pelo autor.

Para efeitos de comparação, é apresentado na Figura 46 um histograma com a distribuição de tempo de resposta dos três testes, sendo que os dados estão normalizados. Com isto é

possível perceber que a grande maioria dos dados teve um tempo de resposta entre 700 e 1600 milissegundos, e principalmente entre 800 e 900 milissegundos. Os três testes apresentam uma distribuição similar, com isto pode-se determinar o tempo entre requisições no mestre da rede Modbus.

Figura 46 – Tempo de resposta geral



Fonte – Elaborado pelo autor.

Apesar da internet não garantir o determinismo necessário para grande parte das aplicações de redes de campo, a determinação do tempo entre requisições pode ser feita baseada na distribuição de probabilidade, onde deve-se escolher o tempo levando em conta a mínima perda de mensagens. No entanto o uso de um meio que não garanta determinismo, só é recomendado para aplicações não críticas, como por exemplo medição e controle de sistemas com grande inércia.

Os testes foram realizados com um tempo entre requisições de 5 segundos, o que pelo resultado dos testes pode ser considerado como o dobro do tempo suficiente, uma vez que apenas uma mensagem de um total de 7983 mensagens, ou 0,01253% das mensagens, teve um tempo superior a 2,5 segundos.

CONCLUSÃO

Sendo o protocolo Modbus um dos mais utilizados em ambientes industriais e em equipamentos para controle e geração de energia distribuída, conforme McConahaV (2011), Zurawski (2005). É notável a necessidade de interligação entre redes Modbus situadas em diferentes localidades para obter uma maior flexibilidade e integração entre sistemas.

Este trabalho que teve por objetivo o desenvolvimento de um dispositivo que atenda esta necessidade, demonstrou que, com base nas técnicas e metodologias pesquisadas foi possível propor e desenvolver tal sistema. Este mostrou-se factível, e atendeu de forma satisfatória os seus objetivos, tanto com equipamentos emulados, quanto com equipamentos reais.

Através do estudo de caso, foi possível ver a sua aplicação em uma situação real, onde a criação de uma VPP para uma usina de energia solar fotovoltaica, onde duas redes Modbus foram interligadas de forma transparente em termos de comunicação, através do *gateway* desenvolvido, independente da distância entre elas. Esta aplicação teve seus tempos de resposta mensurados para a avaliação da performance do sistema proposto, a qual demonstrou-se extremamente satisfatória.

A partir dos dados medidos, foi possível determinar que o tempo entre as requisições, para a situação do estudo de caso onde a distância entre as duas redes ultrapassa 13 mil quilômetros, deve ser de 2,5 segundos ou mais para obter uma baixa taxa de perdas de mensagens.

O sistema possui grande potencial para ser explorado em plantas com leiaute reconfigurável, onde os equipamentos podem estar em movimento, ou são reposicionados com frequência, sem a necessidade de realizar um novo cabeamento para a rede EIA-485, podendo utilizar a rede sem fios IEEE 802.11, e diminuindo assim o custo de infraestrutura da planta.

Para a continuidade deste trabalho em trabalhos futuros, fica proposta a implementação do *gateway* em um sistema embarcado em forma de produto para possível comercialização, de modo que não dependa de um computador para o seu funcionamento, possa ser facilmente aplicado em sistemas reais que utilizem o protocolo Modbus RTU para comunicação.

REFERÊNCIAS

ALLAMARAJU, S. *RESTful Web Services Cookbook: Solutions for Improving Scalability and Simplicity*. [S.l.]: O'Reilly Media, 2010. ISBN 9781449388843. Citado na página 17.

ANEEL. *Resolução Normativa nº482/2012*. [S.l.], 2012. Citado na página 30.

ANEEL. Micro e Minigeração Distribuída Sistema de Compensação de Energia Elétrica. *Cadernos temáticos. Agência Nacional de Energia Elétrica*, p. 32, 2014. Citado 2 vezes nas páginas 28 e 30.

CASSAGRAS. *CASAGRAS and The Internet of Things: Definition and Vision Statement Agreed*. Disponível em: <<http://www.rfidglobal.eu/userfiles/documents/CASAGRAS26022009.pdf>>. Citado na página 14.

CERP-IOT. *Internet of Things: Strategic Research Roadmap*. Disponível em: <http://www.grifs-project.eu/data/File/CERP-IoT%20SRA_IoT_v11.pdf>. Citado na página 14.

DRATH, R.; HORCH, A. Industrie 4.0: Hit or hype? *IEEE Industrial Electronics Magazine*, v. 8, n. 2, p. 56–58, 2014. ISSN 19324529. Citado na página 21.

DURKOP, L.; CZYBIK, B.; JASPERNEITE, J. Performance evaluation of m2m protocols over cellular networks in a lab environment. In: *Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on*. [S.l.: s.n.], 2015. p. 70–75. Citado 5 vezes nas páginas 33, 34, 35, 36 e 39.

EPOSS. *Internet of Things in 2020: Roadmap for the Future*. Disponível em: <<http://www.smart-systems-integration.org/public>>. Citado na página 14.

FREITAS B. M. R. DE; HOLLANDA, L. *Micro e minigeração no Brasil: Viabilidade econômica e Entraves do Seror*. [S.l.], 2015. Citado 2 vezes nas páginas 28 e 29.

GANDHI, P.; ADHYARU, D. Critical data reliability issues with Modbus protocol in smart grid application. In: *2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO)*. IEEE, 2015. p. 1–5. ISBN 978-1-4799-7676-8. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7253788>>. Citado 7 vezes nas páginas 32, 33, 34, 35, 38, 39 e 41.

GREENGARD, S. *The Internet of Things*. [S.l.]: MIT Press, 2015. (The MIT Press Essential Knowledge series). ISBN 9780262527736. Citado na página 21.

Hermann, Mario; Pentek, Tobias; Otto, B. Design Principles for Industrie 4.0 Scenarios: A Literature Review. 2015. Citado na página 21.

HIVEMQ. *MQTT Essentials Part 5: MQTT Topics & Best Practices*. 2015. Disponível em: <<http://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices>>. Acesso em: 16 dez. 2015. Disponível em: <<http://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices>>. Citado 2 vezes nas páginas 18 e 19.

IEC - *International Electrotechnical Commission. Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 1: Overview and guidance for the IEC 61158 series*. 1. ed. [S.l.], 2003. Citado na página 22.

INEE. *O QUE É GERAÇÃO DISTRIBUÍDA*. 2015. Acesso em: 21 dez. 2015. Disponível em: <http://www.inee.org.br/forum/_ger/_distrib.asp>. Citado na página 28.

JAFFEY, T. *MQTT and CoAP, IoT Protocols*. 2014. Acesso em: 16 dez. 2015. Disponível em: <https://eclipse.org/community/eclipse{_}newsletter/2014/february/article2.> Citado 3 vezes nas páginas 16, 17 e 18.

LAMPKIN, V. et al. *Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry*. IBM Redbooks, 2012. (IBM redbooks). ISBN 9780738437088. Disponível em: <https://books.google.com.br/books?id=F_HHAgAAQBAJ>. Citado 5 vezes nas páginas 18, 31, 34, 38 e 39.

LEE, S. et al. Correlation analysis of mqtt loss and delay according to qos level. In: *Information Networking (ICOIN), 2013 International Conference on*. [S.l.: s.n.], 2013. p. 714–717. ISSN 1976-7684. Citado 4 vezes nas páginas 36, 37, 38 e 39.

LI, H.; ZHANG, H.; PENG, D. Research and design of industrial ethernet communication gateway on power station. In: *Transportation, Mechanical, and Electrical Engineering (TMEE), 2011 International Conference on*. [S.l.: s.n.], 2011. p. 986–989. Citado 3 vezes nas páginas 31, 32 e 39.

LOPES, Y.; FERNANDES, N. C.; CHRISTINA, D. M. Geração Distribuída de Energia: Desafios e Perspectivas em Redes de Comunicação. *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, p. 2–55, 2015. Citado 2 vezes nas páginas 28 e 29.

MACDOUGALL, W. *INDUSTRIE 4.0. Smart Manufacturing for the Future*. [S.l.], 2014. Disponível em: <http://www.gtai.de/GTAI/Content/EN/Invest/{_}SharedDocs/Downloads/GTAI/Brochures/Industries/industrie4.0-smart-manufacturing-for-the-future-en.> Citado na página 21.

MCCONAHAV, J. Using modbus for process control and automation. *measurements*, v. 100, p. 1, 2011. Citado 3 vezes nas páginas 22, 26 e 52.

MCEWEN, A.; CASSIMALLY, H. *Designing the Internet of Things*. Wiley, 2013. ISBN 9781118430651. Disponível em: <<http://books.google.com.br/books?id=iYkKAgAAQBAJ>>. Citado 3 vezes nas páginas 14, 15 e 16.

MESNIL, J. *Mobile and Web Messaging: Messaging Protocols for Web and Mobile Devices*. [S.l.]: O'Reilly Media, 2014. ISBN 9781491944769. Citado na página 18.

Modbus Organization. *MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE V1.0b*. [S.l.], 2006. 1–46 p. Citado 3 vezes nas páginas 26, 27 e 28.

Modbus Organization. *MODBUS over Serial Line Specification and Implementation Guide V1.02*. [S.l.], 2006. 1–44 p. Citado 3 vezes nas páginas 25, 26 e 27.

Modbus Organization. *MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3*. [S.l.], 2012. 1–50 p. Citado 3 vezes nas páginas 23, 24 e 25.

OASIS Open. *MQTT Version 3.1.1 OASIS Standard*. [S.l.], 2014. Citado na página 19.

REYNDERS, D.; MACKAY, S.; WRIGHT, E. *Practical Industrial Data Communications: Best Practice Techniques*. Elsevier Science, 2004. (Practical professional books from Elsevier). ISBN 9780080480138. Disponível em: <<https://books.google.com.br/books?id=hZ-VvrufuyEC>>. Citado na página 22.

RICHARDSON, L.; AMUNDSEN, M.; RUBY, S. *RESTful Web APIs*. [S.l.]: O'Reilly Media, 2013. ISBN 9781449359737. Citado na página 17.

RICHARDSON, L.; RUBY, S. *RESTful Web Services*. [S.l.]: O'Reilly Media, 2008. ISBN 9780596554606. Citado na página 16.

SABOORI, H.; MOHAMMADI, M.; TAGHE, R. Virtual Power Plant (VPP), Definition, Concept, Components and Types. *2011 Asia-Pacific Power and Energy Engineering Conference*, p. 1–4, 2011. ISSN 2157-4839. Disponível em: <<http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5749026>>. Citado na página 30.

SAP. *SAP: Internet of Things: An Integral Part of the Future Internet*. Citado na página 14.

STEMMER, M. R. *Redes locais industriais: a integração da produção através das redes de comunicação*. [S.l.]: Ed. da UFSC, 2010. ISBN 9788532804921. Citado 4 vezes nas páginas 22, 23, 24 e 25.

TANENBAUM, A. *Computer Networks*. Prentice Hall PTR, 2003. (Computer Networks, p. 3). ISBN 9780130661029. Disponível em: <<https://books.google.com.br/books?id=Pd-z64SJRbAC>>. Citado 2 vezes nas páginas 18 e 22.

TEICH, P. *Behaviorally Segmenting the IoT*. [S.l.], 2013. Citado 2 vezes nas páginas 16 e 17.

TEMIZ, A. et al. A smart grid controller for low voltage pv network. In: *Power Engineering, Energy and Electrical Drives (POWERENG), 2015 IEEE 5th International Conference on*. [S.l.: s.n.], 2015. p. 263–267. Citado 4 vezes nas páginas 6, 31, 38 e 39.

THANGAVEL, D. et al. Performance evaluation of mqtt and coap via a common middleware. In: *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*. [S.l.: s.n.], 2014. p. 1–6. Citado 7 vezes nas páginas 6, 34, 35, 36, 37, 39 e 41.

UCKELMANN, D.; HARRISON, M.; MICHAHELLES, F. *Architecting the Internet of Things*. [S.l.: s.n.], 2011. 1–353 p. ISBN 978-3-642-19156-5. Citado na página 14.

ZHANG, Z.-K. et al. IoT Security: Ongoing Challenges and Research Opportunities. *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications*, p. 230–234, 2014. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6978614>>. Citado na página 20.

ZHOU, H. *The Internet of Things in the Cloud: A Middleware Perspective*. Taylor & Francis, 2012. ISBN 9781439892992. Disponível em: <<http://books.google.com.br/books?id=nIYKwoy1Z6oC>>. Citado 2 vezes nas páginas 14 e 15.

ZURAWSKI, R. *INDUSTRIAL COMMUNICATION TECHNOLOGY*. [S.l.: s.n.], 2005. ISBN 0849330777. Citado 2 vezes nas páginas 22 e 52.