

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Estratégia Distribuída e Modular para a Gestão Inteligente de Carregamento de Veículos Elétricos

Túlio Soares



Mestrado em Engenharia Eletrotécnica e de Computadores

Orientador: David Emanuel Rua

28 de outubro de 2024

Resumo

Esta dissertação aborda a crescente importância da gestão de sessões de carregamento de veículos elétricos (VEs) no contexto da expansão da mobilidade elétrica. Principal contribuição deste trabalho é de uma arquitetura inovadora para a gestão descentralizada dessas sessões, utilizando o protocolo Raft para estabelecer um sistema distribuído entre as estações de carregamento. Esta abordagem visa aumentar a resiliência, flexibilidade e eficiência do sistema, permitindo a coordenação e a gestão eficaz das sessões de carregamento, mesmo em caso de falhas de comunicação ou de nós individuais.

A arquitetura proposta visa integrar as normas ISO 15118 e OCPP, *Open Charge Point Protocol*, garantindo a interoperabilidade e compatibilidade com as tecnologias existentes no ecossistema de carregamento de veículos elétricos. A norma ISO 15118 estabelece a comunicação digital entre o veículo elétrico e o carregador, abrangendo funcionalidades como autenticação, identificação, pagamento e agendamento de carregamento. Por sua vez, o protocolo OCPP é a norma estabelecida pela indústria para a comunicação entre as estações de carregamento e os sistemas de gestão, facilitando a interoperabilidade entre diferentes dispositivos e fabricantes.

A inovação da arquitetura reside na adição de uma rede distribuída de consenso entre os servidores OCPP e na incorporação desses módulos de gestão dentro das unidades de carregamento. Esta abordagem permite a criação de um sistema de gestão local dos carregadores, coordenado pela ativação dos servidores através do mecanismo Raft.

A avaliação experimental da arquitetura proposta, realizada com carregadores reais e simulados, demonstrou resultados promissores. Quando em operação adequada, a rede Raft apresenta um impacto mínimo no desempenho da comunicação e execução do caso de uso OCPP. No entanto, falhas na comunicação ou atrasos na eleição do líder podem comprometer a disponibilidade do servidor OCPP e, conseqüentemente, a gestão das sessões de carregamento. Para mitigar esse risco, a adição à rede distribuída de um servidor redundante assegura a continuidade dos serviços e a disponibilidade dos dados necessários para as sessões de carregamento.

Palavras-Chave: Infraestrutura de carregamento de VEs, Sistemas distribuídos, Gestão das Sessões de Carregamento, OCPP, Agregadores.

Abstract

This dissertation addresses the growing importance of managing electric vehicle (EV) charging sessions in the context of expanding electric mobility. The main contribution of this work is an innovative architecture for the decentralised management of these sessions, using the Raft protocol to establish a distributed system between charging stations. This approach aims to increase the resilience, flexibility and efficiency of the system, enabling effective coordination and management of charging sessions, even in the event of communication or individual node failures.

The proposed architecture aims to integrate the ISO 15118 and OCPP, *Open Charge Point Protocol* standards, ensuring interoperability and compatibility with existing technologies in the electric vehicle charging ecosystem. The ISO 15118 standard establishes digital communication between the electric vehicle and the charger, covering functionalities such as authentication, identification, payment and charging scheduling. The OCPP protocol is the industry standard for communication between charging stations and management systems, facilitating interoperability between different devices and manufacturers.

The architecture's innovation lies in the addition of a distributed consensus network between the OCPP servers and the incorporation of these management modules within the charging units. This approach allows for the creation of a local management system for the chargers, coordinated by the activation of the servers through the Raft mechanism.

The experimental evaluation of the proposed architecture, carried out with real and simulated chargers, showed promising results. When operating properly, the Raft network has minimal impact on communication performance and execution of the OCPP use case. However, communication failures or delays in electing the leader can jeopardise the availability of the OCPP server and, consequently, the management of loading sessions. To mitigate this risk, the addition of a redundant server to the distributed network ensures continuity of services and the availability of the data required for upload sessions.

Keywords: EV charging infrastructure, distributed systems, charging session management, OCPP, aggregators.

Agradecimentos

Elevo meus mais sinceros agradecimentos ao meu Senhor Jesus Cristo, cuja graça e amor me sustentam a cada dia. Agradeço à minha amada esposa, Ana Beatriz Durvino de Oliveira, companheira leal e fortaleza em cada batalha, e a toda a minha família, pilares de força e apoio em cada decisão, em especial a minha mãe, Deiselene de Oliveira, meu pai, e meus irmãos, Igor Soares e Gôndia Soares.

Estendo a minha profunda gratidão à Igreja no Porto, que me acolheu como família durante estes cinco anos, e à FEUP, pela excelência da sua estrutura e corpo docente, e ao INESC TEC, que me proporcionou não somente a realização deste trabalho no seu laboratório, mas também um ambiente de aprendizado e crescimento profissional ao longo do curso.

Agradeço especialmente ao coorientador da tese Rui Lopes Campos pelo tempo dedicado, pelo cuidado em cada detalhe e pelo modelo de paciência e resiliência que ajudou a finalizar este trabalho.

Túlio Soares

*“Todos se dedicavam de coração ao ensino dos apóstolos, à comunhão, ao partir do pão e à oração.
Havia em todos eles um profundo temor, e os apóstolos realizavam muitos sinais e maravilhas.
Os que criam se reuniam num só lugar e compartilhavam tudo que possuíam.
Vendiam propriedades e bens e repartiam o dinheiro com os necessitados, adoravam juntos no templo diariamente, reuniam-se nos lares para comer e partiam o pão com grande alegria e generosidade,
sempre louvando a Deus e desfrutando a simpatia de todo o povo. E, a cada dia, o Senhor lhes acrescentava aqueles que iam sendo salvos. ”*

Atos 2:42-47

Conteúdo

1	Introdução	1
1.1	Contextualização e Motivação	5
1.2	Objetivos	6
1.3	Ambiente de realização da dissertação	8
1.4	Estrutura da dissertação	8
2	Estado da Arte	9
2.1	Infraestrutura de carregamento	9
2.1.1	Tipos de veículos elétricos	9
2.1.2	Tipos de carregadores	10
2.1.3	Agregadores	11
2.1.4	Normas relacionadas ao carregamento de veículos elétricos	12
2.2	Parametrização da infraestrutura de carregamento	18
2.3	Topologias para carregamento coordenado	20
2.3.1	Arquitetura de controlo centralizada	20
2.3.2	Arquitetura de controlo distribuída	21
2.4	Estratégias de carregamento coordenado	22
2.4.1	Algoritmos de carregamento coordenado	23
2.4.2	Algoritmos de consenso	23
2.5	Raft	25
2.5.1	Mecanismos de consenso em mobilidade elétrica	26
2.5.2	Metodologia de testes para comunicação distribuída	31
2.6	Sumário	32
3	Descrição do Sistema Proposto	35
3.1	Plataforma distribuída proposta	37
3.1.1	Blocos funcionais implementados	38
3.1.2	Sistema distribuído de comunicação	43
3.2	Mecanismo de consenso	48
3.3	Sumário	51
4	Avaliação experimental	53
4.1	Unidades de Carregamento	53
4.1.1	Módulo computacional	55
4.1.2	Desempenho computacional	55
4.1.3	Serviços do sistema operativo	55
4.1.4	Componentes eletrónicos	56
4.2	Arquitetura de testes	58

4.2.1	Cenário de sessão de carregamento	58
4.2.2	Parâmetros do protocolo Raft	58
4.2.3	Primeiro ensaio - Sistema centralizado e inter-processos	59
4.2.4	Segundo ensaio - Sistema distribuído e inter-processos	61
4.2.5	Terceiro ensaio - Sistema centralizado e inter-dispositivos	62
4.2.6	Quarto ensaio - Sistema distribuído e inter-dispositivos	62
4.3	Resultados	63
4.3.1	Resultados Computacionais e da Comunicação do protocolo OCPP . . .	64
4.3.2	Resultados da rede Raft	69
4.4	Comparação dos resultados	71
4.5	Sumário	73
5	Conclusões e Trabalho Futuro	75
	Referências	79
A	Implementação do Sistema Proposto	83

Lista de Figuras

1.1	Estações de carregamento por veículos elétricos segundo relatório da agência internacional de energia (IEA) <i>Global EV Outlook</i> [5]	2
1.2	Potência disponível (kW) por VE segundo relatório da agência internacional de energia (IEA) <i>Global EV Outlook 2024</i> [6]	3
2.1	Modos de carregamento	11
2.2	Tomada de conexão <i>Combined Charging System</i> (CCS)	11
2.3	Infraestrutura de carregamento com atores e normas de comunicação	13
2.4	Sinal PWM enviado do Carregador para o VE [40]	14
2.5	Circuitos internos do EVSE e VE	14
2.6	Documentos da ISO 15118 relacionados as camadas OSI	15
2.7	Exemplo de comunicação digital entre VE e EVSE baseado na norma ISO 15118-2 e IEC 61851-1	17
2.8	Topologias de controlo conforme [26]	22
2.9	Estados de um nó do protocolo Raft [28]	25
2.10	Mecanismo de consenso PRAFT [22]	28
2.11	Mecanismo de consenso RBFT [22]	29
2.12	Resultados de estudo dos algoritmos PRAFT e RPBFT [22]	30
3.1	Componentes do sistema informático das unidades de carregamento	35
3.2	Representação real dos atores e os seus módulos de comunicação	36
3.3	Sistema distribuído proposto	37
3.4	Componentes do sistema informático das unidades de carregamento	38
3.5	Componentes do sistema informático das unidades de carregamento	39
3.6	Diagrama lógico dos nós Raft com servidor OCPP	41
3.7	Comunicação entre as unidades de carregamento no sistema proposto	44
3.8	Diagrama de sequência de caso de uso AC - Início da sessão de carregamento	45
3.9	Diagrama de sequência de caso de uso AC - Processo de autenticação e serviços	46
3.10	Diagrama de sequência de caso de uso AC - Negociação energética e agendamento de carregamento	47
3.11	Diagrama de sequência de caso de uso AC - Paragem da sessão de carregamento	48
4.1	Unidade de carregamento EVSE nas instalações externa da INESC TEC (inescevse010)	54
4.2	Unidade de carregamento EVSE utilizadas nos testes no SGEVL, INESC TEC (inescevse005 e 006)	54
4.3	Experiência local com servidor OCPP centralizado	60
4.4	Experiência local com rede Raft (Segundo ensaio)	62
4.5	Experiência na infraestrutura com topologia centralizada (Terceiro ensaio)	63

4.6	Experiência na infraestrutura com rede Raft (Quarto ensaio)	64
4.7	Comparação da comunicação OCPP utilizados durante experiências com topologia Distribuída (com Raft) e Centralizada (sem Raft) em ensaio inter-processos (Simulado) vs inter-dispositivos (Real)	66
4.8	Comparação da comunicação da rede Raft utilizados durante experiências nos carregadores(inter-dipositivos) e na unidade computacional (inter-processos) . . .	66
4.9	Comparação da comunicação OCPP utilizados durante experiências com topologia Distribuída (com Raft) e Centralizada (sem Raft) em ensaio inter-processos (Simulado) vs inter-dispositivos (Real)	66
4.10	Quantidade de mensagens registada nos clientes OCPP no final de cada experiência	67
4.11	Quantidade de mudanças de líder no ensaio real com rede Raft	70
4.12	Comparação dos valores calculados e do ensaio do débito das mensagens Raft (TPS)	72

Lista de Tabelas

2.1	Documentos que constituem a documentação do protocolo OCPP v2.0.1	18
2.2	Especificação dos requisitos temporais de desempenho do sistema	19
2.3	Comparação entre algoritmos de consenso [11].	24
3.1	Seleção de bibliotecas que implementam Raft em Python	50
4.1	Especificação computacional das unidades de carregamento	55
4.2	Serviços e processos executados pelo SO nas unidades de carregamento	56
4.3	Configuração e parâmetros dos carregadores	59
4.4	Parâmetros do mecanismo Raft	59
4.5	Especificação computacional da máquina para inter-processos	60
4.6	Quantidade de mensagens dos clientes OCPP no fim da experiência centralizado .	68
4.7	Quantidade de mensagens dos clientes OCPP no fim do ensaio distribuído (9 nós)	68
4.8	Taxa de sucesso das sessões de carregamento (%) em cada experiência	69
4.9	Latência das mensagens Raft (s) em cada configuração de rede entre processos .	69
4.10	Débito das mensagens Raft (s) em cada configuração de rede entre processos (valores aproximados)	70
4.11	Latência das mensagens Raft (s) em cada configuração de rede nos carregadores .	71
4.12	Débito das mensagens Raft (s) em cada configuração nos carregadores	71

Abreviaturas e Símbolos

BEV	Battery Electric Vehicles (Veículos Elétricos a Bateria)
CSMS	Charging Station Management System (Sistema de Gestão de Estações de Carregamento)
EVSE	Electric Vehicle Supply Equipment (Equipamento de Fornecimento de Veículo Elétrico)
IEC	International Electromechanical Commission (Comissão Eletromecânica Internacional)
MER	Máquina de Estado Replicáveis (protocolo de máquina de estado replicáveis)
OCPP	Open Charge Point Protocol (Protocolo Aberto de Ponto de Carregamento)
PHEV	Plug-in Electric Vehicles (Veículos Elétricos Plug-in)
Raft	Um protocolo de consenso utilizado em sistemas distribuídos
SAE	Society of Automotive Engineers (Sociedade de Engenheiros Automotivos)
SOC	State of Charge (Estado de Carga) de uma bateria
SOC	System on a Chip (Sistema em um Chip)
TWh	Tera Watt-hora (medida de energia em ordem de Tera)
TPS	Transações por segundo

Capítulo 1

Introdução

Veículos elétricos têm-se tornado a nova norma para mobilidade globalmente. Este crescimento só será possível a partir de uma boa cobertura de estações de carregamento. Segundo a Agência Internacional de Energia, IEA, o número de veículos elétricos de passageiros será de 125 milhões em 2030 e de 240 milhões em 2040. Esta previsão aponta para um aumento relevante de veículos eletrificados: as vendas globais do ano de 2022 superaram as do ano anterior em 10 milhões de veículos [4]. O esperado aumento significativo destes veículos nos próximos anos, previsto pela referida agência, causará um aumento no consumo de energia elétrica global para veículos elétricos (VEs), sendo possível atingir 11000 TWh, na avaliação APS, indicando que estes veículos terão um importante papel nos sistemas de potência [4].

O aumento do consumo de energia elétrica decorrente da crescente adoção de veículos elétricos (VEs) exige uma integração adequada destes à rede de distribuição. A ausência de gestão das sessões de carregamento pode levar à sobrecarga da rede, resultando em diversos problemas, como perdas elétricas, quedas de tensão, sobrecarga, circulação de correntes harmônicas, aumento da temperatura e envelhecimento prematuro dos transformadores, conforme apontado por Zheng et al. (2019) [46].

Para mitigar este impacto e garantir a máxima utilização dos recursos, é fundamental implementar sessões de carregamento coordenadas. Esta abordagem permite otimizar a utilização da infraestrutura de carregamento, evitando picos de procura e garantindo uma distribuição mais equilibrada da carga na rede elétrica. Além disso, a gestão inteligente do carregamento pode contribuir para a redução dos custos de energia e para a promoção da sustentabilidade do sistema elétrico, incentivando o uso de fontes renováveis e a participação dos VEs na gestão da procura.

A Figura 1.1 ilustra a relação entre o número de veículos elétricos em circulação e a quantidade de estações de carregamento públicas em diferentes regiões, com foco nos principais mercados de VEs: Estados Unidos da América (verde), Noruega (azul), Dinamarca (amarelo) e China (vermelho). Na sua maioria, verifica-se uma tendência atual de uma taxa inferior a 0,5 estações de carregamento por veículo elétrico de passageiros leve (BEV LDV). Esta proporção indica que, para cada ponto de carregamento público disponível, existem mais de dois veículos elétricos em circulação.

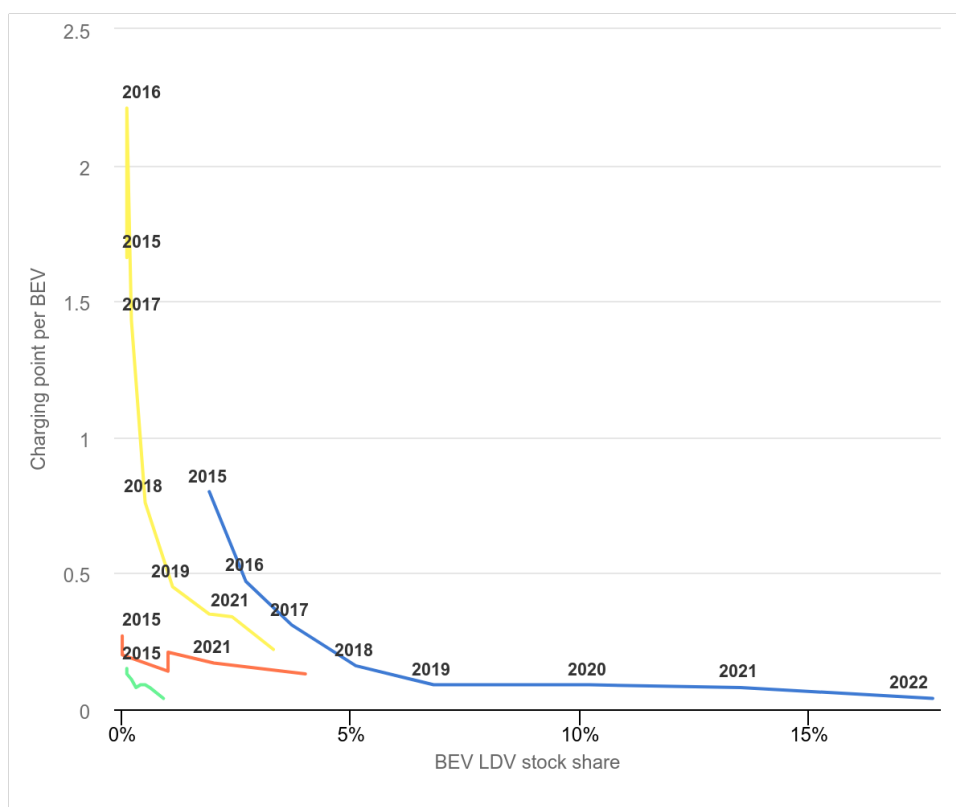
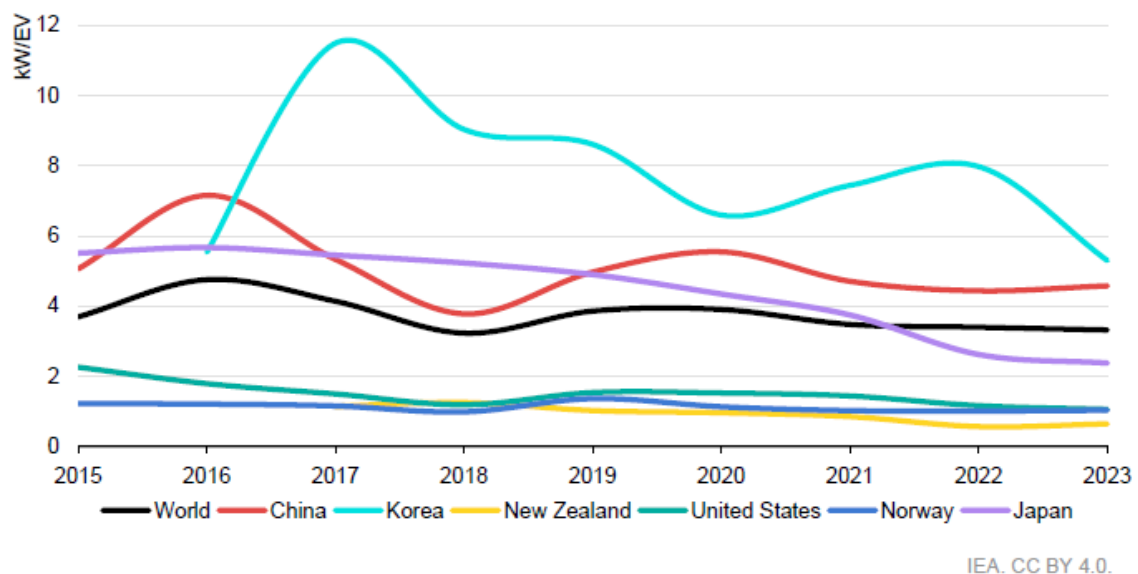


Figura 1.1: Estações de carregamento por veículos elétricos segundo relatório da agência internacional de energia (IEA) *Global EV Outlook* [5]



Notes: Kilowatts per EV are estimated assuming 11 kW for slow and 50 kW for fast chargers. Official national statistics, which benefit from access to more granular data, may differ from these values.
Sources: IEA analysis based on country submissions.

Figura 1.2: Potência disponível (kW) por VE segundo relatório da agência internacional de energia (IEA) *Global EV Outlook 2024* [6]

A implantação de carregadores de veículos elétricos deve ser coordenada com o desenvolvimento da rede elétrica para garantir novas ligações que estejam alinhadas com o planejamento de longo prazo da rede. Quando o carregamento não é gerido adequadamente, pode provocar picos de procura de energia. As estratégias de gestão de carregamento, como tarifas baseadas no tempo de utilização e o carregamento inteligente, tornar-se-ão mais necessárias à medida que a adoção de VEs continuar a crescer [6].

A Figura 1.2 mostra a evolução da potência disponível para veículos elétricos, considerando o número de estações de carregamento em cada região. Para este cálculo, considerou-se que as estações de carregamento lento têm uma potência nominal de 11 kW, enquanto as de carregamento rápido oferecem 50 kW. Em 2023, no cenário global, a média disponível foi inferior a 4 kW por veículo elétrico, destacando a necessidade de uma gestão eficiente da carga para otimizar os recursos disponíveis.

Segundo Sobrinho D. [38], diferentes estudos relacionam o impacto na inserção dos veículos elétricos na infraestrutura de carregamento, como o impacto em políticas e incentivos, em modelo de negócio, na previsão de utilização e gestão de baterias. A integração eficiente dos VEs na rede elétrica constitui um desafio complexo, envolvendo múltiplas variáveis e exigido soluções que considerem tanto as necessidades dos utilizadores como as restrições da infraestrutura. Diversos algoritmos têm sido propostos para auxiliar neste processo, cada um com objetivos específicos, como a proteção da rede, o fornecimento de serviços auxiliares ou a otimização do custo para o utilizador.

Algoritmos de natureza distribuída apresentam vantagens em termos de flexibilidade e escalabilidade em relação aos centralizados. No entanto, a escolha da melhor estratégia para implementação em infraestruturas de carregamento ainda carece de consenso. A definição e implementação de uma única estratégia limita a infraestrutura à otimização de um único algoritmo, restringindo a exploração de outras abordagens potencialmente mais eficientes em diferentes cenários.

Além dos problemas já mencionados, um estudo realizado por H.S. Das e colaboradores [14] em Berlim revelou uma distribuição desigual das estações de carregamento após a expansão da infraestrutura pública na cidade. A densidade de estações é maior no centro, resultando numa procura de carregamento desigual, com maiores taxas de ocupação dos veículos elétricos na periferia após o término do carregamento. O estudo propõe políticas para solucionar o problema e pode-se evidenciar a importância da gestão das sessões de carregamento para melhorar a disponibilidade das estações.

Diante deste contexto, o presente trabalho não se concentra na exploração de algoritmos específicos de otimização do agendamento de carregamento, mas sim na proposição de uma plataforma que considera as normas da indústria atual. Flexível a diferentes casos de uso, proporcionando maior liberdade e adaptabilidade às entidades gestoras.

Para permitir uma descentralização das tarefas executadas mediante entidades centrais é preciso garantir a capacidade computacional e de comunicação com agentes periféricos, como os carregadores. As tarefas executadas por um agente central podem envolver autenticação, identificação, controlo das sessões de carregamento de forma dinâmica ou por agendamento, ou gestão de pagamento das sessões, como especificado na norma ISO 15118. Todas as tarefas descritas exigem uma infraestrutura de comunicação bem estabelecida, o que é um problema difícil de ser ultrapassado como apresentado por Lopes-Behar em [21]. Entende-se assim que para o avanço das infraestruturas de carregamento acontecer, o protocolo de comunicação tem um papel essencial.

Diferentes casos de uso para carregamentos inteligentes foram definidos como, por exemplo, em [18], e os pilotos envolvidos nesse trabalho definiram dois tipos de regras que guiam os casos de uso, carregamento inteligente guiado pelo preço das tarifas e os compatíveis com a rede elétrica. Isso agrupa as tarefas citadas acima que podem ser executadas com diferentes propósitos, mas devem respeitar as restrições dinâmicas da rede elétrica, garantir a disponibilidade de recursos aos utilizadores, a privacidade dos utilizadores e das centrais de operação, pode-se ainda ter que cumprir outros requisitos como minimização de custos ou maximização do uso de energia de fontes renováveis.

Dentre as tarefas acima citadas a que exige maior trabalho é a do controlo das sessões de carregamento, pois é um problema que depende de múltiplas variáveis e que possui um efeito em escala, pois deve considerar dados globais e locais e ainda enviar comandos às unidades de carregamento de forma individual.

1.1 Contextualização e Motivação

Dado o aumento de veículos elétricos no mercado global, cria-se a necessidade de uma estrutura de estações de carregamento que suporte a crescente procura inerente a estes veículos. Essas, por sua vez, necessitam ser geridas e para isso trocam informações com servidores centrais. Estes servidores são dedicados a gerir as estações de carregamento, compreendendo em si a função de receber e manipular as informações recebidas com outros dados provenientes do contexto global.

Neste sentido, o constante crescimento observado nos últimos anos e o expectável para os próximos, geram a eminente sobrecarga destes servidores devido à quantidade de informações recebidas. A consequência evidente será a falta de resposta em tempo útil, promovendo indisponibilidade das tais estações para os seus clientes/utilizadores.

Por conseguinte, a própria capacidade computacional encontrada da conjuntura atual nas estações de carregamento, estratégias de gestão distribuída tornam necessária a sua exploração. Essas estratégias procuram contribuir para o alívio da carga computacional e de comunicação nos servidores, através da inclusão dos carregadores como dispositivos capazes de realizar a gestão das suas próprias sessões de carregamento e a de outros carregadores locais, assim resultando numa solução mais flexível para o futuro da gestão de estações de carregamento e consequentemente o futuro da mobilidade elétrica.

Esta investigação tem como objetivo principal investigar estratégias de computação modular e comunicação distribuída, visando um controlo local e periférico de carregadores de veículos elétricos. Pretende-se, desta forma, desenvolver uma plataforma que auxilie na resolução dos desafios inerentes à gestão de redes de carregamento, facilitando a operação das entidades gestoras deste setor. Adicionalmente, estas estratégias poderão conferir maior tolerância a falhas e garantir uma maior disponibilidade das infraestruturas de carregamento para o utilizador, para além de proporcionarem uma gestão mais flexível dos recursos energéticos.

A motivação subjacente a este trabalho surgiu durante o desenvolvimento de unidades de carregamento lento para a infraestrutura de carregamento do INESC TEC. Observou-se que os recursos computacionais dessas unidades de carregamento eram subutilizados em relação à sua capacidade total. Em resposta a esta constatação, propõe-se que tarefas tradicionalmente executadas por agentes centrais sejam, doravante, processadas localmente pelas próprias unidades de carregamento. Este redirecionamento visa otimizar a utilização dos recursos computacionais disponíveis nos equipamentos de carregamento de veículos elétricos, contribuindo, de forma indireta, para a resolução de desafios de larga escala previamente identificados no setor da mobilidade elétrica, com um enfoque particular na infraestrutura de sistemas de carregamento.

Atualmente, diversos trabalhos propõem a migração da carga computacional para dispositivos periféricos de carregamento, implementando algoritmos específicos. Estes recebem dados de entrada variados, como o estado de carga (SOC) inicial e final, tempo de partida planeado, dados históricos de utilizadores, dados da infraestrutura elétrica, dados de dispositivos de medição de energia locais, entre outros. Adicionalmente, os objetivos do controlo das sessões de carregamento podem ser diversos, orientados para o utilizador, a rede ou o preço. Os dados de saída de

cada algoritmo apresentam formatos e significados variados, sendo interpretados de forma distinta por cada sistema, resultando numa multiplicidade de configurações e parâmetros.

Sistemas descentralizados dependem de mecanismos de consenso para garantir a confiabilidade e estabilidade da rede. Estes mecanismos permitem que os diferentes nós da rede cheguem a um acordo sobre o estado do sistema, mesmo na presença de falhas ou comportamentos maliciosos. A escolha do mecanismo de consenso adequado é crucial para o desempenho e segurança do sistema, como definido pela fundação Hyperledger[16]. A tecnologia *blockchain*, por exemplo, utiliza mecanismos de consenso para validar e adicionar novos blocos à cadeia, garantindo a integridade dos dados e a imutabilidade do histórico de transações. Este trabalho analisa a aplicabilidade de diferentes mecanismos de consenso em infraestruturas de carregamento de veículos elétricos.

O presente trabalho visa explorar a capacidade computacional dos carregadores de veículos elétricos (VEs), integrando-os no âmbito de sistemas embarcados, como a Internet das Coisas (IoT), e experimentando a utilização de uma rede de consenso. Para tal, será avaliada a aplicabilidade de um mecanismo de consenso nestes dispositivos, incluindo testes de desempenho e a medição de indicadores-chave de desempenho (KPIs) relacionados à comunicação e ao desempenho computacional.

Sabendo que é necessário gerir as sessões de carregamento e controlá-las, diferentes topologias para interação entre os atores envolvidos no sistema de carregamento são propostas, sendo possível serem classificadas em topologias centralizadas, descentralizadas, distribuídas.

A arquitetura centralizada exige requisitos de comunicação e computação elevados num único agente, reduz a privacidade ao concentrar os dados, pode enfrentar sérios problemas quando se fala em escalabilidade para além de ter um único ponto de processamento e armazenamento que pode comprometer todo o sistema em caso de falhas, mas como vantagem produz geralmente resultados otimizados por possuir todas as informações do sistema.

Face a essas características dos sistemas centralizados, esse trabalho visa propor uma estratégia alternativa por meio de uma experiência utilizando um algoritmo descentralizado, como resultado espera-se apresentar um demonstrador aplicado a um caso real de uso, como referência para o trabalho realizado as características desejáveis para o algoritmo é que o mesmo seja flexível, eficiente e robusto.

1.2 Objetivos

O uso de algoritmos de consenso para diferentes aplicações vem sendo um assunto amplamente estudado nos últimos tempos, sendo a utilização mais conhecida em redes *blockchain*, esse trabalho visa realizar uma aplicação num ambiente real de um protocolo de máquina de estado replicáveis (MER), como explicado por [37], a base destes protocolos estão os algoritmos de consenso, utilizados para manter a consistência entre os nós, em suma, a aplicação de um algoritmo num ambiente ainda inexplorado.

Diversas estratégias distribuídas têm sido propostas para otimizar a gestão de redes de carregamento de veículos elétricos. Contudo, levanta-se a questão de saber se os dispositivos atualmente utilizados nas unidades de carregamento possuem a capacidade computacional e de comunicação necessária para lidar com as exigências impostas por estes algoritmos.

Este trabalho visa validar uma rede de comunicação distribuída, baseada em mecanismos de consenso, através da análise de desempenho numa infraestrutura real de carregamento de veículos elétricos. A proposta inovadora reside na utilização das próprias unidades de carregamento como gestoras autónomas, explorando a capacidade computacional limitada de cada unidade para estabelecer e manter o consenso distribuído entre os módulos de gestão. Esta abordagem descentralizada visa otimizar a gestão da rede de carregamento, conferindo maior autonomia e flexibilidade ao sistema. A validação da rede de comunicação será realizada por meio de testes de desempenho em ambiente real, considerando as restrições computacionais das unidades de carregamento.

Para o objetivo principal ser cumprido são definidos os seguintes objetivos específicos:

1. Estudo de mecanismos de consenso e estratégias utilizadas para selecionar o mecanismo de consenso a utilizar.
2. Pesquisa e análise de bibliotecas que implementem corretamente o algoritmo de consenso selecionado.
3. Implementação em unidades de carregamento reais.
4. Especificação de testes a serem executados e respetivas métricas de avaliação.
5. Análise de resultados e análise da viabilidade de implementação do mecanismo de consenso em unidades de carregamento reais.

O presente trabalho visa explorar a capacidade computacional dos carregadores de veículos elétricos (VEs), integrando-os no âmbito de sistemas embarcados, como a Internet das Coisas (IoT), e validando a utilização de uma rede de consenso nesses dispositivos. Posteriormente, o desempenho da rede de consenso será analisado utilizando o método de análise de desempenho de sistemas distribuídos apresentado em [36].

O valor esperado com esse trabalho é uma análise de uma rede de consenso num ambiente real e uma especificação de parâmetros mínimos para o bom funcionamento de uma rede de consenso em sistemas embarcados periféricos.

A rede de comunicação proposta baseia-se em algoritmos de consenso com eleição de líder, permitindo a comunicação entre os nós, procura garantir a flexibilidade energética através da limitação a valores máximos de potência, e tendo a capacidade em cada carregador de calcular o agendamento de carregamento da sessão em que se encontra.

1.3 Ambiente de realização da dissertação

O Laboratório de Redes Elétricas Inteligentes e Veículos Elétricos (SGEVL) do INESC TEC, situado no Porto, Portugal, é um centro em pesquisa e desenvolvimento na área de sistemas de energia e mobilidade elétrica [17]. O laboratório oferece uma infraestrutura completa para a realização de testes e validação de tecnologias inovadoras, incluindo redes elétricas inteligentes, carregamento de veículos elétricos, armazenamento de energia e integração de energias renováveis.

O SGEVL possui uma equipa multidisciplinar de investigadores, engenheiros e estudantes, com vasta experiência nos diversos temas relacionados ao setor energético. O laboratório colabora com empresas e instituições, tanto nacionalmente como internacional, promovendo a transferência de tecnologia e a inovação no setor energético [17].

No âmbito deste trabalho, o SGEVL proporcionou o ambiente adequado para a experimentação e validação da utilização de uma rede de consenso em carregadores de veículos elétricos, permitindo a realização de testes em condições reais e a avaliação do desempenho do sistema proposto.

1.4 Estrutura da dissertação

Após o capítulo introdutório, o presente documento prossegue com o capítulo dedicado ao estado da arte, abordando a fundamentação teórica, os tipos de veículos elétricos, os tipos de carregadores existentes, a infraestrutura de carregamento e os algoritmos relevantes.

O capítulo 3 apresenta uma descrição detalhada do sistema proposto, focando na comunicação entre os dispositivos e na função de cada módulo constituinte do sistema. Adicionalmente, será apresentado o caso de uso implementado para o carregamento de veículos elétricos.

O capítulo 4 descreve as unidades de carregamento utilizadas nos experimentos reais, bem como a arquitetura de testes desenvolvida para comparar os resultados obtidos.

Por fim, o capítulo 5 apresenta as conclusões deste trabalho e as perspetivas futuras de desenvolvimento, com base nos resultados alcançados.

Capítulo 2

Estado da Arte

Neste capítulo, será apresentado a fundamentação teórica e o estado da arte dos sistemas de gestão de carregamento de veículos elétricos. Discutiremos os principais atores que compõem a infraestrutura de carregamento, os métodos propostos para o carregamento coordenado, e as normas vigentes na indústria.

Além disso, exploraremos os sistemas distribuídos e as técnicas mais comuns utilizadas, uma vez que este trabalho propõe a adição de uma camada distribuída ao gestor das estações de carregamento. Cabe destacar que o conceito de distribuição mencionado neste contexto não se refere às topologias distribuídas em sistemas de potência, mas sim à comunicação distribuída entre dispositivos computacionais.

2.1 Infraestrutura de carregamento

Sistemas relacionados à mobilidade elétrica são intrinsecamente heterogêneos e escaláveis, como apontado por Cruz e Dario em [9]. Consequentemente, a infraestrutura de carregamento deve ir além do simples fornecimento de energia, desempenhando um papel crucial no suporte a funções essenciais e auxiliares para a rede, o veículo e o utilizador. Nesse contexto, os intervenientes nas sessões de carregamento podem ser agrupados em quatro categorias principais: o veículo elétrico, o carregador, o gestor de carregamentos e os agentes externos.

2.1.1 Tipos de veículos elétricos

Os veículos elétricos (VEs), apesar de existirem há bastante tempo, têm conquistado um espaço cada vez maior nas estradas nos últimos anos [12]. Ao contrário dos veículos com motor de combustão interna (ICE), que dependem de combustíveis líquidos (gasolina ou gásóleo), os VEs são impulsionados por energia elétrica, representando uma alternativa mais sustentável e eficiente para a mobilidade.

Há vários tipos de veículos elétricos disponíveis no mercado atualmente, mas o foco deste trabalho é nos veículos que possuem conexão por tomada, conhecidos como veículos elétricos *plug-in*. Estes podem ser categorizados em dois tipos principais [9]:

- **Veículos elétricos a bateria (BEV):** São completamente tracionados por motores elétricos alimentados por uma bateria elétrica. Utilizam a eletricidade para carregar a bateria, dessa forma a energia elétrica é fornecida aos motores pela bateria que converte em energia mecânica para tracionar o veículo, proporcionando o movimento do veículo, com emissão zero no processo descrito e sem ruído. O sistema de Travagem Regenerativa (TR), que no movimento passivo da roda permite o motor agir como gerador e fornecer energia para a bateria pela inércia, ajuda a armazenar energia em situações de travagem ou redução da velocidade [9].
- **Veículos Híbrido-Elétricos Plug-In (PHEV):** Os *Plug-in Hybrid Electric Vehicles* (PHEVs) utilizam duas fontes distintas de energia. A propulsão e movimento desses veículos podem ser gerados tanto por um motor elétrico quanto por um motor de combustão, ambos instalados no veículo. Dessa forma, podemos dizer que a bateria permite que o mesmo tenha um alcance variável, dependendo da sua capacidade, além do depósito de combustível fóssil. PHEVs também podem instalados com o sistema TR. Esses veículos possuem uma bateria relativamente menor do que os BEVs citados anteriormente [9].

2.1.2 Tipos de carregadores

Existem diferentes tipos de carregadores de veículos elétricos, classificados conforme o tipo de energia elétrica transferida, os protocolos de comunicação que utilizam e o contexto em que são instalados.

Um carregador pode fornecer tanto corrente contínua (DC) quanto corrente alternada (AC). Além disso, pode implementar vários protocolos de comunicação, como CHAdeMO, Tesla Supercharger (SC), IEC 61851-1 ou ISO 15118. As normas que definem as tomadas de conexão variam, assim como a aplicação do carregador, que pode ser destinada a ambientes residenciais ou comerciais, resultando em diferentes características visuais e funcionalidades personalizadas conforme o uso.

Além das configurações mencionadas, os carregadores podem ser classificados pelos modos de operação, que se diferenciam pelos níveis de potência fornecida ao veículo. Estes modos são classificados progressivamente em modos 1, 2 e 3, cada um oferecendo um nível crescente de potência.

A Figura 2.1 fornece uma visão abrangente dos tipos de carregadores disponíveis e os diferentes contextos em que são utilizados.

Os veículos que utilizam baterias elétricas para armazenamento de energia, conforme mencionado na seção anterior, podem ser recarregados externamente utilizando a rede elétrica mediante um dispositivo de carregamento, geralmente denominado Electric Vehicle Supply Equipment (EVSE). Um carregador (EVSE) é essencial no processo de recarga da bateria de um veículo elétrico, pois atua como intermediário no processo: a rede elétrica fornece energia em corrente alternada (CA), enquanto a bateria dos veículos elétricos armazena energia em corrente contínua (CC) [44].

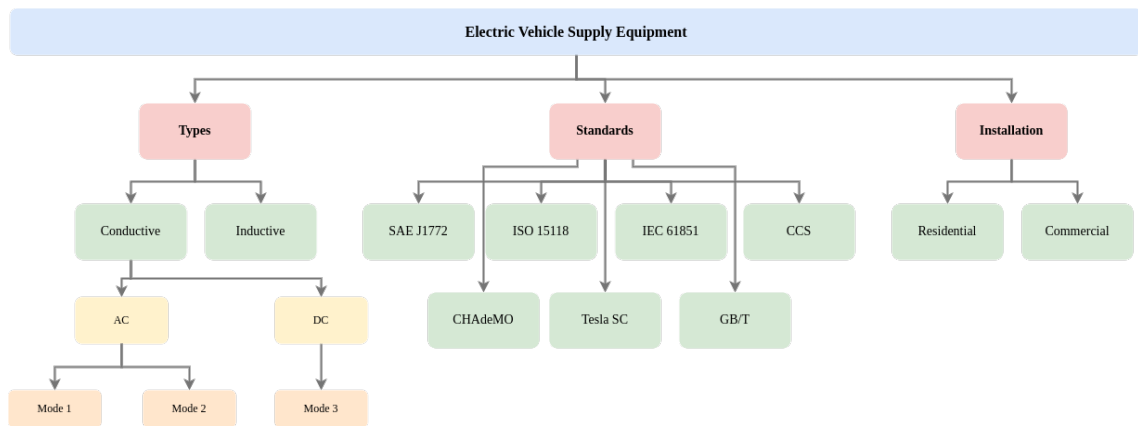
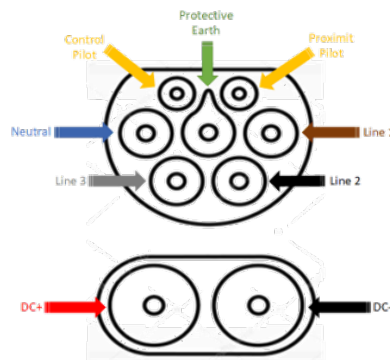


Figura 2.1: Modos de carregamento

Existem diferentes tipos de carregadores de veículos elétricos disponíveis, incluindo carregamento indutivo e condutivo. Esses carregadores desempenham um papel crucial na integração dos veículos elétricos na infraestrutura de recarga, garantindo a transferência eficiente de energia.

Figura 2.2: Tomada de conexão *Combined Charging System* (CCS)

Já discutimos os tipos de veículos elétricos e apresentamos os diversos tipos de carregadores que podem fornecer energia a esses veículos. No entanto, é igualmente importante compreender como esses agentes se interligam e a dinâmica de comunicação e interação entre eles. As próximas seções abordarão essa interconexão, trazendo clareza e compreensão sobre o funcionamento integrado dos sistemas de carregamento de veículos elétricos.

2.1.3 Agregadores

O conceito de agregador é amplo e abordado de diferentes perspectivas na literatura. Contudo, neste trabalho, o foco recairá sobre os agregadores que servem como interface entre a rede e os veículos elétricos (VEs). Estes agregadores recebem constantemente informações dos VEs por meio dos carregadores ou de medidores inteligentes de energia. Assim, podem determinar as necessidades de carregamento dos VEs e agendar o carregamento com base nas informações adquiridas através da comunicação com o operador da rede [10].

Os três principais atores envolvidos na infraestrutura de carregamento são os VEs, os EVSEs e o gestor dos EVSEs (agregador). Os dois primeiros foram abordados nas Secções 2.1.2 e 2.1.1. Nesta secção, analisaremos a entidade responsável pela gestão das sessões de carregamento, o agregador, sendo o componente principal no contexto deste trabalho.

O controlo dinâmico dos postos de carregamento é um processo essencial. Para tal, existem os agregadores, que são atores que comunicam diretamente com um carregador ou com um grupo de carregadores para verificar o estado atual em que se encontram. Adicionalmente, podem possuir informações sobre o contexto global da rede elétrica, obtidas através da comunicação com os operadores de rede. Conforme discutido no estudo de Nimalsiri et al. [26], o agregador atua como uma espécie de proxy entre os VEs e a rede elétrica, gerindo as interações com o mercado energético de maneira inteligente.

Em conjunto com o ORD (Operador do Sistema de Distribuição), o agregador pode prever a demanda de energia para o dia seguinte e preparar os preços de compra/venda. Pressupõe-se que o agregador de VEs possua informações sobre o preço da eletricidade e a localização das estações de carregamento, as quais são disponibilizadas aos proprietários de VEs [10].

Além da previsão dos preços de mercado, o agregador também enfrenta incertezas relacionadas aos comportamentos e preferências dos proprietários de VEs. As principais fontes de incerteza incluem os horários de partida/chegada, as distâncias percorridas e as preferências de carregamento, como, por exemplo, quando e quanto carregar a bateria [10].

Para além dessas funções, sendo a interface entre a rede e os VEs, o agregador, executa também a autenticação e identificação dos veículos elétricos através do acesso das devidas plataformas. Isto permite habilitar o processo de negociação de energia, no qual o veículo apresenta as suas necessidades energéticas e os seus limites de carregamento. O agregador, então, calcula um agendamento de carregamento para o veículo com base num algoritmo de otimização utilizando os dados disponíveis. Se o VE aceitar o agendamento, inicia-se a transferência de energia; caso contrário, o processo de negociação realiza uma nova iteração [2].

2.1.4 Normas relacionadas ao carregamento de veículos elétricos

A análise dos padrões de carregamento apresentada por Sobrinho D. [38] examina os três maiores mercados de veículos elétricos: americano, europeu e asiático. No mercado americano, o foco recai predominantemente sobre os Estados Unidos. A análise evidencia a predominância de diferentes padrões de carregamento em cada região, destacando-se SAE J1772, Mennekes, CHAdeMO, GB/T e Tesla.

Dada a variabilidade das aplicações existentes e a característica previamente mencionada de heterogeneidade dos sistemas de carregamento, torna-se essencial o estabelecimento de normas para a interoperabilidade entre equipamentos [9]. Estas normas englobam VEs, estações de carregamento, gestores das estações de carregamento e a rede elétrica na totalidade.

A Figura 2.3, de autoria própria e baseada nas normas vigentes, apresenta os protocolos mais relevantes e amplamente utilizados na atualidade, os quais compõem a infraestrutura de carregamento dos veículos elétricos.

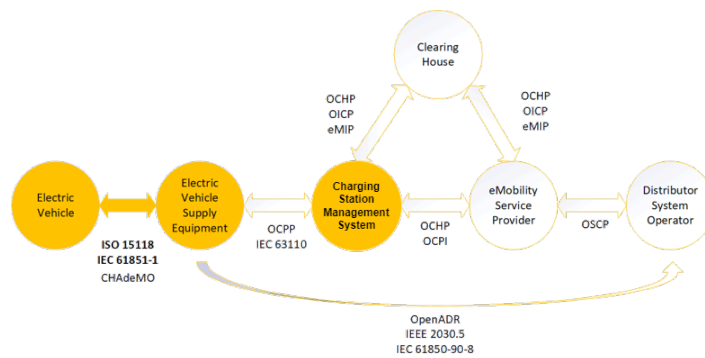


Figura 2.3: Infraestrutura de carregamento com atores e normas de comunicação

Os principais atores no processo de integração dos veículos elétricos na rede são os próprios veículos e, diretamente relacionados a estes, os carregadores, conforme indicado pelo diagrama através da diferenciação de cores.

2.1.4.1 Norma IEC 61851-1

Geralmente, a comunicação entre o veículo e o carregador é realizada por meio de uma comunicação unidirecional, conforme estabelecido pela norma IEC 61851-1 [40]. De acordo com esta norma, o sinal de comunicação é transmitido através de um cabo de secção reduzida, integrado no cabo de carregamento. Este arranjo permite que o sinal seja emitido pelo carregador e recebido pelo veículo, estabelecendo o circuito através do cabo de proteção (PE). Este sinal de comunicação é designado por *Control Pilot*. As características do sinal são as seguintes:

- Amplitudes máximas: $\pm 12V$
- Frequência: 1kHz
- Tipo: PWM (Pulse Width Modulation)

Abaixo segue um diagrama que exemplifica o sinal, a temporização aceite pela norma e o seu funcionamento básico:

No polo negativo, o sinal deve manter sempre uma amplitude de pico de -12V. No polo positivo, o valor pode variar em função dos diferentes estados do processo de carregamento, sendo estas mudanças determinadas pelo veículo elétrico.

Quando o veículo é conectado ao carregador, o circuito formado entre o cabo *Control Pilot* (CP) e o cabo de proteção (PE) cria um divisor de tensão, o que reduz a amplitude positiva do sinal de +12V para +9V. Quando o veículo está pronto para iniciar o carregamento e o sinal do CP apresenta um ciclo de trabalho (*duty cycle*) entre 10% e 96%, o veículo fecha o interruptor interno (S2). Esta comutação altera a amplitude positiva do sinal para +6V no caso de carregamento sem ventilação interna, ou para +3V no caso de carregamento com ventilação interna. Esta alteração na amplitude sinaliza ao carregador para modificar o estado do relé, permitindo, assim, a transferência de energia.

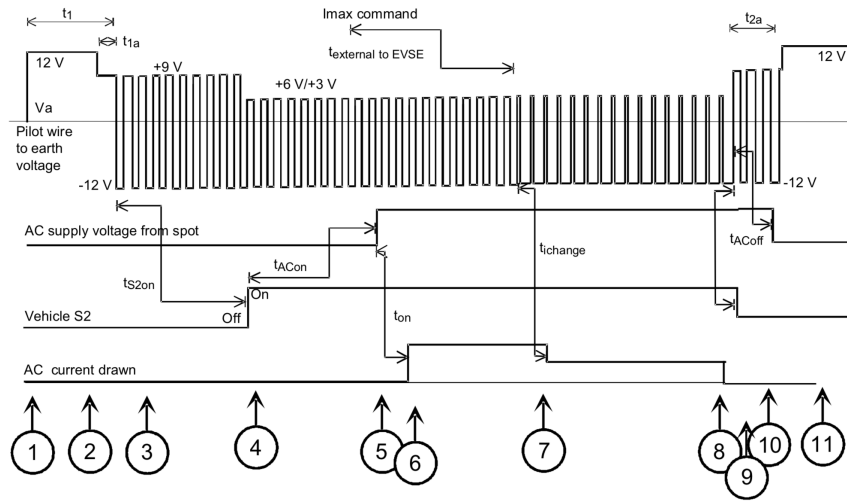


Figura 2.4: Sinal PWM enviado do Carregador para o VE [40]

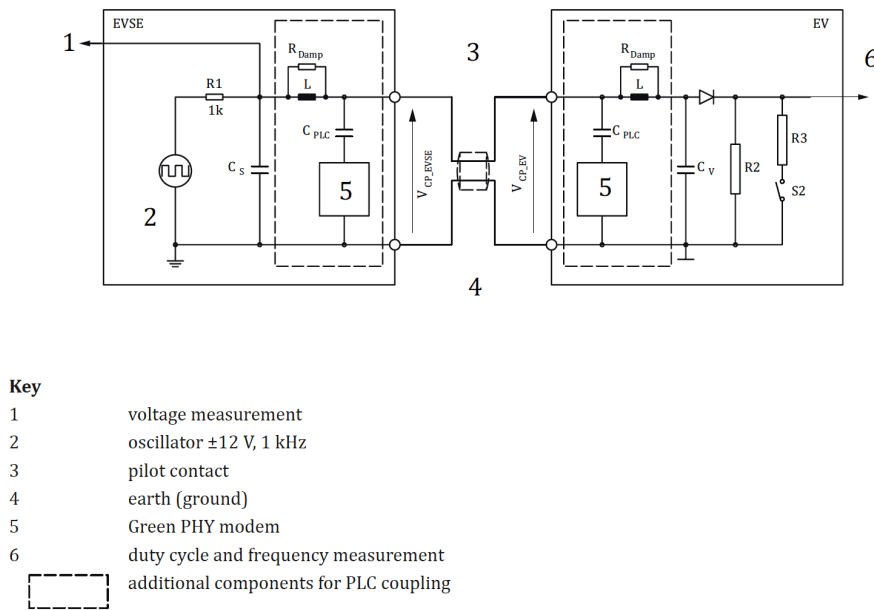


Figura 2.5: Circuitos internos do EVSE e VE [40]

2.1.4.2 Norma ISO 15118

A comunicação entre o VE e o carregador é uma etapa essencial para a infraestrutura de carregamento. Para ser possível planejar uma sessão de carregamento, é necessário obter informações tanto do veículo como do utilizador. O planeamento das sessões de carregamento pode possibilitar a utilização dos veículos como recursos energéticos para a rede elétrica. A norma ISO 15118 [1], que inclui uma série de documentos, especifica os requisitos para a comunicação digital entre o veículo elétrico e o carregador. Esta norma permite o uso de diversas ferramentas inovadoras

na comunicação entre o VE e o carregador e serve de base para a função Plug&Charge. Esta função abrange a identificação e autenticação do utilizador, o processo de cobrança da sessão de carregamento, bem como o fornecimento de informações técnicas do veículo para a negociação da transferência de energia durante a sessão de carregamento atual. Além disso, a norma possibilita a utilização de *smart contracts*, aumentando a segurança e a conveniência para o utilizador. Estas ferramentas visam simplificar o processo de carregamento para o utilizador. Além das necessidades do utilizador, um dos principais recursos desta norma é a capacidade de habilitar a tecnologia Vehicle-to-Grid (V2G) no processo de carregamento. Esta tecnologia permite a transferência de energia do veículo para a rede elétrica.

Na Figura 2.6, é possível observar a relação entre a divisão da ISO 15118 e as 7 camadas do modelo Open System Interconnection (OSI), de modo a estabelecer a comunicação definida pela norma, uma comunicação digital entre o VE e o EVSE (carregador).

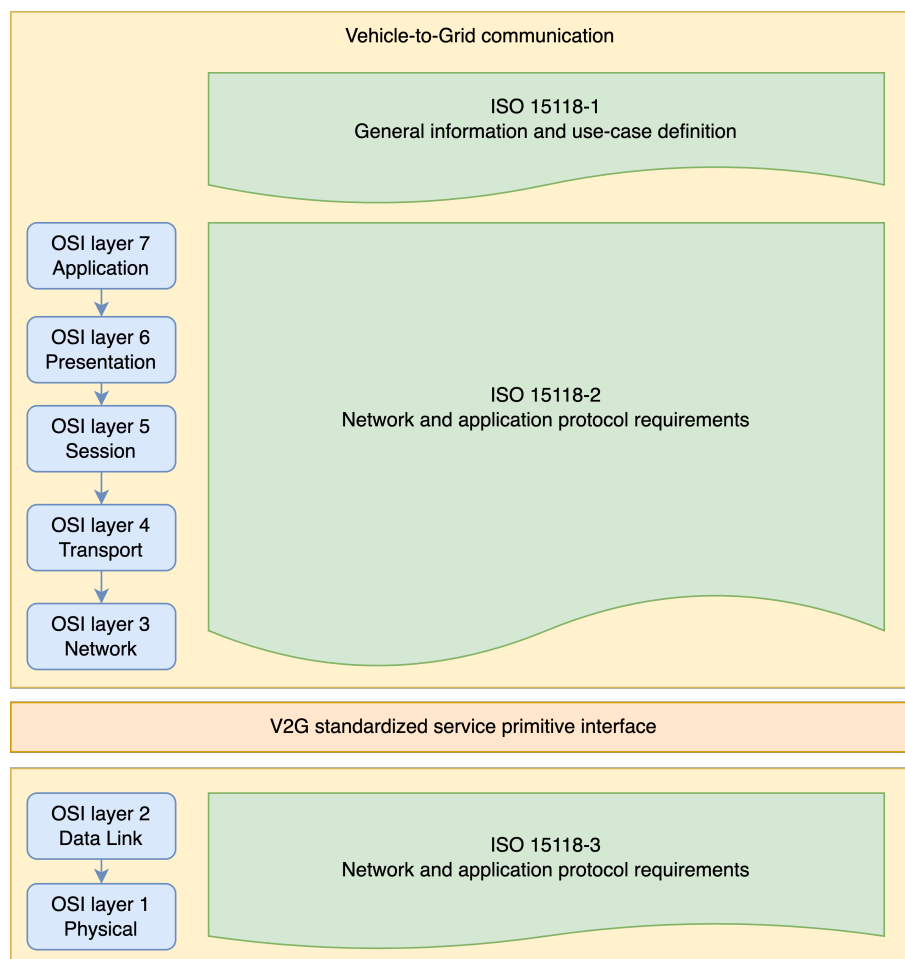


Figura 2.6: Documentos da ISO 15118 relacionados às camadas OSI

Conforme discutido por Santos, Joni e colaboradores em [35], cada camada do modelo OSI é construída baseada na camada subjacente. A primeira camada, a camada física, define o meio físico e a força do sinal. O método de comunicação mais utilizado e definido pela norma de

comunicação é a tecnologia de comunicação em *Power Line Communication* (PLC), denominada *HomePlugGreen PHY* (HPGP), possui banda mais larga, comunicação e alta velocidade e um menor impacto no ambiente em comparação com tecnologias semelhantes.

A camada de ligação de dados especifica as características do sinal PLC, as principais delas são o *Signal Level Attenuation Characterization* (SLAC), que determina se o sinal foi recebido ou através de medir a intensidade do sinal e o *Service Access Point* (SAP), que é utilizado para fornecer sinais entre as camadas do modelo OSI.

Na camada de rede os endereços IPs são trocados para permitir a comunicação IP. A versão do protocolo IP foi definido na norma como sendo IPv6 o protocolo por defeito e opcional o uso do IPv4.

Na camada de transporte, a comunicação é realizada através do *User Datagram Protocol* (UDP) e do *Transmission Control Protocol* (TCP). Inicialmente utiliza-se a comunicação UDP para o mecanismo *SECC Discovery Protocol* (SDP) específico da norma ISO 15118, o UDP é utilizado nesta etapa, pois essa não necessita de estabelecimento de uma conexão para envio de dados. Após esse processo as mensagens de alto nível definidas pela norma são trocadas utilizando TCP ou de forma opcional acrescenta-se uma camada de segurança e utiliza-se o *Transport Layer Security* (TLS).

Na camada de sessão é utilizado o *Vehicle-to-Grid Transfer Protocol* (V2GTP) para definir a estrutura de dados de comunicação, a estrutura do cabeçalho e o endereço da porta, assim como é usado para definir o formato de mensagem na camada seguinte, em termos práticos o V2GTP é inserido no header das mensagens de alto nível, como forma de controlo e padronização na troca de mensagens.

Na camada de apresentação é utilizado o *Efficient XML Interchange* (EXI), que reduz o encargo de comunicação do *Extensive Mark-UP Language* (XML) padrão, para além de adicionar a codificação das mensagens em formato XML.

E na última camada, de aplicação, as mensagens de alto nível com informações relevantes para o processo são populadas com dados coerentes a medida que as mensagens são trocadas, possibilitando a comunicação entre os módulos *Electric Vehicle Communication Controller* (EVCC) do lado dos VEs e do *Supply Equipment Communication Controller* (SECC) do lado dos carregadores ou EVSEs.

No diagrama de sequência representado pela Figura 2.7 é demonstrado um caso de uso real de troca de mensagens desde o início até o fim de uma sessão de carregamento:

A norma ISO 15118, embora promissora, ainda não está amplamente disponível em veículos até o momento deste trabalho, devido ao seu carácter recente e à falta de consenso no mercado, como apontado por Santos e Joni [35]. No entanto, o estado do carregamento do veículo elétrico é expresso pela sinalização do *Control Pilot* (CP), conforme definido pela norma IEC 61851, e não pelo sinal PLC ou pelas mensagens ISO 15118-2.

Deste modo, o protocolo definido pela IEC 61851 deve ser implementado em todos os veículos e carregadores que utilizem os sistemas de tomada tipo 2 e tipo 1, que são os mais comuns

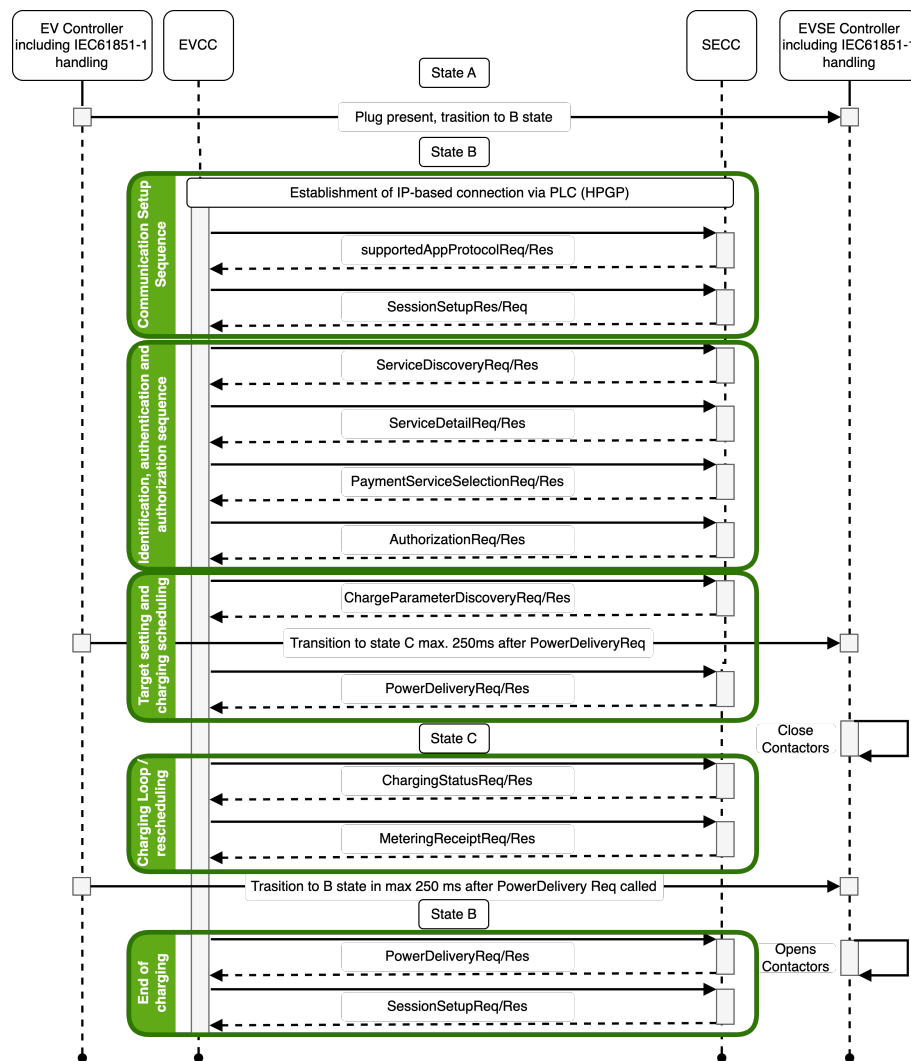


Figura 2.7: Exemplo de comunicação digital entre VE e EVSE baseado na norma ISO 15118-2 e IEC 61851-1

no contexto europeu. O diagrama apresentado na Figura 2.7 ilustra que, mesmo com a mensagens trocadas pela comunicação digital estabelecida pela ISO 15118, ainda é necessário manter o sinal definido pela IEC 61851. Assim, esses protocolos de comunicação são complementares e coexistem no processo de carregamento.

2.1.4.3 Open Charge Point Protocol - OCPP

Atualmente, observa-se um grande número de fabricantes de estações de carregamento, o que, por sua vez, levanta a questão da interoperabilidade. Esta se refere à capacidade de cada estação de carregamento e veículo de integrar-se à infraestrutura com o mínimo de esforço e custo. Nesse contexto, o Open Charge Point Protocol (OCPP) desempenha um papel crucial. Este protocolo facilita a transição entre dispositivos com menor esforço, sendo comparável à simplicidade da troca de um cartão SIM num telemóvel, conforme discutido por Venkata e colaboradores [41].

Em outras palavras, o OCPP é a norma de facto suportada pela indústria para a comunicação entre as estações de carregamento e os sistemas de gestão dessas estações, conhecidos como Charging Station Management Systems (CSMS). O protocolo foi formulado para acomodar qualquer técnica de carregamento e é uma norma aberta, sem barreiras de licença para sua utilização [2].

O estudo realizado por Raboaca e colaboradores [31], procura preencher o vazio no que se refere a literatura relacionada a norma OCPP. Apresentando os problemas na implementação, uma avaliação do desempenho do protocolo e uma aplicação real utilizando esse protocolo. Apresenta como vantagens da implementação do protocolo baseado na literatura atual como o aumento da estabilidade e integridade do sistema, oferecer ao utilizador serviços customizados e flexibilidade, a redução de investimentos e custos na infraestrutura, uma implementação eficiente de controlo da carga local [34].

As infraestruturas de carregamento podem ter um papel importante na exploração de tecnologias para *smart grids*, um ecossistema mais amplo que envolve diversos atores, dispositivos e protocolos. Neste ecossistema, normas de comunicação abertas são fundamentais para um desenvolvimento mais rápido e sustentável. O OCPP, criado pela Open Charge Alliance (OCA), é compatível com os protocolos OpenADR e Open Smart Charging Protocol (OSCP), além de uma série de documentos da norma ISO 15118. O documento original da norma é dividido em 6 partes, sendo elas descritas na Tabela 2.1

Tabela 2.1: Documentos que constituem a documentação do protocolo OCPP v2.0.1

Documento	Descrição
Parte 0	Introdução
Parte 1	Arquitetura e Topologia
Parte 2	Especificação: Casos de uso e requisitos, mensagens, tipos de dados e variáveis
Parte 3	
Parte 4	
Parte 5	
Parte 6	Casos de teste

Na Tabela 2.1 é apresentado a divisão de documentos da norma OCPP, a Parte 2 apresenta a descrição detalhada das transações realizadas e foi o documento mais consultado para este trabalho. Este documento define os casos de uso, a interação entre os diferentes atores envolvidos em cada caso e, adicionalmente, os pré-requisitos, a sequência de eventos e os requisitos posteriores associados a cada caso. Estes requisitos indicam o estado do sistema após a conclusão bem-sucedida da interação.

2.2 Parametrização da infraestrutura de carregamento

Nesta secção serão apresentados os requisitos funcionais e não funcionais relacionados a infraestrutura de carregamento como forma de enquadrar os ensaios realizados para que os resultados

possam ser comparados conforme as necessidades da infraestrutura de carregamento, cumprindo requisitos temporais, de desempenho e de funcionamento.

Para as unidades de carregamento no que se refere a comunicação com os VEs os requisitos da norma IEC 61851-1 e ISO 15118 são apresentados na Tabela 2.2.

Tabela 2.2: Especificação dos requisitos temporais de desempenho do sistema

Requisito	Especificação
ISO 15118-2 message SupportedAppProtocol	2s
ISO 15118-2 message SessionSetup	2s
ISO 15118-2 message ServiceDiscovery	2s
ISO 15118-2 message PaymentServiceSelection	2s
ISO 15118-2 message Authorization	2s
ISO 15118-2 message ChargeParameterDiscovery	2s
ISO 15118-2 message ChargingStatus	2s
ISO 15118-2 message MeteringReceipt	2s
ISO 15118-2 message CableCheck	2s
ISO 15118-2 message PreCharge	2s
ISO 15118-2 message WeldingDetection	2s
ISO 15118-2 message SessionStop	2s
ISO 15118-2 message ServiceDetail	5s
ISO 15118-2 message PaymentDetails	5s
ISO 15118-2 message PowerDelivery	5s
ISO 15118-2 message CertificateInstallation	5s
ISO 15118-2 message CertificateUpdate	5s
ISO 15118-2 message CurrentDemand	0.25s
Acionamento do relé após identificação de mudança de estado do VE de conectado para carregando ⁰	3s
Tempo máximo de atraso entre um comando externo e resposta no sinal piloto (CP)	10s

Como se observa na Tabela 2.2 os requisitos temporais para as respostas aos pedidos realizados pelos veículos elétricos (VEs) aos carregadores estabelecem valores de 5s, 2s e 0,25s. Em contraste, as exigências de resposta definidas pela norma IEC 61851-1 são mais flexíveis. Considerando a possibilidade de eventos simultâneos, espera-se que um sistema composto por 6 nós, que utiliza replicação de dados, precise estar sincronizado no intervalo temporal máximo de 250 ms, conforme o protocolo Raft para sincronizar 6 nós com o novo estado.

De acordo com o projeto desenvolvido por Alzahriani e colaboradores [7], um carregador de veículos elétricos implementado utiliza o protocolo CAN para comunicação e a norma SAE J1939 da *Society of Automotive Engineers*. Nesse projeto, os dados de comunicação entre o VE e as unidades de carregamento são transmitidos através de uma rede criada para facilitar a informação processada pela ECU (*Electronic Control Unit*) do veículo. Assim, o BMS, que controla a bateria, comunica diretamente com o carregador através do protocolo CAN. O projeto faz uso dos

⁰Tempo para acionamento do relé pode ser estendido no caso da comunicação digital ter sido estabelecida durante esse tempo

protocolos CAN e Modbus para a comunicação entre o veículo e o carregador, empregando um microcontrolador Arduino UNO para gerir as comunicações.

Para utilizar o protocolo CAN de forma eficaz, o intervalo de tempo entre duas mensagens CAN pode ser de 10 ms, 50 ms, 100 ms, 250 ms ou 500 ms, conforme determinado pela norma SAE J1939. No projeto mencionado, são utilizadas mensagens enviadas com intervalos de 250 ms [7].

Com base nas informações obtidas, conclui-se que as unidades de carregamento devem ser capazes de processar os dados e sincronizar-se de maneira eficiente em um sistema distribuído, com a capacidade de enviar mensagens no intervalo máximo de 250 ms. Isso indica que, para a comunicação entre o veículo e o carregador, o sistema deve ser capaz de realizar todas as atividades necessárias de processamento e comunicação dentro de uma janela temporal de 250 ms.

No caso do carregador não conseguir responder a pedidos do VE, a norma ISO 15118 prevê o envio de respostas temporárias indicando que o processo ainda está em andamento. Por exemplo, durante o processo de autenticação, o carregador deve comunicar-se com um agregador para verificar a autorização. Isso reduz a dependência de outros meios de comunicação e agentes externos, sendo necessário apenas que o carregador realize o processamento em tempo hábil.

Ongaro D. [28], no seu trabalho, demonstrou que é possível alcançar uma média de 1 ms de Round Trip Time (RTT) entre mensagens enviadas num *cluster*, ajustando os valores máximos e mínimos dos intervalos para o envio de mensagens de confirmação e sincronização numa rede de consenso Raft.

2.3 Topologias para carregamento coordenado

2.3.1 Arquitetura de controlo centralizada

O principal conceito de carregamento centralizado é utilizar a estrutura centralizada para adquirir as informações dos VEs, processá-las centralmente e fornecer uma solução ótima considerando todas as restrições da rede e do utilizador, a unidade central de processamento executa a tomada de decisão relativamente à taxa de carregamento e horário dos VEs. A limitação do carregamento centralizado é do tamanho do problema de otimização torna-se grande com o aumento do número de veículos elétricos numa determinada área [10].

Por outras palavras, no controlo centralizado todos os parâmetros relevantes são recolhidos e os cálculos são centralizados e executados por uma única entidade, que coleta os requisitos de carregamento, resolve um problema de otimização para resolver um problema de determinar a taxa na qual cada veículo irá carregar, e comunica assim o agendamento ótimo do carregamento produzido de volta para os EVSEs.

Dessa forma, as soluções mais otimizadas estão disponíveis ao agregador, por possuir acesso a todo o sistema.

No trabalho realizado por Gonçalves, José [13], a arquitetura centralizada, também referida como arquitetura direta, é caracterizada pela comunicação direta entre o operador de rede e VE,

desta forma cada veículo pode ser tratado individualmente como um recurso determinístico, isto é, aproximações são feitas para desconsiderar incertezas relacionadas ao processo de conexão e desconexão dos VEs. As sessões de carregamento são comandada pelo operador de rede. O autor apresenta como desvantagens para essa abordagem, dois aspetos:

- Quantidade de informação de sinalização e controlo necessária para monitorizar o veículo, o conjunto de informações de cada veículo que será listado abaixo pode sobrecarregar o sistema em larga escala, as informações de sinalização são, por exemplo:
 - Estado de ligação
 - Valor de potência disponível
 - Estado de carga das baterias
 - Valor de potência elétrica máxima durante a sua carga
- O segundo aspeto negativo dessa abordagem apresentado seria a localização do veículo e a sua capacidade energética limitada, devido a sua natureza pode estar a carregar em sítios distintos, tornando a distribuição de potência da rede mais difícil de controlar, como explicado por W. Kempton em *Vehicle-to-grid power fundamentals* [19].

O trabalho citado se refere ao contexto do uso da tecnologia V2G e assume alta capacidade de trocar informações entre o veículo e o carregador, mas essa capacidade não se estende a atual frota de VEs disponível no mercado, limitando a comunicação a somente a sinalização básica definida na IEC 61851-1. Por isso, estratégias que sejam capazes de processar o carregamento de forma local e ainda carregar de forma inteligente os veículos elétricos sem precisar de extrair todas essas informações dos veículos pode ser uma estratégia interessante para aplicações reais em que o carregamento inteligente será possível com qualquer veículo que se conecte a rede mediante um carregador.

2.3.2 Arquitetura de controlo distribuída

Em sistemas distribuídos as ações são executadas por diferentes entidades para obter parâmetros relevantes através da comunicação. Mais especificamente, em estruturas distribuídas a carga pode ser dividida entre vários agentes, sendo que cada um resolve o seu problema em menor escala, de tal forma que a cada indivíduo possui certa responsabilidade no controlo e definição de um planeamento de carregamento [26].

Um estudo realizado por Tang Q. e colaboradores [39] propõe dois algoritmos de carregamento eficientes: a Estratégia de Carregamento Centralizada (ECC) e a Estratégia de Carregamento Distribuída (ECD). A ECC apresenta um desempenho ligeiramente superior à ECD, pois considera todas as informações e toma decisões numa unidade de controle central. Por outro lado, a ECD não requer informações privadas dos VEs e permite uma tomada de decisão descentralizada. Embora esses algoritmos se refiram ao agendamento de carregamento, tema que não é o foco deste

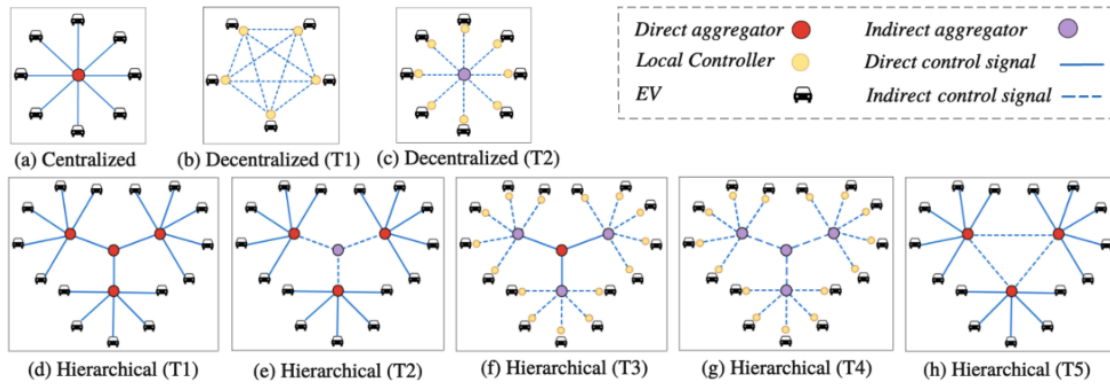


Figura 2.8: Topologias de controlo conforme [26]

trabalho, eles servem como referência relevante relativamente às estratégias de comunicação e propostas distribuídas existentes na literatura para a infraestrutura de carregamento.

Sistemas distribuídos podem ser estruturados de forma descentralizada e hierárquica, onde esquemas descentralizados compartilham o peso computacional entre os carregadores, e em esquemas hierárquicos o peso computacional fica dividido entre carregadores e os agregadores, entidades que gerem um grupo de carregadores.

Em sistemas completamente descentralizados todos os dados que o sistema possui são os dados obtidos localmente pelo grupo em que os carregadores estão incluídos. Sendo assim, utilizando uma estratégia híbrida seria possível descentralizar o sistema para permitir o processamento local em tarefas não necessárias e ainda permitir que o sistema tenha acesso a informações externas necessárias para incluir o grupo no contexto global potencialmente resolvendo problemas de menor escala e de larga escala essa estratégia se aproxima da estrutura hierárquica.

O algoritmo proposto pretende assim aproveitar as informações que não possui localmente, como, por exemplo, da rede elétrica recebendo do agregador um limite máximo de potência para o grupo que pode mudar de forma dinâmica, reagindo assim as mudanças em tempo real e redistribuindo a potência conforme o estado atual de cada carregador do grupo de forma cooperativa.

2.4 Estratégias de carregamento coordenado

Nesta subsecção, serão explorados trabalhos que empregam diferentes estratégias para a integração de veículos elétricos na rede elétrica. Inicialmente, serão apresentados os algoritmos de consenso desenvolvidos pelos autores em [38] e [24]. Em seguida, serão discutidas estratégias que combinam o mecanismo de consenso Raft com a tecnologia *blockchain*, conforme proposto por Luo, Haoxiang em [22] e por Ran, Xiumin em [32].

2.4.1 Algoritmos de carregamento coordenado

Diversos estudos conduzidos apontam para o impacto da penetração de VEs na rede elétrica, como é o caso de [23] onde se estudam estratégias para lidar com o aumento do consumo em alturas de pico. Os autores determinam que o carregamento não coordenado de VEs com alta penetração durante o período de ponta pode ser superior à capacidade de geração de energia elétrica. Considerando constante a capacidade máxima de geração, o estudo conclui, que aproximadamente 90% dos carregamentos deveria ser deslocado no tempo para intervalos de menor procura. A arquitetura de uma infraestrutura de carregamento é definida com o objetivo resolver um problema de otimização do sistema de carregamento para assim realizar um controlo para minimizar os impactos da integração dos veículos elétricos a rede ou até mesmo fornecer serviços auxiliares [26].

2.4.2 Algoritmos de consenso

Como apresentado nas estratégias distribuídas para controlo das sessões carregamento, estratégias que envolvem mecanismos de consenso, tais algoritmos têm sido cada vez mais utilizados em aplicações de natureza distribuída.

Algoritmos de consenso permitem um conjunto de máquinas a trabalhar de forma coerente como um grupo que pode sobreviver mesmo com falhas de algum dos seus membros [28]. O consenso envolve o acordo relativamente a um valor ou estado entre múltiplos servidores, agem como um grupo coerente, sobrevivendo a falhas de algum membro, é necessário para garantir a consistência e a tolerância a falhas. Por causa disso, os mecanismos de consenso são essenciais na construção de sistemas informáticos de larga escala.

Como apontado pelo estudo de Sobrinho D. [38], sistemas distribuídos vem sendo estudados desde 1960, e o mecanismo de consenso Paxos foi o algoritmo que teve maior influência no cenário, se tornou o veículo primário para se lecionar a respeito de consenso, como descrito por Ongaro D. em [28].

Algoritmos de consenso necessitam apresentar algumas propriedades para tornar um sistema distribuído prático como garantir confiança (*safety*), nunca retornando um comportamento inadequado e atingindo o consenso em condições não bizantinas, isso inclui em atrasos de comunicação, partições, perdas de pacotes, duplicações e reordenação. Em tais condições um algoritmo de consenso deve ser capaz de manter os dados corretos, também deve garantir a disponibilidade evitando erros bizantinos.

Falhas bizantinas, em sistemas distribuídos, referem-se a falhas arbitrárias em componentes individuais que podem apresentar comportamentos imprevisíveis e maliciosos, dificultando o consenso entre os nós da rede. Tal como descrito por Lamport, Shostak e Pease (1982) no artigo seminal "The Byzantine Generals Problem", estes erros podem levar a decisões inconsistentes e comprometer a confiabilidade do sistema na totalidade [27].

Comparação entre os algoritmos

A Tabela 2.3 apresenta uma classificação e comparação dos algoritmos de consenso distribuído mais utilizados. Lista os algoritmos distribuídos frequentemente empregues em redes *blockchain*, nas quais este mecanismo desempenha um papel fundamental. Os resultados demonstram que o algoritmo Raft obteve o maior débito de transações, atingindo 700 Transações Por Segundo (TPS). Para além deste desempenho notável, o algoritmo Raft destaca-se pelos elevados níveis de segurança e eficiência, aliados a um baixo consumo de recursos computacionais.

Tabela 2.3: Comparação entre algoritmos de consenso [11].

Algoritmo de Consenso	Tolerante a Falhas Bizantinas	Probabilidade Tolerância a falhas	Segurança	Eficiência	Consumo de potência	TPS
PoW	Sim	49%	Alta	Baixa	Alta	7tps, 15tps
PoS	Sim	49%	Média	Alta	Baixa	70tps
Bitcoin-NG	Sim	25%	Alta	Altíssima	Altíssima	200tps, 400tps
Byzcoin	Sim	25%	Alta	Altíssima	Altíssima	100tps
MPoW	Sim	49%	Alta	Altíssima	Média	500tps
PBFT	Sim	33%	Altíssima	Altíssima	Baixa	200tps, 300tps
HotStuff	Sim	33%	Alta	Alta	Baixa	Sem informação
Paxos	Não	49%	Alta	Alta	Baixa	Sem informação
Raft	Não	49%	Alta	Alta	Baixa	700tps

Os algoritmos de consenso distribuídos podem ser comparados com base numa série de critérios, incluindo:

- Robustez: Capacidade de lidar com falhas de nós.
- Eficiência: Capacidade de atingir um consenso rapidamente e com baixo custo de comunicação.
- Simplicidade: Facilidade de implementação e compreensão.
- Segurança: Capacidade de resistir a ataques maliciosos.

Como o objetivo desse trabalho é validar a possibilidade de usos de mecanismos de consenso em infraestrutura de carregamento de veículos elétricos, investigação na literatura em relação a estudos de desempenho desse algoritmo foi realizado e o estudo [15] apresenta uma análise sistemática indicando a probabilidade de divisão da rede em caso de falhas na comunicação do líder para os seguidores através das mensagens de pulsação enviadas periodicamente a uma taxa fixa, chamados *heartbeats*, sendo assim esse estudo realizou mais de 10 000 simulações variando a quantidade de nós na rede, o tempo limite para eleições a taxa de perdas de pacotes.

O algoritmo Raft é aplicado geralmente em redes *ad hoc*. Isso significa que contribui para adição e remoção de nós.

2.5 Raft

Raft é um algoritmo de consenso projetado para gerir logs e máquinas de estado replicadas. A sua principal característica é a busca pela simplicidade e facilidade de compreensão, alcançada através da decomposição do problema em três subproblemas: eleição de líder, replicação de *logs* e segurança [28]. Essa abordagem permite uma análise gradual do problema do consenso e dos sistemas distribuídos, facilitando a compreensão dos conceitos envolvidos.

Num *cluster* Raft os nós podem tomar três estados diferentes: líder, seguidor e candidato. Sendo que num *cluster* comum, existe somente um líder entre os participantes, todos os outros nós servem como seguidores. Todos os seguidores são nós passivos que redirecionam os pedidos recebidos pelo cliente para o líder. O Tempo é dividido em termos de tamanho arbitrário, e no fim de cada termo um novo processo de eleição é iniciado. A Figura 2.9 retirada de [28] ilustra os estados e devidas transições que cada nó num cluster Raft podem tomar.

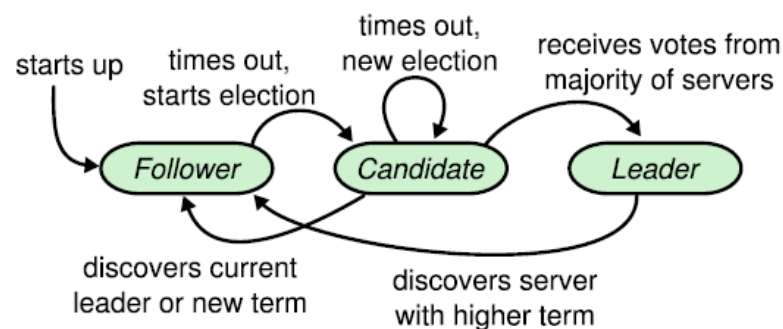


Figura 2.9: Estados de um nó do protocolo Raft [28]

Os líderes mantêm o seu estado através do envio de mensagens cíclicas temporizadas que confirmam a presença do líder ativamente, chamada mensagens de *heartbeats*. Quando um seguidor deixa de receber uma dessas mensagens por um intervalo de tempo definido por referência, mas com uma variação aleatória para que outros nós não acordem no mesmo instante, o seguidor assume que não existe um líder disponível no momento e inicia um processo de eleição, mudando para o estado de candidato. Então o nó que agora é candidato vota por si próprio e realiza pedidos aos outros nós para votarem nele. O nó continuará nesse estado até que ganhe a eleição, caso não haja uma decisão na eleição atual, após um período aleatório um novo nó inicia o processo de eleição e esse procedimento repete-se até que um único líder seja eleito [30].

Devido a essas propriedades o algoritmo Raft é considerado tolerante a falhas, mas não tolerante a falhas bizantinas, isso acontece porque durante as eleições dois nós maliciosos podem ignorar os *timeouts* e mudar de líder entre eles, impedindo que o sistema saia da fase de eleição.

Todos os pedidos dos clientes são armazenados nos *logs* do líder e então replicados para todos os outros nós. Isso é realizado através de um único líder continuamente enviar eventos até que todos os nós tenham os seus dados replicados nos seus *logs* [28].

Cada nó possui um comando que foi executado, organizados por índices, e o registro de cada termo que os comandos foram executados. Um *log* é confirmado pelo líder quando a maioria dos nós da rede estão sincronizados de forma bem sucedida no seu estado. Uma vez que os *logs* são confirmados, o líder executa o comando e todos os relatórios de volta ao cliente. Porque os comandos são executados em ordem, a confirmação dos *logs* também é realizada com pedidos confirmados anteriores [28].

Inconsistências são resolvidas pela imposição da replicação dos *logs* do líder, garantindo a coerência em caso de conflitos entre inserções nos *logs* [28].

A segurança do sistema é assegurada por meio da regra que só permite um único líder por *cluster* num determinado termo. Adicionalmente, durante o processo de eleição, um nó irá recusar a votar num candidato que possui o seu *log* mais recente continua desatualizado do seu próprio [28].

O mínimo número de nós exigidos para executar ações num *cluster* é de $2/N+1$, sendo N o número de nós integrantes do *cluster*, isso significa que o Raft tolera até metade dos seus nós estarem desconectados [28].

O processo de eleição no Raft serve para garantir o forte líder que o algoritmo promete, quando os servidores Raft iniciam, começam no estado de seguidores e permanecem nesse estado enquanto recebem mensagens RPCs válidas de um líder ou candidato.

Raft usa mecanismos de *heartbeats* como gatilho para as eleições do líder. Enquanto o líder estiver ativo e funcionando corretamente, os seguidores irão receber mensagens que confirmam que o líder está ativo. Se as mensagens desaparecem, o seguidor espera um certo tempo, com uma margem aleatória, e quando esse tempo passa o tempo o seguidor inicia um novo processo de eleição, vota em si e envia pedidos de votos para outros nós. Para iniciar uma eleição, o seguidor incrementa o seu termo atual e realiza a transição para o estado de candidato. Enquanto espera por votos, o candidato pode receber mensagens de *AppendEntry* de outro servidor proclamando ser o líder. Se o termo do líder é maior que o termo do candidato, então ele aceita o pedido e volta a tornar-se um seguidor. Se não, continua o processo de eleição.

A eleição de líder no protocolo Raft é um processo complexo que envolve uma série de regras e procedimentos. O processo é projetado para ser robusto e tolerante a falhas, mesmo na presença de nós não confiáveis.

No geral, o protocolo Raft é um protocolo eficaz e tolerante a falhas que pode ser usado numa variedade de aplicações. No entanto, é importante estar ciente dos problemas e dificuldades potenciais ao implementar o protocolo.

2.5.1 Mecanismos de consenso em mobilidade elétrica

Como explicado por Ongaro D. [28], um mecanismo de consenso em sistemas distribuídos é um processo que permite que um grupo de entidades (computadores, servidores, etc.) chegue a um acordo sobre um valor ou decisão, mesmo na presença de falhas ou informações inconsistentes. Este acordo é crucial para garantir a confiabilidade e a coerência dos dados em sistemas onde múltiplas entidades interagem e podem ter visões diferentes do estado do sistema.

Alguns trabalhos implementam algoritmos de consenso em conjunto com técnicas de otimização para realizar o controlo do carregamento ou algum serviço auxiliar. Tais esforços visam mitigar problemas de forma descentralizada, trabalhos que desenvolvem algoritmos de consenso próprio são apresentados, além de trabalhos que utilizam mecanismos bem estabelecidos e amplamente utilizados.

A plataforma REVS: Reliability-Aware EV Charging Scheduler, propõe um algoritmo distribuído que gera um agendamento de carregamento ótimo de VEs para instalações residenciais, baseado nas preferências do consumidor e se atualiza iterativamente até as restrições definidas pelo operador de rede sejam satisfeitas [24].

REVS propõe um modelo de programação linear inteira mista (MILP) com variáveis binárias que indicam o estado que provem das unidades de carregamento. O problema poderia ser resolvido por uma central local, mas como se trata de aplicações residenciais o operador não poderia ter acesso a informações de carga por questões de privacidade, por isso uma abordagem distribuída foi realizada nesse trabalho. Da mesma forma a topologia da rede e parâmetros obtidos da rede não podem ser compartilhados com os consumidores.

A estratégia de otimização utilizada utiliza o método ADMM (*Alternating Direction Method of Multipliers*), que foi provado converge para soluções ótimas de problemas convexos, problemas esses que regressões lineares não conseguem chegar a soluções ótimas, neste trabalho, por utilizar restrições binárias para unidades de carregamento o que torna o problema não convexo.

O procedimento descentralizado proposto envolve conjuntos de mensagens duplas que são trocadas entre o operador de rede e os consumidores residenciais. O sistema continua a realizar iterações até que o operador de rede atualize a trajetória da potência segundo os objetivos calculados, sendo assim, garante consenso com a trajetória do uso de potência nas residências [24].

O trabalho realizado por [38] propõe um método para se realizar um controlo das sessões de carregamento utilizando um mecanismo de consenso baseado em grafos, o algoritmo é desenvolvido pelo próprio autor, o algoritmo visa a execução de um carregamento eficiente.

A exploração de mecanismos de consenso no contexto do carregamento de veículos elétricos está na sua fase inicial até o momento deste trabalho, tais mecanismos estão geralmente enquadrados em redes *blockchain* pois garantem a estabilidade, replicabilidade e tolerância a falhas nessas aplicações.

Blockchain é um sistema de transações e armazenamento de dados distribuído e encriptado, onde todos os integrantes, *peers*, compartilham informações de uma forma descentralizada e segura. As suas principais características são: descentralização, imutabilidade, atomicidade, ser auditável, capacidade de confirmar ou rejeitar uma transação já realizada. Blockchain tem sido um ativo muito útil cobrindo diversos campos científicos diferentes [15].

No estudo realizado por Ran, Xiumin em [32], apresenta-se um novo sistema que realiza o cálculo de pilhas de carregamento baseado na tecnologia *blockchain* e no algoritmo Raft. O sistema é dividido em três módulos: módulo de controlo, módulo de pagamento e módulo de iteração com o utilizador.

Algoritmo de consenso desenvolvido utilizando uma combinação de dois mecanismos de consenso de baixa complexidade, PBFT e Raft [22]. Inicialmente é feita a combinação de PBFT e Raft foi nomeado PRAFT, que consiste em separar o sistema em nós primários e secundários e fazer com que os nós primários participem no consenso segundo PBFT e os nós secundários participem no consenso segundo o mecanismo Raft, a Figura 2.10 representa os mecanismos de consenso de cada nó:

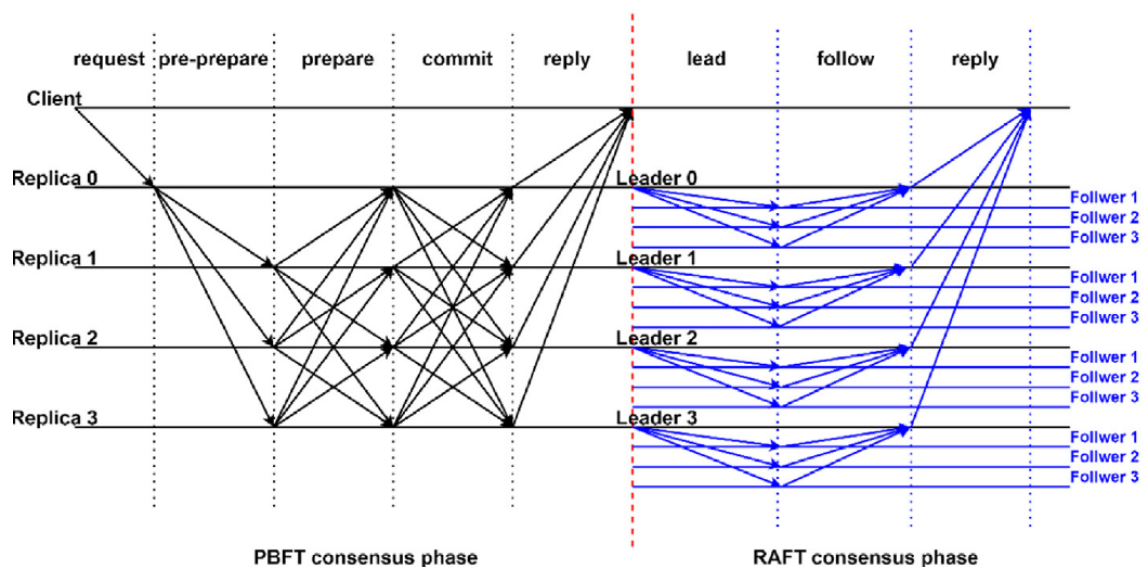


Figura 2.10: Mecanismo de consenso PRAFT [22]

Na Figura 2.10 é representado o processo a seguir descrito:

1. O cliente notifica a réplica 0 no nó primário para habilitar o consenso PBFT.
2. Os nós primários completam todos os passos e retornam de acordo com o processo PBFT.
3. O Cliente, após receber as mensagens de cada nó primário, codifica as mensagens (*hashes*) em sequência. (Esse processo evita que o nó primário emita mensagens falsas) Agora em relação à fase de consenso do Raft, em que os nós primários descritos acima se comportam como líderes no Raft e vão atingir consenso com os seus seguidores.
4. Cada líder envia os seus resultados de consenso na fase PBFT para os seus seguidores.
5. Cada Seguidor irá codificar (hash) os valores armazenados na nuvem pelo cliente no passo 3. Se eles forem os mesmos então vai para o passo 6, se não volta para o passo 4.
6. Quando os valores codificados são consistentes, seguidores irão iniciar o consenso Raft em cada grupo, confiando nas mensagens e armazenando a mensagem dos seus líderes.
7. Finalmente, o consenso PRAFT termina, a esse ponto todos os nós na rede atingiram consenso. E o resultado do consenso são armazenados.

Em seguida é realizada a combinação de Raft e PBFT, formando o algoritmo nomeado RPBFT, nesse processo de consenso descrito na Figura 2.11 os nós na rede necessitam de uma forma diferente de agrupamento de acordo com algumas regras.

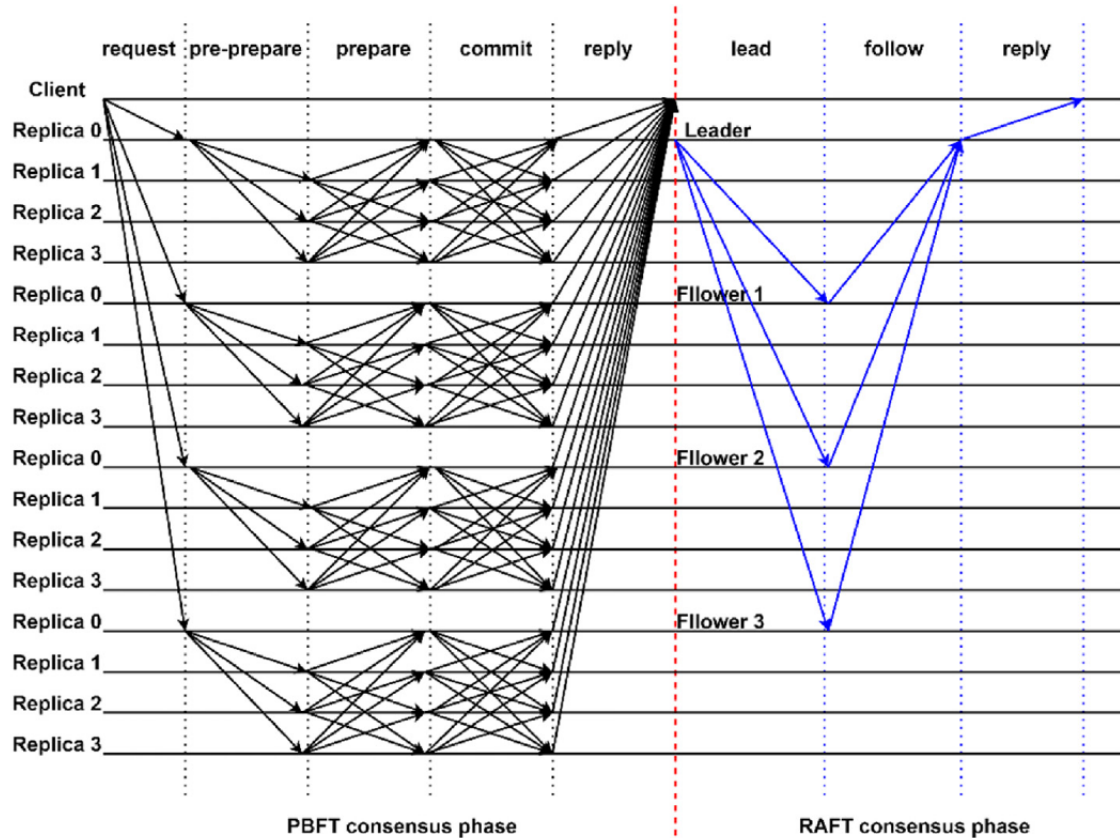


Figura 2.11: Mecanismo de consenso RPBFT [22]

O processo de consenso específico é descrito abaixo:

1. Os nós na rede são inicialmente divididos em diferentes grupos e o consenso PBFT é iniciado em cada grupo.
2. Quando o consenso PBFT termina, o Cliente codifica o resultado de cada grupo e envia os valores codificados para a nuvem.
3. O nó primário então é eleito dentro de cada grupo por uma certa regra que conduz o novo consenso Raft que permite que todos os nós na rede interajam com a informação.
4. Dentre os nós primários, o líder é eleito conforme regras próprias do mecanismo de consenso, os outros nós primários tornam-se Seguidores. Então, o Líder entrega as mensagens de consenso para os Seguidores.
5. Cada Seguidor codifica as mensagens recebidas e compara com as mensagens codificadas no passo 2 que estão na nuvem. Se elas são iguais, vai para próxima etapa, se não, volta para passo 4.

6. Quando os valores codificados são consistentes, seguidores iniciam o consenso Raft e então confiam nas mensagens recebidas e armazenam para o Líder.
7. Finalmente, o processo de consenso do RPBFT finaliza. Neste ponto, todos os nós na rede atingiram consenso e o resultado do consenso é armazenado.

Após isso foi realizada a análise dos algoritmos, sendo algoritmos que permitem alta escalabilidade, baixa complexidade de comunicação, pouca carga de armazenamento, débito alto e latência baixa, além de serem tolerante a falhas. Nesse trabalho os algoritmos são apresentados num cenário de carregamento real.

Como apresentado nesse trabalho para o cálculo de complexidade de comunicação de cada algoritmo o mecanismo PBFT necessita de ao menos 3 nós para atingir consenso quando um nó falha, por isso o mínimo viável para que o sistema funcione com o mínimo de tolerância a falhas seria quatro nós. Sendo assim, o Raft é um algoritmo mais versátil, pois bastam dois nós estarem disponíveis que o processo de eleição se inicia e um dos nós se tornará o líder e assim atingirem o consenso do sistema.

Como resultado do trabalho o realizado segue as Figuras 2.12a e 2.12b representam o débito e latência medidos dos algoritmos:

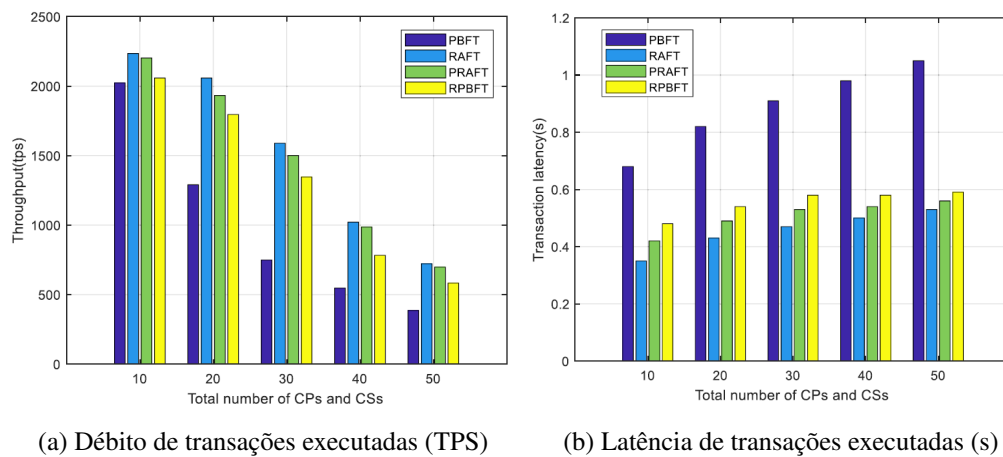


Figura 2.12: Resultados de estudo dos algoritmos PRAFT e RPBFT [22]

O estudo [36], fez uma análise dos algoritmos Raft, IBFT, QBFT e Clique, neste trabalho foi explorado de forma sistemática o estado da arte de cada um desses algoritmos, para análise dos trabalhos levantados, foi realizado um filtro baseado nas seguintes questões abaixo a serem respondidas:

1. Qual a métrica de eficiência utilizada?
2. Qual a métrica de tolerância a falhar utilizada?
3. Qual a *framework* ou estratégia para teste de desempenho (*benchmark*) dos algoritmos?

A primeira pergunta se difere da terceira, pois a eficiência, está mais relacionada a latência e débito de resposta ou até mesmo no aspeto relacionado a comunicação do algoritmo, são medidas mais específicas de desempenho, já o teste de desempenho, ou *benchmark*, consiste numa série de testes para se determinar o desempenho do algoritmo considerando diferentes aspetos e em diferentes cenários, em resumo, geralmente o benchmark é escolhido um problema complexo conhecido e é executado nos diferentes dispositivos sobre testes, no caso algoritmos de consenso, é feita uma análise do resultado por meio do desempenho assim é possível realizar uma comparação mais justa, as estratégias contempladas no estudo [36] são *Synthetic application*, *Hyperledger Caliper*, *Blockbench* *Chaos Engineering*.

No capítulo 3 será explorado a técnica de benchmark e as métricas aplicadas neste trabalho para se obter os resultados aplicados a infraestrutura de carregamento de VEs.

2.5.2 Metodologia de testes para comunicação distribuída

Para além das estratégias de *benchmark* foram definidos indicadores de desempenho (KPIs), algoritmos de consenso distribuídos são mecanismos que permitem que um grupo de nós numa rede chegue a um acordo sobre um valor comum, mesmo que alguns dos nós falhem.

No estudo [30], para realizar a pesquisa de métricas válidas que indicassem valores palpáveis para validar e comparar modelos de sistemas distribuídos foram formuladas duas perguntas-chave:

1. Qual o efeito dos algoritmos de consenso no desempenho?
2. Qual o efeito dos algoritmos de consenso na tolerância a falhas?

O Hyperledger Performance and Scale Working Group (PSWG) propuseram uma série de termos básicos e métricas que servem de norma para análise de desempenho de tecnologias distribuídas [30].

Latência e débito nas transações de leitura de escrita são identificados como as principais métricas para determinar o desempenho de um sistema. Em adição, consumo de recursos e escalabilidade também auxiliam na medida para se avaliar qual bem o sistema se comporta em diferentes casos, resultando na seguinte lista de métricas:

- Débito de transações
- Débito de leituras
- Latência de transações
- Latência de leituras
- Consumo de recursos
- Escalabilidade: Relação entre o número de nós e a latência de transações

- Taxa de sucesso nas transações: única métrica para se avaliar a capacidade de ser tolerante as falhas

Uma transação é considerada quando um estado dentro da rede de consenso é modificado. Débito de transação é a taxa em que transações validadas são confirmadas através do consenso num determinado intervalo temporal. Sendo assim representa a taxa para toda o grupo conectado e não somente para um único nó [16].

$$\text{Débito de transação} = \frac{\text{Total de transações confirmadas}}{\text{Total de tempo em segundos}} \quad (2.1)$$

Latência de uma transação também é medida a nível da rede de consenso, e representa o tempo necessário para uma transação submetida por um cliente até que o seu resultado esteja disponível em toda a rede. Isso significa que a propagação do mecanismo de consenso e o tempo para estabilização estão incluídos na latência.

$$\text{Latência de transação} = \text{Tempo de confirmação} - \text{Tempo de submissão do pedido} \quad (2.2)$$

As leituras são medidas diretamente no cliente, sendo assim, são muito mais simples de serem definidas. Débito de leitura representam quantas operações de leitura são completadas em um intervalo de tempo.

$$\text{Débito de leitura} = \frac{\text{Total de operações lidas}}{\text{Tempo total em segundos}} \quad (2.3)$$

Latência de leitura, é por consequência, a diferença entre o tempo que o pedido foi enviado e o tempo de resposta recebido.

$$\text{Latência de leitura} = \text{Tempo de recebimento da resposta} - \text{Tempo de submissão do pedido} \quad (2.4)$$

É importante notar que todos os valores de latência médios, máximos e mínimos serão obtidos para serem comparados posteriormente.

Nos testes realizados posteriormente neste trabalho, o consumo de recursos irá ser obtido através da leitura do estado de consumo da CPU e memória RAM durante o uso do algoritmo para testes de desempenho (*benchmark*).

2.6 Sumário

Este capítulo apresentou o estado da arte dos sistemas de gestão de carregamento de veículos elétricos. Foram discutidos os principais atores que compõem a infraestrutura de carregamento, os

métodos propostos para o carregamento coordenado e as normas da indústria. Além disso, foram explorados os sistemas distribuídos e as técnicas mais comuns utilizadas, com um enfoque particular no mecanismo de consenso Raft, que define a interação entre os integrantes de um sistema distribuído. Este mecanismo de consenso será o utilizado para o sistema distribuído proposto nesta tese.

Através do estudo dos protocolos e normas de comunicação, foi determinado o que é possível de ser realizado nas estações de carregamento, revelando as limitações e oportunidades existentes. Com base na literatura apresentada, podemos concluir que a gestão de carregamento de veículos elétricos é um desafio complexo que exige a consideração de diversos fatores, como a infraestrutura de carregamento, os métodos de carregamento coordenado, as normas da indústria e as técnicas de comunicação. Embora diversas soluções tenham sido propostas, nenhuma delas se destaca como ideal.

Além disso, o estudo da literatura atual evidencia uma ausência de investigação em sistemas distribuídos aplicados à mobilidade elétrica. Assim, é necessário o desenvolvimento de novas soluções que sejam mais eficientes, robustas e escaláveis, levando em consideração as lacunas identificadas e as possibilidades oferecidas pelos sistemas distribuídos.

Capítulo 3

Descrição do Sistema Proposto

Este capítulo pretende elucidar às características do sistema desenvolvido, no qual se pretende explorar as novas abordagens para a gestão das sessões de carregamento de veículos elétricos. O sistema consiste numa plataforma distribuída, projetada para permitir que os carregadores de veículos elétricos realizem funções de gestão de forma coordenada e eficiente.

Esta plataforma possibilita a interação e colaboração entre um grupo de carregadores, assegurando robustez e eficiência na gestão coletiva das sessões de carregamento. O desenvolvimento deste sistema visa aumentar a eficiência e a confiabilidade da infraestrutura de carregamento de veículos elétricos, oferecendo uma solução inovadora para os desafios atuais do setor.

Neste capítulo o propósito é clarificar a estrutura do sistema proposto, bem como cada ator envolvido no processo.

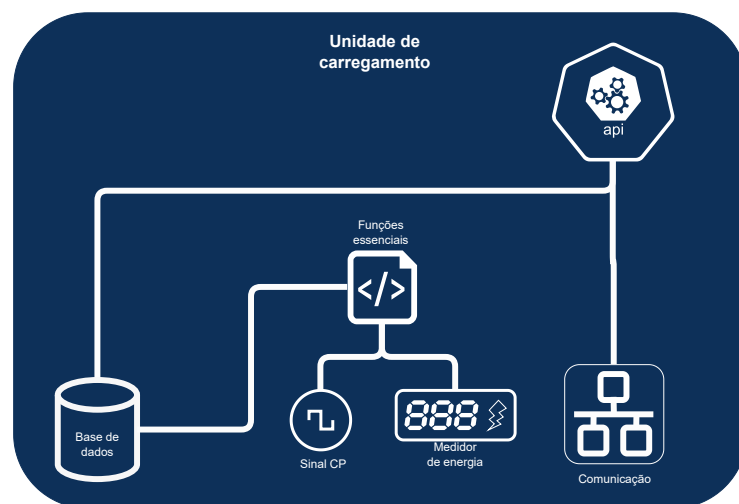


Figura 3.1: Componentes do sistema informático das unidades de carregamento

O sistema proposto é constituído por unidades de carregamento ou EVSEs (*Electric Vehicle Supply Equipment*). Estas unidades de carregamento foram desenvolvidas no laboratório de sistemas de potência do INESC TEC, SGEVL (*Smart grids and Electric Vehicles Laboratory*), e realiza

o carregamento AC de veículos elétricos, a Figura 3.1 representa as funções dessas unidades de carregamento na infraestrutura atual, as ademais funcionalidades e módulos serão apresentados no próximo capítulo na Secção 4.1

A comunicação entre unidades de carregamento no sistema de recarga de veículos elétricos nas instalações da INESC TEC possui uma infraestrutura de rede robusta. Cabos Ethernet estabelecem a conectividade física, vinculando as unidades a um *switch* que atua como ponto central de comutação. O direcionamento do tráfego de dados entre os dispositivos é realizado pelo *switch*, garantindo a comunicação eficiente. Ademais, as unidades de carregamento operam numa mesma rede virtual (VLAN), assegurando a segurança e o isolamento do tráfego, aspetos cruciais para a confiabilidade do sistema.

Para o sistema de carregamento coordenado proposto operar corretamente, as normas ISO 15118 e OCPP devem trabalhar em conjunto e em diferentes camadas para habilitar funcionalidades que enquadrem diferentes configurações da infraestrutura de carregamento, a fim de permitir diferentes estratégias de controlo e arquiteturas de comunicação. Para tal fim, este trabalho utiliza bibliotecas fornecidas por tais normas para assegurar testes com um sistema que integre todos os agentes previamente citados. Além disso, também realiza testes com uma proposta distribuída que inclui o mecanismo Raft para explorar a capacidade das unidades de carregamento de gerirem-se por si mesmas durante as sessões de carregamento do *cluster*.

A Figura 3.5 apresenta 4 atores: o servidor OCPP responsável por gerir as sessões de carregamento; o cliente OCPP que interage com o servidor mediante a ocorrência de eventos no processo; O SECC (*Supply Equipment Communication Controller*) e o EVCC (*Electric Vehicle Communication Controller*), como é definido pela norma ISO 15118-2. Os dois últimos atores são responsáveis por gerirem a comunicação em alto nível entre o veículo e o carregador. Detalhes mais aprofundados de cada módulo e do caso de uso utilizado neste trabalho, demonstrando como eles interagem entre si, serão apresentados posteriormente neste capítulo.

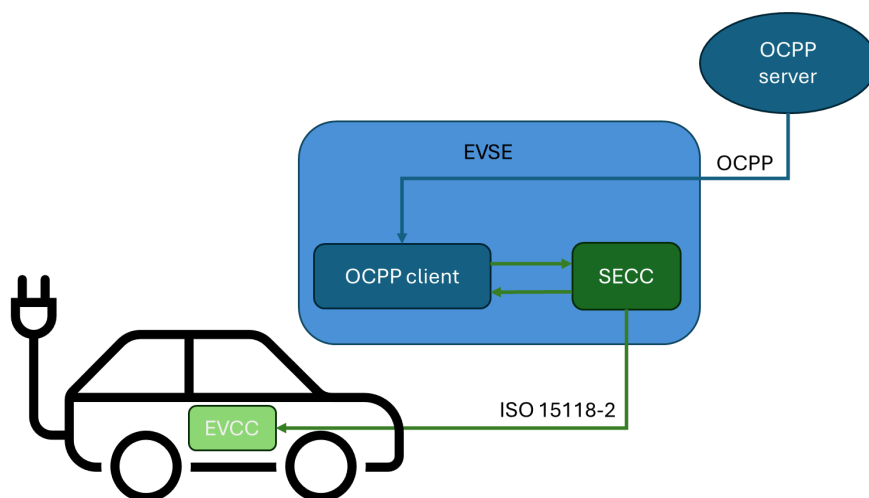


Figura 3.2: Representação real dos atores e os seus módulos de comunicação

Para que múltiplos servidores possam coabitar e trabalhar em conjunto, é necessário estabelecer uma comunicação entre eles. Neste trabalho, é utilizado o protocolo Raft para realizar a comunicação entre os servidores OCPP, responsáveis pela gestão das sessões de carregamento. Esta comunicação permite a eleição de um líder, assegurando que apenas um único servidor esteja ativo de cada vez. Além disso, o mecanismo Raft permite a replicação de dados ou do estado do sistema, uma característica fundamental deste protocolo.

3.1 Plataforma distribuída proposta

A plataforma distribuída proposta neste trabalho tem como base o protocolo de comunicação OCPP e ISO 15118-2. Para a realização dos testes foi modificado o modelo da unidade de carregamento representada na Figura 3.1. A Figura 3.3 apresenta uma idealização do sistema proposto.

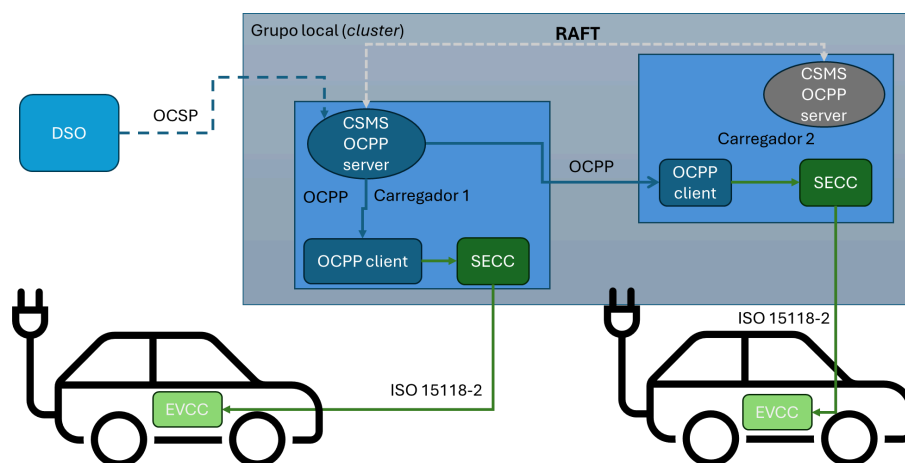


Figura 3.3: Sistema distribuído proposto

A organização distribuída dos servidores OCPP é a abordagem utilizada nesse trabalho, ainda não documentada na literatura até o momento desta pesquisa. Esta estrutura permite maior tolerância às falhas na gestão e propõe a inclusão deste sistema nas unidades de carregamento, aproveitando, dessa maneira, a capacidade computacional desses dispositivos e adicionando flexibilidade à infraestrutura para casos como instalações remotas.

A rede distribuída realiza a coordenação entre as instâncias gestoras e permite a adição de nós heterogêneos que não sejam carregadores. Estes nós, assim como os carregadores, devem possuir conectividade em rede local ou global, garantindo comunicação estável e com latência adequada, conforme os parâmetros das mensagens de *heartbeats*, ou mais tecnicamente, *append entries* da rede Raft, que garante a posição do líder.

Embora o mecanismo distribuído se recupere de falhas de mensagens, o sistema depende de um líder estabelecido para funcionar corretamente, por isso a exigência de uma rede estável tem o objetivo de garantir o duradouro funcionamento do sistema.

Para além do líder forte, a rede Raft precisa de um tempo de resposta baixo, taxa de envio das mensagens inferiores aos da comunicação do protocolo OCPP e das mensagens ISO 15118 entre

EV e EVSE, para que o processo de eleição ocorra o mais rápido possível, garantindo assim a disponibilidade de um agregador para o sistema.

A temporização das mensagens OCPP é flexível, a variar conforme o caso de uso definido pela entidade que implementa o protocolo. No entanto, a norma ISO 15118-2 exige tempos de respostas mais restritos, como apresentado na Tabela 2.2 no Capítulo 2.

As mensagens ISO 15118-2 que geralmente dependem de informações externas do gestor dos pontos de carregamento, como autenticação e agendamento de carregamento, podem ser enviadas iterativamente entre o veículo e o carregador. Isto garante que, quando o servidor estiver disponível para reagir ao pedido, a informação será fornecida ao veículo e a sessão de carregamento pode prosseguir sem falhas.

A camada distribuída adicionada à infraestrutura de carregamento não altera a natureza da interação entre os atores quatro atores envolvidos (servidor OCPP, cliente OCPP, EVCC, SECC). Sendo assim, o sistema não consolida uma única solução, mas fornece uma plataforma em que permita a personalização da infraestrutura prevista por norma.

Na Figura 3.4 os itens adicionados para a experiência são sinalizados com a coloração laranja para diferenciar dos componentes nativos da unidade de carregamento, representados na coloração branca. Essa representação demonstra que durante os testes as funcionalidades básicas do carregador esteve em operação normal aproximando do cenário real em que na comunicação de alto nível da norma ISO 15118-2 acontece sobre a comunicação de baixo de nível da norma IEC 61851.

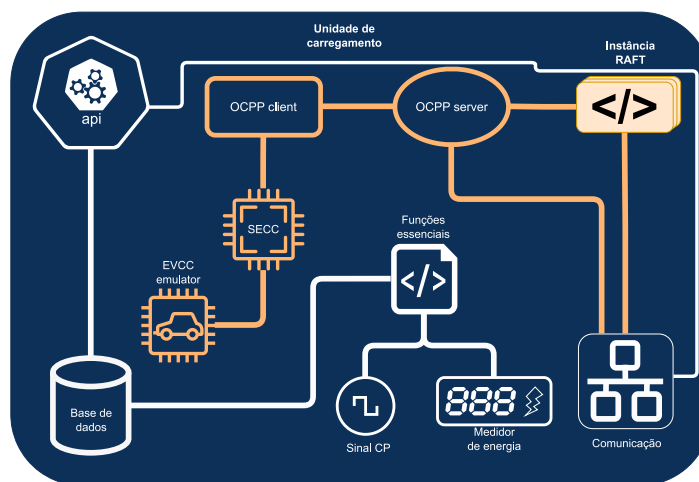


Figura 3.4: Componentes do sistema informático das unidades de carregamento

3.1.1 Blocos funcionais implementados

No que se refere a funcionalidade de cada componente do sistema, a Figura 3.5 apresenta as principais funções exercidas por cada módulo funcional do sistema para assim ser possível a gestão das sessões de carregamento e a execução dos testes para verificar se o sistema é viável de ser implementado em casos reais.

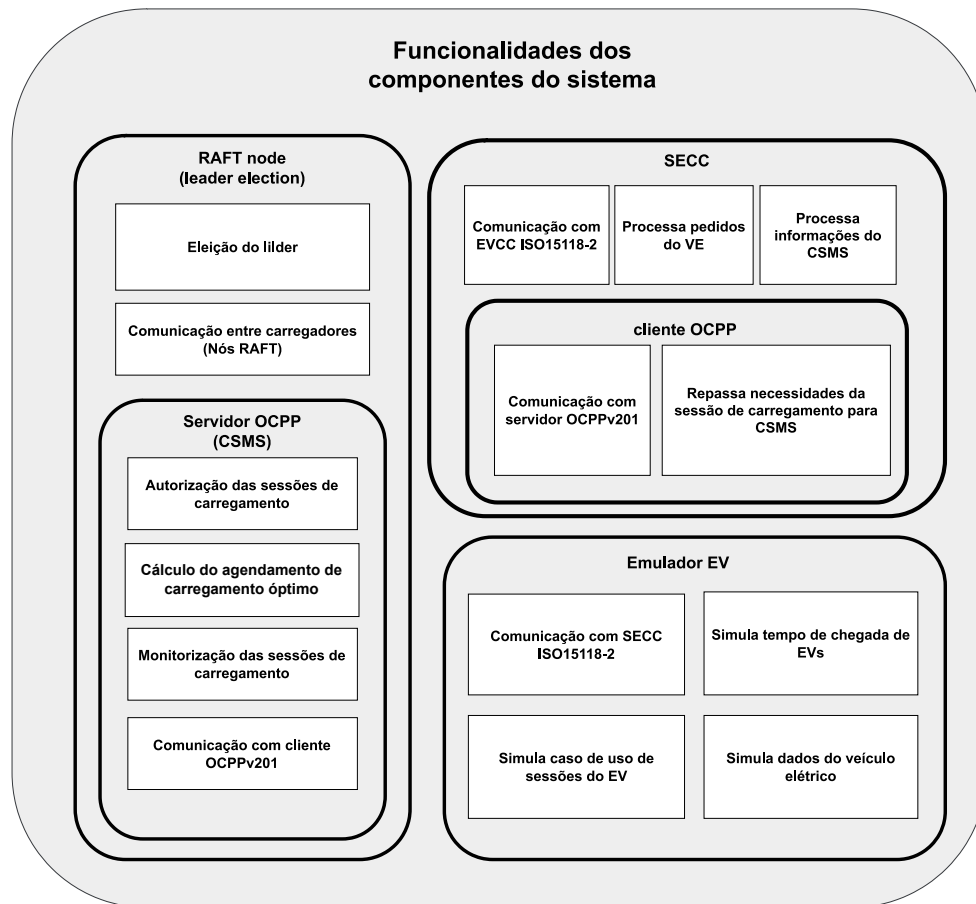


Figura 3.5: Componentes do sistema informático das unidades de carregamento

3.1.1.1 Módulo servidor Raft

O servidor Raft executa um mecanismo de consenso que permite determinar quais servidores OCPP estarão ativos. Os nós que constituem o grupo, ou *cluster*, quando inicializados executam o processo de eleição do líder. Nessa etapa é necessário que a maioria dos nós estejam ativos, isto é, em um *cluster* constituído por N nós, $N/2 + 1$ nós devem estar ativos e responder aos pedidos realizados para que o processo seja capaz de atingir consenso.

Uma limitação identificada na biblioteca utilizada é que ela não possui um mecanismo nativo para diferenciar nós inativos de nós ativos, mantendo sempre o número total de nós definido na configuração inicial para o cálculo do *quorum*, mesmo quando alguns destes nós estão inativos. Como consequência, em casos onde não existe a maioria dos nós respondendo, o sistema permaneceria indefinidamente aguardando um *quorum* que inclui nós que podem não estar mais disponíveis.

Para contornar esta limitação específica da biblioteca, foi implementado um mecanismo que

permite identificar e remover temporariamente da contagem de *quorum* os nós que não respondem às requisições. O processo verifica periodicamente a conectividade dos nós e, quando não existe maioria ativa, remove um nó inativo por vez da contagem total até que seja possível estabelecer um *quorum* com os nós efetivamente disponíveis. Este processo se repete até que a condição de maioria seja satisfeita, permitindo que a eleição do líder prossiga e o servidor OCPP seja ativado.

Como representado na Figura 3.6 a sequência lógica de eventos e operação de um único nó Raft. Através desse módulo é incluída assim a propriedade distribuída intencionada para esse trabalho.

3.1.1.2 Módulo servidor OCPP

O *Open Charge Point Protocol* (OCPP) disponibiliza uma biblioteca em *Python* que implementa o protocolo de comunicação padrão entre Pontos de Carregamento (clientes) e um Sistema Central de Gestão (servidor). Nesta arquitetura cliente-servidor, o Sistema Central atua como gestor das transações de carregamento, enquanto os Pontos de Carregamento enviam requisições e respondem a comandos do servidor, permitindo assim o controle e monitoramento remoto das estações de carregamento.

A biblioteca utiliza a tecnologia de comunicação por *websockets* como base para o estabelecimento da conexão entre os agentes envolvidos. É fornecido as estruturas das mensagens em JSON com a especificação dos parâmetros obrigatórios e opcionais de cada mensagem.

Neste protocolo há ainda uma exceção no que se refere ao paradigma de comunicação entre o servidor e o cliente, o qual permite que o servidor envie pedidos e solicite dados ao cliente em casos de uso específicos. Em outros termos, durante o estabelecimento inicial da ligação o servidor fica a espera do pedido do cliente, mas no processo de agendamento de carregamento, por exemplo, o servidor envia o pedido ao cliente para assim receber uma resposta de confirmação.

3.1.1.3 Módulo cliente OCPP

Os clientes OCPP são responsáveis por enviar os pedidos ao servidor OCPP conforme os casos de uso determinados pela norma. Através da troca de mensagens com o servidor OCPP, o cliente fornece as informações necessárias para realizar a gestão das sessões de carregamento; por isso, é necessário estar em coordenação com o SECC e o servidor OCPP para que a comunicação realizada entre os módulos não possua interrupções indesejadas. Os diagramas de sequência das próximas seções representam essa interação sincronizada entre o veículo e o carregador baseado no carregamento inteligente da norma ISO 15118.

3.1.1.4 Módulo SECC

No contexto da norma ISO 15118 temos uma biblioteca fornecida pelos fundadores da norma na linguagem de programação *Python*. A biblioteca fornece uma implementação das funções do SECC (*Supply Equipment Communication Controller*), a abstração do módulo que está dentro das unidades de carregamento permite a compreensão das suas funções e adequação a diferentes casos

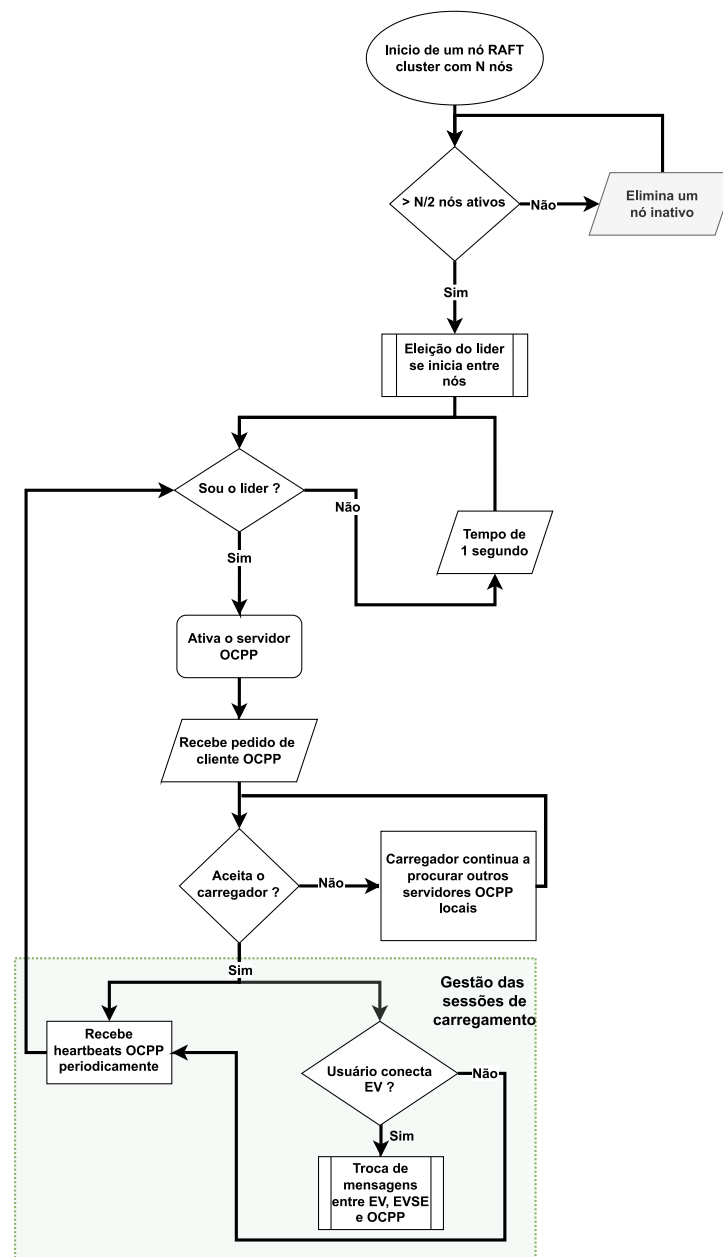


Figura 3.6: Diagrama lógico dos nós Raft com servidor OCPP

de uso. Permite comunicação com os veículos elétricos em alto nível (*High level communication*) como explicado no Capítulo 2, mediante trocas de mensagens seguindo o mesmo paradigma cliente-servidor utilizado pelo OCPP, mas com a restrição dos pedidos somente poderem ser iniciados do lado do veículo. Em alguns casos de uso específico, quando o veículo necessita finalizar

as sessões de carregamento é ativamente enviado uma *flag* no cabeçalho de uma das respostas das mensagens do SECC e o próprio SECC em seguida envia uma mensagem de finalização da sessão, *Session Stop*.

3.1.1.5 Módulo EVCC - Emulador

Electric Vehicle Communication Controller é o módulo que integra o veículo elétrico dando a capacidade de comunicação por meio do sinal CP e utilizando o protocolo HPGP para realizar a comunicação em alto-nível pelo protocolo ISO 15118-2/20. Entretanto, o módulo utilizado neste trabalho não possui a conexão física e de enlace, mas utiliza *sockets* TCP/TLS para trocar as mensagens, pois emula o comportamento do lado do veículo durante o processo. Para isso foi utilizado uma biblioteca em *Python* fornecida pela norma ISO 15118, entretanto, algumas modificações foram necessárias para permitir um comportamento estocástico a fim de se aproximar de cenários factuais. As modificações envolvem o intervalo inter-chegadas de veículos e os parâmetros elétricos. Para o intervalo inter-chegadas, entre cada sessão de carregamento, é determinado de forma estocástica por meio de uma distribuição exponencial. No que se refere aos parâmetros elétricos do carregamento AC e das necessidades de carregamento foram adicionados valores baseados numa distribuição gaussiana. Os Códigos 3.1 e 3.2 apresentam as modificações realizadas.

```

1 import numpy as np
2
3 def simulate_next_ev_arrival(arrival_rate):
4     """
5     Simulate the waiting time until the next EV arrival based on the given \
        arrival rate (lambda).
6
7     Parameters:
8     arrival_rate (float): The average arrival rate (lambda) of EVs per unit \
        time.
9
10    Returns:
11    float: The time until the next EV arrival.
12    """
13    waiting_time = np.random.exponential(1 / arrival_rate)
14    return waiting_time
15
16 # Example usage
17 #Arrival rate closer to 0.0 will have bigger inter-arrival intervals
18 #Arrival rate closer to 1.0 will have lower inter-arrival intervals
19 arrival_rate = 0.1 # Average arrival rate of EVs per unit time
20
21 # Simulate the waiting time until the next EV arrival
22 next_arrival_time = simulate_next_ev_arrival(arrival_rate)
23 print(f"Waiting time until the next EV arrival: {next_arrival_time:.2f} seconds\
    ")

```

Listing 3.1: Método para simular tempo de chegadas de EVs

```

1 def simulate_gaussian_distribution(mean, stddev, lower_bound, upper_bound):
2     value = random.gauss(mean, stddev)
3     return max(lower_bound, min(value, upper_bound))
4
5 if (await self.get_energy_transfer_mode(protocol)).startswith("AC"):
6     e_amount_value = simulate_gaussian_distribution(60, 10, 30, 90) # \
7         Simulating energy amount in kWh
8     ev_max_voltage_value = simulate_gaussian_distribution(230, 50, 220, 250) # \
9         Simulating voltage in volts
10    ev_max_current_value = simulate_gaussian_distribution(32000, 5000, 20000, \
11        32766) # Simulating current in milliamps
12    ev_min_current_value = simulate_gaussian_distribution(6, 3, 0, 10) # \
13        Simulating min current in amps

```

Listing 3.2: Adição de distribuição gaussiana para os parâmetros nas sessões de carregamento

3.1.2 Sistema distribuído de comunicação

No documento da norma OCPP na Parte 2 citam-se as topologias que o protocolo OCPP suporta, como apresentado no Capítulo 2. Entre essas topologias, foi utilizado um conceito diferente: o servidor OCPP e suas funcionalidades estão presentes dentro dos carregadores. Assim, os servidores estão distribuídos por todos os carregadores, mas apenas um único carregador tem o servidor OCPP ativo. Este servidor ativo gere os outros carregadores incluídos na rede por meio de mensagens trocadas entre os nós da rede.

Na Figura 3.7 é possível observar o comportamento do sistema após a eleição de um líder usando o mecanismo Raft: assim que o líder é definido como servidor OCPP e se torna ativo para que as conexões com os clientes sejam estabelecidas. Dessa forma, as sessões de carregamento são geridas através das próprias unidades de carregamento. Todas as unidades de carregamento terão a capacidade de se tornarem gestoras das sessões de carregamento, mas somente uma unidade de carregamento será o gestor.

Comunicação no caso de uso implementado

Para a gestão das sessões de carregamento existem diferentes casos de uso, que dependem do meio de autenticação do utilizador, se o sistema exige que o utilizador conecte primeiro o veículo para permitir o início do processo, ou se existe detecção de quando um veículo se estaciona a frente do carregador, o tipo de carregamento que será realizado, se existem outros veículos conectados no momento, a forma de pagamento, dentre outros. Cada um desses fatores direciona

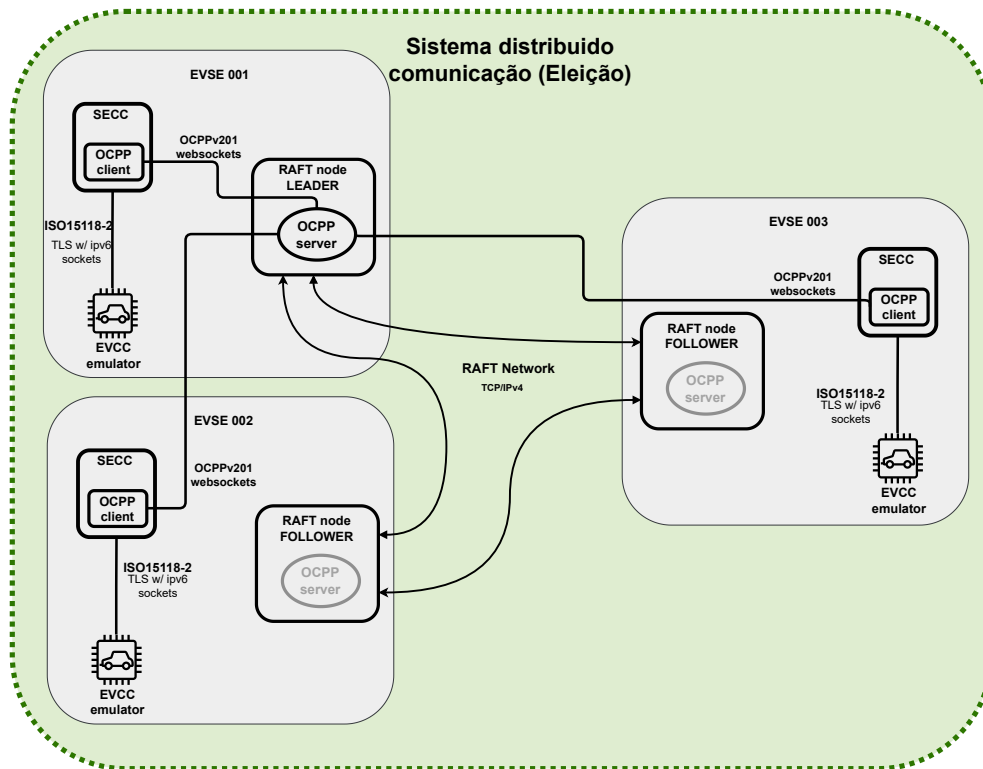


Figura 3.7: Comunicação entre as unidades de carregamento no sistema proposto

o processo para uma diferente sequência de eventos resultando assim em diferentes casos de uso, para cobrir todos os casos de uso é preciso uma implementação mais robusta e uma integração entre sistemas externos e internos bem coordenada. No âmbito deste trabalho, foi definido um caso de uso comum para o carregamento AC. As Figuras 3.8 a 3.11 representam, por meio da descrição da sequência de eventos, a interação de todos os atores envolvidos no processo.

A Figura 3.8 representa o início do processo de uma sessão de carregamento, no qual é necessário que a estação de carregamento, *OCPP_CLIENT*, estabeleça comunicação com o servidor OCPP (*OCPP_SERVER*) através do envio da mensagem *BootNotification(StartUp)*. Se o servidor responde com uma mensagem de aceitação (*Accepted*), a conexão é estabelecida, e o cliente passa a enviar mensagens de confirmação, chamadas Heartbeats, no intervalo de tempo definido pelo servidor OCPP.

Assim, o próximo evento que pode desencadear todo o processo é a conexão de um veículo elétrico. Isso faz com que o EVCC inicie o processo chamado SDP (*SECC Discovery Protocol*), definido pela norma ISO 15118, que envolve o envio de mensagens UDP para um canal *broadcast*, conforme especificado pela norma. Um SECC configurado corretamente tem a capacidade de receber e processar esse pedido, enviando uma resposta com o seu endereço e porta para comunicação. Em seguida, o SECC atualiza internamente os seus dados para indicar a presença de um veículo elétrico, e o cliente OCPP envia uma mensagem de *StatusNotification* ao servidor OCPP, informando a mudança do estado de conexão daquela estação de carregamento.

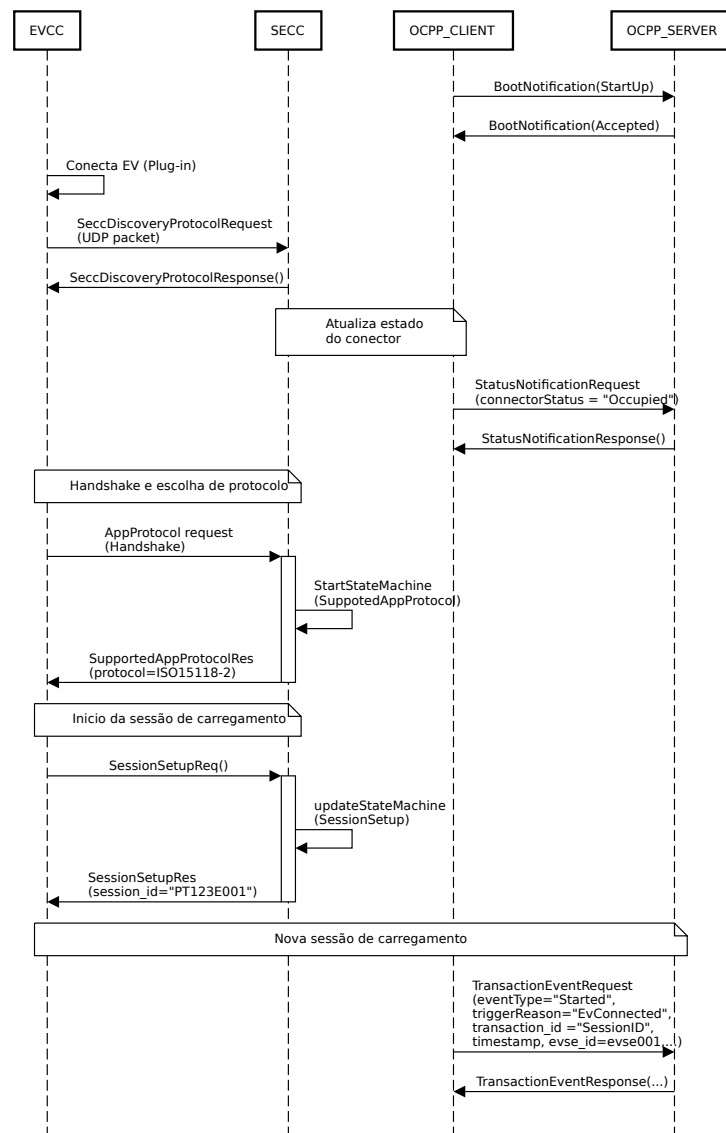


Figura 3.8: Diagrama de sequência de caso de uso AC - Início da sessão de carregamento

Após esse processo inicia-se o processo de comunicação da norma ISO 15118 com a mensagem SupportedAppProtocol que envolve o envio do veículo com uma lista de protocolos de comunicação que suporta, como, por exemplo, ISO 15118-2 ou ISO 15118-20. Após isso, o carregador envia uma resposta com o protocolo selecionado, as próximas mensagens devem estar em conformidade com o protocolo escolhido.

A mensagem SessionSetup estabelece o identificador da sessão que será utilizado em todas as próximas mensagens que serão trocadas durante a sessão atual, assim que a sessão é estabelecida com sucesso a mensagem TransactionEvent permite que o servidor seja atualizado.

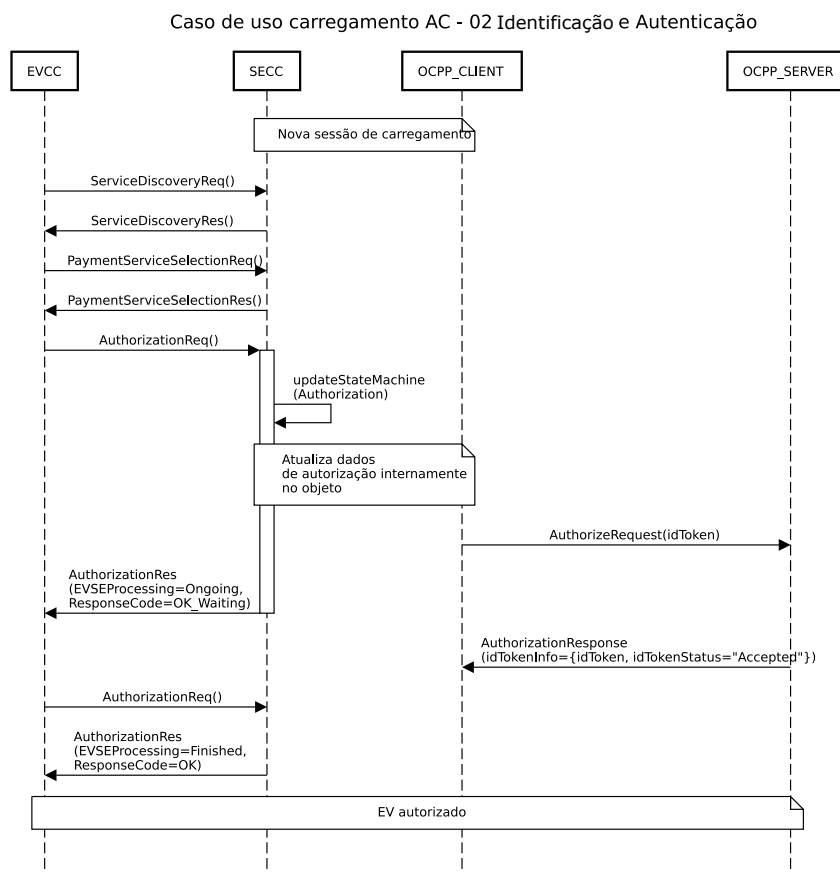


Figura 3.9: Diagrama de sequência de caso de uso AC - Processo de autenticação e serviços

Na Figura 3.9 a mensagem de *ServiceDiscovery* permite identificar quais serviços são fornecidos pelo carregador atual, a mensagem *PaymentServiceSelection* realiza a seleção da forma de pagamento que será utilizado, o método escolhido nesse processo é o carregamento grátis que não exige dados para pagamento como cartão de crédito ou certificados inteligentes que a norma suporta. Já a mensagem de *Authorization* permite a autenticação da sessão de carregamento, diferentes métodos podem ser utilizados nessa etapa como cartão RFID, cartão de crédito somente para autenticação ou até mesmo uma assinatura eletrônica provida dos certificados. O método escolhido para simplificação do processo foi a criação de *idtokens* que são confirmados do lado do servidor OCPP. Se o servidor OCPP não responder a tempo da resposta da mensagem de pedido do EVCC o SECC responde com o parâmetro *EVSEProcessing* igual a *Ongoing* e o veículo continuará a enviar mensagens de *Autorization* permitindo assim que o processo continue sem falhar a comunicação.

Na Figura 3.10 são representados os eventos relacionados a negociação energética necessária para o início do carregamento, nesta etapa o EV envia a mensagem de *ChargeParameterDiscovery* que informa ao carregador a quantidade de energia solicitada durante a sessão atual (*Eamount*) para além dos parâmetros elétricos máximos e mínimos que o veículo suporta, o carregador envia

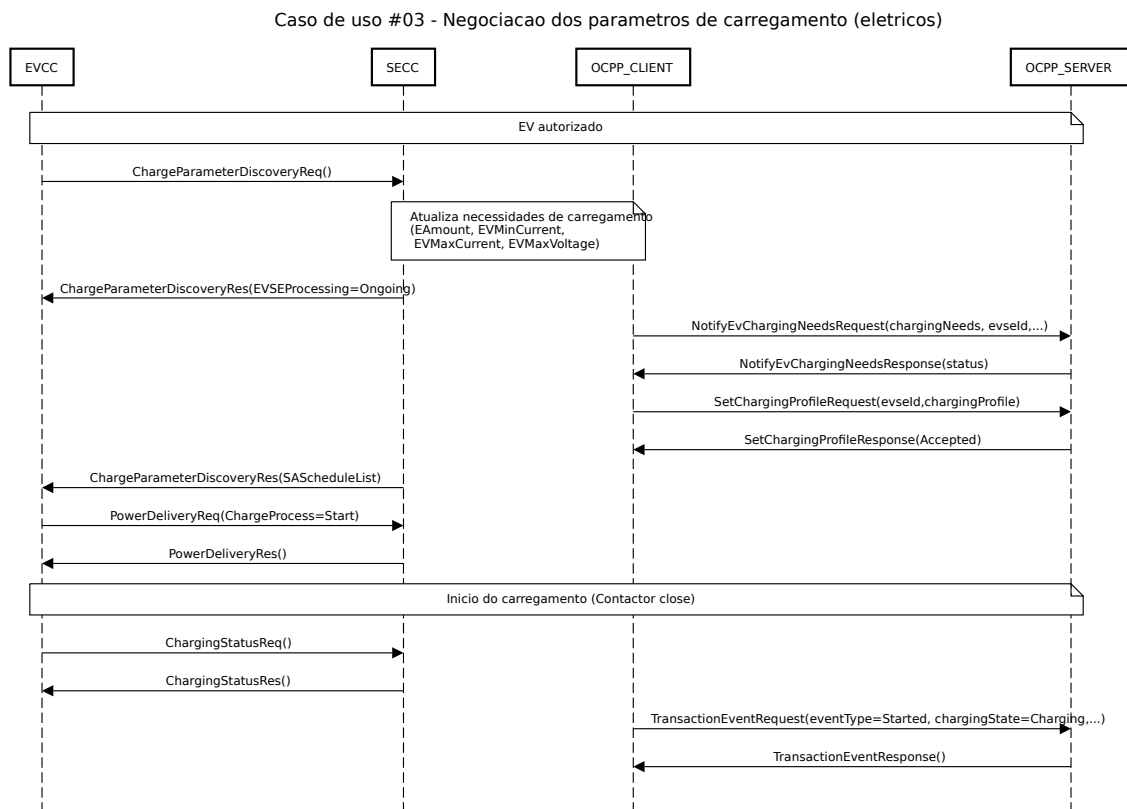


Figura 3.10: Diagrama de sequência de caso de uso AC - Negociação energética e agendamento de carregamento

esses dados ao servidor OCPP que processa e calcula um agendamento de carregamento adequado para o momento. O cálculo do agendamento de carregamento é um ramo de investigação extenso no qual permite otimizar e realizar serviços auxiliares a rede, porém não é o âmbito do trabalho atual cobrir as estratégias utilizadas nesse método, para isso é utilizado somente uma função de programação linear que obtém os parâmetros enviados pelo veículo com uma matriz de custos horária retirada do histórico de preços horários de Portugal de um dia estocástico para permitir um agendamento de carregamento dinâmico a cada nova sessão de carregamento, a fim de realizar uma aproximação fiel a casos de uso reais. Após o recebimento do agendamento de carregamento o veículo inicia o processo de carregamento. Durante o processo de transferência de energia, o veículo permanece a enviar a mensagem ChargingStatus periodicamente até o fim da sessão de carregamento.

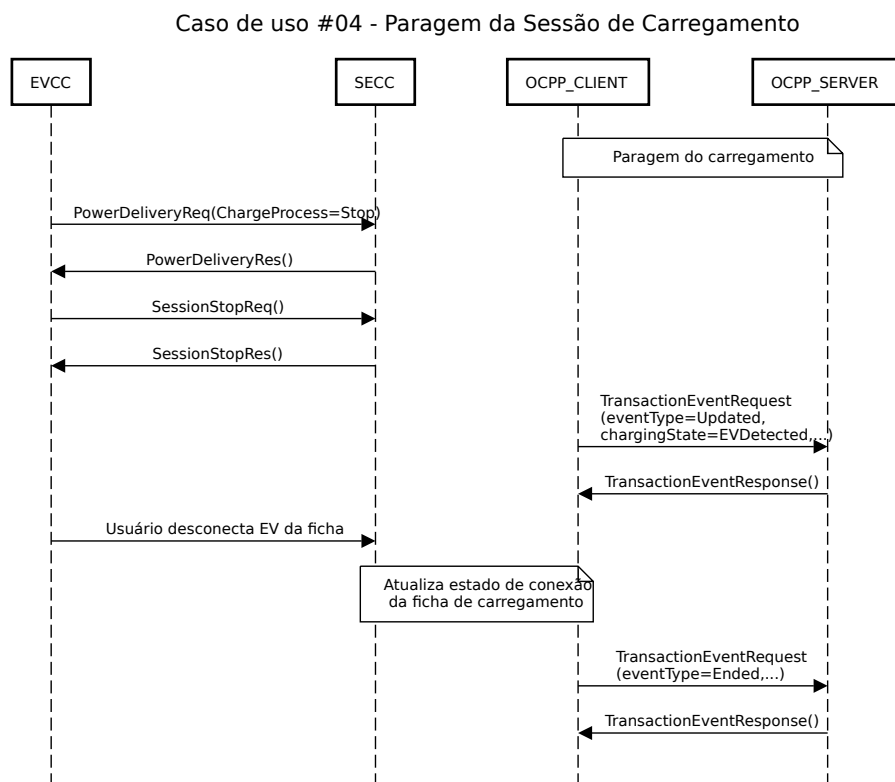


Figura 3.11: Diagrama de sequência de caso de uso AC - Paragem da sessão de carregamento

Na Figura 3.11 a mensagem de *PowerDelivery* é responsável pela finalização da transferência de energia, essa mesma mensagem deve ser enviada previamente ao início desse processo com o valor da variável *ChargeProgress* adequado ao evento que reproduz, pode-se tomar os valores de Start, Stop e Pause. A pausa na transferência de energia serve para o processo de renegociação das sessões de carregamento, que também é um caso de uso importante para as sessões de carregamento para o caso de, por exemplo, entrada e saída de outros veículos das estações de carregamento pode afetar a potência disponível para transferência. A mensagem *SessionStop* finaliza a sessão de carregamento e libera o conector da ficha do veículo para ser retirada pelo utilizador.

3.2 Mecanismo de consenso

Segundo o recente estudo de Pedro J. [30], uma comparação entre algoritmos de consenso foi realizada, mediante diferentes cenários de testes onde em cada um a única variável era o algoritmo de consenso, nesse estudo foi comparado os algoritmos Raft, Clique, IBTF 1.0 e QBFT através da plataforma *blockchain*, designada por *Quorum*. Essa plataforma estende o *Ethereum* que utiliza uma técnica chamada *proof-of-stake*, e substitui por um mecanismo chamado *proof-of-authority*, tais conceitos não são o âmbito deste trabalho, por isso não serão explorados, o escopo está mais direcionado para a componente do mecanismo de consenso.

O trabalho realizado por Pedro J. visa avaliar o desempenho dos algoritmos e o algoritmo que obteve o melhor desempenho foi o Raft, considerando para transações públicas ou privadas, em seguida vieram o QBFT e o IBFT, com desempenhos similares e o algoritmo Clique foi o que obteve o pior desempenho. [30]

Baseado nos estudos e na facilidade de compreensão e implementação do Raft em comparação a outros algoritmos de consenso foi escolhido esse algoritmo para teste na infraestrutura de carregamento de VEs, por meio da análise e desempenho do Raft encontrado na literatura a sua utilização será uma forma de validar trabalhos existentes.

O Raft, como explicado no capítulo 2, implementa uma máquina de estados replicável, que por essa razão vem sendo utilizado em muitas implementações de base de dados distribuídas. Neste trabalho essa característica visa ser aproveitada para que cada carregador de VEs que estejam incluídos na mesma rede de consenso conheçam o estado atual dos outros carregadores à sua volta. Dessa forma, podem realizar transações e até mesmo agendamentos de carregamento a partir dos dados do grupo que são partilhados e sincronizados de forma iterativa.

Para implementar o algoritmo aplicado a infraestrutura de veículos elétricos, a estratégia encontrada foi procurar bibliotecas que implementem as características e regras definidas pelo Raft, para que, sobre tal biblioteca, o sistema fosse desenvolvido. Alguns fatores importantes deveriam ser considerados para a escolha da biblioteca a ser utilizada, primeiramente a linguagem em Python foi escolhida pela sua versatilidade e por já estar a ser usada em outros processos utilizados nos carregadores. Como fator comparativo entre as bibliotecas, foi considerado, que implementem as seguintes funcionalidades Raft:

- Eleição de líder e Replicação dos Logs
- Persistência
- Mudança dinâmica de membros
- Compressão dos logs

A lista de bibliotecas encontradas na linguagem selecionada foi:

A seleção das bibliotecas para implementação da rede de consenso foi realizada com base em diversos critérios, visando garantir a qualidade e adequação ao caso de uso pretendido. Os critérios considerados foram:

- **Suporte:** Nível de suporte técnico disponível, incluindo documentação, fóruns e comunidade de desenvolvedores.
- **Popularidade:** Número de estrelas de recomendação no GitHub, como indicador da relevância e utilização da biblioteca.
- **Atividade de Desenvolvimento:** Tempo decorrido desde o último *commit* do repositório, indicando a frequência de atualizações e correções de *bugs*.

Tabela 3.1: Seleção de bibliotecas que implementam Raft em Python

Nome	Referência	Licença	Eleição de líder Replicação dos logs	Persistência	Mudança dinâmica de membros	Compressão dos logs
PySyncObj	[29]	MIT	Sim	Sim	Sim	Sim
Zatt	[3]	AGPL-3.0	Sim	Sim	Sim	Sim
raftos	[45]	MIT	Sim	-	Não	Não
simpleRaft	[33]	MIT	Não	-	Não	Não
Vesper	[25]	MIT	Sim	-	Sim	Não
raft	[42]	Apache-2.0	Sim	Sim	Sim	Sim
aioraft	[20]	AGPL-3.0+	Não	Não	Sim	Não
py-raft	[8]	Unlicense	Não	Não	Sim	Não
Chillaxd	[43]	Apache-2.0	Sim	-	Não	Não

- **Resolução de Issues:** Eficiência na resposta e resolução de problemas reportados na plataforma GitHub.
- **Compatibilidade:** Adequação da biblioteca ao caso de uso específico, considerando as funcionalidades e características necessárias.

Para além dos critérios mencionados, a seleção também considerou a cobertura das seguintes funcionalidades:

- **Eleição de Líder:** Mecanismo para determinar o nó responsável pela coordenação da rede em um determinado momento.
- **Replicação de Logs:** Distribuição segura e confiável dos logs entre os nós da rede.
- **Persistência:** Armazenamento permanente dos dados da rede, garantindo sua disponibilidade mesmo em caso de falhas.
- **Mudança Dinâmica de Membros:** Capacidade de adicionar ou remover nós da rede de forma dinâmica, sem interromper o funcionamento do sistema.
- **Compressão de Logs:** Redução do tamanho dos logs para otimizar o uso de recursos de armazenamento e rede.

A escolha das bibliotecas foi realizada considerando os critérios e funcionalidades mencionados. A biblioteca PySyncObj foi selecionada inicialmente por apresentar a melhor combinação de características para o caso de uso específico, mas posteriormente apresentou muitas dificuldades para integração com o sistema e por isso foi utilizada a biblioteca raftos com as devidas modificações. [45]

A biblioteca raftos é desenvolvida em Python destinada à construção de sistemas distribuídos tolerantes a falhas. Disponibiliza a capacidade de replicar os dados de uma aplicação entre múltiplos servidores, possuindo as características de protocolo Raft para eleição de líder e replicação

dos *logs*, permite a execução do ciclo de eventos em diferentes tarefas de forma assíncrona, pois utiliza a biblioteca *asyncio* como dependência, interface amigável ao desenvolvedor permitindo a replicação de dados em diferentes estruturas de dados, como, por exemplo, listas e dicionários.

3.3 Sumário

Neste capítulo, detalhamos o sistema proposto para a gestão coordenada das sessões de carregamento de veículos elétricos, destacando a inovação e eficiência introduzidas. A integração das normas ISO 15118 e OCPP, juntamente como mecanismo de consenso Raft, proporciona uma solução robusta e eficiente para os desafios atuais da infraestrutura de carregamento. A implementação da rede de consenso Raft como uma camada adicional demonstra o potencial das unidades de carregamento de gerir autonomamente as sessões de carregamento assegurando a robustez e a fiabilidade do sistema.

A infraestrutura de rede descrita permite uma comunicação eficiente entre as unidades de carregamento, suportada por uma plataforma distribuída que adapta e modifica os protocolos OCPP e ISO 15118-2 para os testes experimentais. A gestão distribuída do servidor OCPP e a eleição de um líder utilizando Raft exemplificam a capacidade do sistema de se adaptar a diferentes cenários de operação, garantindo um serviço de carregamento eficiente e fiável.

Concluimos que o sistema proposto representa uma abordagem inovadora para a gestão de carregamento de veículos elétricos, oferecendo uma solução flexível que aumenta a tolerância a falhas do sistema e pode ser implementada em diversas infraestruturas de carregamento.

Capítulo 4

Avaliação experimental

Neste capítulo são apresentados os resultados dos testes realizados em ambiente simulado e real, verifica-se o impacto do mecanismo de consenso em carregadores reais enquanto os mesmos operam normalmente.

Como demonstrado no estudo [36] é preciso validar o sistema proposto utilizando dados extraídos entre os nós distribuídos, para isso foi programado testes que permitem obter em cada nó obter os dados desejados, em seguida uma comparação entre os dados obtidos é feita. A título de comparação entre os dados, os testes foram realizados em diferentes cenários e assim foram divididos em testes inter-processos e inter-dispositivos.

Os testes inter-processos são realizados num mesmo dispositivo, mas em diferentes processos, executados paralelamente. Já os testes inter-dispositivos são realizados na infraestrutura de carregamento do instituto de investigação INESC TEC.

O objetivo é avaliar o impacto da integração dessa tecnologia no desempenho e confiabilidade do sistema, considerando diferentes ambientes de operação. Avaliando assim, a viabilidade da rede de consenso Raft que foi implementada como uma camada adicional sobre a arquitetura padrão.

4.1 Unidades de Carregamento

Nesta secção apresentam-se as especificações físicas e virtuais dos carregadores, para posteriormente ser apresentado o sistema proposto que utiliza as unidades de carregamento. A Figura 3.1 ilustra as funções essenciais para o carregamento de veículos elétricos, as quais asseguram a monitorização e o controlo das sessões de carregamento. Essas funções foram desenvolvidas especificamente para a infraestrutura em questão, entretanto neste trabalho o sistema proposto irá realizar a gestão das sessões de carregamento de acordo com normas da indústria.

Para identificar o estado da ligação entre o veículo e o carregador, o EVSE tem de gerar um sinal PWM (modulação por largura de pulso) de frequência fixa e, simultaneamente, ler a amplitude desse sinal para atualizar internamente o estado atual da ligação. Isso permite que o sistema controle o relé que permite ou interrompe a transferência de energia entre a rede e



Figura 4.1: Unidade de carregamento EVSE nas instalações externa da INESC TEC (inescevse010)



Figura 4.2: Unidade de carregamento EVSE utilizadas nos testes no SGEVL, INESC TEC (inescevse005 e 006)

o veículo durante o carregamento AC. Esse processo requer um módulo responsável pelo sinal PWM, denominado sinal CP *Control Pilot* pela norma IEC 61851.

Para realizar a monitorização do consumo energético sem a necessidade de comunicação direta com o veículo elétrico, é necessário adicionar um módulo de medição de energia. Este módulo permite a extração de valores de tensão, corrente, energia, potência e frequência. Os dados obtidos são extraídos e enviados para um servidor externo por meio de uma API (*Application programming interface*), que realiza a interface do sistema interno através do método HTTPs POST para transmitir as informações a um servidor previamente configurado.

A API em HTTP também permite a adição de um serviço de autenticação, realizado com o auxílio de uma aplicação desenvolvida localmente para a autenticação dos utilizadores. Somente após a autenticação é liberada a potência para o carregamento do veículo elétrico. Este mecanismo foi customizado para a infraestrutura de carregamento da INESC TEC. Além disso, a conexão com

o servidor externo permite a parada e reinício do carregamento durante as sessões, bem como a limitação dinâmica da potência máxima em cada unidade de carregamento.

Todas essas funcionalidades visam aumentar o nível de controlo das sessões de carregamento, permitindo a implementação de futuras estratégias de investigação. Essas estratégias visam otimizar as sessões de carregamento conforme apresentado no capítulo 2. Para cada aplicação, pode-se realizar uma otimização dos recursos energéticos direcionada a diferentes atores, como, por exemplo, o utilizador, a rede elétrica ou o operador de rede.

4.1.1 Módulo computacional

O módulo computacional é responsável pelo processamento dos dados provenientes dos módulos de controlo de carga e medição de energia. Neste módulo, são processados os dados do sinal piloto (Control Pilot) e do medidor de energia para diferentes finalidades. Para esta tarefa, utiliza-se um computador em placa única, conhecido como Raspberry Pi.

Utilizando o módulo computacional, *Raspberry Pi*, as informações obtidas são armazenadas numa base de dados local, e posteriormente enviadas para um servidor remoto. Cada carregador possui uma interface de programação de aplicações (API) que possibilita receber comandos e solicitações do servidor, permitindo o controlo da potência, o início e a pausa do carregamento, além do processo de autenticação de utilizadores, realizado com o suporte de uma aplicação externa.

As especificações do Raspberry Pi utilizado nos carregadores estão detalhadas na Tabela 4.1.

Tabela 4.1: Especificação computacional das unidades de carregamento

Aspeto	Especificação
Modelo	Raspberry Pi 3b
Clock da CPU	4 Núcleos de 1,2 GHz
Memória RAM	1 GB
Cartão de Memória(ROM)	32 GB ultrarrápido
Sistema Operativo	RaspbianOS 32 bits
Arquitetura do processador	ARMv7

4.1.2 Desempenho computacional

Para avaliar o desempenho computacional do Raspberry Pi, foram conduzidos ensaios considerando o seu funcionamento durante a execução de funções essenciais no contexto do carregamento de veículos elétricos. Os ensaios realizados envolveram a monitorização periódica do consumo da CPU e do uso da memória volátil como meios de comparação para os ensaios posteriores realizados com a rede de consenso.

4.1.3 Serviços do sistema operativo

Dentro de cada unidade de carregamento, múltiplos serviços são executados simultaneamente para desempenhar as funções do próprio sistema operativo, além disso, foram criados serviços que

inicializam automaticamente para ser possível o funcionamento adequado das sessões de carregamento. Estes serviços são organizados para interagirem entre si em diferentes etapas durante o processo de carregamento. Cada serviço é um processo independente executado dentro do sistema operativo, com um identificador único do processo (PID).

Tabela 4.2: Serviços e processos executados pelo SO nas unidades de carregamento

Serviço	Descrição
iec61851_charger	Leitura de sinais, controlo de sinal CP, medidor de energia, escrita e leitura da base de dados
evse_api	Servidor REST API que habilita o acesso por meio de endpoints HTTP
mariadb	Executa a gestão e acesso à base de dados do sistema

- O serviço "iec61851_charger" é responsável por diferentes tarefas, incluindo a leitura de sinais, o controlo do sinal piloto (CP), a medição de energia e a manipulação da base de dados, que envolve a escrita e leitura de informações.
- O serviço "evse_api" implementa um servidor REST API que permite acesso aos recursos do carregador por meio de endpoints HTTP, fornecendo uma interface para interação com o sistema.
- O serviço "mariadb" é responsável pela gestão e acesso à base de dados do sistema, possibilitando o armazenamento e recuperação eficientes das informações relevantes para o funcionamento do carregador.

Relativamente às configurações do sistema operativo, a maioria das unidades de carregamento utiliza o RaspbianOS Lite, uma distribuição do sistema operativo Raspbian baseado em Linux, com uma arquitetura de 32 bits.

4.1.4 Componentes eletrónicos

Para que as aplicações informáticas sejam viáveis em sistemas de carregamento de veículos elétricos (VE), é necessário a integração de componentes eletrónicos e físicos, cada qual desempenhando funções específicas e imprescindíveis para o funcionamento do sistema. Os componentes eletrónicos são tipicamente montados numa placa de circuito impresso (PCB), que atua como o núcleo do sistema, interligando e suportando os diversos módulos eletrónicos. Nesta secção, serão detalhados os módulos que compõem a PCB nas unidades de carregamento utilizadas.

Os componentes eletrónicos nas unidades de carregamento viabilizam o processamento de sinais, a comunicação entre o VE e o carregador, o controlo da transferência de energia pelo relé, a indicação visual do estado do carregador para o utilizador por meio de código de cores, o processamento e a computação de todas as informações referentes ao carregador e o estabelecimento de comunicação do EVSE com agregadores ou outros atores. Em termos técnicos, a PCB é composta

por diversos componentes eletrônicos, como microcomputadores (SBC), sensores, transdutores, *drivers* e interfaces de comunicação. Esses elementos trabalham em conjunto para garantir o funcionamento adequado de cada unidade de carregamento.

Alimentação da placa

A energia da rede elétrica (230V CA) necessita ser convertida para tensões mais baixas (3,3V, 5V, 12V CC) para alimentar os componentes eletrônicos da placa de circuito impresso.

Adicionalmente, é necessário empregar módulos de conversão para transformar a tensão de 12V em -12V, essenciais para suportar o sinal piloto (Control Pilot). Este processo de conversão de tensão é crucial para garantir a compatibilidade entre fonte de energia disponível e as necessidades específicas dos componentes eletrônicos.

Aquisição do sinal do Control Pilot (CP)

Conforme mencionado na Secção 2.1.4.1, o sinal necessário para a comunicação entre o veículo elétrico (VE) e o Equipamento de Carregamento de Veículos Elétricos (EVSE) é um sinal de PWM com amplitude entre 12V e -12V. Este sinal é originado pelo módulo PWM na saída dos pinos da Raspberry Pi.

Inicialmente, o valor do sinal gerado pela Raspberry Pi situa-se entre 3V e -3V. Para ajustar a amplitude do sinal para os valores nominais de saída entre 12V e -12v, é necessário utilizar um amplificador de precisão com capacidade de operação *rail-to-rail*. Este amplificador é responsável por garantir que a amplitude do sinal esteja dentro dos limites requeridos pela norma em questão. Este processo é crucial para assegurar que o sinal esteja conforme as exigências técnicas e de segurança estabelecidas pela norma aplicável.

Módulo de medição de energia

O módulo de medição de energia permite a obtenção de informações detalhadas sobre a potência consumida durante as sessões de carregamento, o consumo total de energia e a monitorização em tempo real das características da rede elétrica. Essa capacidade de monitorização em tempo real possibilita uma compreensão mais completa do que está a ocorrer durante as sessões de carregamento.

Este módulo realiza medições de diversos parâmetros elétricos, incluindo tensão nominal, corrente, potência, energia e frequência, com uma taxa de amostragem na ordem dos milissegundos. Os valores medidos são então processados para calcular médias e registados periodicamente numa base de dados, com uma resolução de 5 minutos entre registos. Além disso, esses dados são enviados para um gestor de carregamento, permitindo uma análise mais abrangente e a geração de relatórios sobre o desempenho do sistema de carregamento.

4.2 Arquitetura de testes

Conforme o modelo OSI, os módulos SECC e EVCC idealmente abrangem todas as camadas, da aplicação à física. No entanto, para otimizar os testes e análises, propõe-se uma modelação do sistema, focando nas camadas superiores à de ligação. Esta decisão se baseia na premissa de que as camadas físicas e de enlace já foram exaustivamente testadas e validadas em trabalhos anteriores, sendo consideradas estáveis para esta investigação.

Os testes propostos visam avaliar a disponibilidade, o desempenho e a carga computacional do sistema. A disponibilidade será medida pelo tempo em que o servidor permanece ativo num único nó e pela taxa de sucesso das sessões de carregamento que são inicializadas a partir da conexão de novos VEs. O desempenho será quantificado pela latência e débito das mensagens Raft e OCPP, trocadas via websockets entre cliente e servidor. A carga computacional será medida pelo consumo médio de CPU e memória durante as experiências.

Com base nesses indicadores, pretende-se aferir a viabilidade do sistema proposto, explorando uma estratégia distribuída e flexível para a gestão de sessões de carregamento coordenadas.

4.2.1 Cenário de sessão de carregamento

Para as experiências, foi escolhido um cenário de sessão de carregamento coordenado baseado em carregamento AC com autenticação externa por meio de um agregador (o servidor OCPP). O serviço simulado calcula o agendamento de carregamento com base nas necessidades energéticas do veículo elétrico e os seus valores nominais, além dos custos horários de energia, determinando a potência máxima para cada hora. Esses custos, que servem de incentivo para o algoritmo, poderiam ser gerados localmente ou recebidos no operador de rede, mas a análise detalhada de estratégias de agendamento não faz parte do escopo deste trabalho. O objetivo é simular um cálculo de agendamento com valores coerentes de dados que reflitam um caso de uso real.

Durante a troca de mensagens entre os servidores Raft em cada unidade de carregamento, estes operarão normalmente com as suas funções essenciais, conforme representado na Figura 3.4.

As unidades de carregamento apresentam *hardware* e configuração de *software* semelhantes, com certa homogeneidade entre os dispositivos. As únicas diferenças identificadas são a versão do kernel em duas unidades e a utilização do armazenamento interno no momento do ensaio. A Tabela 4.3 detalha as características de cada unidade de carregamento, é configurada uma porta de rede comum em todos os carregadores para comunicação.

4.2.2 Parâmetros do protocolo Raft

A biblioteca Raft, utilizada para implementar o mecanismo de consenso Raft, possui parâmetros que regem o seu funcionamento em sistemas distribuídos, conforme apresentado na Tabela 4.4. O ajuste adequado destes parâmetros é crucial para o desempenho e estabilidade do sistema, dependendo das características da rede e do número de nós envolvidos.

Tabela 4.3: Configuração e parâmetros dos carregadores

Identificador	Armazenamento interno	Kernel
inescevse001	21%	Linux 6.1.21-v7+ armv7l
inescevse002	21%	Linux 6.1.21-v7+ armv7l
inescevse003	38%	Linux 5.15.76-v7+ armv7l
inescevse004	21%	Linux 6.1.21-v7+ armv7l
inescevse006	46%	Linux 5.15.76-v7+ armv7l
inescevse007	37%	Linux 6.1.21-v7+ armv7l
inescevse008	20%	Linux 6.1.21-v7+ armv7l
inescevse009	15%	Linux 6.1.21-v7+ armv7l
inescevse010	15%	Linux 6.1.21-v7+ armv7l

Nas experiências realizadas, optou-se por utilizar os mesmos parâmetros em todos os cenários, visando uma comparação mais justa entre os resultados. No entanto, é importante salientar que estes parâmetros podem ser ajustados para se adequarem a redes com diferentes capacidades. Em redes menos robustas, por exemplo, parâmetros mais relaxados podem ser necessários, mas é fundamental garantir que o sistema ainda seja capaz de atingir o consenso. A definição dos valores ideais para cada parâmetro deve ser realizada de forma experimental, considerando as particularidades de cada sistema.

Tabela 4.4: Parâmetros do mecanismo Raft

Propriedade	Valor	Descrição
Heartbeat interval Raft	100 ms	Intervalo em que líder deve enviar mensagens <i>append entries</i>
Step down missed heartbeats (Contador)	5	Quantidade de <i>heartbeats</i> não respondidos para o líder deixar de ser líder
Election interval Spread	3	Determina tamanho do intervalo da aleatoriedade dos <i>timeouts</i> de recebimento dos <i>heartbeats</i>

4.2.3 Primeiro ensaio - Sistema centralizado e inter-processos

Para realizar uma comparação dos valores obtidos, os mesmos ensaios foram realizados em ambiente local em um computador com as características apresentadas na Tabela 4.5. O objetivo da realização dos ensaios em ambiente local serve para eliminar a componente da rede de comunicação física que as unidades de carregamento utilizam para comunicar entre si e também para verificar o desempenho num dispositivo com recursos computacionais mais elevados.

A experiência foi realizada no sistema operativo Windows 11, mas no ambiente *Windows Subsystem for Linux (wsl)* em que a memória disponível RAM é de 7.6 GB e 10 núcleos lógicos de processamento.

Tabela 4.5: Especificação computacional da máquina para inter-processos

Aspecto	Especificação
Modelo	Lenovo 21DJ
Clock da CPU	Intel Core i7-1255U, 9 núcleos físicos
Memória RAM	16 GB (7.6 GB available on wsl)
Cartão de Memória(ROM)	512 GB SSD
Sistema Operativo	Windows 11 (wsl)
Arquitetura do processador	x64

Sendo assim, os atores apresentados no capítulo 3 (cliente OCPP, SECC e EVCC) com o servidor OCPP ativo em um único processo, simulam um caso real de uso de um servidor OCPP centralizado que suporta o caso de uso de sessão de carregamento inteligente da norma ISO 15118-2. Este ensaio servirá como referência do que é possível realizar sem a latência da camada de comunicação física *Ethernet* de uma rede local, embora seja limitado pela capacidade computacional do dispositivo ao processar dados de sessões simultâneas.

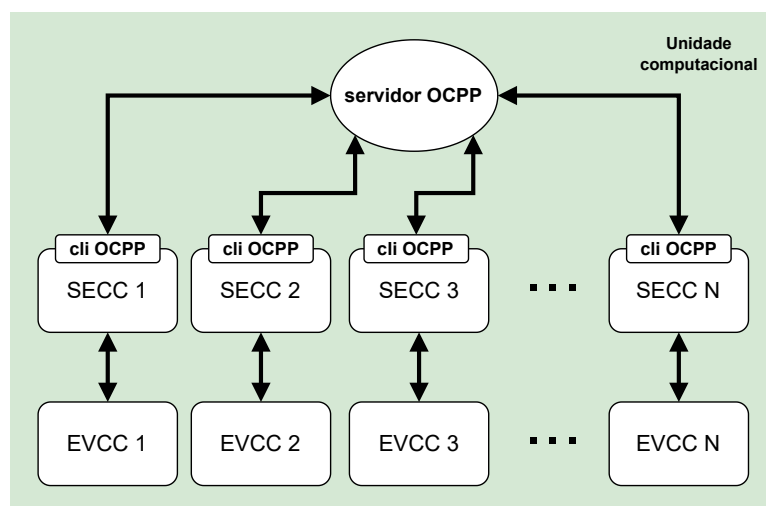


Figura 4.3: Experiência local com servidor OCPP centralizado

Conforme ilustrado na Figura 4.3, a comunicação entre os dispositivos SECC (Supply Equipment Communication Controller) e EVCC (Electric Vehicle Communication Controller) ocorre de forma individualizada, ou seja, cada SECC se comunica com um único EVCC e vice-versa. No entanto, o protocolo de comunicação ISO 15118 prevê um mecanismo de descoberta de SECCs (SECC Discovery Protocol - SDP) para que o EVCC identifique os SECCs disponíveis na rede. Este mecanismo consiste no envio de mensagens UDP em broadcast pelo EVCC, respondidas pelos SECCs com os seus respectivos endereços.

Para evitar conflitos na comunicação durante os testes locais, foi desenvolvido um *script* que configura portas diferentes para cada par SECC-EVCC no mecanismo SDP. Adicionalmente, a

biblioteca da norma ISO 15118 foi adicionada a capacidade de receber este parâmetro, permitindo a individualização da comunicação entre os dispositivos.

```
1 # run_in_parallel.py
2 import multiprocessing
3 import subprocess
4
5
6 def run_evcc(config_file, sdp_port):
7     subprocess.run(["python3", "iso15118/evcc/main.py", config_file, str(\
8         sdp_port)])
9
10 if __name__ == "__main__":
11     num_processes = 10 #Max of EVCCs simultaneously
12     processes = []
13     config_file = "iso15118/shared/examples/evcc/iso15118_2/evcc_config_eim_ac.\
14         json"
15     sdp_port = 15118
16
17     for _ in range(num_processes):
18         p_evcc = multiprocessing.Process(target=run_evcc, args=(config_file, \
19             sdp_port,))
20         p_evcc.start()
21         processes.append(p_evcc)
22         sdp_port += 1
23
24     for p in processes:
25         p.join()
```

Listing 4.1: Metodo para executar SECCs e EVCCs paralelamente

O Código 4.1 representa os processo paralelos do EVCC que são instanciados, para o SECC o script é semelhante. Esse mecanismo não é necessário para as experiências inter-dispositivos, pois em cada dispositivo terá somente uma instância do SECC e EVCC.

4.2.4 Segundo ensaio - Sistema distribuído e inter-processos

O segundo ensaio replica os mesmos inter-processos, porém, em vez de um único servidor OCPP, são instanciadas diversas réplicas em paralelo, distribuídas entre os nós Raft, onde apenas o líder se manterá ativo, conforme a proposta do sistema. Assim, obtêm-se os dados da rede Raft de consenso com a menor latência e o maior débito viável, limitados apenas pelo tempo de *runtime* intrínseco do sistema programado da linguagem de programação, para além da capacidade computacional do dispositivo utilizado na execução paralela dos processos.

A experiência foi conduzida sob diferentes configurações da rede Raft, com 3, 5, 7 e 9 nós, para estabelecer uma comparação entre os resultados obtidos.

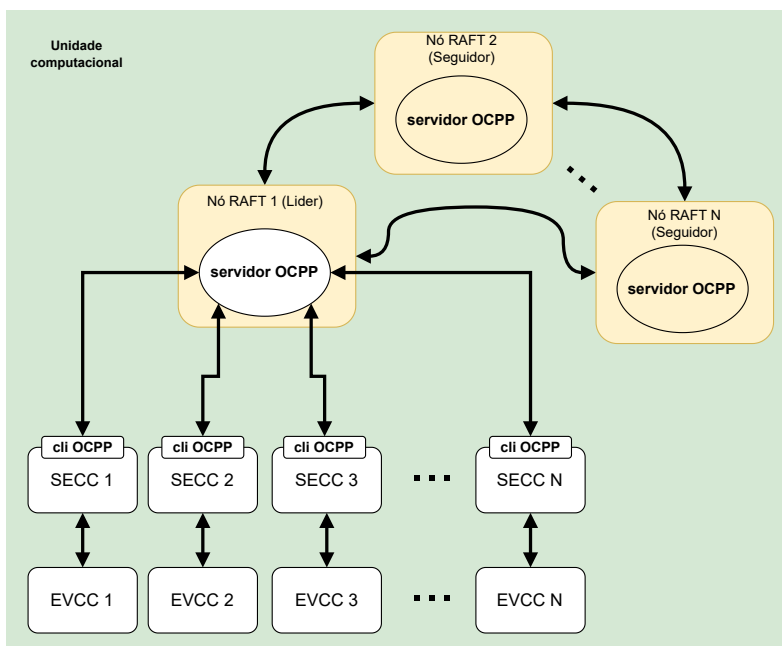


Figura 4.4: Experiência local com rede Raft (Segundo ensaio)

4.2.5 Terceiro ensaio - Sistema centralizado e inter-dispositivos

O terceiro ensaio, diferentemente dos anteriores que utilizavam processos simulados e exploravam a rede Raft, será realizado em um ambiente real, utilizando os carregadores da infraestrutura de carregamento da INESC TEC, conforme detalhado no Capítulo 4. Essa mudança permitirá avaliar o desempenho do sistema proposto em condições reais de operação, considerando as características e limitações dos carregadores físicos, como a comunicação em rede local e a capacidade de processamento de cada unidade. Além disso, será possível analisar o impacto da latência da rede e do tráfego de dados real na troca de mensagens entre os clientes OCPP e o servidor centralizado.

Neste ensaio, o foco será analisar o desempenho do sistema sem a utilização da rede Raft, concentrando-se na comunicação direta entre os clientes OCPP e o servidor centralizado. Serão avaliados os tempos de resposta, a latência da rede e a capacidade do servidor em gerir múltiplas requisições simultâneas.

4.2.6 Quarto ensaio - Sistema distribuído e inter-dispositivos

Na quarta experiência, a comunicação será estabelecida entre os carregadores reais da INESC TEC, apresentados neste capítulo, incorporando o protocolo Raft. Essa camada adicional de comunicação permitirá avaliar o desempenho do sistema em um ambiente real, considerando a coordenação e o consenso entre os nós da rede Raft.

Assim como nos ensaios anteriores, serão utilizadas diferentes quantidades de nós na rede Raft (3, 5, 7 e 9), possibilitando a comparação do desempenho sob distintas configurações. Essa análise

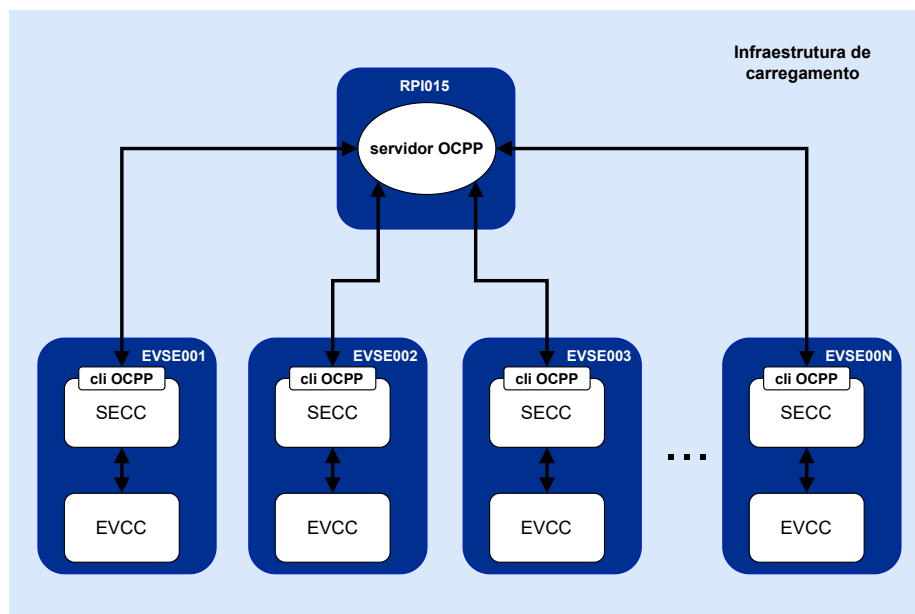


Figura 4.5: Experiência na infraestrutura com topologia centralizada (Terceiro ensaio)

permitirá identificar o impacto do número de nós na latência da comunicação, no tempo de resposta do sistema e na eficiência do consenso entre os carregadores. Além disso, será possível avaliar a escalabilidade do sistema e a sua capacidade de lidar com um número crescente de dispositivos na rede.

A biblioteca utilizada para implementar o protocolo Raft utiliza mensagens UDP com identificadores únicos, permitindo a transmissão simultânea para todos os nós. Ao receber respostas dos nós ativos, o identificador é verificado. Os nós que recebem mensagens respondem com o mesmo identificador. No mecanismo Raft, existem dois tipos de mensagens:

Append Entries: Enviadas pelo líder, podem ser vazias (servindo como heartbeats) ou conter dados a serem replicados na rede. As respostas são verificadas para confirmar se a maioria dos nós atualizou os dados com sucesso.

Request Vote: Enviadas pelo candidato durante a eleição do líder. Os seguidores apenas recebem e respondem a mensagens.

Em suma, a biblioteca Raft utiliza mensagens UDP identificadas para comunicação eficiente. As mensagens *append entries* servem para replicação de dados e heartbeats, enquanto as mensagens *request vote* são usadas na eleição do líder.

4.3 Resultados

Durante as experiências realizadas ficheiros de texto são criados automaticamente e adicionado os valores dos resultados, um segundo script converte e agrupa esses ficheiros de texto dos

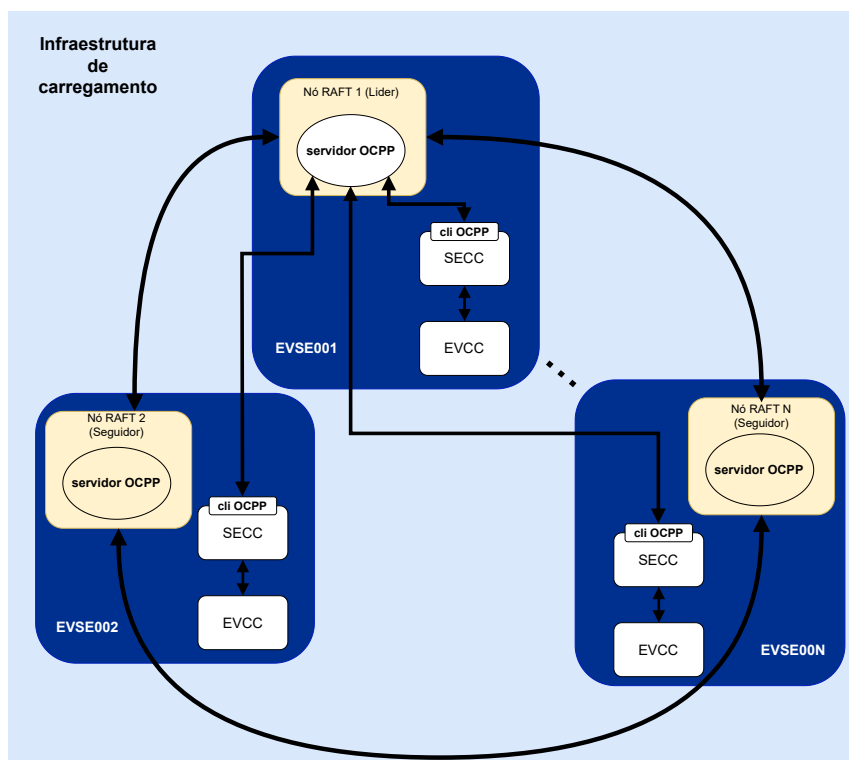


Figura 4.6: Experiência na infraestrutura com rede Raft (Quarto ensaio)

resultados que são extraídos de cada unidade de carregamento para um computador e posteriormente os ficheiros individuais são convertidos para um arquivo de texto para análise. Os resultados subdividem-se em dois aspetos, o desempenho computacional e de comunicação.

O tempo máximo para cada experiência foi fixado há 4 horas de duração para uniformização e melhor comparação dos dados obtidos. A quantidade de sessões de carregamento realizadas em cada experiência tem um valor estocástico devido à utilização do intervalo entre chegadas de VEs aleatório implementado, as experiências sem a rede distribuída envolveu 10 clientes OCPP inter-processos e 7 clientes OCPP na experiência com os carregadores.

Durante a segunda experiência, foram concluídas 132 sessões de carregamento ao longo de 4 horas, com um total máximo de 29.211 mensagens OCPP, das quais a maioria eram mensagens de *heartbeats* enviadas ao servidor OCPP. Estas mensagens servem para garantir a monitorização da disponibilidade das estações de carregamento. O ensaio envolveu 9 servidores OCPP (nós) da rede Raft. Foram utilizados 10 processos SECCs e EVCCs em paralelo para simular constantemente 10 carregadores e veículos elétricos.

4.3.1 Resultados Computacionais e da Comunicação do protocolo OCPP

As análises experimentais revelaram um incremento consistente na utilização da memória RAM disponível. Um exame pormenorizado evidenciou que, durante as sessões de carregamento, os processos do EVCC, nomeadamente o *gateway java* responsável pelo acesso ao codificador

de mensagens EXI (empregado na comunicação ISO 15118), desencadeiam um aumento substancial no uso de memória RAM. Diversas estratégias de mitigação, incluindo a restrição da *heap* do processo Java e a ativação do garbage collector, foram implementadas, contudo, sem lograr os resultados desejáveis, mas diminuindo a taxa de incremento de utilização da memória RAM.

Nas figuras apresentadas dos resultados são apresentados os dados com os valores médios obtidos em cada ensaio do consumo da CPU, memória RAM, latência e débito das mensagens OCPP e das mensagens da rede distribuída. Os valores médios não representam a mediana e sim a média aritmética de todos os valores obtidos e os valores máximos representam somente um único valor registado, dos valores mínimos foram excluídos os valores nulos, os zeros, que não expressam nenhuma conclusão para as experiências.

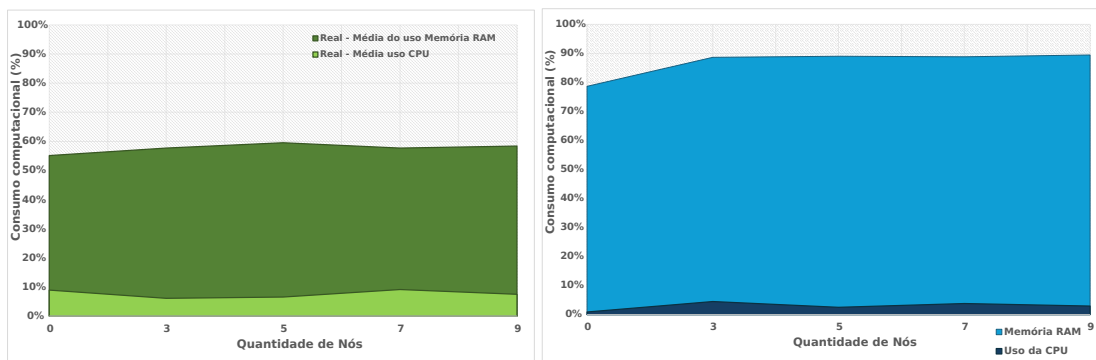
A análise dos dados recolhidos revela informações relevantes sobre o desempenho do sistema em diferentes cenários. A Figura 4.7 mostra uma maior utilização média da CPU nas unidades de carregamento relativamente ao ensaio simulado, devido à menor capacidade computacional dos carregadores. No entanto, a comparação entre o terceiro (sem Raft) e o quarto ensaio (com Raft), realizado em ambiente real, em comparação não é possível observar uma variabilidade grande entre os dados dos sistemas sem e com a rede distribuída, pode-se concluir que o impacto da rede Raft no consumo de CPU é baixo.

O elevado consumo de memória RAM, observado na Figura 4.7, não está relacionado à sobrecarga dos módulos de gestão de carregamento, mas sim a questões específicas do módulo EVCC, como já mencionado anteriormente.

A latência das mensagens numa rede distribuída deve ser baixa para evitar interferências na próxima rodada de mensagens enviadas pelo líder do grupo. Se o sistema apresentar alta latência nas mensagens da rede Raft, o líder pode interpretar algumas mensagens como não respondidas pelos nós seguidores, o que pode comprometer o processo de consenso. Dado que as mensagens periódicas do líder foram configuradas para ocorrer a cada 100 ms, os resultados de latência obtidos apresentaram valores médios e máximos significativamente abaixo desse intervalo, evidenciando uma boa estabilidade da rede durante o seu funcionamento. Na Figura 4.8 é possível evidenciar o que foi descrito previamente.

A latência das mensagens OCPP, apresentada na Figura 4.9a, aumenta progressivamente do primeiro à última experiência, refletindo a transição de um cenário idealizado para um mais próximo da realidade. Apesar disso, mesmo no pior caso, a latência máxima permanece abaixo do limite de tolerância das mensagens ISO 15118, como apresentado no estado da arte 2, indicando que o servidor OCPP consegue responder em tempo útil.

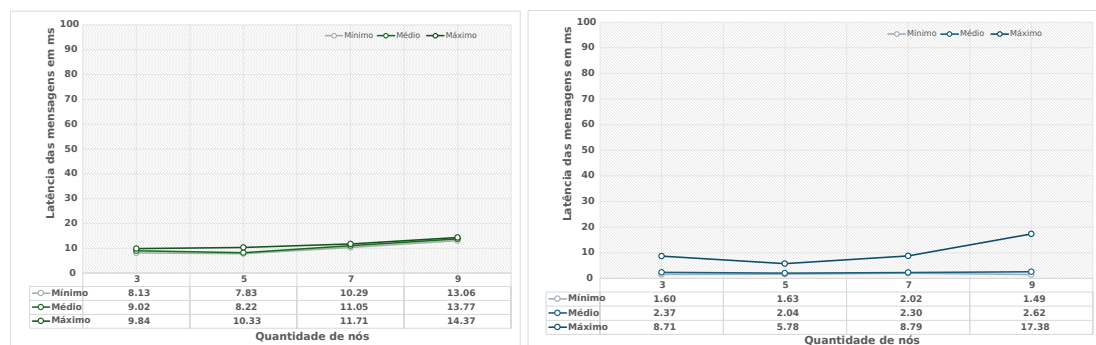
Por fim, a Figura 4.9b revela uma queda significativa no débito da comunicação entre as experiências inter-processos (simulado) e inter-dispositivos (real). Essa redução pode ser atribuída às diferenças entre os ambientes simulados e reais, como a capacidade de processamento dos dispositivos e a latência da rede. Apesar da queda, o débito ainda é suficiente para atender aos requisitos da aplicação, como demonstrado pela latência das mensagens OCPP e taxa de sucesso das mensagens OCPP mostrado na Tabela 4.8.



(a) Consumo computacional (inter-dispositivos)

(b) Consumo computacional (inter-processos)

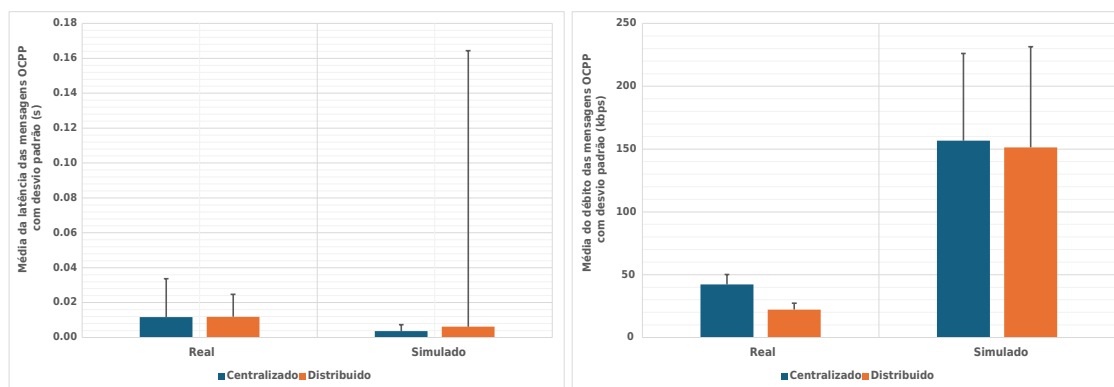
Figura 4.7: Comparação da comunicação OCPP utilizados durante experiências com topologia Distribuída (com Raft) e Centralizada (sem Raft) em ensaio inter-processos (Simulado) vs inter-dispositivos (Real)



(a) Latência mensagens teste inter-dispositivos

(b) Latência mensagens teste inter-processos rede distribuída (ms)

Figura 4.8: Comparação da comunicação da rede Raft utilizados durante experiências nos carregadores(inter-dipositivos) e na unidade computacional (inter-processos)



(a) Latência das mensagens OCPP em segundos

(b) Débito das mensagens OCPP em kbps

Figura 4.9: Comparação da comunicação OCPP utilizados durante experiências com topologia Distribuída (com Raft) e Centralizada (sem Raft) em ensaio inter-processos (Simulado) vs inter-dispositivos (Real)

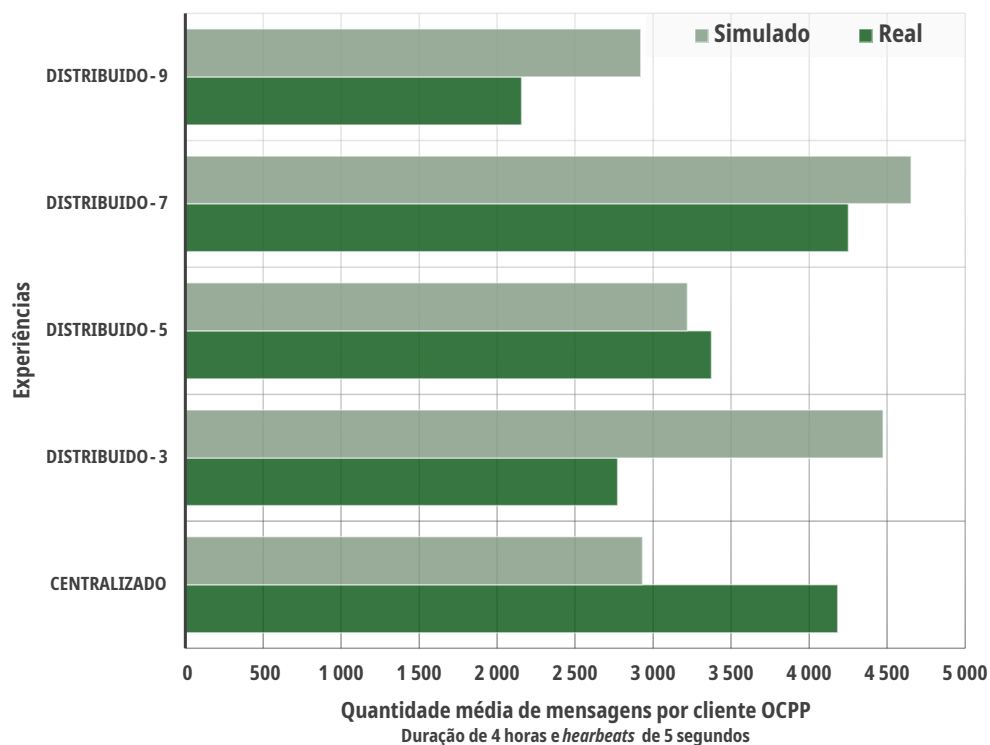


Figura 4.10: Quantidade de mensagens registada nos clientes OCPP no final de cada experiência

A Figura 4.10 ilustra a quantidade média de mensagens OCPP trocadas em cada cliente OCPP no final cada experiência realizada, tanto em ambiente simulado como real. A comparação entre a configuração puramente centralizada e a configuração distribuída baseada na rede Raft, com diferentes números de nós, revela uma relação entre o número de mensagens trocadas e a disponibilidade do servidor OCPP.

Nas experiências com menor volume de mensagens OCPP, a menor disponibilidade do servidor OCPP pode ser atribuída ao tempo necessário para a eleição do líder na rede Raft. Este fenómeno é particularmente evidente na configuração com 9 nós, onde o prolongado tempo de eleição resultou em um número reduzido de mensagens trocadas.

Em contraste, a experiência com a rede Raft e 7 nós apresenta um número de mensagens superior ao do sistema centralizado. Este resultado pode ser atribuído à maior eficiência na gestão computacional estocástica das tarefas de processamento, resultando numa maior precisão na transmissão de mensagens.

Nas Tabelas 4.6 e 4.7 é apresentado a quantidade de mensagens OCPP trocadas totais durante o primeiro e última experiência. O primeiro ensaio realizado dentro do mesmo dispositivo entre processos e sem uma rede Raft, já a Tabela 4.7 representa a experiência realizada nos carregadores da INESC TEC e com uma rede Raft entre os servidores OCPP de 9 nós.

Dos valores observados nas tabelas que representam a quantidade das sessões de carregamento, pode-se observar que as experiências que utilizam a rede Raft apresentam metade das sessões de carregamento. Ao analisar os dados das sessões de forma mais aprofundada, pode-se observar

Tabela 4.6: Quantidade de mensagens dos clientes OCPP no fim da experiência centralizado

Mensagens OCPP	Quantidade
Authorize	121
BootNotification	10
Heartbeat	28711
NotifyEVChargingNeeds	131
StatusNotification	121
TransactionEvent	242
Total	29336

Tabela 4.7: Quantidade de mensagens dos clientes OCPP no fim do ensaio distribuído (9 nós)

Mensagens OCPP	Quantidade
Authorize	61
BootNotification	13
Heartbeat	14790
NotifyEVChargingNeeds	66
StatusNotification	61
TransactionEvent	121
Total	15112

que a rede Raft inicialmente mostrou dificuldade em atingir o consenso na eleição do líder, limitando assim a quantidade de sessões realizadas durante o ensaio. Uma explicação para esse comportamento pode ser dada baseada na componente estocástica do mecanismo Raft, que a cada mensagem recebida do líder gera uma contagem regressiva, numa certa janela temporal configurada inicialmente. Este tempo aleatório e regressivo serve para que, caso o líder falhe em enviar uma mensagem de resposta, os nós seguidores acordem em tempos diferentes para iniciar uma nova eleição, limitando assim conflitos em que dois nós se tornem candidatos simultaneamente. Essa janela temporal deveria ser alargada à medida que a quantidade de nós utilizados na rede aumenta. Por isso, é importante realizar um ajuste dos parâmetros da rede Raft adaptado aos atributos da rede em específico. Os parâmetros utilizados nas experiências foram os mesmos para permitir uma comparação mais coerente entre as experiências com diferentes quantidades de nós.

A taxa de sucesso das sessões de carregamento é o cálculo de quantas sessões iniciadas foram finalizadas sem falhas durante o processo, para isso basta observar a quantidade de mensagens Status Notification relativamente à metade da quantidade de mensagens *Transaction Event*. Pois quando um novo EV é detetado a mensagem Status Notification é enviada, durante uma sessão de carregamento é enviada uma mensagem de Transaction Event no início e uma no final para a entidade gestora, sendo assim, numa sessão de carregamento bem-sucedida teremos duas mensagens de *Transaction Event* e uma de *Status Notification*, sendo assim para verificar a taxa de sucesso basta utilizar a Fórmula 4.1

$$\text{Taxa de sucesso} = \frac{\# \text{ Mensagens } Transaction \text{ Event}}{2 * \# \text{ Mensagens } Status \text{ Notification}} \quad (4.1)$$

A Tabela 4.8 evidencia que nos ensaios distribuídos a taxa de sucesso, embora inferior à dos ensaios centralizados, mantém-se acima de 95% mesmo no cenário menos favorável, que corresponde à rede Raft com 9 nós.

Tabela 4.8: Taxa de sucesso das sessões de carregamento (%) em cada experiência

Experiência	Taxa de sucesso
Primeiro ensaio	100%
Segundo ensaio (9 nós)	95%
Terceiro ensaio	100%
Quarto ensaio (9 nós)	99%

4.3.2 Resultados da rede Raft

Na Tabela 4.9 é apresentado a latência, isto é, o intervalo de tempo entre o envio até o recebimento da resposta das mensagens na rede Raft.

Tabela 4.9: Latência das mensagens Raft (s) em cada configuração de rede entre **processos**

Número de nós Raft	Mínimo	Médio	Máximo
3 nós	1.60 ms	2.04 ms	4.63 ms
5 nós	1.63 ms	2.04 ms	5.78 ms
7 nós	2.02 ms	2.48 ms	4.54 ms
9 nós	1.49 ms	2.62 ms	17.38 ms

Observa-se que os valores mínimos de latência mantêm-se relativamente estáveis e não aumentam de forma linear com o acréscimo do número de nós. Os valores médios, por sua vez, apresentam um leve aumento à medida que o número de nós cresce. Já os valores máximos, até 7 nós, são similares, mas no ensaio com 9 nós, há um aumento considerável, de cerca de 350%.

Apesar desse aumento expressivo, a ordem de grandeza dos valores máximos de latência permanece abaixo dos 100 ms do intervalo entre os *heartbeats* enviados pelo líder, garantindo a consistência da rede Raft e, conseqüentemente, a disponibilidade do servidor OCPP ativo.

Conforme a Tabela 4.10, numa rede com N nós, onde um é líder, espera-se o envio de $(N - 1) * (1 / \text{heartbeats_interval})$ mensagens. Com *heartbeats* a cada 100 ms, o líder enviaria 10 mensagens por segundo aos demais nós $(N - 1)$. Portanto, numa rede de 9 nós, o débito esperado seria de 80 TPS (Transações por segundo). Os dados da tabela são baseados no débito calculado no nó líder, mas os valores mínimos registrados dizem respeito aos valores obtidos nos nós seguidores que se recebem mensagens a cada 100 ms devem ter um valor de débito de 10 transações por segundo teoricamente, e pode-se confirmar que os valores do débito em tais nós possuem valores muito próximos ao esperado.

Tabela 4.10: Débito das mensagens Raft (s) em cada configuração de rede entre processos (valores aproximados)

Número de nós Raft	Mínimo	Médio	Máximo
3 nós	10 TPS	28 TPS	30 TPS
5 nós	10 TPS	39 TPS	40 TPS
7 nós	10 TPS	58 TPS	60 TPS
9 nós	9 TPS	78 TPS	79 TPS

Na Figura 4.11, observa-se que a quantidade de mudanças de liderança nos ensaios Raft nos carregadores não apresenta dependência do número de nós, sugerindo a presença de um líder forte após a eleição ser concluída. Contudo, a análise dos resultados revela um aumento significativo no tempo de eleição do líder à medida que o número de nós aumenta. Este comportamento está relacionado na maior dificuldade de alcançar consenso à medida que o número de nós aumenta, mas pode ser minimizado com a alteração das configurações específicas da rede Raft.

Na Tabela 4.11, são apresentados os dados obtidos do ensaio com a rede Raft executada nos carregadores. Observa-se um aumento nos valores de latência à medida que a rede aumenta o número de nós, sendo essa alteração sutil entre as redes com 3 a 7 nós e apresentando um salto significativo de aproximadamente 143% quando a rede é composta por 9 nós.

Apesar desse salto, é importante destacar que os valores de latência permanecem numa ordem de grandeza muito inferior aos 2 segundos de tolerância definidos pela ISO 15118 para as mensagens de carregamento inteligente entre o veículo e o carregador como apresentado no Capítulo 2.

No que diz respeito ao débito de transmissão da rede Raft, na secção anterior foram apresentados os valores esperados para cada configuração de rede. Na Tabela 4.12 é possível verificar que, na rede Raft implementada entre os carregadores, os valores obtidos se aproximam bastante dos valores esperados e dos resultados obtidos no ensaio inter-processos. Esta proximidade comprova a consistência da rede Raft, mesmo em diferentes ambientes, incluindo ambientes reais.

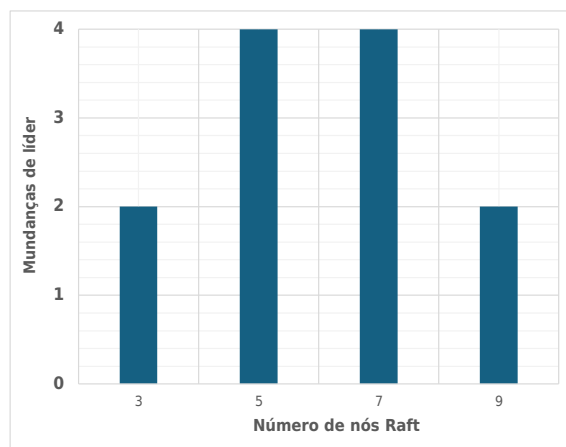


Figura 4.11: Quantidade de mudanças de líder no ensaio real com rede Raft

Tabela 4.11: Latência das mensagens Raft (s) em cada configuração de rede nos carregadores

Número de nós Raft	Mínimo	Médio	Máximo
3 nós	8.13 ms	9.02 ms	9.84 ms
5 nós	7.83 ms	8.22 ms	10.33 ms
7 nós	10.29 ms	11.05 ms	11.71 ms
9 nós	13.06 ms	13.77 ms	14.37 ms

Tabela 4.12: Débito das mensagens Raft (s) em cada configuração nos carregadores

Número de nós Raft	Mínimo	Médio	Máximo
3 nós	10 TPS	19 TPS	20 TPS
5 nós	9 TPS	39 TPS	40 TPS
7 nós	9 TPS	58 TPS	60 TPS
9 nós	9 TPS	79 TPS	80 TPS

4.4 Comparação dos resultados

Não foram encontrados, até a publicação deste trabalho, abordagens semelhantes do mecanismo de consenso na infraestrutura de carregamento. Por isso, foram elaborados ensaios que pudessem ser comparados entre si. Entretanto, o trabalho de [22], que utiliza mecanismo de consenso em mobilidade elétrica, foi usado como referência para comparação.

Comparando-se com o estudo de Luo et al. (2023), o desempenho de débito e latência da rede Raft neste trabalho apresentou valores distintos. Enquanto Luo et al. (2023) obtiveram cerca de 2200 transações por segundo (TPS) de débito e 350 ms de latência com 10 nós, os ensaios com 9 nós neste trabalho resultaram em 79 TPS e 17.38 ms de latência entre processos, e 80 TPS e 14.37 ms de latência nos carregadores reais.

Os valores de débito obtidos neste trabalho foram limitados pelo intervalo de 100 ms dos heartbeats, mas mesmo que fossem maiores, não teriam aplicação prática para a finalidade do algoritmo Raft aqui implementado.

Na Figura 4.12, é apresentada uma comparação entre o valor teórico do débito e os valores de débito obtidos nas experiências inter-dispositivos, ou seja, durante a utilização dos carregadores. Os dados utilizados são extraídos pelos líderes no momento em que recebem as respostas às mensagens enviadas. A partir desses dados, calcula-se a quantidade de mensagens recebidas num determinado intervalo de tempo, que é então normalizada em termos de mensagens por segundo ou transações por segundo (TPS).

Nos testes realizados, o intervalo entre as mensagens enviadas pelo líder é definido em 100 ms, o que implica que, para cada nó na rede, o líder deve enviar 10 mensagens por segundo. Considerando que o número de mensagens que o líder deve enviar a cada transação é de $N-1$, onde N representa o total de nós na rede, temos o seguinte: numa rede com 3 nós, espera-se um débito teórico de 20 mensagens por segundo; numa rede com 5 nós, o valor esperado é de 40 TPS, 7 nós, 60 TPS e finalmente com 9 nós é esperado 80 TPS.

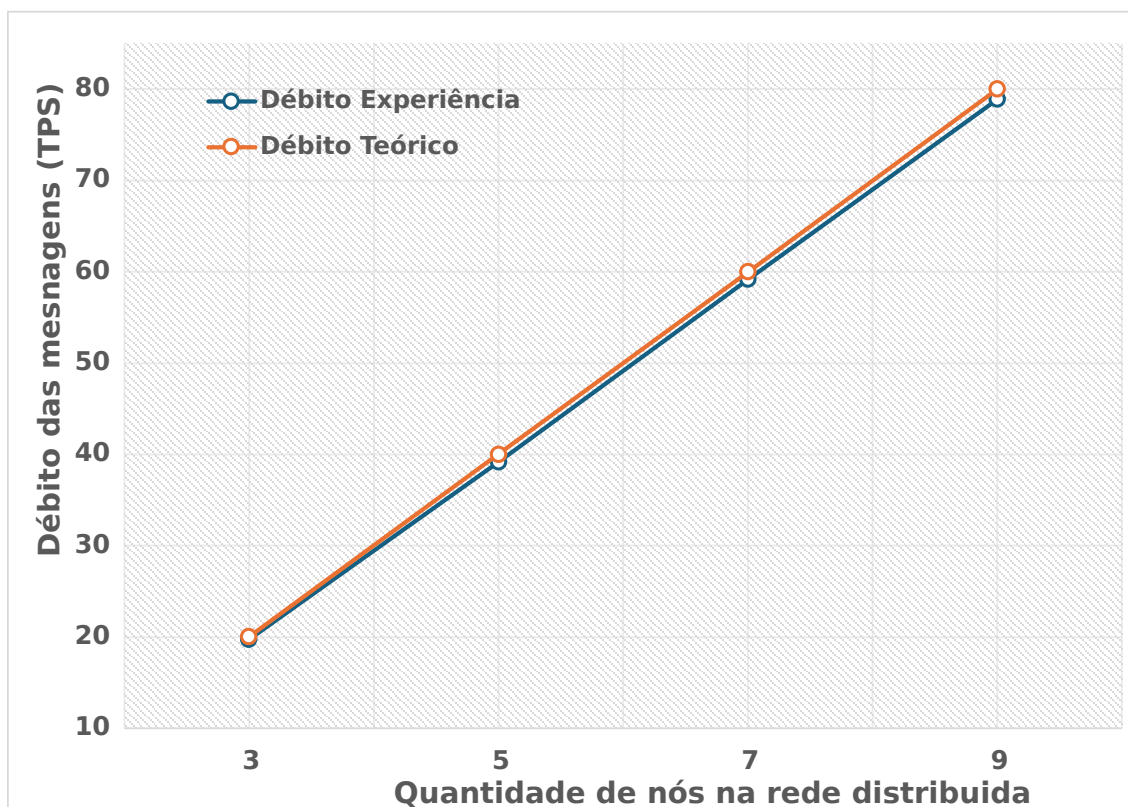


Figura 4.12: Comparação dos valores calculados e do ensaio do débito das mensagens Raft (TPS)

Considerando os tempos de resposta esperados pela ISO 15118 e OCPP, bem como o tempo de espera dos utilizadores na autenticação, os valores obtidos são mais do que suficientes. Além disso, a análise da latência revela que os valores medidos foram significativamente inferiores aos observados por [22].

A baixa latência é crucial, pois garante que o líder possa enviar novas mensagens sem atrasos que comprometam o processo de consenso. Como mencionado, se a latência for elevada, o líder pode erroneamente considerar algumas mensagens como não respondidas, interferindo na comunicação entre os nós. No nosso sistema, com mensagens periódicas configuradas para ocorrer a cada 100 ms, os resultados de latência mostraram valores médios e máximos bem abaixo desse intervalo, o que demonstra uma robusta estabilidade da rede durante o seu funcionamento. Essa eficiência não só melhora a experiência do utilizador, reduzindo o tempo de espera na autenticação, mas também assegura que o processo de consenso seja mantido de forma eficaz, promovendo a fiabilidade do sistema na totalidade.

Ao comparar os ensaios com processos simulados e com carregadores reais, observou-se que os valores máximos de latência da rede Raft com 9 nós, foram superiores nos carregadores reais, mesmo que estes possuam menor capacidade computacional. Essa diferença pode ser atribuída à sobrecarga de processamento no ensaio simulado, onde um único dispositivo precisou simular múltiplos processos simultaneamente: a comunicação ISO 15118 entre 10 EVCCs e SECCs, a rede Raft com 9 nós e a comunicação OCPP com o líder ativo. Esta simulação intensiva pode ter

impactado o desempenho da rede Raft, resultando em latências superiores às observadas nos carregadores reais, que operavam com um único SECC e EVCC. Esta hipótese baseia-se na discussão apresentada neste capítulo, que aborda a utilização da memória RAM pelo módulo EVCC. Nos testes simulados, múltiplos módulos EVCC eram executados em um único dispositivo computacional, afetando potencialmente o desempenho da rede Raft.

4.5 Sumário

Neste capítulo, os ensaios realizados evidenciam que a rede Raft, quando em operação adequada, exerce um impacto mínimo no desempenho da comunicação e execução do caso de uso OCPP. No entanto, eventuais falhas ou atrasos na eleição do líder podem comprometer a disponibilidade do servidor OCPP e, conseqüentemente, a gestão das sessões de carregamento. Nesses cenários, a implementação de um servidor redundante pode assegurar a continuidade dos serviços e a disponibilidade dos dados necessários para as sessões de carregamento.

Em aplicações futuras, a rede Raft poderá ser utilizada para a replicação de dados e do estado do sistema de gestão, tornando ainda mais crucial a sua alta disponibilidade. Para tal, será necessário ajustar os parâmetros da rede Raft em função do número de nós e considerar a adoção de uma implementação mais robusta, bem como uma linguagem de programação mais eficiente. Adicionalmente, a exploração de protocolos de comunicação alternativos, com latência e débito otimizados, poderá ser benéfica.

Apesar das limitações da implementação utilizada, observou-se que a ordem temporal de operação da rede Raft é inferior à exigida pelos protocolos ISO 15118 e OCPP, indicando que a troca de mensagens e a execução de operações ocorrem mais rapidamente do que o necessário. Esta vantagem garante a coexistência harmoniosa desses sistemas no mesmo ambiente de comunicação e processamento.

Em suma, os resultados demonstram a viabilidade de uma aplicação desta natureza e o seu potencial para garantir sessões de carregamento otimizadas, fornecendo serviços ininterruptos para o utilizador e a rede, independentemente da disponibilidade das unidades de carregamento, desde que devidamente configurada.

Capítulo 5

Conclusões e Trabalho Futuro

Os resultados desta dissertação demonstram a viabilidade e o potencial da arquitetura proposta para a gestão descentralizada de sessões de carregamento de VEs, utilizando o algoritmo Raft. A arquitetura mostrou-se eficiente e robusta, com um impacto mínimo no desempenho da comunicação e execução do caso de uso OCPP, mesmo em cenários de falhas. Adicionalmente, a arquitetura proposta oferece maior flexibilidade em comparação com as arquiteturas centralizadas tradicionais, que podem apresentar dificuldades em lidar com o crescente número de VEs e estações de carregamento.

No trabalho realizado nesta dissertação, a maioria dos objetivos foram alcançados:

1. **Estudo de mecanismos de consenso e estratégias utilizadas para selecionar o mecanismo de consenso a utilizar:** O capítulo 2 da dissertação apresenta um estudo abrangente de diferentes mecanismos de consenso, incluindo Paxos, Raft, PBFT e outros. A escolha do Raft foi justificada por sua simplicidade, facilidade de implementação e comprovada eficácia em sistemas distribuídos.
2. **Pesquisa e análise de bibliotecas que implementem corretamente o algoritmo de consenso selecionado:** A dissertação descreve a pesquisa e seleção de bibliotecas Python que implementam o algoritmo Raft, culminando na escolha da biblioteca "raftos" após uma análise comparativa detalhada (Tabela 3.1 no Capítulo 3).
3. **Implementação em unidades de carregamento reais:** O capítulo 3 detalha a implementação do sistema proposto, incluindo os módulos de servidor e cliente OCPP, o emulador EVCC e a integração com a biblioteca Raft, possibilitando a execução de testes em cada unidade de carregamento.
4. **Especificação de testes a serem executados e respectivas métricas de avaliação:** O capítulo 4 apresenta a arquitetura de testes, incluindo os cenários de carregamento, os parâmetros do protocolo Raft e a descrição dos quatro ensaios realizados (dois inter-processos e dois inter-dispositivos), com a coleta de dados sobre consumo de CPU, memória RAM, latência e débito das mensagens OCPP e Raft.

5. **Análise de resultados e análise da viabilidade de implementação do mecanismo de consenso em unidades de carregamento reais:** os resultados das experiências, apresentados no capítulo 4, demonstram a viabilidade da implementação do mecanismo de consenso Raft em unidades de carregamento, com baixo impacto no desempenho e alta taxa de sucesso nas sessões de carregamento, mesmo em cenários com falhas simuladas.

Em suma, esta dissertação apresenta uma contribuição significativa para o domínio da mobilidade elétrica, propondo uma solução que visa contribuir para o problema da gestão de sessões de carregamento. As questões em aberto identificadas neste trabalho abrem caminho para futuras pesquisas e desenvolvimentos, que poderão aprimorar ainda mais a eficiência, confiabilidade e sustentabilidade da infraestrutura de carregamento de veículos elétricos.

As questões em aberto mencionadas podem ser abordadas em trabalhos futuros, como, por exemplo, o desenvolvimento de um sistema de gestão de falhas capaz de lidar com falhas mais complexas, como a perda de dados ou a corrupção de mensagens. Outra questão é a necessidade de realizar testes em larga escala, com um número maior de carregadores e veículos elétricos, para avaliar a escalabilidade do sistema.

Além disso, seria interessante explorar a utilização de outras tecnologias de comunicação, como redes sem fio ou protocolos de comunicação mais eficientes para rede Raft, para aumentar a flexibilidade e a escalabilidade do sistema. A integração do sistema com outras fontes de energia renovável, como painéis solares ou turbinas eólicas, também poderia ser investigada para aumentar a sustentabilidade da infraestrutura de carregamento.

Outra possibilidade para trabalhos futuros é a implementação de um algoritmo de otimização da sessão de carregamento no *backend* do servidor OCPP, permitindo a seleção de diferentes estratégias de otimização, considerando os dados de todos os VEs conectados na infraestrutura. Adicionalmente, a transição entre servidores OCPP no mesmo cluster poderia ser aprimorada, utilizando a característica de replicação de logs para replicar dados sensíveis e importantes sobre as transações ativas, garantindo a continuidade do processo mesmo após a eleição de um novo líder.

Pode-se adaptar o algoritmo de sistemas distribuídos para no caso de falha para definição do *schedule* de forma distribuída ou que o *schedule* não seja totalmente otimizado o líder daquele *cluster* se tornar o agregador indireto, assim o sistema será ainda mais flexível e atingirá com mais facilidade o valor ótimo do perfil de carregamento para aquele *cluster*. Podendo trabalhar em conjunto ou ser implementado numa arquitetura centralizada sem muitos esforços de adaptação.

Transição entre servidores OCPP no mesmo *cluster*: É necessário desenvolver uma estratégia para transição entre servidores OCPP dentro do mesmo *cluster*. Pode-se utilizar a característica de replicação de *logs* para replicar dados sensíveis e importantes sobre as transações ativas, permitindo que mesmo após a eleição de um novo líder o processo continue sem necessidade de intervenção do utilizador e com o menor esforço possível.

Implementação de um algoritmo de otimização da sessão de carregamento: A estrutura para a implementação de tal algoritmo está pronta, mas é necessário integrar essa funcionalidade no

backend do servidor OCPP. Isto permitirá a seleção de diferentes estratégias de otimização, considerando os dados de todos os VEs conectados na infraestrutura.

Referências

- [1] Iso 15118-2: Road vehicles – vehicle to grid communication interface – part 2: Network and application protocol requirements. <https://www.iso.org/standard/55366.html>, 2014. ISO 15118-2:2014.
- [2] Open charge point protocol (ocpp) v201. As-fact standard, Open Charge Alliance (OCA), 03 2020. <https://openchargealliance.org/protocols/open-charge-point-protocol/#OCPP2.0.1>.
- [3] Simon Accascina. Zatt. <https://github.com/simonacca/zatt.git>, 2014.
- [4] International Energy Agency. World energy outlook 2022, 2022.
- [5] International Energy Agency. Public charging point per battery-electric ldv ratio in selected countries against battery electric ldv stock share, 2015-2022, 2023.
- [6] International Energy Agency. Global ev outlook 2024, 2024.
- [7] Ahmad Alzahrani, Shriya Makarand Wangikar, Vairavasundaram Indragandhi, Rassaiah Raja Singh, e Vairavasundaram Subramaniaswamy. Design and implementation of sae j1939 and modbus communication protocols for electric vehicle. *Machines*, 11, 2 2023.
- [8] Toby Burress. pyraft. <https://github.com/kurin/py-raft.git>, 2014.
- [9] Dario Medina da Cruz. Avaliação de algoritmos de carregamento de veículos elétricos em parques que integrem sistemas fotovoltaicos em autoconsumo, 2018.
- [10] H.S. Das, M.M. Rahman, S. Li, e C.W. Tan. Electric vehicles standards, charging infrastructure, and impact on grid integration: A technological review. *Renewable and Sustainable Energy Reviews*, 120:109618, 2020.
- [11] Xiaohong Deng, Kangting Li, Zhiqiang Wang, Juan Li, e Zhiqiong Luo. A survey of blockchain consensus algorithms. páginas 188–192. Institute of Electrical and Electronics Engineers Inc., 2022.
- [12] Azzara G e Mauri A. ISO15118-20. Design and implementation of bidirectional power transfer for V2X applications. *Master Thesis*, 1(1):891–921, 2022.
- [13] Jose Miguel Ribeiro GONCALVES. Solucoes de tecnologias de informação e comunicação para implementação do conceito v2g, 2015.
- [14] Michael Hardinghaus, Markus Loecher, e John Anderson. Real-world insights on public charging demand and infrastructure use from electric vehicles. *Environmental Research Letters*, 15, 10 2020.

- [15] Dongyan Huang, Xiaoli Ma, e Shengli Zhang. Performance analysis of the raft consensus algorithm for private blockchains. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50:172–181, 1 2020.
- [16] Hyperledger. Hperledger blockchain performance metrics white paper. Relatório técnico, Hyperledger Foundation, May 2023.
- [17] INESC TEC. Laboratório de redes elétricas inteligentes e veículos elétricos (sgev). <https://www.inesctec.pt/pt/centros/cite>. Acedido em 29 de julho de 2024.
- [18] Doris Johnsen, Lars Ostendorf, Mischa Bechberger, e Daniel Strommenger. Review on smart charging of electric vehicles via market-based incentives, grid-friendly and grid-compatible measures. *World Electric Vehicle Journal*, 14, 1 2023.
- [19] Willett Kempton e Jasna Tomić. Vehicle-to-grid power fundamentals: Calculating capacity and net revenue. *Journal of Power Sources*, 144(1):268–279, 2005.
- [20] lisacl. aioraft. <https://github.com/lisacl/aioraft.git>, 2014.
- [21] Diana Lopez-Behar, Martino Tran, Jerome R. Mayaud, Thomas Froese, Omar E. Herrera, e Walter Merida. Putting electric vehicles on the map: A policy agenda for residential charging infrastructure in canada. *Energy Research & Social Science*, 50:29–37, 2019.
- [22] Haoxiang Luo, Hongfang Yu, e Jian Luo. Praft and rpbft: A class of blockchain consensus algorithm and their applications in electric vehicles charging scenarios for v2g networks. *Internet of Things and Cyber-Physical Systems*, 3:61–70, 1 2023.
- [23] Dean McCarthy e Peter Wolfs. The hv system impacts of large scale electric vehicle deployments in a metropolitan area. 01 2011.
- [24] Rounak Meyur, Swapna Thorve, Madhav Marathe, Anil Vullikanti, Samarth Swarup, e Henning Mortveit. A reliability-aware distributed framework to schedule residential charging of electric vehicles. 2 2022.
- [25] Peng Ding Nicola Manzini. Vesper. <https://github.com/Oaklight/Vesper-raft.git>, 2019.
- [26] Nanduni I. Nimalsiri, Chathurika P. Mediwaththe, Elizabeth L. Ratnam, Marnie Shaw, David B. Smith, e Saman K. Halgamuge. A survey of algorithms for distributed charging control of electric vehicles in smart grid. *IEEE Transactions on Intelligent Transportation Systems*, 21:4497–4515, 11 2020.
- [27] Sergey Novikov, Olga Mikheenko, Natalia Kulagina, e Oleg Kazakov. Digital registry of professional competences of the population drawing on distributed registries and smart contracts technologies. *Business Informatics*, 2018:43–53, 12 2018.
- [28] Diego Ongaro e John Ousterhout. In search of an understandable consensus algorithm (extended version). Relatório técnico, Stanford University, 2014.
- [29] Filipp Ozinov. Pysyncobj. <https://github.com/bakwc/PySyncObj.git>, 2024.
- [30] Lopes A Pedro J. Exploração de algoritmos de consenso no quorum, 2023.

- [31] Maria Raboaca, Maria Meheden, Andrei Musat, Andrei Viziteu, Alexandru Creanga, Valentin Vlad, Constantin Filote, Mihai Rata, e Alexandru Lavric. An overview and performance evaluation of open charge point protocol from an electromobility concept perspective. *International Journal of Energy Research*, 46, 09 2021.
- [32] Xiumin Ran, Chaoshan Xin, Wengang Li, Cheng Chen, e Zhongping Yu. New energy charging pile computing system based on blockchain technology and raft algorithm. Em *2022 World Automation Congress (WAC)*, páginas 384–388, 2022.
- [33] Sean Reed. simpleraft. <https://github.com/streed/simpleRaft.git>, 2015.
- [34] Á. Rodríguez-Serrano, A. Torralba, E. Rodríguez-Valencia, e J. Tarifa-Galisteo. A communication system from ev to ev service provider based on ocpp over a wireless network. Em *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, páginas 5434–5438, 2013.
- [35] Jóni Nichal Buchinho dos Santos. Sistema de comunicação entre veículos elétricos e posto de carregamento segundo a norma iso 15118, 2020.
- [36] Fernando Augusto Medeiros Silva e Ricardo Augusto Rabelo de Oliveira. Análise de algoritmos distribuídos híbridos para consenso em rede.
- [37] Cecília Conceição de Oliveira Soares. Formalização da reconfiguração de protocolosde consenso usando alloy, 2022.
- [38] Dário Macedo Sobrinho, Janaína Barbosa Almada, Fernando Lessa Tofoli, Ruth Pastora Saraiva Leão, e Raimundo Furtado Sampaio. Distributed control based on the consensus algorithm for the efficient charging of electric vehicles. *Electric Power Systems Research*, 218:109231, 2023.
- [39] Qiang Tang, Kezhi Wang, Kun Yang, e Yuan-sheng Luo. Congestion-balanced and welfare-maximized charging strategies for electric vehicles. *IEEE Transactions on Parallel and Distributed Systems*, 31(12):2882–2895, 2020.
- [40] Peter Van Den Bossche. *IEC 61851-1: Electric vehicle conductive charging system - Part 1: General requirements*, páginas 1–99. IEC, Nov 2010.
- [41] Thota Venkata Pruthvi, Niladri Dutta, Phaneendra Bobba, e B Vasudeva. Implementation of ocpp protocol for electric vehicle applications. *E3S Web of Conferences*, 87:01008, 01 2019.
- [42] Yfan Xing. raft. <https://github.com/xingyif/raft.git>, 2018.
- [43] yassine Lamgarchal. Chillaxd. <https://github.com/ylamgarchal/chillaxd.git>, 2015.
- [44] Jia Ying Yong, Vigna K. Ramachandaramurthy, Kang Miao Tan, e N. Mithulananthan. A review on the state-of-the-art technologies of electric vehicle, its impacts and prospects. *Renewable and Sustainable Energy Reviews*, 49:365–385, 2015.
- [45] Alexander Zhebrak. raftos. <https://github.com/zhebrak/raftos.git>, 2023.
- [46] Yanchong Zheng, Songyan Niu, Yitong Shang, Ziyun Shao, e Linni Jian. Integrating plug-in electric vehicles into power grids: A comprehensive review on power interaction mode, scheduling methodology and mathematical foundation. *Renewable and Sustainable Energy Reviews*, 112:424–439, 2019.

Anexo A

Implementação do Sistema Proposto

O sistema proposto foi implementado em python e o código fonte está disponível nos seguintes endereços:

- Módulo Raft e servidor OCPP
- Módulo SECC, EVCC e cliente OCPP