



FERRAMENTA DE TESTE VIRTUAL PARA SISTEMAS DE PROTEÇÃO & CONTROLE CENTRALIZADOS - IEC 61850

Alailton José Alves Júnior¹, Alan Petrônio Pinheiro¹, Fabrício Augusto Matheus Moura¹, Leandro José Duarte¹, Gabriel Machado Bernardes¹, Daniel Braga Melo¹, Fabio Romero de Souza Junior¹ e Andreia Coelho Domingos¹

¹FEELT - Universidade Federal de Uberlândia

Resumo - Este trabalho descreve o desenvolvimento de um testador virtual para Dispositivos Eletrônicos Inteligentes (IEDs) de proteção, conforme a norma IEC 61850. O sistema é projetado para uma arquitetura centralizada do sistema proteção, automação e controle (PAC). Nesse modelo, os IED não possuem hardware dedicado e são hospedados em servidores dentro da subestação. O software desenvolvido é implementado em conjunto com esses IEDs virtuais, dentro de uma máquina virtual ou em um container. Os resultados alcançados foram positivos, demonstrando a viabilidade do uso do testador virtual para realizar testes de proteção em IEDs, tanto os convencionais quanto os virtuais.

Palavras-Chave - IEC 61850, Redes Inteligentes, Subestações Digitais, Testador Virtual, Virtualização PAC

VIRTUAL TEST SET FOR CENTRALIZED PROTECTION & CONTROL SYSTEMS - IEC 61850

Abstract - This work presents the results obtained in the development of a virtual test set for Intelligent Electronic Devices (IEDs) based on the IEC 61850 standard. In this context, a centralized architecture of the protection, automation, and control system was considered. Within this framework, the IEDs are virtualized machines within a server at the substation. From this scope, the developed software is implemented alongside virtual IEDs, in a virtual machine or container. The obtained results proved to be satisfactory and enabled the application of the virtual test set for protection tests on both conventional and virtual IEDs.

Keywords - Digital Substations, IEC 61850, smart grid, Virtualized PAC, Virtual Test Set

I. INTRODUÇÃO

No contexto dos sistemas de energia modernos, a evolução da tecnologia trouxe mudanças profundas na forma como concebemos, projetamos e gerenciamos componentes críticos, como subestações. Essas instalações servem como pontos de acesso para uma variedade extensa de elementos do sistema

de energia, abrangendo linhas de transmissão, transformadores, geradores, fontes renováveis e as mais diversas variedades de cargas [1].

A crescente integração de fontes renováveis, como a energia solar e eólica, e a constante expansão com novas linhas de transmissão e crescentes demandas energéticas têm tornado as subestações tradicionais inadequadas para enfrentar as complexidades da rede elétrica atual. Anteriormente, as subestações eram projetadas principalmente para direcionar o fluxo unidirecional de energia, dos geradores para as cargas. No entanto, com a proliferação de sistemas de geração distribuída e micro redes, a energia flui em ambas as direções, dificultando o seu gerenciamento e proteção. Por esse motivo, a busca por maior confiabilidade, flexibilidade e interoperabilidade tem impulsionado a utilização de tecnologias mais modernas e a implementação da concepção de redes inteligentes dentro das subestações [2, 3].

Abordando essas demandas, o conceito de *smart grid* tem ganhado força, impulsionando a necessidade de digitalização das subestações e a incorporação de padrões abertos, como o IEC 61850 [4]. Nesse novo paradigma, a comunicação digital da norma permitiu que fluxos contínuos de valores medidos amostrados e sinais de controle trafeguem pela rede da subestação, utilizando protocolos padronizados e específicos para esse tipo de aplicação. Nesse cenário, é retratado que os sistemas modernos de automação e proteção de subestações, baseados na norma IEC 61850, são cruciais na implementação de uma infraestrutura inteligente nos sistemas de energia [5].

Diante disto, no desenvolvimento e implementação de subestações puramente digitais, surgiu uma nova arquitetura centralizada do sistema de Proteção, Controle e Automação (PAC). Neste modelo, os dispositivos eletrônicos inteligentes (IED), que antes eram equipamentos físicos e isolados, passariam a ser dispositivos virtualizados, desacoplando totalmente da necessidade de um hardware dedicado [6].

No trabalho desenvolvido em [7], foi apresentado uma avaliação do desempenho de uma aplicação end-to-end, do sistema elétrico, virtualizada e distribuída, realizando teste de aplicabilidade e *benchmark* para implementação do software *OpenDSS*. Ao final, os autores concluíram que as aplicações que demandam um tempo de resposta menor que 1ms poderiam ser implementadas em um ambiente virtualizado.

No *whitepaper* [8] foi demonstrado uma arquitetura do sis-

tema de proteção totalmente virtualizada, no qual foram realizados testes em um relé virtual que estava implementado dentro de uma máquina virtual em um servidor. Os resultados obtidos pelos autores apresentaram baixos valores de atraso no tempo de *trip* do relé, viabilizando a sua aplicação.

Com base no contexto supracitado, o trabalho explanado neste documento teve como finalidade criar uma aplicação viável de um testador de IEDs virtuais hospedados dentro de um servidor em uma subestação digital IEC 61850. Para isso, foi desenvolvido um software para ser implementado junto aos IED em uma máquina virtual ou container no servidor, de modo que seja possível alcançar, em uma aplicação puramente de software, funcionalidades de tempo crítico com respostas semelhantes aos equipamentos que possuem hardware dedicado.

A organização subsequente deste artigo é delineada da seguinte maneira: na Seção II faz uma breve introdução dos protocolos de comunicação da norma IEC 61850 utilizados para transmitir dados. Na Seção III são apresentadas as tecnologias implementadas junto ao software que tornam a aplicação viável. Na Seção IV é mostrado as interfaces de configuração do programa e a ideia da lógica implementada. O desempenho do algoritmo é mostrado na Seção V, seguida pela apresentação das conclusões na Seção VI.

II. PROTOCOLOS DA NORMAL IEC 61850

A norma IEC 61850 é um padrão internacional fundamental no campo da automação de sistemas de energia elétrica. Desenvolvida pela Comissão Eletrotécnica Internacional (IEC), essa norma estabelece diretrizes para a comunicação, controle e gerenciamento de equipamentos em subestações e redes elétricas. A IEC 61850 tem como objetivo principal aumentar a eficiência, confiabilidade e interoperabilidade das redes elétricas por meio da adoção de padrões comuns.

Dentro norma IEC 61850, existem dois protocolos que foram explorados neste trabalho, o *Sampled Values* (SV) e o Generic Object Oriented Substation Event (GOOSE).

O protocolo *Sampled Values* é projetado para a transmissão eficiente e precisa de dados de medição, como corrente e tensão, a partir dos dispositivos de campo para os sistemas de automação. Ele permite a transmissão em tempo real desses valores, com alta precisão e sincronização, o que é crucial para aplicações que demandam monitoramento e controle em tempo real.

O protocolo GOOSE é uma parte integral da IEC 61850, focado na rápida troca de informações de eventos entre dispositivos em uma subestação. Ele permite a comunicação rápida de eventos críticos, como alarmes ou mudanças de estado, sem a necessidade de um endereçamento explícito. Isso reduz o tempo de reação dos sistemas de proteção e controle, contribuindo para a segurança e estabilidade das redes elétricas.

III. TECNOLOGIAS IMPLEMENTADAS

Conforme mencionado na Seção I, inicialmente o software foi desenvolvido para operar em um servidor dentro de uma subestação, servindo como ferramenta de teste para futuros

IEDs virtuais. No entanto, a sua utilização também é viável na avaliação de IEDs microprocessados convencionais, desde que seja integrado em um ambiente que assegure a precisão requerida para essa finalidade. A seguir, apresentam-se as tecnologias empregadas neste trabalho que viabilizaram a aplicação do testador virtual.

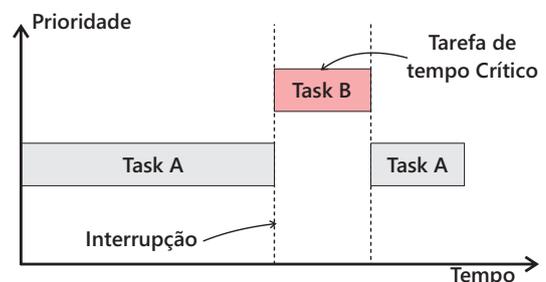
A. Real-Time kernel

O *kernel* Linux em tempo real (*Fully Preemptible Kernel*) [9], é uma versão especializada do *kernel* projetada para sistemas que requerem tempos de resposta previsíveis e consistentes. O motivo para utilizar esse tipo de *kernel* é a garantia de determinismo e consistência na execução de tarefas.

Nesses ambientes, a previsibilidade é crucial para a execução correta do sistema. Este *kernel* permite que as tarefas sejam programadas com prioridades específicas e tempos de execução garantidos. Isso garante que tarefas críticas sejam concluídas dentro de limites estritos, reduzindo riscos, melhorando a eficiência e mantendo a integridade do sistema.

Para que o software seja utilizado em um ambiente de testes controlado, os pacotes *Sampled Value* precisam ser transmitidos em uma taxa fixa e sincronizados com a rede ethernet da subestação. Neste contexto, para que não haja atraso no envio do pacote, a utilização do RT-Kernel é necessária, pois este consegue priorizar tarefas de tempo crítico e disponibiliza a utilização de interrupções imediatas dentro do sistema operacional.

Figura 1: Exemplo de uma interrupção entre tarefas no *kernel* fully preemptible



Como mostrado na Figura 1 a interrupção do *kernel* é o principal motivo da sua utilização. Ele é projetado para responder a essas interrupções no momento em que elas são invocadas, garantindo que a tarefa crítica seja executada sem atrasos. Isso é essencial para cumprir os requisitos de tempo real e garantir que os pacotes de SV sejam transmitidos nos intervalos fixos necessários.

B. KVM

Kernel-based Virtual Machine (KVM) [10] é um *hypervisor* do tipo *bare-metal* incorporado ao próprio *kernel* Linux. E por esse motivo, esse *hypervisor* permite a execução de máquinas virtuais (VM) ou containers utilizando diretamente o hardware do servidor, ou seja, não há necessidade de um Sistema Operacional adicional entre os recursos físicos do servidor e as VM e containers.

C. Open vSwitch

Considerando novamente a utilização do programa em um servidor, o Open vSwitch (OVS) [11] é um software de comutação virtual utilizado como uma alternativa dos Switch físicos em ambientes de virtualização de redes.

Essa tecnologia permite a criação de portas virtuais que podem ser conectadas a máquinas virtuais ou contêineres. Ele oferece recursos avançados de rede, como encaminhamento baseado em VLANs, balanceamento de carga, tunelamento VXLAN e GRE e monitoramento de rede.

Neste trabalho, o OVS foi implementado em conjunto com o *hypervisor* KVM e o software DPDK para controle do fluxo de rede e direcionamento de vLans dentro do servidor.

D. DPDK

O *Data Plane Development Kit* (DPDK) [12] é um conjunto de bibliotecas e drivers utilizado em ambientes de redes de alta densidade de pacotes. Ele possibilita a aceleração de software utilizando das capacidades de hardware modernas, como processadores multi-core e controladores de rede de alto desempenho.

Neste trabalho, o DPDK foi aplicado para melhoria no desempenho de envios e recebimentos dos pacotes GOOSE e SV. Ele também possui a capacidade de ser utilizado como mediador para interfaces de redes que necessitam de API próprias (*Smart NICs*) e as máquinas virtuais do servidor. Porém, no escopo deste estudo, essa aplicação não foi explorada.

IV. TESTADOR VIRTUAL

As arquiteturas apresentadas a seguir tiveram como finalidade apresentar uma prova de conceito de um testador de relés digitais que utilizam a norma IEC 61850, partindo de uma metodologia puramente de software, e visando a concepção de virtualização dos dispositivos de proteção (IEDs) em uma subestação.

Para tal, foi desenvolvido uma arquitetura de software de baixo nível para ser utilizada dentro de uma máquina virtual ou container em um servidor. Nela, foram utilizadas ferramentas de tempo crítico descritas na Seção III com o intuito de se aproximar de uma precisão de hardware de um equipamento de teste convencional.

O controle do testador virtual é realizado por meio de um programa *desktop* amigável para o usuário desenvolvido para Windows. Nele é possível de parametrizar os parâmetros das mensagens Sample Value e GOOSE, e permite a realização de dois modos de ensaio, um dele é o envio contínuo de pacotes *Sampled Value* e o outro é reprodução de sequências definidas pelo usuário.

Nos tópicos a seguir, serão mostradas as interfaces do programa de controle e a lógica do algoritmo implementado no testador virtual.

A. Telas do Software de controle

Tela Inicial

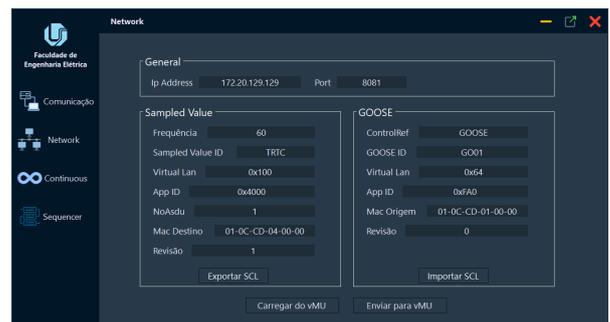
Figura 2: Tela inicial do software de controle.



Comunicação com a Máquina Virtual no servidor

Nessa tela é configurado os parâmetros do pacote Sampled Value que será enviado e o pacote GOOSE que será recebido

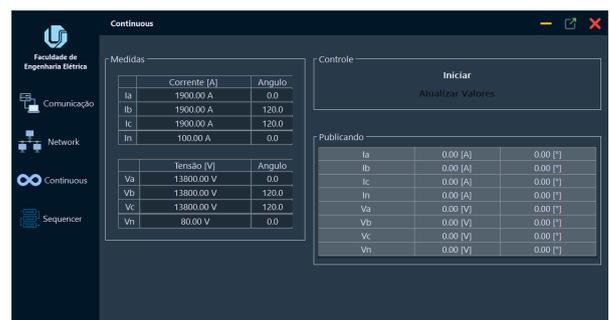
Figura 3: Tela comunicação do software de controle.



Nesta interface é parametrizado o teste de envio contínuo dos pacotes Sampled Value, nele é possível determinar os valores fasoriais das correntes e tensões. Uma vez que o teste já esteja iniciado, existe também a possibilidade de atualizar esses valores em tempo real.

Ensaio de envio contínuo

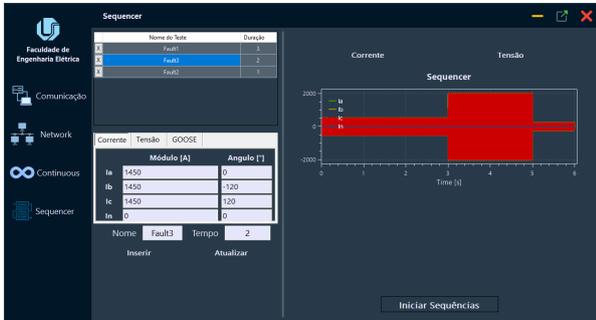
Figura 4: Tela do ensaio contínuo do software de controle.



Ensaio de reprodução de sequências

Neste tela é configurado o ensaio de sequência, este ensaio foi criado para o operador testar as funções de proteção do relê, podendo criar cenários de pre-faulta, faulta e pos-faulta.

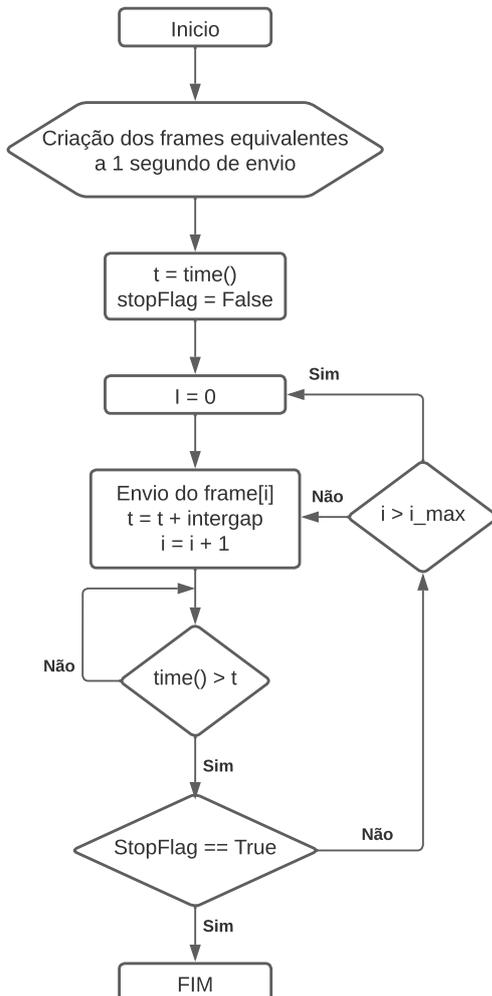
Figura 5: Tela do ensaio de sequências do software de controle.



B. Testador Virtual

O algoritmo foi escrito utilizando as linguagens Python e C, a preparação e criação dos pacotes foi realizada em Python e o processamento de envio e recebimento na linguagem C.

Figura 6: Fluxograma da lógica implementada no envio de pacotes SV



Durante a execução do ensaio, são executados os seguintes serviços de forma independentes:

- Script para interface de controle entre o algoritmo e o software de controle;
- Script para criação do buffer contendo os frames;
- Script para envio dos pacotes *Sampled Value*;
- Script para recebimento e processamento dos pacotes GOOSE.

No fluxograma da Figura 6 é indicado a lógica implementada para envio dos frames.

O *intergap* é o tempo de espera entre cada envio de pacote, nos testes de aplicabilidade do programa, foi considerado a frequência de amostragem do SV *light edition* igual a 4800Hz, ocasionando no *intergap* de 208,3333us. A variável de parada "stopFlag" pode ser alterada pelo usuário utilizando o software de controle ou por meio de um pacote GOOSE.

V. RESULTADOS

Os ensaios apresentados a seguir foram implementados em uma máquina virtual ou container inseridos dentro do servidor Dell PowerEdge R730. As especificações de ambos são indicadas nas tabelas a seguir.

Componente	Máquina Virtual	Servidor
OS/Modelo	Linux Server 22.04	Dell PowerEdge R730
Modelo do Processador	Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz	
Adaptador de Rede	Intel(R) I350-AM4	
Número de Processadores	2	40
Memória HD	40 Gb	8,96 Tb
Memória RAM	4096 Mb	128 Gb
Interface de Rede	Virtualizada	Intel I350 1GbE RJ45 4-Port

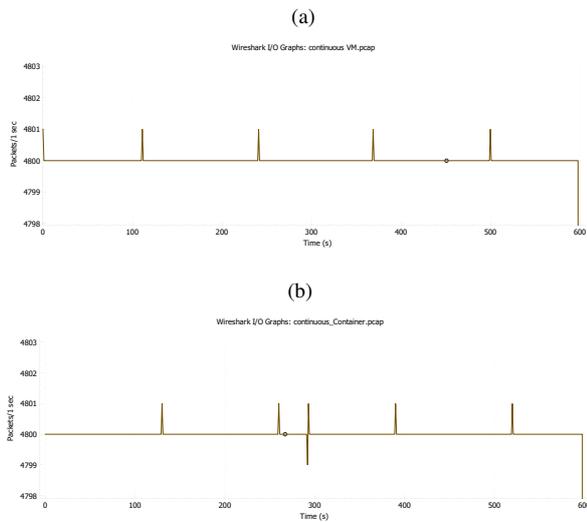
Tabela 1: Configurações do hardware e máquina virtual utilizada.

A fim de avaliar a aplicabilidade do testador virtual desenvolvido, os ensaios realizados foram voltados na avaliação da velocidade e constância no envio dos pacotes *Sampled Values* em uma máquina virtual e em container. Desse modo, espera-se que os resultados indiquem se a precisão obtida puramente em software assemelha-se a precisão de equipamento físicos.

Para tal, foi utilizado o modo de teste contínuo durante 10 minutos, os pacotes enviados pelo programa foram registrados utilizando o programa *tcpdump* do Linux em uma máquina virtual diferente do envio. Simulando, assim, o tempo em que um IED virtual receberia os dados do testador.

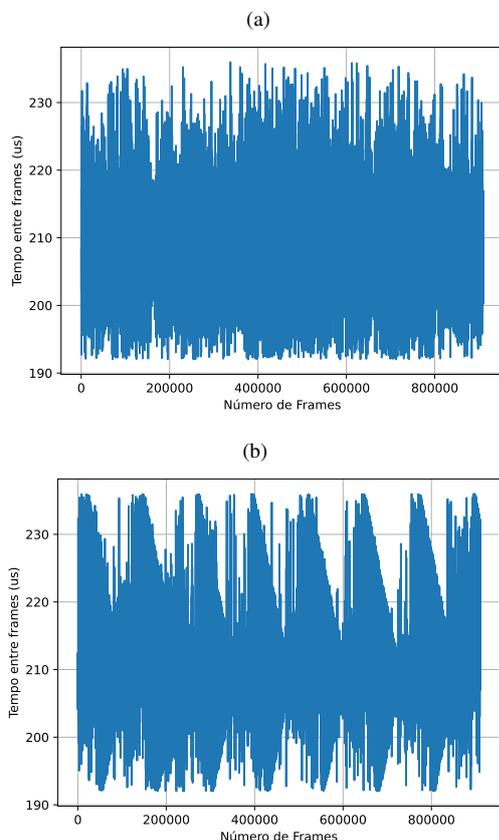
As figuras a seguir mostram a taxa de envio de pacotes por segundo no software Wireshark. Nessa análise, foi implementado programa em uma máquina virtual e em um container.

Figura 7: Valor médio de frames por segundo utilizando a ferramenta Wireshark, (a) Máquina Virtual e (b) Container.



Com base na Figura 7, o software desenvolvido consegue cumprir um envio fixo de 4800Hz durante longos períodos. Porém, analisando o tempo de envio entre cada frame individual, percebe-se que ocorreu uma certa variação entre os tempos de envio (*jitter*), nos gráficos a seguir demonstram o *jitter* durante o teste.

Figura 8: Resultado da variação do tempo entre frames (inter-gap) utilizando (a) Máquina Virtual e (b) Container.



Considerando uma frequência de envio de 4800Hz e os resultados da Figura 7, esperava-se um tempo entre frames constante de 208,3333us. Porém, como indicado pela Figura 8, esse valor variou-se entre 236us e 192us.

Após analisar os tempos de execução do código, foi verificado que a função de envio de pacotes era precisamente chamada a cada 208,3333us. Todavia, o tempo para o pacote ser enviado era volátil, resultando na variação supracitada.

Apesar da ocorrência do *jitter*, o erro introduzido no envio do pacote não era recorrente e os atrasos de um pacote era compensado pelo pacote seguinte. Logo, com base na análise estatística indicada na Tabela 2, o valor médio ficou constante e igual a 1/4800 e o desvio padrão e variância não foram significativos.

	Máquina Virtual	Container
Valor Médio	208,3330 us	208,3330 us
Máximo	232,9420 us	235,9900 us
Mínimo	194,0230 us	192,0060 us
Desvio Padrão	1,7544 us	1,7398 us
Variância	3,0781 us	3,0271 us

Tabela 2: Resultados estatísticos obtidos nos ensaios.

VI. CONCLUSÃO

Esse trabalho apresentou o desempenho alcançado pelo testador virtual frente a uma nova arquitetura centralizada do sistema PAC em uma subestação digital. Os resultados indicaram que o software, mesmo sendo um modelo unicamente virtual, possui capacidade de ser implementado para testes de proteção de tempo crítico em IEDs convencionais e virtuais, devido a baixa variação de tempo e consistência no envio dos pacotes Sampled Value.

REFERÊNCIAS

- [1] H. Zeynal, M. Eidiyani and D. Yazdanpanah, "Intelligent Substation Automation Systems for robust operation of smart grids," in *2014 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA)*, Kuala Lumpur, Malaysia, 2014, pp. 786-790
- [2] Q. Huang, "Smart Substation: State of the Art and Future Development," in *2018 IEEE Power & Energy Society General Meeting (PESGM)*, Portland, OR, USA, 2018, pp. 1-1.
- [3] Y. R. Kafle, K. Mahmud, S. Morsalin and G. E. Town, "Towards an internet of energy," in *2016 IEEE International Conference on Power System Technology (POWERCON)*, Wollongong, NSW, Australia, 2016, pp. 1-6.
- [4] IEC. *IEC 61850: Communication networks and systems for power utility automation*. Edição 2.1. International Electrotechnical Commission, 2018.
- [5] Q. Song et al., "Smart substation integration technology and its application in distribution power grid," in *CSEE*

- Journal of Power and Energy Systems*, vol. 2, no. 4, pp. 31-36, dezembro de 2016.
- [6] N. Kabbara et al., "Towards Software-Defined Protection, Automation, and Control in Power Systems: Concepts, State of the Art, and Future Challenges," in *Energies*, vol. 15, no. 24, p. 9362, 10 de dezembro de 2022.
- [7] N. Das, Tin Tun Aung and S. Islam, "Process-to-bay level peer-to-peer network delay in IEC 61850 substation communication systems," in *2013 Australasian Universities Power Engineering Conference (AUPEC)*, Hobart, TAS, Australia, 2013, pp. 1-6.
- [8] Kalkitech. "Virtual Protection Relay: Concepts and Applications." Whitepaper. Disponível em: <https://kalkitech.com/whitepaper-virtual-protection-relay/> Acessado em: 22/08/2023.
- [9] Linux Foundation. "Real-Time Linux Project." Disponível em: <https://wiki.linuxfoundation.org/realtime/> Acessado em: 22/08/2023.
- [10] Linux-KVM. "Kernel-based Virtual Machine (KVM)." Disponível em: <https://www.linux-kvm.org/> Acessado em: 22/08/2023.
- [11] Open vSwitch Project. "Open vSwitch." Disponível em: <https://www.openvswitch.org/> Acessado em: 22/08/2023.
- [12] DPDK Project. "Data Plane Development Kit (DPDK)." Disponível em: <https://www.dpdk.org/> Acessado em: 22/08/2023.