Comunicação Resiliente e Autonômica para Proteção e Controle das Smart Grids

Yona Lopes, Natalia Castro Fernandes, Débora Christina Muchaluat-Saade

***Resumo*— Nas redes elétricas inteligentes, todo o sistema elétrico de potência passa a ser interconectado, incluindo o usuário final. Pontos de geração distribuída passam a se interligar com o sistema elétrico de potência introduzindo a necessidade de implementação de soluções de proteção e controle semelhantes às usadas em redes de alta tensão. Com isso, aumenta a exigência por resiliência nas redes de comunicação da distribuição. Contudo, os métodos de resiliência usados atualmente ainda não atendem aos requisitos de QoS das *smart grids*. Esse artigo propõe o *framework* ARES para prover uma comunicação resiliente e autonômica para as *smart grids*. A proposta é avaliada e comparada às principais propostas da literatura, alcançando tempos de recuperação de falha de no máximo 610 microssegundos, o que representa um grande avanço com relação às demais propostas. Além disso, o ARES é implementado de forma transparente para os dispositivos finais, guardando compatibilidade com equipamentos de medição e atuação legados.**

***Palavras-Chave*— Redes Elétricas Inteligentes, Recuperação de Falhas, Qualidade de Serviço, Arquitetura de Redes, Redes Definidas por Software.**

***Abstract*— In smart grids, the power system communication is interconnected, including final users. The broad usage of distributed energy resources in distribution networks introduces the need of protective relaying schemes similar to those used in high voltage networks. Hence, power systems experience an increased demand for resilience in the distribution communication network. However, the resilience methods currently in use still cannot meet these protection requirements. This work proposes ARES, a framework for autonomic and resilient communication for smart grids. The proposal is evaluated and compared to the main proposals of the literature. ARES presents maximum recovery time of 610 microseconds, which is an important advance compared to the other proposals. In addition, ARES is transparent to end devices, keeping compatibility with legacy measurement and actuation devices.**

***Keywords*— Smart Grids, Failure Recovery, Quality of Service, Network Architecture, Software Defined Networking.**

I. Introdução

Atualmente, a matriz energética existente depende principalmente de matéria prima finita, dando a ela um caráter frágil e de rendimento questionável. Somado a isso, existem ainda as interferências de cunho político, econômico e ambiental, que muitas vezes impedem o seu crescimento. Esta fragilidade se contrasta com o aumento do consumo energético e da necessidade de um fornecimento de energia de alta qualidade aos consumidores, deixando o setor elétrico cada vez mais interessado na modernização do Sistema Elétrico de Potência (SEP). Com esse intuito, a rede elétrica inteligente, conhecida como *smart grid*, traz propostas inovadoras que mudam de forma profunda a maneira como a energia é provida desde a geração até os consumidores finais. Cabe observar que, para que o desenvolvimento da rede elétrica inteligente seja possível, a comunicação entre dispositivos, antes existente apenas em parte do SEP, se torna imprescindível da geração até o consumidor final [1].

Para implementação segura de esquemas de proteção, as mensagens de controle da rede elétrica precisam ser trocadas com atrasos máximo na ordem de milissegundos [2], o que exige uma alta Qualidade de Serviço (*Quality of Service* (QoS)) nas redes de comunicação para os sistemas elétricos inteligentes. No entanto, os métodos usados para recuperação de falha atualmente ainda enfrentam muitos problemas para prover a resiliência com esta qualidade [3], [4], [5], [6]. Muitos possuem tempos de recuperação de falhas na rede de comunicação muito altos, o que inviabiliza o seu uso para uma rígida restrição temporal. Outros possuem premissas para implementação muito rígidas e não escaláveis.

Esse trabalho propõe um *framework* de comunicação resiliente, baseado em redes definidas por *software*, que atende as rígidas necessidades da proteção do sistema elétrico de potência desde a geração até o consumidor final. O *framework*, intitulado ARES (*Autonomic and Resilient Environment for Smart Grids*), permite que a rede de comunicação se recupere de falhas em microssegundos se mostrando melhor do que os métodos usados atualmente. Além disso, visando escalabilidade e simplicidade, a rede é configurada de forma autonômica, permitindo que uma grande quantidade de pontos sejam colocados na rede sem necessidade de configuração manual.

O ARES foi implementado e testado usando o controlador RYU e o emulador de redes Mininet [7]. O *framework* configura a rede automaticamente, identificando dispositivos conectados, construindo árvores multicast para reduzir o impacto da inundações de camada 2 [2], além de fazer a recuperação de falhas em tempo real. De fato, a principal contribuição do ARES é relacionada a redução do tempo de recuperação de falhas para microssegundos, muito menor que o RSTP, o qual é o protocolo mais utilizado atualmente em redes de proteção. Além disso, o ARES é transparente para os dispositivos finais, que não precisam de nenhuma modificação de hardware ou software. Com isso, o ARES possui compatibilidade com equipamentos de medição e atuação legados.

O restante do trabalho está estruturado da seguinte forma. Os requisitos de QoS são abordados na Seção II. Os trabalhos relacionados são discutidos na Seção III. Na Seção IV, a proposta é detalhada e os resultados são discutidos na Seção V. Por fim, a Seção VI conclui o trabalho.

Yona Lopes, Débora C. Muchaluat-Saade (Departamento de Ciência da Computação), Natalia C. Fernandes (Departamento de Engenharia de Telecomunicações) - Universidade Federal Fluminense (UFF), Niterói– RJ, E-mails: yonalopes,nataliacf,deboracms@id.uff.br. Este trabalho foi parcialmente financiado pela CAPES, FAPERJ e pelo CNPq.

II. Qualidade de Serviço em *Smart Grids*

Muitas das aplicações de energia para *smart grids* têm exigências rigorosas em termos de disponibilidade e atraso na comunicação [8]. Nesse sentido, algumas iniciativas para definir valores de atraso aceitáveis nessas redes têm sido feitas. A norma IEC 61850-7 [2], de 2009, aborda a padronização da comunicação para os recursos de energia distribuídos usando o mesmo limiar temporal estabelecido para proteção e controle de subestações. Com isso, são recomendados para proteção valores de atraso de 3ms até 100ms de acordo com o tipo de mensagem utilizada[1]. Também nesse sentido, o departamento de Energia dos Estados Unidos, em 2010, analisou os requisitos de comunicação para *smart grids* e definiu valores de latência máximos de milissegundos e níveis de confiabilidade para cada aplicação da *smart grid* [4]. Isso confirma a necessidade de implementação de uma rede de comunicação resiliente entre os dispositivos que compõem o sistema.

A. Resiliência em Smart Grids

Uma falha no sistema elétrico de potência produz uma série de mensagens de proteção e controle para que a falha seja isolada e não se propague para o restante da malha elétrica. Esse comportamento pode gerar uma falha em cascata, onde falhas na rede elétrica podem produzir falhas na rede de comunicação. Caso as mensagens de proteção e controle não sejam entregues ao destino devido à uma falha na comunicação, os equipamentos de proteção não atuam, deixando o sistema desprotegido, o que pode resultar em falhas elétricas de enormes proporções.

Apesar dos avanços tecnológicos obtidos nessa área, a garantia de resiliência evitando que a comunicação fique fora por um tempo que afete o sistema ainda é um desafio. O *Rapid Spanning Tree Protocol* (RSTP) [9], amplamente utilizado na camada de enlace, tem tempos de recuperação de até alguns segundos, valor bastante distante dos limiares exigidos para *smart grids* [3].

Apesar de existirem versões mais recentes do RSTP que apresentam tempos de recuperação da rede muito melhores, estes são geralmente soluções proprietárias ou envolvem uma topologia muito específica e configurações manuais dos dispositivos [10].

Outros dois protocolos que foram propostos para uso em redes de comunicação de subestações que apresentam um baixo tempo para recuperação em caso de falha são o *Parallel Redundancy Protocol* (PRP) e o *High-availability Seamless Redundancy* (HSR) [11], ambos da *International Electrotechnical Commission* (IEC). No PRP, o método consiste no envio duplicado do pacote por duas redes distintas e similares. Com isso, caso uma rede sofra uma falha, o pacote enviado pela outra rede chegará. O HSR funciona de forma similar, porém na topologia em anel. O dispositivo envia o pacote duplicado, um por cada porta, de forma que em caso de rompimento de um lado do anel, o outro pacote chegará. Dessa forma, considera-se que o sistema não fica indisponível em caso de falha. No entanto, este tipo de solução não é escalável, sendo insuficiente para a introdução massiva dos recursos de geração distribuída e proteção em redes de distribuição elétrica [3]. A implementação de duas redes semelhantes e independentes entre todos os dispositivos deixa a solução cara, sendo economicamente viável apenas para subestações.

[1]Valores relacionados as mensagens GOOSE ou SV da norma IEC 61850.

III. Trabalhos Relacionados

Tendo em vista a necessidade de resiliência, o alto tempo de recuperação do RSTP e as premissas para a instalação do PRP e do HSR, alguns autores propõem outras soluções. Selga et al. [3] propõem variações nos algoritmos de cálculo , de forma a diminuir o tempo de convergência. Baseado no SPB (*Shortest Path Bridging*), protocolo especificado no IEEE 802.1AQ, e no TRILL, os autores propõe uma solução baseada na junção de ambos os métodos para aplicação em *smart grids*. Porém, a solução é baseada no encapsulamento do frame ethernet tradicional, o que imprime atraso na comunicação, característica que os autores confirmam com os testes. Como solução, os autores propõem combinações entre o método e o PRP o que resulta nos mesmos problemas descritos para o PRP, inviabilizando o uso da proposta para o cenário em questão.

Outro ponto importante de trabalhos relacionados ao ARES, além dos protocolos para provimento de recuperação de falhas, diz respeito ao uso de SDNs (*Software Defined Networks*) em redes de comunicação para *smart grids*. Goodney et al. [12] demonstram que as redes SDN podem ser usadas como uma boa opção de arcabouço para o SEP, com a implementação de uma rede *multicast* para uso com PMUs (*Phasor Measurement Unit*) e mostram que, com SDN, é possível atingir valores de atraso muito bons. Sydney et al. [13] propõe o uso de OpenFlow para redes MPLS. Cahn et al. [14] propõe o SDECN (*Software-Defined Energy Communication Network*) para uso em subestacões. Também para uso em subestações, Lopes et al. [6] propõe o SMARTFlow, mostrando que o acabouço OpenFlow atende muito bem aos tempos de atraso na rede imposto pela Norma IEC 61850. No entanto, os trabalhos citados têm enfoque apenas no tempo de atraso das mensagens na rede e não em recuperação da rede em caso de falha. Pfeiffenberger et al. [5] e Reitblatt et al. [15] abordam o uso do OpenFlow 1.3 e do grupo *fast failover* para *smart grids*, ressaltando as vantagens do uso do mecanismo para recuperação de falhas. Porém, Pfeiffenberger et al. não descrevem os algoritmos usados nem implementam a solução para testes e avaliação. Reitblatt et al., apesar de descreverem com mais detalhes a solução, intitulada Fattire, não apresentam avaliações motivadoras.

IV. A proposta ARES

Com o intuito de prover uma comunicação resiliente, robusta e flexível para as *smart grids* e permitir uma interação inteligente entre os recursos de energia distribuídos, cargas e sistemas de gerenciamento, é proposto o ARES (*Autonomic and Resilient Environment for Smart Grids*). O ambiente é baseado em redes definidas por *software* e segue os requisitos

temporais e a modelagem da Norma IEC 61850, visando interoperabilidade.

O ARES tem como um de seus principais objetivos prover informações importantes para as aplicações de energia permitindo a implementação de um sistema de controle e supervisão eficiente e automático onde os recursos de energia distribuídos e as cargas podem ser automaticamente mapeados e agrupados no sistema. Além disso, a proposta objetiva evitar que falhas nas redes de comunicação interfiram nas redes elétricas inteligentes. Para isso, é necessário que, em caso de falhas na rede de comunicação, essa seja capaz de se recuperar de forma transparente para as aplicações elétricas com restrição temporal rígida. Como a proposta também objetiva facilitar o planejamento e a configuração da rede de comunicação para o setor elétrico, o ARES permite que a rede de comunicação seja autonômica e com a qualidade de serviço exigida.

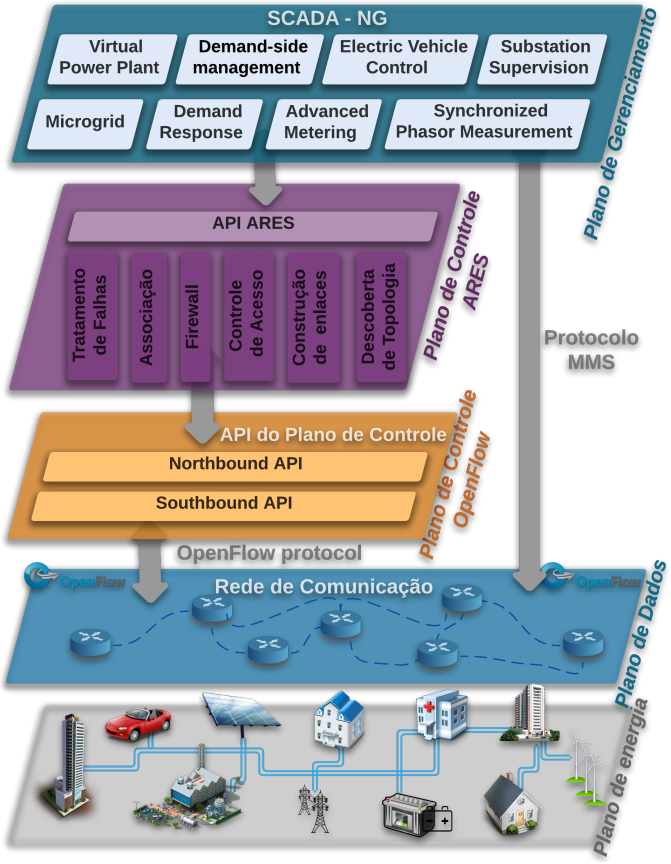


Fig. 1. Estrutura ARES e seus componentes do plano de controle

O *framework* ARES, ilustrado na Fig. 1, é modelado em 5 planos que trocam informações entre si. O plano superior é intitulado plano de gerenciamento e é onde se localizam os sistemas de supervisão. Propõe-se que o sistema localizado nesse plano seja intitulado SCADA-NG (*Next Generation*) por ser uma extensão do *Supervisory Control and Data Acquisition* (SCADA) atual. Com isso, o SCADA-NG engloba, além do supervisório tradicional, as aplicações de energia para as *smart grids* que demandam interação com o núcleo da rede. Além de controlar e coletar dados das subestações, o SCADA-NG também interage com os sistemas de *Distributed Energy Resources* (DERs), veículos elétricos e medidores inteligentes. Assim como o SCADA tradicional, o SCADA-NG é configurável de forma que a concessionária visualize apenas aplicações de energia de interesse. Como o plano de gerenciamento troca informações com o plano inferior, as aplicações de energia do SCADA-NG têm mapeados e relacionados, de forma automática, o sistema e seus componentes como os recursos de energia distribuídos e as cargas, tornando as aplicações de energia mais escaláveis e flexíveis permitindo a execução de aplicações de tempo real da *smart grid*.

O plano de controle ARES é o plano seguinte e é responsável pelo controle da rede de comunicação, tarefa efetuada pelos componentes ARES que mapeiam, calculam e configuram os *switches* da rede. Dentre os componentes ARES, está o `Tratamento de Falhas`, que é detalhado na Seção IV-A. Os demais módulos são responsáveis pela implementação segura de enlaces e pela detecção de dispositivos no sistema. Para isso, com base nas informações passada pelas aplicações de energia do SCADA para a API ARES, os componentes definem as regras de acesso e configuram os dispositivos da rede com base nessas regras. Com base nas informações passadas pelos componentes de descoberta da rede e associação, a API ARES informa para o SCADA-NG qual dispositivo conectou ou desconectou da microgrid. Além disso, esse plano contém a API ARES, responsável pela interação entre as aplicações de energia do SCADA-NG e os componentes ARES. Esses componentes podem ser executados sobre um ou mais controladores de rede e podem ser implementados em qualquer controlador OpenFlow. A API ARES fornece os serviços que a rede pode prestar para o SCADA-NG, tais como escolha de rota e construção de enlaces de forma segura, detecção de entradas e saídas de cargas e recursos de geração distribuídos incluíndo veículos elétricos e baterias, definição e configuração de regras de segurança, definição e configuração de controle de acesso, de forma simples e bem estruturada. Isso torna o *framework* mais modular, permitido a proposição de novas aplicações para o SCADA-NG que dependam do suporte provido pela rede.

Na sequência, o plano de controle OpenFlow se comunica de forma bidirecional com o plano de dados, onde se localizam os *switches* da rede, que podem ser configurados de forma automática e eficiente.

O último plano representa os dispositivos finais das redes elétricas inteligentes, intitulado plano de energia. Neste plano, se localizam as construções inteligentes, medidores inteligentes, DERs, e todo dispositivo que seja considerado uma carga, um gerador, ou ainda uma forma de armazenamento. O SCADA-NG se comunica com esses dispositivos utilizando a rede OpenFlow e o protocolo MMS, conforme estabelecido pelo IEC 61850.

A. Componentes ARES e Resiliência das Redes

Os componentes ARES são baseados em configurações proativas, reativas e híbridas com o uso do OpenFlow. Quando proativas, os componentes calculam as configurações necessárias e configuram automaticamente a rede assim que os

switches são ligados. Quando reativas, a rede reage a determinado evento, como a entrada/saída de um recurso na rede, e as configurações são feitas também de forma automática, porém de acordo com esse evento. Contudo, uma abordagem híbrida também é utilizada visando maior velocidade de configuração e robustez.

Para que a rede de comunicação das *smart grids* seja resiliente, é proposto o componente `Tratamento de Falhas`. O componente é responsável por construir os enlaces *multicast* e *unicast* da rede. Esse componente pode ser chamado pela API ARES, quando uma aplicação de energia precisa configurar os seus recursos, por outro componente que precise configurar os enlaces, ou ainda na inicialização da rede. O componente é capaz de achar o caminho mais curto entre origem e destino, além de gerar caminhos secundários que podem ser usados em caso de falha no caminho principal. O componente `Tratamento de Falhas` objetiva permitir a recuperação de falhas de forma eficiente na rede sendo transparente para os dispositivos finais. Para isso, o componente usa uma tabela de grupo OpenFlow, com um tipo de grupo intitulado *fast failover*, de forma que opções de encaminhamento em caso de falha fiquem instaladas previamente no *switch*. Assim, tem-se opções de saída com prioridades distintas. Caminhos melhores recebem maior prioridade. Como o *switch* executa o caminho ativo de maior prioridade primeiro, caso o caminho principal sofra uma falha o *switch* imediatamente encaminhará o pacote para a saída descrita na próxima opção ativa.

O componente segue o Algoritmo 1, que recebe como entrada os eventos da rede, sejam eventos de falha ou de inicialização, os grupos *multicast*, e os pares *unicast* com origem e destino de elementos que precisam comunicar. Essas informações são providas pela API ARES, por outros componentes ARES ou ainda pela API do plano de controle OpenFlow. Ao final do procedimento, todas as regras de encaminhamento, incluindo os caminhos de recuperação de falhas, são inseridos nos switches. Inicialmente a lista de possíveis caminhos está vazia. O algoritmo calcula para cada possível origem e destino(s) da rede o menor caminho. Em seguida, como descrito a partir da linha 3, para cada switch de cada possível caminho, ele remove a porta de saída escolhida na topologia (linha 5), simulando uma falha naquela porta e recalcula o caminho a partir daquele switch (linhas 6-8), caso ele exista. Com isso, tem-se um caminho secundário para uma falha que ocorra no enlace, porta ou equipamento associado a essa porta de saída. Este cálculo é refeito para todas as opções de falha de forma que possam ser pré-configurados no *switch* os caminhos alternativos em caso de falha como grupos *failovers*. Após criar o grupo que reconfigura o switch em caso de falha (linha 10), esse grupo é adicionado ao conjunto de regras (linha 11), assim como toda a configuração do restante caminho auxiliar para caso falha (linhas 12-14). Por fim, a entrada do caminho principal para a qual foi calculada a falha também é adicionada (linha 16), a topologia original é reestabelecida (15) e o cálculo continua até todos os caminhos terem sido estabelecidos.

Algoritmo 1: Algoritmo de detecção de falhas e restauração da rede

Input: ofp_event,$topo\_net$,$multicast$,$unicast$

1 $paths = []$
2 $list\_flows\_temp = calc\_paths(topo\_net, multicast, unicast)$
3 $list\_flows = []$
4 **for** $entry$ ***in*** $list\_flows\_temp$ **do**
5 $\quad topo\_net.remove_{port}(entry)$
6 $\quad multicast = []$
7 $\quad unicast = [entry.sw, entry.dst]$
8 $\quad flows\_failover = calc\_paths(topo\_net, multicast, unicast)$
9 $\quad$ **if** $len(flows\_failover) > 0$ **then**
10 $\quad\quad group\_entry = create\_group(entry, flows\_failover[0])$
11 $\quad\quad list\_flows.append(group\_entry)$
12 $\quad\quad$ **for** i ***in*** $(1..len(flows\_failover) - 1)$ **do**
13 $\quad\quad\quad list\_flows.append(flows\_failover[i])$
14 $\quad\quad$ **end**
15 $\quad$ **end**
16 $\quad list\_flows.append(entry)$
17 $\quad topo\_net.add_{port}(entry)$
18 **end**
19 $install\_paths(list\_flows)$

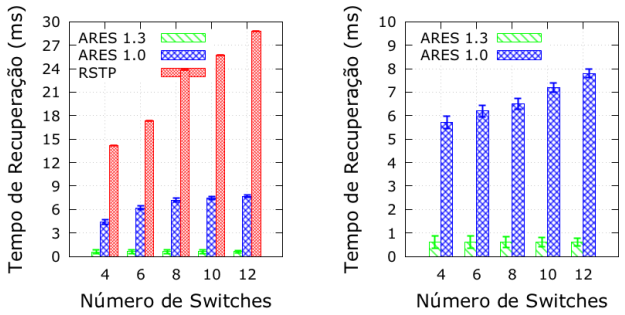
V. Experimentos Emulados e Análise dos Resultados

Os componentes ARES foram desenvolvidos em Python e implementados no controlador RYU na versão 1.0 e 1.3 do OpenFlow. Os experimentos foram emulados usando o Mininet[2] [7] versão 2.2.1. Foi criado um módulo no Mininet que constrói topologias e usa o gerador de pacotes GEESE [16] para emular um tráfego real de mensagens GOOSE, conforme estabelecido pela IEC 61850. A API ARES foi desenvolvida também em Python e implementada na mesma máquina do controlador e do Mininet. Os testes foram executados com 20 rodadas de 100 segundos de duração cada. As falhas foram realizadas em pontos e momentos aleatórios. Foram variados parâmetros como a quantidade de pontos finais na rede, quantidades de *switches*, tipo de topologia, quantidade de pontos por grupo *multicast* e método utilizado. Todos os resultados apresentam um intervalo de confiança de 95%.

Para os experimentos, levou-se em consideração a topologia em anel e em malha. Os testes foram feitos com uma quantidade de 4 até 12 *switches*, pois comportam uma grande quantidade de hosts e ainda estão dentro do valor suportado para emulação pelo mininet. De forma geral, após a estabilização da rede, um enlace é desconectado usado o comando *ifdown* do Linux em momentos aleatórios para cada rodada. O tempo de recomposição é computado subtraindo-se T_{fault} de T_{nc}, onde T_{fault} representa o tempo de entrega da mensagens do ponto A para o B no momento da falha e T_{nc} em condições

[2]O Mininet é uma plataforma flexível para emulação de redes OpenFlow que provê um ambiente de experimentação bem próximo do real.

normais da rede.



(a) Topologia em anel (b) Topologia em malha

Fig. 2. tempo de recuperação da rede em caso de falha.

Os gráficos da Fig. 2 apresentam o resultado para topologia em anel, Fig. 2(a), e para a topologia em malha, Fig. 2(b). Para fins de comparação, no gráfico da Fig. 2(a), também foram descritos os tempos encontrados por Pustylnik et al. [10] para o RSTP na versão melhorada incorporada no padrão 802.1D. No gráfico da Fig. 2(b) os tempos não foram plotados pois o RSTP na versão melhorada não foi testado para topologia em malha, somente em anel. Os outros métodos abordados nesse artigo, o PRP e o HSR, são métodos de redundância e não de recuperação, não cabendo comparação por terem características muitos distintas, como necessidade de duplicação da rede e do tráfego. O RSTP tradicional tem tempos muito altos saindo da ordem de milessegundos dos gráficos apresentados.

O cenário é composto por cinco grupos *multicast* distintos e dez dispositivos finais, distribuídos uniformemente entre os *switches*. Foi instanciado um gerador GEESE em cada dispositivo final a fim de gerar um tráfego de proteção e controle real. Os componentes ARES com arcabouço OpenFlow 1.0 e 1.3 foram implementados e testados no cenário descrito.

Na Fig. 2, nota-se que o tempo de recuperação na rede controlada pelo ARES não passou de 8 ms mesmo com a versão 1.0 OpenFlow[3]. Para o uso do OpenFlow 1.3, o ARES apresenta tempos de recuperação de falhas excelentes, não ultrapassando 0,6 ms. Isso mostra que o ARES, usando OpenFlow 1.0 ou 1.3, atende aos requisitos rígidos de tempo das *smart grids*, apresentando tempos melhores do que o RSTP.

VI. Conclusões e Trabalhos Futuros

O conceito de redes elétricas inteligentes tem trazido muitas vantagens, mas ainda existem muitos desafios a serem resolvidos. Relacionar e delimitar requisitos relacionados a QoS, definindo restrições temporais que atendam às necessidades dos novos sistemas de proteção [4], [8], ainda é um assunto muito discutido. Contudo, como citado por [3], os métodos usados atualmente para recuperação de falhas não atendem essa restrição.

Esse trabalho identificou e discutiu as questões relacionadas ao QoS das redes elétricas inteligentes em momentos de falha da rede, os métodos usados e suas vantagens e desvantagens. Além disso, foi proposto, desenvolvido e avaliado um *framework* resiliente e autonômico para atender os requisitos de QoS com resultados muito positivos. A proposta, intitulada ARES, foi capaz de recuperar a rede no cenário descrito muito mais rápido que uma versão aprimorada do RSTP, que mostrou um tempo de resposta ainda acima do desejável para as *smart grids*. O ARES pode ser implementado de forma transparente para os elementos finais, não aumenta o processamento dos dispositivos nem a quantidade de *switches* na rede como acontece com o PRP, nem duplica o tráfego na rede como o HSR. Assim, acredita-se que a proposta para estabelecimento de caminhos e recuperação de falhas avança o estado da arte, viabilizando a construção de ambientes de comunicação confiáveis para as *smart grids*.

Como trabalhos futuros, pretende-se implantar e testar os componentes de segurança e realizar todos os testes em uma rede real OpenFlow, pois acredita-se que, fora do ambiente emulado/virtualizado do Mininet, os tempos podem ser ainda melhores.

[3]Na versão 1.0 do OpenFlow, não existem os grupos de *fast failover*, o que leva a instanciação da rota alternativa previamente calculada apenas após o controlador da rede ser notificado da falha no enlace.

Referências

[1] Y. Lopes, N. C. Fernandes, and D. C. Muchaluat-Saade, "Geração Distribuída de Energia: Desafios e Perspectivas em Redes de Comunicação," in *Minicursos do XXXIII SBRC*, 1st ed. SBC, 2015, pp. 55–109.

[2] International Electrotechnical Commission, "IEC 61850-7-420: Basic communication structure - Distributed Energy Resources logical nodes," IEC, Tech. Rep., 2009.

[3] J. M. Selga, A. Zaballos, and J. Navarro, "Solutions to the computer networking challenges of the distribution smart grid," *IEEE Communications Letters*, vol. 17, no. 3, pp. 588–591, 2013.

[4] U.S. Department of Energy, "Communication requirements of smart grid technologies," International Electrotechnical Commission, Tech. Rep., Oct. 2010.

[5] T. Pfeiffenberger, J. L. Du, P. B. Arruda, and A. Anzaloni, "Reliable and flexible communications for power systems: Fault-tolerant multicast with sdn/openflow," in *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*, July 2015, pp. 1–6.

[6] Y. Lopes, N. C. Fernandes, C. A. M. Bastos, and D. C. Muchaluat-Saade, "SMARTFlow: A Solution for Autonomic Management and Control of Communication Networks for Smart Grids." 30th ACM SAC, 2015, pp. 2212–2217.

[7] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop," in *ACM SIGCOMM - Hotnets'10*, 2010.

[8] IEC, "IEC 61850: Communication networks and systems for power utility automation," International Electrotechnical Commission, Tech. Rep. IEC 61850, 2002- 2013.

[9] LAN/MAN Standards Committee, "801.1D: IEEE Standard for Local and metropolitan area networks - MAC Bridges," IEEE, Tech. Rep., 2004.

[10] M. Pustylnik, M. Zafirovic-Vukotic, and R. Moore, "Performance of the Rapid Spanning Tree Protocol in Ring Network Topology," Siemens, White Paper, 2007.

[11] International Electrotechnical Commission, "IEC 62439-3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR)," IEC, Tech. Rep. 62439, 2010.

[12] A. Goodney, S. Kumar, A. Ravi, and Y. H. Cho, "Efficient PMU networking with software defined networks," in *SmartGridComm*, 2013.

[13] A. Sydney, D. S. Ochs, C. Scoglio, D. Gruenbacher, and R. Miller, "Using GENI for experimental evaluation of Software Defined Networking in smart grids," in *Computer Networks*, 2014.

[14] A. Cahn, J. Hoyos, M. Hulse, and E. Keller, "Software-defined energy communication networks: From substation automation to future smart grids," in *IEEE SmartGridComm*, 2013.

[15] M. Reitblatt, M. Canini, A. Guha, and N. Foster, "FatTire: Declarative Fault Tolerance for Software-Defined Networks," *HotSDN*, 2013.

[16] Y. Lopes, D. C. Muchaluat-Saade, N. C. Fernandes, and M. Z. Fortes, "Geese: A traffic generator for performance and security evaluation of IEC 61850 networks," in *IEEE 24th ISIE*, June 2015, pp. 687–692.