

SMARTFlow: Uma Proposta para a Autoconfiguração de Redes de Subestação IEC 61850 Baseada em OpenFlow

Yona Lopes¹, Natalia Castro Fernandes², Carlos Alberto Malcher¹

¹Laboratório GTECCOM, ²Laboratório MidiaCom
Dep. de Telecomunicações - Universidade Federal Fluminense (UFF)

Abstract. *Smart grids depend on a solid foundation of communications. Although standards like IEC 61850 proposes solutions to achieve interoperability, the auto configuration and the automatic control of the communication network are still an open problem. This work proposes an efficient solution for autonomic management and control of communication networks for substations based on IEC 61850. The proposed solution, called SMARTFlow, uses OpenFlow in order to achieve granularity and flexibility in the treatment of data flows. SMARTFlow proactively calculates Layer 2 multicast trees to forward GOOSE and Sampled Values messages and reconfigures all flow entries in case of network failures. Also, the proposed system monitors the network and defines on-demand the configuration of client-server flows. The proposal was implemented and tested using Mininet and showed a total load up to 44 % lower than the load using typical switches. The proposal also showed advantages when compared to other multicast solutions, such as GMRP(GARP Multicast Registration Protocol)*

Resumo. *As Smart Grids dependem de uma base sólida de comunicação. Apesar de normas como a IEC 61850 apresentarem soluções para alcançar a interoperabilidade, ainda existem problemas em aberto com relação à autoconfiguração e ao controle automatizado da rede de comunicação. Esse trabalho propõe uma solução eficiente de controle e gerenciamento autônomo de redes de comunicação para subestações baseadas na norma IEC 61850. A solução proposta, chamada de SMARTFlow, é baseada no uso do OpenFlow para obter granularidade e flexibilidade no tratamento dos fluxos de dados. O SMARTFlow, pró-ativamente, calcula as árvores multicast de camada 2 para encaminhamento de mensagens GOOSE e Sampled Values e reconfigura todas as entradas de fluxos em caso de falhas, além de monitorar a rede, definindo sob demanda a configuração de fluxos cliente-servidor. A proposta foi implementada e testada utilizando o emulador Mininet e apresentou uma carga total até 44% menor do que a de switches típicos. A proposta também demonstrou vantagens quando comparada com outras soluções de multicast, como o GMRP.*

1. Introdução

A energia elétrica é essencial para o desenvolvimento da sociedade. A falta de energia, mesmo que por um período curto de tempo, acarreta em prejuízos enormes além de danos causados pela falta de serviços essenciais. Por exemplo, no Brasil, uma auditoria do Tribunal de Contas da União mostrou que os apagões de 2001 e 2002 causaram um prejuízo de R\$ 45,2 bilhões para os brasileiros¹. O responsável por prover a qualidade do serviço elétrico oferecido aos consumidores, controlando e direcionando o fluxo de energia e fornecendo energia durante o maior tempo possível aos consumidores finais é o

¹<http://www.correiobraziliense.com.br> e <http://epocaestadobrasil.wordpress.com>

Sistema Elétrico de Potência (SEP), que possui uma infraestrutura complexa para gerar, transmitir e distribuir essa energia. Contudo, o SEP está sujeito a anormalidades e defeitos em sua operação. Para que o impacto das falhas seja reduzido, sistemas de proteção e supervisão são indispensáveis.

Apesar da importância do SEP, pouca inovação tem sido feita nos últimos anos, e, conseqüentemente, os equipamentos utilizados e as tecnologias, algumas vezes, são os mesmos de 40 anos atrás [Gungor et al. 2011]. Da necessidade inevitável de modernização dos sistemas de automação do SEP, surgiram as *Smart Grids*, tornando imprescindível a implantação de um sistema de comunicação mais “inteligente” interligando os sistemas de proteção e supervisão e os centros de controle [Lopes et al. 2012]. Uma das principais normas que especificam essa comunicação inteligente é a norma IEC 61850 [IEC 61850]. Apesar da norma IEC 61850 apresentar soluções para modelagem da comunicação dentro do SEP e interoperabilidade dos equipamentos, ainda existem problemas em aberto. A comunicação de mensagens prioritárias da norma, *Sampled Values*(SV) e GOOSE, é feita em camada 2 e com o uso de *multicast* [IEC 61850]. Contudo, devido ao comportamento padrão dos *switches* para *multicast* de camada 2, essas mensagens são enviadas em *broadcast*. Esse comportamento traz uma sobrecarga tanto para rede quanto para os dispositivos finais que recebem mensagens mesmo não estando no grupo [Sivanthi and Goerlitz 2013]. Existem protocolos dinâmicos para encaminhamento *multicast* de camada 2, como o GMRP (*GARP Multicast Registration Protocol*), padronizado pelo IEEE 802.1D. Porém, essa solução exige que o protocolo seja implementado nos dispositivos finais, o que ainda não ocorre em redes IEC 61850.

Outra questão importante é relativa aos sistemas de recuperação de falhas para redes de subestação recomendados pela norma. Esses sistemas garantem a entrega do pacote duplicando o tráfego, ou duplicando todos os dispositivos na rede. Com isso, o tráfego na rede aumenta, o que pode sobrecarregar os nós da rede, aumentando inclusive o processamento dos dispositivos finais, que tem baixo poder de processamento, e passam a processar o dobro de pacotes. Além disso, esses sistemas só podem ser usados em topologias específicas e necessitam de implementação nos dispositivos finais [Tan and Luan 2011].

Esse trabalho visa melhorar o desempenho na comunicação dentro de subestações utilizando de forma eficiente os recursos de rede de comunicação, além de autoconfigurar as redes de subestação IEC 61850. A proposta, chamada de SisteMa Autoconfigurável para Redes de Telecomunicações IEC 61850 com arcabouço OpenFlow (SMARTFlow), utiliza o OpenFlow [McKeown et al. 2008] para fazer um melhor controle dos dados em tráfego dentro de uma subestação. Inicialmente, as entradas de fluxos de alta prioridade definidos na norma IEC 61850 são configuradas com base no arquivo de configuração de subestação, definido na norma. Essa configuração é feita pró-ativamente para garantir que não serão inseridos atrasos para as mensagens sensíveis. Para tanto, o SMARTFlow possui um componente que calcula e configura as árvores *multicast* para encaminhamento das mensagens de camada 2, evitando o uso do *broadcast* e sem a necessidade de implementação de protocolos nos dispositivos finais. Foi desenvolvido também, um componente para tratamento de falhas, que recalcula as árvores dinamicamente sempre que houver falhas na rede. Além disso, o SMARTFlow define um modelo para aplicação de prioridade nos diferentes tipos de mensagens na rede, garantindo os requisitos de atraso.

O SMARTFlow foi implementado e testado utilizando o controlador POX e o

emulador de redes Mininet [Lantz et al. 2010] e se mostrou muito eficiente, atendendo a todos os requisitos das mensagens da norma IEC 61850. O sistema proposto reduziu em até 20 vezes o atraso gerado pelas aplicações de controle padrão do POX e apresentou uma carga total até 44% menor do que a de *switches* típicos. Os testes foram realizados considerando diferentes configurações típicas de subestação, de forma a validar a proposta em termos de atraso e carga de controle.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta uma visão geral da norma IEC 61850 e da comunicação *multicast* de camada 2. Os trabalhos relacionados são apresentados na Seção 3. A proposta é apresentada no Capítulo 4 e a Seção 5 apresenta a análise dos resultados experimentais. As conclusões são apresentadas na Seção 6.

2. Redes IEC 61850

A norma IEC 61850 modela os sistemas e a comunicação da rede para a automação no SEP. Seu principal objetivo é garantir interoperabilidade entre dispositivos eletrônicos inteligentes (*Intelligent Electronic Device* (IEDs)) de diferentes fabricantes que permitem, dentre outros, a supervisão, o controle e a proteção em tempo real do SEP.

O modelo de comunicação da IEC 61850 usa três diferentes tipos de mensagem: GOOSE e SV com alta restrição temporal, chegando a 3 ms, e MMS (*Manufacturing Message Specification*) variando de 100ms até 1000ms [IEC 61850]. As mensagens são encaminhadas de duas formas, no modelo publicador/assinante com os endereços *multicast* padronizados pela norma, e no modelo cliente-servidor. A norma define o modelo publicador/assinante para mensagens GOOSE, e o modelo cliente-servidor para mensagens MMS. A SV pode ser enviada nos dois modelos. As mensagens GOOSE e SV são usadas para serviços críticos na subestação, por esse motivo são mapeadas diretamente na camada 2 a fim de prover um tempo de resposta mais rápido. Com isso, estas mensagens são diretamente encapsuladas em camada Ethernet e transmitidas com um endereço de destino MAC *multicast* [McGhee and Goraj 2010].

A norma padroniza uma linguagem de descrição, chamada de Linguagem de Configuração de Subestação (*Substation Configuration Language* - SCL), que norteia a configuração do sistema. Isto significa dizer que são configurados desde os canais de comunicação até a alocação de funções para os sistemas de automação [IEC 61850]. A SCL é baseada em XML (*eXtensible Markup Language*) e seu objetivo principal é padronizar os atributos de configuração, ou seja, criar uma nomenclatura uniformizada, de maneira a permitir configurações de IEDs com maior segurança e confiabilidade. O intuito é manter a interoperabilidade, garantindo a troca de dados entre IEDs independente do fabricante. Dentre os arquivos de configuração que compõe a SCL, este artigo ressalta o SCD, pois é este arquivo que descreve detalhadamente a subestação no que tange à comunicação. O arquivo SCD contém uma seção de configuração de comunicação e uma seção de descrição da subestação. Desta maneira, este arquivo pode conter, por exemplo, aonde está alocado cada nó lógico do sistema, endereços de rede, endereços de grupos *multicast*, etc.

2.1. Comunicação *multicast* de camada 2 em redes IEC 61850

A norma define o uso de grupos *multicast* de camada 2 para mensagens GOOSE e SV. *Switches*, por padrão, quando recebem um pacote endereçado a um destino *multicast* de

camada 2, enviam este pacote por todas as portas. É esse método que garante que o pacote vai chegar ao destino qualquer que seja o destinatário². Porém, esse método consome banda no enlace, aumenta o atraso nos *switches* e introduz uma sobrecarga significativa nos IEDs. Para contornar este problema, os fluxos de dados *multicast* devem ser restritos apenas ao grupo em questão, ou seja, à árvore *multicast* [McGhee and Goraj 2010].

As árvores *multicast* podem ser configuradas de forma estática ou dinâmica. Na configuração estática, o endereço *multicast* é mapeado para a porta, ou mais de uma se for o caso, por onde os pacotes daquele grupo específico devem ser encaminhados. Se o estado da rede muda, não existe uma atualização automática na tabela de encaminhamento. Qualquer configuração ou alteração deve ser feita manualmente, por esse motivo não é muito utilizada. Na configuração dinâmica, o gerenciamento do *multicast* é feito com o uso de protocolos, que se encarregam do tráfego *multicast* encaminhando-o para os dispositivos que manifestaram interesse em recebê-lo. A árvore *multicast* é construída dinamicamente, e, em caso de atualização na rede, o algoritmo deve recalcular a árvore. Nesse caso, os IEDs deveriam implementar protocolos para serem capazes de entrar ou sair de um grupo. Atualmente, o protocolo *multicast* de camada 2 usado em redes IEC 61850 é o GMRP. Seu princípio básico de funcionamento é baseado em mensagens intituladas *join* e *leave*, as quais o *host* deve ser capaz de enviar quando quiser entrar ou sair de um grupo. O *switch* registra a porta pela qual recebeu a mensagem *join* e associa essa porta ao grupo *multicast* da mensagem, montando a sua tabela de encaminhamento. Em seguida, o *switch* envia a mensagem *join* via *broadcast* para toda a rede, garantindo que o novo membro do grupo *multicast* passe a ser conhecido por toda a rede. Todos os *switches* que suportam GMRP podem receber essa informação de outros *switches* para atualizar seu registro local. Com a tabela configurada e atualizada, quando o publicador envia mensagens *multicast*, o *switch* envia essas mensagens apenas para as portas necessárias para que a mensagem chegue a todos os membros do grupo [Yong-hui et al. 2011].

3. Trabalhos Relacionados

Existem trabalhos que mostram que o uso de *multicast* em redes de subestação simplifica o controle do tráfego e diminui os atrasos [Ingram et al. 2011, Sivanthi and Goerlitz 2013, Moore et al. 2010, McGhee and Goraj 2010], além de reduzir o volume de dados recebidos por dispositivos [XiCai et al. 2011].

Yong-hui et al. mostram as vantagens do uso do GMRP em redes de subestação. Os autores exploram as vantagens do GMRP com aplicações em laboratório e aplicações práticas. Nesse mesmo contexto, XiCai et al. apresentam um exemplo do uso do GMRP em subestações Chinesas para reduzir o volume de dados recebidos por dispositivos [XiCai et al. 2011]. Muitas vezes, o GMRP é usado em conjunto com as VLANs (*Virtual Local Area Network*) para restringir o tráfego [Ingram et al. 2011]. Ingram et al. combinam o uso de VLANs e filtro *multicast* para separar o tráfego por aplicação e por grupos de interesse, de forma que o tráfego *multicast* fique restrito a ser distribuído apenas dentro de determinada VLAN [Ingram et al. 2011]. Sivanthi e Goerlitz propõem uma abordagem sistemática para melhorar o uso de filtros *multicast* e VLANs agrupando caminhos comuns [Sivanthi and Goerlitz 2013]. Os autores apresentam um algoritmo que agrupa destinos que seguem o mesmo caminho na rede, ou seja, se dois

²Nesse artigo, entende-se por *switch* típico o comportamento padrão de *switches*.

grupos *multicast* têm a mesma árvore, o algoritmo coloca os dois no mesmo grupo *multicast* [Sivanthi and Goerlitz 2013]. As propostas de uso do GMRP em redes de subestação, demandam que os fabricantes de IEDs implementem o protocolo em seus dispositivos [Yong-hui et al. 2011]. Contudo, os IEDs, até o momento, não possuem a capacidade de implementar o protocolo e enviar mensagens *join* e *leave* para a rede. Com isso, parte da configuração tem que ser feita manualmente, na tabela *multicast* estática dos *switches*. Assim, a configuração da rede se torna trabalhosa e a maioria dos fabricantes recomenda a utilização de VLANs para limitar os domínios de *broadcast* [Ingram et al. 2011]. Portanto, o *multicast* acaba não sendo utilizado na prática, o que reduz o desempenho da rede [McGhee and Goraj 2010].

A proposta deste artigo dinamicamente cria árvores *multicast*, sem a necessidade de implementação nos IEDs e configura os fluxos *unicast* e *multicast* no contexto de sistemas de automação de subestação usando o OpenFlow. O trabalho leva em consideração os requisitos temporais rígidos da norma IEC 61850 e a confiabilidade requerida em redes de subestação. Além disso, reconfigura dinamicamente toda a rede em caso de falha ou de mudança na rede.

Com relação ao uso de OpenFlow em *Smart Grids*, Sydney et al. propõem o uso de OpenFlow para prover recursos de MPLS em redes de longa distância [Sydney et al. 2014]. Cahn et al. propõem o SDECN (*Software-Defined Energy Communication Network*), que usa redes definidas por software (SDN) como solução para as *smart grids* [Cahn et al. 2013]. Os autores sugerem o uso do openflow em redes de subestação e fazem uma implementação simples com base no controlador Ryu. Poucos detalhes são apresentados sobre algoritmos e lógicas de controle e os cenários de teste são muito restritos.

Existem, ainda, alguns trabalhos que utilizam o OpenFlow para gerenciamento de rede e encaminhamento *multicast* [Marcondes et al. 2012, Silva et al. 2012], porém fora do contexto de subestações.

4. A proposta SMARTFlow

Para aumentar o desempenho das redes de subestação, é proposto o SMARTFlow, que é um sistema autoconfigurável para redes de telecomunicações IEC 61850 baseado no arcabouço OpenFlow. Seus objetivos principais são o desenvolvimento de um encaminhamento apropriado de mensagens IEC 61850 e o controle eficiente de redes de dados de subestações elétricas. Além disso, com o SMARTFlow, é possível fazer a autoconfiguração dos grupos *multicast* de forma automática, toda vez que for inserido ou retirado um IED da rede, sem a necessidade de implementação no IED. Como a proposta é facilitar o planejamento e a configuração da rede, é possível configurar automaticamente a rede de telecomunicações com a solução adequada aos requisitos da rede. Dessa forma, os principais módulos do SMARTFlow são:

- autoconfiguração inicial da rede de telecomunicações com base nos dados obtidos pelos arquivos de configuração da subestação providos pela norma IEC 61850;
- criação automática de árvores *multicast* de camada 2 para o envio de mensagens GOOSE e SV;
- recuperação de falhas através do recálculo e atualização automática das rotas proativamente sempre que um dos enlaces da rede falhar;

4.1. O IEC 61850 e o OpenFlow

Nesse trabalho, o OpenFlow foi escolhido como arcabouço, pois possibilita um controle mais flexível e orientado às necessidades de cada sistema de comunicação, como é o caso das *smart grids* e da Norma IEC 61850. Com isso, torna-se possível experimentar novos métodos e novos algoritmos que garantam uma boa solução e aumentem o desempenho das redes. De fato, essa tecnologia traz ganhos por permitir a implementação de todas as recomendações da norma e de recursos para otimizar a rede. Além disso, por existir um controlador centralizado, as redes OpenFlow oferecem flexibilidade de programação e uma visão unificada da rede, facilitando a implementação de novos algoritmos, a configuração e a gestão da rede. Além disso, a manutenção preventiva da rede é realizada de uma maneira simples, uma vez que a migração de fluxos é simples em redes OpenFlow. Como o uso do arcabouço OpenFlow simplifica a virtualização da rede, diferentes fabricantes podem implementar políticas para controlar a sua rede no mesmo *switch* sem interromper ou interferir com outros serviços.

4.2. Arquitetura do SMARTFlow

A arquitetura detalhada do SMARTFlow é ilustrada na Figura 1. O SMARTFlow é executado sobre um elemento central da rede, chamado controlador, que se comunica com todos os *switches* para configurar as tabelas de fluxo via um canal seguro. O SMARTFlow pode ser implementado em qualquer controlador OpenFlow.

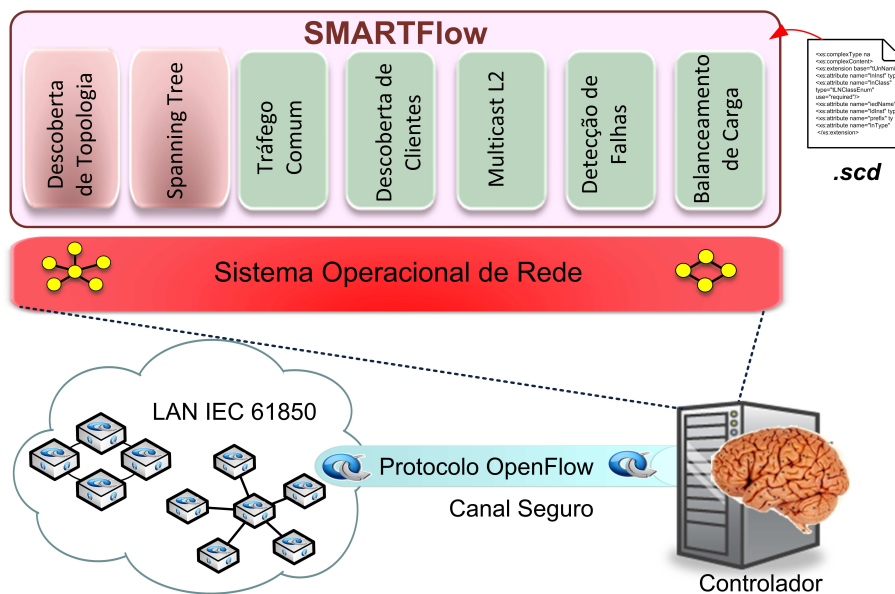


Figura 1. Estrutura do SMARTFlow, o qual é composto por um conjunto de aplicações de controle para redes de subestações baseadas em IEC 61850.

No SMARTFlow, o plano de controle da rede de telecomunicações é responsável por monitorar e configurar a rede automaticamente a partir de parâmetros do arquivo SCD da Norma. Os algoritmos de controle do SMARTFlow [Lopes 2013], usam como base informações contidas nesse arquivo, que pode conter, dentre outras informações, a que grupo *multicast* cada IED pertence e os endereços de rede do mesmo. Dessa forma, torna-se possível fazer um mapeamento dos grupos *multicast* existentes na rede para o algoritmo que calcula as árvores, gerando uma lista de grupos *multicast*.

Propõe-se o uso de sete componentes, os quais todos foram desenvolvidos para o SMARTFlow, com exceção do Descoberta de Topologia e do Spanning Tree, que são componentes geralmente encontrados nos controladores OpenFlow:

Descoberta de Topologia: Este componente é essencial para o funcionamento dos outros componentes do SMARTFlow. O componente Descoberta de Topologia instrui cada um dos *switches* da rede a enviar mensagens *Link Layer Discovery Protocol* (LLDP) para seus vizinhos, de modo que o controlador possa descobrir a topologia da rede. Quando um *switch* recebe um pacote LLDP, ele encaminha o cabeçalho do pacote para o controlador, já que não existe regra de encaminhamento para esse fluxo. Com isso, o controlador pode inferir a conectividade dos enlaces combinando os dados de cada pacote LLDP recebido.

Spanning Tree: Já conhecendo a visão geral da topologia da rede, pelo componente Descoberta de Topologia, pode-se, então, construir a *spanning tree*. O objetivo do componente *Spanning Tree* é calcular a árvore de cobertura da rede e, de acordo com os dados obtidos, desativar a primitiva de inundação em alguns enlaces específicos. Isso faz com que topologias que formem ciclos evitem *loops* infinitos no envio de mensagens *broadcast*.

Tráfego Comum: Este componente encaminha, sob demanda, o tráfego unicast de baixa prioridade das redes baseadas em IEC 61850, como por exemplo, *mensagens MMS*. O componente Tráfego Comum é uma versão do componente do Openflow que funciona como um *switch* de aprendizagem de forma reativa. Isso significa dizer que, sempre que um novo fluxo chegar a um *switch*, este dispositivo encaminha o cabeçalho do primeiro pacote para o controlador, o qual aciona o módulo Tráfego Comum para calcular e definir as entradas de fluxo correspondentes com base no estado atual da rede.

Descoberta de Clientes: Este componente captura os pacotes da rede e faz o mapeamento dos endereços MAC de todos os IEDS na rede automaticamente, armazenando em um dicionário todos os endereços MAC dos IEDs, a qual portas e *switches* estão ligados. Esse dicionário é intitulado *mac_map*. Cada vez que um IED é acrescentado ou retirado da rede, esse componente atualiza esse dicionário. Os endereços MAC dos IEDs e servidores, na prática, podem ser obtidos do arquivo SCD da norma IEC 61850. Contudo, essa configuração inicial não é suficiente para detectar erros de ligação, ou ainda, mudanças na topologia geradas durante o funcionamento da rede. Assim, optou-se pelo uso de uma solução mais genérica. O componente proposto monitora os eventos da rede e mantém atualizado o dicionário que correlaciona o MAC e a porta de saída de cada *switch*.

Balanceamento de carga: Esse componente observa o estado da rede e distribui a carga dos fluxos *unicast* menos prioritários entre os enlaces, através do uso de migrações ao vivo. Com isso, sempre que se observa que um enlace está próximo de um certo limiar de uso, alguns fluxos são migrados para outros enlaces menos sobrecarregados. Essa migração é feita criando-se uma nova rota a partir do destino até a origem. A rota original é apagada apenas quando a nova rota já está pronta para uso, garantindo, assim, que não serão perdidos pacotes. Isso ajuda a melhorar o desempenho da rede e a minimizar a latência de resposta, o que é essencial para redes de comunicação que auxiliam mecanismos de proteção do SEP.

Multicast L2: Este componente implementa o Algoritmo 1 para calcular as árvores de distribuição para os grupos *multicast* de camada dois, necessários para as mensagens GOOSE e SV. Utilizando como entrada o arquivo SCD, esse componente identifica os grupos *multicast* e, em seguida, com base nos dados das aplicações Descoberta de Topologia e Descoberta de Clientes, calcula as árvores *multicast* para cada grupo na rede IEC 61850 da subestação e configura pró-ativamente os respectivos *switches*. Essa configuração pró-ativa é importante porque os grupos *multicast* são utilizados por mensagens de alta sensibilidade a atrasos no IEC 61850. A configuração pró-ativa impede que a entrega das mensagens seja atrasada pela consulta reativa ao controlador, o que seria o comportamento padrão do OpenFlow.

O Algoritmo 1, baseado nos dados do componente Descoberta de Topologia, é capaz de ter uma visão completa da rede, podendo armazenar em uma lista, todos os enlaces e nós da rede. A lista de enlaces (E), a lista de nós (N), a lista de grupos *multicast* (*gruposmulticastgerados*) e o dicionário *mac_map* são as entradas desse algoritmo. A saída é o caminho completo da árvore *multicast*, intitulado *caminho_completo*, que o próprio algoritmo usa para configurar os fluxos em cada *switch* da rede. A lista de grupos *multicast* é percorrida para criação e configuração de cada

Algoritmo 1: Algoritmo de cálculo de árvores *multicast* e estabelecimento de rotas

```

Input:  $E, N, mac\_map, gruposmulticastgerados$ 
Output:  $caminhos\_completos$ 
1 for grupo in gruposmulticastgerados do
2    $end\_GM = descobre\_end(grupo)$ 
3    $raiz = descobre\_raiz(mac\_map, grupo)$ 
4    $sws\_destino = descobre\_dst(mac\_map, grupo)$ 
5    $melhores\_rotas = SPF\_multicast(raiz, N, E)$ 
6   for rota in melhores_rotas do
7     if rota[dst] in sws_destino then
8        $caminhos\_arvore.append(rota)$ 
9     end
10  end
11   $arvore\_multicast = remove\_redundancias(caminhos\_arvore)$ 
12   $caminhos\_completos =$ 
    $acrescenta\_portas(mac\_map, arvore\_multicast)$ 
    $instala\_fluxos(caminhos\_completos, end\_GM, raiz)$ 
    $return\ caminhos\_completos$ 
13 end

```

árvore *multicast*, como mostrado na linha 1. A função *descobre_end*, na linha 2, descobre qual o endereço MAC do grupo *multicast*. A função *descobre_raiz*, na linha 3, com base no *map_mac* descobre qual o endereço MAC do nó raiz do grupo *multicast* e em qual *switch* está conectado. É importante observar que serão considerados como raiz todos os nós que possam emitir mensagens para o grupo *multicast*. A função *descobre_dst*, na linha 4, retorna uma lista com os *switches* que estão conectados a pelo menos um dos membros do grupo, com as respectivas portas. Com os nós, os enlaces e a raiz do grupo

multicast, a função *SPF_multicast* calcula todas as rotas mais curtas da raiz para cada outro nó restante da rede, possíveis nós receptores do grupo. Com isso, é criada uma lista de caminhos mais curtos, intitulada *melhores_rotas*. Conforme descrito na linha 6, as rotas da lista *melhores_rotas* que tiverem como destino os nós receptores do grupo são armazenadas na lista *caminhos_arvore* que, ao final, contém os melhores caminhos do IED publicador até os IEDs receptores do grupo *multicast*. A lista *caminhos_arvore* é processada para remoção de redundâncias com a função *remove_redundancias*. Essa função gera a lista *arvore_multicast*, contendo o caminho por qual os pacotes deverão passar para alcançar todos os destinos daquele grupo, conforme linha 11. Para que as tabelas dos *switches* possam ser configuradas, além do caminho é necessário o acréscimo das portas por qual esse caminho está conectado. Essa tarefa é realizada pela função *acrescenta_portas*, que gera como saída a lista intitulada *caminhos_completos*, contendo uma lista para cada *switch* com o seu número de identificação e as portas por onde os pacotes serão encaminhados, ou seja, a tabela de fluxos que deverá ser configurada. Por fim, o controlador pode configurar as tabelas de fluxos nos *switches* pertencentes à lista *caminhos_completos* criando assim a *árvore_multicast* do grupo em questão.

Cabe observar que o algoritmo proposto não busca a melhor árvore de cobertura, mas a entrega mais rápida dos pacotes, tendo como métrica o número de saltos. Devido às fortes restrições de atraso das mensagens GOOSE e SV, a proposta prioriza a velocidade da entrega.

Detecção de Falhas: Este componente é chamado sempre que ocorre mudanças na topologia da rede. Desta forma, sempre que um enlace cai, o componente Detecção de Falhas chama o componente *Multicast L2*, o qual recalcula as árvores³. O componente e seu algoritmo são detalhados no Algoritmo 2. É importante notar que,

Algoritmo 2: Algoritmo de detecção de falhas e restauração da rede

Input: *evento, E, N, mac_map, grupos_multicast_gerados, caminhos_completos*

Output: *caminhos_completos*

- 1 *enlaces_afetados = busca_falhas(evento, E, N)*
 - 2 *grupos_afetados, grupos_nao_afetados = busca_grupos(enlaces_afetados, caminhos_completos, grupos_multicast_gerados)*
 - 3 *novos_grupos = multi_tree(E, N, mac_map, grupos_afetados)*
 - 4 *caminhos_completos = novos_grupos + grupos_nao_afetados*
 - 5 *return caminhos_completos*
-

se a versão do OpenFlow suportar o *Fast Failover*, então existem pequenas alterações nos Algoritmos 1 e 2. O Algoritmo 1, após calcular a árvore de distribuição para um par (endereço MAC multicast, raiz), irá aumentar os custos de todos os enlaces utilizados nessa árvore de cobertura e irá buscar uma nova árvore de distribuição na rede. Em seguida, essa nova árvore é adicionada como caminho para casos de falha na tabela de grupo dos *switches* OpenFlow. É importante observar que o aumento do custo garante que uma segunda árvore será encontrada, mesmo que não existam caminhos totalmente disjuntos.

³Cabe observar que o comportamento desse componente varia de acordo com a versão do OpenFlow em uso, de acordo com a disponibilidade ou não da tabela de grupo e do mecanismo de *fast failover*.

As modificações no Algoritmo 2 ocorrem apenas para apagar a rota original que está com uma falha, de forma que a segunda opção se torne a rota principal. O Algoritmo, naturalmente, já buscará uma terceira opção de árvore de distribuição, para prevenir novas falhas na rede. Ressalta-se que a rede deverá prover mais de um caminho, caso a rede só tenha um caminho possível, esse componente não se aplica. Maiores detalhes sobre os algoritmos podem ser encontrados em [Lopes 2013].

5. Experimentos com ferramentas de emulação e Análise dos Resultados

Os componentes do SMARTFlow foram desenvolvidos em Python e implementados no controlador POX 0.1.0 na versão 1.0.0 do OpenFlow. Os experimentos foram emulados usando o Mininet [Lantz et al. 2010] versão 2. O Mininet é uma plataforma flexível para emulação de redes OpenFlow que provê um ambiente de experimentação bem próximo do real. Foi criado um módulo no Mininet que constrói topologias LAN de subestações, todas distribuindo os IED uniformemente na rede. Este módulo também contém a classe que chama os componentes do SMARTFlow para controlar a rede. Para esses experimentos, focou-se na entrega do tráfego de mensagens GOOSE, que tem grande restrição de atrasos nas redes das subestações. Para tanto, foi desenvolvido um gerador de mensagens GOOSE na ferramenta Scapy para simular o tráfego dos IEDs. A duração dos experimentos foi de 100 segundos para cada rodada, incluindo, neste tempo, a estabilização da rede, a configuração dos fluxos, a troca de mensagens e o tempo da simulação. Foram variados parâmetros como quantidade de IEDs na rede, quantidades de *switches*, tipo de topologia e quantidade de IED por grupo *multicast*.

O ambiente foi emulado por meio de virtualização em um notebook com processador Intel Core i5-3210M, e 4GB de memória RAM. Os testes foram realizados com três instâncias de máquinas virtuais simultâneas, cada uma delas com uma CPU virtual, 1024 MB de memória e executando o sistema operacional Ubuntu 11.10. Todos os resultados apresentam um intervalo de confiança de 95%. Para os experimentos, levou-se em consideração os cenários típicos de subestação descritos na parte 1 da norma IEC 61850 [IEC 61850] onde é descrito o tamanho da subestação e sua importância no sistema. Em todos os casos, a quantidade de IEDs depende muito do projeto e funções que serão utilizadas na subestação. Assumindo uma quantidade de 3 até 12 IEDs pode-se assumir que os testes englobam, senão todos, a maior parte dos cenários típicos em subestações. Além disso, para todos os experimentos foram testadas topologias em anel e estrela pois são as topologias encontradas em subestação. Os gráficos da Figura 2 apresentam o resultado para topologia em anel, já que esta apresentou uma carga de controle um pouco mais alta do que a topologia em estrela. O cenário é composto por cinco grupos *multicast* distintos e nove IEDs, distribuídos uniformemente entre os *switches*. A quantidade de *switches* variou de um a nove. Foram estimulados dez eventos na rede, gerando tráfego GOOSE. A ideia consiste em criar cenários que representam desde pequenas até grandes subestações.

Na Figura 2(a), nota-se que o atraso na rede controlada pelo SMARTFlow não passou de 1,5 ms. Isso mostra que o SMARTFlow atende bem aos requisitos rígidos de tempo da norma, que determinam atraso máximo de 3 ms para mensagens GOOSE, mesmo em redes de subestação com grande número de switches. Além disso, verifica-se que, na rede controlada pelos componentes reativos do controlador OpenFlow, o atraso é muito superior, chegando a ser até 20 vezes maior que o atraso do SMARTFlow. Esse

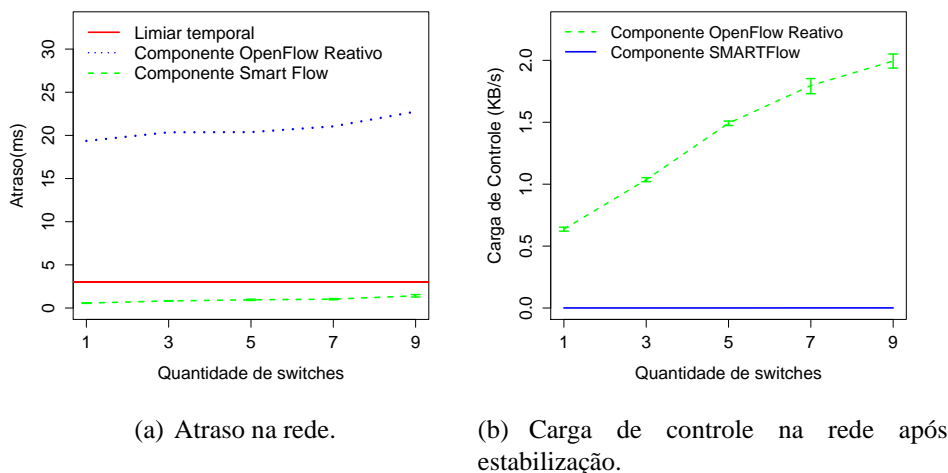


Figura 2. Comparação ao se utilizar o SMARTFlow e o OpenFlow Nativo em uma topologia em anel.

comportamento era esperado pela característica reativa desses componentes, no qual cada pacote de um novo fluxo que chega ao *switch* é enviado ao controlador. Quando o controlador identifica que é um pacote *multicast*, configura uma entrada de fluxo para enviar o pacote para todas as portas de saída, já que o comportamento padrão dos *switches* é tratar o *multicast* como *broadcast*. Todo esse processo, naturalmente, sobrecarrega a rede e aumenta o atraso dos pacotes. Com isso, conclui-se que o componente padrão de encaminhamento do controlador OpenFlow não é adequado para o controle de mensagens GOOSE, pois não garante os requisitos mínimos de atraso. A Figura 2(b), analisa a carga de controle gerada na rede, assumindo um sistema estabilizado, ou seja, durante o comportamento padrão da subestação. Para isso, calculou-se toda a carga de pacotes de controle na rede, como pacotes LLDP, *packet_in*, *packet_out*, etc. Em uma rede estabilizada, o SMARTFlow apresenta uma carga de controle muito baixa, com valores muito próximos de 0, pois a carga de controle da proposta se concentra, principalmente, na inicialização da rede. Isso é importante, pois a rede não é sobrecarregada durante o seu funcionamento normal, evitando o aumento dos atrasos para mensagens GOOSE e SV. Os valores mais altos do OpenFlow se devem à troca de mensagens entre controlador e *switches* durante todo o tempo.

Por fim, a Figura 3, apresenta os resultados da avaliação da carga total na rede, incluindo a carga de controle no caso dos componentes OpenFlow. Além dos sistemas anteriores, OpenFlow Nativo e SMARTFlow, acrescenta-se um sistema baseado em *switches* típicos. Para emular os *switches* típicos, criou-se um componente apenas para instalar proativamente regras que tratam o tráfego *multicast* como *broadcast* configurando os fluxos proativamente. Desta forma, simula-se um *switch* comum, onde os fluxos já se encontram definidos encaminhando o pacote por todas as portas. O cenário considerou uma LAN com 5 *switches* e uma quantidade de IEDs variando entre 3 e 12. Emulou-se dois tipos de topologia, anel e estrela, cinco grupos *multicast* e dez eventos na rede elétrica, em momentos aleatórios, que refletem em um aumento do tráfego GOOSE. Observa-se, na Figura 3, que a carga total de dados no OpenFlow Nativo é um pouco mais alta do

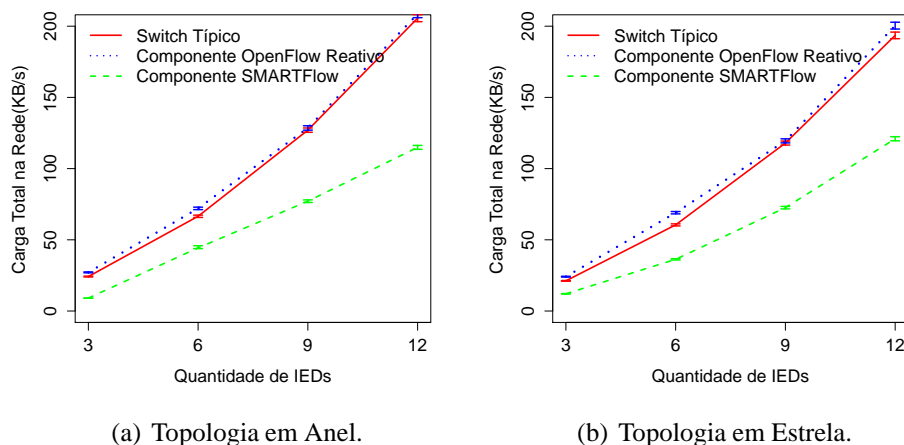


Figura 3. Carga total de dados na rede em função do aumento do número de IEDs.

que a carga do *switch* típico. Isso acontece pois, quando o componente reativo do OpenFlow emula *switches* típicos é acrescida à carga da inundação na rede a carga de controle para setar o fluxo. Ressalta-se que, a carga de controle para este cenário é muito pequena quando comparada a quantidade total de dados na rede e, com isso, o comportamento das duas curvas é muito próximo. Além disso, verifica-se que o SMARTFlow diminui a carga total na rede em até 44% para 12 IEDs na topologia em Anel. Isso acontece, pois o SMARTFlow calcula uma árvore *multicast* para encaminhar os pacotes evitando a inundação natural de *switches* típicos. Quando o pacote chega ao *switch*, ao invés de ser encaminhado por todas as portas, é enviado apenas para a porta relativa a árvore *multicast*.

O tempo para recuperação de falha, mesmo sem a implementação do *fast failover* ficou em torno de 8ms. Esse experimento e outros resultados como os tempos para configuração da rede, descoberta da rede, etc, são encontrados em [Lopes 2013].

Realizou-se também uma análise qualitativa comparando as características das soluções *multicast* de camada 2 com as características da solução SMARTFlow. Os resul-

Tabela 1. Comparação entre as atuais soluções *multicast* de camada 2 e a proposta SMARTFlow

Características	<i>Multicast</i> típico	<i>Multicast</i> Estático	GMRP	SF 1.0	SF
Complexidade de configuração	Baixa	Alta	Média	Baixa	Baixa
Dependência de mensagens <i>Join</i> e <i>leave</i>	Não	Não	Sim	Não	Não
Consumo de Banda pelo tráfego de dados	Alto	Baixo	Baixo	Baixo	Baixo
Carga de Controle	Baixa	Baixa	Alta	Baixa	Baixa
Inundação da Rede	Sim	Não	Não	Não	Não
Controle aprimorado do tráfego	Não	Não	Não	Sim	Sim
Simplicidade e Flexibilidade	Baixa	Baixa	Baixa	Alta	Alta
Convergência Rápida	Sim	Não	Não	Sim	Sim
Tempos de atraso na rede	Alto	Baixo	Baixo	Baixo	Baixo

tados são mostrados na Tabela 1, onde SF 1.0 refere-se ao SMARTFlow implementado no arcabouço do OpenFlow versão 1.0, SF refere-se a implementação em versões superiores, *Multicast* Típico refere-se ao comportamento padrão de *switches* e *Multicast* Estático a configuração da árvore de forma manual. Uma vantagem do SMARTFlow sobre o GMRP é a ausência de mensagens de controle entre os *switches*. No GMRP, os *switches* são responsáveis por encaminhar pacotes e trocar mensagens de controle para montar a árvore *multicast*. Isto cria uma sobrecarga extra em termos de consumo de banda, pois o controle de mensagens, como *join* e *leave* ou atualizações de árvores, são enviadas através dos mesmos enlaces que o tráfego de dados. Além disso, no SMARTFlow, os IEDs são automaticamente incluídos na árvore *multicast* com base no arquivo SCD, de modo que não é necessário que os IED implementem protocolos *multicast* cliente ou que mensagens de atualização da árvore sejam enviadas frequentemente. Mudanças na topologia de rede são automaticamente detectadas no SMARTFlow, que desencadeia a reconfiguração das árvores *multicast*.

6. Conclusões

A norma IEC 61850 tem ganhado cada vez mais espaço, sendo implantada em novas subestações trazendo inúmeros benefícios como redução de custos e de erros humanos, automação, implementação de novas capacidade, dentre outros. Contudo, mesmo com a inovação, ainda existem problemas a serem resolvidos. Esse trabalho identificou e abordou alguns desses problemas, assim como propôs, desenvolveu e avaliou um serviço de gerenciamento e encaminhamento autoconfigurável para redes IEC 61850 baseada em uma técnica promissora que tem possibilitado um controle mais flexível, o OpenFlow. A proposta, chamada SMARTFlow, diminuiu a carga total da rede gerada pelo OpenFlow Nativo e pelo *switch* de camada 2 típico em até 44% no cenário apresentado. Os testes mostraram, também, que o atraso na rede controlada pelo OpenFlow em sua forma habitual chegou a ser até 20 vezes maior do que a rede controlada pelo SmartFlow, passando de 20ms. Contudo, o atraso na rede controlada pela proposta SMARTFlow não ultrapassou 1,5ms, que é metade do valor mais rígido de tempo estabelecido pela norma (3ms). Com isso, esse trabalho verificou que o SMARTFlow é capaz de cumprir os requisitos impostos pela norma IEC 61850 ao usar as aplicações desenvolvidas como uma prova de conceito. Além disso, o uso de *multicast* com um controle centralizado e com dados oriundos do arquivo SCD permitem que sejam usados IEDs mais simples e também reduz o tráfego de controle, o tempo de convergência dos algoritmos de controle e o atraso de entrega de dados, quando comparado com os protocolos habituais.

Como trabalhos futuros, pretende-se implantar a proposta em uma rede real OpenFlow, e em uma rede tradicional para uma análise mais profunda. Uma outra questão é o estudo, desenvolvimento e implementação do SMARTFlow nos IEDs que possuem *switches* embarcados para a construção de topologias em anel. Portanto, seria necessária a implementação do OpenFlow nesses *switches* embarcados e a realização de testes de desempenho utilizando o SMARTFlow. Pode-se também investigar, implementar e avaliar a proposta em um contexto entre subestações, podendo inclusive estender a pesquisa para às Smart Grids como um todo. Além disso, pretende-se validar a recuperação de falhas do SMARTFlow implantando também o *fast failover*.

Referências

- [Cahn et al. 2013] Cahn, A., Hoyos, J., Hulse, M., and Keller, E. (2013). Software-defined energy communication networks: From substation automation to future smart grids. In *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*, pages 558–563.

- [Gungor et al. 2011] Gungor, V., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C., and Hancke, G. (2011). Smart grid technologies: Communication technologies and standards. *IEEE Transactions on Industrial Informatics*, 7(4):529–539.
- [IEC 61850] IEC 61850. Communication networks and systems for power utility automation. Technical report, International Electrotechnical Commission (IEC).
- [Ingram et al. 2011] Ingram, D., Schaub, P., and Campbell, D. (2011). Multicast traffic filtering for sampled value process bus networks. In *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, pages 4710–4715.
- [Lantz et al. 2010] Lantz, B., Heller, B., and McKeown, N. (2010). A network in a laptop. In *Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks - Hotnets '10*, pages 1–6. ACM Press.
- [Lopes 2013] Lopes, Y. (2013). SMARTFlow: Sistema Autoconfigurável para Redes de Telecomunicações IEC 61850 com arcabouço OpenFlow. Mestrado em Engenharia de Telecomunicações, Universidade Federal Fluminense, UFF.
- [Lopes et al. 2012] Lopes, Y., Frazão, R. H., Molano, D. A., dos Santos, M. A., Calhau, F. G. a., Bastos, C. A. M., Martins, J. S. B., and Fernandes, N. C. (2012). Smart Grid e IEC 61850: Novos Desafios em Redes e Telecomunicações para o Sistema Elétrico. In *Minicursos do XXX Simpósio Brasileiro de Telecomunicações*, pages 1–44. (SBrT), Sociedade Brasileira de Telecomunicações, 1 edition.
- [Marcondes et al. 2012] Marcondes, C., Santos, T., Godoy, A., Viel, C., and Teixeira, C. (2012). CastFlow: Clean-slate multicast approach using in-advance path processing in programmable networks. In *2012 IEEE Symposium on Computers and Communications (ISCC)*, pages 94–101.
- [McGhee and Goraj 2010] McGhee, J. and Goraj, M. (2010). Smart High Voltage Substation Based on IEC 61850 Process Bus and IEEE 1588 Time Synchronization. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 489–494.
- [McKeown et al. 2008] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: enabling innovation in campus networks. *SIGCOMM Computer Communication Review*, 38(2):69–74.
- [Moore et al. 2010] Moore, R., Midence, R., and Goraj, M. (2010). Practical experience with IEEE 1588 high Precision Time Synchronization in electrical substation based on IEC 61850 Process Bus. In *Power and Energy Society General Meeting, 2010 IEEE*, pages 1–4.
- [Silva et al. 2012] Silva, F. d. O., Goncalves, M. A., de Souza Pereira, J. H., Pasquini, R., Rosa, P. F., and Kofuji, S. T. (2012). On the analysis of multicast traffic over the entity title architecture. In *Networks (ICON), 2012 18th IEEE International Conference on*, pages 30–35.
- [Sivanthi and Goerlitz 2013] Sivanthi, T. and Goerlitz, O. (2013). Systematic real-time traffic segmentation in substation automation systems. In *Emerging Technologies Factory Automation (ETFA), 2013 IEEE 18th Conference on*, pages 1–4.
- [Sydney et al. 2014] Sydney, A., Ochs, D. S., Scoglio, C., Gruenbacher, D., and Miller, R. (2014). Using GENI for experimental evaluation of Software Defined Networking in smart grids. In *Computer Networks*.
- [Tan and Luan 2011] Tan, J. C. and Luan, W. (2011). IEC 61850 based substation automation system architecture design. In *Power and Energy Society General Meeting, 2011 IEEE*, pages 1–6.
- [XiCai et al. 2011] XiCai, Z., ShuChao, W., Lei, X., and YaDong, F. (2011). Practice and trend of DSAS in China. In *Advanced Power System Automation and Protection (APAP), 2011 International Conference on*, volume 3, pages 1762–1766.
- [Yong-hui et al. 2011] Yong-hui, Y., Lei-tao, W., and Yong-jian, T. (2011). Research of network transmission of process bus based upon IEC 61850. In *Advanced Power System Automation and Protection (APAP), 2011 International Conference on*, volume 2, pages 1578–1582.