Métodos de Resiliência em Redes Definidas por Software

Rodrigo S. Amancio, Rodrigo S. Couto,
Marcelo G. Rubinstein

***Resumo*— As Rede Definidas por Software facilitam o gerenciamento e a configuração dinâmica da rede; porém, a capacidade de responder prontamente a falhas em um curto período de tempo também é essencial. Neste artigo, são analisados métodos de resiliência que permitem a diminuição do *jitter* e da perda de pacotes em caso de falhas, além de melhorar o tempo de recuperação. Com a implantação desses métodos é possível diminuir em até oito vezes a perda de pacotes, além de ter uma diminuição em torno de 0,8 ms do *jitter* em comparação a cenários sem os métodos de resiliência implementados.**

***Palavras-Chave*— Software, Resiliência, Falhas, Jitter, Perda de pacotes.**

***Abstract*— Software Defined Networking facilitates the management and the dynamic configuration of the network, however, the ability to quickly respond to failures in a short period of time is also essential. In this article, we analyze methods of resilience that allow the reduction of jitter and the packet loss in case of failures, moreover improving the recovery time. With the aforementioned mechanisms it is possible to increase network reliability by reducing packet loss by up to eight times, in addition to reducing about 0.8 ms of the jitter in comparison to scenarios without the implemented resilience methods.**

***Keywords*— Software, Resilience, Failures, Jitter, Packet loss.**

I. Introdução

A capacidade de responder prontamente a falhas em um curto período de tempo é essencial em redes de grande porte, como *datacenters* e provedores de serviço de Internet. Tendo em vista um cenário altamente competitivo, a busca por melhor desempenho nos serviços oferecidos e a necessidade de redes cada vez mais estáveis e resilientes, arquiteturas que utilizam SDN (*Software Defined Networking*) [1] têm se tornado as principais opções de implementação.

Um dos benefícios da entidade controladora central introduzida em SDN é sua possibilidade de monitorar a rede quanto ao desempenho e funcionalidade, assim como sua capacidade de reprogramar os fluxos quando necessário. O controlador pode monitorar a integridade geral da rede e observar as características de fluxos, como a vazão, o atraso e a perda de pacotes. A tarefa mais básica do controlador é configurar como é realizado o encaminhamento de tráfego dos fluxos da rede [2]. Portanto, quando um enlace é interrompido, o controlador precisa reconfigurar a rede para restaurar ou manter a conectividade [3]. Todavia, o tempo de restauração de um caminho quebrado, além do tempo de detecção da quebra, inclui o atraso introduzido pelo tempo de propagação da notificação do evento ao controlador e o atraso da reconfiguração da rede. Todas essas etapas, consequentemente, atrasam a execução do método de resiliência da rede [3].

Este artigo apresenta uma análise de desempenho de métodos de resiliência em redes SDN. Esses métodos utilizam funcionalidades que combinam caminhos primários e secundários pré-configurados. A implementação da detecção de falha por enlace é realizada por meio da Detecção de Encaminhamento Bidirecional (*Bidirectional Forwarding Detection* - BFD) [4], um protocolo que detecta falhas através da análise da perda de pacotes e fluxos frequentes de mensagens de controle. Além disso, o mecanismo de agregação de enlaces, que utiliza o Protocolo de Controle de Agregação de Enlace (*Link Aggregation Control Protocol* - LACP) [5], também é implementado em redes resilientes e o seu desempenho é analisado quando implementado na borda da rede.

O trabalho está organizado da seguinte forma. A Seção II apresenta os trabalhos relacionados, enquanto a Seção III apresenta os métodos de resiliência e suas classificações. Na Seção IV, os resultados obtidos são apresentados e analisados. Finalmente, a Seção V conclui o trabalho e aponta direções futuras.

II. Trabalhos Relacionados

Métodos de resiliência em SDN são objetos de estudos em diversos trabalhos da literatura. Van Adruchem *et al.* propõem um método de recuperação rápida baseado na identificação de falhas de enlaces, combinando caminhos primários e secundários configurados por um controlador central [3]. Além disso, o trabalho implementa a identificação de falhas por enlace do BFD. Com a implementação desta abordagem proativa proposta no artigo, é possível alcançar uma recuperação abaixo de 50 ms.

A ideia de pré-instalar entradas de fluxo redundantes no plano de dados também já foi abordada em [6], no qual propõe-se o aprimoramento do plano de dados por

Rodrigo Amancio e Marcelo G. Rubinstein, Programa de Engenharia Eletrônica (PEL) - Universidade do Estado do Rio de Janeiro (UERJ). Rodrigo S. Couto, Grupo de Teleinformática e Automação (GTA) - PEE/COPPE - Universidade Federal do Rio de Janeiro (UFRJ). E-mails: rodrigo.s.amancio@gmail.com, rodrigo@gta.ufrj.br, rubi@uerj.br

meio de rotas pré-configuradas. Esta abordagem possibilita que 60% dos pedidos de reconexão possam ser manipulados neste plano. Ademais, como apenas uma fração das solicitações é manipulada pelo controlador, ocorre uma diminuição na sobrecarga de controle.

Outra solução analisada como um método para recuperação rápida em caso de falhas corresponde à implementação da agregação de enlaces por meio do uso do LACP [7], sendo possível integrar várias portas físicas em conjunto para criar um único enlace lógico de comunicação.

Artigos relacionados como [3], [6] e [7] analisam o tempo de recuperação dos métodos de resiliência e de seus respectivos mecanismos, porém não fazem uma análise de desempenho da rede de forma mais profunda. Assim, neste artigo é realizada uma análise de desempenho da rede quantificando o *jitter* e a perda de pacotes em cenários com falha. Além disso, é realizada uma comparação entre cenários que utilizam os métodos de resiliência analisados neste artigo com cenários sem a implementação desses métodos.

III. Métodos de Resiliência e suas Classificações

Os mecanismos de resiliência podem ser classificados como métodos de proteção do caminho e métodos de proteção do enlace. Nos métodos de proteção do caminho, os nós de origem e de destino reservam estaticamente seus caminhos secundários de uma extremidade a outra durante a configuração. Já nos métodos de resiliência que utilizam a proteção do enlace, um enlace secundário é definido em torno do enlace primário. Esta definição do enlace secundário ocorre entre dois nós e não necessariamente de uma extremidade a outra do caminho. Neste método, todas as conexões que atravessam o enlace com falha são redirecionadas para o enlace secundário [8].

A análise de desempenho realizada neste artigo engloba os resultados dos métodos de proteção do caminho, como os fluxos pré-configurados, os fluxos pré-configurados com o BFD, e também os resultados utilizando o LACP que é um caso especial de proteção do enlace, no qual provisiona-se um enlace redundante para o enlace protegido.

A. Fluxos pré-configurados

Com a implantação dos fluxos pré-configurados, um caminho secundário é preestabelecido no plano de dados. Caso ocorra uma falha, este caminho é utilizado, evitando o envio de solicitações para o controlador e economizando o atraso de recuperação de ida e volta entre o plano de dados e o plano de controle [6].

Os fluxos pré-configurados podem ser implantados por meio da Tabela de Grupo de Recuperação Rápida, que é uma funcionalidade suportada a partir da versão 1.1 do protocolo *OpenFlow*. Esta funcionalidade pode ser configurada para monitorar o status de portas, interfaces e para alternar ações de encaminhamento independentemente do controlador [3]. Ademais, como mostra a Figura 1, após um tráfego ser recebido pela porta de entrada, uma ação de saída é definida. A Tabela de Grupos monitora continuamente um conjunto de portas de saída que podem encaminhar este tráfego caso seja necessário. Se o enlace de saída principal falhar, a Tabela de Grupos automaticamente encaminhará o tráfego para umas das portas que já estão sendo monitoradas e estão à disposição para serem utilizadas.

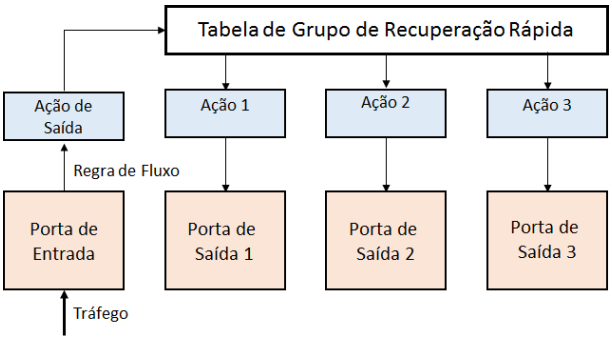


Fig. 1: Exemplo de Grupo de Recuperação Rápida [3].

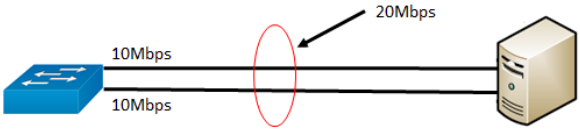
B. Detecção de Encaminhamento Bidirecional (BFD)

O protocolo de Detecção de Encaminhamento Bidirecional (BFD) implementa um mecanismo de controle de mensagens de eco para verificar se os enlaces estão ativos. Cada nó transmite mensagens de controle com o estado atual do enlace. Um nó que recebe uma mensagem de controle responde com uma mensagem de eco contendo seu respectivo status de sessão. Após esse processo, as mensagens de controle, que são enviadas de forma frequente, confirmam a ausência de uma falha no enlace [4].

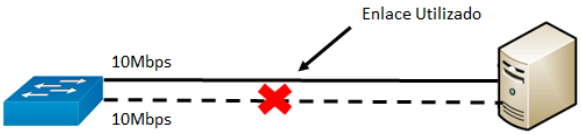
O BFD foi projetado para ser independente de protocolo e é utilizado para fornecer ou melhorar a detecção de falhas na rede [4]. Neste artigo, o BFD foi implementado no *Open vSwitch* [9], uma implementação popular de comutador *OpenFlow* [10].

C. Protocolo de Controle de Agregação de Enlace (LACP)

O Protocolo de Controle de Agregação de Enlace (LACP), definido no padrão IEEE 802.3ad, é um recurso de configuração automática que permite que várias interfaces físicas agregadas tornem-se um único enlace lógico, chamado de Grupo de Agregação de Enlace (*Link Aggregation Group* - LAG). A Figura 2(a) mostra o aumento da largura de banda devido à agregação de dois enlaces entre dispositivos. Já a Figura 2(b) mostra como ocorre a tolerância a falhas em caso de queda de um dos enlaces. O LACP assegura a comunicação confiável por meio de caminhos multiplexados que são usados como caminhos secundários em caso de falhas [7].



(a) Aumento da largura de banda



(b) Tolerância a Falhas

Fig. 2: Função da Agregação de Enlaces [7].

D. Resiliência pelo Controlador

Quando ocorre uma falha na rede o controlador precisa reconfigurar a rede para manter a conectividade entre os nós. O processo funciona de forma que o comutador detecta uma alteração e então notifica o controlador. Após a notificação, o controlador calcula as ações de reparo e envia atualizações para os elementos afetados, que por sua vez, atualizam suas tabelas de encaminhamento [11]. Todavia, todo esse processo, inclui o atraso introduzido pelo tempo de propagação da notificação ao controlador e o atraso da reconfiguração da rede [3].

IV. Análises e Resultados Obtidos

Para realizar as análises dos métodos de proteção foi utilizada a topologia da Figura 3. Todos os enlaces são de 10 Mbps e os testes são realizados utilizando o *Iperf* com tráfego UDP em uma taxa de 10 Mbps. O controlador SDN utilizado para os experimentos é o *OpenDayLight* e a emulação dos comutadores e estações é realizada pelo *Mininet*.

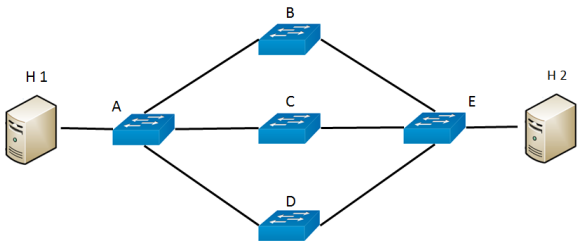


Fig. 3: Topologia de rede utilizada nos testes.

Para analisar o desempenho dos métodos apresentados, utiliza-se uma máquina Intel Core i5 com 4 GB de RAM e CPU de 2,50 GHz. O sistema operacional utilizado foi o Ubuntu 18.04.5 LTS.

A. Métricas de Desempenho

O *jitter* (variação de atraso) e a perda de pacotes são as métricas utilizadas para analisar os métodos de resiliência.

O *jitter* é a variação do tempo decorrido entre o momento em que um pacote é gerado na fonte e o momento em que é recebido no destinatário [12]. Através desta métrica, também se pode determinar o nível de estabilidade da rede com a utilização dos métodos de resiliência analisados neste artigo [13]. A métrica de perda de pacotes é utilizada para medir a quantidade de pacotes que não são entregues, sendo possível definir o desempenho da rede com a utilização dos métodos analisados [12].

Com exceção dos gráficos que mostram a vazão ao longo do tempo, todas as métricas foram apresentadas com médias e intervalos de confiança de 95%.

B. Resultados Obtidos nos Métodos de Proteção do Caminho

Para realizar as análises dos métodos de proteção do caminho, ou seja, dos fluxos pré-configurados e dos fluxos pré-configurados com BFD, são definidos caminhos primários e secundários como mostra a Figura 4. Na topologia, o caminho primário é o A-B-E e o caminho secundário é o A-D-E. Para a realização dos testes o enlace A-B é derrubado.

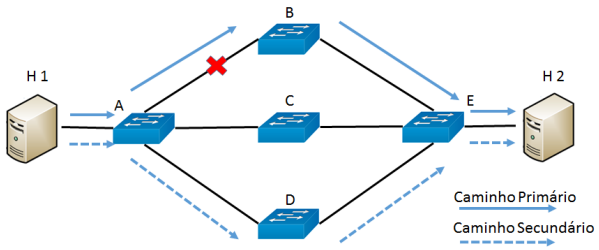


Fig. 4: Topologia com caminhos primário e secundário no núcleo da rede.

A Figura 5 mostra os resultados do *jitter* para cada um dos mecanismos de proteção do caminho. O *jitter* em um cenário em que existe uma rota pré-configurada no comutador (histograma laranja) é cerca de 0,6 ms menor em comparação com uma rota definida pelo controlador (histograma cinza). Essa diferença ocorre já que as solicitações tratadas no plano de dados não serão enviadas para o plano de controle, economizando assim o atraso de recuperação de ida e volta entre o plano de dados e o plano de controle [6].

É possível verificar também uma melhora no *jitter* após ser acrescentado aos mecanismos de fluxo pré-configurado e rota definida pelo controlador, a configuração do BFD nos enlaces. Nota-se uma diminuição do *jitter* de cerca de 0,5 ms em uma rota definida pelo controlador e 0,2 ms para uma rota pré-configurada, aproximando-se ainda mais do *jitter* de um cenário sem queda. Os melhores resultados em relação ao *jitter* ocorrem devido à troca de mensagens de eco proporcionada pelo BFD, que propicia a vantagem de testar frequentemente o caminho percorrido entre dois nós. Isso pode reduzir a variação de atraso de ida e volta e, assim, permitir um melhor tempo de

detecção, além de potencialmente detectar falhas que poderiam não ser detectadas [4].

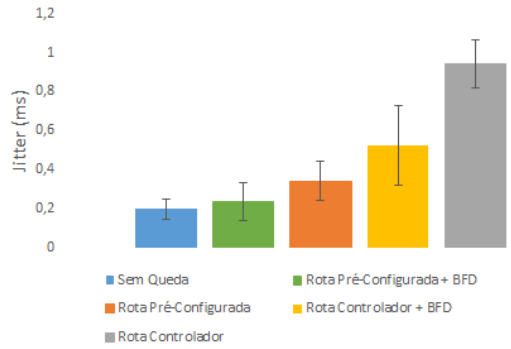


Fig. 5: Variação de atraso - Mecanismos de proteção do caminho.

A Figura 6 mostra os resultados de perda de pacotes para os mecanismos de proteção do caminho. É possível analisar que o cenário com a rota pré-configurada perde cerca de oito vezes menos pacotes em comparação com o cenário em que a decisão é tomada apenas pelo controlador. O cenário sem queda foi omitido nessa figura, já que em todas as rodadas a taxa de perda de pacotes é zero. Para ilustrar melhor o momento da queda, a Figura 7 mostra a vazão na linha do tempo para uma das amostras de experimento. Essa figura mostra a estabilidade da utilização do fluxo pré-configurado.

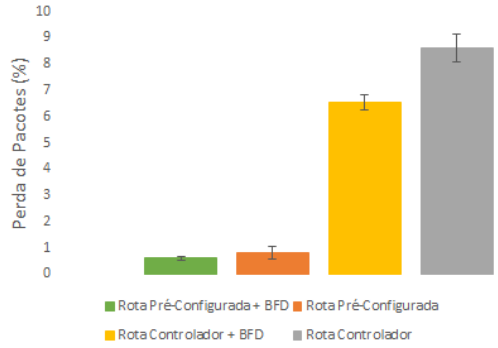


Fig. 6: Perda de Pacotes - Mecanismos de proteção do caminho.

Após a implantação do BFD, também é possível notar uma diminuição de 2,1 pontos percentuais no número de pacotes perdidos na rota definida pelo controlador e cerca de 0,25 pontos percentuais na rota pré-configurada, conforme a Figura 6. Na Figura 8 é ilustrada a vazão em função do tempo ao se usar o BFD em conjunto, tornando-se possível concluir que, mesmo com a implantação do BFD, a rota pré-configurada continua sendo melhor que a do controlador.

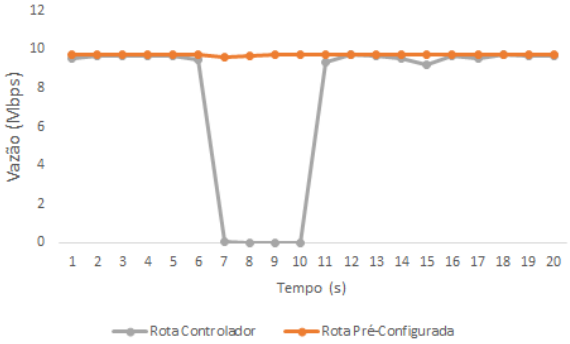


Fig. 7: Vazão ao longo do tempo - Rota Pré-configurada e Rota Controlador.

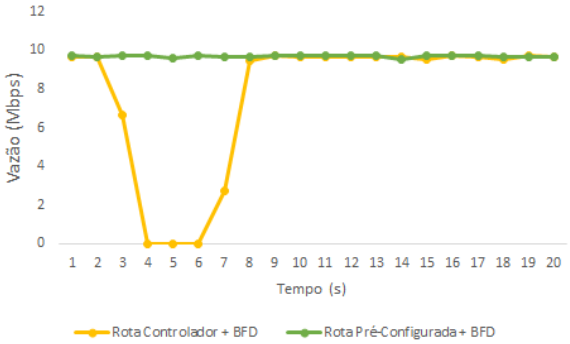


Fig. 8: Vazão ao longo do tempo - Rota Pré-configurada + BFD e Rota Controlador + BFD.

C. Resultados Obtidos nos Métodos de Proteção do Enlace

Para realizar as análises do método de proteção do enlace, ou seja, das agregações de enlaces utilizando o LACP, a topologia anterior foi modificada de acordo com a Figura 9. Todos os enlaces da topologia são de 10 Mbps, porém entre H1 e A existem dois enlaces agregados, ou seja, um LAG. Para a realização dos testes de proteção do enlace, um dos enlaces físicos do LAG é derrubado.

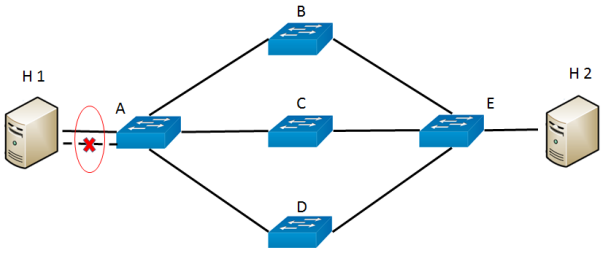


Fig. 9: Topologia com caminhos primário e secundário na borda da rede.

A Figura 10 mostra o *jitter* para duas situações: uma sem a queda do enlace e a outra com a queda do enlace. É possível analisar que mesmo com uma queda em um dos enlaces físicos do LAG, o *jitter* é estatisticamente igual em comparação com um cenário sem falhas na

rede.

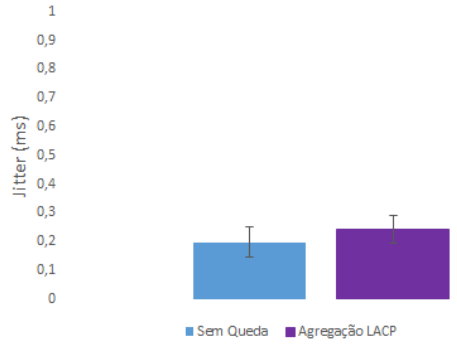


Fig. 10: Variação de atraso - Agregação de enlace formado pelo LACP.

Com relação à perda de pacotes, a taxa de perdas foi zero mesmo com a queda de um dos enlaces físicos. Além disso, é possível notar na Figura 11 que a resiliência ocorre de forma automática mantendo a transferência de pacotes. Com base nos resultados encontrados, foi possível concluir que a transferência de dados apenas não será realizada em caso de falha no LAG, ou seja, no enlace lógico formado por todos os enlaces físicos [14].

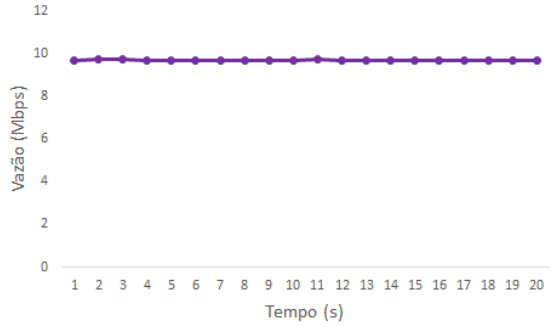


Fig. 11: Vazão ao longo tempo - Agregação de enlace formado pelo LACP.

V. Conclusões e Trabalhos Futuros

Este artigo fez uma análise de desempenho de métodos de resiliência que podem ser implantados em uma rede SDN. Com relação ao método de proteção do caminho, que foi implantado no núcleo da rede, foi possível concluir que a funcionalidade de fluxo pré-configurado diminui consideravelmente a variação de atraso e a perda de pacotes em comparação com uma rota definida pelo controlador. Além disso, com a implementação do BFD nos enlaces é possível melhorar ainda mais o desempenho. Isso se dá devido à capacidade do BFD de monitorar constantemente se os enlaces estão ativos.

O método de proteção do enlace, que foi testado na borda da rede, mostrou-se eficaz em caso de falhas. A variação de atraso manteve-se estatisticamente igual à de uma rede sem falhas e não houve perda de pacotes mesmo com a queda de um dos enlaces do LAG. Além disso, foi possível concluir que o tráfego não é interrompido devido a falha no enlace [15].

Como trabalhos futuros pretende-se analisar a resiliência em redes com mais de um controlador, no intuito de evitar a existência de um único ponto de falha na rede. Além disso, pode ser estudado como os métodos analisados neste artigo podem melhorar a estabilidade e confiabilidade da rede neste cenário.

Agradecimentos

O presente trabalho foi realizado com apoio do CNPq, da FAPERJ, da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), auxílios no. 2015/24494-8 e 2015/24490-2 e da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Código de Financiamento 001.

Referências

[1] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, pp. 14–76, Jan. 2015.

[2] S. Shenker, M. Casado, T. Koponen, N. McKeown *et al.*, "The future of networking, and the past of protocols," *Open Networking Summit*, vol. 20, pp. 1–30, 2011.

[3] N. L. Van Adrichem, B. J. Van Asten, F. A. Kuipers *et al.*, "Fast recovery in software-defined networks." *EWSDN*, vol. 14, pp. 61–66, 2014.

[4] D. Katz and D. Ward, "Bidirectional forwarding detection (BFD)," Tech. Rep., 2010.

[5] "IEEE Standard for Ethernet Link Aggregation:IEEE802.3ad," http://www.ieee802.org/3/ad/ - Acessado em abril de 2019.

[6] A. Xie, X. Wang, W. Wang, and S. Lu, "Designing a disaster-resilient network with software defined networking," in *2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)*. IEEE, 2014, pp. 135–140.

[7] I. D. Irawati, S. Hadiyoso, and Y. S. Hariyani, "Link aggregation control protocol on software defined network," *International Journal of Electrical and Computer Engineering*, vol. 7, no. 5, p. 2706, 2017.

[8] S. Ramamurthy and B. Mukherjee, "Survivable wdm mesh networks. part i-protection," in *IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320)*, vol. 2. IEEE, 1999, pp. 744–751.

[9] B. Pfaff, J. Pettit, K. Amidon, M. Casado, T. Koponen, and S. Shenker, "Extending networking into the virtualization layer." in *Hotnets*, 2009.

[10] F. L. Rodriguez and D. R. Campelo, "Rede SDN-openflow para o caso de um ISP: Desafios e oportunidades." *SBrT*, 2013.

[11] S. H. Yeganeh, A. Tootoonchian, and Y. Ganjali, "On scalability of software-defined networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 136–141, 2013.

[12] J. F. Kurose and K. W. Ross, "Redes de computadores e a internet (5ª edição)," 2010.

[13] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," Tech. Rep., 2003.

[14] B. Sevcik, H. Zeilinger, T. Turek, and G. Zucker, "Network layer based redundancy for time-critical voip applications," in *AFRICON 2009*. IEEE, 2009, pp. 1–5.

[15] P. M. Nair, S. V. Nair, M. Marchetti, G. Chiruvolu, and M. Ali, "Bandwidth sensitive fast failure recovery scheme for metro ethernet," *Computer Networks*, vol. 52, no. 8, pp. 1603–1616, 2008.