Autocura para Redes Definidas por Software

Natal Vieira de Souza Neto[1], **Pedro Frosi Rosa**[1],
Flávio de Oliveira Silva[1], **Luiz A. DaSilva**[2]

[1]Faculdade de Computação – Universidade Federal de Uberlândia (UFU)

[2]Commonwealth Cyber Initiative, Electrical and Computer Engineering, Virginia Tech

Abstract. *The introduction of software-defined networking and network functions virtualisation brings many advantages. Nonetheless, the flexibility and programmability characteristics expected in such technologies require new operation components in the control and management layers. The quick failure recovery in these layers is essential because, without control or management, the data plane is also inoperable. This paper summarises a thesis that explores the self-healing autonomic computing fundamental as a solution for managing such layers. The results prove self-healing efficiency in network slices with strict quality of service requirements and demonstrate that the proposed solution can heal the degraded environment and heal itself.*

Resumo. *O advento de redes definidas por software e virtualização de funções de rede trouxe inúmeras vantagens; contudo, para atingir a flexibilidade e a programabilidade previstas nessas tecnologias, novos componentes nos planos de controle e gerenciamento foram introduzidos. Tais componentes requerem recuperação rápida pois, sem gerenciamento, todo o plano de dados fica inoperável. Para lidar com falhas nesses planos, recorre-se à técnica de autocura, explorada na tese que é resumida neste documento. Os resultados provam a eficácia de autocura em fatias de rede com rigorosos requisitos de qualidade e, também, demonstram que o framework introduzido é capaz de autocurar, ou seja, curar o ambiente degradado bem como curar a si mesmo.*

1. Introdução

As Redes Definidas por Software – *Software-defined Networking* (SDN) – e a Virtualização de Funções de Rede – *Network Functions Virtualisation* (NFV) – são abordagens consolidadas na academia e promissoras na indústria por permitirem programabilidade e flexibilidade em redes de computadores e redes móveis, as quais tradicionalmente possuem ambientes fechados e pouco flexíveis [Ramirez-Perez and Ramos 2016, Barona López et al. 2017]. Para atingir essas características, novos componentes são introduzidos e novos planos idealizados, como os planos de controle, gerenciamento e aplicações [Rehman et al. 2019]. Curiosamente, apesar da introdução de novos planos facilitar a implantação de funções em software, ela traz consigo novos problemas para gerenciamento [d. R. Fonseca and Mota 2017].

O principal problema em aberto tratado neste artigo é a saúde da rede. Pode-se afirmar que o gerenciamento do plano de dados avançou nos últimos anos, ao contrário do gerenciamento do plano de controle e do próprio plano de gerenciamento [Abdelsalam 2018, Cox et al. 2017]. No plano de controle há componentes, como por

exemplo controladores SDN, que requerem funcionamento ininterruptível. No plano de gerenciamento, aplicações de *Operations, Administration, and Maintenance* (OAM) [Mizrahi et al. 2014] cuidam da saúde da rede e, por isso, carecem de disponibilidade a todo momento. Visando autogerenciamento que englobe os planos de dados, controle e gerenciamento, a tese[1] aqui sintetizada parte dos fundamentos de computação autônoma [Ganek and Corbi 2003, 3GPP 2018a, 3GPP 2018b], em especial autocura – *self-healing* – para mitigar ou evitar falhas que comprometam o funcionamento saudável das redes. Por funcionamento saudável, entende-se que as aplicações e fatias de rede estejam entregando a Qualidade de Serviço – *Quality of Service* (QoS) – pré-estabelecida durante todo o tempo de execução [3GPP 2018c].

Problema. A adoção de tecnologias de software como SDN, NFV e *Software-defined Radio* (SDR) em sistemas de computação traz consigo novos elementos para controle, orquestração, gerenciamento, etc. Esses elementos requerem gerenciamento pois sem eles toda a comunicação fica comprometida, visto que eles que operam o ambiente do plano de dados. O impasse está na questão: Quem gerencia os próprios elementos de gerenciamento? Para tratar este problema, pode-se recorrer ao fundamento de autogerenciamento.

Hipótese. Considerando a premissa de que o fundamento de autocura aplicado às redes de comunicação é essencial para a existência de redes autonômicas, toma-se como hipótese: técnicas de autocura podem ser aplicadas nos três planos de redes autonômicas para prover qualidade de serviço e experiência de entidades comunicantes.

Para demonstrar a teoria supramencionada, foi desenvolvido um modelo conceitual que se materializou em uma arquitetura para autogerenciamento intitulada *Self-organising Networks Architecture* (SONAr), a qual serve como um modelo de referência para se atingir o nível de redes autonômicas. Tendo a SONAr como ponto de partida, um sistema de autogerenciamento, denominado *Network Operations Self-healing* (NOSH), com foco no fundamento de autocura, foi desenvolvido para experimentar autocura nos planos de controle, gerenciamento e dados.

1.1. Motivação

A principal motivação deste trabalho está na lacuna que existe no gerenciamento do plano de dados com o gerenciamento dos planos de controle e gerência. Esses últimos possuem diversas questões em aberto, sendo uma das principais a saúde da rede [Chandrasekaran et al. 2016, d. R. Fonseca and Mota 2017, Azab and Fortes 2017]. Este trabalho também é motivado pela iminente adoção de Fatiamento de Rede – *Network Slicing* – [Afolabi et al. 2018], no qual fatias de rede virtuais proverão a comunicação de aplicações com requisitos de QoS rigorosos. Uma rede pode estar com todos os elementos da topologia em funcionamento (*up*) e mesmo assim ser considerada não-saudável caso os requisitos de QoS não estejam sendo atingidos por quaisquer motivos.

No estado da arte, encontra-se arquiteturas de sistemas preparados para lidar com recuperação do plano de dados, mas não para problemas do plano de controle e do próprio plano de gerenciamento. Ainda, esses sistemas costumam curar o ambiente degradado, mas não autocurar, ou seja, curar a si mesmos. As contribuições desta tese estão relacionadas à autocura englobando também os planos de controle e gerenciamento.

[1]Tese disponível em: http://doi.org/10.14393/ufu.te.2021.246

2. Objetivos

O objetivo geral deste trabalho é a incorporação de aspectos de autocura na arquitetura SONAr, para garantir o funcionamento de redes SDN/NFV, permitindo que falhas nos planos de Controle, Dados e Gerenciamento possam ser resolvidas com a mínima intervenção humana. Para se alcançar o objetivo supracitado, propõe-se os seguintes objetivos específicos:

1. Especificação dos componentes e interoperabilidade dentro da SONAr necessários para rodar as funcionalidades de autocura;
2. Implementação dos serviços para monitoramento do plano de Dados considerando-se três abordagens: *Control loop*, Agentes Locais e Interceptação de primitivas na *Southbound Interface* (SBI);
3. Implementação dos serviços para monitoramento do plano de Controle e plano de Gerenciamento;
4. Aplicação de ações de recuperação clássicas, praticadas em *Self-organising Networks* (SON), para recuperar falhas nos planos de Dados – considerando *Network Slice*s (NSs) –, Controle e Gerenciamento;
5. Avaliação de técnicas de monitoramento por detecção *vs* predição e técnicas de recuperação *real-time* e *bucket-based*;
6. Demonstração da solução final, o *framework* NOSH, em um ambiente SONAr, executando todos os serviços e componentes em forma de funções de rede virtualizadas.

3. Sistema de Gerenciamento de Autocura

O modelo de referência criado a partir desta tese, SONAr [Souza Neto 2021], considera que um domínio de redes de comunicação possui três camadas principais, quais sejam: infraestrutura, contendo hardware físico ou virtual; controle, contendo orquestradores e controladores de SDN e NFV; e aplicação, contendo as aplicações de funções de rede. Para gerenciar essas três camadas, idealizou-se um Plano de Gerenciamento no qual a SONAr é executada.

3.1. Arquitetura para Redes Auto-organizáveis

A SONAr prevê entidades categorizadas em quatro grupos principais, sendo *Collector Entities* (CoEs), *Self-learning Entities* (SLEs), *Self-organising Entities* (SOEs) e *Integration Entities* (IEs). As CoEs são responsáveis por coletar e receber todo tipo de informação de todas as camadas do domínio. As SOEs aplicam algoritmos de monitoramento e recuperação em tempo real. As SLEs fornecem serviços de aprendizado de máquina para predição, no intuito de mitigar problemas futuros. Por fim, as IEs são entidades que possuem serviços para integrar a SONAr com componentes de controle do domínio (como controladores, orquestradores, ferramentas de OAM, etc.).

A SONAr prevê um *Network Event Manager* (NEM) para que eventos possam ser enviados e recebidos de/para qualquer entidade. Por exemplo, um evento *link down* pode ser recebido pela *Self-healing Entity* (SHE) e pela *Self-optimisation Entity* (SOPE) concomitantemente visto que esse tipo de evento faz sentido para essas duas SOEs. Ainda, a SONAr tem um *Network Database* (NDB) para armazenar informações operacionais e eventos recebidos do domínio (que servem de insumos para as SLEs). Por fim, a SONAr

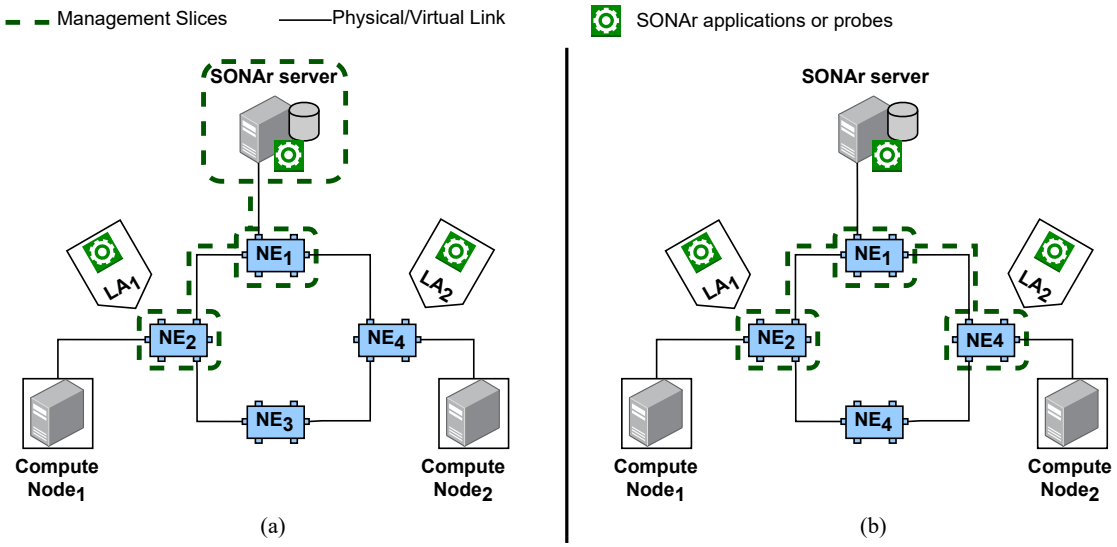

Figura 1. Fatias de Gerenciamento.

ainda possui um *Control Primitives Interceptor* (CPI) para interceptar primitivas na SBI e *Local Agent*s (LAs), que são agentes para *probing* local. Com esse modelo de referência, é possível implementar um sistema de autogerenciamento para redes com as tecnologias contemporâneas.

3.2. Rede Lógica para Gerenciamento

O ponto fulcral da tese aqui relatada está na rede lógica para gerenciamento, posto de outro modo, na idealização de Fatias de Gerenciamento. Essas fatias funcionam conforme o conceito de fatiamento de rede, mas cuidam exclusivamente do tráfego de primitivas de controle e gerenciamento.

A Figura 1 apresenta dois exemplos de fatias de gerenciamento. Nota-se que a Figura 1a possui um LA (LA_1) se comunicando com entidades da SONAr presentes no **SONAr server**. O NOSH aprovisiona recursos para a fatia de gerenciamento de modo que as primitivas de controle entre LA_1 e **SONAr server** sejam trafegadas prioritariamente. O NOSH deve monitorar o LA_1 (servidor e aplicação), o **SONAr server** (servidor e aplicações), o elemento de rede NE_1, o elemento de rede NE_2, e os links NE_2–NE_1 e NE_1–**SONAr server**. Dessa forma, prioriza-se a manutenção em tempo real da comunicação de controle (através da fatia de gerenciamento).

Na Figura 1b está exemplificada outra fatia de gerenciamento, que garante a comunicação entre dois LAs. O NOSH, desenvolvido fielmente à luz da arquitetura SONAr, é capaz de manter um catálogo dos componentes de controle e gerenciamento (inclusive componentes do próprio NOSH) executando no momento, seus servidores e links. Assim, o NOSH mantém a saúde dos componentes de controle e gerenciamento através das fatias de gerenciamento, pelas quais as primitivas de controle/gerência trafegam. Na Figura 1, os serviços do NOSH executam no **SONAr server**.

3.3. Inicialização Autônoma

A inicialização autônoma é outro ponto forte do NOSH. Por motivos óbvios, no dia zero deve-se ativar o *script* de *booting* manualmente. A partir daí, o NOSH possui mecanismos

para inicializar todas as entidades previstas na SONAr, controladores SDN, orquestradores, etc. Nenhuma intervenção manual é mais necessária.

Além do *start* dos serviços, o NOSH aprovisiona os recursos necessários para estabelecimento das fatias de gerenciamento, isto é, inicializa as aplicações necessárias e aplica as regras de encaminhamento nos *Network Element*s (NEs) (*switches*, roteadores, *gateways*, etc.). Para exemplificar, na topologia da Figura 1b o NOSH estabeleceria regras para todas as rotas entre LA_1 e LA_2. Para coletar métricas de latência *multi-path*, é necessário que todas as possíveis rotas estejam em funcionamento. Na prática, para enviar pacotes por diferentes rotas, a implementação dos LAs utiliza interfaces virtuais do Linux, de modo que diferentes IPv4 cheguem aos NEs e estes saibam para qual porta direcionar.

3.4. Monitoramento e Recuperação de Falhas

Para o monitoramento e identificação de falhas, o NOSH coleta diversos tipos de informação, tais como métricas, alterações topológicas, alarmes, *logs*, dentre outras. Existem três formas de coleta de informação. A primeira, *Autonomic Control Loop* (ACL), funciona na direção CoEs $\rightarrow$ NEs. A segunda, LAs, funciona na direção NEs $\rightarrow$ CoEs. Finalmente, a terceira, CPI, intercepta primitivas na SBI para corrigir alguma primitiva que possa afetar a comunicação do controlador ou descartar primitivas maliciosas. Entende-se que essas três técnicas de coleta abrangem o estado da arte.

Com as informações coletadas, a identificação de falhas ocorre por detecção, através da SHE, ou predição, através da *Prediction Self-learning Entity* (PSLE). A predição visa evitar ou mitigar falhas futuras, mas a detecção ainda é necessária pois uma falha pode eventualmente ocorrer mesmo que os melhores algoritmos de predição estejam atuando. Uma vez identificada uma falha, o NOSH parte para a fase de recuperação.

Quatro tipos de ações de recuperação são executados, sendo eles: reinicialização, que trata do *reboot* de componentes (como NEs virtuais, containers de aplicações, etc.); reconfiguração de recursos, que trata de reconfigurar parâmetros computacionais (como vCPU, memória, etc.) ou rotas entre entidades; redefinição, que trata do *reset* de unidade de software de componentes; e migração, que trata de migrar aplicações, controladores, orquestradores, etc. para outro ponto do domínio. O *workflow* (sequência de execução das ações) ocorre pela combinação das ações (caso uma ação não resolva o problema, parte-se para outra).

4. Resultados

Para comprovar a eficácia do NOSH, duas implementações fiéis à SONAr foram realizadas. A primeira, implantada na infraestrutura da FACOM/UFU, buscou medir o desempenho do NOSH para autogerenciamento (inicialização autônoma, recuperação com diferentes tipos de ações, etc.). A segunda, implantada no IRIS Testbed[2], buscou avaliar o NOSH como ferramenta OAM integrada a um orquestrador de fatias de rede com diversas fatias sendo provisionadas com requisitos de QoS rigorosos (o intuito é que o NOSH seja capaz de manter as fatias de rede entregando o QoS acordado e, no caso de falhas, reconfigurar as fatias autonomamente).

[2]Iris testbed, Trinity College Dublin.

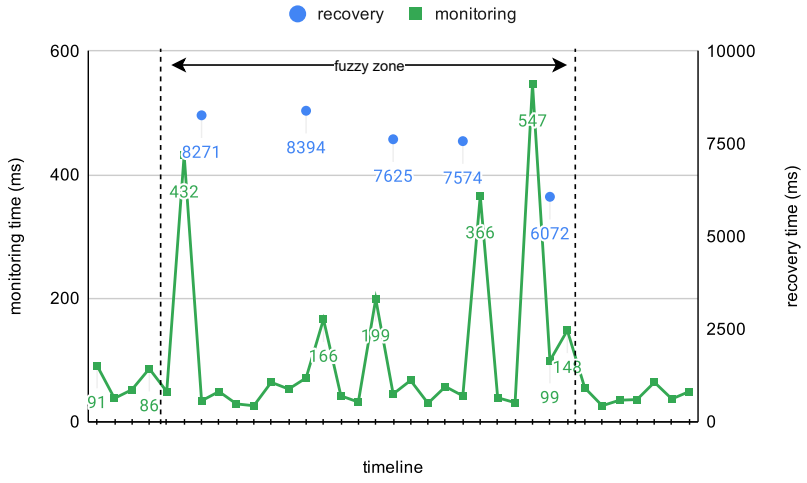

Figura 2. Avaliação de desempenho de monitoramento. Quando há recuperação de falhas, o tempo de monitoramento é diretamente proporcional ao de recuperação pois engloba o procedimento de identificação de falhas.

4.1. Resultados Técnicos

Os experimentos realizados na tese buscavam: avaliar o tempo de inicialização do NOSH e desempenho de coleta de métricas; comparar diferentes técnicas de recuperação; avaliar o desempenho de monitoramento *vs* recuperação; avaliar a detecção de anomalias em fatias de *Ultra-Reliable and Low Latency Communications* (URLLC); comparar recuperação por detecção *vs* predição; avaliar o estabelecimento autônomo das fatias de gerenciamento; e avaliar o monitoramento e recuperação de centenas de fatias de rede executando concomitantemente. As Figuras 2 e 3 trazem dois resultados, sendo que os demais constam no documento original da tese.

4.2. Resultados Científicos

Abaixo uma listagem dos artigos científicos publicados resultantes desta pesquisa.

- de Souza Neto N.V., Oliveira D.R.C., Gonçalves M.A., de Oliveira Silva F., Rosa P.F. (2021) Self-healing in the Scope of Software-Based Computer and Mobile Networks. In: Ferguson D., Pahl C., Helfert M. (eds) Cloud Computing and Services Science. CLOSER 2020. Communications in Computer and Information Science, vol 1399. Springer, Cham. https://doi.org/10.1007/978-3-030-72369-9_14.
- Neto, N.; Oliveira, D.; Gonçalves, M.; Silva, F. and Rosa, P. (2020). A Self-healing Platform for the Control and Management Planes Communication in Softwarized and Virtualized Networks.In Proceedings of the 10th International Conference on Cloud Computing and Services Science – Volume 1: CLOSER, ISBN 978-989-758-424-4, pages 415-422. DOI: 10.5220/0009465204150422.
- Oliveira, D.; Neto, N.; Gonçalves, M.; Silva, F. and Rosa, P. (2020). Network Self-configuration for Edge Elements using Self-Organizing Networks Architecture (SONAr).In Proceedings of the 10th International Conference on Cloud Computing and Services Science – Volume 1: CLOSER, ISBN 978-989-758-424-4, pages 408-414. DOI: 10.5220/0009465104080414.

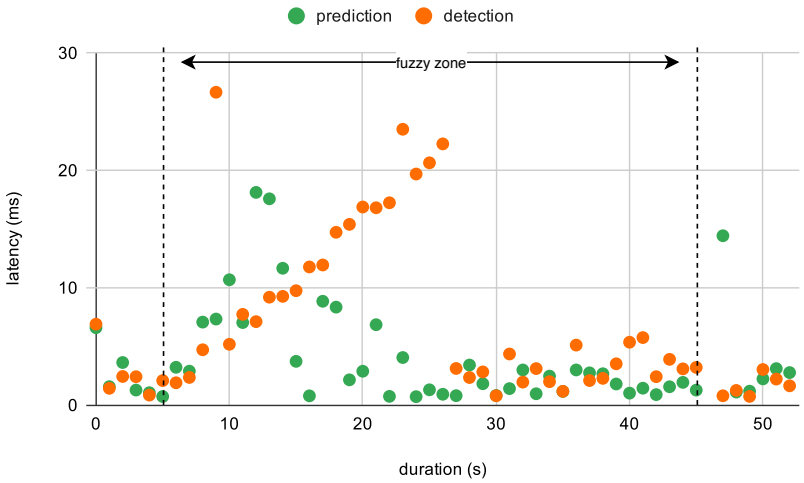

Figura 3. Detecção *vs* predição. Ao aplicar predição, a fatia de rede não apresentou latência fim-a-fim maior que o requisitado (20 ms) pois o NOSH previu o aumento linear de latência e reconfigurou as rotas.

- Gonçalves, M.; Neto, N.; Oliveira, D.; Silva, F. and Rosa, P. (2020). Bootstrapping and Plug-and-Play Operations on Software Defined Networks: A Case Study on Self-configuration using the SONAr Architecture.In Proceedings of the 10th International Conference on Cloud Computing and Services Science – Volume 1: CLOSER, ISBN 978-989-758-424-4, pages 103-114. DOI: 10.5220/0009406901030114.
- J. F. Santos; Wei Liu; Xianjun Jiao; Natal V. Neto; Sofie Pollin; Johann M. Marquez-Barja; Ingrid Moerman and Luiz A. DaSilva. "Breaking Down Network Slicing: Hierarchical Orchestration of End-to-End Networks," in IEEE Communications Magazine, vol. 58, no. 10, pp. 16-22, October 2020, doi: 10.1109/MCOM.001.2000406.
- Oliveira, D.; Gonçalves, M.; Neto, N.; Silva, F. and Rosa, P. (2021). Specialized Network Self-configuration: An Approach using Self-Organizing Networks Architecture (SONAr). In Proceedings of the 11th International Conference on Cloud Computing and Services Science - CLOSER, ISBN 978-989-758-510-4, pages 161-168. DOI: 10.5220/0010403601610168.

5. Conclusão

A pesquisa realizada para esta tese permitiu, por meio de uma minuciosa revisão bibliográfica e análise de experiências reais, a criação de um modelo de referência para redes autonômicas, que se materializou na arquitetura SONAr. Ainda, um sistema para autogerenciamento foi entregue implementado e avaliado sob duas diferentes perspectivas: autogerenciamento dos componentes do plano de controle e gerenciamento; e autocura de fatias de rede com requisitos de QoS pré-estabelecidos. Pode-se afirmar que o framework entregue, o NOSH, é capaz de curar componentes degradados nos planos de controle, gerenciamento ou dados e, ainda, autocurar, ou seja, o NOSH é capaz de curar a si mesmo.

Referências

3GPP (2018a). Telecommunication management; Self-configuration of network elements; Concepts and requirements. Technical Specification (TS) 32.501, 3rd Generation Partnership Project (3GPP).

3GPP (2018b). Telecommunication management; Self-Organizing Networks (SON); Self-healing concepts and requirements. Technical Specification (TS) 32.541, 3rd Generation Partnership Project (3GPP).

3GPP (2018c). Telecommunication management;Study on management and orchestration of network slicing for next generation network. Technical Report (TR) 28.801, 3rd Generation Partnership Project (3GPP). Version 15.1.0.

Abdelsalam, M. A. (2018). *Network Application Design Challenges and Solutions in SDN*. PhD thesis, Carleton University.

Afolabi, I., Taleb, T., Samdanis, K., Ksentini, A., and Flinck, H. (2018). Network slicing and softwarization: A survey on principles, enabling technologies, and solutions. *IEEE Communications Surveys Tutorials*, 20(3):2429–2453.

Azab, M. and Fortes, J. A. B. (2017). Towards proactive sdn-controller attack and failure resilience. In *2017 International Conference on Computing, Networking and Communications (ICNC)*, pages 442–448.

Barona López, L., Valdivieso Caraguay, Á., Sotelo Monge, M., and García Villalba, L. (2017). Key technologies in the context of future networks: operational and management requirements. *Future Internet*, 9(1):1.

Chandrasekaran, B., Tschaen, B., and Benson, T. (2016). Isolating and tolerating sdn application failures with legosdn. In *Proceedings of the Symposium on SDN Research*, SOSR '16, pages 7:1–7:12, New York, NY, USA. ACM.

Cox, J. H., Chung, J., Donovan, S., Ivey, J., Clark, R. J., Riley, G., and Owen, H. L. (2017). Advancing software-defined networks: A survey. *IEEE Access*, 5:25487–25526.

d. R. Fonseca, P. C. and Mota, E. S. (2017). A survey on fault management in software-defined networks. *IEEE Communications Surveys Tutorials*, 19(4):2284–2321.

Ganek, A. G. and Corbi, T. A. (2003). The dawning of the autonomic computing era. *IBM systems Journal*, 42(1):5–18.

Mizrahi, T., Sprecher, N., Bellagamba, E., and Weingarten, Y. (2014). An Overview of Operations, Administration, and Maintenance (OAM) Tools. RFC 7276.

Ramirez-Perez, C. and Ramos, V. (2016). Sdn meets sdr in self-organizing networks: fitting the pieces of network management. *IEEE Communications Magazine*, 54(1):48–57.

Rehman, A. U., Aguiar, R. L., and Barraca, J. P. (2019). Fault-tolerance in the scope of software-defined networking (sdn). *IEEE Access*, 7:124474–124490.

Souza Neto, N. V. d. (2021). *Autocura para redes definidas por software*. PhD thesis, Universidade Federal de Uberlândia, Uberlândia.