

Uma Proposta de Arquitetura para o Monitoramento Multidomínio de Cloud Network Slices

André L. Beltrami Rocha¹, Matheus N. Garcia¹, Vinicius F. Carvalho¹,
Paulo Ditarso Maciel Jr.^{1,2}, Fábio Luciano Verdi¹

¹Departamento de Computação (DComp)
Universidade Federal de São Carlos (UFSCar)
Sorocaba - SP, Brasil

²Unidade Acadêmica de Informação e Comunicação
Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB)
João Pessoa - PB, Brasil

{beltrami, verdi}@ufscar.br

{vinicius.furukawa, matheus.nadaleti}@dcomp.sor.ufscar.br

paulo.maciel@ifpb.edu.br

Abstract. *The main feature of the cloud network slicing concept is the provision of a physical and virtual end-to-end infrastructure, capable of supporting a variety of vertical industries. Such a pool is instantiated over a large number of administrative and technological domains, which brings a greater challenge for the management and monitoring of allocated resources. Resource monitoring in this new entity so called slice is extremely important to address MANO operations. Therefore, in this work we propose an architecture for the monitoring of physical and virtual resources in cloud network slices, considering multiples administrative and technological domains. Moreover, we present an information model for the monitoring of slices and a proof of concept implementation, in which we observe the slice monitoring resources metrics over time.*

Resumo. *O conceito de cloud network slicing oferece como sua principal característica o provisionamento de uma infraestrutura física e virtual fim-a-fim capaz de dar suporte a uma variedade de indústrias verticais. Tal infraestrutura é instanciada ao longo de múltiplos domínios administrativos e tecnológicos, o que torna um desafio ainda maior gerenciar e monitorar os recursos alocados. O monitoramento dos recursos desta nova entidade chamada de slice é de suma importância para que as operações de gerência e orquestração sejam possíveis. Portanto, este trabalho propõe uma arquitetura para o monitoramento de recursos físicos e virtuais em cloud network slices, considerando multidomínios administrativos e tecnológicos. Além disso, este trabalho apresenta um modelo de informação preliminar para o monitoramento de slices e implementa uma prova de conceito, na qual foi possível observar algumas métricas das slices sendo monitoradas ao longo do tempo.*

1. Introdução

Por meio de tecnologias emergentes como Redes Definidas por Software (SDN, do inglês *Software-Defined Networking*) e Virtualização de Funções de Rede (NFV, do inglês

Network Functions Virtualization), além de tecnologias consolidadas de computação na nuvem, a “softwarização” de rede pode fornecer a flexibilidade e modularidade necessárias para criar várias redes (virtuais) lógicas, cada uma adaptada para um determinado caso de uso, em cima de uma infraestrutura de rede comum.

A partir destas tecnologias habilitadoras e da necessidade de prover uma infraestrutura fim-a-fim, surge o conceito de *cloud network slicing*. Este novo conceito encapsula em uma única unidade, a *slice*, categorias de elementos oriundos da computação em nuvem, redes e armazenamento. Portanto, o principal motivador deste novo conceito é prover a instanciação dos recursos de computação em nuvem, rede e armazenamento ao longo de múltiplos domínios administrativos e/ou tecnológicos de maneira integrada. Uma vez instanciados, o locatário que solicitou a criação da *slice* poderá acessar, configurar e gerenciar os recursos como preferir, de modo completamente isolado, até mesmo sublocando tais recursos para outros locatários.

Diversas entidades e projetos vêm estudando este tema para definir algumas opções de arquitetura para a criação e gerência de uma *slice*. Entretanto, operações importantes como o monitoramento e a orquestração dos recursos estão ainda incipientes na literatura e possuem muitos desafios a serem explorados. Em um ambiente onde os elementos envolvidos são tão diversos, um serviço de orquestração é extremamente necessário para coordenar os recursos alocados, realizar operações de elasticidade, entre outras. De acordo com a *Open Network Foundation* (ONF) [TR-521 2016], a orquestração é definida como um processo contínuo de seleção de recursos para atender às demandas de atendimento ao cliente de maneira ideal.

No cenário de *cloud network slicing* contextualizado acima, a orquestração dos recursos físicos ou virtuais não pode ser realizada eficientemente sem uma entidade capaz de monitorar diferentes recursos alocados em múltiplos domínios. Com o surgimento deste novo conceito *slice*, uma nova entidade fim-a-fim necessita ser monitorada, composta conceitualmente pelo monitoramento de recursos relacionados à computação, rede e armazenamento.

Este artigo busca equacionar alguns destes problemas através da proposta de uma arquitetura para o monitoramento de *cloud network slices*, considerando o desafio de monitorar múltiplos domínios administrativos e tecnológicos. Como contribuições deste trabalho, pode-se destacar a proposta de uma arquitetura e um modelo de informação para o monitoramento de *slices*. Foi implementada, uma prova de conceito que visa demonstrar de forma minimalista um cenário composto por duas *slices* sendo monitoradas ao longo do tempo. Com tais informações, diferentes ferramentas de visualização podem ser desenvolvidas de forma a facilitar a navegabilidade e operações de orquestração para esta nova entidade *slice* e seus respectivos recursos e/ou serviços.

Este trabalho está organizado como descrito a seguir. A Seção 2 apresenta alguns conceitos básicos para um melhor entendimento sobre *cloud network slicing*. A Seção 3 descreve alguns trabalhos relacionados à gerência e ao monitoramento de *slices*. As propostas de uma arquitetura multidomínio e de um modelo de informação para o monitoramento de *slices* estão presentes na Seção 4. A Seção 5 aborda questões relacionadas à implementação da proposta, como por exemplo as tecnologias habilitadoras. A Seção 6 apresenta a prova de conceito e os resultados preliminares obtidos. Por fim, a Seção 7

finaliza este artigo expondo as conclusões encontradas e perspectivas de trabalhos futuros.

2. Conceitos Básicos

Baseado na definição em [Clayman 2018], *cloud network slice*¹ é uma infraestrutura fim-a-fim que envolve recursos relacionados à computação em nuvem, rede e armazenamento. Esta variedade de recursos deve ser instanciada de forma transparente para o locatário, para que ele possa realizar operações de gerência e orquestração. Por ser um conceito fim-a-fim, *slices* incluem diversos componentes que pertencem a domínios administrativos diferentes e, portanto, exigem um nível de abstração mais alto para a gerência e monitoramento dos mesmos. Além disso, permite novos modelos de negócios, onde alguns atores desempenham funções comerciais distintas, dos quais valem a pena serem destacados (atores e terminologias):

- *Provedor de infraestrutura*: possui e gerencia uma determinada infraestrutura. Esses recursos, físicos ou virtualizados, são oferecidos para compor uma *slice* requisitada por um único ou vários locatários;
- *Provedor de slice*: possui ou não infraestrutura física própria, mas é responsável por oferecer “serviços” de *slices* fim-a-fim entre domínios administrativos distintos, como por exemplo criação, remoção e gerência de *slices*;
- *Slice Parte*: uma *slice* pode estar instanciada ao longo de muitos provedores de infraestrutura onde, para cada domínio administrativo a porção dos recursos que correspondem à determinada *slice* é denominada de *slice parte*;
- *Locatário*: contrata recursos de um ou mais provedores de *slices*, onde o locatário pode realizar, gerenciar e fornecer serviços para seus usuários. Um serviço é uma composição de elementos de rede, computação e/ou armazenamento, e é definido em termos de como tais elementos são acessados e conectados;
- *Domínio administrativo*: conjunto de recursos e infraestruturas gerido por uma única entidade, organização ou empresa; que mantém e governa tais recursos sob uma administração comum, podendo ou não utilizar diferentes domínios tecnológicos (tecnologias distintas de gerenciamento dos recursos);
- *Domínio entrante*: provedor de *slice* onde o locatário solicita a formação de uma *cloud network slice* (multidomínios administrativos), geralmente escolhido por critérios de localidade e/ou afinidade comercial.

A atuação dos membros acima é naturalmente recursiva, onde, por exemplo, provedores e locatários podem atuar em um padrão vertical multicamadas, em que um locatário em uma camada atua como provedor de infraestrutura ou *slice* na camada imediatamente acima. A recursão mencionada aqui implica que um locatário pode fornecer serviços de rede para um usuário final, mas também para outro locatário. Na Figura 1, observa-se as interações de cada uma das terminologias descritas, assim como um componente responsável por monitorar as *slices* instanciadas, que será descrito posteriormente.

Apesar da natureza efêmera dos recursos virtuais instanciados em serviços de nuvem, é provável que os dados gerados por tais recursos sejam importantes de serem coletados e observados. O monitoramento destes dados oferece não apenas uma medida de desempenho dos serviços, mas também uma valiosa fonte de informação do comportamento do sistema como um todo [Tusa et al. 2018].

¹Para facilitar à leitura, abreviaremos *cloud network slice* para *slice* no decorrer de todo o texto.

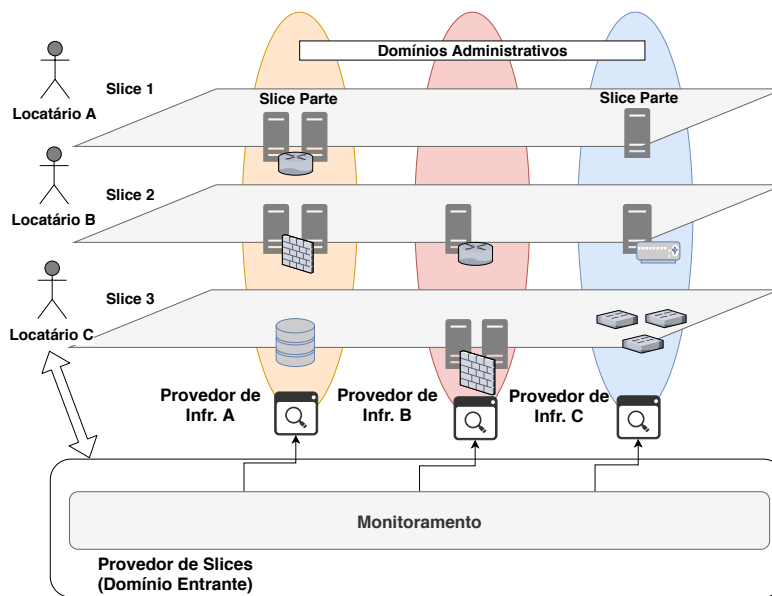


Figura 1. Ilustração da terminologia utilizada.

Dentre as características importantes desejadas em uma *cloud network slice* podem ser citadas: gerenciamento automático do ciclo de vida, operação autônoma, provisionamento e gerência de recursos simplificados, re-provisionamento dinâmico, alta escalabilidade e confiabilidade, eficiência econômica e rápida implantação da rede ou do serviço, possibilidade de novos modelos de negócios, dentre outras.

Há diversos esforços de padronização de *slices* por diferentes entidades, como por exemplo 3GPP [de Foy and Rahman 2017], ETSI [ETSI ISG NFV 2018] e NGMN [NGMN Alliance 2018]. Definir um modelo de informação faz parte destes esforços [Qiang et al. 2018], com o intuito de padronizar a comunicação entre os requisitos de um *slice*, que podem ser agnósticos as tecnologias, e o que realmente é implementado/provisionado nos provedores de *slices*, principalmente quando considerado o multidomínio tecnológico.

Contudo, apesar de todo o empenho em se criar padrões pelas entidades descritas acima, não há uma proposta definida especificamente para o monitoramento de *slices*, muito menos uma proposta de modelo de informação específico para o monitoramento de recursos de uma *slice*. Esta lacuna precisa ser preenchida e o objetivo do trabalho proposto neste artigo é dar um primeiro passo nesta direção.

3. Trabalhos Relacionados

O monitoramento de *slices* é de extrema importância para que os provedores e locatários possam gerenciar e orquestrar os recursos instanciados. Na literatura há um número considerável de trabalhos que abordam o monitoramento de infraestruturas físicas, de rede e de recursos virtuais. Entretanto, trabalhos que discorrem sobre o monitoramento de *cloud network slice* no mesmo contexto que o apresentado neste artigo não foram encontrados. Ou seja, nenhum trabalho propõe uma arquitetura capaz de monitorar tanto os recursos físicos quanto virtuais de *slices* com múltiplos domínios administrativos e tecnológicos. A seguir, são descritos alguns dos trabalhos estudados para fomentar esta proposta.

No artigo [Ordonez-Lucena et al. 2017] os autores descrevem os principais aspectos de *network slicing*, com atenção especial em sua aplicação às redes 5G. O artigo ressalta como *network slicing* oferta uma solução para estabelecer simultaneamente um amplo escopo de serviços em uma infraestrutura comum de rede 5G, cobrindo a demanda de casos de uso específicos das indústrias verticais. Os autores apresentam uma breve visão geral da arquitetura SDN proposta pela ONF, que indica ferramentas para o suporte a *slices*. De acordo com os autores, apesar da falta de recursos essenciais que podem ser fornecidos por NFV, a arquitetura fornece meios para implementações de *network slicing*.

No artigo [Fatema et al. 2014], é apresentada uma visão geral do termo monitoramento com ênfase em ambientes na nuvem e *datacenters*. O objetivo do trabalho é comparar qualitativamente algumas das ferramentas de monitoramento mais utilizadas, considerando alguns pilares do monitoramento, como por exemplo: a escalabilidade, a portabilidade, o ambiente com múltiplos locatários e a habilidade de adaptação de acordo com o locatário. Estas características e outras citadas no artigo devem ser consideradas para melhor satisfazerem as necessidades dos locatários.

O artigo [Kuklinski and Tomaszewski 2018] apresenta uma extensão da arquitetura MANO para suportar *network slicing* no contexto da gerência e orquestração de funções de rede virtualizadas. O trabalho traz uma discussão sobre possíveis atualizações da arquitetura MANO para abranger o conceito de *slice*. No artigo os autores descrevem detalhadamente os componentes propostos e suas responsabilidades. Além disso, discorrem qualitativamente os pontos fortes da proposta apresentada. Porém, não é apresentada uma análise quantitativa ou implementação de uma prova de conceito da arquitetura.

Em [Tusa et al. 2018] os autores apresentam uma proposta de monitoramento inteligente utilizando a ferramenta Lattice². O escopo do monitoramento é voltado para sistemas de computação em nuvem, embora os autores mencionem que o Lattice pode ser estendido para suportar o conceito de *slice*. Em suma, eles monitoram um ambiente em computação em nuvem coletando algumas métricas e, ao detectar um aumento ou diminuição do tráfego, algumas taxas são ajustadas dinamicamente, como por exemplo o intervalo em que as métricas são coletadas e enviadas para o orquestrador. Entretanto, como o próprio trabalho destaca, os autores ainda não estenderam o trabalho para suportar o conceito de *slice*.

É importante fazer uma comparação entre os trabalhos descritos acima e o proposto neste artigo. O foco deste trabalho é apresentar uma arquitetura genérica que consiga fornecer as métricas de monitoramento referentes as *slices*, compostas por múltiplos domínios administrativos e/ou tecnológicos. A arquitetura irá coletar as métricas dos recursos individuais e agregar as métricas relacionadas a uma determinada *slice* como um todo. Além disso, será apresentado um modelo de informação preliminar para o monitoramento dos componentes que podem compor uma *cloud network slice*. O monitoramento poderá ser utilizado como uma visualização adaptada para cada locatário e/ou provedor de *slices*, bem como para realizar operações de gerência e orquestração dos recursos.

4. Proposta

Esta seção apresenta uma arquitetura e um modelo de informação para o monitoramento de *slices*, que levam em consideração multidomínios administrativos e tecnológicos. No

²<http://clayfour.ee.ucl.ac.uk/lattice/>.

contexto desse novo conceito de *slicing*, surge a necessidade de monitorar recursos de diferentes tipos em tais infraestruturas, como por exemplo, computacionais, de armazenamento e de rede. Portanto, esta proposta visa ampliar a discussão sobre este tema apresentando uma possível arquitetura e um modelo de informação minimalista para esta nova unidade de monitoramento, denominada *slice*.

Na Seção 2, a Figura 1 ilustra uma visão mais ampla de como seria o monitoramento de diferentes *slices* que estão alocadas em domínios administrativos distintos. Assume-se que o componente responsável pelo monitoramento faz parte do provedor de *slice*, que representa o domínio entrante para os três locatários ilustrados na figura. Os provedores de infraestrutura A, B e C possuem recursos instanciados que pertencem a *slices* de diferentes locatários. É responsabilidade do provedor de *slice* disponibilizar uma visualização de cada uma das *slices* para os seus respectivos locatários. Para isso, o componente de monitoramento deverá ser capaz de coletar e apresentar as informações de cada *slice* parte.

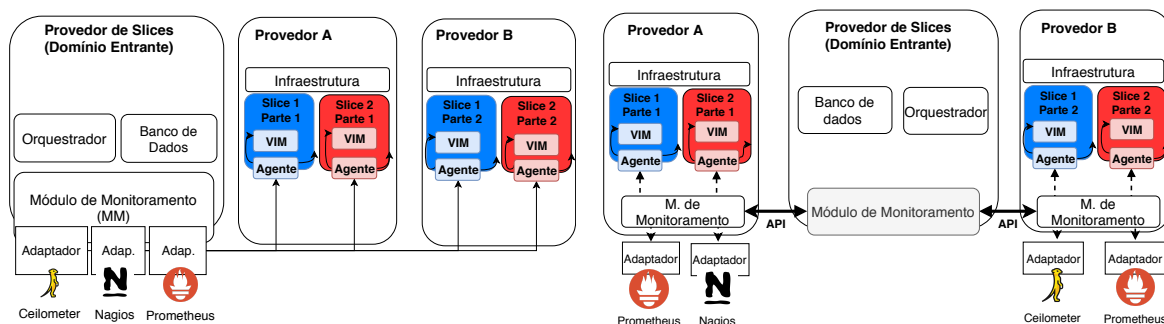
A tarefa de monitorar os recursos de uma *slice* pode ser considerada por diferentes perspectivas. A primeira é a perspectiva do locatário que, ao contratar uma *slice*, requer monitorar os recursos e/ou serviços que ele instanciou. Além do locatário, uma outra perspectiva em relação ao monitoramento é a do provedor de *slice*, que também precisa coletar métricas dos recursos alocados. Esta informação é utilizada para validar os contratos de utilização dos recursos previamente acordados, assim como, prover a orquestração inteligente dos recursos utilizando as informações providenciadas pelo componente de monitoramento. Considerando estas diferentes perspectivas baseadas nos interesses dos locatários e provedores, dois perfis podem ser elencados:

- **Perfil de monitoramento do locatário:** poderá requisitar informações relacionadas a recursos físicos, rede, armazenamento e serviços; baseadas no contrato definido no momento da instanciação de uma *slice*;
- **Perfil de monitoramento do provedor de *slice*:** deverá fornecer informações relacionadas aos recursos físicos, rede, armazenamento e serviços, instanciados por *slice*, que serão consumidas pelo locatário para as tarefas de orquestração.

4.1. Arquitetura

Este artigo busca apresentar duas possíveis arquiteturas que podem ser utilizadas com o objetivo de coletar métricas de diferentes tipos de recursos que estarão alocados em múltiplos domínios administrativos e tecnológicos. Como pode ser observado nas Figuras 2a e 2b, tal arquitetura possui como principais elementos: o Módulo de Monitoramento (MM), o Banco de Dados, o Orquestrador, os Adaptadores, os Agentes de Monitoramento (ou Ferramentas de Monitoramento) e os *Virtual Infrastructure Managers* (VIMs).

O MM é o principal componente da arquitetura, responsável pelo gerenciamento e monitoramento dos recursos alocados para uma determinada *slice*, de modo a disponibilizar informações sobre o uso dos recursos para o locatário e para o próprio provedor de *slices*. Para que o MM possa coletar as informações dos recursos e/ou serviços instanciados, ele deverá ser capaz de se comunicar com os diferentes provedores de infraestrutura que compõem a *slice*. Ao menos duas abordagens distintas podem ser implementadas para a comunicação do MM com os componentes de monitoramento. Estas abordagens representam a principal diferença entre as Figuras 2a e 2b e serão descritas a seguir.



(a) Opção 1: Comunicação MM domínio entrante e Agentes. (b) Opção 2: Comunicação MM domínio entrante e MM locais.

Figura 2. Arquiteturas para o monitoramento de *slices* multidomínios.

Na Figura 2a observa-se apenas um componente MM instanciado no domínio do provedor de *slices*. Este MM será capaz de coletar e agregar as informações das diferentes *slice* partes de forma direta. Ou seja, o MM do domínio entrante deverá se comunicar com os agentes de monitoramento instanciados, através da implementação de diferentes adaptadores. Na Figura 2b, além do MM no domínio entrante, será necessário instanciar um MM em cada provedor de infraestrutura. Desta forma, o MM no domínio entrante, ao invés de se comunicar diretamente com os agentes de monitoramento, utiliza-se das interfaces com os MMs instanciados nos diferentes provedores de infraestrutura que compõem a *slice*. Em outras palavras, o papel que era desempenhado pelo MM no domínio entrante de coletar as métricas diretamente com os agentes de monitoramento, passa a ser responsabilidade dos MMs locais em cada um dos provedores de infraestrutura.

A partir das duas abordagens de implementação mencionadas acima, pode-se ressaltar as principais diferenças entre as mesmas:

- Opção 1: Comunicação do MM domínio entrante com os Agentes (Figura 2a).
 - O número de requisições é menor;
 - A informação estará concentrada no domínio entrante;
 - A implementação dos adaptadores será realizada no domínio entrante, retirando tal responsabilidade dos provedores de infraestrutura.
- Opção 2: Comunicação do MM domínio entrante com outros MMs (Figura 2b).
 - O número de requisições e o isolamento das informações é maior;
 - O provedor de infraestrutura tem a liberdade de configurar a visualização, podendo adaptar as métricas coletadas em um nível de granularidade maior;
 - A implementação dos adaptadores não precisa ser genérica, ou seja, cada provedor de infraestrutura possui diferentes implementações;
 - A implementação de uma API para a comunicação entre os MMs é necessária.

É válido ressaltar que as duas abordagens não são necessariamente excludentes e a arquitetura é genérica o suficiente para suportar as duas abordagens de modo a se adequar melhor ao funcionamento de cada provedor. Por exemplo, é provável que um provedor de *slices* que não possui infraestrutura física própria opte por implementar uma comunicação entre MMs, uma vez que não possui adaptadores para VIMs.

O adaptador é o componente responsável por interagir com diferentes agentes de monitoramento e/ou VIMs. Os adaptadores se comunicam com o Módulo de Monitoramento para obter as métricas de uma determinada *slice* parte. Como exemplo, pode-se destacar a implementação de adaptadores para as seguintes tecnologias: OpenStack, Docker, Prometheus, OpenStack-Telemetry, VLSP, Lattice, Gnocchi, SNMP, Nagios, entre outros.

O agente de monitoramento é o componente responsável por coletar as métricas, tanto da infraestrutura física quanto dos serviços. Cada provedor de infraestrutura pode disponibilizar para o locatário diferentes agentes de monitoramento. Alguns exemplos de agentes de monitoramento para recursos físicos são: SNMP, sFlow, IPMI, Prometheus, Nagios, entre outros. Alguns exemplos para recursos virtuais são: OpenStack-Telemetry, Prometheus, entre outros.

Os VIMs são responsáveis por instanciar os recursos virtuais relacionados à computação, ao armazenamento ou à rede. Alguns exemplos de VIMs são: OpenStack, VLSP, Kubernetes/Docker, entre outros. O Orquestrador utiliza as métricas monitoradas pelo MM com o intuito de otimizar a operação dos recursos alocados, além de armazenar constantemente no banco de dados as métricas monitoradas por *slice*. As outras funções que este componente desempenha estão relacionadas ao gerenciamento do *slice* e não fazem parte do escopo deste trabalho. O banco de dados armazena as métricas dos recursos que compõem um determinado *slice*, para que o Orquestrador ou o locatário possam acessar as informações de monitoramento.

4.2. Modelo de Informação para o Monitoramento em Slices

Esta subseção apresenta um modelo de informação com ênfase no monitoramento das *slices*. A Figura 3 ilustra um diagrama UML que detalha tal modelo, que pode ser estendido de acordo com a necessidade dos provedores de *slices*. Primeiramente, uma *slice* é composta basicamente por três diferentes tipos de infraestruturas principais: computação, armazenamento e rede. Sendo assim, uma solução de monitoramento para *slices* deve contemplar estas três categorias. A proposta a seguir foi baseada no draft do IETF³ que aborda um modelo de informação para o conceito de *slice*, entretanto não possui como foco o monitoramento dos recursos desta nova entidade.

Discorrendo os relacionamentos presentes nesta entidade tem-se que, uma *slice* é composta por um ou mais domínios administrativos, onde cada domínio provê elementos de computação, armazenamento e/ou rede. Um domínio administrativo pode alocar um ou mais hosts, assim como elementos de rede. Os servidores físicos podem ser formados por nenhum ou vários elementos virtuais, que estão relacionados diretamente aos serviços, sejam eles instanciados em máquinas virtuais e/ou contêineres. Os elementos de rede, por sua vez, podem ser switches e/ou roteadores. Tanto os elementos de rede quanto as suas portas podem ser monitoradas pelo provedor de *slice* e/ou locatário. Um domínio pode conter um ou mais links nos quais estão associadas um conjunto de portas.

Os componentes descritos acima devem ser monitorados pelos provedores de modo a checar se os *KPIs* ou *SLAs* (do inglês, *Key Performance Indicators* e *Service Level Agreements*) acordados com o locatário serão cumpridos. Estes *KPIs* e *SLAs*, podem ser

³<https://tools.ietf.org/html/draft-qiand-coms-netslicing-information-model-02>

definidos para cada *slice* e estendidos para cada recurso que a compõem. Os exemplos de métricas a serem monitoradas por *slice* em cada componente podem ser visualizados na Figura 3. É importante salientar também que, caso seja necessário, tal modelo pode ser estendido de acordo com as requisições de cada locatário. Tanto a arquitetura de monitoramento quanto o modelo de informação descritos nesta seção, são propostas que visam ser genéricas o suficiente para que qualquer tipo de componente ou infraestrutura possam ser monitorados(as), considerando o conceito de *slices* multidomínios.

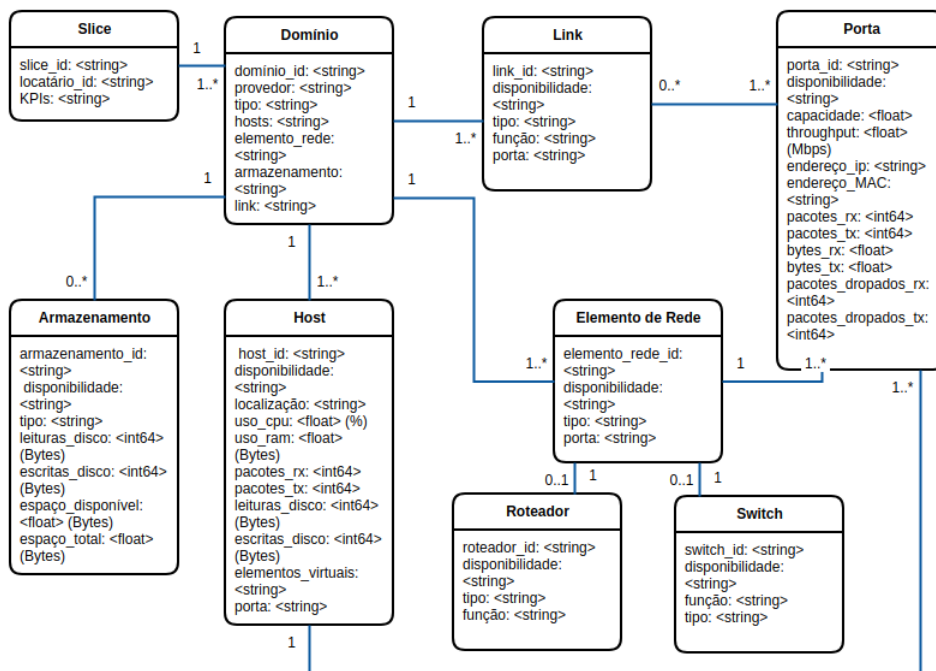


Figura 3. Modelo de informação para o monitoramento de *slices*.

5. Tecnologias Habilitadoras e Implementação

Esta seção expõe a implementação do MM e uma prova de conceito que busca exercitar tanto a arquitetura proposta quanto o modelo de informação. Para este trabalho, considera-se apenas o monitoramento dos elementos virtuais (máquinas virtuais e/ou contêineres) relacionados à infraestrutura de computação das *slices*. Inicialmente, optou-se pela arquitetura de comunicação representada na Figura 2a, assim como escolheu-se o perfil de monitoramento sob o ponto de vista do locatário (explicado na Seção 4). Em trabalhos futuros pretende-se implementar ambas as arquiteturas e analisar as diferenças das abordagens quantitativamente, o que neste trabalho foi exercitado qualitativamente.

5.1. Tecnologias Habilitadoras

Como mencionado anteriormente, uma *slice* pode ser formada por diferentes domínios, tanto administrativos como tecnológicos. Logo, para implementação de uma prova de conceito que represente tanto o monitoramento deste ambiente quanto o modelo de informação, decidiu-se por utilizar as seguintes tecnologias habilitadoras:

- O VIM utilizado será o Docker, responsável por instanciar contêineres;

- O Banco de dados responsável por armazenar as métricas coletadas foi construído em MySQL de acordo com o modelo de informação descrito na Subseção 4.2;
- O Agente de Monitoramento utilizado foi o Prometheus ⁴, que ficou responsável por coletar as métricas de monitoramento;
- O MM foi implementado na linguagem de programação Python e comunica-se com o Agente de monitoramento (Prometheus) via uma API REST.

O Prometheus é um framework capaz de monitorar diferentes tipos de infraestruturas e até mesmo aplicações por meio de elementos denominados *exporters*. Recentemente foi reconhecido pela Cloud Native Computing Foundation⁵ e vêm sendo amplamente estudado e aprimorado. De certa forma, o Prometheus é capaz de disponibilizar as informações de diferentes *exporters*, como por exemplo, um *exporter* para cada tipo de infraestrutura como por exemplo, elementos físicos, recursos virtuais e até mesmo métricas de aplicações. Tal funcionalidade é muito interessante quando relacionada ao conceito de *slices*, pois facilita a forma em que diferentes métricas podem ser agregadas por componentes, como por exemplo os Adaptadores que compõem a arquitetura proposta neste artigo.

5.2. Implementação

Esta subseção detalha a implementação do componente MM, bem como discorre o esquema utilizado no banco de dados. O esquema baseia-se no modelo de informação apresentado na Figura 3 e abrange apenas o monitoramento das métricas dos elementos virtuais que compõem uma *slice*.

O *Algoritmo 1* representa o pseudocódigo da implementação do componente MM. Na linha 1, observa-se os pré-requisitos para o início do algoritmo, que contempla um vetor contendo as *slices* que o MM irá monitorar e os agentes que o MM deverá se comunicar para obter as métricas. Então, da linha 4 a 6, uma nova *thread* é criada por *slice*, passando os agentes de monitoramento para o método *monitoreaSlice*. Este método é observado entre as linhas 9 e 13, que tem como função se comunicar com cada um dos agentes informados para obter as métricas de monitoramento através do método *coletaMetricas()* presente na linha 11. Este método recebe como parâmetro informações relacionadas ao agente de monitoramento. Após as métricas serem obtidas para cada agente, as mesmas são adicionadas no banco de dados.

Pelo fato deste trabalho considerar apenas a infraestrutura virtual relacionada ao elemento de computação de uma *slice*, o esquema do banco de dados criado engloba apenas os seguintes objetos apresentados na Figura 3: *slice*, domínio, servidores físicos e elementos virtuais. Portanto, no escopo atual do trabalho, as métricas relacionadas aos servidores físicos não são coletadas. Mais especificamente, as métricas coletadas dos elementos virtuais são: percentual de uso da CPU, quantidade de bytes utilizados de memória RAM, quantidade de bytes transmitidos e enviados pela rede, além da quantidade de bytes escritos no disco. As visualizações destas métricas são apresentadas na Seção 6.

6. Prova de Conceito e Resultados

Esta seção discorre sobre o ambiente de testes utilizado para validar a arquitetura proposta e sobre o modelo de informação. Esta prova de conceito está relacionada ao monitora-

⁴<https://prometheus.io/>

⁵<https://www.cncf.io/>

Algoritmo 1 Pseudocódigo - MM

```
1: Slices = {Slice, [Agentes]}
2: procedure MÓDULO DE MONITORAMENTO
3:   enquanto true faça
4:     para cada slice em Slices faça
5:       novaThread = monitoraSlice(Agentes)
6:     fim para
7:   fim enquanto
8: fim procedure
9: procedure MONITORASLICE(Agentes de monitoramento)
10:  para cada agente em Agentes faça
11:    metricas = coletaMetricas(agente)
12:    adicionaBanco(metricas)
13:  fim para
14: fim procedure
```

mento de *slices* previamente instanciadas, através de multidomínios administrativos.

6.1. Prova de Conceito

Na Figura 4, observa-se a infraestrutura utilizada que consiste em duas *slices* instanciadas ao longo de dois provedores de infraestrutura, denominados *Provedor A* e *Provedor B*. Cada *slice* parte pertencente aos provedores de infraestrutura possui um servidor físico dedicado ilustrado por $SF\#$ ($SF1 - SF4$). Em cada servidor $SF\#$ há um contêiner representado por $C\#$ e um agente de monitoramento representado por $A\#$. O agente de monitoramento para todos os quatro $C\#$ é o Prometheus. O provedor de *slice* está representado mais à direita na Figura 4. A figura ilustra também os componentes: Banco de Dados, MM e Orquestrador. Entretanto, para os testes realizados, apenas os dois primeiros foram implementados e fazem parte do escopo deste trabalho.

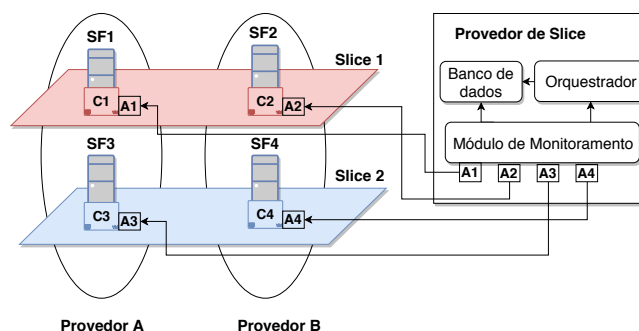


Figura 4. Ambiente de Testes.

De acordo com a arquitetura proposta, para cada agente ($A\#$) contido nos provedores de infraestrutura existe um adaptador implementado no MM. Contudo, como o framework de monitoramento é o mesmo (Prometheus) para todos os agentes na implementação da prova de conceito, foi implementado um adaptador no MM capaz de coletar as métricas dos diferentes agentes Prometheus. Em suma, o ambiente de testes consiste de duas *slices* instanciadas ao longo de dois provedores de infraestrutura (*Provedor A* e *Provedor B*), contendo os seguintes componentes por *slice*: dois servidores físicos ($SF\#$), dois contêineres ($C\#$) e dois agentes de monitoramento ($A\#$). Externamente aos provedores de infraestrutura, o *Provedor de Slice* executa o MM, que é responsável por monitorar e armazenar informações de cada uma das *slices*.

Para o cálculo das métricas relacionadas às *slices*, obteve-se a média aritmética dos valores individuais⁶. Mais especificamente, a cada intervalo de tempo $\Delta t = 10s$, foram coletadas as métricas dos contêineres pertencentes a uma *slice* e calculada a média das métricas em cada intervalo de tempo. Para cada um dos contêineres, simulou-se uma carga de trabalho adicional com o intuito de aumentar o uso dos recursos (CPU, memória, rede e disco). Para aumentar a carga de trabalho na CPU, memória e disco, utilizou-se o programa *stress-ng* disponível no Linux. Com relação à rede, gerou-se um tráfego de comunicação entre os contêineres e um servidor a partir do programa *iperf*⁷. Além disso, três perfis distintos de consumo dos recursos foram considerados: baixo, médio e alto. Nos testes realizados, estes perfis foram simulados a partir de uma variação sequencial a cada 10 minutos, resultando em testes de 30 minutos. Então, para cada uma das duas *slices* foram iniciados dois contêineres, onde um seguiu uma ordem crescente de perfil de consumo (baixo, médio e alto) e o outro a ordem inversa (alto, médio e baixo).

6.2. Resultados

Os resultados expostos a seguir têm o intuito de ilustrar algumas das métricas descritas acima, monitoradas ao longo do tempo. Vale ressaltar que os gráficos e visões apresentados nesta seção representam parte de um subconjunto do que poderia ser explorado com relação às métricas monitoradas atualmente. Por exemplo, optou-se por exibir as métricas por *slices* e também por recursos virtuais relacionados à uma determinada *slice*. Entretanto, poderia haver outras formas de exibir as métricas coletadas, uma vez que a arquitetura em conjunto com o modelo de informação proposto proporcionam o isolamento das métricas monitoradas por *slice*.

Na Figura 5, dois gráficos ilustram as métricas coletadas no ambiente de testes descrito na Subseção 6.1. Ambos os gráficos apresentam as métricas monitoradas (eixo y) ao longo do tempo (eixo x). A Figura 5a mostra o uso médio de CPU dos recursos virtuais para as *Slices* 1 e 2. Pela Figura 5b, pode-se observar um detalhamento nos valores de monitoramento de cada um dos recursos virtuais específicos para a *Slice* 1. Como dito anteriormente, nos experimentos executados, o *C1* iniciou com um perfil de carga de trabalho definido como baixo, enquanto o *C2* iniciou com o perfil alto.

A análise dos gráficos na Figura 6 é análoga a descrita no parágrafo anterior, onde o gráfico mais à esquerda (Figura 6a) ilustra a quantidade de bytes transmitidos pelos contêineres que compõem a *Slice* 2. O gráfico na Figura 6b detalha o consumo de memória RAM dos contêineres que fazem parte da *Slice* 1. Como este ainda é um trabalho preliminar, o foco dos resultados apresentados é apenas ilustrar alguns componentes de uma *slice* sendo monitorados ao longo do tempo.

A partir das Figuras 5 e 6, é possível observar a variação dos perfis de carga de trabalho no comportamento de consumo dos contêineres instanciados. Nota-se, por exemplo, que no primeiro terço do tempo de coleta um dos contêineres apresenta um alto perfil de carga de trabalho, enquanto o outro simula uma carga baixa de trabalho. Como esperado, este comportamento se inverte no terceiro terço do tempo de coleta, enquanto no segundo terço o comportamento dos dois é similar com uma carga média de trabalho.

⁶Tais métricas têm o intuito apenas de ilustrar o monitoramento dos recursos, pois a informação da média de uso de um recurso de uma *slice* pode não agregar tanta informação para possíveis ações de orquestração.

⁷<https://iperf.fr/>

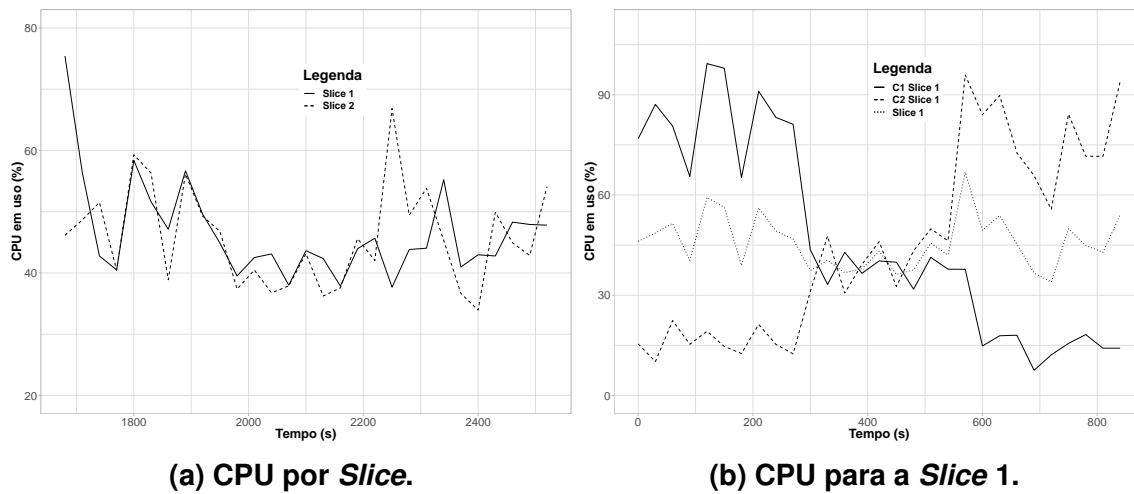


Figura 5. Métricas relacionadas à CPU (%).

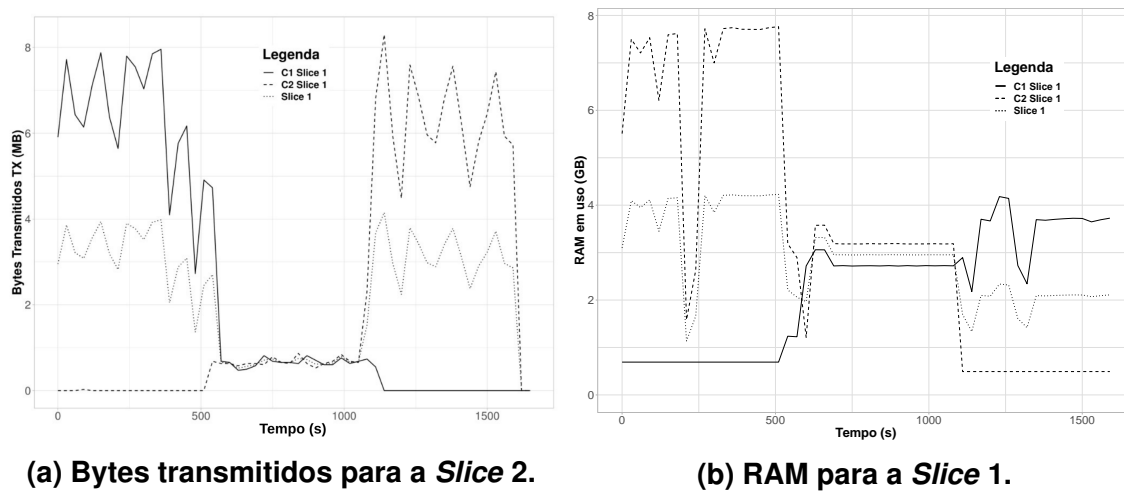


Figura 6. Métricas relacionadas a Rede 6a e a Memória 6b.

7. Conclusões e Trabalhos Futuros

O conceito de *cloud network slicing* traz novos desafios que precisam ser investigados em relação ao monitoramento dessa nova entidade. Coletar informações dos recursos fim-a-fim através de múltiplos domínios administrativos e tecnológicos é de suma importância para aprimorar operações de gerência e orquestração.

Neste contexto, este artigo propõe uma arquitetura para o monitoramento de *slices* fim-a-fim, considerando o cenário de multidomínios administrativos e/ou tecnológicos. Outra contribuição importante é a proposta de um modelo de informação para o monitoramento de *slices*, que envolve diferentes infraestruturas. Por fim, pode-se destacar a implementação de uma prova de conceito baseada na arquitetura proposta. A partir desta prova de conceito foi possível visualizar métricas relacionadas aos recursos virtuais que haviam sido instanciados, bem como as métricas por *slice* de forma isolada.

A extensão imediata deste trabalho envolve avaliar quantitativamente a escalabilidade da arquitetura proposta, aumentando o número de métricas, *slices*, recursos, dentre outros. Pretende-se também adicionar adaptadores para outras infraestruturas e/ou agen-

tes de monitoramento, como por exemplo Kubernetes e OpenStack. Por fim, planeja-se aproveitar a arquitetura proposta para servir de base para a implementação do componente responsável pela gerência e orquestração dos recursos alocados, levando em consideração as métricas monitoradas.

Agradecimentos

Este trabalho é financiado pela chamada conjunta Europa-Brasil H2020 número 777067, no contexto do Projeto NECOS (*Novel Enablers for Cloud Slicing*), co-financiado pela União Europeia e pelo Ministério de Ciência, Tecnologia, Inovação e Comunicação (MCTIC), através do Centro de Pesquisa e Desenvolvimento em Tecnologias Digitais para Informação e Comunicação (CTIC) da Rede Nacional de Pesquisa (RNP).

Referências

- Clayman, S. (2018). D3.1: NECOS System Architecture and Platform Specification. V1. Technical report.
- de Foy, X. and Rahman, A. (2017). Network Slicing - 3GPP Use Case. Internet-Draft draft-defoy-netslices-3gpp-network-slicing-02, IETF Secretariat.
- ETSI ISG NFV (2018). GR NFV-IFA 022 - V3.1.1 - Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Report on Management and Connectivity for Multi-Site Services. Technical report.
- Fatema, K., Emeakaroha, V. C., Healy, P. D., Morrison, J. P., and Lynn, T. (2014). A survey of cloud monitoring tools: Taxonomy, capabilities and objectives. *Journal of Parallel and Distributed Computing*, 74(10):2918 – 2933.
- Kuklinski, S. and Tomaszewski, L. (2018). Dasmo: A scalable approach to network slices management and orchestration. *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–6.
- NGMN Alliance (2018). 5G End-to-End Architecture Framework v2.0. Technical report.
- Ordonez-Lucena, J., Ameigeiras, P., Lopez, D., Ramos-Munoz, J. J., Lorca, J., and Folgueira, J. (2017). Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges. *IEEE Communications Magazine*, 55(5):80–87.
- Qiang, L., Galis, A., Geng, L., Makhijani, K., Martinez-Julia, P., Flinck, H., and de Foy, X. (2018). Technology Independent Information Model for Network Slicing. Internet-Draft draft-qiang-coms-netslicing-information-model-02, Internet Engineering Task Force. Work in Progress.
- TR-521, O. (2016). SDN Architecture. Technical report, Open Networking Foundation.
- Tusa, F., Clayman, S., and Galis, A. (2018). RealTime Management and Control of Monitoring Elements In Dynamic Cloud Network Systems. In *2018 IEEE 7th International Conference on Cloud Networking (CloudNet)*, pages 1–7. IEEE.