

Destravando o *Element Management System*: Permitindo a Gerência Holística de Funções de Rede Virtualizadas

Vinicius Fulber-Garcia, José Flauzino, Elias P. Duarte Jr.

Universidade Federal do Paraná (UFPR)
Caixa Postal 19018 81531-990, Curitiba - PR

{vfgarcia, jwvflauzino, elias}@inf.ufpr.br

Abstract. *The development of efficient management solutions is a fundamental step towards the broad adoption of Network Function Virtualization (NFV) technology. The Element Management System (EMS) is a key component of the NFV reference management architecture defined by the European Telecommunications Standards Institute – ETSI. The EMS actuates directly on the Virtualized Network Functions (VNF), executing operations issued by the VNF Manager (VNFM) and Operation and Business Support Systems (OSS/BSS). In recent works however, the EMS has been either (i) completely ignored or (ii) employed in limited settings, typically for executing simple operations in the context of a particular application. In this work, we propose a comprehensive EMS architecture that is compliant with the NFV reference architecture. This architecture enables the creation of a holistic EMS to manage VNF instances while abstracting communication protocols and technologies, allowing the interoperability among different VNF platforms and OSS/BSS. A prototype was implemented, and evaluation results are presented for the overhead of including the EMS in the execution of typical management operations.*

Resumo. *O gerenciamento eficiente de funções de rede virtualizadas é um desafio importante para a adoção em larga escala da tecnologia Network Function Virtualization (NFV). O Element Management System (EMS) é um componente da arquitetura de referência NFV que atua no gerenciamento de instâncias de Virtualized Network Functions (VNF), executando operações requeridas pelo VNF Manager (VNFM) e Operation and Business Support Systems (OSS/BSS). Porém, atualmente, ou (i) o EMS é completamente ignorado, ou (ii) é usado de forma limitada, permitindo a execução de operações simples no contexto de uma única aplicação. Neste trabalho, uma arquitetura EMS abrangente e compatível com a arquitetura de referência NFV é proposta. Essa arquitetura permite ao EMS executar um gerenciamento holístico das instâncias de VNF, abstraindo protocolos e tecnologias de comunicação e provendo interoperabilidade entre as diferentes plataformas de execução de VNF, VNFM e OSS/BSS. Um protótipo da arquitetura foi implementado e avaliado pela sobrecarga imposta à execução de operações típicas do gerenciamento de funções de rede virtualizadas.*

1. Introdução

A virtualização de redes apresenta-se como uma alternativa para expandir, flexibilizar, e reduzir custos operacionais e de capital da infraestrutura de rede, tradicionalmente baseada em equipamentos dedicados para a execução de suas funções e serviços. Entre os paradigmas que seguem essa tendência está a Virtualização de Funções de Rede (*Network*

Function Virtualization - NFV). Esse paradigma visa desacoplar as funções e serviços de rede de seus equipamentos dedicados, os implementando em um plano de *software* através de tecnologias de virtualização existentes, como virtualização completa, para-virtualização e containerização. Através do paradigma NFV objetiva-se programar redes personalizadas capazes de atender políticas e requisitos específicos, atingir alta mobilidade facilitando processos de migração, além de descentralizar e expandir o mercado de desenvolvimento de funções de rede, entre outros benefícios.

Recentemente, diversos esforços têm sido feitos no sentido de padronizar o paradigma NFV. Dentre esses esforços destaca-se a definição de uma arquitetura de referência pela *European Telecommunications Standards Institute* (ETSI) [ISG 2014]. Essa arquitetura é formada por três domínios de trabalho chamados de Infraestrutura Virtualizada (*Virtualized Infrastructure* - VI), Funções de Rede Virtualizadas (*Virtualized Network Functions* - VNF) e Gerenciamento e Orquestração de NFV (*NFV Management and Orchestration* - NFV-MANO). É importante ressaltar que, do ponto de vista de gerenciamento, diversos componentes operacionais são previstos. Os componentes de gerência que recebem maior destaque são o Gerente de Infraestrutura Virtualizada (*Virtualized Infrastructure Manager* - VIM), Gerente de VNF (*VNF Manager* - VNFM), Orquestrador NFV (*NFV Orchestrator* - NFVO) e o Sistema de Gerenciamento de Elementos (*Element Management System* - EMS).

De maneira geral, os componentes de gerência NFV são amplamente explorados na literatura. Porém, o EMS não é particularmente abordado em trabalhos recentes. A utilização desse componente é tipicamente isolada e orientada a aplicações específicas, gerando um conjunto de implementações heterogêneas, e por vezes conflitantes, do mesmo. O uso específico do EMS ocorre, por exemplo, na alocação de recursos em redes 5G [Hawilo et al. 2014], implementação de sistemas multimídias em NFV [Carella et al. 2014] e aceleração de funções de rede [Sharma et al. 2019]. Além disso, soluções de EMS não são desenvolvidas considerando uma arquitetura de referência. Esse cenário pode levar a dificuldades, ou até mesmo a impossibilidade, de integração entre EMS e NFV-MANO (VNFM), assim como com Sistemas de Suporte a Operações e Negócios (*Operation and Business Support System* - OSS/BSS). Esse problema é agravado visto que virtualmente não existem trabalhos baseados na interface protocolar de comunicação entre EMS e VNFM (Ve-Vnfm-em [IFA 2020]), nem modelos holísticos e padronizados de comunicação entre EMS e diferentes plataformas de execução de VNF.

Considerando o cenário apresentado, o presente trabalho propõe uma arquitetura interna para o EMS. Essa arquitetura é projetada de forma abrangente e extensível, permitindo o desenvolvimento de diferentes tipos de soluções de EMS, sejam elas de uso genérico ou orientadas a aplicações específicas. Além disso, os módulos operacionais internos e interfaces de comunicação definidos na arquitetura EMS são totalmente compatíveis com a arquitetura de referência NFV. Um protótipo foi desenvolvido como prova de conceito da arquitetura proposta. Esse protótipo foi experimentado em diferentes cenários de forma a validar sua capacidade em exercer diversas operações esperadas de um EMS. Resumidamente, as contribuições deste trabalho consistem de: (i) proposta de uma arquitetura EMS abrangente e extensível; e (ii) implementação e disponibilização de uma solução de EMS genérica que segue a arquitetura definida.

O restante do trabalho está organizado como segue. A Seção 2 apresenta conceitos básicos relacionados ao paradigma NFV, detalhando, em particular, o componente

operacional de EMS. Também na Seção 2 são apresentados os principais trabalhos relacionados. A Seção 3 descreve a arquitetura EMS proposta e discute seus módulos internos e interfaces de comunicação. Na Seção 4 são apresentados detalhes de implementação do protótipo desenvolvido. Os cenários de experimentação e os resultados obtidos são discutidos na Seção 5. Finalmente, a Seção 6 conclui o trabalho.

2. O EMS: Sistema de Gerenciamento de Elemento

A arquitetura de referência NFV [ISG 2014] prevê múltiplos componentes operacionais para o gerenciamento das infraestruturas de rede virtualizadas a partir de diferentes escopos de execução, denominados domínios de trabalho. Entre esses componentes destacam-se o NFVO, VNFM e VIM presentes no domínio de trabalho de NFV-MANO; o EMS localizado no domínio de trabalho de VNF; e o OSS/BSS alocado externamente aos domínios de trabalho NFV. De forma secundária, é possível destacar os agentes de gerenciamento presentes nas plataformas de execução de VNF (domínio de trabalho de VNF), utilizados para a execução de operações do ciclo de vida das funções de rede virtualizadas. Os componentes operacionais de gerenciamento comunicam-se entre si para realizar a manutenção de ambientes NFV, sendo estes responsáveis por tarefas como interfaceamento com os operadores de rede, coordenação de funções e serviços, monitoramento de métricas de desempenho, atualização de configurações e recuperação de falhas.

Em particular, o EMS aplica operações de FCAPS (*Fault, Configuration, Accountability, Performance and Security*) internamente em instâncias de VNF [IFA 2020]. Esse componente é fortemente relacionado ao VNFM e aos módulos de gerenciamento presentes nas plataformas de execução de VNF, conhecidos como Agentes de Gerência (*Management Agent - MA* [Fulber-Garcia et al. 2019a]). Apesar de apresentarem similaridades quanto às operações que executam, VNFM, EMS e MA trabalham em níveis de abstração diferentes e cooperam entre si para obterem sucesso na execução de suas tarefas de gerenciamento. A principal atribuição exclusiva do EMS em relação aos demais componentes é a de interfaceamento entre VNFM e MA. Ou seja, esse componente deve comunicar-se com o VNFM através de um protocolo padrão definido pela ETSI [SOL 2020] e, com ele, abstrair a heterogeneidade operacional do MA, que pode ter variações dependendo, por exemplo, da função de rede executada (*e.g.*, firewall, inspetores e roteadores), do sistema hospedeiro utilizado (*e.g.*, contêiner, *unikernel*, *cloud* e servidor) e das interfaces disponibilizadas (*e.g.*, SSH, HTTP e socket).

Um EMS pode ser categorizado de diferentes formas considerando seu modo de operação e abrangência de atuação no ambiente NFV e, em especial, no domínio de VNF. A primeira categoria de EMS apresenta considerações em relação a **integração** deste à arquitetura de referência NFV. Nesse caso, duas categorias são possíveis: MANO e não-MANO [IFA 2016]. Um EMS MANO é consciente do ambiente NFV do qual faz parte, nativamente cooperando com o domínio de NFV-MANO através do VNFM. Por outro lado, um EMS não-MANO pode ser compreendido como qualquer sistema de gerenciamento que atue sobre instâncias de VNF, sendo que o mesmo não permite, não reconhece ou não diferencia interações realizadas com o VNFM. Ainda, um EMS MANO pode ser classificado considerando sua **implantação** no ambiente NFV: VNFC, VNF e Independente [IFA 2016, IFA 2018]. A classe de EMS MANO VNFC é executada internamente a uma instância de VNF, sendo reconhecida como um componente interno da mesma e estando subordinada ao ciclo de vida de sua respectiva função de rede. Um EMS da classe MANO VNF é implantado externamente às instâncias de VNF gerenci-

adas por ele, porém é interpretado e controlado pelos demais componentes operacionais da arquitetura NFV como uma instância de VNF. Por último, um EMS MANO Independente é integrado ao ambiente NFV, mas não está sob o controle de nenhum componente operacional do mesmo, tendo um nível de independência similar ao OSS/BSS.

Quanto à **localização** do EMS em relação às instâncias de VNF gerenciadas por ele, duas categorias são definidas: *Interno* e *Externo* [IFA 2018]. Um EMS da categoria *Interno* é contido em uma instância de VNF. Dessa forma, o EMS pode substituir qualquer agente de gerência da plataforma de execução de VNF, rodando diretamente no sistema operacional da mesma. Opcionalmente, um EMS *Interno* também pode ser implantado como um VNFC (*i.e.*, EMS MANO VNFC). Por outro lado, a categoria *Externo* determina a independência operacional do EMS das instâncias de VNF gerenciadas, realizando suas tarefas externamente a elas. Dessa forma, um EMS pode ser implantado tanto como um sistema independente, quanto como uma instância de VNF subordinada ao NFV-MANO.

Atentando ao **gerenciamento** das instâncias de VNF possibilitado pelo EMS, outras duas categorias são aplicáveis: *Genérico* e *Restrito* [IFA 2018]. Um EMS pertencente à categoria *Genérico* fornece apenas um conjunto básico de operações de gerenciamento (definido em [IFA 2020]) e uma interface de comunicação que permite o acesso às mesmas. Já um EMS *Restrito*, além de responder pelos mesmos requisitos de um EMS genérico, ainda provê operações restritas de um ou mais tipos de plataformas de execução de VNF. Dado esse cenário, não há garantias de funcionamento de um EMS restrito (em particular de suas operações não protocolares) para plataformas de execução de VNF diferentes daquelas presentes no conjunto de projeto do mesmo. Por fim, duas categorias existem quanto à **abrangência** (*i.e.*, número de instâncias de VNF) de gerenciamento de um EMS: *Dedicado* e *Guarda-chuva* [ISG 2014, IFA 2018, REL 2015]. Um EMS dedicado é limitado ao gerenciamento de uma única instância de VNF. Ao contrário, um EMS *Guarda-chuva* pode gerenciar múltiplas instâncias de VNF em simultâneo.

Trabalhos Relacionados: O EMS nas Principais Plataformas de NFV-MANO

Apesar do componente operacional de EMS pertencer ao domínio de trabalho de VNF da arquitetura de referência NFV, sua implementação está tipicamente associada a projetos de NFV-MANO, como OpenStack Tacker [OpenStack 2020], Open Source MANO (OSM) [ETSI 2020], Open Baton [OpenBaton 2020] e CloudStack Vines [Flauzino et al. 2020]. De maneira prática, esse fenômeno ocorre devido à existência de protocolos de comunicação específicos, projetados para cada plataforma de NFV-MANO. Assim, esses mesmos protocolos são utilizados para o estabelecimento da comunicação entre VNFM e EMS. Também, é comum que sejam empregadas operações das próprias plataformas de NFV-MANO no desenvolvimento das operações de EMS. A seguir, essas soluções de EMS são discutidas em maiores detalhes.

O OpenStack Tacker [OpenStack 2020] é uma plataforma de NFV-MANO que conta com VNFM e NFVO nativos, além de utilizar o OpenStack como seu VIM/VI. O OpenStack Tacker não disponibiliza um EMS como parte de sua plataforma. Entretanto, para possibilitar a execução de operações de configuração inicial interna nas funções de rede, o Tacker suporta o envio de arquivos do tipo *Shell Script* para plataformas de execução de VNF que implementam um sistema chamado Cloud-Init. O Cloud-Init não pode ser considerado um EMS completo, mas sim um agente de configuração básico que executa *scripts* que, de forma independente, realizam operações na instância de VNF.

Feitas essas observações e considerando apenas as suas características de configuração, o Cloud-Init é categorizado como um EMS **não-MANO, interno e dedicado**.

O OSM é um projeto gerido pela ETSI que oferece uma plataforma de NFV-MANO através de contêineres [ETSI 2020]. Essa plataforma permite a utilização tanto de nuvens OpenStack quanto de contêineres Kubernetes como seu VIM/VI. O OSM não dispõe de um EMS nativo e oferece apenas um agente de configuração interna através de seu sistema *day-like*. A execução desse agente de configuração depende de ferramentas pré-instaladas nas plataformas de execução de VNF. Assim, aplicando as mesmas limitações consideradas para o Cloud-Init, o sistema *day-like* é categorizado como um EMS **não-MANO, interno e dedicado**.

O Open Baton [OpenBaton 2020] é uma plataforma de NFV-MANO com VNFM, NFVO e suporte ao OpenStack como seu VIM/VI. Além disso, o Open Baton permite a implementação de *drivers* para habilitar o uso de outras plataformas de nuvem como seu VIM/VI. Essa plataforma de NFV-MANO possui um EMS que deve ser executado internamente a plataformas de execução de VNF. Tal EMS é programável, permitindo definir operações de gerência interna e disponibilizá-las através de interfaces REST (*Representational State Transfer*) ou AMQP (*Advanced Message Queuing Protocol*). Diferente do OpenStack Tacker e OSM, o EMS do Open Baton mantém-se disponível após a fase de configuração da instância de VNF. Dessa forma, esse EMS é categorizado como **MANO (independente), interno, restrito e dedicado**.

A plataforma de NFV-MANO CloudStack Vines provê um VNFM e um NFVO que atuam com uma nuvem CloudStack (VIM/VI) [Flauzino et al. 2020]. Essa plataforma disponibiliza naturalmente um EMS implementado como um elemento de rede do CloudStack. Esse EMS pode comunicar e gerenciar múltiplas instâncias de funções de rede que utilizam a plataforma de execução de VNF Leaf (proposta no contexto do Vines). Além disso, esse EMS pode ser adaptado, através da implementação de *drivers*, para gerenciar outras plataformas de execução de VNF. Sendo assim, o EMS do CloudStack Vines é categorizado como **MANO (VNF), externo, restrito e guarda-chuva**. Observe-se, entretanto, que ambos os EMS do Open Baton e CloudStack Vines não implementam o protocolo de comunicação com o VNFM especificado pela ETSI (*i.e.*, [SOL 2020]).

3. Uma Arquitetura Abrangente para EMS

Atualmente, não existe nenhuma arquitetura de referência para a estruturação e desenvolvimento do componente de EMS. Dessa forma, as características de integração, localização, gerenciamento e abrangência são definidas nos EMS existentes como uma consequência de sua utilização conjunta a uma plataforma de NFV-MANO específica, e não como uma finalidade de projeto do próprio EMS. A existência de soluções de EMS flexíveis e abrangentes é uma necessidade primordial para atender a diversos fundamentos do paradigma NFV (*e.g.*, integração e portabilidade), além destas funcionarem como ferramentas essenciais no gerenciamento de redes modernas baseadas em *software*. Grande parte da importância do EMS vem da sua competência em estabelecer e interfacear a comunicação entre diferentes componentes operacionais de um ambiente NFV (*i.e.*, VNF, VNFM, OSS e BSS). Assim, o EMS é relacionado à interoperabilidade por definição, abstraindo o gerenciamento do domínio de VNF independentemente das tecnologias e fornecedores dos componentes envolvidos no processo.

Neste trabalho, uma arquitetura genérica, abrangente e expansível para o com-

ponente de EMS é proposta. A arquitetura, ilustrada na Figura 1, é constituída por seis módulos fundamentais: (i) Base de Informações de Gerência de EMS (EMIB), (ii) Sub-sistema de Acesso, (iii) Roteador Interno, (iv) Subsistema de Monitoramento, (v) Subsistema de VNF e (vi) Módulo de Configuração. Cada módulo é responsável por executar operações específicas de tratamento, encaminhamento e/ou aplicação de requisições, sejam estas referentes a instâncias de VNF gerenciadas pelo EMS (plano de dados, indicado por setas com linhas contínuas) ou a configuração do próprio componente de EMS (plano de gerência, estabelecido pelas setas com linhas tracejadas). Um terceiro plano, de informações, trata do acesso e manutenção de configurações que definem o modo de operação de um EMS instanciado. Essas configurações são armazenadas na EMIB, sendo o acesso e manipulação das mesmas realizadas exclusivamente por módulos internos da arquitetura EMS (linhas pontilhadas). As interfaces de comunicação do EMS na arquitetura de referência do paradigma NFV são demonstradas por linhas duplas contínuas. Interfaces fora do escopo deste trabalho são representadas por linhas duplas tracejadas.

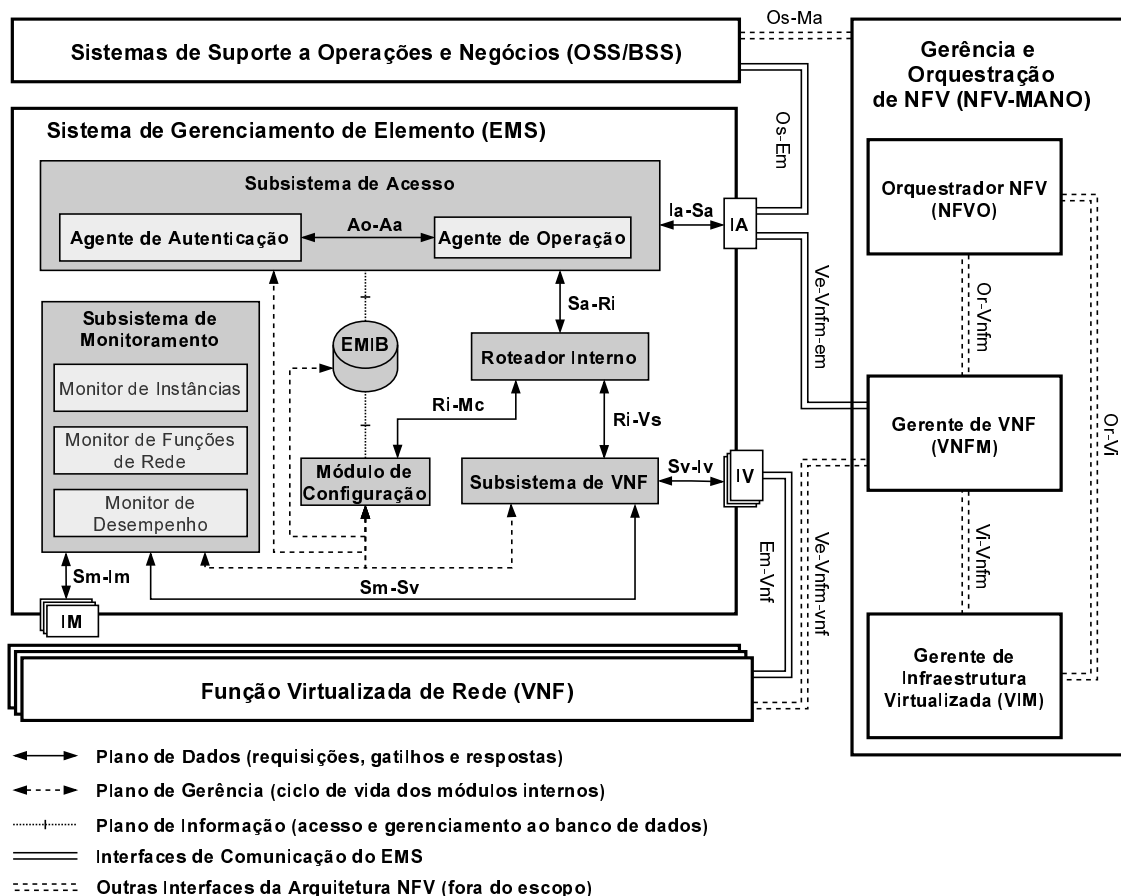


Figura 1. Arquitetura do Componente Operacional de EMS

Todos os módulos da arquitetura EMS foram projetados para serem pouco acoplados entre si, sendo que o funcionamento de uma solução e a cooperação entre módulos são garantidos a partir da programação de pontos de acesso definidos. Assim, módulos individuais podem ser substituídos de maneira natural de acordo com a evolução e surgimento de novas tecnologias e ferramentas. As descrições de competências e dos relacionamentos dos módulos da arquitetura de EMS proposta são apresentados a seguir.

Base de Informações de Gerência de EMS (EMIB) – este módulo consiste em um ponto de armazenamento de informações relacionadas à configuração e administração de um EMS. A EMIB pode ser programada como um simples sistema de pastas e arquivos, ou mesmo como um banco de dados complexo. A escolha do modelo de implementação deste módulo deve considerar a complexidade e quantidade de informações mantidas pelo EMS pretendido. Exemplos de informações presentes na EMIB são: dados e credenciais de usuários do EMS; descrição, localização e modelos de acesso das instâncias de VNF gerenciadas; agentes e políticas de monitoramento executados continuamente; e informações de suporte a comunicação e compatibilidade com plataformas de outros componentes da arquitetura NFV (*e.g.*, VNF, VNFM e OSS/BSS).

Subsistema de Acesso (SA) – uma vez enviadas requisições para um EMS, o SA é o módulo responsável pelo recebimento, validação, autenticação e encaminhamento das mesmas para o Roteador Interno. Do mesmo modo, após processada uma requisição, o SA deve encaminhar o seu resultado para o requerente. Este módulo contempla dois agentes: Agente de Operação (AO) e Agente de Autenticação (AA). O AO define e hospeda a Interface de Acesso (IA) do EMS. É no AO que estão disponibilizadas todas as operações de acesso externo, sejam estas relacionadas ao gerenciamento de instâncias de VNF ou a administração do próprio EMS. Sendo assim, minimamente, o AO deve prover acesso as operações protocolares da interface Ve-Vnfm-em [IFA 2020]. O AA, no que lhe concerne, verifica se as requisições recebidas são oriundas de usuários válidos e com privilégios suficientes para as realizarem. Esse agente, em particular, é de utilização opcional, porém é recomendado em documentos de especificação que abordam o EMS [SEC 2019, SEC 2020]. Quando utilizado, a autenticação do AA deve ocorrer após a validação de uma requisição pelo AO e antes do envio desta para o Roteador Interno.

Roteador Interno (RI) – o RI direciona requisições (recebidas do SA) internamente ao EMS. Ou seja, requisições por operações de gerenciamento de instâncias de VNF são enviadas para o Subsistema de VNF. Ainda, requisições manutenção do próprio EMS são repassadas ao Módulo de Configuração. O RI deve ser capaz de rastrear a totalidade do fluxo de encaminhamento de uma requisição no EMS, ligando a ele os seus resultados e eventuais exceções e erros. Ao final, os resultados do processamento de uma requisição são retornadas ao SA para envio ao seu respectivo requerente.

Subsistema de Monitoramento (SM) – este módulo oferece um ambiente de execução para monitores de instâncias de VNF. Em outras palavras, usuários do EMS podem estabelecer, a partir das operações disponíveis no Subsistema de VNF, agentes de monitoramento que são executados continuamente. Esses agentes devem prover suporte ao desenvolvimento de gatilhos, de forma a possibilitar o envio de notificações, alertas e relatórios para destinatários específicos quando eventos de interesse são identificados. O envio dessas mensagens é realizado por Interfaces de Monitoramento (IM) próprias dos agentes, evitando assim a sobrecarga do SA. Apesar de, *a priori*, um agente de monitoramento ter acesso a qualquer operação disponibilizada no Subsistema de VNF, aquelas mais comumente utilizadas estão relacionadas ao monitoramento da instância virtualizada (*e.g.*, envio de *heartbeats*) e da função de rede sendo executada (*e.g.*, filtragens e inspeções), além de métricas genéricas de desempenho (*e.g.*, atraso e *jitter*).

Subsistema de VNF (SV) – o SV é um dos principais módulos da arquitetura proposta. É a partir dele que se estabelece a comunicação entre o EMS e um ou mais tipos de plataformas de execução de VNF. Este módulo deve reconhecer previamente o

conjunto de operações de VNF possíveis, além dos procedimentos de execução e modelos de resultados de cada uma delas. Assim, mais do que aplicar uma operação em uma instância de VNF, o EMS pode interpretar o seu resultado e responder adequadamente ao requerente. Toda a comunicação descrita ocorre através de Interfaces de VNF (IV). A gerência dessas interfaces varia de acordo com o projeto e implementação do EMS, mas a existência de múltiplos IV é natural. Exemplos de cenários onde vários IV podem ser usados são: definição de um IV para cada tipo de plataforma de execução de VNF suportada pelo EMS e criação de um IV exclusivo para atender um agente de monitoramento do SM (evitando a sobrecarga das interfaces de uso comum).

Módulo de Configuração (MC) – o objetivo primário do MC é estabelecer as configurações de um EMS (armazenadas na EMIB) em seus respectivos módulos. Assim, após a inicialização do EMS, este se encontrará completamente operacional e responsivo a requisições. Em seguida, o MC se torna um módulo administrativo disponível. Nesse caso, operações como de mudanças de configuração do EMS, ativação e desativação de agentes de monitoramento, inclusão e exclusão de instâncias de VNF gerenciadas e registro de novos usuários são conduzidas através dele. Essas operações administrativas são também disponibilizadas para os usuários através do SA, e suas efetivações não devem comprometer o desempenho das tarefas de gerenciamento de instâncias de VNF sendo executadas. É importante ressaltar que o MC mantém o controle generalizado do EMS, sendo capaz de acessar e modificar qualquer um de seus módulos configuráveis. Sendo assim, é importante garantir que o acesso a esse módulo seja realizado apenas por usuários com nível de privilégio adequado.

De maneira geral, quando uma requisição é recebida pelo SA através do IA (Ia-Sa), o AO procede com a sua validação considerando o conjunto de operações disponibilizadas por ele. Caso a requisição esteja conforme, esta é encaminhada ao AA (Ao-Aa) que autentica o requerente e verifica se o mesmo possui privilégios para a execução da operação pretendida. Caso a requisição seja válida e autorizada, ela é então encaminhada do SA para o RI (Sa-Ri). O RI realiza o registro interno para rastreamento da requisição e a encaminha para o SV (no caso de uma operação de gerenciamento de VNF – Ri-Vs) ou ao MC (no caso de uma operação de (re)configuração do EMS – Ri-Mc). Considerando, primeiramente, que a requisição se trata de um gerenciamento de VNF, o SV realiza a identificação da plataforma de execução de VNF que está sendo requisitada e o endereço de sua respectiva instância virtualizada (essas informações são incluídas pelo SA durante o processo de validação da requisição). Atentando a essas informações, a requisição é remodelada e encaminhada através da IV adequada (Sv-Iv). Por outro lado, se a requisição for por uma operação de configuração do EMS, o MC identifica o módulo alvo da mesma e atua com intermediário para sua execução através do plano de gerência/informação. Os resultados das operações seguem o caminho inverso no EMS, sendo finalmente encaminhados ao requerente pelo SA. Os demais pontos de acesso interno que devem ser destacados são o Sm-Sv, que é utilizado pelo SM para requisitar periodicamente ao SV operações de monitoramento de VNF de seus agentes; e o Sm-Im, sendo usado no envio de mensagens geradas pelos agentes de monitoramento do SM.

4. HoLMES: *Holistic, Lightweight and Malleable EMS Solution*

Um protótipo foi desenvolvido como prova-de-conceito da arquitetura EMS proposta neste trabalho. O protótipo, chamado *Holistic, Lightweight and Malleable EMS Solution* (HoLMES - disponível em <https://github.com/ViniGarcia/HoLMES>), im-

plementa todos os módulos internos e interfaces previstas na arquitetura EMS, sendo completamente compatível com a arquitetura de referência do paradigma NFV [ISG 2014]. O HoLMES foi projetado para ser genérico e comunicar-se com diferentes plataformas de execução de VNF e de NFV-MANO. Além disso, sua implementação permite que várias instâncias de VNF sejam gerenciadas ao mesmo tempo por uma única instância de EMS. Assim, o HoLMES é classificado como **MANO (independente), externo, restrito e guarda-chuva**. O protótipo é oferecido como um pacote de instalação isolado ou como uma imagem pré-configurada no sistema operacional *Ubuntu Cloud* [Canonical 2021]. Esse sistema operacional provê um núcleo de operações e um ambiente para a instalação de aplicações, sendo relativamente econômico em termos da demanda de recursos computacionais e executando os módulos da plataforma com total compatibilidade de *software*.

O HoLMES foi implementado através da linguagem de programação Python3. Em particular, o Subsistema de Acesso utilizou-se da biblioteca Flask para operacionalizar uma Interface de Acesso. Ainda, nesse subsistema, o Agente de Operação disponibiliza nativamente as funções do protocolo *Ve-Vnfm-em* [SOL 2020] cujo fluxo de requisição acontece do VNFM para o EMS. Como destaque, o Agente de Operação suporta a execução de *drivers* de comunicação entre EMS e plataformas de NFV-MANO (em específico, com o VNFM dessas plataformas). Esses *drivers* providenciam as funções do *Ve-Vnfm-em* em que o fluxo de requisição parte do EMS para o VNFM. Para isso, um *template* é disponibilizado pelo protótipo (*VnfmDriverTemplate*), habilitando seus operadores a programar *drivers* de VNFM sob demanda. O Agente de Autenticação oferece métodos com diferentes níveis de segurança para verificar a autenticidade dos emissores das requisições que chegam ao EMS, exemplos desses métodos vão desde a aceitação incondicional (ausência de autenticação) até aqueles baseados em usuário e senha. O Agente de Autenticação é programado de maneira modular e pode ser facilmente estendido com poucas alterações em seu código fonte. O Roteador Interno consiste de um ponto central de troca de mensagens entre os módulos do EMS. O Roteador Interno é responsável por criar rótulos de identificação de requisições e resultados, encaminhando os mesmos para os módulos necessários através de chamadas de função.

O Subsistema de Monitoramento foi implementado utilizando a biblioteca de processamento paralelo *multiprocessing*. Esse subsistema gerencia múltiplos agentes de monitoramento executados como processos. Os agentes comunicam-se com instâncias de VNF através do Subsistema de VNF, executando rotinas de monitoramento a partir das operações disponibilizadas em suas respectivas plataformas de execução. Um agente de monitoramento é assinado por usuários autorizados de uma instância do HoLMES, os quais passam a receber notificações de monitoramento disparadas na ocorrência de eventos preestabelecidos. Essas notificações são enviadas por Interfaces de Monitoramento definidas pelos próprios agentes. A programação de agentes de monitoramento deve ser realizada considerando um modelo importável próprio do HoLMES, este fornecido como parte do protótipo (*MonitoringAgentTemplate*). O último subsistema, o Subsistema de VNF, realiza a comunicação entre o HoLMES e os agentes de gerência das plataformas de execução de VNF. Como plataformas de execução de VNF heterogêneas podem ser utilizadas, o Subsistema de VNF suporta a criação e aplicação de *drivers* de forma similar à dinâmica empregada para a comunicação com o VNFM no Agente de Operações. Assim, cada plataforma de execução de VNF apresenta um *driver* correspondente que especifica suas operações de gerenciamento, além de definir uma Interface de VNF de acordo com a

sua tecnologia e protocolo de comunicação em particular (*e.g.*, REST, *Socket* e RMI). O *template* usado para a criação dos *drivers* relacionados ao Subsistema de VNF também é provido junto ao protótipo (*VnfDriverTemplate*).

A Base de Informações de Gerência de EMS do HoLMES é banco de dados relacional construído com SQLite. Nessa base de dados são armazenadas todas as informações necessárias para a execução e manutenção do sistema, contendo tabelas como de usuário, plataformas de execução de VNF, instâncias de VNF, agentes de monitoramento, entre outras. A EMIB, assim como os demais módulos do protótipo, é gerenciada pelo Módulo de Configuração, o qual funciona como um eixo de execução de operações de configuração e atualização do sistema. Assim, qualquer requisição que não seja relacionada a uma instância de VNF em particular, passa pelo Módulo de Configuração para que seja aplicada de fato dentro do HoLMES. Essas operações de configuração e atualização do protótipo também são disponibilizadas (utilizando um protocolo próprio) pelo Agente de Operação, sendo acessíveis por usuários com privilégios de administradores do sistema. Em relação à comunicação interna, o HoLMES utiliza três tipos principais de mensagens: (i) *MensagemRI*, modelo padrão de troca de mensagens com o RI, mantém o rastreamento de origem e destino do conteúdo (utilizado nas conexões Sa-Ri, Ri-Mc e Ri-Vs); (ii) *MensagemGI*, conteúdo enviado em uma *MensagemRI* quando o destinatário é o MC (Ri-Mc), contém informações necessárias para uma operação de configuração/atualização do EMS; e (iii) *MensagemVS*, conteúdo de uma *MensagemRI* quando o destinatário é o VS (Ri-Vs), contém informações para a execução de requisições em instâncias de VNF.

5. Experimentação e Resultados

A arquitetura proposta foi avaliada através do protótipo HoLMES na execução de operações típicas de gerenciamento de funções de rede virtualizadas. Com isso objetivava-se determinar a sobrecarga imposta no tempo de operação devido ao uso do EMS intermediando a comunicação entre um OSS e diferentes plataformas de execução de VNF. Para tanto, dois cenários são comparados: (i) OSS implementando uma interface de comunicação para cada uma das plataformas de execução de VNF testadas, requisitando operações de gerência diretamente aos seus MA; (ii) OSS implementando uma interface de comunicação única com o EMS, o qual intermedeia as requisições por operações de gerência com as plataformas de execução de VNF. Três plataformas de execução de VNF foram testadas: Click-On-OSv (COO – interface HTTP) [Marcuzzo et al. 2017], COVEN (interface HTTP e via Socket TCP) [Fulber-Garcia et al. 2019b] e Leaf (interface HTTP) [Flauzino et al. 2020]. Três operações providas de forma similar nessas plataformas foram executadas: (i) obter o *status* de execução da plataforma, operação leve (*Get Status*); (ii) enviar uma função de rede para a plataforma, operação de transferência de dados (*Post NF*); e (iii) configurar e iniciar uma função de rede na plataforma, operação de computação intensiva (*Post Configure and Start*). O tempo de operação é medido a partir da ótica do OSS, sendo a contagem iniciada no momento do envio de uma requisição e cessada no momento do recebimento de sua resposta.

Duas máquinas foram utilizadas na realização dos experimentos, uma para a instanciação dos componentes operacionais de gerência e outra para a instanciação das funções de rede virtualizadas. A máquina de gerência hospeda tanto o OSS que emite requisições, quanto uma instância do HoLMES. Essa máquina conta com um processador Intel Core I3 4010-U 1.9GHz, 8GB de memória RAM DDR3 e Ubuntu 16.04. A segunda máquina é dedicada à instanciação e execução das funções de rede virtualizadas,

sendo a virtualização de cada plataforma de execução de VNF mantida somente durante a execução de seus respectivos experimentos. Essa máquina conta com processador Intel Core I5-3330 3.0GHz, 8GB de memória RAM DDR3 e Ubuntu 16.04. A virtualização das funções de rede foi feita através do hipervisor KVM. Todos os sistemas utilizados estavam conectados a uma rede local GbE e contavam com um IP dedicado. Cada experimento foi executado 150 vezes, sendo 10% dos resultados extremos (*i.e.*, melhores e piores) descartados, totalizando o processamento útil de 120 execuções.

O primeiro experimento recupera o *status* das plataformas de execução de VNF como uma cadeia de caracteres. Essa é uma operação de monitoramento considerada leve, pois não envolve transferência de arquivos e o tempo de processamento para sua efetivação é desprezível. A média e a distribuição por execução do tempo dessa operação nas plataformas testadas são apresentadas, respectivamente, nas Figuras 2 e 3. Nesse experimento, a sobrecarga no tempo de execução originada pela utilização do HoLMES variou entre 150% (Leaf) e 300% (COO), sendo as sobrecargas intermediárias iguais a 225% e 250% para, respectivamente, COVEN-HTTP e COVEN-Socket. A maior discrepância bruta de resultados, contudo, entre o cenário (i) - sem EMS - em relação ao cenário (ii) - com EMS - ocorreu durante a testagem do COVEN-Socket (10ms). Nesse caso, a maior sobrecarga é atribuída, além da presença da indireção de requisições para o HoLMES, à necessidade do processamento de um protocolo de camada sete (HTTP) no EMS utilizado, o que é dispensado no primeiro cenário devido à comunicação ser realizada exclusivamente via sockets TCP. A sobrecarga no tempo de operação nas demais plataformas deve-se exclusivamente à indireção decorrente do EMS. Assim, a diferença no percentual médio do tempo de operação é atribuída às características operacionais das plataformas testadas, refletindo na complexidade dos *drivers* de interfaceamento do EMS.

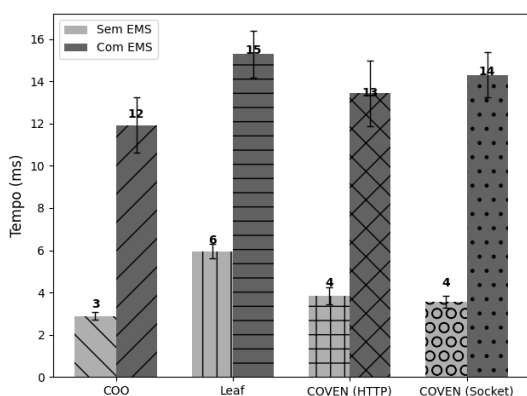


Figura 2. Média - *Get Status*

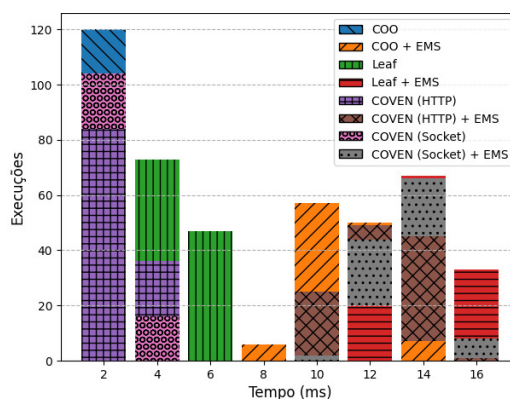


Figura 3. Distribuição - *Get Status*

No segundo experimento são enviadas funções de rede para as plataformas de execução de VNF. Essa operação de configuração é dedicada à transferência de dados e arquivos. Os resultados de média e distribuição por execução do tempo de operação são exibidos, em ordem, nas Figuras 4 e 5. A média da sobrecarga do tempo de operação decorrente do HoLMES variou entre 367% (COO) e 875% (COVEN-Socket), sendo os resultados intermediários iguais a 173% para o Leaf e 389% para o COVEN-HTTP. A maior variação do tempo de operação em relação ao primeiro experimento (*i.e.*, *Get Status*) acontece devido à necessidade da transferência de arquivos das funções de rede para um intermediário (*i.e.*, o EMS), que então realiza o envio para a plataforma destino. As

funções de rede enviadas para cada plataforma testada diferem entre si, uma vez que estas plataformas apresentam requisitos, capacidades e características específicas. Para o COO, a função de rede é um simples encaminhador de tráfego com 352 octetos; para o Leaf, a função de rede encontra-se pré-instalada na plataforma (servidor Apache2) e o pacote de configuração enviado contém 2797 octetos; para o COVEN, a função de rede enviada consiste em um encaminhador de tráfego com quatro componentes, sendo o pacote enviado contando com 9287 octetos. A diferença na quantidade de dados transmitida impacta ativamente na variação da sobrecarga produzida pela utilização do EMS. Por fim, no caso do COVEN-Socket, a maior sobrecarga é gerada por motivos análogos aos do primeiro experimento, estes agravados pela necessidade de transferência de dados. Destaca-se que, no pior caso de sobrecarga (875%), o tempo extra não superou 35 milissegundos.

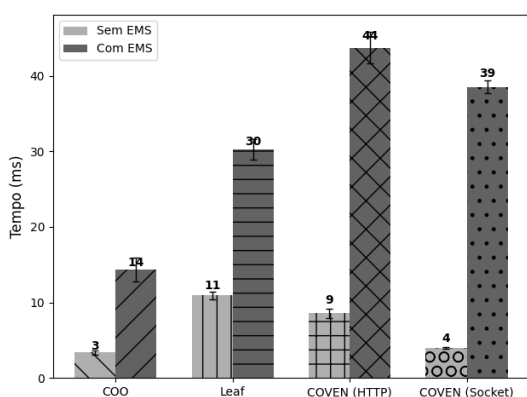


Figura 4. Média - Post NF

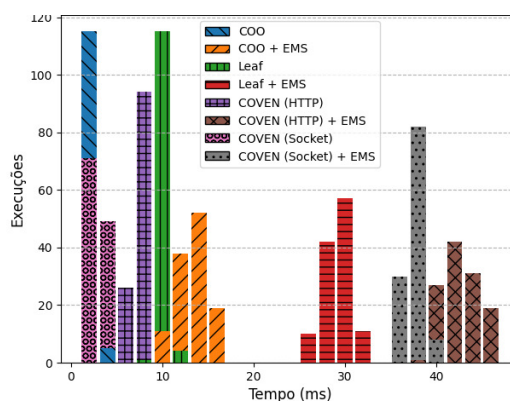
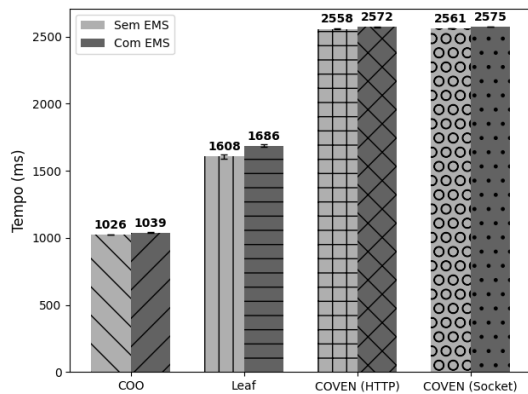
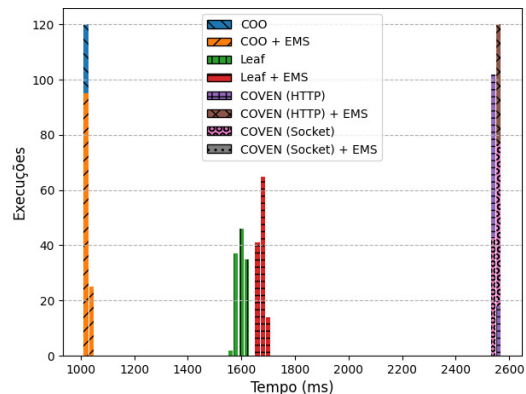


Figura 5. Distribuição - Post NF

O terceiro experimento realiza a configuração e inicialização das funções de rede previamente enviadas para as plataformas de execução de VNF. Essas operações de configuração resultam em situações de computação intensiva internamente às plataformas. A média e a distribuição por execução do tempo de operação nos cenários testados são expostas nas Figuras 6 e 7, respectivamente. A média de sobrecarga no tempo de operação proveniente do HoLMES variou entre 0,6% (COVEN-HTTP e COVEN-Socket) e 4,8% (Leaf), sendo o resultado intermediário igual a 1,3% (COO). Esse experimento apresenta a menor sobrecarga relativa entre a utilização ou não de um EMS intermediando a comunicação. Tal fenômeno ocorre devido às características de execução das operações de configuração e inicialização de funções de rede: requisição leve (sem transferência de dados), resultando em baixo tempo de transmissão, e operação custosa, requerendo maior tempo de processamento interno às plataformas de execução de VNF. A plataforma Leaf representa o teto da variação de sobrecarga. Esse comportamento decorre da necessidade de duas chamadas HTTP para iniciar uma função de rede na plataforma, ocasionando uma sobrecarga acumulada. Em contrapartida, as demais plataformas exigem apenas uma chamada HTTP ou uma conexão via *socket* TCP para executar uma operação de inicialização equivalente, culminando nas menores sobrecargas observadas. É importante ressaltar, por fim, que o tempo absoluto elevado resultante da experimentação com a plataforma COVEN decorre de seu sistema complexo, com uma arquitetura modular e uma variedade de características que são aplicadas em tempo de inicialização, além desta suportar a instanciação de vários componentes compondo suas funções de rede.



**Figura 6. Média - Post Configu-
re/Start**



**Figura 7. Distribuição - Post Con-
figure/Start**

Os resultados dos experimentos demonstram que a utilização de um EMS (em particular, do HoLMES) acresce o tempo total de operação para a aplicação de operações de gerência em plataformas de execução de VNF. Porém, o HoLMES foi capaz de abstrair a comunicação com diferentes plataformas, tratando a heterogeneidade dos agentes de gerência e suas respectivas tecnologias. Esse tratamento permite que, através uma interface única e padronizada, o OSS seja capaz de comunicar-se com uma variedade de plataformas, facilitando seu desenvolvimento e usabilidade. Quanto à variação da sobrecarga de tempo imposta pelo uso do HoLMES, esta pode representar, percentualmente, maiores (875%) ou menores (0,6%) impactos de acordo com as características da operação requisitada e da plataforma de execução de VNF destino. Porém, mesmo com casos de percentuais elevados, o acréscimo de tempo não superou 78 milissegundos, sendo uma variação bruta aceitável para uma miríade de cenários sem requisitos estritos de tempo.

6. Conclusão

O gerenciamento de funções de rede virtualizadas é um dos desafios do paradigma NFV. O EMS é o componente responsável pela requisição de operações de gerência diretamente nas instâncias de funções de rede virtualizadas. Esse componente é pouco abordado na literatura recente, sendo frequentemente ignorado ou, quando utilizado, desenvolvido de forma ingênua, sem considerar nenhuma orientação arquitetônica. Esse cenário gera implementações de EMS restritivas e com aplicabilidade limitada, contribuindo de forma periférica para a evolução do gerenciamento NFV. O presente trabalho apresentou uma arquitetura interna para o componente de EMS totalmente compatível com a arquitetura de referência NFV. Essa arquitetura consiste em uma série de módulos pouco acoplados que se comunicam através de interfaces padronizadas, permitindo abstrair e intermediar de forma holística a comunicação entre VNF, OSS/BSS e plataformas de execução de VNF. A arquitetura foi implementada como um protótipo (HoLMES), o qual foi avaliado em relação à sobrecarga de tempo imposta na execução de diferentes operações de gerenciamento de funções de rede virtualizadas. Os resultados mostraram que o HoLMES realizou suas funções de interfaceamento com sucesso, impondo uma baixa ou moderada sobrecarga no tempo de operação, com valores brutos médios contidos entre 9 e 78 milissegundos. Trabalhos futuros incluem a disponibilização de *drivers* nativos do HoLMES para plataformas de NFV-MANO (e.g., Tacker [OpenStack 2020], OSM [ETSI 2020], Open Baton [OpenBaton 2020] e Vines [Flauzino et al. 2020]).

Referências

- Canonical (2021). Ubuntu cloud operating system. <https://cloud-images.ubuntu.com>.
- Carella, G. et al. (2014). Cloudified ip multimedia subsystem (ims) for network function virtualization (nfv)-based architectures. In *Symposium on Computers and Communications*, pages 1–6. IEEE.
- ETSI, E. T. S. I. (2020). Osm: Open source mano. <https://osm.etsi.org>.
- Flauzino, J. et al. (2020). Além do openstack: Disponibilizando o suporte para funções virtualizadas de rede nfv-mano no cloudstack. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 435–448. SBC.
- Fulber-Garcia, V. et al. (2019a). An nsh-enabled architecture for virtualized network function platforms. In *International Conference on Advanced Information Networking and Applications*, pages 376–387. Springer.
- Fulber-Garcia, V. et al. (2019b). On the design of a flexible architecture for virtualized network function platforms. In *Global Communications Conference*, pages 1–6. IEEE.
- Hawilo, H. et al. (2014). Nfv: State of the art, challenges, and implementation in next generation mobile networks (vepc). *Network*, 28(6):18–26.
- IFA, E. N. (2016). Network functions virtualisation (nfv); management and orchestration; report on architectural options.
- IFA, E. N. (2018). Network functions virtualisation (nfv); management and orchestration; report on management of nfv-mano and automated deployment of em and other oss functions.
- IFA, E. N. (2020). Network functions virtualisation; management and orchestration; ve-nfm reference point; interface and information model specification.
- ISG, E. N. (2014). Network functions virtualization architectural framework.
- Marcuzzo, L. et al. (2017). Click-on-osv: A platform for running click-based middleboxes. In *Symp. on Integrated Network and Service Management*, pages 885–886. IEEE.
- OpenBaton (2020). Open Baton: An extensible and customizable NFV MANO-compliant framework. <https://openbaton.github.io>.
- OpenStack (2020). Tacker: Openstack nfv orchestration. <https://wiki.openstack.org/wiki/Tacker>.
- REL, E. N. (2015). Network functions virtualisation (nfv); resiliency requirements.
- SEC, E. N. (2019). Network functions virtualisation (nfv); trust; report on certificate management.
- SEC, E. N. (2020). Network functions virtualisation (nfv); security; access token specification for api access.
- Sharma, G. P. et al. (2019). Dynamic hardware-acceleration of vnfs in nfv environments. In *International Conference on Software Defined Systems*, pages 254–259. IEEE.
- SOL, E. N. (2020). Network functions virtualisation; protocols and data models; restful protocols specification for the ve-vnfm reference point.