

NIEP eXperience: Aprimorando a Usabilidade do Emulador de Redes com Suporte à NFV NIEP

Marcelo Gyovani Pereira
marcelop2307@gmail.com
Universidade Federal do Paraná
Curitiba, Paraná, Brasil

Vinicius Fulber-Garcia
vinicius@inf.ufpr.br
Universidade Federal do Paraná
Curitiba, Paraná, Brasil

Abstract

New networking paradigms have been developed to provide greater flexibility and enable finer-grained management of computer networks. One of the most prominent new network paradigm is the Network Function Virtualization (NFV), which can leverage the *NFV Infrastructure Emulation Platform* (NIEP) as an environment for prototyping and testing virtualized network functions and services. Despite its versatility and comprehensiveness, NIEP poses a steep learning curve for new users, particularly in the process of specifying virtualized network topologies. To address this challenge, this work proposes NIEPx, an assistive interface designed to simplify topology specification within NIEP while minimizing common errors in the process. The effectiveness of NIEPx was evaluated through a qualitative-quantitative model, which demonstrated its ability to reduce human errors during topology specification. Furthermore, the interface was tested for communication overhead with the NIEP emulator, showing negligible impact in the analyzed scenarios. Thus, NIEPx emerges as an effective solution to facilitate the onboarding of new users to NIEP, while also supporting experienced users in handling more complex network topology specification tasks.

Keywords

Rede, Emulador, Interface, Experiência, NFV

1 Introdução

A virtualização de funções de rede (NFV) é uma maneira de virtualizar serviços de rede normalmente executados em hardware especializado, como *firewalls*, *switches* e roteadores. Dessa maneira, essas funções são empacotadas em máquinas virtuais que podem ser executadas em máquinas de propósito geral, não necessitando do hardware especializado. Por isso, arquiteturas baseadas em NFV oferecem baixo custo, flexibilidade e escalabilidade, sendo interessantes tanto para a indústria quanto para a academia.

Apesar dos benefícios da NFV, ainda há muitos desafios para a sua implementação em ambientes de produção, sendo um deles a predição do seu impacto em tais situações. Para resolver esse desafio, foi proposta a *NFV Infrastructure Emulation Platform* (NIEP) [14], uma plataforma de emulação de redes com suporte a NFV que facilita o *design* e teste de redes NFV.

Entretanto, o emulador NIEP apresenta uma curva de aprendizado muito acentuada e um processo de criação de topologias complexo. A definição de uma topologia de rede no NIEP utiliza múltiplos arquivos com informações interconectadas, o que torna esse processo difícil e a edição de uma topologia ainda mais desafiadora. Além disso, o emulador é majoritariamente baseado no Click-on-OSv [4] e exige que o usuário saiba utilizar a linguagem

Click para configurar funções de rede executando no NIEP. Dessa forma, fica claro que a curva de aprendizado e o esforço necessário para utilizar o NIEP são altos, o que acaba desincentivando novos usuários.

Sendo assim, o desenvolvimento de uma documentação completa com exemplos de definições de topologias diferentes tem potencial de diminuir a curva de aprendizado do NIEP, facilitando a utilização para um usuário novo. Porém, essa abordagem não melhora a usabilidade do emulador em si, nem ajuda usuários experientes com o NIEP. Outra alternativa é a criação de uma interface gráfica capaz de abstrair a definição de uma topologia e automatizar parte do trabalho do usuário, ajudando usuários independentemente do nível de experiência com a plataforma e aprimorando a usabilidade.

Assim, o objetivo deste trabalho é simplificar o processo de definição e edição de topologias de rede e de serviços virtualizados, suavizar a curva de aprendizado e melhorar a experiência do usuário através da introdução de uma interface gráfica para a plataforma NIEP, chamada de *NIEP eXperience* (NIEPx). Essa plataforma visa, principalmente, facilitar o aprendizado e aprimorar a produtividade de todos os usuários.

Tecnicamente, o NIEPx é uma interface gráfica para o NIEP que possibilita definir uma topologia de forma intuitiva, assistida e simplificada, utilizando elementos gráficos para representar a rede de maneira visual e abstraída. Após o desenvolvimento, o método baseado na técnica *Human Error Assessment and Reduction Technique* (HEART) [2] foi utilizado para avaliar a interface de forma quantitativa. A avaliação demonstrou que a interface reduziu drasticamente a chance de ocorrerem erros durante a definição de uma topologia, em comparação com o método tradicional de configurá-la manualmente, melhorando a eficácia e a eficiência do uso do NIEP.

O restante desse trabalho é organizado da seguinte forma. Na Seção II, o contexto e o trabalho relacionado é revisado. Na Seção III, é apresentado as funcionalidades e detalhes do NIEPx. Na Seção IV, é demonstrado o processo de avaliação da interface. Na Seção V, há os detalhes de implementação. Por fim, na Seção VI, são apresentadas as conclusões e possibilidades de trabalhos futuros.

2 Virtualização de Funções de Rede e a Plataforma NIEP

A Virtualização de Funções de Rede (*Network Function Virtualization* - NFV) é um paradigma proposto pelo *European Telecommunications Standards Institute* (ETSI) para o desenvolvimento, gerenciamento e implementação de redes de forma mais econômica e flexível em comparação às implementações tradicionais baseadas em *physical appliances* [3]. Nesse paradigma, as funções de rede são desvinculadas de um hardware específico e implementadas em software,

por meio da virtualização, na forma de Funções Virtuais de Rede (*Virtualized Network Functions* - VNFs).

Particularmente, uma VNF possui dois componentes principais: (i) a plataforma de execução e (ii) a função de rede (*Network Function* - NF) [6]. A plataforma de execução refere-se ao ambiente de virtualização da VNF, podendo ser uma máquina virtual ou um contêiner, enquanto a função de rede corresponde às rotinas e operações que serão executadas nessa plataforma. Exemplos de funções de rede incluem roteamento, filtragem, análise e tunelamento de tráfego de rede.

Entre as vantagens do paradigma NFV estão a redução de custos e a flexibilidade proporcionada pelo uso de hardware genérico, além da agilidade na implantação, configuração e escalabilidade dos serviços. NFV também possibilita o uso de Cadeias de Funções de Serviço (*Service Function Chains* - SFCs) [5], que consistem em topologias formadas por múltiplas VNFs com propósitos específicos, permitindo a implementação de serviços sofisticados e complexos.

Entretanto, o paradigma NFV continua em fase experimental e necessita de ambientes de teste robustos para verificar a eficiência e eficácia das funções e serviços implementados. Assim, o desenvolvimento de ferramentas para emular, avaliar e depurar VNFs é de extrema importância para a maturação e popularização desse paradigma. Além disso, o monitoramento de funções virtualizadas excede as capacidades das ferramentas legadas, como *ping*, *tcpdump* e *traceroute*, sendo necessárias plataformas e elementos de gerência especialistas.

Nesse contexto, surge o *NFV Infrastructure Emulation Platform* (NIEP) [7, 14], uma plataforma de emulação de redes com suporte à infraestrutura NFV, que utiliza tanto virtualização leve quanto completa para executar funções de rede. O NIEP é ideal para testar e prototipar redes baseadas em virtualização. Sua implementação é composta principalmente por quatro componentes: (i) o Mininet [10] para emulação de redes, (ii) o *KVM Hypervisor* para gerenciamento de máquinas virtuais, (iii) o *Click-on-OSv* [4] como plataforma nativa de VNF, e (iv) um módulo de orquestração, o qual é o núcleo do emulador. A definição de topologias no NIEP é feita de forma nativa por meio de um arquivo *JavaScript Object Notation* (JSON), onde são declaradas todas as funções de rede, serviços de rede, elementos virtualizados, *links*, conexões, entre outros recursos disponíveis.

Dessa forma, o NIEP se destaca como uma plataforma de emulação de rede completa, oferecendo um conjunto amplo de funcionalidades que vai além das plataformas de emulação tradicionais, como o Mininet. Também, a plataforma NIEP foi projetada para ser altamente extensível, permitindo a integração com novas plataformas de execução e funções de rede de maneira simples e eficaz. Esse nível de extensibilidade não apenas facilita a adição de novas VNFs, mas também proporciona flexibilidade para acomodar tecnologias emergentes e novos paradigmas de rede.

3 Trabalhos Relacionados

Com o aumento da popularidade das redes de computadores, que são um dos principais meios modernos de comunicação humana, diversas plataformas de emulação de redes têm sido propostas para implementar e testar novas tecnologias nesse contexto. Entre essas

plataformas, destacam-se, além do próprio NIEP, o Mininet [10], o EsCAPE [13] e o MeDICINE [11].

O Mininet [10] é uma popular plataforma de emulação de redes com suporte nativo a hosts, switches e links. O Mininet pode ser configurado sequencialmente por meio de uma interface de linha de comando ou por meio de uma interface gráfica para a criação de topologias, chamada Miniedit. No entanto, apesar de fornecer uma interface amigável e intuitiva para novos usuários, o Mininet não é capaz de emular ambientes NFV nem de instanciar e gerenciar máquinas virtuais nativamente.

O *Graphical Network Simulator 3* (GNS3) [8] é uma plataforma de simulação de redes que conta com uma interface gráfica, tornando-se bastante acessível para novos usuários. No entanto, o GNS3 não oferece suporte nativo ao paradigma NFV, o que dificulta a simulação de topologias de rede que utilizam funções e serviços virtualizados. Além disso, o foco principal do GNS3 está na simulação, enquanto o uso de elementos virtualizados reais, como em plataformas de emulação, não é seu objetivo.

O EsCAPE [13] é uma plataforma de emulação de rede com suporte a VNFs, porém sua definição de VNF é limitada, sem suporte à execução de plataformas de VNF reais ou qualquer tipo de máquina virtual genérica. Além disso, o EsCAPE não possui uma interface gráfica, sendo necessário escrever manualmente as topologias em um documento formatado em JSON, o que torna o processo verboso e dificulta até mesmo a definição de topologias de rede simples.

O MeDICINE [11] é uma plataforma de prototipação capaz de emular um ambiente com múltiplos pontos de presença (PoPs), executando funções de rede em contêineres Docker. O MeDICINE é baseado no ContainerNET2, uma extensão do Mininet que adiciona suporte para VNFs utilizando contêineres. Assim como o EsCAPE, não suporta plataformas de VNF reais, nem gerenciamento e orquestração de VNFs genéricas, além de não possuir uma interface gráfica. Entretanto, a definição de topologias é feita por meio de *scripts*, ou pode ser gerada algorítmicamente.

Finalmente, o NIEP oferece suporte nativo e explícito à criação de VNFs, SFCs e máquinas virtuais, o que o diferencia significativamente das demais plataformas de emulação. No entanto, assim como o EsCAPE e o MeDICINE, o NIEP não provê uma interface simples e intuitiva para a definição das topologias de rede a serem emuladas, sendo todo o processo realizado por meio da escrita de um documento em formato JSON, o que torna a curva de aprendizado para o uso da plataforma alta e onerosa.

De maneira geral, as plataformas de emulação de rede carecem de interfaces intuitivas que possam simplificar a interação com usuários menos experientes e otimizar fluxos de trabalho para a criação e configuração de topologias de rede. Além disso, essas interfaces, quando existem, não focam em processos de assistência para a criação e execução das topologias de rede, impedindo que erros de definição ocorram durante o processo.

4 NIEPx: NIEP *eXperience*

Considerando as lacunas nas soluções voltadas para facilitar e assistir usuários de emuladores de redes, e, em especial, a ausência de soluções compatíveis com emuladores que suportam novas tecnologias e paradigmas de rede, este trabalho propõe o NIEP *eXperience* (NIEPx). Esta seção apresentará a arquitetura da interface dedicada

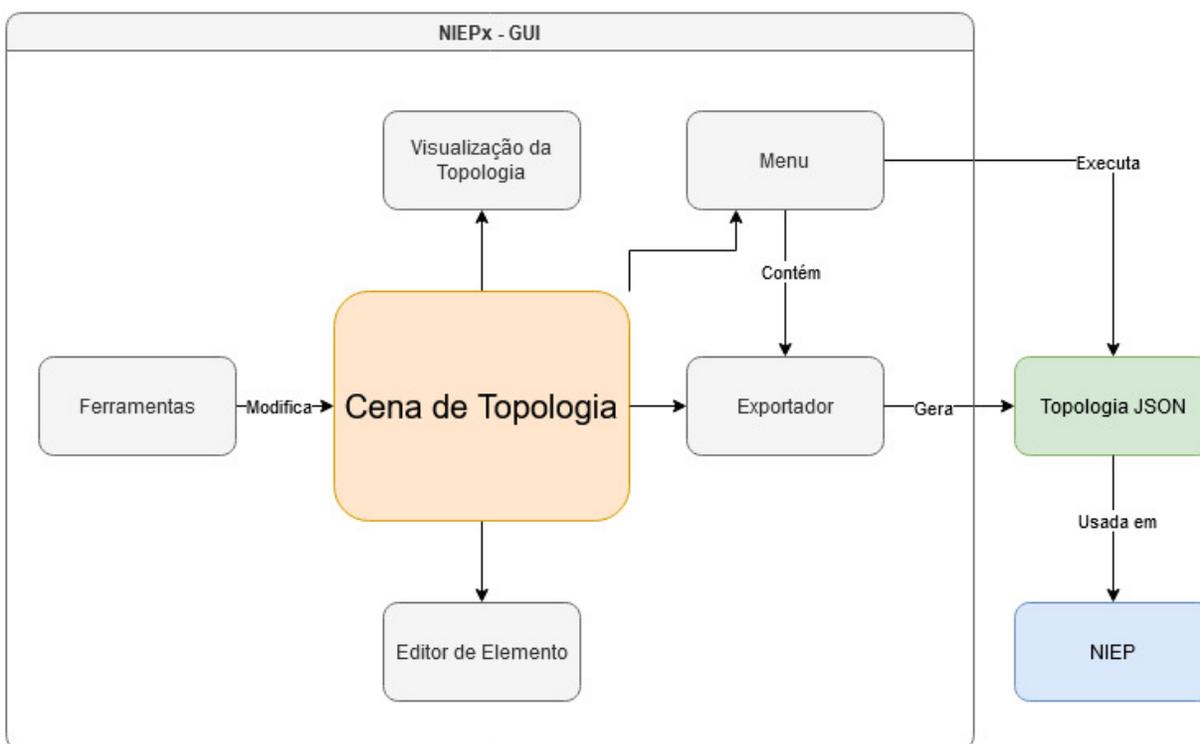


Figura 1: Arquitetura para Sistemas de Interfaceamento Gráfico para o NIEP

ao emulador NIEP, assim como o processo de desenvolvimento da solução em forma de um protótipo completamente funcional.

É importante destacar que o projeto do NIEPx é de código aberto e está disponível em um repositório online¹. Entre as funcionalidades dessa solução de interface, destaca-se o suporte à visualização da topologia, ferramentas de edição de topologia, salvamento e carregamento de topologias em arquivos, exportação da topologia para uso no NIEP e uma interface para executar a topologia diretamente no emulador.

4.1 Arquitetura

A arquitetura de um sistema de interfaceamento gráfico para o NIEP, especialmente do NIEPx, é apresentada na Figura 1, onde é possível observar seus principais componentes e as interações entre eles.

Nesse contexto, o principal componente do sistema de interfaceamento gráfico é a cena de topologia, que atua como o núcleo da interface. Essa cena é uma representação interna da topologia da rede, composta por um grafo que descreve os elementos e as conexões entre eles. Cada vértice do grafo corresponde a um elemento gráfico na interface, que pode ser um *host*, uma máquina virtual (VM), um *switch* ou um controlador. As arestas indicam as conexões entre esses elementos da topologia, sendo representadas por linhas na cena.

A cena, juntamente com outros componentes, é responsável pela criação e organização de novos elementos e conexões em uma topologia, evitando que sejam estabelecidas conexões inválidas no contexto do NIEP. Além disso, a cena também gerencia a atribuição de nomes e endereços IPv4 e MAC padrão para os novos elementos, facilitando a criação de topologias rapidamente, sem a necessidade de configuração manual desses endereços e nomes.

A visualização da topologia é a janela gráfica onde a cena de topologia é renderizada, proporcionando uma visão clara e interativa da rede. Ela permite ao usuário manipular os elementos diretamente, rearranjando ou conectando nós de forma visual. Além disso, a visualização gerencia a renderização da cena, exibindo apenas uma parte específica da topologia, permitindo que o usuário navegue por ela. Essa abordagem facilita a edição de topologias grandes, pois o usuário pode se concentrar em áreas específicas que deseja modificar, em vez de visualizar e interagir com a topologia completa de uma só vez.

O módulo de ferramentas deve oferecer, no mínimo, quatro funcionalidades básicas para interagir com a cena de topologia: selecionar, adicionar, conectar e remover. A ferramenta de seleção permite escolher um ou mais elementos da cena, possibilitando que sejam movidos e suas propriedades editadas. A ferramenta de adição é utilizada para criar novos elementos, representados por vértices na cena, enquanto a ferramenta de conexão é responsável por estabelecer as conexões (arestas) entre esses elementos. Por fim, a ferramenta de remoção pode ser usada para excluir tanto vértices quanto arestas da topologia.

¹<https://github.com/marzelop/NIEP-GUI>

O editor de elementos, por sua vez, é responsável pela configuração detalhada de qualquer elemento selecionado na cena. Ele permite a edição completa das propriedades dos elementos, incluindo a adição, remoção e modificação das interfaces de rede de hosts e VMs. O editor também valida automaticamente os valores inseridos, prevenindo o uso de caracteres ou valores inválidos. Além disso, oferece a possibilidade de alterar as interfaces de rede utilizadas nas conexões entre os elementos, garantindo flexibilidade e controle sobre a topologia.

O menu deve ser composto por, no mínimo, dois submenus, sendo um terceiro opcional, e atua principalmente como interface para a manipulação e utilização de arquivos dentro do sistema de interfaceamento. O primeiro submenu é o de arquivo, que permite salvar e carregar topologias na forma de arquivos *NIEP Graphical Interface* (NPGI), além de possibilitar a exportação da topologia para o formato JSON, necessário para a execução no NIEP. Em segundo, há um submenu de ajuda que direciona o usuário à documentação oficial do sistema. Esse recurso facilita a resolução de dúvidas e a consulta a instruções sem sair da interface. Por fim, o último submenu é o de execução, que permite iniciar e interromper a execução de topologias no NIEP através de uma interface HTTP local — um recurso nativo do emulador.

O exportador, além de ser uma opção no menu, atua como a interface entre o sistema de interfaceamento e o NIEP, possibilitando a utilização desses sistemas em conjunto. A topologia gerada é salva como uma estrutura de diretórios, contendo um arquivo principal de definição da topologia e subdiretórios que armazenam as definições de VMs e VNFs. Além disso, também é possível realizar a exportação utilizando um arquivo compactado, evitando a criação imediata de diretórios.

4.2 Interface

Conforme a arquitetura apresentada na Subseção 4.1, o NIEPx foi desenvolvido para facilitar o uso e a manipulação de topologias de rede de maneira visual e intuitiva. A tela principal do NIEPx é apresentada, segmentada em áreas numeradas, na Figura 2. Cada uma dessas áreas desempenha um papel específico para guiar o usuário na criação e modificação de uma topologia de rede NIEP. A seguir, são detalhadas as funcionalidades de cada uma dessas áreas.

Na **Área 1** está o menu, composto por três submenus: arquivo, ajuda e execução. O submenu de arquivo possui as opções de criar uma nova topologia, salvar/carregar a topologia, ou exportar a topologia pronta para execução no NIEP. A topologia NIEPx pode ser salva num único arquivo NPGI e posteriormente carregada utilizando o mesmo arquivo. A topologia também pode ser exportada para o formato NIEP, em estrutura de diretórios e com um arquivo JSON principal que serve para iniciar a topologia na plataforma de emulação. Ainda, a exportação pode gerar um arquivo compactado com os mesmos diretórios e arquivos (.zip), facilitando a transferência e compartilhamento destes.

O submenu de ajuda contém um atalho para a documentação web do NIEPx para ajudar o usuário caso encontre problemas durante a utilização da interface. O submenu de execução serve para executar ou interromper topologias no NIEP por meio de uma interface HTTP, mas não tem a capacidade de iniciar o NIEP em si.

A barra de ferramentas do NIEPx está localizada na **Área 2** e possui quatro ferramentas: seleção, conexão, adição e remoção de elementos. A ferramenta de seleção é usada para selecionar elementos da topologia, permitindo sua edição e movimentação. A ferramenta de conexão cria um *link* entre diferentes elementos. Para estabelecer uma conexão/*link* entre os elementos, o usuário, com a ferramenta de conexão selecionada, deve clicar no primeiro elemento a ser conectado e, em seguida, no segundo elemento. Assim, a conexão será criada utilizando, por padrão, as primeiras interfaces de cada elemento, caso necessário.

Vale destacar que a ferramenta de conexão só permite a criação de conexões válidas. Por exemplo, no contexto do emulador NIEP, não é possível conectar *switches* diretamente. Se a tentativa de conexão for inválida, a ferramenta irá ignorá-la, evitando erros na topologia. Além disso, o *OVSwitch* é um elemento que deve estar conectado a um único controlador. Nesse caso, o NIEPx notifica o usuário ao tentar conectar um *OVSwitch* a um segundo controlador, desfazendo automaticamente a conexão com o primeiro. Esses avisos podem ser desabilitados, caso o usuário prefira um fluxo de trabalho sem interrupções.

A ferramenta de adição permite criar um novo elemento do tipo selecionado e adicioná-lo à topologia. O tipo do elemento é escolhido pelo usuário na **Área 3**, onde é possível selecionar o tipo desejado. Após a seleção, o usuário clica em um ponto na **Área 4**, de visualização da topologia, e o novo elemento será criado nesse local, com um nome único padrão, que pode ser alterado posteriormente. Vale destacar que essa ferramenta só permite a criação de elementos representados por vértices na topologia, ou seja, ela não cria arestas ou conexões, uma funcionalidade dedicada à ferramenta de conexão. Por outro lado, a ferramenta de remoção exclui um elemento selecionado, independentemente do tipo, incluindo arestas e conexões. Importante notar que a **Área 3** tem um único propósito: selecionar o tipo do elemento a ser criado pela ferramenta de adição.

A visualização da topologia está concentrada na **Área 4** e é responsável pela representação visual da rede a ser emulada, sendo a parte mais importante da interface. Essa área é a principal forma de interação do usuário com a interface e o meio pelo qual as ferramentas da **Área 2** são utilizadas. A topologia é representada como um grafo, onde cada elemento é um vértice e suas conexões são representadas por arestas. Cada tipo de elemento da topologia possui uma cor distinta para facilitar a identificação: por exemplo, o elemento *host* é representado em azul-claro, enquanto uma VM é representada em azul-escuro. Foram usadas cores semelhantes para elementos relacionados, como *host* e VM, diferenciados por dois tons de azul, enquanto *switch* e *OVSwitch* são representados em tons de amarelo. Além disso, elementos selecionados são destacados por um contorno branco mais espesso, em contraste com o contorno preto padrão, facilitando a sua visualização.

A **Área 5** corresponde ao editor de elementos, que exhibe as informações de um elemento selecionado e permite a edição de suas propriedades. Por exemplo, no caso de uma VM, é possível editar propriedades como nome, quantidade de memória, número de VCPUs, gerenciador de disco e interfaces de rede. Além disso, é possível adicionar ou remover interfaces de rede com um simples clique em um botão. Quando necessário, as interfaces utilizadas pelas conexões do elemento são ajustadas automaticamente, o que

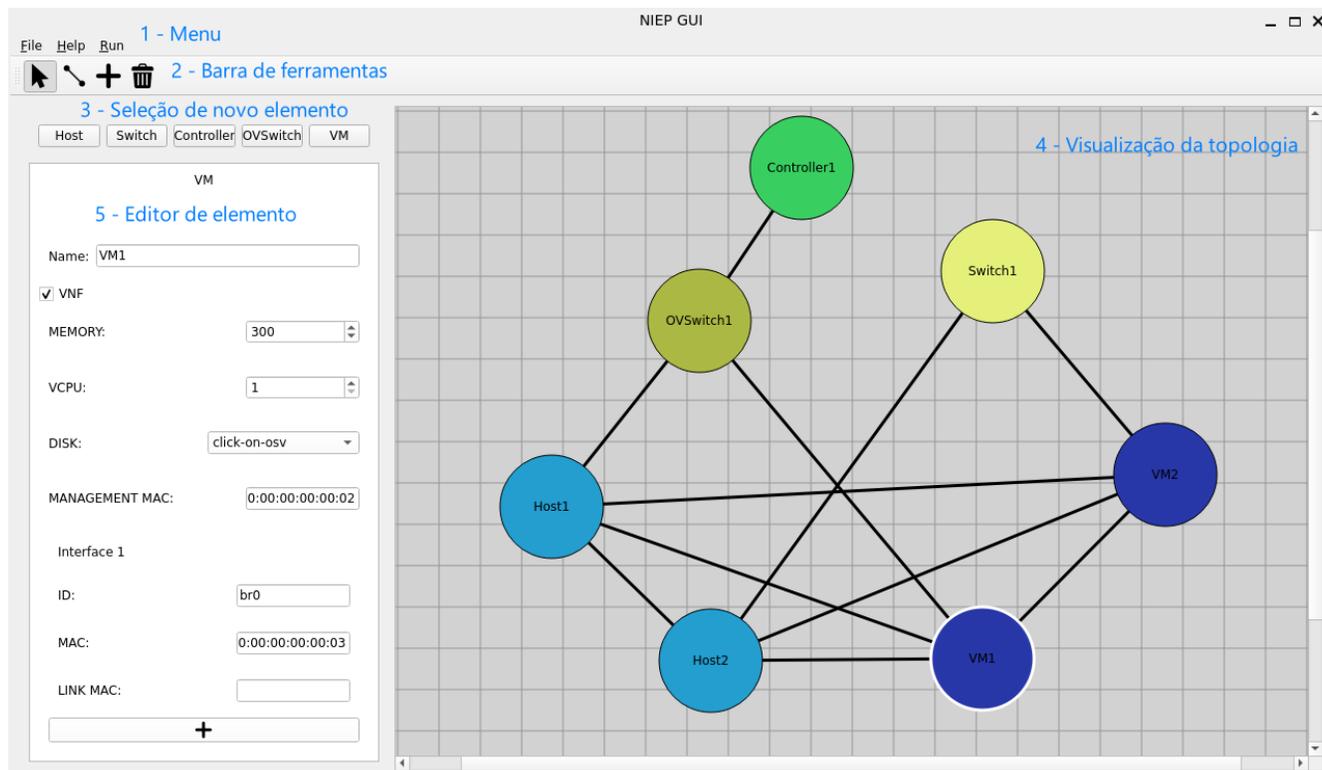


Figura 2: Interface NIEPx

economiza o esforço do usuário e previne erros decorrentes do uso de interfaces de rede inválidas.

Paralelamente, o NIEPx oferece diversos atalhos de teclado para agilizar ações como salvar, carregar e exportar topologias, além de selecionar ferramentas, permitindo um uso mais eficiente da aplicação.

O gerenciamento de erros do NIEPx é baseado principalmente na prevenção, evitando, sempre que possível, que um elemento tenha uma propriedade com valor inválido, seja bloqueando a inserção de dados incorretos, seja impedindo a criação de conexões que não façam sentido no contexto do NIEP. Quando a prevenção não é viável, o sistema realiza ajustes automáticos na topologia, como alterar a interface utilizada em uma conexão automaticamente quando a interface original é removida. Caso haja um possível erro, mas sem confirmação, o usuário é apenas notificado. Um exemplo ocorre quando uma topologia contém VMs com endereços IPv4 duplicados, mas o usuário deseja observar o comportamento da rede nesse cenário.

Dessa forma, o NIEPx oferece uma interface simples e intuitiva para a definição de topologias no NIEP, prevenindo erros antes que ocorram, facilitando o uso para novos usuários e proporcionando agilidade para os mais experientes. Vale ressaltar que o NIEPx não está totalmente integrado à plataforma NIEP; ambos funcionam de forma independente. No entanto, o NIEPx permite exportar topologias no formato JSON, que podem ser posteriormente executadas de forma nativa no NIEP, além de possibilitar a inicialização da topologia diretamente via a interface HTTP fornecida pelo emulador.

4.3 Implementação

O NIEPx foi implementado utilizando a linguagem Python 3.10.12 em um ambiente Ubuntu 22.04. Para a interface gráfica, foi utilizado o framework Qt, por meio da biblioteca PySide6 (versão 6.6.2), que permite o uso do Qt em Python. Além disso, a biblioteca NetworkX (versão 3.2.1) foi usada para o gerenciamento do grafo da cena. Para garantir a reprodutibilidade, as dependências podem ser instaladas via pip, e os arquivos de configuração estão disponíveis no repositório do [GitHub](#).

5 Experimentos e Avaliações

Nesta seção, os experimentos realizados para avaliar a efetividade do NIEPx na criação e manipulação de topologias NIEP são apresentados. Primeiramente, na Subseção 5.1, a avaliação de usabilidade do NIEPx é detalhada, focando na experiência do usuário e na facilidade de interação com a interface. Em seguida, na Subseção 5.2, discutem-se os testes de sobrecarga para a inicialização de uma topologia gerada pela interface gráfica. O objetivo central dos experimentos é validar as melhorias no processo de definição de topologias sem comprometer o desempenho durante a execução da mesma na plataforma de emulação.

5.1 Avaliação da Interface

Para conduzir o primeiro experimento, foram considerados dois métodos para a definição de topologias: (i) o método tradicional, que utiliza um editor de texto para configurar a topologia por meio

ID	Descrição	Multiplicador	Proporção	Impacto
4	Meios triviais demais de suprimir ou sobrescrever informação.	9	0,2	2,6
5	Sem meios de exibir informações espaciais e funcionais de forma intuitiva para o operador.	8	0,7	5,9
6	Conflito entre a visão do designer e o modelo do operador.	8	0,1	1,7
7	Sem forma óbvia de reverter operações.	8	0,1	1,7
12	Discrepância entre risco real e risco perceptível.	4	0,15	1,45
13	Feedback precário do sistema.	4	1	4
14	Sem confirmação clara da ação intendida por parte do sistema.	3	0,3	1,6
17	Falta de verificação ou testes do resultado.	3	0,8	2,6

Tabela 1: EPCs do Editor de Texto

ID	Descrição	Multiplicador	Proporção	Impacto
6	Conflito entre a visão do designer e o modelo do operador.	8	0,05	1,35
7	Sem forma óbvia de reverter operações.	8	0,1	1,7
12	Discrepância entre risco real e risco perceptível.	4	0,15	1,45
14	Sem confirmação clara da ação intendida por parte do sistema.	3	0,6	2,2

Tabela 2: EPCs do NIEPx

da criação de múltiplos arquivos JSON, e (ii) uma abordagem alternativa, que emprega a interface NIEPx. Dessa forma, foi realizada uma comparação entre os erros identificados em ambos os métodos de definição de topologias.

Para realizar a comparação entre os métodos, foi aplicado o modelo *Human Error Assessment and Reduction Technique* (HEART) [2]. Esse modelo, ao avaliar uma tarefa genérica, exige a determinação da incerteza associada à tarefa. Essa incerteza é definida com base em uma tabela fornecida pelo próprio modelo, considerando uma estimativa da experiência do usuário responsável pela execução da tarefa. Além disso, é necessário identificar as *Error Producing Conditions* (EPCs) presentes na tarefa e definir sua proporção de efeito. Em seguida, calcula-se o impacto de cada EPC, com base no multiplicador definido na tabela de EPCs do HEART e na proporção de efeito determinada durante a aplicação do método. O impacto total é calculado como o produtório do impacto de todas as EPCs. Por fim, calcula-se a chance de uma execução da tarefa apresentar pelo menos um erro, dada pelo produto do impacto total e a incerteza da tarefa.

Contudo, foi utilizada uma versão modificada do modelo HEART, na qual a chance de erro é descartada para generalizar o resultado do processo. Sendo assim, o impacto total (um número adimensional e sem limite superior – **cujo objetivo é minimização**) e a quantidade de EPCs (número absoluto de condições de erro) são utilizados para a comparação entre dois métodos na definição de uma mesma topologia. Também, o nível de experiência do usuário foi apenas considerado na determinação da proporção de efeito das EPCs.

A tarefa considerada na avaliação envolve a definição completa de uma topologia, pois as múltiplas relações de dependência entre os diferentes elementos do NIEP tornam desafiador definir EPCs para tarefas isoladas, como a configuração de uma única VNF. Assim, optou-se por considerar o cenário completo, abrangendo todas

	Editor de Texto	NIEPx
Impacto Total de EPCs	1069.65	7.32

Tabela 3: Impacto Total

as etapas: criação dos elementos, conexão entre eles e edição das propriedades tanto dos elementos quanto das conexões.

A tarefa de definição de topologia foi classificada como do tipo E, conforme as predefinições do modelo HEART, caracterizando-a como uma tarefa rotineira e frequentemente praticada pelo usuário. Partiu-se do pressuposto de que o usuário é um técnico especializado na área, com pleno domínio dos conceitos envolvidos na atividade. Para os dois cenários analisados, identificaram-se EPCs distintas. As EPCs do método tradicional, que utiliza um editor de texto, estão listadas na Tabela 1, totalizando 8 entradas. Por outro lado, as EPCs referentes ao uso do NIEPx estão apresentadas na Tabela 2, com 4 entradas. A Tabela 3 resume os resultados do impacto total de cada cenário avaliado.

Os resultados mostram que a diferença no impacto total entre os cenários decorre, principalmente, da visualização intuitiva da topologia e do *feedback* automático fornecido pelo NIEPx. Esses aspectos correspondem a duas EPCs de alto impacto presentes na Tabela 1, mas ausentes na Tabela 2. Além disso, os erros evitados pela interface são destacados na Tabela 4, corroborando para a redução no número de EPCs no contexto do NIEPx.

Assim, conforme as análises realizadas, destaca-se a capacidade do NIEPx de simplificar o processo de definição de topologias NIEP. A interface é capaz de prevenir que o usuário cometa erros, sejam eles erros lógicos, léxicos ou sintáticos, por meio de uma representação intuitiva da topologia em definição e de diversos mecanismos de validação que impedem a execução de operações indevidas.

Erros Previnidos	Observações
Nomes/IDs duplicados de entidades da rede	-
Conexões inválidas, como switch-switch ou host-controller	-
Usar uma interface inexistente ou inválida em uma conexão	Por exemplo, ao modificar um MAC address de uma interface, não é necessário modificar as conexões que utilizam esse MAC address, como seria se o processo fosse realizado manualmente.
Conexões com entidades de ID inválido	Para renomear uma entidade, não é necessário mudar sua identificação em todas as conexões manualmente.
Utilizar um gerenciador de disco inválido para uma VM	O usuário não precisa digitar o nome do gerenciador de disco durante a configuração de uma VM, apenas selecionar uma das opções disponíveis.
Erros de nomeação de arquivos	O usuário não define o nome dos arquivos de definições da topologia diretamente, impedindo erros por ligação de arquivos.
Utilização de caracteres problemáticos em IDs	O caractere pode gerar erros no parsing dentro do NIEP.
Remoção incorreta de elementos	A interface exclui todas as entidades que dependem totalmente de outra entidade quando essa é excluída. Por exemplo, ao apagar um host, todas as suas conexões também são apagadas.

Tabela 4: Prevenções de Erros no NIEPx

5.2 Sobrecarga para Inicialização de uma Topologia NIEP

O NIEPx pode inicializar uma topologia no emulador NIEP através de uma interface HTTP. Para que essa inicialização ocorra de forma adequada, o cliente (NIEPx) deve fornecer um arquivo ZIP, gerado pela funcionalidade de exportação da interface, para um servidor NIEP (ou seja, uma instância). O endereço e a porta do servidor NIEP precisam ser configurados no NIEPx por meio de um processo assistido.

Devido à necessidade de comunicação via rede, existe uma pequena sobrecarga, tanto para o envio de mensagens de controle e transferência dos arquivos de configuração da topologia, quanto pelo processamento adicional do servidor para extrair os arquivos do ZIP e inicializar a topologia como um subprocesso do emulador.

Assim, para medir a sobrecarga causada pela interface HTTP, foram realizados testes de dois comandos na interface de linha de comando (CLI) e na interface HTTP: os comandos de TOPOUP e KILL, que iniciam e interrompem a execução de uma topologia no NIEP, respectivamente. Cada comando foi executado 50 vezes em cada interface, sendo que os dados obtidos foram utilizados para calcular a média de tempo da execução de cada comando e o desvio padrão. O NIEP foi executado em uma máquina virtual com Ubuntu 16, sendo utilizado para os testes de CLI e como servidor no caso

		CLI	HTTP
TOPOUP	Tempo (s)	1.849	1.877
	Desvio	0.1783	0.1688
KILL	Tempo (s)	0.057	0.085
	Desvio	0.008	0.1492

Tabela 5: Testes de Sobrecarga

do HTTP que se comunicou com um cliente em outra máquina virtual, com Ubuntu 22. A topologia utilizada para testes foi uma topologia simples com apenas 3 hosts. O tempo de interação do usuário com a interface não foi considerado. Os resultados dos testes são apresentados na Tabela 5.

A sobrecarga de tempo, conforme descrita anteriormente, realmente existe, mas é insignificante no cenário apresentado: cerca de 30ms por comando, com variações potenciais dependendo do estado da rede. Embora a interface HTTP exija a transferência de arquivos, esses arquivos são pequenos, geralmente com poucos bytes, enviados em um único quadro, o que não impacta de forma significativa o tempo de execução dos comandos no emulador.

5.3 Discussão

Assim, é possível concluir que o NIEPx, conforme os testes realizados, de fato simplifica o processo de definição de topologias no NIEP e reduz a ocorrência de erros de definição ao eliminar fatores associados a erros humanos. Esse fenômeno é particularmente evidenciado pelo modelo HEART, que compara os EPCs entre os dois cenários, demonstrando a eficácia do NIEPx em mitigar os erros durante a tarefa.

Cabe destacar, entretanto, que outros modelos de avaliação de confiabilidade humana, como o *Technique for Human Error-Rate Prediction* (THERP) [12] e o *Justification of Human Error Data Information* (JHEDI) [9], podem ser empregados para avaliar a interface. No entanto, esses métodos exigem uma maior quantidade de recursos para serem aplicados em avaliações mais detalhadas. Além disso, é possível utilizar rotinas de testes automatizados para verificar o tratamento de erros realizado pela interface [1].

Além disso, foi demonstrado que a comunicação via interface HTTP do NIEP não gera uma sobrecarga significativa no contexto de uso apresentado, sendo eficaz para estabelecer a conexão entre o NIEPx e o emulador, mesmo quando o emulador está instanciado remotamente. Em termos numéricos, os testes de sobrecarga indicaram que a interface HTTP adiciona uma latência de aproximadamente 30ms em comparação com a interface de linha de comando, desconsiderando o tempo de interação do usuário. Esse fator de interação pode, na verdade, fazer com que a interface HTTP seja mais eficiente do que a CLI.

Outro ponto de destaque, porém, é a explicitação de que os resultados foram obtidos a partir da instanciação do NIEP na rede local para executar uma topologia simples. A sobrecarga de rede pode variar segundo a qualidade da conexão e a distância geográfica entre a máquina do NIEPx e o servidor NIEP, além da complexidade

da topologia. Essa complexidade, por sua vez, influencia o tamanho dos arquivos de definição que precisam ser transmitidos pela rede.

6 Conclusão

O paradigma NFV tem ganhado crescente atenção na literatura devido aos seus diversos benefícios, como flexibilidade, agilidade e escalabilidade na implementação de redes. No entanto, esse paradigma também apresenta diversos desafios. Um dos principais desafios é a previsão do impacto do uso de funções e serviços de rede virtualizados em cenários reais, e para isso, os emuladores desempenham um papel fundamental. O NIEP é um emulador de redes que suporta NFV e facilita a *design* e os testes de redes baseadas em virtualização. Contudo, o emulador apresenta uma dificuldade significativa: uma curva de aprendizado bastante alta, o que aumenta a suscetibilidade a erros durante o uso da ferramenta.

Nesse contexto, este trabalho apresentou o NIEPx, uma interface gráfica assistiva desenvolvida para simplificar a definição de topologias no NIEP. O NIEPx visa reduzir a curva de aprendizado associada ao NIEP, oferecendo uma representação visual intuitiva das topologias e simplificando os processos necessários para sua definição, facilitando o uso inicial da ferramenta. Além disso, a interface minimiza a ocorrência de erros durante a configuração das topologias por meio da validação automática e imediata das propriedades dos elementos, prevenindo equívocos, e eliminando fatores que poderiam gerar erros no processo.

O NIEPx também aprimora o gerenciamento de topologias, permitindo seu empacotamento no formato ZIP, além da tradicional exportação para o formato JSON compatível com o emulador NIEP. Esses arquivos podem ser transmitidos e executados diretamente pelo emulador, seja através de sua interface de linha de comando ou de sua interface HTTP, tornando o processo mais ágil e acessível para os usuários.

Para validar a eficácia e eficiência do NIEPx, foi realizada uma avaliação quali-quantitativa utilizando o método HEART. Os resultados dessa avaliação indicaram que o NIEPx elimina diversas EPCs presentes no processo tradicional de definição de topologias no NIEP, reduzindo significativamente a probabilidade de ocorrência de erros humanos durante a tarefa.

Adicionalmente, foram conduzidos testes de sobrecarga para avaliar se o uso da interface HTTP, empregada na execução de topologias definidas no NIEPx, introduz aumento na latência das operações do NIEP. Os resultados demonstraram uma diferença de latência desprezível nos cenários avaliados, confirmando que a interface HTTP não impacta de forma significativa o desempenho do emulador na execução das topologias.

Como trabalhos futuros, planeja-se incorporar suporte à definição de *Service Function Chaining* no NIEPx, abrangendo suas múltiplas variações estruturais, funcionalidade já disponível no NIEP, mas que ainda não pode ser configurada através da interface gráfica atual. Além disso, busca-se aprimorar a experiência do usuário, permitindo a personalização da interface de acordo com suas preferências e necessidades, tornando o uso do NIEPx e do emulador mais intuitivo e eficiente. Também, pretende-se implementar uma opção para exportar imagens das topologias criadas. Essa funcionalidade permitirá a geração de representações visuais claras e

organizadas, sem elementos específicos da interface, com resolução adequada para inclusão em relatórios técnicos e publicações acadêmicas.

Além do anteriormente citado, é um objetivo futuro a realização de uma comparação abrangente entre as interfaces gráficas de diferentes simuladores e emuladores de rede, considerando tanto a completude operacional (ou seja, se a interface gráfica permite a execução de todas as operações disponíveis em outras interfaces das plataformas) quanto a probabilidade de ocorrência de erros (avaliando se e em que medida o uso da interface gráfica reduz a probabilidade de execução de ações com erro). Por fim, serão realizados testes com usuários reais do NIEP no NIEPx, visando coletar dados para a avaliação e o aprimoramento contínuo da interface.

Referências

- [1] Pekka Aho and Tanja Vos. 2018. Challenges in automated testing through graphical user interface. In *International Conference on Software Testing, Verification and Validation Workshops*. IEEE, Västerås, Sweden, 118–121.
- [2] Tiffany Miller Alexander. 2017. Human error assessment and reduction technique (HEART) and Human factor analysis and classification system (HFACS). In *Collaboration on Quality in the Space and Defense Industries Forum*. NASA Technical Reports Server, 1–6.
- [3] Margaret Chiosi, Don Clarke, Peter Willis, Andy Reid, James Feger, Michael Bugenhagen, Waqar Khan, Michael Fargano, Chunfeng Cui, Hui Deng, et al. 2012. Network functions virtualisation. In *SDN and OpenFlow World Congress*. ONF, Darmstadt, Germany, 22–24.
- [4] Leonardo da Cruz Marcuzzo, Vinicius F Garcia, Vitor Cunha, Daniel Corujo, Joao P Barraca, Rui L Aguiar, Alberto E Schaeffer-Filho, Lisandro Z Granville, and Carlos RP dos Santos. 2017. Click-on-osv: A platform for running click-based middleboxes. In *Symposium on Integrated Network and Service Management*. IFIP/IEEE, Lisbon, Portugal, 885–886.
- [5] Vinicius Fulber-Garcia, Alexandre Huff, Carlos R. P. dos Santos, and Elias P. Duarte Jr. 2020. Network service topology: Formalization, taxonomy and the CUSTOM specification model. *Computer Networks* 178 (2020), 107337.
- [6] Vinicius Fulber-Garcia, Leonardo da C. Marcuzzo, Alexandre Huff, Lucas Bondan, Jeferson C. Nobre, Alberto Schaeffer-Filho, Carlos R. P. dos Santos, Lisandro Z. Granville, and Elias P. Duarte. 2019. On the design of a flexible architecture for virtualized network function platforms. In *Global Communications Conference*. IEEE, Waikoloa, Hawaii, USA, 1–6.
- [7] Vinicius Fulber-Garcia, Giovanni Venâncio De Souza, Elias Procópio Duarte Jr, Thales Nicolai Tavares, Leonardo Da Cruz Marcuzzo, Carlos RP Dos Santos, Muriel Figueredo Franco, Lucas Bondan, Lisandro Zambenedetti Granville, Alberto Egon Schaeffer-Filho, et al. 2020. On the Design and Development of Emulation Platforms for NFV-based Infrastructures. *International Journal of Grid and Utility Computing* 11, 2 (2020), 230–242.
- [8] Jose Gomez, Elie F Kfoury, Jorge Crichigno, and Gautam Srivastava. 2023. A survey on network simulators, emulators, and testbeds used for research and education. *Computer Networks* 237 (2023), 110054.
- [9] Barry Kirwan. 1996. The validation of three human reliability quantification techniques—THERP, HEART and JHEDI: Part 1—technique descriptions and validation issues. *Applied Ergonomics* 27, 6 (1996), 359–373.
- [10] B. Lantz, B. Heller, and N. McKeown. 2010. A network in a laptop: rapid prototyping for software-defined networks. In *SIGCOMM Workshop on Hot Topics in Networks*. ACM, New Delhi, India, 1–6.
- [11] Manuel Peuster, Holger Karl, and Steven Van Rossem. 2016. MeDICINE: Rapid prototyping of production-ready network services in multi-PoP environments. In *Conference on Network Function Virtualization and Software Defined Networks*. IEEE, Palo Alto, CA, USA, 148–153.
- [12] Rachel Benish Shirley, Carol Smidts, Meng Li, and Atul Gupta. 2015. Validating THERP: Assessing the scope of a full-scale validation of the Technique for Human Error Rate Prediction. *Annals of Nuclear Energy* 77 (2015), 194–211.
- [13] Balázs Sonkoly, János Czentye, Robert Szabo, Dávid Jocha, János Elek, Sahel Sahhaf, Wouter Tavernier, and Fulvio Risso. 2015. Multi-domain service orchestration over networks and clouds: A unified approach. *ACM SIGCOMM Computer Communication Review* 45, 4 (2015), 377–378.
- [14] Thales Nicolai Tavares, Leonardo da Cruz Marcuzzo, Vinicius Fulber Garcia, Giovanni Venâncio de Souza, Muriel Figueredo Franco, Lucas Bondan, Filip De Turck, Lisandro Zambenedetti Granville, Elias Procópio Duarte Junior, Carlos Raniery Paula dos Santos, et al. 2018. Niep: Nfv infrastructure emulation platform. In *International Conference on Advanced Information Networking and Applications*. IEEE, Kraków, Poland, 173–180.