

Uma Solução Flexível e Personalizável para a Composição de Cadeias de Função de Serviço

Vinícius F. Garcia¹, Marcelo Caggiani Luizelli²,
Elias P. Duarte Júnior¹, Carlos Raniery P. dos Santos³

¹Universidade Federal do Paraná (UFPR)
Caixa Postal 19018 81531-990, Curitiba - PR

²Universidade Federal do Pampa (UNIPAMPA)
Av. Tiaraju, 810 - Ibirapuitã, Alegrete - RS

³Universidade Federal de Santa Maria (UFSM)
Av. Roraima, 1000 - Camobi, Santa Maria - RS

vfgarcia@inf.ufpr.br, marceloluizelli@unipampa.edu.br,
elias@inf.ufpr.br, csantos@inf.ufsm.br

Abstract. *One of the greatest challenges of the Network Function Virtualization paradigm is related to virtualized services deployment (composition, embedding, and scheduling). In particular, the service function chain composition process is currently executed by solutions that are static and do not present customizable operational behavior and evaluation metrics. This paper proposes a new service composing solution that is able to deal with multiple requirements from network operators. An adaptative methodology is employed in order to evaluate an elastic and heterogeneous metric set, conciliating different granularities and semantics. Finally, results obtained on an experimental environment show the feasibility of the proposed solution for specific services composition.*

Resumo. *Um dos principais desafios do paradigma de Virtualização de Funções de Rede reside na implantação de serviços virtualizados (composição, integração e execução). Em especial, a composição de cadeias de função de serviço é atualmente realizada por soluções estáticas e não personalizáveis em relação aos seus comportamentos operacionais e métricas de avaliação. Este artigo propõe uma nova solução de composição capaz de tratar requisitos específicos dos operadores de redes. Para isso, uma metodologia adaptativa é adotada para avaliar um conjunto elástico de métricas heterogêneas, conciliando diferentes granularidades e semânticas. Finalmente, resultados obtidos em um ambiente experimental demonstraram a viabilidade da solução proposta na composição de serviços específicos.*

1. Introdução

As infraestruturas de telecomunicações dependem de diferentes Funções de Rede (*Network Functions* - NF) para operarem. Atualmente, essas NFs são majoritariamente implementadas em equipamentos dedicados – chamados de *physical appliances* – capazes de identificar, tratar e executar protocolos e serviços. Porém, tais equipamentos tipicamente apresentam alto custo financeiro, baixa escalabilidade e processos de

manutenção complexos, contribuindo com o problema conhecido como ossificação da Internet [Handley 2006]. Nesse contexto, o paradigma de Virtualização de Função de Rede (*Network Function Virtualization* - NFV) surge objetivando, através do uso de técnicas de virtualização já existentes, instanciar funções de rede sobre um plano de *software* (*Virtualized Network Functions* - VNF) em equipamentos de propósito geral [NFVISG 2012]. Dessa forma, são esperados como benefícios a redução de custos financeiros e a flexibilização do monitoramento e da manutenção de NFs, além da simplificação do processo de desenvolvimento e implantação de novos serviços de rede criados a partir de uma ou mais funções virtualizadas.

O paradigma NFV também prevê a criação de serviços de rede através da conexão de diferentes funções em uma estrutura de encaminhamento e processamento de tráfego conhecida como Cadeia de Funções de Serviço (*Service Function Chain* - SFC) [Quinn and Nadeau 2015]. A implantação de um serviço de rede em um ambiente virtualizado acontece a partir de uma série de etapas inter-relacionadas que constituem o processo de implantação. Resumidamente, a implantação em NFV é chamada de Alocação de Recursos (NFV *Resource Allocation* - NFV-RA) [Herrera and Botero 2016] e é dividida em três etapas: composição (*composition*), integração (*embedding*) e agendamento de execução (*scheduling*). Assim, os desafios particulares associados a cada etapa do NFV-RA vêm sendo extensivamente explorados pela comunidade acadêmica, resultando em soluções que são empregadas no cumprimento de cada uma delas [Gil-Herrera and Botero 2017, Ocampo et al. 2017, Baek et al. 2017, Allybokus et al. 2018, Faraci et al. 2016, Kulkarni et al. 2017].

Porém, em geral, a implantação é realizada considerando apenas a topologia da rede, os recursos físicos disponíveis e algumas outras características pre-definidas. Assim, aspectos das próprias cadeias de serviço (*e.g.*, heterogeneidade de funções de rede, simetria de conexões, compartilhamento de funções virtualizadas) e possíveis necessidades específicas a serem definidas pelos clientes e/ou operadores de rede são desconsideradas. Em especial, soluções de composição existentes [Mehraghdam et al. 2014, Dräxler and Karl 2017, Gil-Herrera and Botero 2017, Ocampo et al. 2017, Wang et al. 2017] apresentam capacidade de análise limitada já que operam sobre um conjunto de métricas de avaliação rigidamente programado e imutável. Entre as métricas comumente observadas nessas soluções estão a taxa de tráfego, requisitos de banda e recursos computacionais, tamanho da cadeia de serviço e níveis de prioridade definidos para funções de rede. Apesar de formarem um conjunto significativo, limitar as soluções a essas métricas leva a uma execução inflexível, restritiva e sem suporte a personalização da etapa de composição. Dessa forma, o operador de rede é forçado a adequar suas necessidades às possibilidades de análise das soluções de composição existentes ou, para que o contrário ocorra, adaptar as mesmas em código.

Cenários de implantação são sobretudo formados pelos diferentes objetivos e necessidades dos operadores de rede (*e.g.*, redução da taxa de tráfego no serviço, do atraso de processamento, do consumo de memória, da quantidade de VNFs), por ambientes de virtualização heterogêneos (*e.g.*, centralizado, distribuído, privado, público) e por tipos variados de serviços (*e.g.*, roteamento, segurança, balanceamento de carga). Considerando a baixa capacidade adaptativa a esses cenários por parte das soluções de composição existentes, este trabalho apresenta o *CUstomizable Service COmposing* (CUSCO), uma

nova solução para a composição de serviços de rede virtualizados que aplica uma avaliação personalizada de métricas de interesse informadas pelo operador de rede. A solução CUSCO executa uma metodologia dinâmica e flexível de avaliação que possibilita a conciliação de múltiplas métricas com diferentes granularidades e semânticas, estas fornecidas através de um modelo de requisição próprio.

O restante deste artigo está estruturado como se segue: a Seção 2 examina e discute os principais trabalhos relacionados; a Seção 3 apresenta a nova proposta do CUSCO; a Seção 4 exhibe os testes realizados; e, finalmente, a Seção 5 conclui o trabalho.

2. Trabalhos Relacionados

Nesta seção são apresentados os principais trabalhos que abordam a etapa de composição de cadeias de serviço. Uma vez que diferentes topologias podem ser construídas para prover um mesmo serviço de rede através de um grupo comum de funções virtualizadas [Herrera and Botero 2016], soluções de composição capazes de determinar, analisar e qualificar as possíveis topologias se tornam necessárias. Dessa forma, as soluções de composição recebem como entrada documentos de requisição com informações necessárias à sua execução (*e.g.*, funções de rede que compõem o serviço, seus comportamentos operacionais e suas inter-relações). Ainda, recomendações de implantação (*i.e.*, dados resultantes de uma investigação prévia que sugerem as configurações ideais para implantação do serviço em diferentes ambientes de execução) e informações obtidas através de *benchmarkings* das funções de rede virtualizadas [Rosa et al. 2015] são frequentemente utilizadas como objetos de análise na etapa de composição.

Em [Mehrghdam et al. 2014] apresenta-se uma solução de composição que requer uma cadeia de serviço parcialmente ordenada, além da taxa de tráfego das funções de rede que a compõem (*i.e.*, o quociente entre a quantidade de dados recebidos e quantidade de dados encaminhados). Nela, uma função objetivo de minimização da taxa de tráfego agregada é adotada através de uma heurística de adiantamento de possíveis descartes de pacotes. Da mesma forma, a solução descrita em [Dräxler and Karl 2017] também opera sobre uma topologia parcialmente ordenada. Entretanto, esta utiliza-se de um modelo de Pareto e simulações sucessivas de integração objetivando, assim, determinar a composição capaz de otimizar a relação custo/benefício entre três métricas que formam sua função objetivo (tamanho da cadeia, taxa de tráfego agregada e necessidade de recursos computacionais).

As soluções presentes em [Ocampo et al. 2017] e [Gil-Herrera and Botero 2017] avaliam os possíveis relacionamentos entre as funções de rede destinadas a executar um serviço. A primeira apresenta uma função objetivo, modelada por Programação Linear Inteira, para minimizar a taxa de tráfego agregada e a quantidade de recursos computacionais necessários para a integração do serviço requisitado. Já a segunda utiliza uma função objetivo que busca assegurar que seja mínima a necessidade de largura de banda do serviço requisitado, calculando a mesma através de uma meta-heurística baseada em busca tabu. Por fim, em [Wang et al. 2017] uma solução de composição automática é descrita. Essa solução determina as dependências de um serviço baseada nas características operacionais de suas funções de rede, visando assim a criação de topologias válidas. Para tanto, diagramas Hasse são usados na definição de ordenamentos e conexões considerando uma função objetivo de prioridade de VNFs informadas pelo operador de rede.

A Tabela 1 sumariza as principais características dos trabalhos relacionados, nela o símbolo ✓ indica que a solução avaliada apresenta determinada capacidade de maneira completa, enquanto o símbolo * indica a parcialidade e ✗ a ausência da mesma.

	Trabalho Referenciado				
	[Mehraghdam]	[Dräxler]	[Ocampo]	[Gil-Herrera]	[Wang]
Modelo de Especificação de SFC	Gramática Livre de Contexto	Grafo de Serviço (Explícito)	Grafo de Serviço (Implícito)	Grafo de Serviço (Implícito)	Grafo de Serviço (Implícito)
Solução/Formulação do Problema	Heurística Redutiva	Análise de Pareto	ILP	Pesquisa Tabu	Algoritmo Hasse e Algoritmo de Prioridades
Função Objetivo	Estática	Estática	Estática	Estática	Estática
Métricas de Avaliação	Taxa de Tráfego	Taxa de Tráfego Requisitos de Recursos Tamanho da SFC	Requisitos de Banda	Requisitos de Banda	Prioridade
Suporte a Criação de Ramificações	*	✓	✓	✓	✓
Suporte a Ordenamento Parcial	✓	✓	✓	✓	✓
Suporte a Dependência de Ordem	✗	✗	✓	✓	✓
Suporte a Dependência de Acoplamento	✗	✗	*	*	*

Tabela 1. Sumarização de Características das Soluções de Composição

Os modelos de especificação de cadeias de serviço indicam o nível de dependência do operador de rede na etapa de composição. Modelos que especificam explicitamente uma topologia parcial do serviço estão nos trabalhos [Mehraghdam et al. 2014] e [Dräxler and Karl 2017]. Já modelos implícitos, onde apenas possíveis relacionamentos entre funções de rede são descritos ou inferidos automaticamente estão presentes em [Ocampo et al. 2017], [Gil-Herrera and Botero 2017] e [Wang et al. 2017]. Porém, a automatização completa da etapa de composição pode resultar em limitações indesejadas que incidem sobre os operadores de rede, ou seja, necessidades e conhecimentos específicos das funções que compõem o serviço e/ou do ambiente onde este será integrado (*e.g.*, fixação de posicionamento, pontos de monitoramento) são desconsiderados. Por outro lado, os modelos topológicos explícitos também apresentam limitações de especificação de algumas estruturas topológicas importantes, a exemplo disto estão a ausência de suporte para a definição de ramificações não terminais ([Mehraghdam et al. 2014]) e de dependências entre funções de rede ou domínios de execução ([Mehraghdam et al. 2014] e [Dräxler and Karl 2017]).

Como todos os trabalhos analisados utilizam funções objetivo estáticas, as soluções são otimizadas para obterem os resultados mais adequados (sejam estes locais ou globais) para um grupo fechado de métricas. Os trabalhos [Mehraghdam et al. 2014] e [Dräxler and Karl 2017] realizam a busca pela melhor composição dentro das combinações possíveis de seus segmentos parcialmente ordenados. Enquanto a primeira simplesmente ordena tais elementos a fim de minimizar a taxa de tráfego agregada, a segunda utiliza uma análise de Pareto para identificar as topologias que maximizam o custo/benefício entre as três métricas consideradas. Já os trabalhos [Ocampo et al. 2017], [Gil-Herrera and Botero 2017] e [Wang et al. 2017] não operam sobre a fixação e combinação de segmentos parcialmente ordenados, mas na criação das topologias por inteiro. Dessa forma, as soluções visam otimizar as métricas assistidas ao mesmo tempo que a topologia é formada, para isto diferentes abordagens podem ser adotadas, como a análise de cada transição através de PLI ([Ocampo et al. 2017]), a decisão da transição através de busca tabu ([Gil-Herrera and Botero 2017]) ou a análise de todas

as possíveis transições descritas em um diagrama Hasse ([Wang et al. 2017]).

As soluções de composição apresentadas no decorrer desta seção, independente de sua eficiência na determinação de resultados consistentes com seus objetivos, não permitem ao operador de rede requisitar avaliações personalizadas de serviços de rede no que diz respeito às suas funções objetivo. Apesar de haver indícios da extensibilidade de alguns métodos de composição, permitindo assim a avaliação de funções objetivo diferentes daquelas prefixadas nas soluções, não existe uma estratégia formal nem uma ferramenta genérica que possibilite que essa mudança seja feita sem a necessidade de manipulação de código fonte. Finalmente, como decorrência da construção de soluções estáticas de composição, os modelos de requisição de SFC apresentam, além de indicações da topologia parcial do serviço, apenas os valores de entrada para métricas de avaliação previamente definidas, não sendo prevista nenhuma forma de descrever as métricas em si e nem mesmo as operações de avaliação a serem aplicadas sobre elas.

3. *C*Ustomazable *S*ervice *C*OMposing

O *C*Ustomazable *S*ervice *C*OMposing¹ (CUSCO) é uma solução de composição de serviços de rede que preza pela flexibilidade de configuração e personalização de avaliação para gerar topologias absolutamente ordenadas válidas a partir de uma cadeia parcial e um conjunto de dependências previamente especificadas através de uma gramática livre de contexto. Essas informações são providas em conjunto com métricas e diretivas de avaliação em um documento estruturado pela linguagem de serialização de dados *YAML Ain't Markup Language* (YAML). Esta seção apresenta em detalhes o modelo de requisição adotado e a metodologia de composição executada pela solução.

3.1. Modelo de Requisição

O modelo de requisição YAML apresenta quatro objetos principais que contemplam metadados (METADADOS), além de informações sobre o serviço (SERVICO), funções de rede (ESPECIFICACAO) e função objetivo (FUNCAO_OBJ). Em especial, o objeto da função objetivo suporta a especificação de múltiplas métricas definidas como uma coleção de sub-objetos. Esses sub-objetos informam, além de uma identificação única, o peso da métrica para a avaliação global (um valor real [0;1], sendo o somatório deste atributo entre todas as métricas igual a um), um valor inicial de entrada, as operações de avaliação (operações matemáticas), a forma de atualização após cada iteração avaliativa e o objetivo de avaliação (maximização ou minimização). A função objetivo é aplicada à topologia inserida no objeto de serviço e é calculada considerando os dados de *benchmarking* das métricas e expectativas de divisão de tráfego (na ocorrência de cada ramificações na cadeia) presentes no objeto de especificação. A Figura 1 ilustra, em alto nível, as chaves que compõem o modelo de requisição adotado.

A topologia do serviço parcialmente ordenado deve ser construída através da gramática livre de contexto denominada *Service ChAin Grammar* (SCAG), desenvolvida como parte deste trabalho. A formalização da gramática proposta é realizada através da quádrupla $SCAG = (\nu, \tau, \rho, \varsigma)$. Nesse caso, ν representa o conjunto de símbolos não terminais (INICIO, BLOCOOP, BLOCOOPR, BLOCOTP, ORDEMP, EXCECAO, RAMIFICACAOT, PRAMIFICACAOT, RAMIFICACAONT, RAMIFICACAONTINT,

¹Disponível em <https://github.com/ViniGarcia/NFV-FLERAS>

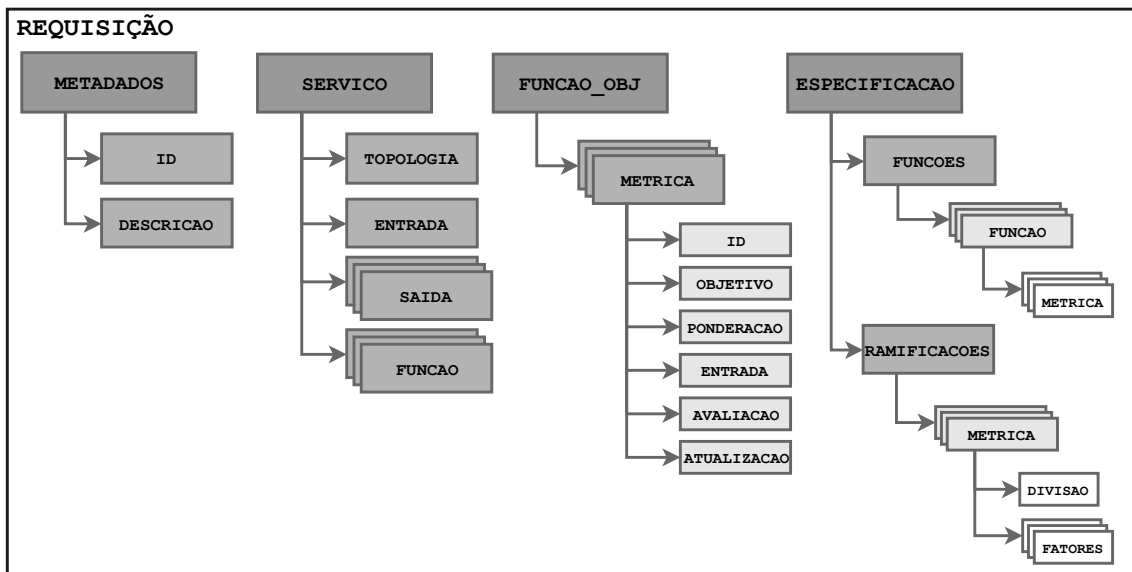


Figura 1. Modelo de Requisição CUSCO

PRAMIFICACAONT, PFUNCI, FUNCI, FUNC, NS, DOMINIO) e τ o conjunto de símbolos terminais (símbolos de controle - '<', '>', '{', '}', '(', ')', '[', ']', '/', '*', 'NE' - e símbolos variáveis - presentes nas regras de produção indicadas por FUNC, NS e DOMINIO) que formam a gramática. Esses símbolos são utilizados nas regras de produção definidas por ρ e apresentadas na Figura 2. Por fim, ς representa a regra de produção inicial da gramática (INICIO).

- 1 **INICIO** → 'NE' BLOCOOP
- 2 **BLOCOOP** → RAMIFICACAOT | RAMIFICACAONT | BLOCOTP BLOCOOP | BLOCOTP NS
- 3 **BLOCOOPR** → RAMIFICACAONTINT | BLOCOTP BLOCOOPR | BLOCOTP
- 4 **BLOCOTP** → ORDEMP | FUNCI
- 5 **ORDEMP** → '[' FUNCI PFUNCI ']' EXCECAO | '[' FUNCI PFUNCI ''
- 6 **EXCECAO** → '(' FUNC FUNC ')' EXCECAO | '(' FUNC FUNC '' | '(' FUNC FUNC '*' ')' EXCECAO | '(' FUNC FUNC '*' ''
- 7 **RAMIFICACAOT** → BLOCOTP '{' BLOCOOP PRAMIFICACAOT '}'
- 8 **PRAMIFICACAOT** → '/' BLOCOOP PRAMIFICACAOT | '/' BLOCOOP
- 9 **RAMIFICACAONT** → BLOCOTP '{' BLOCOOPR PRAMIFICACAONT '}' BLOCOOP
- 10 **RAMIFICACAONTINT** → BLOCOTP '{' BLOCOOPR PRAMIFICACAONT '}' BLOCOOPR
- 11 **PRAMIFICACAONT** → '/' BLOCOOPR PRAMIFICACAONT | '/' BLOCOOPR
- 12 **PFUNCI** → FUNCI PFUNCI | FUNCI
- 13 **FUNCI** → FUNC '<' DOMINIO '>' | FUNC
- 14 **FUNC** → 'FNC1' | 'FNC2' | 'FNC3' | ... | 'FNCn'
- 15 **NS** → 'NS1' | 'NS2' | 'NS3' | ... | 'NSn'
- 16 **DOMINIO** → 'DOM1' | 'DOM2' | 'DOM3' | ... | 'DOMn'

Figura 2. Regras de Produção da SCAG

A SCAG permite a criação de cadeias de serviço contendo tanto funções de rede absolutamente ordenadas, quanto em segmentos parcialmente ordenados delimitados pelos símbolos '[' e ']'. Sobre os segmentos parcialmente ordenados podem incidir dependências inseridas entre os símbolos '(' e ')', estas podem ser de ordem (precedência

em qualquer posição, sem o modificador '*' ou de acoplamento (precedência em primeira posição anterior, com o modificador '*'). Também, cada função de rede pode conter uma dependência de infraestrutura indicando, entre os símbolos '<' e '>', o domínio de execução da mesma. Além disso, ramificações terminais (*i.e.*, ramos irmãos) e não terminais (*i.e.*, com intersecção de ramos irmãos em uma função comum) podem ser especificadas entre os símbolos '{' e '}', com ramos separados pelo símbolo '/'. Finalmente, o modelo assume a existência de um único nó de entrada de dados e diversos nós de saída, além de encaminhamento unidirecional de tráfego. Todas as características relatadas são analisadas pela solução de composição em diferentes etapas do seu processamento.

3.2. Metodologia de Composição

A metodologia de composição CUSCO é dividida em dois procedimentos: expansão de topologias e avaliação de topologias. A expansão de topologias determina, através da resolução de ordenamentos parciais e reduções de ramificações, um conjunto de composições candidatas capazes de prover o serviço requisitado. Já a avaliação de topologias identifica e processa as métricas presentes na requisição, constrói funções parciais de avaliação e uma função objetivo geral e as utiliza para avaliar cada uma das composições candidatas, atribuindo a elas um Índice de Adequabilidade Topológica (IAT) que caracteriza uma composição candidata em relação às demais considerando a função objetivo geral. O IAT é então empregado no ranqueamento das composições candidatas, possibilitando que a solução sugira uma ou mais organizações topológicas a serem adotadas nas etapas seguintes da implantação do serviço.

3.2.1. Expansão de Topologias

O procedimento de expansão de topologias é subdividido em dois passos: resolução de ordenamentos parciais e redução de ramificações. Esses passos recebem como argumentos os dados fornecidos pelo objeto de serviço presente em uma requisição. No objeto os atributos ENTRADA, SAIDA e FUNCAO são empregados para distinguir os identificadores presentes na topologia requisitada (atributo TOPOLOGIA), além de serem diretamente usados na validação da mesma. O atributo TOPOLOGIA, por sua vez, fornece a cadeia de serviço parcialmente ordenada e informações extras que possibilitam a execução da expansão (*i.e.*, dependências, organização e ordenamentos topológicos), a execução desses passos resulta em um conjunto de composições candidatas absolutamente ordenadas que implementam o serviço requisitado.

A resolução de ordenamentos parciais é processada analogamente a uma permuta com restrições onde as restrições são as dependências de ordem e acoplamento. Dependências de ordem reduzem o número de possíveis posições de uma função de rede, sendo o tamanho desta redução relativa à quantidade de posições que sucedem aquela ocupada pela segunda função participante da dependência. Já uma dependência de acoplamento implica na fixação da primeira função na posição logo anterior a segunda, sendo ambas compreendidas como um bloco indivisível e tratadas como uma única função. A solução gera todas as permutas possíveis para um segmento parcialmente ordenado e verifica a validade das mesmas (*i.e.*, força bruta). Assim, a quantidade total de ordenamentos absolutos analisados para um segmento parcialmente ordenado é determinado pelo fatorial do número de funções de rede presentes no mesmo, enquanto a quantidade

de composições válidas ao final deste passo corresponde ao produto entre o número de ordenamentos absolutos válidos gerados em cada segmento processado.

A redução de ramificações consiste no processamento dos ramos de uma topologia absolutamente ordenada para detectar segmentos que podem ser alocados em uma seção comum antes do início ou após a conclusão de uma ramificação em análise. Ademais, segmentos equivalentes localizados em posições intermediárias dos ramos também podem ser unidos. Para isso, uma ramificação qualquer é decomposta em duas ramificações, uma de característica não terminal, estabelecida do ponto inicial da ramificação original até sua conclusão que ocorre antes do início do segmento a ser unificado (sendo a primeira função de rede deste segmento o ponto de intersecção dos ramos), e outra com a mesma característica da ramificação original, iniciada após o segmento unificado e perpetuada até a conclusão da ramificação original, ou seja, até os pontos de saída no caso de uma ramificação terminal ou até o elemento operacional de intersecção no caso de uma ramificação não terminal.

A solução realiza a redução de ramificações em segmentos iniciais e finais dos ramos, desconsiderando o caso de segmentos intermediários. As composições candidatas retornadas pela resolução de ordenamentos parciais são examinadas exaustivamente para identificar possíveis reduções em suas ramificações. Uma ramificação pode ser reduzida quando existem funções comuns a todos os seus ramos (tanto em tipo quanto em posição), além de, quando presentes, manifestarem dependências de infraestrutura compatíveis. Assim, a quantidade de composições candidatas geradas pela redução de uma ramificação é igual ao somatório do total de funções de rede presentes nos segmentos comuns máximos inicial e final (*i.e.*, maior conjunto de funções análogas entre os ramos), já que estes podem ser decompostos em subsegmentos menores. Também, a quantidade de novas composições resultantes desse passo quando aplicado a uma composição candidata é igual ao produtório do tamanho dos segmentos comuns máximos de todas as ramificações reduzidas, uma vez que estas reduções podem ocorrer simultaneamente.

3.2.2. Avaliação de Topologias

A avaliação de topologias é o segundo procedimento executado pela solução de composição e consiste do processamento das composições candidatas de um serviço de rede em vista das métricas, operações, ponderações e demais características informadas no objeto de função objetivo presente na requisição de um serviço. Esse procedimento é dividido em três passos: o primeiro compreende a definição de funções parciais e a avaliação das funções de rede e demais estruturas e nós especificados nas composições candidatas; o segundo realiza a depuração dos resultados parciais, os padronizando em um intervalo comum de valores; por fim, o terceiro passo utiliza os resultados padronizados para gerar um IAT destinado a cada composição candidata avaliada, possibilitando o ranqueamento das mesmas. As notações que formalizam as operações realizadas durante este procedimento são sumarizadas pela Tabela 2.

O primeiro passo trata da avaliação individual das composições candidatas, iterando sobre as posições de suas topologias para agregar resultados de avaliação. Para isso, inicialmente, funções parciais precisam ser construídas. Uma função parcial destina-se à avaliação das composições candidatas em relação a uma única métrica requisitada. Dessa

Notação	Descrição
CET	Conjunto de composições candidatas (topologias absolutamente ordenadas) gerado durante o passo de expansão de topologias.
CC	Composição candidata composta por funções de rede e nós de borda absolutamente ordenados.
FNC	Função de rede descrita em um documento de requisição.
P_{mtc}	Peso da métrica "mtc".
E_{mtc}	Valor de entrada da métrica "mtc".
FNC_{mtc}	Valor da métrica "mtc" informado para a função de rede "FNC".
$A_{mtc}(E_{mtc}, FNC_{mtc})$	Operação de avaliação da métrica "mtc" aplicada sobre a sua entrada "E" e o valor da mesma métrica colhido para uma função "FNC".
$AVA_{mtc}^{CC CET}$	Resultado da avaliação da função parcial referente a métrica "mtc" para a composição candidata "CC" pertencente a "CET".
$DA_{mtc}^{CC CET}$	Distância absoluta máxima entre o resultado da avaliação da composição candidata "CC" e o conjunto de resultados de avaliação das composições candidatas existentes em "CET" pela métrica "mtc".
$NORM_{mtc}^{CC CET}$	Normalização do resultado da avaliação da composição candidata "CC" em relação ao conjunto de resultados de avaliação das composições candidatas existentes em "CET" pela métrica "mtc".
$RPP_{mtc}^{CC CET}$	Resultado parcial padronizado da composição candidata "CC" em relação ao conjunto de resultados de avaliação das composições candidatas existentes em "CET" pela métrica "mtc".
IAT_{CC}^{CET}	Índice de adequabilidade topológica da composição candidata "CC" em relação ao conjunto de composições candidatas "CET".

Tabela 2. Notações de Formalização da Avaliação de Topologias

forma, cada uma das métricas presentes no objeto de função objetivo de uma requisição é transcrita para uma função parcial a ser calculada. O cálculo das funções parciais retornam valores agregados brutos que consistem do somatório dos resultados de avaliação de cada função operacional, estes devem estar disponíveis para todas as composições candidatas do conjunto retornado pelo primeiro procedimento antes da execução próximo passo. A Equação 1 generaliza o cálculo e o resultado de uma função parcial aplicada sobre funções de rede de uma composição candidata.

$$AVA_{mtc}^{CC CET} = \sum_{FNC}^{CC} A_{mtc}(E_{mtc}, FNC_{mtc}) \quad (1)$$

Ainda, o cálculo de uma função parcial é sucedido pela atualização das entradas e pela definição, caso necessária, de novos fluxos de execução, concluindo uma iteração avaliativa. Logo após a avaliação das métricas, estas são atualizadas de acordo com a sua própria operação de avaliação, substituindo a entrada pelo seu resultado, ou, da mesma forma, pelo resultado de avaliação de outra métrica presente na função objetivo. Também, uma terceira opção é a manutenção de um valor de entrada estático. Em seguida, o fluxo padrão de iterações, que consiste da avaliação de posições subsequentes da cadeia de serviço, pode ser alterado na eminência do início de uma ramificação. Nesse caso, a solução avalia todos os ramos (*i.e.*, do ponto inicial até o ponto de intersecção, onde ocorre uma agregação das entradas, ou nó de saída, encerrando a avaliação) através de uma busca em profundidade. Com isso, a entrada das métricas é atualizada por uma

operação de divisão de tráfego sobre fatores determinados para cada ramo (objeto de especificação), encaminhando estes valores ao início do novo fluxo de execução.

O segundo passo recebe os resultados provenientes da avaliação das composições candidatas pelas funções parciais de todas as métricas que compõem a função objetivo, os organizando no intervalo contínuo padrão $[0;1]$. A padronização do resultado de avaliação de uma métrica sobre uma composição candidata consiste em determinar a representatividade deste em relação aos demais resultados obtidos para a mesma métrica, mas provenientes de composições candidatas diferentes. Assim, o conjunto contínuo mínimo que compreende a totalidade dos resultados de avaliação de uma métrica, iniciado pelo resultado de Borda Inferior (BI) e encerrado no resultado de Borda Superior (BS) $[BI;BS]$, deve ser mapeado ao intervalo contínuo $[0;1]$ de forma que zero represente BI e um represente BS . Para isso, inicialmente, o intervalo $[BI;BS]$ deve ser deslocado de modo que a sua origem se iguale a origem do intervalo objetivado (*i.e.*, zero), sendo necessária a transcrição dos resultados para o intervalo de distância absoluta máxima $[0;BS - BI]$ utilizando a Equação 2.

$$DA_{mtc}^{CC\ CET} = AVA_{mtc}^{CC\ CET} - BI \quad (2)$$

Em seguida, o processo de normalização transcreve os valores do intervalo de distância absoluta máxima $[0;BS - BI]$ de cada uma das métricas avaliadas para o intervalo padrão $[0;1]$. Isso é alcançado pelo cálculo da razão entre a distância absoluta verificada para as composições candidatas ($DA_{mtc}^{CC\ CET}$) e a distância absoluta máxima ($BS - BI$), como demonstra a Equação 3. Como exceção, a Equação 4 é empregada em cenários onde todas as composições candidatas apresentam o mesmo resultado para uma mesma métrica avaliada ($BI = BS$). O processo de normalização indica a proximidade de um dado resultado à borda superior, assim este pode ser naturalmente usado como o resultado parcial padronizado para métricas cujo objetivo é sua maximização (Equação 5). Por outro lado, métricas com o objetivo de minimização devem ter suas normalizações complementadas para a geração dos resultados parciais padronizados, como mostra a Equação 6, para que revelem a sua proximidade à borda inferior e possam ser analisados da mesma maneira e conjuntamente a métricas com objetivo de maximização.

$$NORM_{mtc}^{CC\ CET} = \frac{DA_{mtc}^{CC\ CET}}{DA_{mtc}^{CET}} \quad (3)$$

$$NORM_{mtc}^{CC\ CET} = 1 \quad (4)$$

$$RPP_{mtc}^{CC\ CET} = NORM_{mtc}^{CC\ CET} \quad (5)$$

$$RPP_{mtc}^{CC\ CET} = 1 - NORM_{mtc}^{CC\ CET} \quad (6)$$

Após a padronização dos resultados, os IAT das composições candidatas podem ser calculados. O IAT é um valor contido no intervalo contínuo $[0;1]$ e indica a proximidade da avaliação geral da composição candidata ao resultado ótimo considerando todos aqueles recuperados no conjunto de composições avaliadas sobre todas as métricas requisitadas. Assim, uma composição adquire o maior IAT possível (*i.e.*, um) quando esta apresenta os resultados de borda inferior para todas as métricas com objetivo de minimização e resultados de borda superior para todas as métricas cujo objetivo é sua maximização.

De maneira prática, uma vez que os resultados já se encontram padronizados no intervalo de valores usado para o IAT, a sua definição consiste do somatório ponderado de todos os resultados parciais padronizados colhidos para uma determinada composição candidata (Equação 7), sendo a função objetivo adotada pela solução definida pela maximização (*MAX*) do IAT (Equação 8).

$$IAT_{CET}^{CC} = P_{mtc_1} * RPP_{mtc_1}^{CC \ CET} + \dots + P_{mtc_n} * RPP_{mtc_n}^{CC \ CET} \quad (7)$$

$$MAX(IAT_{CC}^{CET}), \forall CC \in CET \quad (8)$$

Finalmente, uma vez que a solução de composição desenvolvida é hierarquizada em uma série de procedimentos e passos que, por mais que sejam interconectados, apresentam uma estrutura de execução independente e podem ser alterados ou substituídos desde que mantenham o modelo padrão de geração de dados. Em outras palavras, apesar de haver uma dependência relacionada à sequência de processamento entre procedimentos e passos, não há influência entre eles quanto ao algoritmo aplicado para a geração de seus dados de saída. Dessa forma, heurísticas podem ser utilizadas no primeiro procedimento sem que modificações sejam necessárias no segundo procedimento, da mesma forma que as funções parciais podem ser alteradas sem que haja necessidade de modificações na função objetivo geral da solução.

4. Experimentação

Para avaliar a solução proposta, um serviço de segurança, controle de conteúdo e balanceamento de carga foi desenvolvido como parte deste trabalho. O serviço usa um total de sete funções de rede virtuais, sendo duas implementadas em *Click Modular Router* (Filtro de Protocolo/Porta - FPP; Classificador de Tráfego - CT) e outras cinco implementadas em Python 3 (Inspetor de Assinaturas em Pacotes HTTPS - IAPH; Inspetor de Conteúdo de Pacotes HTTP - ICPH; Filtro de Marcação - FM; Sistema de Prevenção de Intrusão HTTP/S - SPIH; Balanceador de Carga - BC). A cadeia de serviço parcialmente ordenada especificada com o modelo SCAG ("NE FPP CT { [IAPH FM SPIH] (IAPH FM*) / [ICPH FM SPIH] (ICPH FM*) } BC NS") balanceia requisições HTTP e HTTPS entre clientes e servidores, sendo as HTTP direcionados a análise de conteúdo e as HTTPS a análise de assinaturas, além de descartar tráfego com características diferentes de tais aplicações (TCP/IP, porta 80 para HTTP e porta 443 para HTTPS), com conteúdos ilegais ou reconhecidamente maliciosos pelo sistema de prevenção de intrusão.

A função objetivo empregue busca minimizar três métricas: taxa de tráfego, taxa de atendimento a requisições e tamanho da topologia. A taxa de tráfego consiste de um valor percentual, com entrada iniciada em 1 (100% do tráfego) e avaliação feita através da multiplicação desta pela taxa de tráfego de cada função iterada. A taxa de atendimento a requisições traduz-se em um número inteiro que indica a expectativa do máximo de requisições recebidas por segundo, sendo sua entrada igual a 100 e avaliação realizada através da diferença deste valor pela capacidade de atendimento a requisições por segundo das funções avaliadas. Por fim, o tamanho da cadeia assume um valor inteiro com entrada igual a 0, sendo uma unidade adicionada a ela a cada função iterada. A taxa de tráfego e capacidade de atendimento a requisições HTTP/S foram definidas por *benchmarking* das funções de rede. Além disso, todas as métricas apresentam o mesmo peso

(*i.e.*, ponderação 0,3333) e não houve preferência na divisão de tráfego no ponto de origem da única ramificação presente na topologia, especulando a equivalência entre tráfego HTTP e HTTPS que resulta em uma distribuição igualitária entre os seus ramos.

O processamento da cadeia de funções parcialmente ordenada do serviço de teste pela solução CUSCO resultou em nove composições candidatas. Dessas, três foram escolhidas para experimentação prática em um ambiente virtualizado real e são referenciadas pelos seus índices: IAT 0,666 ("NE FPP SPIH CT {IAPH / ICPH} FM BC NS"), IAT 0,500 ("NE FPP CT {SPIH IAPH FM / SPIH ICPH FM} BC NS") e IAT 0,327 ("NE FPP CT {IAPH FM / ICPH FM} SPIH BC NS"). A IAT 0,666 foi escolhida por apresentar o maior índice geral calculado, este oriundo de sua cadeia de tamanho mínimo (sete instâncias) e da menor taxa de tráfego constatada devido a antecipação da função SPIH. Entretanto, a IAT 0,666 obteve a pior taxa de atendimento a requisições visto que a existência de apenas uma instância da função SPIH no segmento principal pode se converter em um gargalo de processamento. A IAT 0,500 foi adotada por ser a composição com melhor índice geral cujo maximiza o RPP da taxa de atendimento a requisições, isso acontece pela replicação de instâncias das funções de rede nos ramos da topologia, ocasionando também, como consequência, o pior resultado de tamanho da topologia e resultados intermediários para a taxa de tráfego. Já a escolha pela IAT 0,327 é estimulada devido ao pior índice geral constatado, obtendo resultados intermediários para tamanho e taxa de atendimento a requisições, além do pior resultado de taxa de tráfego dada a postergação máxima das funções que descartam pacotes indesejados (FM e SPIH).

A implantação das composições candidatas selecionadas usou duas máquinas físicas para prover um ambiente virtualizado: a hospedeira dos clientes do serviço HTTP/S (8GB RAM DDR3, Core I3 4010U, Ubuntu 14.04 e virtualizador KVM); e a hospedeira do serviço de rede e dos servidores HTTP/S (8 GB RAM DDR3, Core I5 3330, Debian 8 e virtualizador KVM). Todas as máquinas virtuais (clientes, função de rede e servidores) foram instanciadas com 512 MB RAM e 1 núcleo virtual de processamento. As máquinas físicas são conectadas por interfaces Gigabit Ethernet e as virtuais por pontes, clientes e servidores encontram-se conectados na mesma rede /24. Nesse cenário são instanciadas dois servidores (ambos provendo serviços HTTP e HTTPS) e quatro clientes, dois para requisição de serviços HTTP e dois para HTTPS. Um cliente de cada aplicação é responsável por enviar requisições legítimas, enquanto o outro envia requisições maliciosas (pacotes anômalos de 1450 bytes, a uma taxa individual de 10Mbps) para indisponibilizar o serviço. Clientes e servidores são intermediados pelo serviço de teste. O ponto inicial da avaliação é o começo simultâneo do envio de requisições por parte dos clientes, já o ponto final remete a conclusão, com sucesso, de 5000 processos de requisições disparados pela ferramenta `http-perf`² em cada cliente legítimo.

A maior taxa de requisições HTTP/S atendidas por segundo, presentes na primeira e segunda colunas da Tabela 3, foi verificada para a composição candidata IAT 0,500. Isso acontece devido a presença de instâncias da função SPIH descartando tráfego malicioso no início dos ramos da topologia, além da alta capacidade de processamento de tráfego das funções anteriores a elas (FPP e CT, implementadas em *Click*), assegurando assim que funções com baixa capacidade de processamento não sejam oneradas com tráfego malicioso. Dessa forma, esse resultado confirma o melhor RPP da métrica de capacidade

²Disponível em <https://www.npmjs.com/package/http-perf>

	Maximização		Minimização	
	Taxa de Requisições HTTP Atendidas (req/s)	Taxa de Requisições HTTPS Atendidas (req/s)	Taxa de Tráfego (Mbps)	Número de Instâncias de Funções de Rede
<i>IAT 0,666</i>	2,4	2,5	37,9	7
<i>IAT 0,500</i>	20,4	13,0	59,6	9
<i>IAT 0,327</i>	0	0	-	8

Tabela 3. Resultados de Experimentação

de atendimento a requisições calculado pela solução de composição. A IAT 0,666 suporta o ataque recebido, entretanto a taxa de requisições atendidas é reduzida em relação a IAT 0,500. Essa situação decorre da existência de apenas uma instância da SPIH no segmento principal da cadeia, convertida em um gargalo durante a análise e descarte do tráfego malicioso. Já a composição IAT 0,327 aloca uma instância da SPIH na penúltima posição da cadeia, resultando no encaminhamento de tráfego malicioso para funções de baixa capacidade de processamento (IAPH e ICPH) e consequente indisponibilização do serviço.

A taxa de tráfego média no serviço (*i.e.*, quociente entre o total de tráfego processado pelas instâncias de função de rede e o tempo de execução até a conclusão da experimentação realizada), mensuradas para as composições candidatas que mantiveram o serviço disponível e exibidas na terceira coluna da Tabela 3, confirma a expectativa calculada pela solução de composição através do RPP da métrica taxa de tráfego. A composição IAT 0,666 adquire os melhores resultados para essa métrica devido à alocação das instâncias da função SPIH na segunda posição da cadeia, evitando que a classificação de tráfego (CT) ocorra sobre tráfego malicioso. Essa classificação, por outro lado, é necessária durante a utilização da composição IAT 0,500. Por fim, o número de instâncias de funções de rede virtualizadas presentes nas composições candidatas testadas, explícito na última coluna da Tabela 3, também confirma o RPP da métrica de tamanho calculada durante a determinação de seus respectivos índices. Dessa forma, é possível concluir que o comportamento operacional e características das cadeias de serviço no cenário e condições de teste demonstraram a viabilidade da solução CUSCO na composição e avaliação das mesmas.

5. Conclusão

A operacionalização de um serviço de rede no paradigma NFV passa pelo processo de implantação deste no ambiente virtualizado. Esse processo é constituído, principalmente, pelas etapas de composição, integração e agendamento de execução. Atualmente, a composição de cadeias de serviço é realizada por soluções estáticas e pouco personalizáveis em relação às suas funções objetivo e características de avaliação. Em vista disso, este trabalho apresentou o *CUsTomazable Service COmposing*, uma solução genérica para a composição de cadeias de função de serviço parcialmente ordenadas orientada por métricas e operações de avaliação personalizadas. Por fim, um serviço de rede experimental foi composto e seus resultados testados em um ambiente virtualizado real, demonstrando a viabilidade da solução desenvolvida. Em trabalhos futuros objetiva-se generalizar e aplicar a metodologia CUSCO nas demais etapas de implantação.

Referências

- Allybokus, Z., Perrot, N., Leguay, J., Maggi, L., and Gourdin, E. (2018). Virtual function placement for service chaining with partial orders and anti-affinity rules. *Networks*, 71(2):97–106.
- Baek, H., Jang, I., Ko, H., and Pack, S. (2017). Order dependency-aware service function placement in service function chaining. In *Information and Communication Technology Convergence (ICTC), 2017 International Conference on*, pages 193–195. IEEE.
- Dräxler, S. and Karl, H. (2017). Specification, composition, and placement of network services with flexible structures. *International Journal of Network Management*, 27(2):1963:1–20.
- Faraci, G., Lombardo, A., and Schembra, G. (2016). A processor-sharing scheduling strategy for nfv nodes. *Journal of Electrical and Computer Engineering*, 2016:1:1–1:1.
- Gil-Herrera, J. and Botero, J. F. (2017). A scalable metaheuristic for service function chain composition. In *Communications (LATINCOM), 2017 IEEE 9th Latin-American Conference on*, pages 1–6. IEEE.
- Handley, M. (2006). Why the internet only just works. *BT Technology Journal*, 24(3):119–129.
- Herrera, J. G. and Botero, J. F. (2016). Resource allocation in nfv: A comprehensive survey. *IEEE Transactions on Network and Service Management*, 13(3):518–532.
- Kulkarni, S. G., Zhang, W., Hwang, J., Rajagopalan, S., Ramakrishnan, K., Wood, T., Arumaithurai, M., and Fu, X. (2017). Nfvnic: Dynamic backpressure and scheduling for nfv service chains. In *ACM Special Interest Group on Data Communication*, pages 71–84. ACM.
- Mehraghdam, S., Keller, M., and Karl, H. (2014). Specifying and placing chains of virtual network functions. In *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on*, pages 7–13. IEEE.
- NFVISG, E. (2012). Network functions virtualization: White paper. Technical report, European Telecommunications Standards Institute.
- Ocampo, A. F., Gil-Herrera, J., Isolani, P. H., Neves, M. C., Botero, J. F., Latré, S., Zambenedetti, L., Barcellos, M. P., and Gasparly, L. P. (2017). Optimal service function chain composition in network functions virtualization. In *IFIP International Conference on Autonomous Infrastructure, Management and Security*, pages 62–76. Springer.
- Quinn, P. and Nadeau, T. (2015). Problem Statement for Service Function Chaining - RFC 7498. Request for comments, Internet Engineering Task Force.
- Rosa, R. V., Rothenberg, C. E., and Szabo, R. (2015). Vbaas: Vnf benchmark-as-a-service. In *Software Defined Networks (EWSDN), 2015 Fourth European Workshop on*, pages 79–84. IEEE.
- Wang, Y., Li, Z., Xie, G., and Salamatian, K. (2017). Enabling automatic composition and verification of service function chain. In *Quality of Service (IWQoS), 2017 IEEE/ACM 25th International Symposium on*, pages 1–5. IEEE.