

UNIVERSIDADE DO ESTADO DE SANTA CATARINA - UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT
MESTRADO EM COMPUTAÇÃO APLICADA

FELIPE RODRIGO DE SOUZA

ALOCÇÃO DE INFRAESTRUTURAS VIRTUAIS EM *DATA*
CENTERS* BASEADOS EM REDES DEFINIDAS POR *SOFTWARE

JOINVILLE

2017

FELIPE RODRIGO DE SOUZA

**ALOCÇÃO DE INFRAESTRUTURAS VIRTUAIS EM *DATA*
CENTERS BASEADOS EM REDES DEFINIDAS POR *SOFTWARE***

Dissertação submetida ao Programa de Pós-Graduação em Computação Aplicada do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, para a obtenção do grau de Mestre em Computação Aplicada.

Orientador: Dr. Guilherme Piêgas Koslovski

JOINVILLE

2017

Souza, Felipe Rodrigo de

Alocação de Infraestruturas Virtuais em *Data Centers* baseados em Redes Definidas por *Software*/ Felipe Rodrigo de Souza. – 2017

74 p. : il. ; 30 cm.

Orientador: Guilherme Piêgas Koslovski

Bibliografia: p.70-74

Dissertação (mestrado) - Universidade do Estado de Santa Catarina - UDESC, Centro de Ciências Tecnológicas, Programa de Pós-Graduação em Computação Aplicada, Joinville, 2017.

1. Sistemas Computacionais. 2. Infraestruturas Virtuais. 3. *Data Center*. 4. SDN. I. . Koslovski, Guilherme Piêgas. II. Universidade do Estado de Santa Catarina. Programa de Pós-Graduação Aplicada III. Título.

CDD 004.2– 23.ed.

**Alocação de Infraestruturas Virtuais em Data Centers Baseados em Redes
Definidas por Software**

por

Felipe Rodrigo de Souza

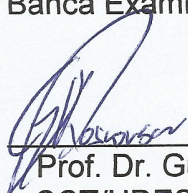
Esta dissertação foi julgada adequada para obtenção do título de

Mestre em Computação Aplicada

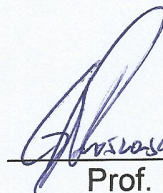
Área de concentração em "Ciência da Computação,
e aprovada em sua forma final pelo

CURSO Mestrado Acadêmico em Computação Aplicada
CENTRO DE CIÊNCIAS TECNOLÓGICAS DA
UNIVERSIDADE DO ESTADO DE SANTA CATARINA.

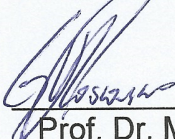
Banca Examinadora:



Prof. Dr. Guilherme Piegas Koslovski
CCT/UDESC (Orientador/Presidente)



Prof. Dr. Adriano Fiorese
CCT/UDESC



Prof. Dr. Marcos Dias de Assunção
ENS/INRIA

Joinville, SC, 07 de março de 2017.

Dedico este trabalho aos meus familiares, amigos, colegas e professores que me acompanharam e me deram forças nessa magnífica trajetória.

AGRADECIMENTOS

Gostaria de agradecer aos meus familiares que sempre me apoiaram e me incentivaram em qualquer escolha que eu fizesse. Gostaria de agradecer meus amigos, por terem me ajudado, me incentivado nos momentos ruins, e comemorado nos momentos bons. Gostaria de agradecer em especial ao Professor Guilherme Piêgas Koslovski, que por mais que diga que desorientava mais do que orientava, fez um excelente trabalho e me ensinou muito. E por fim gostaria de agradecer a Deus, que sem ele nada seria possível.

“Independentemente das circunstâncias,
devemos ser sempre humildes, recatados
e despidos de orgulho.”

Dalai Lama

RESUMO

Infraestruturas Virtuais (IVs), que são compostas por recursos computacionais e de comunicação, foram difundidas como uma possível solução para a evolução das redes e criação de novos serviços na Internet. Embora promissora, sua adoção enfrentou obstáculos, principalmente ocasionados pela rigidez no gerenciamento dos recursos de comunicação. Recentemente, o paradigma de redes definidas por *software* (SDN) popularizou a customização e a flexibilidade no gerenciamento devido a separação entre os planos de controle e dados. SDN fornece mecanismos para garantia de largura de banda e controle de latência nos equipamentos de rede, e não somente nas extremidades da infraestrutura. Ainda, através da separação dos planos de dado e controle, SDN permite redução da utilização da infraestrutura física. Entretanto, os benefícios introduzidos por SDN não são trivialmente aplicados para o provisionamento de IVs em *data centers* de nuvens computacionais. Existem lacunas na seleção e alocação de recursos de SDN para hospedar IVs. Os mecanismos para alocação necessitam de informações combinadas do plano de controle (e.g, políticas para isolamento de IVs, identificação de fluxos) e de dados (e.g, capacidade de armazenamento, número de entradas por fluxo) para encontrar uma solução eficiente. Diante dos benefícios e desafios introduzidos por SDN, é proposta uma formulação baseada em programação inteira mista para o problema de alocação de IVs em *data centers* de nuvem baseados em SDN. Posteriormente, algumas restrições em variáveis são relaxadas e uma heurística para aproximação dos resultados é discutida. Os resultados indicam que com SDN é possível reduzir a utilização da infraestrutura física respeitando a qualidade de serviço solicitada pelos clientes.

Palavras-chaves: infraestruturas virtuais, alocação, *data center*, SDN.

ABSTRACT

Virtual Infrastructures were disseminated as a potential solution for network evolution and creation of new services on the Internet. Although promising, its adoption faces many obstacles, mainly due to rigidity in the management of networking resources. Recently, the Software Defined Networks (SDN) paradigm has popularized the customization and flexibility in management by splitting control and data planes. Moreover, SDN provides mechanisms for bandwidth guarantee and latency control on network switches, not just in the endpoints. Through the separation of the data and control planes, SDN allows to reduce the data center usage. However, the benefits introduced by SDN are not trivially applied for virtual infrastructure embedding in cloud data centers. Allocation frameworks require combined information from the control plan (e.g., isolation policies, flows identification) and data (e.g., storage capacity, flow table configuration) to find a suitable solution. In this context, the present work proposes a formulation based on mixed integer programming to the virtual infrastructure allocation problem. The formulation comprises the main challenges related to SDN-based cloud data centers. Later, some constraints are relaxed composing a linear program, and a heuristic is introduced. Experimental results of the mechanism, termed QVIA-SDN, highlight that an allocation solution with SDN knowledge can reduce the data center usage and improve the quality-of-service perceived by hosted tenants.

Key-words: virtual infrastructures, allocation, data center, SDN.

LISTA DE ILUSTRAÇÕES

Figura 1 – Alocação de Infraestrutura Virtual (IV). Diversos equipamentos de um <i>data center</i> são selecionados para alocar uma requisição de IV.	21
Figura 2 – Infraestrutura de uma rede <i>Software-Defined Networking</i> (SDN).	24
Figura 3 – Alocação de IVs em provedores de nuvens baseados em SDN.	33
Figura 4 – Grafo aumentado combinando recursos físicos e virtuais.	37
Figura 5 – Distribuição de zonas e regiões no <i>data center</i> com topologia <i>fat-tree</i> .	50
Figura 6 – Taxa de aceitação da alocação de IVs.	52
Figura 7 – Distribuição Acumulada da latência média normalizada	54
Figura 8 – Variabilidade da latência média	55
Figura 9 – Fragmentação do <i>Data Center</i>	56
Figura 10 – Razão <i>cost-revenue</i>	57
Figura 11 – Tempo médio para alocação de requisições de IV	58
Figura 12 – Taxa de aceitação da alocação de IVs.	60
Figura 13 – Distribuição Acumulada da latência média normalizada	61
Figura 14 – Variabilidade da latência média	62
Figura 15 – Fragmentação do <i>data center</i>	63
Figura 16 – Razão <i>cost-revenue</i>	64
Figura 17 – Tempo médio para alocação de requisições de IV	65

LISTA DE TABELAS

Tabela 1 – Tabela de trabalhos relacionados.	28
Tabela 2 – Notação utilizada ao longo do artigo. i e j representam recursos virtuais, enquanto u e v são utilizados para recursos físicos.	31
Tabela 3 – Detalhamento dos recursos das instâncias $m3$ da Amazon EC2. . .	58
Tabela 4 – Publicações realizadas.	69

LISTA DE SIGLAS E ABREVIATURAS

IaaS *Infrastructure-as-a-Service*

LP *Linear Programming*

MV Máquina Virtual

MIP *Mixed Integer Program*

PaaS *Platform-as-a-Service*

NaaS *Network-as-a-Service*

QoS *Quality-of-Service*

SaaS *Software-as-a-Service*

SDN *Software-Defined Networking*

SLA *Service Level Agreement*

VPC *Virtual Private Clouds*

IV Infraestrutura Virtual

BDC Busca Determinística de Caminhos

TI Tecnologia da Informação

ONF *Open Network Foundation*

NC N-Camadas

SCL Sem Controle de Latência

NSDN Non-SDN

OSI *Open System Interconnection*

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVOS DO TRABALHO	16
1.1.1	Objetivos Específicos	16
1.2	PRINCIPAIS CONTRIBUIÇÕES	16
1.3	ORGANIZAÇÃO DO TRABALHO	17
2	REVISÃO DE LITERATURA	18
2.1	Provedores de Nuvens Computacionais IaaS	18
2.1.1	<i>Data centers</i>	19
2.1.2	Infraestruturas Virtuais	20
2.1.3	Gerenciamento de Infraestruturas Virtuais	21
2.2	<i>SOFTWARE-DEFINED NETWORKING</i>	23
2.2.1	<i>Conceitos e Definições</i>	23
2.2.2	Padrão OpenFlow	25
2.3	Alocação de IVs em SDN	26
2.4	TRABALHOS RELACIONADOS	27
2.4.1	Alocação de Recursos para Hospedar IVs	27
2.4.2	Provisionamento de IVs em ambiente SDN	29
2.5	CONSIDERAÇÕES PARCIAIS	29
3	MODELO ÓTIMO PARA ALOCAÇÃO DE INFRAESTRUTURAS VIRTUAIS EM <i>DATA CENTERS</i> SDN	31
3.1	FORMULAÇÃO DO PROBLEMA	31
3.1.1	<i>Data center</i> de um Provedor IaaS	32
3.1.2	Requisições de IVs	32
3.1.3	Alocando recursos físicos para hospedar IVs	33
3.1.4	Objetivos do provedor de nuvem <i>Infrastructure-as-a-Service</i> (IaaS)	34
3.2	MIP ÓTIMO PARA ALOCAÇÃO DE IVs COM REQUISITOS DE QoS	34
3.2.1	Seleção de candidatos para alocar recursos virtuais	35
3.2.2	Variáveis e objetivo	36
3.2.3	Restrições para alocação ótima	38
3.2.3.1	<i>Restrições de capacidade dos servidores e enlaces</i>	38
3.2.3.2	<i>Restrições de SDN</i>	38
3.2.3.3	<i>Restrições de fluxo de dados</i>	39
3.2.3.4	<i>Restrições de latência</i>	39

3.2.3.5	<i>Meta restrições e restrições binárias</i>	40
3.3	CONSIDERAÇÕES PARCIAIS	40
4	QOS-AWARE VI ALLOCATION ON SDN-BASED DATA-CENTERS	42
4.1	RELAXANDO AS VARIÁVEIS	42
4.2	REDUÇÃO DO NÚMERO DE CANDIDATOS FÍSICOS	42
4.2.1	Servidores e <i>switches</i>	42
4.2.2	Enlaces de comunicação	43
4.3	HEURÍSTICAS	43
4.3.1	QVIA-SDN	43
4.3.2	Busca Determinística de Caminhos	45
4.4	CONSIDERAÇÕES PARCIAIS	46
5	ANÁLISE EXPERIMENTAL	48
5.1	MÉTRICAS	48
5.2	CENÁRIOS DE SIMULAÇÃO	49
5.2.1	Topologia do <i>data center</i> de nuvem computacional	49
5.2.2	Requisições de IVs	49
5.3	Simulações	50
5.3.1	Simulação com cargas uniformemente distribuídas	51
5.3.1.1	<i>Especificação</i>	51
5.3.1.2	<i>Resultados</i>	51
5.3.2	Simulação baseada na instância <i>m3</i> da Amazon	54
5.3.2.1	<i>Especificação</i>	55
5.3.2.2	<i>Resultados</i>	59
5.3.3	Discussão e principais observações	62
5.4	CONSIDERAÇÕES PARCIAIS	66
6	CONCLUSÃO	67
6.1	Trabalhos Futuros	68
6.2	Trabalhos Publicados	68
	REFERÊNCIAS	70

1 INTRODUÇÃO

Atualmente, uma tendência presente em muitas organizações de pequeno e médio porte, é a de migração parcial ou completa dos serviços de Tecnologia da Informação (TI) para *data centers* remotos, acessíveis pela Internet, popularmente conhecidos como nuvens computacionais (STALLINGS, 2015). Essa tendência é motivada pelo provisionamento de recursos computacionais sob demanda, reduzindo os custos de manutenção e gerenciamento de TI. Ainda, a computação em nuvem pode não ser vista como uma revolução tecnológica, mas sim comercial, que introduziu mudanças no gerenciamento de recursos de TI. Sobretudo, o modelo *pay-as-you-go* proporcionou a redução de custos para adquirir e manter uma infraestrutura de *data center*.

Os recursos computacionais oferecidos pelos provedores de nuvem compreendem servidores, infraestrutura de comunicação, unidades de armazenamento, entre outros (HASHEM et al., 2015). Tais recursos são comercializados seguindo diversos modelos de serviço, como por exemplo *Software-as-a-Service (SaaS)*, *Platform-as-a-Service (PaaS)*, *Network-as-a-Service (NaaS)* e *Infrastructure-as-a-Service (IaaS)*, sendo os dois últimos o foco do presente trabalho, que disponibilizam recursos virtuais de processamento, armazenamento e rede. Porém, em IaaS o cliente não gerencia a infraestrutura física, tendo controle sobre os sistemas operacionais, unidades de armazenamento, aplicações hospedadas e em alguns casos um limite controlado sobre componentes de rede (e.g., *firewalls* e balanceadores de carga) (MELL; GRANCE, 2011).

Provedores de IaaS aplicam tecnologias de virtualização para disponibilizar Infraestruturas Virtuais (IVs) temporárias. As IVs são camadas lógicas provisionadas sobre a infraestrutura física, que coabitam e compartilham os recursos do *data center*. A virtualização permite a personalização do ambiente: cada cliente pode ter uma IV especializada de acordo com os requisitos das aplicações hospedadas. (MANVI; SHYAM, 2014; FISCHER et al., 2013; CHOWDHURY; BOUTABA, 2009).

O modelo de negócio para provisionamento de IVs tem um foco maior na reserva de recursos computacionais e de armazenamento, entretanto, é necessário destacar a importância que a infraestrutura de rede tem no desempenho da aplicação. Usualmente, aplicações hospedadas na nuvem geram grandes volumes de tráfego, destinando uma fatia do seu tempo de execução para a atividade de rede. Um exemplo seriam os *clusters* do Facebook, que despendem até 33% do seu tempo de execução para realização de transferências de dados (ROST; FUERST; SCHMID, 2015). Ou seja, se a rede não for considerada como um fator limitante e portanto estar de-

vidamente configurada, torna-se impossível superar os desafios de gerenciamento a fim de oferecer *Quality-of-Service (QoS)* para as aplicações hospedadas (HUU et al., 2011).

Nesse contexto, este trabalho tem como foco o provisionamento de IVs, tarefa executada pelo provedor IaaS, que por meio das tecnologias de virtualização fornecem recursos virtuais escaláveis, com garantia de um nível mínimo de qualidade de serviço. A *QoS* pode ser expressa através de uma largura de banda mínima, uma latência máxima tolerável, configuração de memória, processamento e armazenamento de Máquinas Virtuais (MVs), capacidade de *switches* virtuais, entre outros requisitos.

Dentre as tarefas gerenciais necessárias para o provisionamento de IVs, a alocação consiste em encontrar, dentro do *data center*, um subconjunto de recursos que respeita as capacidades solicitadas para os recursos virtuais. Esse processo essencial pertence a classe de problemas NP-Difícil, pois consiste na busca de um subgrafo, e se torna ainda mais complexo por representar um ambiente heterogêneo em termos de equipamentos de rede, compartilhado por diversas IVs, com solicitações individuais de capacidade virtual e garantias de *QoS* (FISCHER et al., 2013; HAIDER; POTTER; NAKAO, 2009; MANVI; SHYAM, 2014).

Diante da complexidade relacionada com o provisionamento de IVs, sobretudo respeitando requisitos de *QoS* informados por múltiplos clientes, *Software-Defined Networking (SDN)* surge como uma alternativa gerencial. Em suma, SDN é um paradigma que separa os planos de dados e controle. Essa separação, extrai o plano de controle para uma unidade externa, logicamente centralizada, que conhece e é responsável por gerenciar toda a infraestrutura de rede. Assim, é possível abstrair os equipamentos de rede no processo de alocação. Ou seja, antes era necessário distinguir equipamentos de diferentes fabricantes que possuem interfaces específicas de configuração, enquanto com SDN, todos podem ser considerados similares em termos de configuração (MCKEOWN et al., 2008). Enquanto outras alternativas falharam no gerenciamento de *QoS* de rede (KARAKUS; DURRESI, 2017), SDN se destaca por fornecer mecanismos para controle de largura de banda e latência de comunicação, adaptável aos *data centers* de nuvens computacionais.

Apesar do ambiente SDN fornecer benefícios, novos desafios de gerenciamento surgiram: atraso no encaminhamento de pacotes, compartilhamento de largura de banda, isolamento de *switches* e compartilhamento de tabelas de encaminhamento (SHERWOOD et al., 2009; SHERWOOD et al., 2010). Quanto a alocação de IVs, SDN aumenta a complexidade em 3 eixos: (i) Devido as limitações de capacidade das tabelas de fluxo dos *switches* SDN, novas restrições sobre as capacidades dos *switches* físicos são criadas. Eventualmente fluxos podem ser alocados no controlador. (ii) Por separar os planos de dado e controle e permitir a alocação de fluxos

tanto nos equipamentos de rede, como no controlador, há um aumento na latência de comunicação. (iii) Enquanto os mecanismos tradicionais de alocação de IaaS fornecem *best-effort* dos recursos de rede, SDN fornece garantias de *QoS* de rede, criando uma demanda de restrições para impor o *QoS* solicitado pelo cliente.

1.1 OBJETIVOS DO TRABALHO

O objetivo geral deste trabalho é propor um mecanismo para alocação de IVs, estabelecendo uma relação eficiente entre a redução da fragmentação e garantias de *QoS* para os clientes da nuvem.

1.1.1 Objetivos Específicos

Tendo em vista o objetivo geral, os seguintes objetivos específicos devem ser alcançados:

- propor e desenvolver um modelo ótimo para alocação de infraestrutura virtuais;
- relaxar variáveis do modelo ótimo, propor uma heurística para interpretar o resultado do relaxamento das variáveis e desenvolver um mecanismo para redução do espaço de busca;
- definir métricas de fragmentação, de *QoS* de latência e largura de banda e introduzi-las ao modelo; e
- realizar simulações que possibilitem avaliar a eficiência do modelo com base em métrica de fragmentação, *cost-revenue*, latência média, tempo médio de alocação e taxa de aceitação.

1.2 PRINCIPAIS CONTRIBUIÇÕES

O presente trabalho compreende três contribuições ao processo de alocação de IVs em ambientes SDN:

- A formulação de um *Mixed Integer Program* (MIP) ótimo para alocação de requisições *online* de IVs. A formulação considera os principais desafios de gerenciamento introduzidos por SDN, modelando controladores, latência de comunicação, largura de banda e restrições de capacidade dos *switches*.
- Uma abordagem baseada em *Linear Programming* (LP), obtido através do relaxamentos do domínio das variáveis do MIP. Uma heurística é aplicada para interpretação dos resultados, inovando por selecionar candidatos não só baseado

nas restrições do LP, mas também considerando as particularidades do ambiente SDN.

- Um *framework* para simulação da alocação de IVs de acordo com a formulação do modelo proposto.

Os resultados indicam que a alocação de IVs em um ambiente SDN melhora o QoS oferecido ao cliente, pois são respeitadas as restrições de largura de banda e latência, e portanto a aplicação do cliente não está sujeita a tantas variações na rede. Ainda, a fragmentação é reduzida permitindo a alocação de mais IVs, a razão *cost-revenue* se mantém equivalente, mas como a fragmentação foi reduzida, ainda é possível alocar mais IVs e permitir um aumento de tal razão.

1.3 ORGANIZAÇÃO DO TRABALHO

A estrutura deste trabalho segue: O Capítulo 2 visa detalhar os principais elementos da literatura relacionados e necessários para a compreensão deste trabalho. O Capítulo 3 contém a formulação do modelo proposto, bem como as técnicas de redução do espaço de busca, que incluem a seleção de candidatos e o relaxamento das variáveis, bem como as heurísticas de interpretação de resultados descritos no Capítulo 4. Uma avaliação do modelo proposto é realizada no Capítulo 5 sob dois diferentes cenários que buscam avaliar o desempenho do modelo tanto com cargas aleatórias como com cargas reais. Por fim, o Capítulo 6 apresenta as conclusões obtidas após a realização deste estudo.

2 REVISÃO DE LITERATURA

O presente capítulo apresenta a revisão de literatura essencial para compreensão do trabalho. Dentre os temas abordados estão *data centers* de nuvens computacionais e a definição de IVs (Seção 2.1). Ainda, os principais conceitos associados a SDN são discutidos (Seção 2.2). Posteriormente, os desafios relacionados com a alocação de IVs em *data centers* baseados em SDN são elencados (Seção 2.3). Após a discussão sobre os trabalhos relacionados (Seção 2.4), os elementos principais do capítulo são resumidos (Seção 2.5). O leitor já familiarizado com o tema pode seguir a leitura no Capítulo 3 que apresenta os detalhes do modelo proposto.

2.1 PROVEDORES DE NUVENS COMPUTACIONAIS IAAS

Computação em nuvem é um modelo computacional que permite acesso sob demanda a um conjunto de recursos computacionais compartilhados, que podem ser rapidamente provisionados. Esses recursos computacionais se estendem a servidores, recursos de rede, armazenamento entre outros (MELL; GRANCE, 2011).

Existem quatro modelos de desenvolvimento de nuvens: Privados, onde a nuvem é de uso exclusivo de uma única instituição ou organização. Comunitárias, que são de uso exclusivo de uma comunidade de consumidores de uma ou mais organizações que têm objetivos compartilhados. Públicas, são provisionadas para uso do público em geral. Estas são nuvens que são provisionadas por provedores de nuvens, como Amazon, Google, Microsoft, etc. E por fim, Híbridas, que são a composição de dois ou mais dos modelos anteriores (Privado, Comunitário e Públicas) (MELL; GRANCE, 2011). O modelo proposto nesse trabalho, pode ser aplicado em qualquer um dos modelos de desenvolvimento de nuvens, desde que a nuvem provisione IVs.

Provedor de nuvem é a entidade ou organização responsável por gerenciar a nuvem computacional, ofertando serviços sobre uma infraestrutura física composta por diversos servidores, *switches*, roteadores, e enlaces interconectando todos esses equipamentos. Além disso, o provedor é responsável por executar tarefas de gerenciamento e manutenção tanto dos servidores como da infraestrutura de rede. Para isso, o provedor faz uso de uma série de ferramentas de gerenciamento, redundância e segurança (STALLINGS, 2015). Ainda, para gerenciar recursos e serviços, o provedor de nuvem computacional faz uso de alguma ferramenta de gerenciamento como OpenStack, CloudStack, entre outros.

O provedor de nuvem comercializa os recursos de TI de diferentes formas e modelos. Especificamente, clientes de IaaS reservam recursos virtuais para hospedar

suas aplicações. Por exemplo, a Amazon comercializa o EC2 (*Elastic Cloud Computing*), que fornece diversos tipos e tamanhos de instâncias de MVs (T2, M4, M2, C4, etc), e o S3 (*Simple Storage Service*), que fornece o armazenamento de objetos, com uma interface para armazenar e recuperar qualquer volume de dados (AMAZON, 2017b). Concorrente da Amazon, o Google oferece o *Google Cloud Storage*, um serviço para armazenamento de objetos, e todos os tipos de dados não estruturados. Além disso, oferece serviços para provisionamento de IVs, como o *Google Compute Engine* e *Google Container Engine* (GOOGLE, 2017a).

2.1.1 *Data centers*

Provedores IaaS concentram seus recursos computacionais em *data centers*. Por definição, um *data center* se refere a qualquer conjunto de equipamentos de computação, sejam servidores, enlaces, *switches* e roteadores, que são gerenciados e de posse de uma organização (BENSON; AKELLA; MALTZ, 2010).

Data centers são desenvolvidos com diversas configurações, variando o propósito e o número de elementos (e.g., pequeno, médio e grande porte). Universidades e algumas empresas estão constantemente consolidando seus recursos de TI em *data centers* de pequeno e médio porte. Por outro lado, grandes organizações como Google, Microsoft e Amazon estão constantemente construindo *data centers* de grande porte geograficamente dispersos (BENSON; AKELLA; MALTZ, 2010). Por exemplo, apesar de o Google não disponibilizar informações a respeito do seu número de servidores, em 2011, era estimado que a companhia tinha aproximadamente 900.000 servidores (número que provavelmente aumentou nos últimos anos), enquanto a Microsoft opera mais de 100 *data centers* distribuídos em 17 países. Ainda, em 2013 o então CEO da companhia, Steve Ballmer, afirmou que os *data centers* da Microsoft totalizavam mais de 1 milhão de servidores (KNOWLEDGE, 2011; KNOWLEDGE, 2013; MICROSOFT, 2017). A Amazon, por sua vez, também não fornece informações a respeito do seu número de servidores, porém, opera em mais de 16 países, totalizando 42 zonas, compostas de um ou mais *data centers* (AMAZON, 2017a).

Todos esses recursos de servidores, *switches* e enlaces devem ser organizados de acordo com uma determinada arquitetura. Além de organizar os recursos as arquiteturas têm outros objetivos igualmente importantes, como alta disponibilidade, balanceamento de carga, entre outros objetivos. Um dos exemplos de arquitetura é a *fat-tree* (AL-FARES; LOUKISSAS; VAHDAT, 2008) que busca fornecer escalabilidade, fazendo uma melhor utilização da largura de banda nos *switches* de agregação com enlaces redundantes. Por sua vez, PortLan (MYSORE et al., 2009), é uma arquitetura baseada em *fat-tree* que propõe um esquema de endereçamento e roteamento IP tolerante a falhas. Outro exemplo, a arquitetura BCube (GUO et al., 2010), reduz a

utilização da infraestrutura física tanto de *switches* como servidores, por utilizar *servidores* tanto para hospedar aplicações como fazer o roteamento de rede.

Para todos os cenários, *frameworks* de gerenciamento de nuvens computacionais têm sido aplicados para controle interno dos recursos físicos e virtualizados. Assim, as infraestruturas de *data centers* podem ser utilizadas para hospedar os mais diversos tipos de serviços. Por exemplo, apesar de Google e Microsoft também fornecerem serviços de hospedagem em nuvem, Google Compute Engine e Microsoft Azure respectivamente, o foco principal da Microsoft é a hospedagem de aplicações como Skype, Office 365, Xbox Live, Outlook entre outros serviços oferecidos pela companhia (MICROSOFT, 2017). No Google os serviços vão desde suas aplicações para usuários finais, como mecanismo de busca, Gmail, YouTube, Google+, até infraestruturas para indexação de páginas na Internet (BENSON; AKELLA; MALTZ, 2010). Por fim, a Amazon AWS EC2 é conhecida por hospedar infraestruturas virtualizadas de acordo com um conjunto de requisitos de servidores e de infraestrutura de rede.

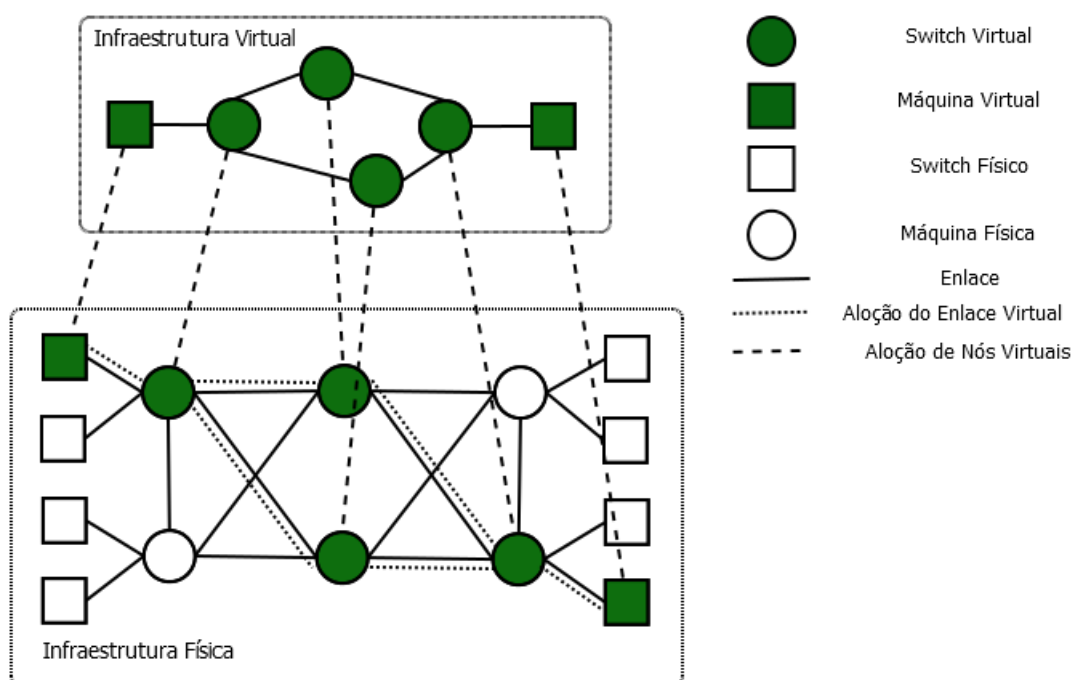
2.1.2 Infraestruturas Virtuais

Infraestruturas Virtuais (IVs) são compreendidas como a combinação de IaaS com provisionamento de redes virtuais (NaaS). Semelhante ao serviço oferecido em IaaS, NaaS é um modelo de provisionamento de recursos de redes personalizadas e sob demanda, com suporte a reserva de largura de banda, roteamento customizado, *firewall*, *switches* e roteadores com suporte a SDN (Seção 2.2) (COSTA et al., 2012).

As IVs são consideradas como uma camada de abstração lógica posicionada sobre o *data center*, sendo compostas por um conjunto de nós virtuais (MVs e *switches* virtuais), interconectados por enlaces virtuais. Cada nó virtual é hospedado por um nó físico de tipo equivalente (servidores hospedam MVs enquanto *switches* físicos hospedam seus pares virtuais), enquanto os enlaces virtuais podem se estender por um caminho composto por enlaces físicos (CHOWDHURY; BOUTABA, 2009). Para exemplificar o provisionamento, na Figura 1, a IV está localizada na parte superior da figura, representada pelos elementos em verde. A associação representada pela linha tracejada ligando os elementos da IV com a infraestrutura física representa a alocação de uma fração da infraestrutura física para hospedar os elementos da IV. Ainda, as máquinas são representadas por quadrados, enquanto os *switches* e roteadores são representados por círculos.

A camada de abstração criada para provisionar IVs facilita o gerenciamento por parte dos clientes. Sobretudo, permite a reorganização dinâmica dos recursos, de acordo com a carga submetida à aplicação hospedada, que pode variar durante a sua execução (HAIDER; POTTER; NAKAO, 2009).

Figura 1 – Alocação de IV. Diversos equipamentos de um *data center* são selecionados para alocar uma requisição de IV.



Fonte: Elaborado pelo autor, 2017.

2.1.3 Gerenciamento de Infraestruturas Virtuais

O ciclo de vida de uma IV contempla, resumidamente, etapas de especificação, alocação, configuração, instanciação e liberação dos recursos virtuais. Na etapa de especificação ocorre a descrição da IV e dos requisitos de QoS que devem ser atendidos. Na alocação, foco deste trabalho, acontece a verificação da possibilidade de posicionamento da IV dentro da infraestrutura física, considerando os elementos descritos na etapa de especificação. A etapa de configuração consiste em gerar a configuração dos equipamentos de rede e de computação, enquanto a instanciação consiste na preparação e disponibilização da IV para o usuário final. A última etapa, liberação, é quando a IV atinge seu objetivo (tempo limite ou solicitação de término) e então deve ser removida da infraestrutura física liberando a capacidade reservada (ANHALT; KOSLOVSKI; PRIMET, 2010).

Especificamente considerando a submissão e alocação de requisições de IVs, existem dois tipos de mecanismos: as requisições *offline* e as requisições *online*. As requisições *offline* já estão todas pré-definidas quando o processo de alocação inicia, então o *framework* de alocação da nuvem já sabe o quanto de cada IV irá consumir da infraestrutura física, e pode alocar todas de uma única vez, facilitando a redução da

fragmentação e portanto, realizando uma melhor alocação. Já as requisições *online* chegam em tempo de execução, sem que o *framework* de alocação da nuvem tenha conhecimento dos requisitos da IV, e assim é alocada uma IV por vez. Ainda, em requisições *online*, é comum que durante o ciclo de vida da IV os seus requisitos sejam alterados, devido a variação de carga de trabalho, redução de custos entre outros motivos. O processo de reconfiguração elástica da IV para se adaptar aos novos requisitos impacta no uso dos recursos do *data center*, aumentando a fragmentação ou desbalanceando a carga. Independente do mecanismo, ao fim do ciclo de vida da IV os recursos reservados para sua alocação são liberados, e podem ser utilizados para alocar novas IVs que vão chegar.

No processo de alocação, tanto a infraestrutura física quanto a IV são tratadas como grafos não direcionados com arestas e vértices com pesos. O processo de alocação consiste em identificar, dentro do grafo que representa o *data center*, um sub-grafo capaz de hospedar os elementos da IV, tendo como parâmetro de comparação a capacidade residual da infraestrutura física e a capacidade virtual solicitada (FISCHER et al., 2013; HAIDER; POTTER; NAKAO, 2009). A capacidade residual é compreendida como a capacidade total de um equipamento subtraída da capacidade previamente reservada para alocar requisições virtuais. A capacidade total do equipamento físico representa CPU, disco, memória para os servidores, capacidade de processamento e quantidades de entradas na tabela de roteamento para os *switches* e largura de banda para os enlaces (HAIDER; POTTER; NAKAO, 2009). Para exemplificar o processo de alocação, na Figura 1, o objetivo consiste em posicionar o grafo da IV (em verde) no grafo representando a infraestrutura do *data center* (em branco).

O problema de alocação de IVs com restrições de enlaces pode ser reduzido ao problema NP-Difícil de separador de múltiplos caminhos (ANDERSEN, 2002). O problema se torna ainda mais complexo a partir do momento que novas dimensões são adicionadas a formulação, como por exemplo as capacidades dos nós, que podem ser de dois tipos (*switches* e servidores) e portanto devem ser separadamente consideradas no processo de alocação. Outra dimensão diz respeito aos objetivos da alocação, do lado do provedor destaca-se uma eficiente utilização da infraestrutura física para reduzir fragmentação e evitar congestionamento do *data center*, ou então uma alocação de acordo com um melhor consumo energético, entre outros. Do lado do cliente, respeitar o QoS da IV, redução de custos, segurança, confiança, entre outros.

Além da alta complexidade na busca por uma solução válida, existe ainda a complexidade de configuração da IV. É comum que a infraestrutura física seja composta por equipamentos de diversos fabricantes, e esses equipamentos possuam linguagens específicas de configuração, ou seja, cada equipamento teria que ser representado e modelado de uma maneira diferente (KREUTZ et al., 2015).

2.2 SOFTWARE-DEFINED NETWORKING

Está cada vez mais difícil de desenvolver e gerenciar redes de computadores, com protocolos complexos e compostas por diversos equipamentos de rede, como *switches*, roteadores e *middleboxes* (equipamentos que manuseiam tráfego, mas não são responsáveis pelo encaminhamento de pacotes, como por exemplo *firewalls*). Os administradores da rede são responsáveis por configurar os recursos de acordo com conjunto de políticas de alto nível. Usualmente, estes devem manualmente converter essas políticas de alto nível em linguagens de comando de configuração dos equipamentos de rede. Em redes heterogêneas, essas linguagens de configuração mudam de acordo com os fabricantes, e às vezes diferentes linguagens de um mesmo fabricante variando de acordo com o modelo são propostas. Assim, a configuração de rede não é performática e está sujeita a diversos erros (NUNES et al., 2014; FEAMSTER; REXFORD; ZEGURA, 2013).

Essa dificuldade no gerenciamento dos equipamentos de rede é devido ao fato de os mesmos serem considerados *black boxes*, ou seja, são verticalmente integrados. A integração vertical significa que o plano de controle e o plano de dados estão incorporados no equipamento de rede. O plano de controle representa a parte que é responsável por decidir como lidar com o tráfego de rede, já o plano de dados é a parte que faz o encaminhamento dos pacotes de rede de acordo com as decisões do plano de controle (KREUTZ et al., 2015; NUNES et al., 2014).

2.2.1 Conceitos e Definições

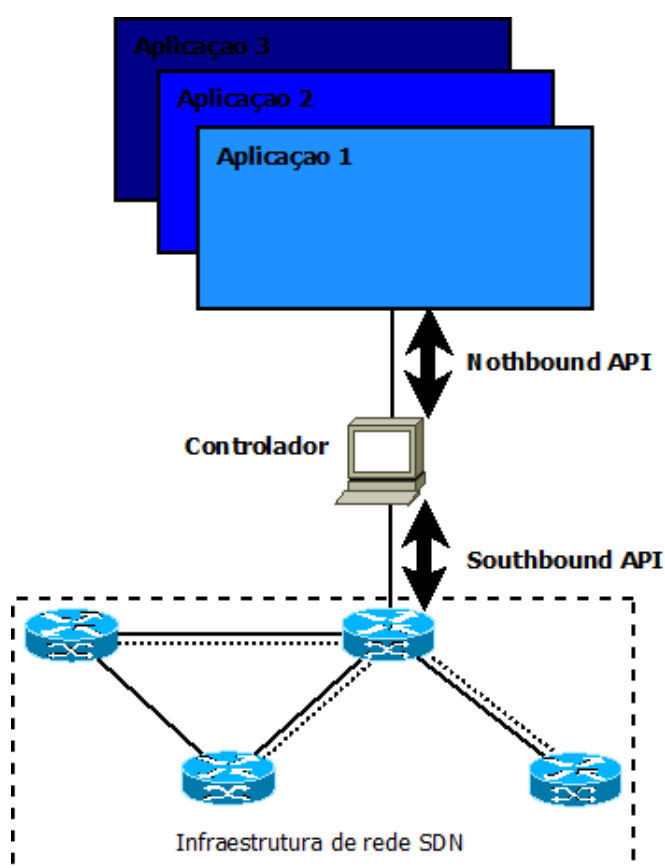
A ideia de redes programáveis se apresenta como uma alternativa para facilitar o processo de configuração e gerenciamento das redes de computadores. Em particular, *Software-Defined Networking (SDN)*, é um paradigma emergente que quebra a integração vertical, deixando para os equipamentos de rede somente a função do plano de dados, extraindo o plano de controle para uma unidade externa e logicamente centralizada. Essa unidade externa e logicamente centralizada é nomeada controlador (KREUTZ et al., 2015; NUNES et al., 2014).

Apesar de ser um conceito antigo SDN ganhou recente destaque tanto na indústria quanto na academia. Um grupo de operadores e fabricantes de equipamentos de rede, se uniu e fundou a *Open Network Foundation (ONF)*, uma organização orientada pela indústria que tem como objetivo promover SDN (FEAMSTER; REXFORD; ZEGURA, 2013; NUNES et al., 2014).

A Figura 2 representa uma rede SDN na qual o controlador está logicamente conectado a todos os equipamentos de rede, e essa conexão lógica é representada pela linha pontilhada. O controlador gerencia diversos planos de dados, que estão

contidos nos equipamentos de rede. Além de se comunicar com os equipamentos de rede, o controlador se comunica com diversas aplicações, que são responsáveis por definir as políticas de gerenciamento da rede. Essas aplicações podem ter como função gerenciar a infraestrutura física, e nesse caso comunicando diretamente com o controlador da rede. Ou então podem ser aplicações finais, que compõem uma IV hospedada na nuvem, e deve se comunicar com o controlador virtual responsável pela IV, para este então comunicar com o controlador físico e este gerenciar a fatia da IV na infraestrutura física.

Figura 2 – Infraestrutura de uma rede SDN.



Fonte: Elaborado pelo autor, 2017

O controlador se comunica com os planos de dados através da interface *south-bound*. Essa interface utiliza um protocolo padronizado de comunicação, facilitando a configuração e gerenciamento de uma infraestrutura de rede heterogênea. Além disso, SDN fornece outras vantagens, como garantias de largura de banda, isolamento, facilidade na coleta de informações (taxa de utilização de entradas e métricas de fluxo) do equipamentos de rede (MCKEOWN et al., 2008).

No ambiente SDN, a tupla armazenada no *switch* que indica a porta para onde cada pacote deve ser encaminhado representa um fluxo de dados. Esta tupla pode ser composta com informações das camadas 2, 3 e 4 do modelo *Open System Interconnection (OSI)* como por exemplo endereços IP, MAC e portas. Nos equipamentos de rede, essa tupla é armazenada nas tabelas de encaminhamento, que são o plano de dados, nas quais o *switch* busca informação sobre como encaminhar um pacote.

Por sua vez, o controlador armazena uma cópia das tabelas de encaminhamento de cada *switch* SDN, porém sem enfrentar a limitação de recursos de armazenamento dos *switches*. Assim, existe possibilidade de armazenamento de parte da tabela de encaminhamento no controlador, reduzindo assim a utilização dos *switches* (MCKEOWN et al., 2008).

2.2.2 Padrão OpenFlow

Apesar de ser vista como uma tecnologia emergente, SDN é um conceito antigo (FEAMSTER; REXFORD; ZEGURA, 2013), que foi trazido à tona atualmente pela definição de um protocolo padrão para comunicação *southbound*, o protocolo OpenFlow. O protocolo foi proposto por McKeown e seus colegas (MCKEOWN et al., 2008), e atualmente é mantido pela ONF, que defende a existência de uma interface padronizada, independente de fabricante para configurar e gerenciar os equipamentos de rede.

Diante da dificuldade de desenvolver e testar novas tecnologias para as redes de computadores, surgiu a motivação para criação de um protocolo que auxiliasse a configuração de redes virtualizadas em um ambiente de redes programáveis. Redes programáveis são aquelas que abstraem a infraestrutura física, onde os equipamentos de rede trabalham como um sistema distribuído, com uma interface programável aberta, que permite que provedores da rede possam facilmente manipular os equipamentos de rede (CHOWDHURY; BOUTABA, 2009). SDN é um exemplo de redes configuráveis.

É através do OpenFlow, que é possível impor garantias de largura de banda, recuperar informações de taxa de utilização de fluxo nas tabelas de encaminhamento, número de bytes e de pacotes encaminhados por fluxo. Ainda, o protocolo permite a criação de grupos de encaminhamento. Esses grupos de encaminhamento permitem que mais de uma ação seja executada quando um pacote chega ao *switch*. Por exemplo, é possível encaminhar o pacote para diversas portas, é possível enviar uma cópia para o controlador, para que este seja modificado. Além de ser possível utilizar esses grupos como balanceadores de carga. Outra vantagem da utilização de grupos, em alguns casos, é a redução do número de entradas nas tabelas de encaminhamento. Por exemplo: um pacote que deve ser encaminhado para diversos destinatários é re-

presentado por um grupo de encaminhamento. Atualmente o protocolo se encontra na versão 1.5.0, entretanto a versão mais recente que é implementada pela maioria dos fabricantes é a 1.3.0.

As funcionalidades descritas são possíveis por que o OpenFlow implementa dois tipos de tabelas de encaminhamento. As tabelas de fluxo são as tabelas de encaminhamento compostas por 3 campos. O primeiro campo é o de combinação, que contém as informações que devem combinar com os pacotes de rede entregues, o segundo campo é o contador de quantidade de vezes que houve a combinação com o pacote, e o terceiro campo contém as instruções de ação sobre o pacote. A segunda tabela (de grupos) contém quatro campos: o primeiro campo um inteiro para identificar o grupo, o tipo do grupo que determina quais conjuntos de ações devem ser tomadas, o contador da quantidade de vezes que houve combinação com o pacote e os conjuntos de ações (cada conjunto é composto por diversas ações) (ONF, 2011).

2.3 ALOCAÇÃO DE IVs EM SDN

SDN oferece diversas vantagens para o desenvolvimento e gerenciamento de redes de computadores, mas além disso, facilita o provisionamento de IVs em nuvens computacionais. Por disponibilizar uma interface padronizada para configuração dos equipamentos de rede, o processo de configuração é simplificado, sem distinção entre equipamentos de diferentes fabricantes. Ainda, SDN combinado com o protocolo OpenFlow fornece mecanismos para prover garantias de QoS, em especial largura de banda e latência. Em suma, as vantagens introduzidas no provisionamento de IVs em ambientes SDN podem ser descritas levando em consideração dois elementos:

- **Largura de Banda:** Através das funcionalidades presentes no protocolo OpenFlow, é possível garantir uma largura de banda mínima entre os recursos virtuais. Esta garantia de largura de banda ocorre nos equipamentos de rede, e não nas pontas da infraestrutura física. Entretanto, garantir a largura de banda nos equipamentos da rede significa encontrar dentro da infraestrutura de rede, um caminho que tenha uma largura de banda que respeite os requisitos de QoS. Como previamente mencionado, encontrar um caminho com restrições de rede pertence a classe de problemas computacionais NP-Difícil, e demanda a criação de uma heurística para aproximação de resultados.
- **Latência:** Outra vantagem no processo de alocação de IVs em ambiente SDN é a garantia de latência. É importante garantir a latência por que a variação de latência pode ter grande impacto no desempenho da aplicação (ALIZADEH et al., 2010). Por conhecer o estado da infraestrutura de rede, o controlador possui mecanismos para identificar caminhos que respeitem as restrições de latên-

cia. Entretanto, os equipamentos de rede SDN possuem uma capacidade de armazenamento limitada (KREUTZ et al., 2015). Uma alternativa diante de tal limitação, é explorar o controlador para armazenar entradas da tabela de fluxo, pois o controlador armazena uma cópia da tabela de fluxo porém sem as limitações de armazenamento. Este processo, também ajuda a reduzir a utilização da infraestrutura de rede, reduzindo os custos de alocação. A desvantagem de tal abordagem é o aumento da latência de comunicação, pois quando o pacote chega no equipamento de rede, este deve enviar para o controlador, para então receber uma resposta de como processar o pacote. E mais uma vez tem que se identificar um caminho que respeite requisitos de qualidade de serviço solicitados.

As vantagens elencadas tornam SDN atrativo para o provisionamento de IVs. Entretanto, SDN introduz novas dimensões no processo de alocação, aumentando assim sua complexidade. Originalmente, deveria ser escolhido um caminho que respeite o QoS de largura de banda, agora deve ser escolhido um caminho que respeite tanto o QoS de largura de banda como de latência, e em paralelo reduza a utilização da infraestrutura física, sobretudo das tabelas de fluxo dos *switches*.

2.4 TRABALHOS RELACIONADOS

Os trabalhos relacionados abordam a alocação de recursos para hospedar infraestruturas virtuais e provisionamento de recursos em ambientes SDN. A Tabela 1 resume os trabalhos revisados identificando as principais contribuições.

2.4.1 Alocação de Recursos para Hospedar IVs

Os benefícios fornecidos pela virtualização levaram a criação de diversas alternativas para alocação tanto de redes como infraestruturas virtuais. As alternativas variam entre si de acordo com a abordagem e o objetivo determinado pelo provedor, que detém os recursos computacionais. Por exemplo, (CHOWDHURY; RAHMAN; BOUTABA, 2012) (YU et al., 2008) (RIGGIO et al., 2013) dividiram o processo de alocação em duas etapas, alocando primeiros os nós, e então alocaram os enlaces virtuais. Por outro lado (OLIVEIRA; KOSLOVSKI, 2016) (MIJUMBI et al., 2015) (JARRAY; KARMOUCH, 2015) realizaram a alocação tanto de nós como de enlaces em única etapa. O principal diferencial observado é quanto a ponderação entre recursos computacionais e de comunicação. Ao alocar IVs em duas etapas, o algoritmo inicialmente restringe o espaço de busca guiado pelo objetivo de alocação de MVs. Posteriormente, os enlaces são alocados. Considerando a importância que as redes

Tabela 1 – Tabela de trabalhos relacionados.

Trabalho	Contribuição
(SHERWOOD et al., 2009)	<i>Hypervisor</i> para alocação e configuração de rede virtualizadas em ambientes SDN.
(POPA et al., 2011)	Compartilhamento proporcional de recursos de rede em <i>data centers</i> de nuvens computacionais. Algumas propostas requerem SDN.
(OLIVEIRA; KOSLOVSKI, 2016)	Mecanismo VITreeM para alocação de IVs com heurísticas baseadas em árvores. As heurísticas reduzem o espaço de busca alcançando baixo tempo de alocação e fragmentação. Entretanto, SDN não é abordado e VITreeM apresenta uma elevada taxa de rejeição de requisições.
(CAVALCANTI; OBE-LHEIRO; KOSLOVSKI, 2014)	Alocação de IVs resilientes reduzindo a fragmentação do <i>data center</i> . SDN não é considerado. O trabalho indica o potencial uso de modelos ótimos para alocação de IVs.
(CHOWDHURY; RAHMAN; BOUTABA, 2012)	Criação de grafos que estendem a infraestrutura física conectando recursos físicos e virtuais para alocação. Ao criar o grafo estendido, os autores transformam o problema de alocação em um problema de encaminhamento de fluxos. A formulação foi utilizada como base para desenvolvimento do presente trabalho.
(DEMIRCI; AMMAR, 2014)	Quantificação do impacto da latência no desempenho de aplicações com fluxos alocados no controlador SDN.
(AL-SHABIBI et al., 2014)	<i>Framework</i> para virtualização de redes utilizando SDN.
(MIJUMBI et al., 2014)	Controlador SDN adaptado para alocação de redes virtuais. O controlador realiza um balanceamento de carga guiado pelos fluxos alocados.
(GUO et al., 2010)	Virtualização de redes com garantias de largura de banda.
(DRUTSKOY; KELLER; REXFORD, 2013)	Uso de controladores virtuais para gerenciamento de redes virtualizadas privadas.

Fonte: o autor

de comunicação têm em nuvens computacionais, um mecanismo de alocação deve, preferencialmente, considerar os dois recursos com o mesmo peso.

A busca por uma solução ótima para alocação de IVs é um processo custoso e de alta complexidade, que levou a busca de alternativas para obter respostas em tempo computacional aceitável. Assim, alguns trabalhos utilizaram heurísticas para alocação de IVs (OLIVEIRA; KOSLOVSKI, 2016) (FISCHER et al., 2013). Tais heurísticas restringem o espaço de busca diminuindo o número de comparações necessárias para encontrar uma solução. A escolha das premissas para compor as heurísticas é um desafio, pois é necessário abstrair alguns elementos do *data center* ou simplificar o objetivo final do provedor ou do cliente.

Quanto à representação do problema a abordagem mais empregada é a utilização de formulação linear (YU et al., 2008) (CHOWDHURY; RAHMAN; BOUTABA, 2012) (RIGGIO et al., 2013) (MIJUMBI et al., 2015) (JARRAY; KARMOUCH, 2015). Em comum, os trabalhos representam o problema de alocação de IVs como um ma-

peamento de grafos e/ou encaminhamento de fluxos.

Ainda, a alocação pode seguir diferentes objetivos, tanto do ponto de vista do cliente, como do provedor. Os mecanismos de alocação, do ponto de vista do provedor, podem ser guiados por taxa de aceitação, redução da fragmentação, ou então a redução de custos, aumento do *revenue*. Já do ponto de vista do cliente as alocações buscam também a redução do custo, o aumento da largura de banda, redução de latência, ou em termos gerais *QoS* (FISCHER et al., 2013).

Embora SDN introduza vantagens no provisionamento de IVs, a introdução de *switches* com suas especificidades (tabelas de fluxos, grupos de encaminhamento) e controladores é um desafio, pois aumenta a dimensão das possíveis soluções que devem ser analisadas.

2.4.2 Provisionamento de IVs em ambiente SDN

Apesar de introduzir novas dimensões no processo de alocação, SDN fornece diversos mecanismos que auxiliam no gerenciamento de IVs. Além disso, a definição do protocolo OpenFlow motivou diversos trabalhos para alocação de IVs em cenários SDN (DEMIRCI; AMMAR, 2014) (DRUTSKOY; KELLER; REXFORD, 2013) (TAO; JUN; KE, 2015) (SHERWOOD et al., 2009) (MCKEOWN et al., 2008).

Dentre as abordagens que exploram os benefícios de SDN para o provisionamentos de IVs, (SHERWOOD et al., 2009) propõem um controlador virtualizado, que é responsável por se comunicar com o controlador físico para alterar a parte da infraestrutura física que representa a IV do respectivo cliente. (GUO et al., 2010) explorou os mecanismos de garantia de largura de banda para alocação de IVs em ambientes SDN.

É importante ressaltar que SDN combinado com OpenFlow permite o controle de largura de banda uniformizados nos equipamentos do núcleo da rede (POPA et al., 2011). Ou seja, as garantias de *QoS* podem ser diretamente configuradas nos *switches* intermediários.

2.5 CONSIDERAÇÕES PARCIAIS

Existe uma tendência cada vez maior de migração de serviços para nuvem, e dentro das alternativas existentes, IVs, que são a combinação de IaaS com NaaS, tem se mostrado uma boa alternativa para os clientes. Por ser o resultado de uma camada de abstração lógica, IVs usufruem de benefícios como provisionamento sob demanda, realocação elástica, redução dos custos de manutenção, entre outros.

Porém o processo de alocação de IVs, que considera tanto a infraestrutura

física como a IV, é um processo computacionalmente custoso, considerado NP-Difícil, e que portanto faz uso de heurísticas que buscam aproximar a solução encontrada do resultado ótimo. Essas heurísticas identificam dentro da infraestrutura física um subgrafo que respeite os requisitos demandados de qualidade da IV. Além da complexidade do processo de busca por uma solução válida existe a complexidade de configuração da infraestrutura física para alocar a IV.

SDN surge como uma alternativa que visa reduzir a complexidade do processo de provisionamento de IVs. Além disso, fornece mecanismos para garantia de QoS de rede. Entretanto o ambiente SDN introduz outras dimensões ao processo de alocação, como por exemplo a seleção de caminhos considerando largura e banda e latência, onde os *switches* do caminho podem ser alocados no controlador.

Os trabalhos relacionados indicam a existência de diversas abordagens para alocação de IVs em ambientes SDN, porém tais trabalhos carecem de uma abordagem que explore a vantagens do ambiente SDN e da garantia de QoS na alocação de IVs. Assim, no Capítulo 3, um modelo ótimo para alocação considerando as particularidades do ambiente SDN é discutido.

3 MODELO ÓTIMO PARA ALOCAÇÃO DE INFRAESTRUTURAS VIRTUAIS EM DATA CENTERS SDN

Este capítulo descreve o modelo de programação inteira mista (*MIP*) que realiza a alocação de infraestrutura virtuais com garantias de *QoS*, reduzindo a fragmentação da infraestrutura física em ambientes SDN. A Seção 3.1 descreve o problema de alocação considerando as informações e características de provedores e clientes. Na Seção 3.2 é descrito o processo de alocação baseado em formulação linear que busca uma solução ótima, desde a seleção de candidatos até a formalização do conjunto de restrições. As limitações do modelo são discutidas na Seção 3.3, finalizando o capítulo.

3.1 FORMULAÇÃO DO PROBLEMA

A formulação do problema considera, individualmente, os componentes relacionados com o provisionamento de uma IV: infraestrutura de *data center*, requisições de IVs e objetivo do provedor. A notação utilizada ao longo do trabalho é resumida na Tabela 2: i e j representam recursos virtuais, enquanto u e v são utilizados para recursos físicos.

Tabela 2 – Notação utilizada ao longo do artigo. i e j representam recursos virtuais, enquanto u e v são utilizados para recursos físicos.

Notação	Descrição
$G^s(N_h^s, N_n^s, C^s, E^s)$	grafo do <i>data center</i>
N_h^s	servidores físicos
N_n^s	<i>switches</i> SDN
C^s	controlador SDN
E^s	enlaces físicos
$R(\cdot)$	capacidade física residual
$G^v(N_h^v, N_n^v, E^v, D^v)$	requisição de IV
N_h^v	MVs
N_n^v	<i>switches</i> virtuais
E^v	enlaces virtuais
D^v	matriz de latência máxima aceita
d_{ij}	latência máxima entre i e j
c_i	capacidade solicitada para i
$\Omega(i)$	candidatos físicos para hospedar i
$P^s(i, j)$	caminhos no <i>data center</i> entre $\Omega(i)$ e $\Omega(j)$

Fonte: o autor

3.1.1 Data center de um Provedor IaaS

A infraestrutura do *data center* é representada por um grafo não direcionado com pesos nas arestas e vértices. Os vértices representam tanto os servidores como os equipamentos de rede, e as arestas os enlaces de comunicação. Assim o *data center* é representado pelo grafo $G^s(N_h^s, N_n^s, C^s, E^s)$, sendo que N_h^s representa o conjunto de servidores e N_n^s o conjunto de *switches* que compõem a infraestrutura física.

Os controladores SDN são representados por C^s . Os enlaces que interconectam os servidores e equipamentos de rede são representados pelo conjunto E^s . Cada recurso (servidores, equipamentos de rede ou enlaces) possui uma capacidade residual, denotada pela função $R(\cdot)$.

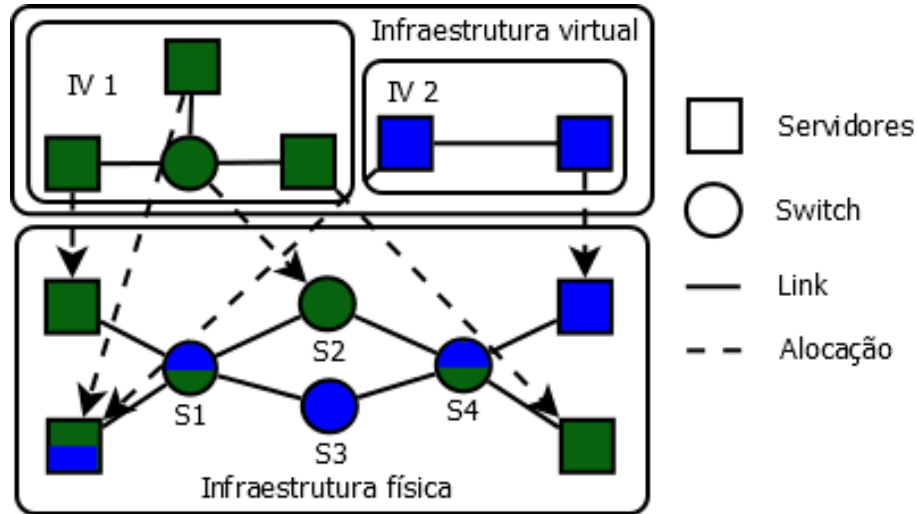
3.1.2 Requisições de IVs

Assim como para o *data center*, uma requisição de IV é representada por um grafo $G^v(N_h^v, N_n^v, E^v, D^v)$, sendo que N_h^v representa o conjunto de MVs, N_n^v o conjunto de equipamentos de rede e E^v o conjunto de enlaces. D^v é uma matriz que contém a latência máxima aceita fim-a-fim, na qual d_{ij} representa a maior latência aceita entre os recursos virtuais i e j . A capacidade solicitada para cada recurso virtual é representada por c .

Como a configuração da rede virtual é um aspecto importante para o desempenho de aplicações hospedadas em IVs, provedores estão buscando a alocação de MVs com requisitos de QoS de rede. Porém para algumas aplicações, a rede não é um fator limitante, como por exemplo aplicações que fazem pouca comunicação e muito processamento. Neste sentido, são abordados dois tipos de requisições de IVs: requisições somente IaaS e outra que combina IaaS com NaaS. As requisições são apresentadas na Figura 3 e detalhadas a seguir.

- **Requisições somente IaaS** representam MVs solicitadas sem a descrição da topologia virtual. Entretanto, além das configurações de MVs, regiões ou zonas, alguns requisitos de rede, como por exemplo a latência máxima fim a fim ou uma largura de banda mínima podem ser especificados. Este tipo de requisição é representado pela IV 2 (Figura 3), na qual duas MVs devem ser, opcionalmente, provisionadas com requisitos de QoS de rede. É importante destacar que os equipamentos de rede utilizados para garantir a comunicação das MVs são abstraídos do cliente sendo de conhecimento somente do provedor.
- **Requisições combinando IaaS e NaaS** permitem a completa descrição da IV. O cliente pode realizar uma descrição detalhada dos equipamentos de rede que compõem a rede virtual, bem como os requisitos de QoS. A IV 1 (Figura 3) exem-

Figura 3 – Alocação de IVs em provedores de nuvens baseados em SDN.



Fonte: Elaborado pelo autor, 2017

plifica que o provedor pode utilizar mais de um recurso físico para alocar um virtual: enquanto a IV solicitou um único *switch* virtual, o provedor utilizou 3 *switches* SDN físicos para garantir a comunicação dos equipamentos virtuais. Nesse caso, o cliente só tem acesso para configuração do *switch* que efetivamente está alocando o equipamento virtual solicitado.

3.1.3 Alocando recursos físicos para hospedar IVs

Conforme discutido na Seção 2.1.3, no presente trabalho serão alocadas requisições *online* de IVs. Ou seja, o *framework* de alocação da nuvem não tem conhecimento do número de requisições que serão submetidas pelos clientes, tampouco dos requisitos de QoS. Cada requisição é individualmente analisada. Além disso, apesar de ter conhecimento que ao longo do ciclo de vida da IV seus requisitos possam ser alterados, este trabalho não realiza a reconfiguração da IV, acreditando que alocação e reconfiguração são processos que devem ser analisados individualmente. Portanto o foco é somente no processo de alocação.

A alocação de IVs em um *data center* de nuvem pode ser dividida em duas etapas: alocação de nós e alocação de enlaces (CHOWDHURY; RAHMAN; BOU-TABA, 2012). O mapeamento de MVs em servidores físicos é dado por $M_h : N_h^v \mapsto N_h^s$, enquanto para *switches* virtuais mapeados em *switches* SDN por $M_n : N_n^v \mapsto N_n^s$, sendo $M_h(i) \in N_h^s$ e $M_n(i) \in N_n^s$. De maneira similar, o mapeamento de um enlace virtual ij é realizado em um caminho $p \in P^s$, entre os nós que alocam os nós virtuais finais (*switch* ou MV) do enlace ij . Para hospedar um recurso virtual, o nó físico deve

ter uma capacidade residual (não reservada) maior ou igual a capacidade virtual solicitada. Para as MVs e os *switches*, $c_i \leq R(M_h(i))$ e $c_j \leq R(M_n(j))$, respectivamente, enquanto para os enlaces virtuais, $c_{ij} \leq R(M_e(ij))$.

3.1.4 Objetivos do provedor de nuvem IaaS

O principal objetivo do modelo proposto é a alocação de IVs com requisito de QoS de rede. Além disso, a formulação busca a redução do custo de alocação, aumentando o *revenue* do provedor. De maneira similar a outros trabalhos, o custo para alocar uma IV é proporcional a capacidade física reservada para garantir o *Service Level Agreement (SLA)*, como mostrado na Eq. (3.1), na qual o custo para alocar a IV G^v é igual ao somatório das capacidades solicitadas para as MV do conjunto N_h^v , acrescido do somatório das capacidades solicitadas para os *switches* virtuais do conjunto N_n^v , mais os recursos de rede consumidos para alocar os enlaces virtuais do conjunto E^v . Sendo que $|ij|$ representa o tamanho do caminho para alocar o enlace virtual ij , em número de saltos (FISCHER et al., 2013). O *revenue* ao alocar uma IV, dado pela Eq. (3.2), é calculado de maneira semelhante a Eq. (3.1), porém sem considerar todos os saltos para alocar o enlace virtual ij .

$$\mathcal{C}(G^v) = \sum_{i \in N_h^v} c_i + \sum_{j \in N_n^v} c_j + \sum_{ij \in E^v} c_{ij} |ij| \quad (3.1)$$

$$\mathcal{R}(G^v) = \sum_{i \in N_h^v} c_i + \sum_{j \in N_n^v} c_{ij} + \sum_{ij \in E^v} c_{ij} \quad (3.2)$$

Além dos requisitos indicados no *SLA*, o QoS fornecido ao cliente é quantificada pela latência média dos recursos virtuais comunicantes. Como descrito na Eq. (3.3), o QoS é calculado através do somatório das latências médias dos caminhos físicos hospedando todos os enlaces virtuais do conjunto E^v , dividido pelo número de enlaces virtuais, representado por $|E^v|$. Ainda, $\mathcal{L}(i, j)$ representa a latência média do caminho físico alocando ij .

$$\mathcal{Q}(G^v) = \frac{\sum_{ij \in E^v} \mathcal{L}(i, j)}{|E^v|} \quad (3.3)$$

3.2 MIP ÓTIMO PARA ALOCAÇÃO DE IVs COM REQUISITOS DE QOS

Um MIP ótimo para alocação de IVs deve considerar todas as possibilidades de mapeamento em um *data center*. Entretanto, o processo de alocação de IVs com requisitos de QoS é computacionalmente custoso. Diante de tal complexidade e de acordo com as definições da Seção 2.1.1, uma alternativa é a redução controlada do

espaço de busca. No modelo proposto, a redução do espaço de busca no processo de alocação de IVs é realizada em duas etapas. Primeiro é reduzido o espaço de busca através da restrição da localização da IV, aplicando o conceito de zonas e regiões, comumente utilizado por provedores de nuvens computacionais públicas. A segunda etapa, consiste em reduzir o conjunto de candidatos aptos a alocar os recursos virtuais de acordo com os requisitos de comunicação especificados para a IV. Após a redução do número de candidatos, é aplicado um conjunto de restrições baseados em MIP. Essas restrições tem como objetivo impor o QoS solicitado pelo cliente, bem como garantir os objetivos do provedor de nuvem.

3.2.1 Seleção de candidatos para alocar recursos virtuais

O primeiro filtro na seleção de candidatos é com base na localização geográfica dos recursos. Em provedores públicos de IaaS, clientes podem selecionar regiões e zonas para alocação das MVs. A Amazon por exemplo tem 42 zonas divididas em 16 regiões, dispersas geograficamente, e cada região é composta por mais de uma zona (AMAZON, 2017a). Google Compute Engine também aplica a mesma ideia de zonas e regiões, porém com 18 zonas distribuídas em 6 regiões (GOOGLE, 2017b). Assim, quando o cliente seleciona uma região ou zona, o espaço de busca é consequentemente reduzido, não permitindo assim a alocação inter-zonas ou inter-regiões.

Neste trabalho, cada recurso virtual requisita uma localização (zona ou região) representado por $loc(i)$. Qualquer recurso físico na localização $loc(i)$ é um candidato para alocar i . Candidatos para alocar o recurso virtual i são representados pelo conjunto $\Omega(i)$.

Além da localização, requisitos de QoS de rede são essenciais para soliciantes de IVs (STALLINGS, 2015). Sobretudo, a latência e o desempenho da rede virtual não são fatores totalmente dependentes da localização física (PERSICO et al., 2015). Sendo assim, além de permitir a seleção de candidatos baseada na localização, como normalmente é realizado (MIJUMBI et al., 2015) (CHOWDHURY; RAHMAN; BOU-TABA, 2012), é proposta a seleção de candidatos baseada em latência fim a fim.

- **Candidatos para alocar *switches* virtuais com requisitos de latência.** Para requisições de IVs com enlaces contendo requisitos de latência e *switches* virtuais, o conjunto de candidatos físicos para alocar o *switch* i , ou seja, $\Omega(i)$, é composto por equipamentos do conjunto N_n^s , capazes de hospedar o pior requisito de latência de i , isto é, que possuem enlaces em que a pior latência é menor do que o maior requisito de latência solicitado para i ($\max(d_{ij})$). Em suma, $\Omega(i) = \{u \in N_n^s | \max(lat(u, v)) < \max(d_{ij}); \forall v \in adj(u); \forall j \in adj(i), \text{ onde } adj(.)$

retorna o conjunto de nós adjacentes, e $lat(u, v)$ indica a latência do caminho físico entre u e v .

- **Candidatos para alocar *switches* virtuais sem requisitos de latência.** Neste caso, todos os *switches* físicos com capacidade residual suficiente são candidatos para alocar o *switch* virtual i . Assim, $\Omega(i) = \{u \in N_n^s | R(u) \geq c_i\}$.
- **Candidatos para alocar MVs sem requisitos de latência.** Os servidores físicos com suficiente capacidade residual são selecionados para alocar a MV i . Em outras palavras, $\Omega(i) = \{u \in N_h^s | R(u) \geq c_i\}$.
- **Candidatos para alocar MVs com requisitos de latência.** Para alocação de MVs com requisitos de latência existem dois cenários:

(a) MVs conectadas com *switches* virtuais: neste caso, os candidatos são selecionados com base na latência de comunicação do *switch* de borda conectado com a MV i . Assim, $\Omega(i)$ é composto por $\{u \in N_h^s | lat(u, v) \leq d_{ij}; \forall v \in \Omega(j); j \in adj(i) \setminus N_h^v\}$.

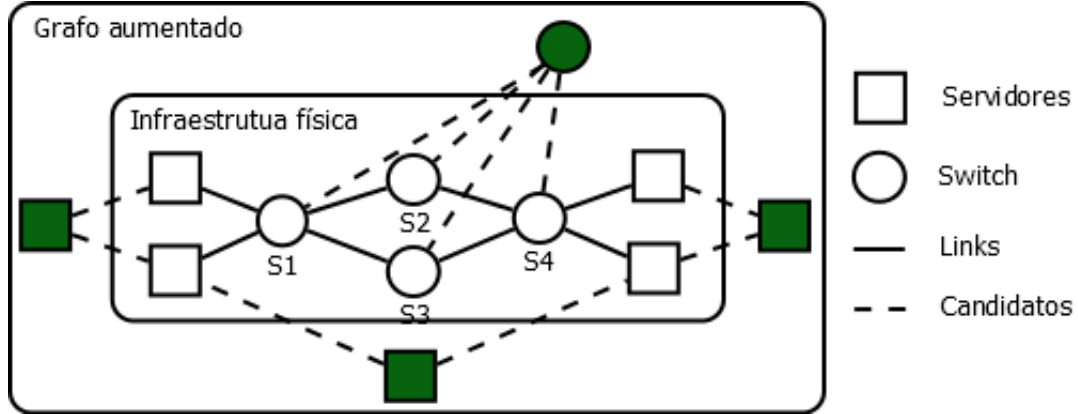
(b) MVs conectadas com MVs: são selecionados os candidatos que possuem um caminho entre um par de servidores, que respeita o requisito de latência, ou seja, baseado na latência fim a fim. Portanto, $\Omega(i) = \{u \in N_h^s | lat(u, v) \leq d_{ij} \forall v \in N_h^s\}$.

Seguindo a abordagem proposta por (CHOWDHURY; RAHMAN; BOUTABA, 2012), cada recurso virtual i é inserido no grafo físico, através de um enlace temporário com seus candidatos, com capacidade infinita e sem latência. O grafo aumentado resultante é representado por $G^{s'}(N^{s'}, C^s, E^{s'})$. Assim, o conjunto de nós do grafo aumentado, $N^{s'}$, é composto pela combinação dos servidores, MV, *switches* físicos e *switches* virtuais, sendo, $N^{s'} = N_h^s \cup N_h^v \cup N_n^s \cup N_n^v$; e o conjunto de enlaces do grafo aumentado, $E^{s'}$, é composto por todos os enlaces da infraestrutura física bem como os enlaces temporários ligando os elementos da IV e seus candidatos, ou seja, $E^{s'} = E^s \cup \{iu | i \in N_h^v, u \in \Omega(i)\} \cup \{ju | j \in N_n^v, u \in \Omega(j)\}$. Os controladores físicos (C^s) não são afetados pela criação do grafo aumentado. A Figura 4 exemplifica um grafo aumentado, conectando a IV 1 da Figura 3 a um *data center* simplificado.

3.2.2 Variáveis e objetivo

Para representar uma solução de mapeamento de IVs com QoS de rede foram utilizadas três variáveis. A variável f_{ijuv} representa o quanto do fluxo (capacidade do enlace virtual) de ij está alocado no enlace do *data center* entre u e v . A variável x_{uv} possui um domínio binário, que representa a presença ou não de um fluxo (requisição de enlace virtual) entre os recursos u e v . Se existe algum fluxo dos enlaces virtuais

Figura 4 – Grafo aumentado combinando recursos físicos e virtuais.



Fonte: os autores

de E^v alocado no enlace entre u e v , sendo $\sum_{ij \in E^v} (f_{ijuv} + f_{ijvu}) > 0$, o valor em x_{uv} é definido como 1, caso contrário 0. Por fim, a variável binária e_{ijpu} representa as entradas das tabelas de encaminhamento de fluxos que estão alocadas no controlador, para o enlace virtual ij no caminho p e no *switch* u . Quando a entrada está alocada no controlador é atribuído o valor 1, caso contrário, a entrada está alocada no *switch* u e o valor é 0.

De acordo com os objetivos do provedor (Seção 3.1.4), uma versão modificada das Equações (3.1), (3.2) e (3.3) compõem a função objetivo (Eq. 3.4). A minimização da função objetivo reduz o custo para alocação de uma IV, através da redução do número de servidores, enlaces e *switches* físicos utilizados, bem como a alocação de fluxos no controlador. Como proposto na literatura, ao ponderar o custo pela capacidade residual dos recursos físicos (servidores, *switches* e enlaces), são selecionados os recursos com maior capacidade residual, balanceando a carga do *data center* (CHOWDHURY; RAHMAN; BOUTABA, 2012). Os parâmetros α_{uv} , β_u , e γ_e controlam a importância de cada recurso de acordo com a visão do provedor, com valores entre 1 e $R(\cdot)$, enquanto δ é um número positivo para evitar divisão por zero.

Assim, os termos com α_{uv} , β_u , e γ_e calculam os pesos para nós (*switches* e servidores), entradas alocadas no controlador e enlaces, respectivamente. O somatório com o termo x_{iu} reduz o número de nós utilizados para alocar uma IV, através do cálculo da capacidade consumida de cada nó físico, pelos nós respectivos (*switches* e servidores) da IV. O somatório com o termo $(1 - e_{ijpu})$ reduz o volume de entradas alocadas nos equipamentos de rede. E por fim, o somatório com o termo f_{ijuv} reduz o número de enlaces utilizados para alocar fluxos do conjunto E^v .

$$\min : \sum_{u \in N_h^s \cup N_n^s} \frac{\beta_u}{R(u) + \delta} \sum_{i \in N^v} x_{iu} c_i + \sum_{u \in N_n^s} \frac{\gamma_e}{R(u) + \delta} \sum_{ij \in E^v} \sum_{p \in P^s(i,j)} (1 - e_{ijpu}) + \sum_{uv \in E^s} \frac{\alpha_{uv}}{R(uv) + \delta} \sum_{ij \in E^v} f_{ijuv} \quad (3.4)$$

3.2.3 Restrições para alocação ótima

Além da função objetivo, para garantia dos requisitos de QoS solicitados no SLA, um conjunto de restrições de capacidade, fluxo, binárias e meta restrições deve ser atendido pelo mecanismo de alocação.

3.2.3.1 Restrições de capacidade dos servidores e enlaces

A Eq. (3.5) define a restrição de capacidade para os enlaces, onde a soma do fluxo tanto em um sentido quanto no outro, para todos os enlaces virtuais, não deve exceder a capacidade residual do enlace. Por sua vez, a Eq. (3.6) aplica as restrições de capacidade aos servidores, de maneira similar a restrição anterior: a capacidade residual não deve ser excedida, considerando todas as MVs hospedadas no candidato. Em suma, os recursos físicos devem ter capacidade residual suficiente para alocar as requisições.

$$\sum_{ij \in E^v} (f_{ijuv} + f_{ijvu}) \leq R(uv) x_{uv} \quad \forall u, v \in N^{s'} \quad (3.5)$$

$$R(u) \geq \sum_{i \in N_h^v} x_{iu} c_i \quad \forall u \in N_h^s \quad (3.6)$$

3.2.3.2 Restrições de SDN.

Os equipamentos com SDN possuem uma particularidade quanto as suas tabelas de fluxo: o controlador mantém uma cópia da tabela de fluxos de todos os *switches*, e isso permite que alguns fluxos do *switch* (representado por x) possam estar alocados somente no controlador (identificado por e). E devido a capacidade limitada dos *switches* SDN, isso se mostra como uma alternativa para redução da infraestrutura física. Neste sentido, a Eq. (3.7) busca garantir que a capacidade residual de um *switch* u deve ser suficiente para alocar todos os fluxos que passam por aquele *switch*, entretanto, as entradas alocadas no controlador não devem ser consideradas. s_{ij} e t_{ij} representam a origem e o destino, do enlace virtual ij . Assim, o primeiro elemento do lado direito da Eq. (3.7), calcula a quantidade de entradas que são alocadas para s_{ij} ou t_{ij} no *switch* u . O segundo elemento, calcula a quantidade de entradas alocada em u para enlaces que passam por ali, mas que não aloquem nem s_{ij} nem t_{ij} em u .

$$\begin{aligned}
R(u) \geq & \sum_{k \in N_n^v} \sum_{ij \in E^v: s_{ij}=k \vee t_{ij}=k} \sum_{p \in P^s(i,j)} (x_{ku} - e_{ijpu}) c_k \\
& + \sum_{ij \in E^v} \sum_{p \in P^s(i,j)} (x_{iu} - e_{ijpu}) \quad \forall u \in N_n^s
\end{aligned} \tag{3.7}$$

3.2.3.3 Restrições de fluxo de dados

As requisições de enlaces virtuais representam um fluxo de dados que deve ser transferido no *data center*. A origem e o destino de cada fluxo são recursos virtuais (máquinas virtuais ou *switches*). Assim, a Eq. (3.8) garante que para cada enlace virtual ij , o fluxo iniciado em s_{ij} deve ser igual a capacidade solicitada para o enlace virtual, enquanto a Eq. (3.9) tem significado equivalente para o fluxo chegando em t_{ij} .

Complementando, a Eq. (3.10) é responsável pelos fluxos encaminhados no *data center* SDN: cada *switch* intermediário deve encaminhar os fluxos que recebeu em sua totalidade. Essas restrições tem como objetivo, garantir que para todos os enlaces virtuais, todo fluxo que entra na rede, deve também sair da rede.

$$\sum_{u \in N^{s'}} f_{ijs_{ij}u} - \sum_{u \in N^{s'}} f_{ijus_{ij}} = c_{ij} \quad \forall ij \in E^v \tag{3.8}$$

$$\sum_{u \in N^{s'}} f_{ijt_{ij}u} - \sum_{u \in N^{s'}} f_{ijut_{ij}} = -c_{ij} \quad \forall ij \in E^v \tag{3.9}$$

$$\sum_{v \in N^{s'}} f_{ijuv} - \sum_{v \in N^{s'}} f_{ijvu} = 0 \quad \forall ij \in E^v, \forall u \in N^{s'} \setminus \{s_{ij}, t_{ij}\} \tag{3.10}$$

3.2.3.4 Restrições de latência

As Equações (3.11) e (3.12) garantem que a latência do caminho físico alocando o enlace virtual é menor ou igual ao solicitado pelo cliente, mesmo quando algumas regras de encaminhamento estão no controlador. A Eq. (3.11) visa garantir que os enlaces virtuais alocados com entradas armazenadas no controlador não devem ter uma latência excedente ao limite determinado pelo SLA. Já a Eq. (3.12) garante que a latência para um enlace virtual, sem nada alocado no controlador, deve ser menor ou igual a solicitada.

$$d_{ij} \leq \sum_{u,v \in p} (lat(u,v)x_{uv} + lat(u,c)e_{ijpu}) \quad \forall ij \in E^v \quad \forall p \in P^s(i,j) \tag{3.11}$$

$$d_{ij} \geq \sum_{u,v \in p} lat(u,v)x_{uv} \quad \forall ij \in E^v \quad \forall p \in P^s(i,j) \tag{3.12}$$

3.2.3.5 Meta restrições e restrições binárias

Pela Eq. (3.13), um recurso virtual será alocado em somente um recurso físico, enquanto a Eq. (3.14) garante que x irá receber valores quando existir algum fluxo passante, tanto num sentido como no outro. As Eq. de (3.15) a (3.17) definem os domínios das variáveis.

$$\sum_{u \in \Omega(i)} x_{iu} = 1 \quad \forall i \in N^v \quad (3.13)$$

$$x_{uv} = x_{vu} \quad \forall u, v \in N^{s'} \quad (3.14)$$

$$f_{ijuv} \geq 0 \quad \forall u, v \in N^{s'}; \forall ij \in E^v \quad (3.15)$$

$$x_{uv} \in \{0, 1\} \quad \forall u, v \in N^{s'} \quad (3.16)$$

$$e_{ijpu} \in \{0, 1\} \quad \forall u \in N_n^s; \forall ij \in E^v; \forall p \in P^s(i, j) \quad (3.17)$$

3.3 CONSIDERAÇÕES PARCIAIS

O processo de alocação de recursos físicos para hospedar infraestruturas virtuais é essencial para o provisionamento em nuvens computacionais. No modelo proposto, tanto infraestruturas físicas como IVs são representadas por grafos no processo de alocação, e as requisições de IVs podem ser de dois tipos: IaaS e NaaS. Requisições do tipo IaaS não detalham infraestrutura de rede, diferente das requisições NaaS que são compostas por MVs e uma infraestrutura de rede detalhada.

Apesar dos requisitos da IV serem alterados ao longo do seu ciclo de vida, este trabalho foca somente no processo de alocação, sem abordar a realocação. O modelo atende o objetivo do provedor reduzindo o custo de alocação e aumentando o *revenue*. Do ponto de vista do cliente, a alocação vai respeitar o QoS solicitado, além de diminuir a latência média percebida na comunicação dos recursos virtuais.

Conforme apresentado na Seção 2.3, um *data center* baseado em SDN introduz novas dimensões de complexidade ao problema de alocação de IVs. Por isso esses desafios foram modelados através das Eq. (3.11) e (3.12) para garantia de latência e para explorar a possibilidade de alocação de fluxos no controlador (Eq. (3.7)). Por sua vez, a restrição de largura de banda foi modelada nas Equações (3.8), (3.9) e (3.10).

Além de não abordar a reconfiguração quando os requisitos da IV mudam, o modelo proposto tem outras limitações. O MIP não permite utilizar um servidor como *switch* virtual. Esta abordagem de utilizar servidores como *switches* virtuais, se mostra interessante, uma vez que nos servidores não haveria limitações do tamanho da tabela de encaminhamento, e não seria necessário se preocupar com a latência para arma-

zenar fluxos nas tabelas do controlador. Outra limitação é a inexistência de elementos no LP para garantir resiliência da infraestrutura, isso fica totalmente sob controle do cliente.

4 QOS-AWARE VI ALLOCATION ON SDN-BASED DATA-CENTERS

É sabido que a solução de um MIP é um problema NP-Difícil (CHOWDHURY; RAHMAN; BOUTABA, 2012). Consequentemente, a aplicabilidade da formulação MIP ótima é limitada em ambientes reais, visto que os clientes de nuvem buscam uma resposta imediata para as requisições.

Diante do exposto, inicialmente ocorre o relaxamento dos domínios das variáveis x e e , compondo um problema de programação linear. Em um segundo momento, o número de candidatos e de caminhos físicos são reduzidos de acordo com as particularidades de *data centers* de nuvem (Seção 2.1.1). Após a solução do LP, os resultados são tratados por uma heurística para obtenção de uma solução. As técnicas listadas compõem QVIA-SDN (*QoS-Aware VI Allocation on SDN-based data-centers*).

4.1 RELAXANDO AS VARIÁVEIS

Para obter o LP, as Equações (3.16) e (3.17) tiveram seus domínios relaxados. Assim, respectivamente, $1 \geq x_{uv} \geq 0; \forall u, v \in N^{s'}$ e $1 \geq e_{ijpu} \geq 0; \forall u \in N_n^s; \forall ij \in E^v; \forall p \in P^s(i, j)$. A função objetivo bem como as demais restrições permanecem inalteradas no LP.

4.2 REDUÇÃO DO NÚMERO DE CANDIDATOS FÍSICOS

Os *data centers* de nuvens computacionais são normalmente compostos por recursos homogêneos, interconectados por uma infraestrutura de rede estruturada (STALLINGS, 2015). Assim, recursos podem ser agrupados de acordo com suas similaridades. Por exemplo, em um *data center* estruturado como uma *fat-tree* (AL-FARES; LOUKISSAS; VAHDAT, 2008), um conjunto de *pods* pode ser internamente composto por enlaces com maior largura de banda. Ainda, grupos de servidores podem ser criados de acordo com suas capacidade de processamento, armazenamento, entre outros.

4.2.1 Servidores e *switches*

Com a aplicação de técnicas de virtualização, a localização dos recursos pode ser abstraída, compondo grupos com recursos geograficamente distribuídos. Por exemplo, os provedores de nuvens públicas Google e Amazon realizam o agrupamento com base na localização geográfica, informação comumente utilizada para composição de grupos de candidatos (ROST; FUERST; SCHMID, 2015) (Seção 3.2.1).

Nessa linha, QVIA-SDN reduz o número de candidatos físicos utilizando somente uma porcentagem de cada região do *data center*. A porcentagem de candidatos em cada região ou zona pode ser ajustada pelo provedor de acordo com a quantidade de recursos físicos disponíveis em cada região ou zona. É esperado que ao reduzir o número de candidatos, a eficiência do algoritmo seja impactada. Essa parametrização é discutida no Capítulo 5.

4.2.2 Enlaces de comunicação

É comum que as topologias dos *data centers* contenham múltiplos caminhos entre os servidores para balanceamento de carga e tolerância a falhas (STALLINGS, 2015; AL-FARES; LOUKISSAS; VAHDAT, 2008), aumentando o espaço de busca por uma solução para hospedar enlaces virtuais. Por natureza, a contabilização de todos os caminhos entre os recursos que compõem o *data center*, é impraticável. Sobretudo quando o procedimento deve ser repetido a cada alocação para atualizar a capacidade residual do *data center*.

Como o LP requer um conjunto P^s para encontrar uma solução aproximada, QVIA-SDN emprega uma redução de caminhos físicos candidatos. Essa redução de candidatos empregada, não remove nenhum enlace de comunicação do *data center*, somente oculta a existência de caminhos, ou seja, qualquer algoritmo de resiliência ou engenharia de tráfego pode ser posteriormente aplicado. Assim, P^s é composto por até dois caminhos entre cada par de candidatos físicos (servidores e *switches*), (i) um caminho mais curto, em números de saltos, e (ii) um caminho com a menor latência, sendo (i) diferente de (ii).

4.3 HEURÍSTICAS

Ao resolver o LP relaxado, a correlação entre as variáveis f , x e e é perdida. Portanto, QVIA-SDN aplica uma heurística para avaliar os valores obtidos e aproximar a solução. A heurística é composta por duas etapas, como descrito a seguir e detalhadas pelos Algoritmos 1 e 2.

4.3.1 QVIA-SDN

A primeira etapa da heurística empregada é baseada em (CHOWDHURY; RAHMAN; BOUTABA, 2012), e descrita no Algoritmo 1. O primeiro passo é a criação do grafo aumentado, conectando os recursos virtuais a seus candidatos, de acordo com as definições da Seção 3.2.1. Então o LP é resolvido nas linhas 1 e 2.

O objetivo é encontrar o candidato mais adequado para alocar cada elemento da IV, de acordo com os valores obtidos na resolução do LP. Para tal, é calculado o

valor p_z para cada elemento k da IV (linhas 4 a 6). p_z é calculado através soma ponderada do total de fluxos passando por kz em ambas as direções e x_{kz} . Esta abordagem reconstrói a correlação entre f e x , identificando o mapeamento de vértices a arestas. O peso α indica a preferência (rede ou nós) durante a combinação das informações. O candidato com o maior p_z é selecionado para alocar o recurso virtual. Quando nenhum candidato é identificado, a alocação da IV é rejeitada (linhas 8 e 9).

As linhas 17 a 21 são executadas de acordo com os resultados obtidos pela execução da segunda parte da heurística, discutida a seguir (em suma, seleciona caminhos para alocar os enlaces virtuais). Caso seja possível alocar todos os enlaces virtuais, as capacidades dos recursos físicos selecionados para hospedar os elementos da IV são atualizadas. Caso contrário a IV é rejeitada.

```

Input:  $G^v, G^s$ 
Output:  $M_n, M_e$ 
1  Cria o grafo aumentado  $G^{s'}$ 
2  Resolve QVIA-SDN com variáveis relaxadas
3  for  $k \in N_h^v \cup N_n^v$  do
4      for  $z \in \Omega(k)$  do
5           $p_z = \alpha(\sum_{ij \in E^v} f_{ijkz} + f_{ijzk}) + (1 - \alpha)x_{kz}$ 
6      end
7       $z_{max} = \text{argmax}\{p_z | z \in \Omega(k)\}$ 
8      if  $z_{max} = \emptyset$  then
9          Rejeita  $G^v$ 
10     end
11     if  $k \in N_h^v$  then
12         Set  $M_h(k) \leftarrow z_{max}$ 
13     else
14         Set  $M_n(k) \leftarrow z_{max}$ 
15     end
16 end
17 if  $BDC(G^v, G^{s'}, M_h, M_n, M_e)$  then
18     Atualizada capacidades residuais dos
        recursos físicos
19     Return  $M_n, M_e$ 
20 else
21     Rejeita  $G^v$ 
22 end

```

Algorithm 1: QVIA-SDN baseado em (CHOWDHURY; RAHMAN; BOUTABA, 2012).

4.3.2 Busca Determinística de Caminhos

Após identificar os candidatos físicos mais adequados para hospedar as MVs e *switches*, os caminhos físicos para alocar os enlaces virtuais são selecionados, considerando tanto as particularidades do ambiente SDN (controladores, *switches* e tabelas de encaminhamento) como o QoS (largura de banda e latência) definido pelo cliente. Além disso a Busca Determinística de Caminhos (BDC) é guiada pelos valores da variável e . O Algoritmo 2 contém os detalhes da segunda parte da heurística.

A BDC segue os objetivos do provedor (Seção 3.1.4) minimizando a utilização de recursos para diminuir os custos. Nesse caso, os principais recursos considerados são os *switches* SDN. QVIA-SDN faz uso da alocação de entradas na tabela de encaminhamento no controlador e busca caminhos com menor número de saltos.

Inicialmente, o Algoritmo 2 verifica se dois recursos virtuais comunicantes estão alocados no mesmo recurso físico. Nesse caso, é assumido que o tal hospedeiro físico possui capacidade suficiente para respeitar as restrições de largura de banda e latência (linhas 2 a 4). São analisados todos os caminhos entre os candidatos selecionados para hospedar i e j , ou seja, todos os caminhos candidatos para hospedar o enlace virtual ij (linha 5). Essa análise considera a presença ou não de entradas na tabela de encaminhamento do controlador, através dos valores da variável e (linhas 8 a 15). Se existe algum valor em e , QVIA-SDN contabiliza a latência até o controlador ao invés de considerar a alocação na tabela de fluxo do *switch* correspondente.

Caminhos físicos com latência superior a requisitada são descartados. Ideia similar é aplicada em relação a largura de banda, ignorando aqueles caminhos que não atendem os requisitos mínimos (linhas 16 a 19). Entretanto, em alguns casos, o caminho pode ser reconfigurado para atender os requisitos de latência (linhas 20 a 26). Ou seja, ocorre a busca por um caminho que considere o máximo número de valores em e , mas que ainda respeite as restrições de latência.

Neste caso, QVIA-SDN verifica se existe uma combinação de *switches* que pode alocar o enlace virtual respeitando os requisitos de QoS, através da resolução do problema da mochila, considerando o caminho e os valores de e para cada salto do caminho. Dentre os caminhos, são selecionados aqueles com o menor número de saltos e maior número de entradas de fluxo alocadas no controlador (reduzindo a carga dos *switches*). Por fim, na ausência de caminhos físicos para alocar um enlace virtual, a IV é rejeitada.

Input: $G^v, G^{s'}, M_h, M_n, M_e$

Output: Verdadeiro ou falso e os caminhos
para M_e

```

1 for  $ij \in E^v$  do
2   if  $M_n(i) == M_n(j) \vee M_h(i) == M_h(j)$ 
3     then
4       continue
5   end
6   for  $p \in P^s(i, j)$  do
7      $path \leftarrow \{\}$ 
8      $lat\_path \leftarrow 0$ 
9     for  $u \in p$  do
10      if  $e_{ijpu} > 0$  then
11         $lat\_path \leftarrow lat\_path + lat(u, c)$ 
12         $path \leftarrow path + u + controller(u)$ 
13      else
14         $path \leftarrow path + u$ 
15      end
16    end
17    if  $R(path) \geq c_{ij} \wedge lat\_path \leq d_{ij}$  then
18      Set  $M_e(ij) \leftarrow path$ 
19      break
20    else
21       $path = resolveMochila(path, d_{ij})$ 
22      if  $path$  then
23         $M_e(ij) \leftarrow path$ 
24        break
25      else
26        Rejeita  $G^v$ 
27      end
28    end
29 end

```

Algorithm 2: Busca Determinística de Caminhos.

4.4 CONSIDERAÇÕES PARCIAIS

Dada a complexidade de solução de um modelo ótimo, QVIA-SDN resolve um LP resultante do relaxamento do domínio das variáveis binárias. A solução do LP é

combinada com técnicas de redução do número de candidatos físicos, tanto para os nós (servidores e *switches*) como para os caminhos entre cada par de nós. Ainda é aplicada uma redução de candidatos com base em sua localização (zona e região), selecionando apenas uma porcentagem dos candidatos dentro da localização selecionada para a IV. Essa redução de candidatos tem como objetivo, reduzir o espaço de busca da aplicação, aumentando o desempenho e reduzindo a fragmentação dentro de uma mesma zona ou região.

Por fim, QVIA-SDN contém uma heurística composta de duas etapas, responsáveis por interpretar os resultados obtidos para as variáveis f , x e e . A primeira parte da heurística consiste em recriar a correlação entre f e x e então selecionar o candidato mais adequado para cada nó da IV. A segunda parte realiza a seleção de caminhos, respeitando os requisitos de QoS, as particularidades do ambiente SDN e os objetivos do provedor.

O modelo proposto neste trabalho explora diversos tipos de estruturas físicas comuns a *data centers*, entretanto falha em alocar IVs naquelas infraestruturas físicas que utilizam servidores tanto para hospedar MVs como realizar o roteamento de pacotes, tal como a infraestrutura B-Cube (GUO et al., 2009). Ainda, o modelo não consegue usufruir ao máximo da largura de banda que não está destinada para alocação de IVs. Visto que esta largura de banda não está sendo utilizada, ela poderia ser disponibilizada para os clientes, melhorando a qualidade de serviço, e retirada caso houvesse necessidade, mas sem interferir o QoS do cliente.

5 ANÁLISE EXPERIMENTAL

Para avaliar a aplicabilidade de QVIA-SDN, foram desenvolvidos dois cenários experimentais, ambos para alocação IVs em *data centers* gerenciados por SDN. O primeiro conjunto de experimentos conta com IVs com carga de recursos gerada aleatoriamente. Já o segundo conjunto é baseado nas configurações da instância *m3* da Amazon EC2, comumente reservada por clientes (PERSICO et al., 2015).

Os experimentos quantificam a utilização do *data center* na perspectiva do provedor, enquanto na perspectiva do cliente, é avaliado a latência percebida, bem como o tempo de resposta de aceitação ou recusa da IV.

5.1 MÉTRICAS

Para representar os objetivos do provedor da nuvem (Seção 3.1.4) e o QoS percebido pelo cliente, cinco métricas foram coletadas.

- A *razão custo-revenue* fornece uma visão dos lucros do provedor ao alocar uma IV. Ao dividir o custo (Equação (3.1)) pelo *revenue* (Equação (3.2)), o provedor obtém uma perspectiva do número de recursos utilizados para atender a requisição.
- *Fragmentação do data center* indica a porcentagem de recursos ativos, calculada através da divisão do número de recursos ativos pelo número total de recursos disponíveis. Ao consolidar os recursos virtuais em hospedeiros físicos, provedores diminuem os custos administrativos e gerenciais. Por exemplo, servidores e portas de *switches* podem ser desativados.
- O *tempo médio de alocação* de uma requisição representa a aplicabilidade da solução em um ambiente real, sendo que clientes buscam respostas imediatas para requisições submetidas.
- A *taxa de aceitação* de requisições indica o volume de requisições aceitas e provisionadas.
- Por fim, a *latência média* de uma IV é calculada pela soma das latências dos caminhos físicos que hospedam os enlaces virtuais, dividido pelo número de enlaces da IV (Equação (3.3)). Essa métrica indica a qualidade do serviço oferecido ao cliente considerando a comunicação entre os recursos virtuais.

5.2 CENÁRIOS DE SIMULAÇÃO

QVIA-SDN e um simulador de eventos discretos foram implementados em Java v1.8, utilizando CPLEX (v12.6.1.0) para resolver o LP. Os experimentos foram executados em um Intel Xeon E5-2620 2.0GHz - 24 núcleos, 256GB (DDR3) de RAM e 2TB. Para avaliar a aplicabilidade de QVIA-SDN em cenários de nuvem, foi selecionado uma topologia de *data center* representativa (STALLINGS, 2015), bem como topologias de requisições de IVs, comumente utilizadas, descritas a seguir.

5.2.1 Topologia do *data center* de nuvem computacional

A topologia física selecionada foi a *fat-tree* (AL-FARES; LOUKISSAS; VAHDAT, 2008). Uma *fat-tree* é baseada em *switches* com k -portas. Assim, k *pods*, cada um contendo duas camadas de $k/2$ *switches*, agregam até $k^3/4$ servidores. Nesse trabalho, são consideradas duas configurações, $k = 4$ e $k = 8$.

A capacidade dos servidores foi calculada atribuindo pesos a cada um dos recursos (CPU, RAM e armazenamento). Os pesos foram definidos empiricamente e representam a importância de cada recurso no processo de alocação de IVs. Assim, a CPU tem um peso de 50%, a RAM e o armazenamento 25% cada. Tomando como base o servidor no qual a análise experimental foi executada, a capacidade ponderada dos servidores do *data center* é igual a 576.

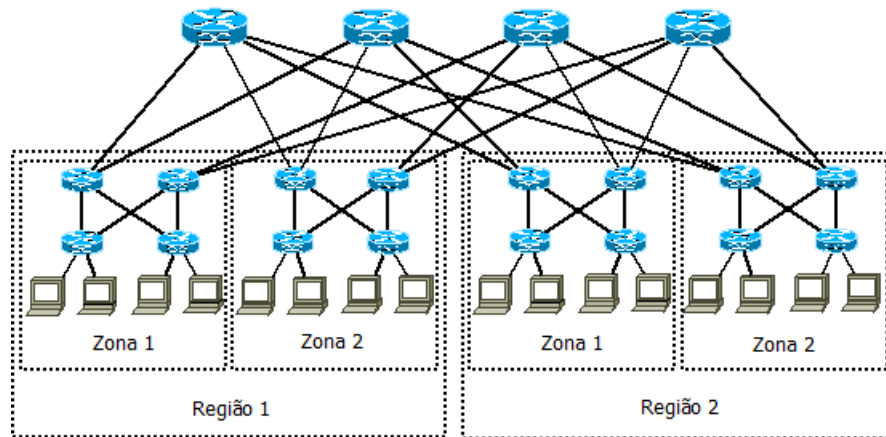
Quanto aos *switches* físicos (núcleo, agregação e aresta), a capacidade foi definida como 100 entradas nas tabelas de encaminhamento. A largura de banda entre os *switches* de núcleo e os *pods* é definida como 10Gbps, e 1Gbps para enlaces dentro dos *pods*. A latência entre qualquer par de recursos físicos é definida como 1ms, enquanto a latência entre os *switches* de núcleo e o controlador SDN é 2ms. Os *switches* de núcleo são diretamente conectados com o controlador SDN, enquanto os outros *switches* estão conectados por caminhos lógicos.

Baseado nos provedores de nuvens públicas Google e Amazon, que definem zonas e regiões de acordo com localizações geográficas, foi desenvolvida uma organização hierárquica de regiões e zonas na *fat-tree* do *data center*. A estrutura hierárquica segue o exemplo da Figura 5: cada *pod* representa uma zona, e cada par de *pods* representa uma região.

5.2.2 Requisições de IVs

A fim de tornar as análises realizadas mais próximas da realidade e evitar a geração de requisições de IVs sem nenhum padrão de comunicação definido, foram selecionadas dois tipos de topologias comumente utilizadas em nuvens privadas e públicas: N-Camadas (NC) e *Virtual Private Clouds* (VPC).

Figura 5 – Distribuição de zonas e regiões no *data center* com topologia *fat-tree*.



Fonte: Elaborado pelo autor, 2017

- **Requisições de IVs N-Camadas.** Diversos inquilinos organizam suas IVs em multi-camadas (JENNINGS; STADLER, 2014), incluindo um balanceador de carga, encarregado de distribuir as requisições para um conjunto de servidores, que eventualmente consultam um banco de dados. Para simulação, cada camada possui um *switch* virtual que solicita QoS de rede, além das MVs. A distribuição de MVs ocorre da seguinte forma: um balanceador de carga, quatro servidores e quatro bancos de dados.
- **Requisições de IVs VPC.** Amazon EC2 introduziu o provisionamento de VPCs¹, que é composta por um conjunto de regras de acesso e um conjunto de MVs, compondo uma rede privada, gerenciada pelo cliente. Para compor as requisições VPC, um conjunto de 9 MVs é conectado a um *switch* SDN, representando as regras de acesso.

5.3 SIMULAÇÕES

Foram realizados dois conjuntos de simulações, ambos com requisições de IVs com NC e VPC, variando em relação a configuração de cada IV. No primeiro cenário a carga das requisições foi gerada aleatoriamente, enquanto o segundo cenário tem cargas geradas baseados nas instâncias *m3* da Amazon.

¹ Virtual Private Cloud (VPC): <<https://aws.amazon.com/vpc/>>.

5.3.1 Simulação com cargas uniformemente distribuídas

Neste primeiro cenário as requisições de IVs tem suas configurações e requisitos gerados de maneira aleatória, de acordo com uma distribuição uniforme. Esta simulação tem como objetivo identificar o comportamento do modelo, a qualidade do serviço oferecida e a carga do *data center*.

5.3.1.1 Especificação

Para ambas as topologias de IVs, a capacidade virtual é definida como uma fração da capacidade física: cada *switch*, MV ou enlace virtual tem entre 5 e 15% da capacidade física (de acordo com uma distribuição uniforme). Com base no exemplo da Figura 5, a localização geográfica das IVs é escolhida aleatoriamente entre as regiões e zonas da *fat-tree*. Primeiro é definida a localização da região, então é definida a zona. Em alguns casos a localização da IV pode conter somente a região, com 50% de probabilidade.

Para representar um período de utilização de um *data center*, um conjunto com 50 requisições (NC e VPC) é submetido para cada cenário físico. Os intervalos de chegada das IVs são uniformemente distribuídos em até 100 intervalos discretos, e o período máximo de atividade de uma IV é de 30 intervalos.

5.3.1.2 Resultados

Os resultados representam a média das amostras com um intervalo de confiança de 95% comparando QVIA-SDN com dois algoritmos base. O primeiro algoritmo segue uma abordagem comum na literatura (CHOWDHURY; RAHMAN; BOUTABA, 2012), representando a alocação em um ambiente de nuvem sem SDN, nomeado de Non-SDN (NSDN). A fim de avaliar a eficácia do controle de latência, o segundo algoritmo não contém tais restrições, identificado por Sem Controle de Latência (SCL). Cada algoritmo, é executado em três cenários diferentes.

A restrição no número de candidatos físicos introduzidos na Seção 4.2 é avaliada variando a porcentagem de candidatos entre 60, 80 e 100%. Tendo como base observações empíricas, QVIA-SDN parametriza $\alpha = 0.9$ (Seção 4.3) e $\beta_u = \gamma_e = \alpha_{uv} = 1$ (Seção 3.2.2).

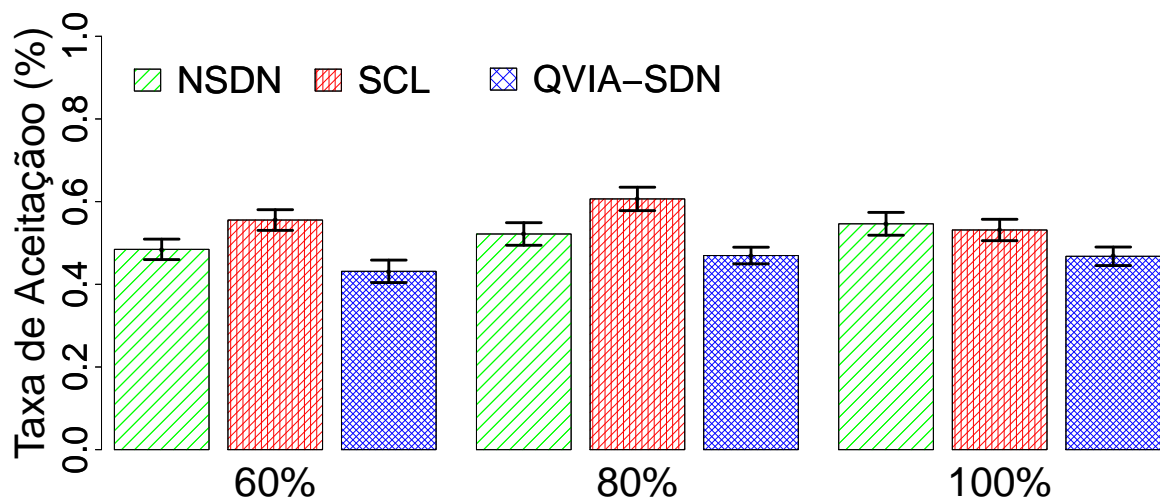
- **Taxa de Aceitação.** A taxa de aceitação para $k = 4$ e $k = 8$ é representada pelas Figuras 6a e 6b, respectivamente. A Figura 6a indica uma baixa taxa de aceitação, com pouca variação, independente do número de candidatos (60, 80 e 100%) analisados. Esta baixa taxa de aceitação é justificada pelo reduzido

número de recursos físicos, indicando um saturamento da ocupação do *data center*.

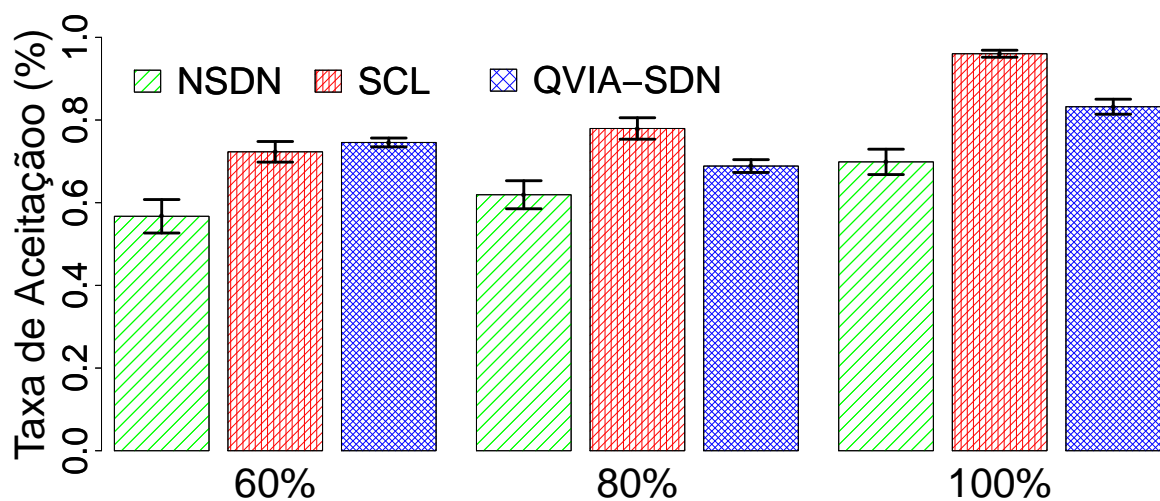
Já na Figura 6b, o cenário é um pouco diferente, pois com mais recursos QVIA-SDN apresenta uma tendência para agrupar recursos para redução da latência fornecida. Essa tendência, de agrupamento leva a uma redução de recursos físicos utilizado, e aumento da taxa de aceitação.

Figura 6 – Taxa de aceitação da alocação de IVs.

(a) $k = 4$



(b) $k = 8$



Fonte: Elaborado pelo autor, 2017.

- **Latência Média.** A Figura 7 contém os resultados da distribuição acumulada da latência média normalizada. A latência média observada por clientes que tiveram suas requisições de IV alocadas com QVIA-SDN é menor quando comparado com as alocadas por SCL, e em alguns casos superando a latência do ambiente sem SDN (NSDN).

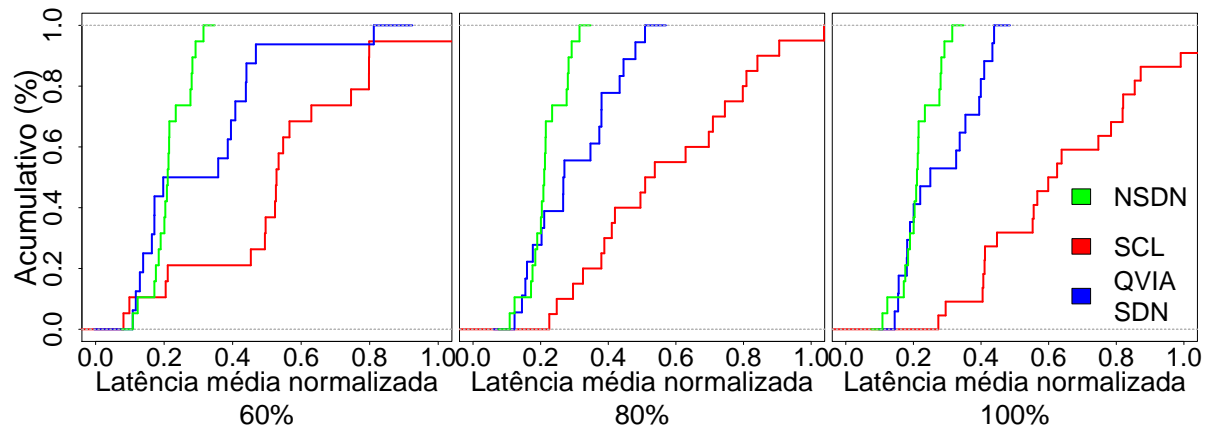
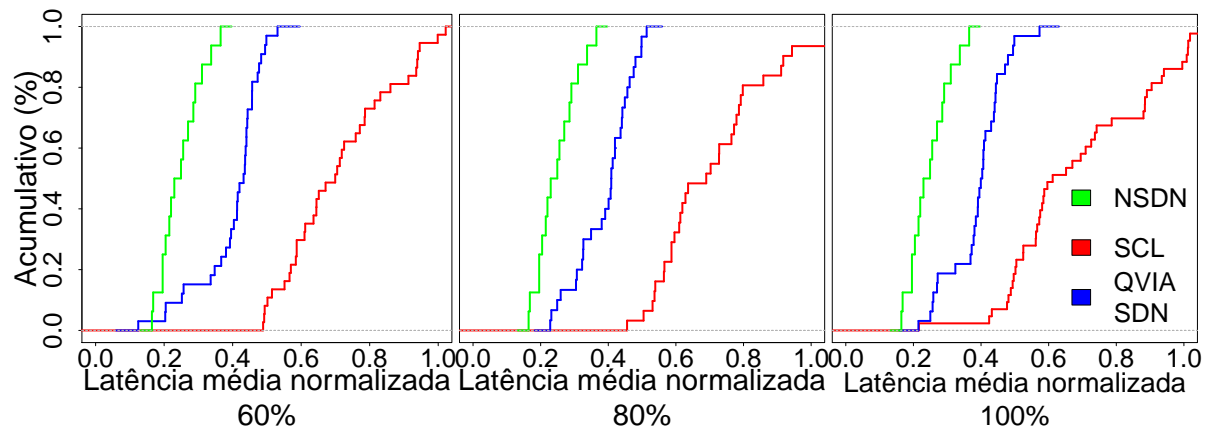
Em média, a latência oferecida pelo ambiente NSDN, é menor, porém com uma menor taxa de aceitação. Por não ter a possibilidade de alocação de fluxos no controlador, NSDN consome mais da infraestrutura física (entradas nas tabelas dos *switches*), levando a uma menor taxa de aceitação. Enquanto isso QVIA-SDN respeita o QoS definido pelo cliente e aloca um número maior de IVs, justificando sua aplicação.

Ainda, a Figura 8 indica que QVIA-SDN possui baixa variabilidade na latência oferecida. SCL obtém a maior variabilidade de latência por não oferecer nenhum mecanismo de controle. Seguindo a tendência da Figura 7 NSDN resulta na menor variabilidade.

- **Fragmentação.** A fragmentação para $k = 4$ e $k = 8$ é exibida nas Figuras 9a e 9b, respectivamente. Embora QVIA-SDN procure condensar os recursos virtuais no *data center*, os resultados indicam uma alta fragmentação. Os números são justificados pelo número de alocações realizadas com sucesso por QVIA-SDN, superior aos demais algoritmos. Ainda, devido a garantia de controle de latência, QVIA-SDN tem uma alta fragmentação de *switches*. É importante destacar, que além de uma maior taxa de aceitação QVIA-SDN e SCL apresentam índices de fragmentação competitivos em relação a NSDN.
- **Razão custo-revenue.** De acordo com as Figuras 10a e 10b todos os 3 modelos de alocação possuem resultados equivalentes em relação a razão entre custo e *revenue*. Estes dados indicam que, mesmo alocando mais IVs é possível ter tanto uma fragmentação, como uma razão custo-revenue competitiva com as demais soluções.
- **Tempo médio para alocação de requisições de IV.** O tempo médio de processamento para análise (alocação ou rejeição) das requisições são apresentados nas Figuras 11a e 11b. Como esperado, conforme o número de requisitos e espaço de busca aumenta, devido a introdução do ambiente SDN ou então um ambiente com mais candidatos ($k = 8$), o número de restrições no LP aumentava, e consequentemente o tempo de processamento.

Assim, QVIA-SDN requer um maior tempo de processamento devido ao grande número de restrições empregadas no modelo. Estas restrições representam as dimensões introduzidas pelo ambiente SDN, bem como as restrições de QoS.

Figura 7 – Distribuição Acumulada da latência média normalizada

(a) $k = 4$ (b) $k = 8$ 

Fonte: Elaborado pelo autor, 2017.

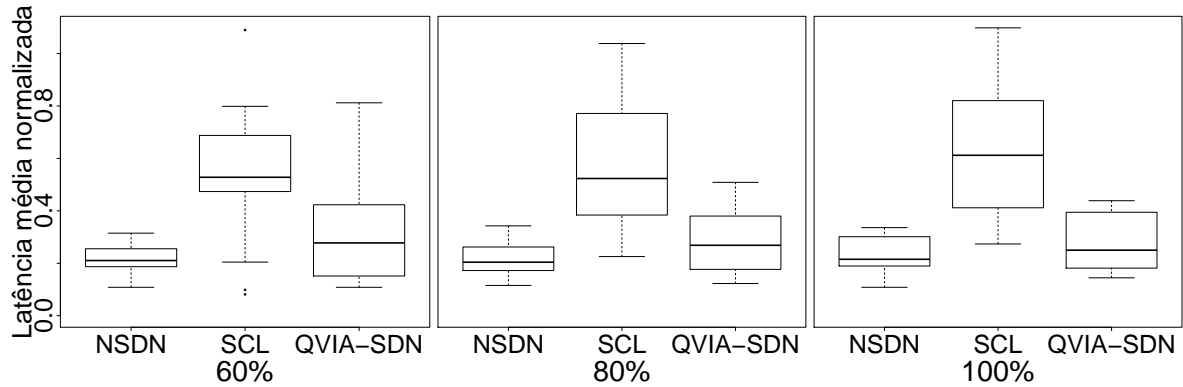
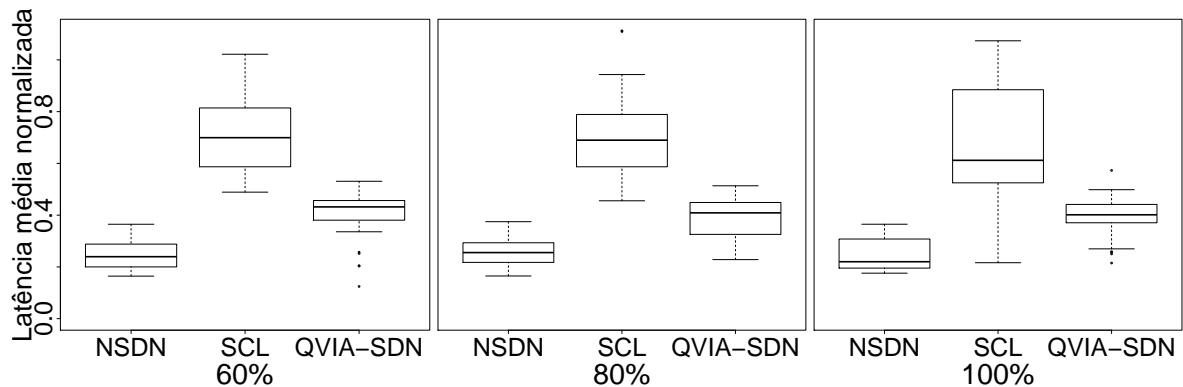
Como os outros dois algoritmos são mais brandos (menos restritivos) são compostos por um número menor de restrições. Porém, apesar da diferença no número de restrições, o tempo de alocação está compreendido em um intervalo de poucos segundos.

5.3.2 Simulação baseada na instância $m3$ da Amazon

Neste segundo cenário, as capacidades solicitadas para composição das IVs são baseadas nas instâncias $m3$ da Amazon EC2². O objetivo é identificar o compor-

² Instâncias $m3$ - Amazon EC2: <<https://aws.amazon.com/pt/ec2/instance-types/>>

Figura 8 – Variabilidade da latência média

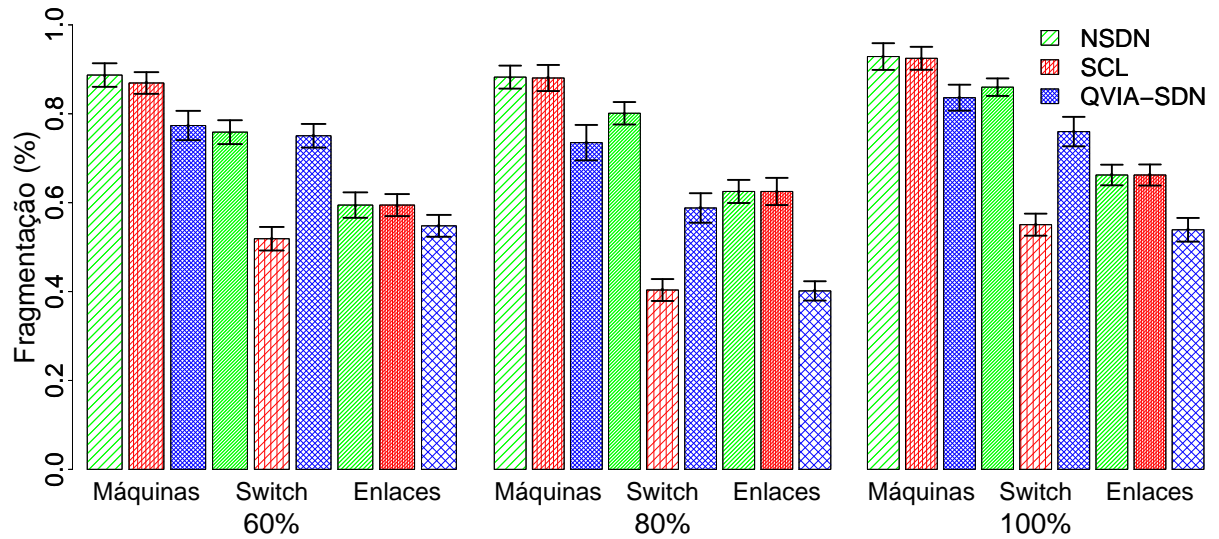
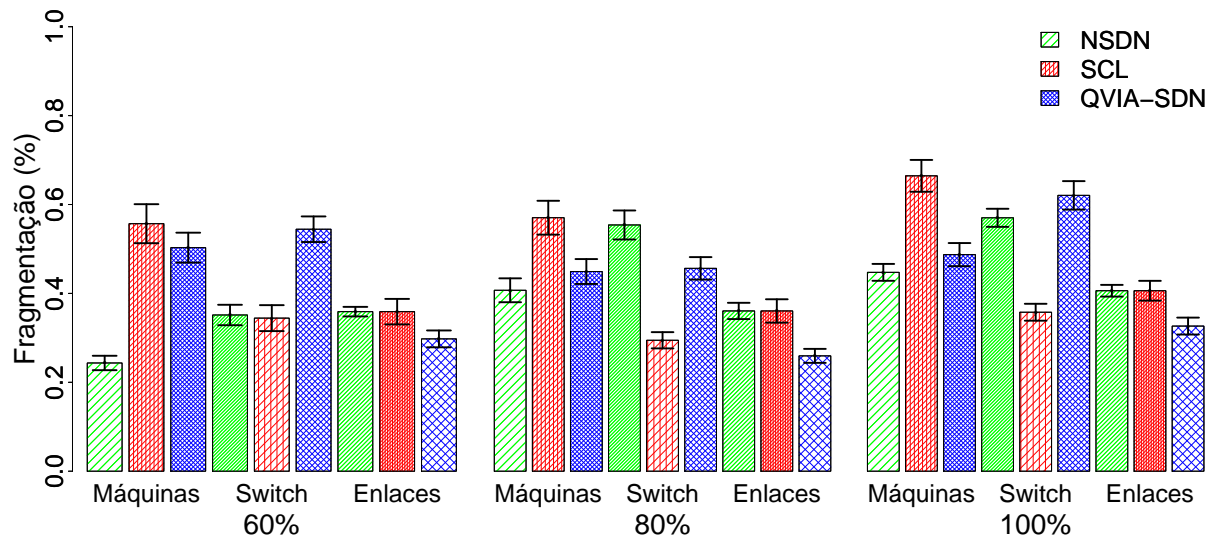
(a) $k = 4$ (b) $k = 8$ 

Fonte: Elaborado pelo autor, 2017.

tamento de QVIA-SDN alocando com configurações populares em nuvens computacionais com ambiente SDN.

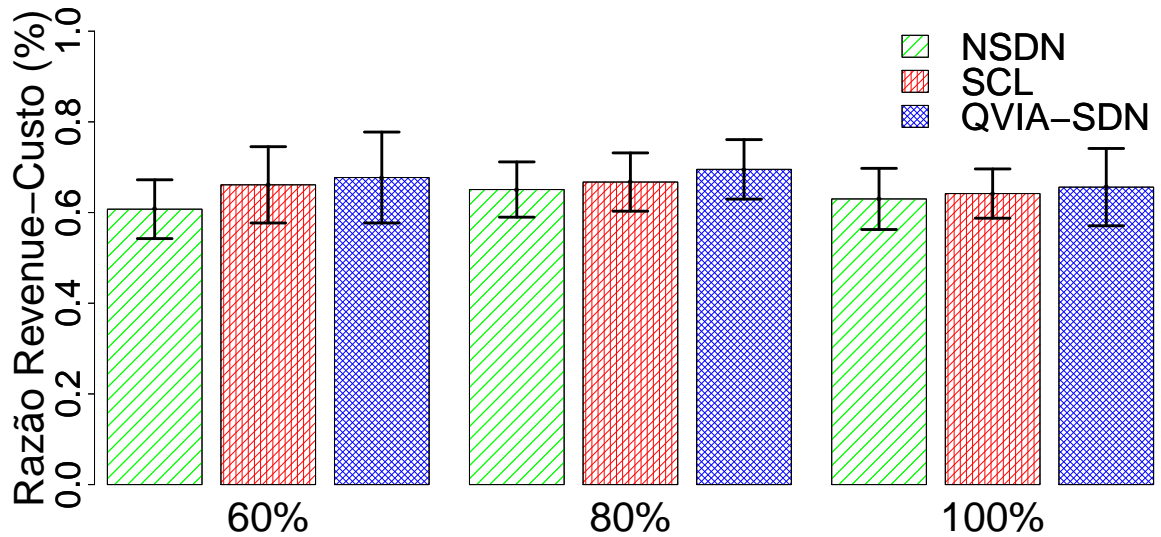
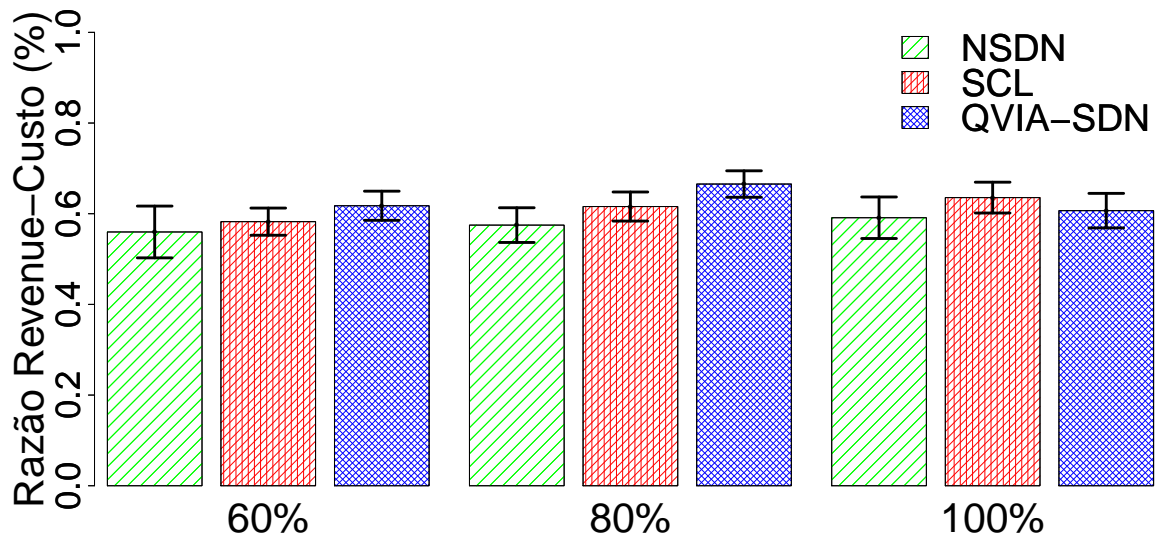
5.3.2.1 Especificação

As capacidades dos recursos virtuais seguem as instâncias *m3* da Amazon nas configurações *medium*, *large*, *xlarge* e *2xlarge*. Foram escolhidas as instâncias *m3* por sua popularidade, e assim representarem uma carga comum dentro da nuvem. A capacidade das MVs é calculada utilizando os mesmos pesos aplicados para o *data center* (Seção 5.2.1) e os valores da Tabela 3. Assim, as capacidades requisitadas para MVs baseadas nas instâncias *medium*, *large*, *xlarge* e *2xlarge*, são respectivamente 2, 10, 25 e 51.

Figura 9 – Fragmentação do *Data Center*(a) $k = 4$ (b) $k = 8$ 

Fonte: Elaborado pelo autor, 2017.

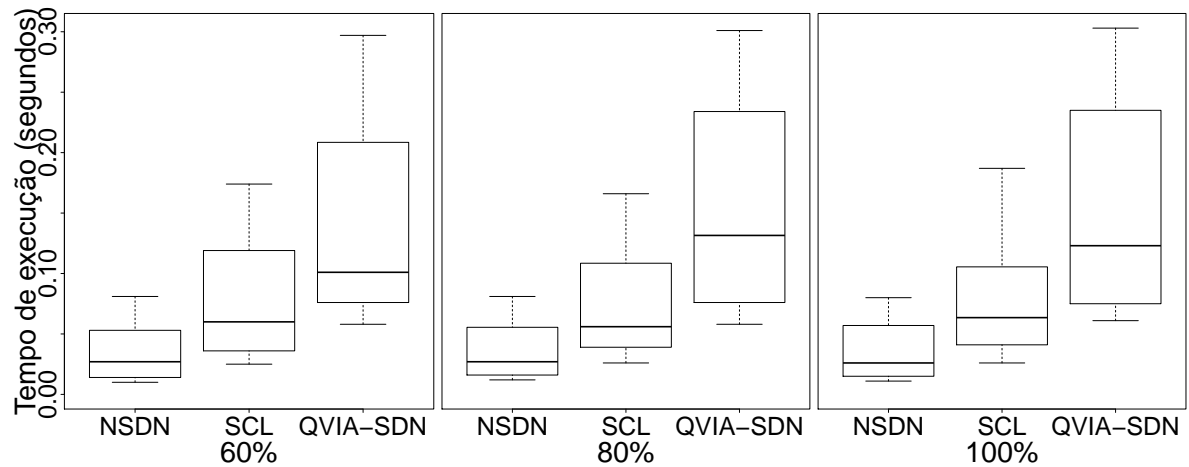
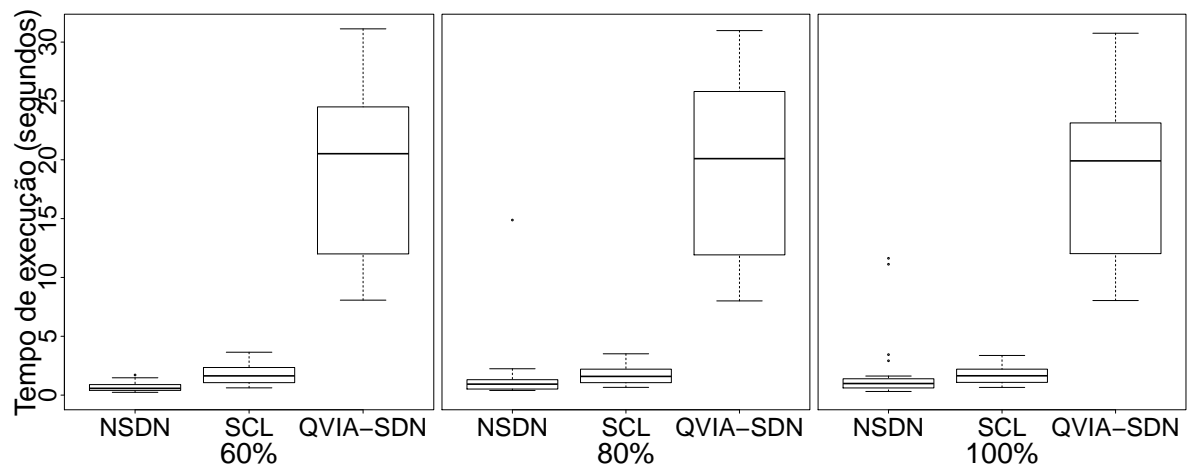
Por sua vez, a capacidade dos enlaces virtuais segue uma distribuição normalizada entre 10, 20, 40 e 80% das larguras de banda média observadas por cada instância (PERSICO et al., 2015), exibidas na Tabela 3. Quanto a localização geográfica, todas as IVs recebem uma região, porém existe 50% de probabilidade de uma IV especificar uma zona.

Figura 10 – Razão *cost-revenue*(a) $k = 4$ (b) $k = 8$ 

Fonte: Elaborado pelo autor, 2017.

Um conjunto com 50 requisições (VPC ou NC) igualmente distribuído entre os quatro tipo de instâncias $m3$ é submetido para cada cenário físico. O tempo de chegada de cada IV é uniformemente distribuído entre 100 intervalos discretos e cada IV permanece ativa por no máximo 30 intervalos.

Figura 11 – Tempo médio para alocação de requisições de IV

(a) $k = 4$ (b) $k = 8$ 

Fonte: Elaborado pelo autor, 2017.

Tabela 3 – Detalhamento dos recursos das instâncias *m3* da Amazon EC2.

Modelo	vCPU	Memória (GB)	Armazenamento (GB)	Largura de banda (Gbps)
<i>medium</i>	1	3,75	4	492
<i>large</i>	2	7,5	32	739
<i>xlarge</i>	4	15	80	958
<i>2xlarge</i>	8	30	160	1188

Fonte: Elaborado pelo autor, 2017.

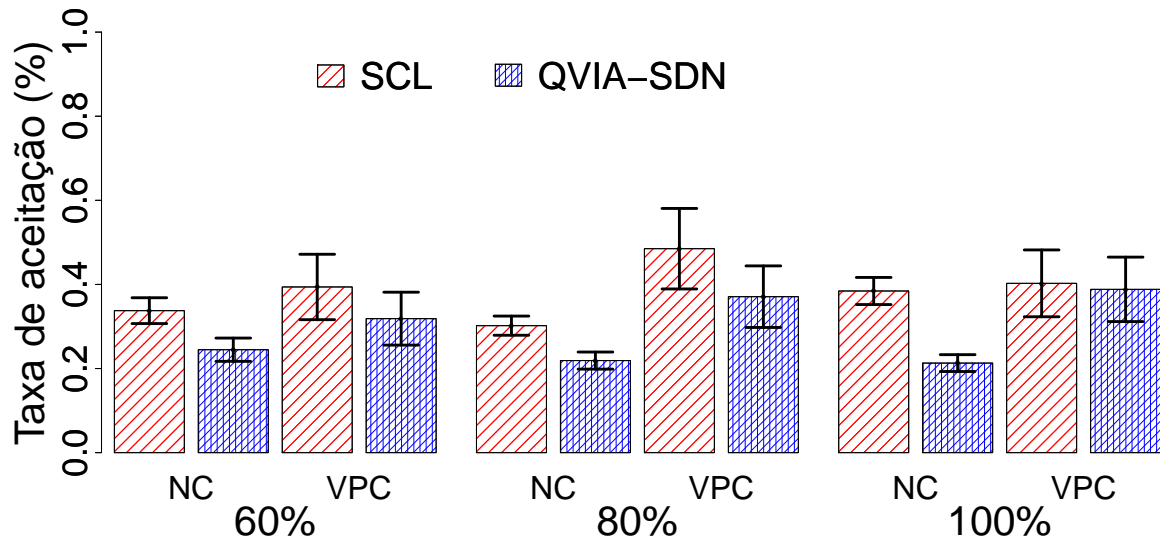
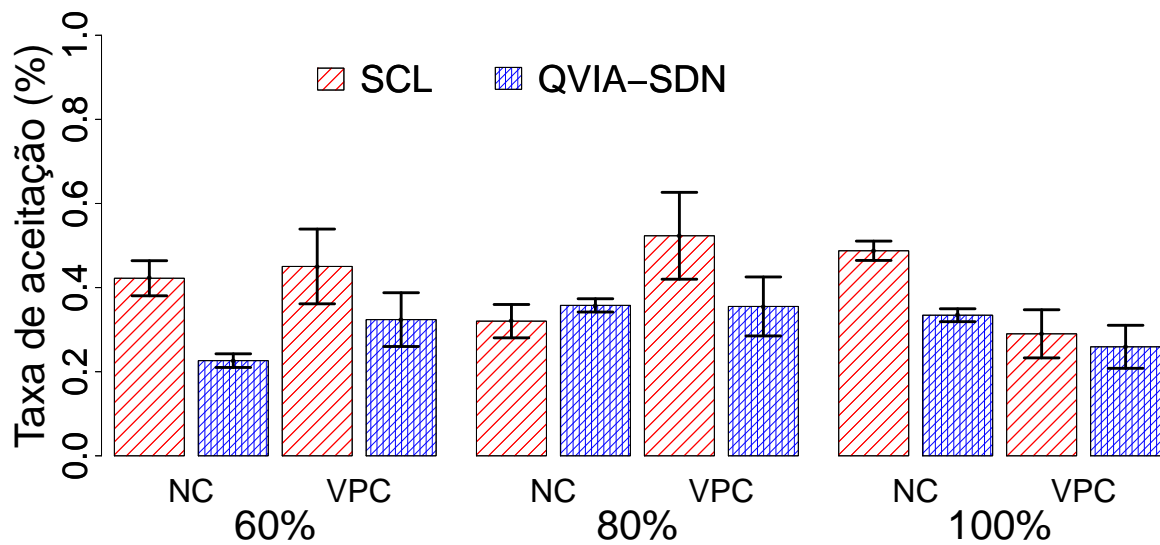
5.3.2.2 Resultados

Os resultados mostram médias com 95% de intervalo de confiança. Visto que o objetivo é identificar o comportamento de alocações em ambientes SDN, o algoritmo NSDN é descartado, portanto QVIA-SDN é comparado com um algoritmo baseado na proposta de (CHOWDHURY; RAHMAN; BOUTABA, 2012) (Sem Controle de Latência (SCL)). Novamente, cada execução difere em relação a porcentagem de candidatos que é avaliada, variando entre 60, 80 e 100%, e as configurações previamente utilizadas foram mantidas $\alpha = 0.9$ (Seção 4.3) e $\beta_u = \gamma_e = \alpha_{uv} = 1$ (Seção 3.2.2).

- **Taxa de Aceitação.** A Figura 12 contém os resultados de taxa de aceitação para $k = 4$ e $k = 8$. Os resultados da Figura 12a indicam que diante de poucos recursos é possível alocar mais requisições de VPCs. Independente do número de candidatos, a taxa de aceitação apresentou pouca variância. Para $k = 8$ (Figura 12b) existe uma equivalência entre os dois tipos de IVs.
- **Latência Média.** A Figura 13 exibe a distribuição acumulada da latência média normalizada. As IVs alocadas com QVIA-SDN apresentam uma latência média inferior, independente de tipo de requisição, enquanto as requisições NC apresentam as menores latências, tanto para $k = 4$ como $k = 8$. Outro ponto que merece destaque é que de acordo com $k = 8$, a diferença de latência ganhou evidência.

NC apresentou uma menor latência média, em outras palavras, utilizou mais entradas nas tabelas de encaminhamento dos *switches*, o que acarretou em uma menor taxa de aceitação. Por fim, devido aos mecanismos de controle de latência introduzidos por SDN, a variabilidade da latência foi menor para QVIA-SDN (Figura 14).
- **Fragmentação.** As Figuras 15a e 15b indicam os resultados de fragmentação para $k = 4$ e $k = 8$, respectivamente. Os resultados de fragmentação estão alinhados com as latências médias observadas. Latências baixas, indicam proximidade dos equipamentos da IV, porém como cada IV possui uma localização diferente, a baixa latência não permite distanciar os elementos da IV para realizar a alocação em equipamentos físicos que já estavam alocando outras IVs. Isso acarreta uma maior fragmentação. Como NC obteve uma menor latência média, teve também uma maior fragmentação, e como mais elementos da sua infraestrutura estavam sendo utilizados, teve uma menor taxa de aceitação.
- **Razão custo-revenue.** As Figuras 16a e 16b indicam que existe uma equivalência entre QVIA-SDN e SCL, bem como entre NC e VPC. Ainda, é importante

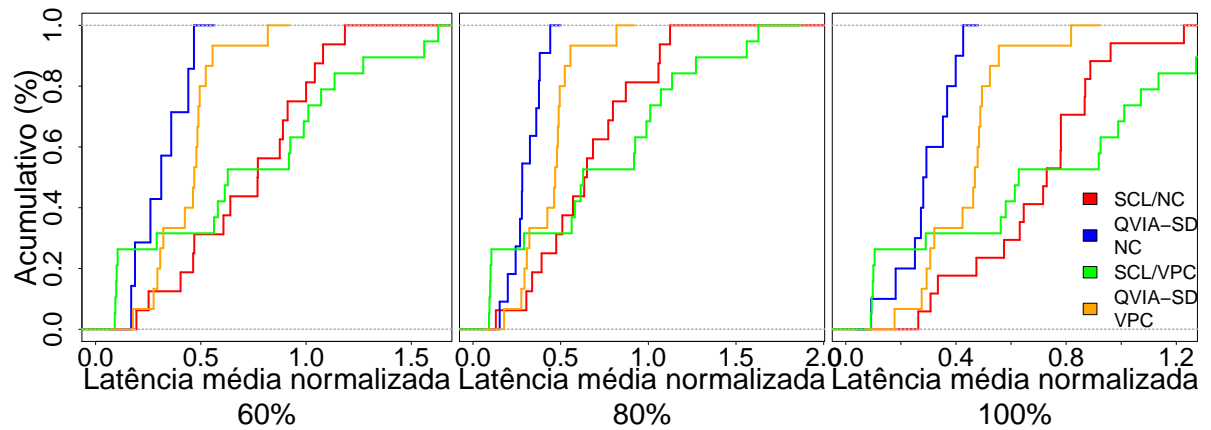
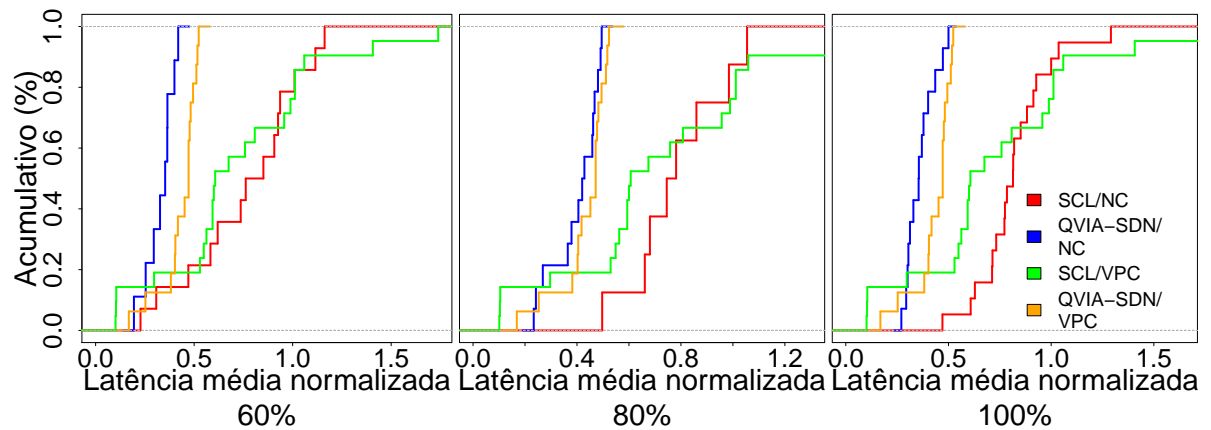
Figura 12 – Taxa de aceitação da alocação de IVs.

(a) $k = 4$ (b) $k = 8$ 

Fonte: Elaborado pelo autor, 2017.

realizar uma relação com a taxa de aceitação. SCL obteve maior aceitação, porém uma menor razão entre custo e *revenue*. Ou seja, os cenários com QVIA-SDN conseguiram alocar IVs com cargas maiores do que aquelas alocadas por SCL. O mesmo raciocínio pode ser aplicado a VPC em relação a NC. Por fim, usando QVIA-SDN é possível melhorar a latência na perspectiva do cliente sem

Figura 13 – Distribuição Acumulada da latência média normalizada

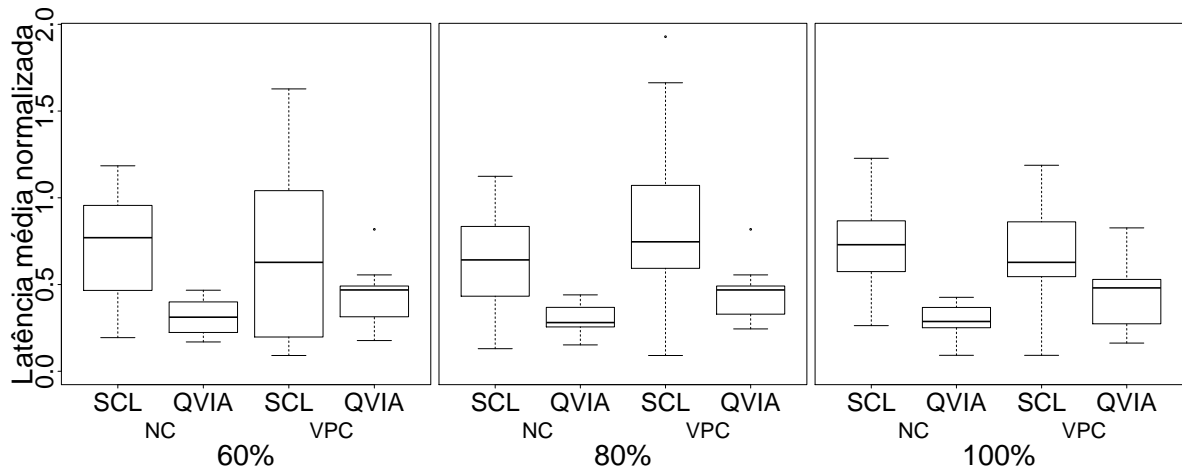
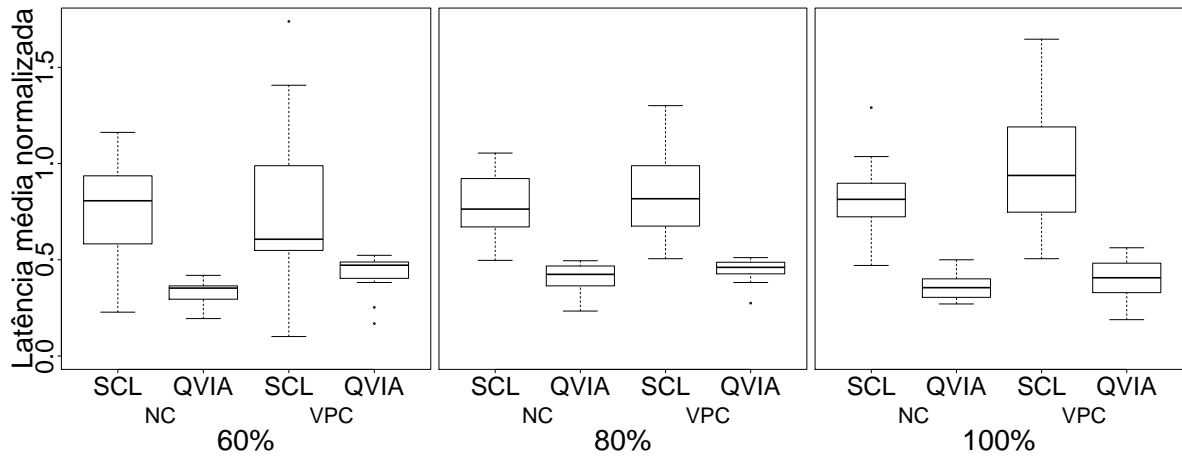
(a) $k = 4$ (b) $k = 8$ 

Fonte: Elaborado pelo autor, 2017.

prejudicar as métricas que representam a perspectiva do provedor.

- **Tempo médio para alocação de requisições de IV.** Os tempos médios de alocação são apresentados na Figura 17. Novamente, devido ao maior conjunto de restrições, QVIA-SDN tem um pior desempenho em termos de tempo de alocação, com o pior caso observado para IVs de NC. Entretanto, o tempo de alocação não ultrapassa 30 segundos no pior caso.

Figura 14 – Variabilidade da latência média

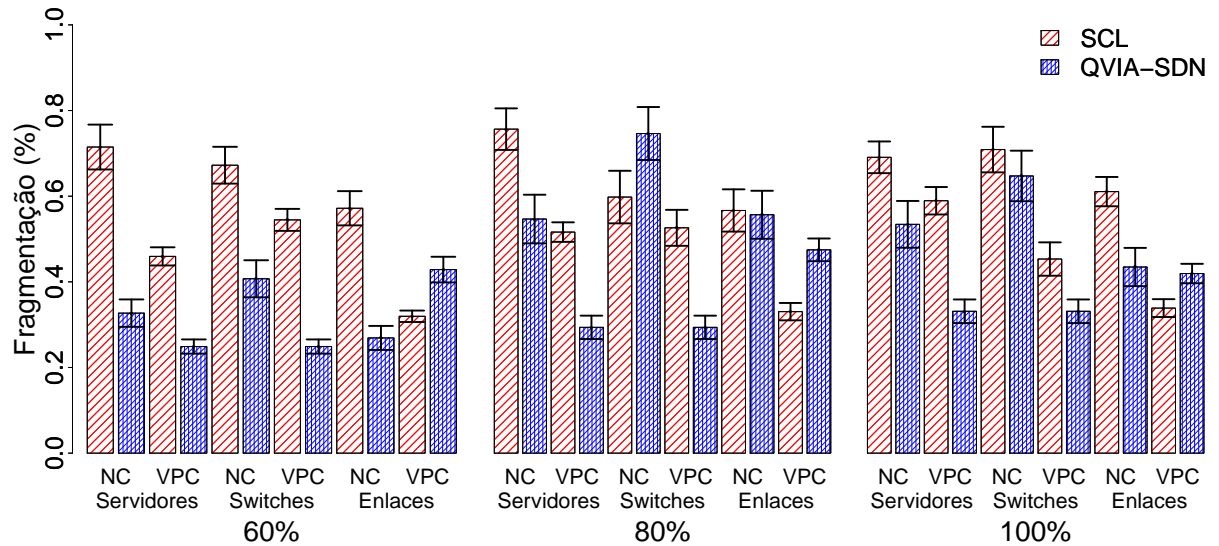
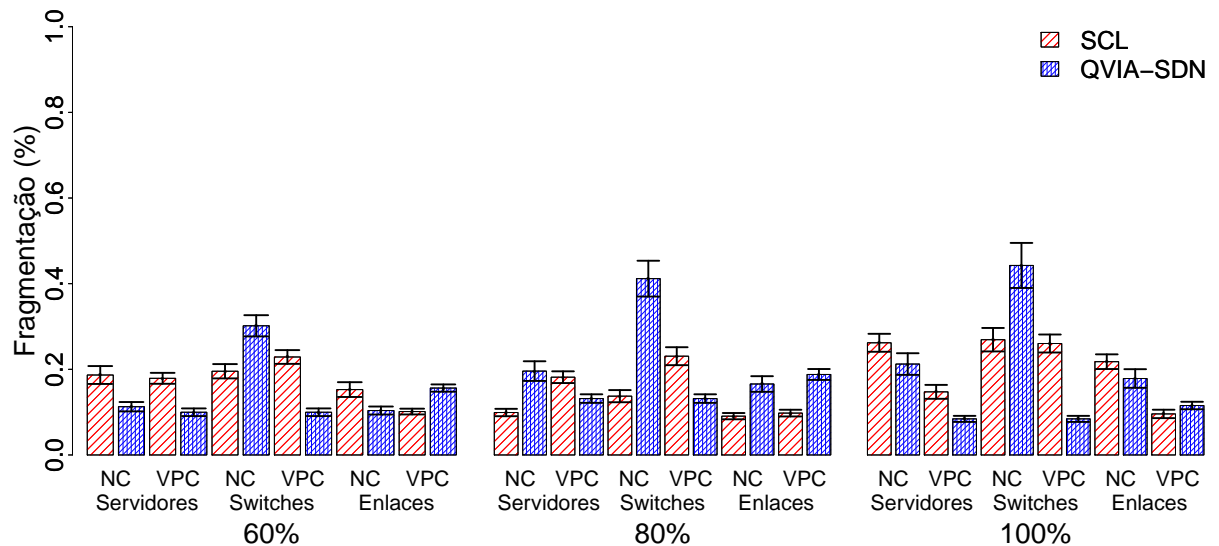
(a) $k = 4$ (b) $k = 8$ 

Fonte: Elaborado pelo autor, 2017.

5.3.3 Discussão e principais observações

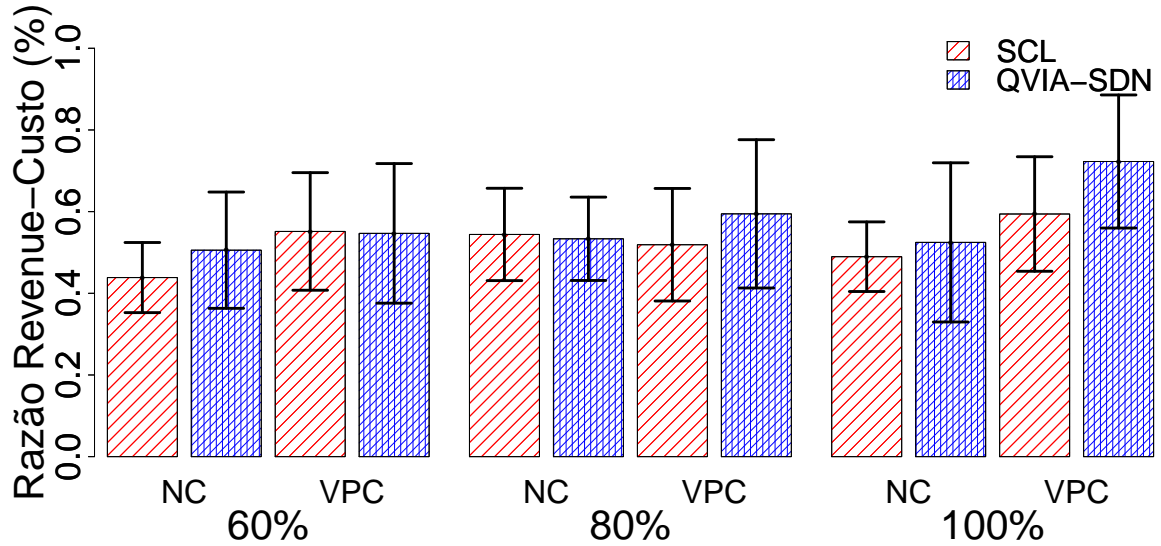
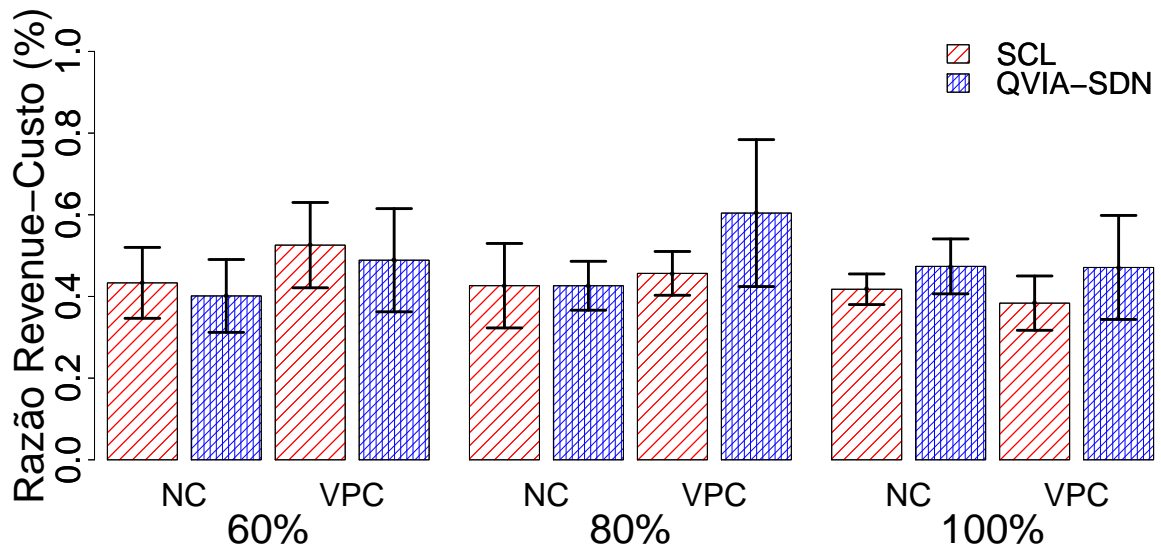
O primeiro cenário de teste, indicou que é possível realizar a alocação em ambientes SDN, fornecendo garantias de QoS, mantendo uma taxa de aceitação competitiva, quando comparado as soluções de base. Como QVIA-SDN posiciona entradas das tabelas de encaminhamento de fluxos no controlador, a fragmentação dos *switches* foi diminuída. Do ponto de vista do provedor, a razão custo-revenue de QVIA-SDN se mostra equivalente aos outros algoritmos comparados.

Já o segundo cenário de testes avaliou o desempenho do mecanismo em ce-

Figura 15 – Fragmentação do *data center*(a) $k = 4$ (b) $k = 8$ 

Fonte: Elaborado pelo autor, 2017.

nários próximos aos vivenciados por provedores de nuvens computacionais. Quando alocadas com QVIA-SDN, NC teve uma melhor latência média, mas VPC pode ser considerada a infraestrutura com melhor custo-benefício, tanto para o cliente como o provedor. Do ponto de vista do cliente, QVIA-SDN aloca requisições VPC respeitando os requisitos de QoS, embora sacrifique latência média. Pela perspectiva do provedor,

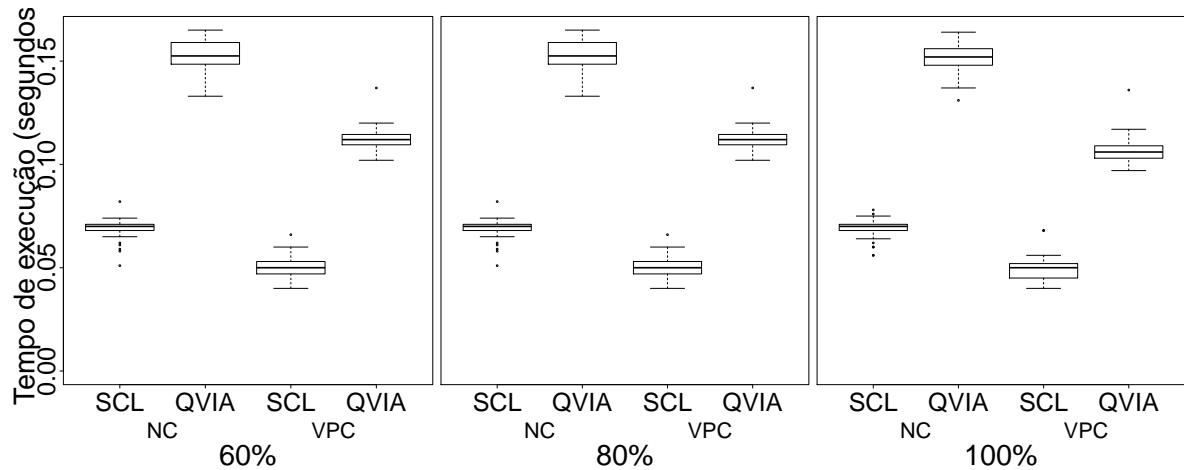
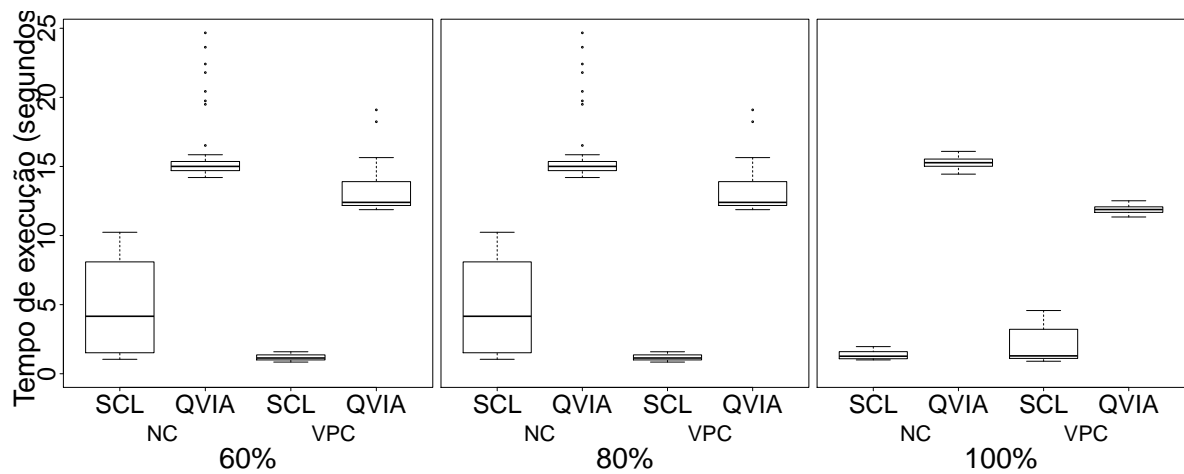
Figura 16 – Razão *cost-revenue*(a) $k = 4$ (b) $k = 8$ 

Fonte: Elaborado pelo autor, 2017.

requisições VPC alocadas com QVIA-SDN obtiveram melhores resultados em todos os aspectos (taxa de aceitação, menor fragmentação e maior razão custo-*revenue*).

Em termos gerais os resultados indicam que é possível fornecer garantias de QoS sem afetar a taxa de aceitação. Ainda, a utilização do ambiente SDN além

Figura 17 – Tempo médio para alocação de requisições de IV

(a) $k = 4$ (b) $k = 8$ 

Fonte: Elaborado pelo autor, 2017.

de fornecer mecanismos para auxiliar na garantia de QoS, aumenta o *revenue* do provedor. As diferentes porcentagens de candidatos avaliada indica que é possível encontrar um solução adequada sem avaliar todo o espaço de busca.

Por fim, devido ao maior conjunto de restrições e a complexidade do processo de alocação QVIA-SDN tem um maior tempo de alocação. Entretanto, diante do volume de restrições e da complexidade do processo, o tempo decorrido de 30 segundos no pior caso pode ser considerado aceitável para alocação *on-line* de IVs.

5.4 CONSIDERAÇÕES PARCIAIS

Apesar de SDN introduzir vantagens no processo de provisionamento de IVs, também acrescentou novas dimensões ao processo de alocação. Portanto, afim de avaliar o mecanismo proposto, foi desenvolvida uma análise experimental, quantificando cinco métricas, que avaliavam a taxa de aceitação, latência média, fragmentação, razão custo-*revenue* e tempo médio de alocação.

QVIA-SDN foi analisado com requisições de dois tipos comuns de IVs, NC e VPC, variando a metodologia de composição das capacidades virtuais solicitadas. As capacidades das requisições variaram em dois cenários. O primeiro cenário teve um aspecto mais generalista, com capacidades definidas de maneira aleatória. Já o segundo cenário, buscou uma avaliação mais realista do modelo, com capacidades reais, baseadas em instâncias *m3* da Amazon.

O ambiente SDN fornece diversos benefícios para alocação de IVs, e os resultados corroboram com esta afirmação. QVIA-SDN atende os objetivos do provedor com uma taxa de aceitação competitiva em relação a algoritmos base, pois exploram a alocação de fluxos no controlador, para redução da fragmentação. Ainda, o QVIA-SDN reduz a razão custo-*revenue* experienciada pelo provedor. Para o cliente QVIA-SDN fornece uma melhora no QoS experienciado, visto a redução de latência média. Além de garantir uma largura de banda mínima.

A redução da porcentagem de candidatos avaliados indica que é possível atingir valores satisfatórios para as métricas do provedor, fornecer QoS para o cliente, sem ter que analisar todo o espaço de busca. Por fim, o modelo proposto apresenta um tempo de que pode ser considerado alto, porém analisando a complexidade do processo e o volume de restrições, este tempo é considerado aceitável.

6 CONCLUSÃO

A migração de aplicações para nuvens computacionais é um fato e continua em constante crescimento. Um dos modelos de negócios adotado para migração em nuvens é o modelo IaaS que fornece uma infraestrutura virtualizada composta de equipamentos de processamento e equipamentos de comunicação. Entretanto, é comum os provedores de nuvem atribuam um foco comercial e técnico diferenciado para os recursos internos de uma IV. Por exemplo, o provisionamento de redes de comunicação é normalmente realizado sem garantia de requisitos de qualidade de serviço. Ainda, em diversos casos, o cliente não possui nenhum controle sobre a comunicação entre as máquinas virtuais. Trabalhos recentes indicaram que o provisionamento de redes de comunicação em nuvens computacionais influenciam diretamente no desempenho das aplicações hospedadas em IVs. Assim, o presente trabalho ressalta o aspecto relacionado com a comunicação interna das IVs, conectando o provisionamento de IaaS com serviços de *Network-as-a-Service* (NaaS). Dentre as tarefas gerenciais efetuadas para o provisionamento de uma IV, o processo de alocação de recursos para hospedar uma infraestrutura virtual é computacionalmente custoso, e consiste em identificar dentro do *data center* quais equipamentos estão aptos a atender a requisição. Além disso, devido a rigidez das redes de computadores, a configuração e manutenção realizada para hospedar uma rede virtual é de alta complexidade.

Diante de tais dificuldades, SDN e OpenFlow surgem como alternativa, pois fornecem mecanismos para garantia de QoS (largura de banda e latência), facilitando o processo de configuração da rede ao separar os planos de dados e de controle, e fornecendo uma interface padronizada de configuração via *software*. Entretanto, SDN introduz novas dimensões que aumentam a complexidade do processo de alocação de recursos físicos para hospedar IVs. Por exemplo, a possibilidade de alocação de fluxos no controlador permite a redução da utilização da infraestrutura física, mas aumenta a complexidade do mecanismo introduzindo um conjunto maior de possibilidades no espaço de busca por uma solução.

Diante das limitações e desafios do processo de alocação e do ambiente SDN, é proposto um mecanismo específico, denominado QVIA-SDN. QVIA-SDN é composto por um modelo representado como um MIP que posteriormente tem suas restrições relaxadas compondo um LP. Ainda, uma heurística, composta de duas etapas, para interpretação dos resultados obtidos com a solução do LP compõe o mecanismo. O MIP proposto busca respeitar os objetivos do provedor de minimização de custos e da taxa custo-*revenue*, bem como os objetivos do cliente (sobretudo requisitos de QoS para latência). Para isso, são aplicadas restrições de capacidade para os servidores, *swit-*

ches e enlaces, tanto físicos como virtuais. Ainda, são aplicadas também restrições relativas ao ambiente SDN que buscam garantir a latência de comunicação enquanto reduzem a utilização da infraestrutura física.

Com o objetivo de avaliar o QVIA-SDN um simulador de eventos discretos foi desenvolvido. QVIA-SDN foi submetido a dois cenários de simulação. O primeiro com cargas geradas aleatoriamente, e o segundo com base em cargas próximas a realidade de provedores de nuvens computacionais. Os resultados indicam que é possível fornecer qualidade de serviço obtendo taxas de aceitação competitivas. A fragmentação observada nos recursos do *data center* indicam que a alocação de fluxos no controlador reduz a utilização dos equipamentos físicos, pois condensa MV em servidores já utilizados e aloca fluxos no controlador sem saturar os *switches*.

Os resultados do segundo cenário, indicam o sucesso de QVIA-SDN ao alocar requisições VPC em um ambiente de nuvem SDN. Tanto provedor quanto clientes percebem um melhor custo-benefício pois o mecanismo respeita o QoS definido pelo cliente e tem melhores taxa de aceitação e resultados para métricas do provedor. Por fim, as reduções de porcentagens de candidatos físicos aplicadas, indicam que é possível alocar IVs sem avaliar todo o espaço de busca. No que diz respeito aos tempos de alocação, QVIA-SDN encontrou uma limitação, entretanto diante do volume de restrições geradas pelo ambiente SDN e pelos requisitos de QoS, o tempo de espera (de no máximo 30 segundos) para alocação de uma IV pode ser considerado aceitável.

6.1 TRABALHOS FUTUROS

Tanto o modelo proposto como a análise realizada apresentam algumas limitações e oportunidades de continuidade do trabalho. Inicialmente, uma implementação de QVIA-SDN em um *framework* de gerenciamento de nuvens computacionais permitirá a quantificação do impacto direto nas aplicações hospedadas. Além disso, o mecanismo pode ser combinado com soluções para realocação de recursos virtuais, que representam o provisionamento elástico de recursos. É esperado que o padrão de comunicação de uma aplicação evolua, resultando em requisições de adaptação da rede virtual. Tais requisições impactam diretamente na configuração das tabelas dos *switches*.

6.2 TRABALHOS PUBLICADOS

A Tabela 4 apresenta a lista de publicações realizadas durante a elaboração do trabalho.

Tabela 4 – Publicações realizadas.

Local	Ano	Título
Escola Regional de Alto Desempenho (ERAD)	2016	Alocação de Infraestruturas Virtuais em Redes Definidas por Software: uma Investigação Preliminar
Revista Brasileira de Computação Aplicada (RBCA)	2016	Explorando a elasticidade de nuvens IaaS para reconfigurar dinamicamente aplicações n-camadas
<i>Network of the Future (NoF)</i>	2016	Executing Distributed Applications on SDN-based Data Center: a Study with NAS Parallel Benchmark
<i>Symposium on Computing Systems Engineering (SBESC)</i>	2016	Análise de Aplicações Distribuídas em SDN: um Estudo com o Benchmark NAS
<i>International Symposium on Cluster, Cloud and Grid Computing (CCGrid)</i>	2017	QoS-Aware Virtual Infrastructures Allocation on SDN-based Clouds
Simpósio Brasileiro de Rede de Computadores e Sistemas Distribuídos (SBRC)	2017	Alocação de Infraestruturas Virtuais em <i>data centers</i> implementados com Redes Definidas por <i>Software</i>

Fonte: Elaborado pelo autor, 2017.

REFERÊNCIAS

- AL-FARES, M.; LOUKISSAS, A.; VAHDAT, A. A scalable, commodity data center network architecture. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 38, n. 4, p. 63–74, ago. 2008. ISSN 0146-4833.
- AL-SHABIBI, A. et al. Openvirtex: Make your virtual sdns programmable. In: **Proc. of the HotSDN '14**. [S.l.]: ACM, 2014. p. 25–30. ISBN 978-1-4503-2989-7.
- ALIZADEH, M. et al. Data center tcp (dctcp). **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 41, n. 4, p. –, ago. 2010. ISSN 0146-4833. Disponível em: <<http://dl.acm.org/citation.cfm?id=2043164.1851192>>.
- AMAZON. **Infraestrutura Global da AWS**. 2017. Disponível em: <<https://aws.amazon.com/pt/about-aws/global-infrastructure/>>.
- AMAZON. **Produtos em nuvem**. 2017. Disponível em: <<https://aws.amazon.com/pt/products/>>.
- ANDERSEN, D. G. **Theoretical Approaches To Node Assignment**. 2002.
- ANHALT, F.; KOSLOVSKI, G.; PRIMET, P. V.-B. Specifying and provisioning Virtual Infrastructures with HIPerNET. **Int. J. Netw. Manag.**, John Wiley & Sons, Inc., New York, NY, USA, v. 20, n. 3, p. 129–148, maio 2010. ISSN 1099-1190. Disponível em: <<http://dx.doi.org/10.1002/nem.732>>.
- BENSON, T.; AKELLA, A.; MALTZ, D. A. Network traffic characteristics of data centers in the wild. In: **Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement**. New York, NY, USA: ACM, 2010. (IMC '10), p. 267–280. ISBN 978-1-4503-0483-2. Disponível em: <<http://doi.acm.org/10.1145/1879141.1879175>>.
- CAVALCANTI, G.; OBELHEIRO, R.; KOSLOVSKI, G. Optimal resource allocation for survivable virtual infrastructures. In: **IEEE Conference on Design of Reliable Communication Networks**. [S.l.: s.n.], 2014. p. 1–8.
- CHOWDHURY, M.; RAHMAN, M.; BOUTABA, R. Vineyard: Virtual network embedding algorithms with coordinated node and link mapping. **Networking, IEEE/ACM Transactions on**, v. 20, n. 1, p. 206–219, Feb 2012. ISSN 1063-6692.
- CHOWDHURY, N. M. M. K.; BOUTABA, R. Network virtualization: State of the art and research challenges. **Comm. Mag.**, IEEE Press, Piscataway, NJ, USA, v. 47, n. 7, p. 20–26, jul. 2009. ISSN 0163-6804.
- COSTA, P. et al. NaaS: Network-as-a-service in the Cloud. In: **Proceedings of the 2Nd USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services**. Berkeley, CA, USA: USENIX Association, 2012. (Hot-ICE'12), p. 1–1. Disponível em: <<http://dl.acm.org/citation.cfm?id=2228283.2228285>>.
- DEMIRCI, M.; AMMAR, M. Design and analysis of techniques for mapping virtual networks to software-defined network substrates. **Computer Communications**, v. 45, p. 1 – 10, 2014. ISSN 0140-3664.

DRUTSKOY, D.; KELLER, E.; REXFORD, J. Scalable network virtualization in software-defined networks. **Internet Computing, IEEE**, v. 17, n. 2, p. 20–27, March 2013. ISSN 1089-7801.

FEAMSTER, N.; REXFORD, J.; ZEGURA, E. The road to sdn. **Queue**, ACM, New York, NY, USA, v. 11, n. 12, p. 20:20–20:40, dez. 2013. ISSN 1542-7730. Disponível em: <<http://doi.acm.org/10.1145/2559899.2560327>>.

FISCHER, A. et al. Virtual network embedding: A survey. **Communications Surveys Tutorials, IEEE**, v. 15, n. 4, p. 1888–1906, Fourth 2013. ISSN 1553-877X.

GOOGLE. **Google Cloud Platform**. 2017. Disponível em: <https://cloud.google.com/?utm_source=google&utm_medium=cpc&utm_campaign=2015-q2-cloud-na-gcp-bkws-freetrial-en&ds_lpt_start&ds_lpt_end>.

GOOGLE. **Regions and Zones**. 2017. Disponível em: <<https://cloud.google.com/compute/docs/regions-zones/regions-zones>>.

GUO, C. et al. Bcube: A high performance, server-centric network architecture for modular data centers. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 39, n. 4, p. 63–74, ago. 2009. ISSN 0146-4833. Disponível em: <<http://doi.acm.org/10.1145/1594977.1592577>>.

GUO, C. et al. Secondnet: A data center network virtualization architecture with bandwidth guarantees. In: **Proceedings of the 6th International Conference**. New York, NY, USA: ACM, 2010. (Co-NEXT '10), p. 15:1–15:12. ISBN 978-1-4503-0448-1.

HAIDER, A.; POTTER, R.; NAKAO, A. Challenges in Resource Allocation in Network Virtualization. **ITC Specialist Seminar**, v. 20, May 2009.

HASHEM, I. A. T. et al. The rise of “big data” on cloud computing: Review and open research issues. **Information Systems**, v. 47, p. 98 – 115, 2015. ISSN 0306-4379. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0306437914001288>>.

HUU, T. T. et al. Joint elastic cloud and virtual network framework for application performance-cost optimization. **Journal of Grid Computing**, v. 9, n. 1, p. 27–47, 2011. ISSN 1572-9184.

JARRAY, A.; KARMOUCH, A. Decomposition approaches for virtual network embedding with one-shot node and link mapping. **Networking, IEEE/ACM Transactions on**, v. 23, n. 3, p. 1012–1025, June 2015. ISSN 1063-6692.

JENNINGS, B.; STADLER, R. Resource management in clouds: Survey and research challenges. **Journal of Network and Systems Management**, Springer US, p. 1–53, 2014. ISSN 1064-7570.

KARAKUS, M.; DURRESI, A. Quality of service (qos) in software defined networking (sdn): A survey. **Journal of Network and Computer Applications**, v. 80, p. 200 – 218, 2017. ISSN 1084-8045. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1084804516303186>>.

KNOWLEDGE, D. C. **Report: Google Uses About 900,000 Servers**. 2011. Disponível em: <<http://www.datacenterknowledge.com/archives/2011/08/01/report-google-uses-about-900000-servers/>>.

KNOWLEDGE, D. C. **Ballmer: Microsoft has 1 Million Servers**. 2013. Disponível em: <<http://www.datacenterknowledge.com/archives/2013/07/15/ballmer-microsoft-has-1-million-servers/>>.

KREUTZ, D. et al. Software-Defined Networking: A Comprehensive Survey. **Proceedings of the IEEE**, IEEE, v. 103, n. 1, p. 14–76, jan. 2015. ISSN 0018-9219.

MANVI, S. S.; SHYAM, G. K. Resource management for infrastructure as a service (iaas) in cloud computing: A survey. **Journal of Network and Computer Applications**, v. 41, p. 424 – 440, 2014. ISSN 1084-8045.

MCKEOWN, N. et al. OpenFlow: Enabling Innovation in Campus Networks. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 38, n. 2, p. 69–74, mar. 2008. ISSN 0146-4833. Disponível em: <<http://doi.acm.org/10.1145/1355734.1355746>>.

MELL, P. M.; GRANCE, T. **SP 800-145. The NIST Definition of Cloud Computing**. Gaithersburg, MD, United States, 2011.

MICROSOFT. **Microsoft Global Data Centers**. 2017. Disponível em: <<https://www.microsoft.com/en-us/cloud-platform/global-datacenters>>.

MIJUMBI, R. et al. A path generation approach to embedding of virtual networks. **IEEE Transactions on Network and Service Management**, v. 12, n. 3, p. 334–348, Sept 2015. ISSN 1932-4537.

MIJUMBI, R. et al. Dynamic resource management in sdn-based virtualized networks. In: **Int. Conf. on Network and Service Management**. [S.l.: s.n.], 2014.

MYSORE, R. N. et al. Portland: A scalable fault-tolerant layer 2 data center network fabric. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 39, n. 4, p. 39–50, ago. 2009. ISSN 0146-4833. Disponível em: <<http://doi.acm.org/10.1145/1594977.1592575>>.

NUNES, B. A. A. et al. A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. **IEEE Communications Surveys Tutorials**, v. 16, n. 3, p. 1617–1634, Third 2014. ISSN 1553-877X.

OLIVEIRA, R. d.; KOSLOVSKI, G. A tree-based algorithm for virtual infrastructure allocation with joint virtual machine and network requirements. **International Journal of Network Management**, John Wiley & Sons, Ltd., 2016.

ONF. **OpenFlow Switch Specification**. 2011. Disponível em: <<http://archive.openflow.org/documents/openflow-spec-v1.1.0.pdf>>.

PERSICO, V. et al. Measuring network throughput in the cloud: The case of amazon ec2. **Computer Networks**, v. 93, p. 408 – 422, 2015. ISSN 1389-1286. Cloud Networking and Comm. {II}.

POPA, L. et al. Faircloud: Sharing the network in cloud computing. In: **Proc. of the Workshop on Hot Topics in Networks**. [S.l.]: ACM, 2011. p. 22:1–22:6. ISBN 978-1-4503-1059-8.

RIGGIO, R. et al. Progressive virtual topology embedding in openflow networks. In: **2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)**. [S.l.: s.n.], 2013. p. 1122–1128. ISSN 1573-0077.

ROST, M.; FUERST, C.; SCHMID, S. Beyond the stars: Revisiting virtual cluster embeddings. In: **In Proc. ACM SIGCOMM Computer Communication Review**. [S.l.: s.n.], 2015.

SHERWOOD, R. et al. Can the production network be the testbed? In: **Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation**. [S.l.]: USENIX, 2010. p. 1–6.

SHERWOOD, R. et al. **FlowVisor: A Network Virtualization Layer**. [S.l.], 2009.

STALLINGS, W. **Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud**. 1st. ed. [S.l.]: Addison-Wesley Professional, 2015. ISBN 0134175395.

TAO, F.; JUN, B.; KE, W. Allocation and scheduling of network resource for multiple control applications in sdn. **Communications, China**, v. 12, n. 6, p. 85–95, June 2015. ISSN 1673-5447.

YU, M. et al. Rethinking virtual network embedding: substrate support for path splitting and migration. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 38, n. 2, p. 17–29, 2008. ISSN 0146-4833.