

# Abordagens Descentralizadas para o Mapeamento de Redes Virtuais em Ambientes Multidomínio

Samuel M. A. Araújo<sup>1</sup>, Fernanda S. H. de Souza<sup>2</sup>, Geraldo R. Mateus<sup>1</sup>

<sup>1</sup>Departamento de Ciência da Computação  
Universidade Federal de Minas Gerais (UFMG)  
Belo Horizonte – MG – Brasil

<sup>2</sup>Departamento de Ciência da Computação  
Universidade Federal de São João del-Rei (UFSJ)  
São João del-Rei – MG – Brasil

samuelcco@gmail.com, fsumika@ufs.br, mateus@dcc.ufmg.br

**Abstract.** *Network virtualization has gained visibility by enabling the development of new functionalities and applications for the internet. With virtualization, a logical view of the hardware is created so that multiple virtual networks can coexist on the same physical network substrate. The virtual network embedding problem belongs to the NP-hard class not allowing the use of exact approaches. In this work, the application of a fast heuristic approach in a multidomain, decentralized and online environment is proposed, through an orchestrator's control. A comparative study regarding the implementation of three different orchestration models is presented: total knowledge, partial and without knowledge of the internal physical network infrastructure of the domains. Experimental results show the performance of the orchestrator along with the heuristic, reporting energy consumption, profit generation and message exchange metrics for the different approaches.*

**Resumo.** *Virtualização de redes tem ganhado visibilidade por permitir a implementação de novas funcionalidades e aplicações para a internet. Com a virtualização, cria-se uma visão lógica do hardware de modo que múltiplas redes virtuais possam coexistir no mesmo substrato físico de rede. O mapeamento de redes virtuais é um problema pertencente à classe NP-difícil não permitindo a utilização de abordagens exatas. Neste trabalho é proposta a aplicação de uma abordagem heurística de rápida execução em um ambiente multidomínio, descentralizado e online, através do controle de um orquestrador. Um estudo comparativo da implementação de três modelos de orquestração distintos é apresentado: conhecimento total, parcial e sem conhecimento da infraestrutura de rede física interna aos domínios. Resultados experimentais mostram o desempenho do orquestrador associado à heurística, evidenciando para as diferentes abordagens, métricas de consumo de energia, geração de lucro e troca de mensagens.*

## 1. Introdução

Dentre outras concepções, o crescimento da internet é impulsionado por uma nova gama de aplicações e dados portáteis que podem ser acessados a partir de qualquer lugar. Outro aspecto importante nesse crescimento é a consolidação da chamada internet das coisas (*Internet of Things* - IoT), que busca conectar aparelhos eletrônicos do dia a dia à internet e trazer mais dinâmica e integração ao cotidiano das pessoas [Bojanova and Voas 2017].

Este crescimento da internet esbarra em alguns fatores inerentes ao fato da internet ser desenvolvida sobre protocolos rígidos e limitados; problema conhecido como “ossificação da internet” [Zhu and Ammar 2006]. Uma alternativa para o problema de ossificação da internet é a aplicação de mecanismos de virtualização de redes. A virtualização de redes permite a implementação de novas funcionalidades ao se criar uma visão lógica do *hardware*, de forma que múltiplas redes virtuais (VNs) com características particulares possam compartilhar o mesmo substrato de rede físico (SN) [Chowdhury et al. 2009]. Nesse contexto, emerge o problema denominado Mapeamento de Redes Virtuais (*Virtual Network Embedding* - VNE). O VNE consiste em determinar o conjunto de componentes do SN que irão hospedar uma dada VN. Este problema pertence à classe NP-difícil [Zhu and Ammar 2006].

A virtualização de redes abre um novo leque para o desenvolvimento das mais diversas aplicações para a internet, viabilizando ao desenvolvedor uma liberdade de criar/implementar funcionalidades inexistentes nos protocolos TPC/IP originais [Anderson et al. 2005]. Como exemplo de aplicações que fazem uso da virtualização de redes podemos citar o provimento de *streaming*, o compartilhamento de dados *peer-to-peer*, telefonia sobre IP etc. Um exemplo de aplicação da computação em nuvens junto à virtualização de redes é a possibilidade do provimento *online* de sistemas computacionais completos para empresas. Nesse caso as empresas não necessitam mais investir em *mainframes*, por ser possível contratar servidores de dados distribuídos acessíveis de qualquer lugar no mundo. De fácil escalabilidade, tais serviços aprovencionam um ambiente computacional completo por meio de emulação. Assim, a pesquisa e desenvolvimento de novas soluções para o VNE atraem a atenção de empresas de tecnologia e serviços sobre a internet.

O VNE envolve diferentes entidades com ações distintas: os provedores de infraestrutura (ISPs), responsáveis por gerir os componentes dos domínios; os provedores de serviço (SPs), responsáveis por instanciar e gerir as VNs, além de contratar os serviços dos ISPs; e os usuários finais, que consomem os serviços prestados pelos SPs [Chowdhury et al. 2009]. Neste trabalho, um domínio de rede é definido como uma entidade que possui um conjunto de equipamentos de infraestrutura de redes, como *switches*, roteadores, fibra ótica etc. Cada ISP envolvido pode gerir um ou mais domínios de rede e cada domínio pode possuir políticas autônomas de gerência de recursos.

Em um cenário de mapeamento *online*, as características e recursos demandados por uma VN somente são conhecidos no momento de sua chegada ao servidor. Isto dificulta a atuação do algoritmo de mapeamento por este desconhecer as demandas futuras. Outra questão relativa ao VNE consiste na composição do SN, que pode ser formado por um único domínio ou pela união de múltiplos domínios (MD). Em ambientes MD, o mapeamento de redes virtuais de forma descentralizada se torna estratégico, pois cada domínio pode aplicar um algoritmo de mapeamento próprio de forma independente e par-

alela, buscando otimizar seus objetivos. Assim, cada domínio possui autonomia total de gerenciamento e otimização local.

Diversos fatores mostram-se importantes em ambientes MD, tais como: (i) a troca de mensagens, que se feita de forma descomedida pode gerar um *overhead*<sup>1</sup> de comunicação, causar gargalos no SN e ser prejudicial em futuros mapeamentos; (ii) o lucro gerado pelos mapeamentos, que está diretamente relacionado com a receita obtida; (iii) a energia consumida pelos mapeamentos, que também impacta no lucro dos domínios. Além disso, em ambientes *online* e MD há necessidade de um mapeamento rápido. Neste caso, abordagens exatas e/ou baseadas em programação linear tendem a serem lentas devido à complexidade do problema e à grande quantidade de componentes presentes em cada domínio.

A fim de tratar o VNE considerando os cenários descritos anteriormente, são propostas neste trabalho abordagens descentralizadas e *online* em ambientes multidomínio. Para garantir um mapeamento rápido, empregou-se uma adaptação da heurística proposta em [Araújo et al. 2015], com um tempo máximo de execução de 300ms (milissegundos) por mapeamento de VN.

O restante deste artigo está organizado como segue. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 apresenta a definição formal do problema. Na Seção 4 são apresentadas as soluções propostas, seguida da Seção 5 que introduz as métricas avaliadas. Na seção 6 são apresentados os experimentos realizados e por fim na Seção 7 o artigo é concluído.

## 2. Trabalhos Relacionados

As abordagens para o VNE apresentadas nesta seção podem ser classificadas quanto a estáticas ou dinâmicas, centralizadas ou distribuídas, concisas ou redundantes, exatas ou heurísticas e *online* ou *offline*, seguindo a taxonomia proposta por [Fischer et al. 2013].

Em uma abordagem dinâmica, permite-se a alteração do mapeamento de uma VN sobre o SN, sendo também chamada de reconfiguração. [Eramo et al. 2016] demonstram que a aplicação de reconfiguração traz vantagens em termos de melhorias na taxa de aceitação, fragmentação do SN e no lucro. No entanto, deve-se atentar para o fato que em ambientes reais a ação de suspender os mapeamentos, remover as conexões ativas e reconfigurá-las é muito custosa, e pode causar problemas na qualidade do serviço prestado. O trabalho aqui proposto adota a linha estática, onde reconfigurações não são permitidas, similar a [Araújo et al. 2015, Nguyen et al. 2017].

Abordagens centralizadas possuem uma visão total do SN, permitindo uma otimização global do mapeamento, como adotado em [Alkmim et al. 2011, Araújo et al. 2015, Nguyen et al. 2017]. Apesar de muito difundido, tal modelo tende a ser não aplicável na prática. Em ambientes reais, centralizar todas as ações sobre uma única entidade pode afetar a escalabilidade, segurança e acarretar em uma fila de requisições à espera de atendimento. Em compensação, a qualidade dos mapeamentos centralizados tende a ser superior em relação a mapeamentos descentralizados

---

<sup>1</sup>Em virtualização, *overhead* de comunicação refere-se a um consumo extra de banda em um enlace físico, gerado pela troca de mensagens necessária entre domínios diferentes para se prover um mapeamento.

[Fischer et al. 2013]. Neste trabalho, o modelo centralizado é usado como base de comparação com as abordagens descentralizadas.

Em [Samuel et al. 2013], um *framework* para o mapeamento de VNs descentralizado é apresentado. A abordagem efetua o particionamento de cada VN em  $k$  sub-VNs a serem mapeadas em um subconjunto de domínios conhecidos. Neste caso, cada domínio é livre para aplicar seu próprio algoritmo de mapeamento. Para tanto, o próprio SP deve conhecer o(s) domínio(s) existentes para requisitar o mapeamento. Vale ressaltar que caso o SP conheça apenas um domínio de rede, pode haver um tipo de monopólio onde somente o domínio conhecido será favorecido. O trabalho proposto se distingue de [Samuel et al. 2013] por manter a unicidade de uma VN, onde fica à cargo de cada domínio uma análise da viabilidade do mapeamento completo, da necessidade de colaboração com domínios adjacentes ou de uma possível rejeição. Além disso, variações relativas a quantos/quais domínios recebem a requisição são consideradas.

De acordo com [Chiaraviglio et al. 2012] dois aspectos estão fortemente relacionados ao lucro de um ISP: o número de clientes ativos, e o consumo de energia gasto para manter uma VN operante. Nesta visão, para obter um lucro alto, é necessário que os ISPs efetuem os mapeamentos de rede a um custo baixo. Assim, os componentes virtuais são mapeados sempre com uso do menor número de recursos físicos disponíveis, viabilizando o desligamento de recursos não utilizados, o que gera a economia de energia. De forma similar, o trabalho aqui proposto visa maximizar o lucro dos domínios através da otimização do consumo de energia elétrica. Ao consumir menos energia para prover o mapeamento, os domínios tendem a lucrar mais. Com esse princípio, cria-se uma concorrência entre os domínios existentes, onde os domínios que possuem os equipamentos de menor consumo vão lucrar mais e potencialmente atrair mais mapeamentos.

Abordagens redundantes tratam o VNE com características de resiliência à falhas. [Oliveira et al. 2015] apresentam uma heurística de mapeamento com duas estratégias: proativa e reativa. O mapeamento proativo é feito com a metaheurística *simulated annealing*, mapeando cada enlace virtual em mais de um caminho físico disjunto. A estratégia reativa atua em caso de ataques e quedas de recursos, com a migração dos mapeamentos originais para os componentes redundantes através de uma estratégia de fluxo máximo. Garantias de resiliência mostram-se importantes; mas em contrapartida, tais reservas levam a recursos ociosos, podendo afetar a taxa de aceitação e o consumo de energia. Diferentemente de [Oliveira et al. 2015], a abordagem proposta neste artigo foca em minimizar o consumo energético com uso conciso de recursos.

Os trabalhos classificam-se ainda como *online* ou *offline*, no que diz respeito à chegada e atendimento das VNs. No cenário *online*, as características de uma VN não são conhecidas antes de sua chegada ao provedor, melhor caracterizando um ambiente real [Chowdhury et al. 2009, Araújo et al. 2015, Nguyen et al. 2017]. No cenário *offline*, o conjunto de VNs é conhecido por completo *a priori*. [Alkmim et al. 2011] abordam o VNE *offline* através de Programação Inteira Binária, onde é aplicada uma abordagem exata e outra relaxada. No trabalho de [Alkmim et al. 2011], o objetivo é minimizar a largura de banda total usada e consequentemente maximizar a quantidade de recursos residuais para futuros mapeamentos. Os autores concluem que mesmo a abordagem relaxada, em instâncias grandes, pode levar um tempo consideravelmente alto para efetuar um mapeamento. Desta forma, a abordagem aqui proposta adota um mapeamento heurís-

tico para garantir um baixo tempo de resposta em um cenário *online*.

O VNE ainda pode requerer o uso de funções específicas, que em uma arquitetura de redes tradicional não podem ser virtualizadas, como bancos de dados, *proxy*, *honeypots* etc. Essa variação do problema é chamada na literatura de virtualização de funções de rede (NFV). [Dalla-Costa et al. 2017] apresentam em seu trabalho uma avaliação detalhada de um orquestrador para NFV em ambientes de redes sem fio chamado Maestro. Em seu trabalho é colocado que uma das funções do orquestrador é posicionar os componentes a serem virtualizados de forma a melhor atender as restrições do problema. Neste trabalho, o orquestrador assume um papel semelhante ao de [Dalla-Costa et al. 2017], onde dentre outras tarefas, ele define os melhores conjuntos de componentes físicos, em termos de gasto energético, a receberem a rede a ser mapeada.

### 3. Definição do Problema

O VNE em MD pode ser definido através de um grafo não direcionado  $G = (N, E)$  representando o SN e demais parâmetros a seguir. Seja  $N = \{N^1 \cup N^2 \dots \cup N^{|D|}\}$ , a união dos subconjuntos de roteadores físicos, onde cada  $N^d \in N$  é um conjunto de roteadores físicos pertencentes ao domínio  $d \in D$  e  $D = \{1, 2, 3, \dots, |D|\}$  o conjunto de domínios existentes. Similarmente  $E = \{E^1 \cup E^2 \dots \cup E^{|D|}\}$  representa a união dos subconjuntos de enlaces físicos, e cada  $E^d \in E$  é um conjunto de enlaces físicos internos ao domínio  $d \in D$  (intra-D). O conjunto  $E^0$  representa os enlaces inter-domínio (inter-D), que são aqueles que fazem fronteira e possuem gerência acordada entre diferentes domínios. Cada nó  $i \in N$  possui uma capacidade máxima  $C_i$  de CPU e um posicionamento geográfico  $(x_i, y_i)$  fixo. Cada enlace  $(i, j) \in E \cup E^0$  possui uma capacidade de banda máxima  $B_{ij}$ . Um exemplo da representação do grafo  $G = (N, E)$  é mostrado na Figura 1.

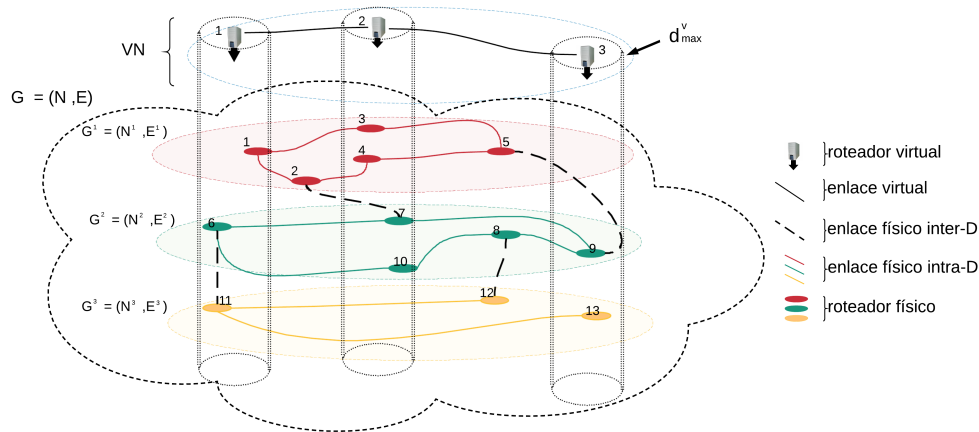


Figura 1. Exemplo da modelagem do SN com três domínios e uma VN.

Seja  $V$  o conjunto de todas as VNs. Cada requisição  $v \in V$  é representada como um grafo não direcionado  $G^v = (N^v, E^v)$ , onde  $N^v$  e  $E^v$  representam os conjuntos de roteadores e enlaces virtuais, respectivamente. Cada nó virtual  $k \in N^v$  possui uma demanda de CPU diferente  $c_k^v$  e uma posição geográfica  $(x_k, y_k)$  que é utilizada para definir uma região de mapeamento com base em um raio de localização  $d_{max}^v$ . Para cada nó virtual  $k \in N^v$  a viabilidade de posicionamento geográfico sobre um nó físico  $i \in N$  é calculada com uso de uma função euclidiana  $d((x_i, y_i), (x_k, y_k))$ . Se

$d((x_i, y_i), (x_k, y_k)) \leq d_{max}^v$ , o roteador físico  $i$  está apto geograficamente a hospedar o nó virtual  $k$ . Cada enlace virtual  $(k, l) \in E^v$  possui uma demanda de banda  $b_{kl}^v$ . Um exemplo de uma VN composta por 3 roteadores físicos pode ser visto na Figura 1.

Uma solução para o VNE em MD consiste em determinar com sucesso o mapeamento  $f : G^v \rightarrow G$ . Assim, o mapeamento de nós é dado por:  $\forall k \in N^v, \exists i \in N : c_k^v \leq \overline{C}_i$ , onde  $\overline{C}_i$  representa a CPU residual do nó físico  $i \in N$ . O mapeamento de enlaces é dado por:  $\forall (k, l) \in E^v, \exists \{(i, j)^1, \dots, (i, j)^m\} : b_{kl}^v \leq \overline{B}_{ij}^1, \dots, b_{kl}^v \leq \overline{B}_{ij}^m$ , onde  $\overline{B}_{ij}$  representa a largura de banda residual dos enlaces  $(i, j) \in E \cup E^0$  que compõem o caminho físico que irá hospedar o enlace virtual. Caso as duas restrições sejam atendidas com sucesso, a VN é aceita; caso contrário é rejeitada.

#### 4. Soluções para o VNE Multidomínio

O algoritmo de mapeamento aplicado neste trabalho é uma adaptação da abordagem proposta por [Araújo et al. 2015], modificada para prover soluções de mapeamentos mais rápidos e sem resiliência. Neste trabalho, o algoritmo GRASP [Feo and Resende 1995] é usado como um filtro que gera um conjunto de soluções durante  $t_1ms$ , seguido pela aplicação de uma busca local na melhor solução gerada, através do algoritmo RVNS [Mladenović and Hansen 1997] por mais  $t_2ms$ . Caso nenhuma solução viável seja encontrada a VN é rejeitada. Tal adaptação deixa o algoritmo mais rápido, o que se adequa melhor em ambientes *online*.

##### 4.1. Abordagem centralizada (VE<sub>ce</sub>)

Neste modelo, o orquestrador faz uma consulta inicial a cada domínio existente e recebe os dados de toda topologia pertencente ao SN (Tabela 1, mensagens 10 e 11). A resposta inclui informações de localização de todos os roteadores físicos pertencentes aos domínios, seus respectivos enlaces, capacidades e consumo de energia. Em cenários reais de redes tradicionais, cada domínio tende a esconder o máximo de informações sobre sua topologia interna; logo, esta abordagem é aplicada apenas com intuito de comparação de resultados.

Nesta abordagem, o orquestrador possui autonomia global para a otimização, onde o algoritmo GRASPF-RVNS é aplicado em todo SN de maneira coordenada e centralizada. Como o orquestrador conhece o estado *a priori* de todo o SN, as trocas de mensagens são geradas somente para a efetivação de um mapeamento (Tabela 1, mensagem 7). O orquestrador é quem define o mapeamento, neste caso, o próprio atualiza internamente o estado *a posteriori* do SN sem a necessidade de demais consultas, como apresentado na Figura 2.

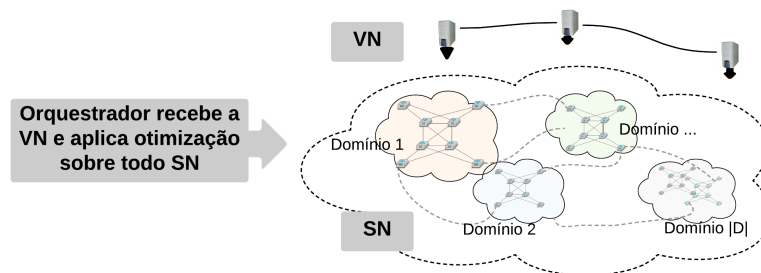


Figura 2. Orquestrador com conhecimento total do SN.

#### 4.2. Abordagem descentralizada, com conhecimento parcial do SN (VE\_d1)

Este modelo se caracteriza pelo compartilhamento do nível de saturação dos domínios existentes com o orquestrador (Tabela 1, mensagem 9). Deste modo, o orquestrador possui a informação de qual domínio possui as capacidades mais próximas do esgotamento e quais estão mais aptos a efetivar um mapeamento. Tal nível de saturação é mensurado pela equação  $h(d) = \frac{\sum_{i \in N^d} \bar{C}_i \cdot \sum_{(i,j) \in E^d \cup E^0} \bar{B}_{ij}}{|N^d|}$ . Este modelo é ilustrado na Figura 3.

Após receber o nível de saturação de cada domínio, o orquestrador fica responsável por repassar o mapeamento da VN para o domínio que possui menor nível de saturação de seus componentes (Tabela 1, mensagem 7). Caso o domínio envolvido não tenha capacidade para mapear algum componente requerido pela VN, ele pode se comunicar com os domínios adjacentes através da comunicação inter-D para requerer o mapeamento do componente faltante (Tabela 1, mensagens 4, 5 e 6). Somente se o domínio escolhido responder pela completa rejeição do mapeamento (Tabela 1, mensagem 3), o orquestrador tenta realizar o mapeamento com o segundo domínio com menor nível de saturação. Assim ocorre sucessivamente até a VN ser mapeada ou até que não haja mais domínios a serem consultados. Ao final, caso algum domínio tenha recebido o mapeamento de algum componente, este deve atualizar junto ao orquestrador seu novo nível de saturação (Tabela 1, mensagem 9). Neste caso a coordenação das informações para o mapeamento é feita pelo orquestrador, e o mapeamento é feito no domínio. Cada domínio aplica a otimização dentro de suas fronteiras, de maneira local.

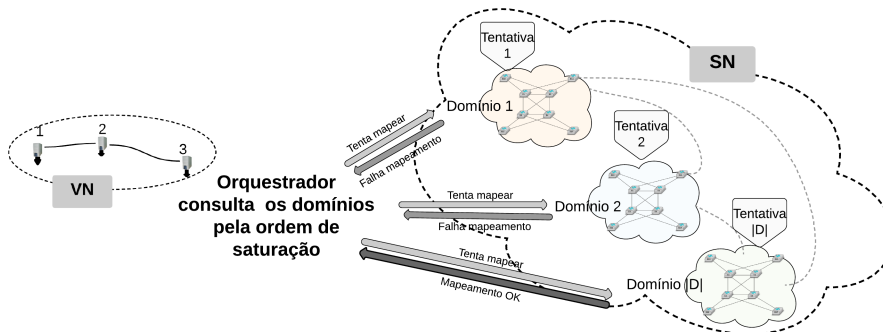


Figura 3. Orquestrador com conhecimento parcial do SN.

#### 4.3. Abordagem descentralizada, sem conhecimento do SN (VE\_d2)

Diferentemente dos modelos anteriores, neste caso o orquestrador não possui nenhuma informação interna aos domínios envolvidos. É de responsabilidade do orquestrador consultar cada domínio, se o mesmo consegue prover o mapeamento, e qual é o custo envolvido (Tabela 1, mensagens 1, 2 e 3). Ao final contrata-se o serviço do domínio mais vantajoso (Tabela 1, mensagens 7 e 8).

Deste modo cada domínio é responsável por mapear e otimizar localmente a VN que lhe foi requerida. Caso por alguma inviabilidade local, algum componente da VN não possa ser mapeado, o domínio pode requisitar a um domínio adjacente que mapeie o componente faltante (Tabela 1, mensagens 4, 5 e 6). Nesse caso, há uma troca de mensagens entre os dois domínios envolvidos. Tal modelo é representado na Figura 4.

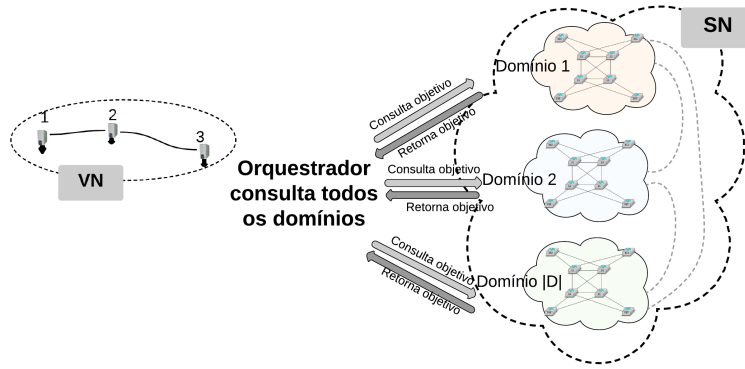


Figura 4. Orquestrador sem conhecimento do SN.

#### 4.4. Mensagens de Coordenação

Sempre que o orquestrador se comunica com algum domínio ou algum domínio se comunica com outro domínio, é gerada uma mensagem de pergunta, resposta ou ordem. Tais mensagens são exemplificadas na Tabela 1.

Tabela 1. Tabela de mensagens

ID	Tipo	Emissor	Receptor	Conteúdo	Abordagem
1	Pergunta	orquestrador	domínio $d$	VN	$VE_{d2}$
2	Resposta	domínio $d$	orquestrador	OK, custo	$VE_{d2}$
3	Resposta	domínio $d$	orquestrador	falha	$VE_{d1}, VE_{d2}$
4	Pergunta	domínio $d'$	domínio $d''$	roteador $u$ , roteador $v$	$VE_{d1}, VE_{d2}$
5	Resposta	domínio $d''$	domínio $d'$	OK, caminho, custo	$VE_{d1}, VE_{d2}$
6	Resposta	domínio $d''$	domínio $d'$	falha	$VE_{d1}, VE_{d2}$
7	Ordem	orquestrador	domínio $d$	OK, VN	$VE_{d1}, VE_{d2}, VE_{ce}$
8	Ordem	domínio $d'$	domínio $d''$	OK, $u, v$ , caminho	$VE_{d1}, VE_{d2}$
9	Resposta	domínio $d'$	orquestrador	$h(d')$	$VE_{d1}$
10	Pergunta	orquestrador	domínio $d'$	$G^d(null, null)$	$VE_{ce}$
11	Resposta	domínio $d'$	orquestrador	$G^d(N^d, E^d)$	$VE_{ce}$

1. Mensagem enviada do orquestrador para o domínio  $d \in D$ , onde é requisitado o custo para atender uma VN.
2. Mensagem enviada do domínio  $d$  ao orquestrador, onde é devolvido o custo envolvido de atender à VN com sucesso.
3. Mensagem enviada do domínio  $d$  ao orquestrador, onde é reportado que não consegue atender à VN.
4. Mensagem enviada do domínio  $d'$  para o domínio adjacente  $d''$ , onde é requerido um roteador físico que atenda às restrições do roteador virtual  $u$  e que possua um caminho com banda mínima  $b_{uv}$  até o roteador de fronteira  $v$ .
5. Mensagem enviada do domínio  $d''$  para o domínio  $d'$ , onde é respondida com sucesso a solicitação de busca de roteador físico junto a rota estabelecida entre origem de fronteira  $v$  e roteador físico  $u$  e o respectivo valor envolvido.
6. Mensagem enviada do domínio  $d''$  para o domínio  $d'$ , onde é respondida a falha no atendimento do roteador físico requisitado.
7. Mensagem enviada do orquestrador para o domínio  $d \in D$ , onde é requisitado que o mesmo efetive o mapeamento definido *a priori*.



8. Mensagem enviada do domínio  $d'$  para o domínio  $d''$ , para que o mesmo efetive o mapeamento do componente  $u$ , no caminho previamente consultado.
9. Mensagem enviada do domínio  $d \in D$  para o orquestrador, onde é reportada a saturação interna dos componentes do domínio  $d$ .
10. Mensagem enviada do orquestrador para o domínio  $d \in D$ , onde é consultada a estrutura topológica interna ao domínio  $d$ .
11. Mensagem enviada do domínio  $d \in D$  para o orquestrador, onde é reportada a estrutura topológica do domínio  $d$ .

## 5. Métricas de Avaliação

De maneira a comparar as abordagens propostas, um conjunto de métricas é definido. Para tal, sejam as notações:

- $\Gamma^i$ , consumo de energia em  $MW$  de um roteador físico em *stand-by*.
- $\gamma^i$ , consumo de energia em  $MW$  de um roteador físico em funcionamento, calculado por  $Mbps$  de CPU processado.
- $\Delta^{ij}$ , consumo de energia em  $MW$  de um enlace físico intra-D em *stand-by*, calculado por  $Km$ .
- $\delta^{ij}$ , consumo de energia em  $MW$  de um enlace físico intra-D em funcionamento, calculado por  $1Mbps$  de banda consumida por  $Km$ .
- $\Omega^{ij}$ , consumo de energia em  $MW$  de um enlace físico inter-D em *stand-by*, calculado por  $Km$ .
- $\omega^{ij}$ , consumo de energia em  $MW$  de um enlace físico inter-D em funcionamento, calculado por  $1Mbps$  de banda consumida por  $Km$ .
- $\kappa$ , receita recebida pelo ISP pelo mapeamento de um roteador virtual, calculado por  $Mbps$  de CPU processado.
- $\varphi$ , receita recebida pelo ISP pelo mapeamento de um enlace virtual, cobrado por banda consumida a cada  $Km$  de distância entre os roteadores virtuais.
- $\chi$ , custo de energia elétrica a cada  $MW$  consumido.

A taxa de aceitação (TA) é dada pelo número total de VNs aceitas que chegaram dentro de um determinado intervalo de tempo  $t$ . Seja  $V^t$  o conjunto de VNs que chegaram dentro do intervalo de tempo  $t$  e  $y^v$  a variável binária que indica se a requisição  $v$  foi atendida ou não. Tem-se que:

$$TA = \sum_{v \in V^t} y^v \quad (1)$$

A receita cobrada pelos ISPs para o mapeamento de determinada VN  $v \in V$ , é dada em função da quantidade de recursos virtuais demandados pela própria VN durante todo seu tempo de vida ( $t^v$ ), expressa por:

$$R^v = t^v \cdot \left( \sum_{k \in N^v} c_k^v \cdot \kappa + \sum_{(k,l) \in E^v} b_{kl}^v \cdot \varphi \cdot d((x_k, y_k), (x_l, y_l)) \right) \quad (2)$$

O custo de energia originado pelo mapeamento de determinada VN  $v \in V$  ativa sobre o SN, é dado pelo gasto dos componentes físicos em *stand-by* acrescido da quantidade de recursos utilizados. Desta forma,  $C1^v$  representa o consumo de energia de todos os roteadores;  $C2^v$  o consumo de energia de todos enlaces intra-D; e  $C3^v$  o consumo de energia de todos enlaces inter-D, usados pela VN  $v \in V$ . Seja  $z_{ki}^v$  a variável que indica se

o nó virtual  $k$  da VN  $v$  foi mapeado no nó físico  $i$ , e  $x_{ki}^{vkl}$  a variável que indica se o enlace virtual  $kl$  da VN  $v$  foi mapeado sobre o enlace físico  $ij$ , tem-se que:

$$C1^v = t^v \cdot \left( \sum_{i \in N} \Gamma^i \cdot \sum_{k \in N^v} \gamma^i c_k^v z_{ki}^v \right) \quad (3)$$

$$C2^v = t^v \cdot \left( \sum_{(i,j) \in E} \Delta^{ij} \cdot \sum_{(k,l) \in E^v} \delta^{ij} \cdot b_{kl}^v \cdot x_{ij}^{vkl} \cdot d((x_i, y_i), (x_j, y_j)) \right) \quad (4)$$

$$C3^v = t^v \cdot \left( \sum_{(i,j) \in E^0} \Omega^{ij} \cdot \sum_{(k,l) \in E^v} \omega^{ij} \cdot b_{kl}^v \cdot x_{ij}^{vkl} \cdot d((x_i, y_i), (x_j, y_j)) \right) \quad (5)$$

Por fim, a função objetivo visa maximizar o lucro dos domínios. Formulada com base na diferença entre a receita (equação 2) e o custo financeiro de energia (equações 3, 4 e 5), é expressa por:

$$\text{Lucro} = \text{Max} \sum_{v \in V} (R^v \cdot y^v - (C1^v + C2^v + C3^v) \cdot \chi) \quad (6)$$

## 6. Experimentos Computacionais

Nesta seção são mostrados os resultados computacionais obtidos através das abordagens propostas, bem como os cenários investigados. Os testes foram realizados em um computador Core 2 Duo E8600 3.33GHz, 1333MHz, 16GB de RAM DDR3, e com uso do sistema operacional Ubuntu 14.04.2 LTS. As abordagens são não determinísticas e foram testadas 33 vezes, onde os dados são mostrados com um intervalo de confiança de 95%.

### 6.1. Cenários de simulação

A geração de cada domínio que compõe o SN foi feita com base no modelo de [Waxman 1988]. A probabilidade de existir comunicação entre cada par de domínios é 0.3. Caso exista, a comunicação é feita por enlaces aleatórios atribuídos com uso de uma distribuição uniforme entre 2 e 6. As simulações consideram 60.000 unidades de tempo ( $t$ ) nas operações simuladas. As requisições são geradas com base nos conceitos de [Fischer and de Meer 2016], onde garante-se que as VNs se mapeadas em uma posição específica são 100% mapeáveis, com consumo de 100% dos recursos disponíveis. A partir da geração dos conjuntos de requisições, são formulados dois cenários de teste em função da carga requisitada:  $S1$  de menor densidade, com 250 VNs a chegar e sair ao longo das 60.000 unidades de tempo; e  $S2$  mais denso, com 2000 VNs, no mesmo intervalo de tempo. Os parâmetros usados nas gerações dos cenários foram extraídos de [Chowdhury et al. 2009] e [Fischer et al. 2013], sendo mostrados nas Tabelas 2 e 3.

Existem diversos fatores envolvidos no consumo de energia elétrica de cada componente pertencente ao SN, o que implica em uma variação dos custos de cada domínio, como: resfriamento de componentes, instalação de módulos, amplificadores, número de portas, tipo de cabeamento, roteadores etc. Abstraindo essa grande variação de custos, foi considerado um modelo simplificado de consumo de energia, com valores embasados em equipamentos de mercado. As tarifas aplicadas pelos ISPs também são baseadas em valores reais. O custo de energia aplicado foi de R\$0.54 /kWh e os valores cobrados para o roteamento são  $\kappa = \text{R\$ } 0.012$  e  $\varphi = \text{R\$ } 0.011$ . Os demais parâmetros adotados nos componentes intra-D e inter-D são mostrados nas Tabelas 4 e 5.

**Tabela 2. Parâmetros para geração do SN**

Parâmetro	SN
Número de domínios	3
Número de nós (roteadores)	25 por domínio
Grid coordenadas	50 × 50
Localização nós ( $x, y$ )	randômica
$\alpha$ (parâmetro Waxman)	0.5
$\beta$ (parâmetro Waxman)	0.2
CPU	Uniformemente distribuído entre 50 e 100
Banda	Uniformemente distribuído entre 50 e 100

**Tabela 3. Parâmetros dos conjuntos de VNs**

Parâmetro	VNs
Número de requisições	250 (S1) / 2.000 (S2)
Número de nós de cada requisição	2 – 12
Grid coordenadas	50 × 50
Raio	distribuição Poisson com $\mu = 8$
Taxa de chegada	distribuição Poisson com $\mu = 200t$ (S1) / $\mu = 25t$ (S2)
Tempo de vida	distribuição Exponencial $\mu = 10.000$ (S1) / $\mu = 1.000t$ (S2)
CPU	Uniformemente distribuído entre 0.001 e 20
Banda	Uniformemente distribuído entre 0.001 e 50

**Tabela 4. Parâmetros inter-D**

$G_d^p$	$\Gamma^d$	$\gamma^d$	$\Delta^d$	$\delta^d$
$G_1^p$	0.50	0.043	0.03	0.0030
$G_2^p$	0.90	0.044	0.04	0.0045
$G_3^p$	0.53	0.056	0.05	0.0050

**Tabela 5. Parâmetros intra-D**

$E_{de}^p$	$\Omega^{de}$	$\omega^{de}$
$E_{1,2}^p$	0.45	0.045
$E_{1,3}^p$	0.50	0.065
$E_{2,3}^p$	0.30	0.021

## 6.2. Resultados

Nesta Seção são mostrados os resultados computacionais obtidos através das simulações realizadas. Foram avaliadas três abordagens de orquestração:

- *VE\_ce* - Orquestração com conhecimento total do SN (Seção 4.1). Tempo de execução total  $t_1 = 60ms$  e  $t_2 = 240ms$  por VN.
- *VE\_d1* - Orquestração com conhecimento parcial do SN (Seção 4.2). Como os testes são realizados em 3 domínios, os tempos das fases de execução são fracionados em  $100ms$  por domínio, sendo  $t_1 = 20ms$  e  $t_2 = 80ms$ .
- *VE\_d2* - Orquestração sem conhecimento do SN (Seção 4.3). Como os testes são realizados em 3 domínios, os tempos das fases de execução são fracionados em  $100ms$  por domínio, sendo  $t_1 = 20ms$  e  $t_2 = 80ms$ .

Os experimentos mostrados na Figura 5 abordam a taxa de aceitação final (Figuras 5(a) e 5(b)) e o lucro acumulado final (Figuras 5(c) e 5(d)). Percebe-se que em ambos cenários, de baixa densidade (S1) e de alta densidade (S2), a aceitação da abordagem *VE\_ce* apresenta-se ligeiramente superior à *VE\_d1* e *VE\_d2*. Este fato é esperado, por *VE\_ce* aplicar a otimização globalmente com foco em maximizar o lucro, que depende de uma aceitação final alta. Ao se observar *VE\_d1* e *VE\_d2*, percebe-se que *VE\_d2* possui uma aceitação final ligeiramente mais alta do que *VE\_d1*. Tal fato ocorre por *VE\_d2* consultar todos os domínios e escolher o que lucra mais, e que conseqüentemente gasta menos energia. Um gasto reduzido de energia implica o menor uso de recursos físicos, o que afeta diretamente no residual dos componentes e na disponibilidade de recursos para mapeamentos futuros.

Em relação ao lucro final, apresentado nas Figuras 5(c) e 5(d), *VE\_d2* lucra mais do que *VE\_d1* em ambos cenários, o que reflete as consultas que o orquestrador faz a fim

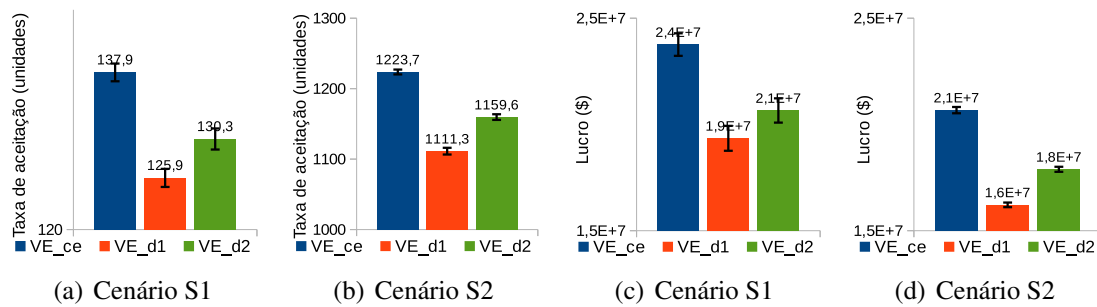


Figura 5. Taxa de aceitação final (a,b) e lucro acumulado final (c,d).

de contratar os serviços do domínio de maior lucro (menor gasto de energia). Além disso, percebe-se que o cenário S2 gera um lucro menor que o cenário S1. Este fato acontece pelas VNs de S1 possuírem um tempo de vida maior, o que gera uma taxa de lucro mais constante. No cenário S2, as requisições possuem um tempo de vida menor e uma taxa de rotatividade maior, logo, as entradas e saídas geram baixas de mapeamento no SN por determinados períodos de tempo, o que afeta o lucro (Figura 6(c) e 6(d)).

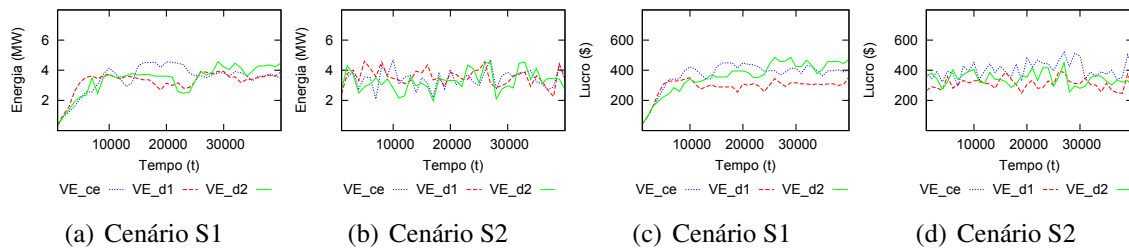


Figura 6. Recortes de tempo da energia consumida (a,b) e lucro gerado (c,d).

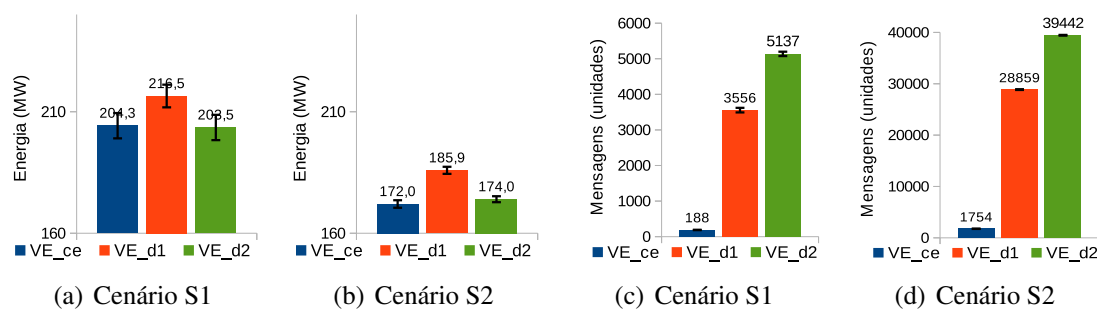


Figura 7. Total de energia consumida (a,b) e total de mensagens trocadas (c,d).

Se comparamos as Figuras 6(a) com 6(c) e 6(b) com 6(d), percebemos que o consumo de energia tem um comportamento similar ao lucro dos domínios, onde no cenário S2 ambos tendem a oscilar muito devido à alta taxa de entrada e saída de VNs. É possível notar ainda que, em ambos os cenários, *VE\_d1* tende normalmente a gastar mais energia e a lucrar menos que as outras abordagens. Este fato ocorre por esta abordagem se preocupar com a saturação dos enlaces (equação de saturação, Seção 4.2) e baixa comunicação entre domínios (Figuras 7(c) e 7(d)) em detrimento de minimizar o consumo de energia

(Figuras 7(a) e 7(b)).

Mesmo que uma VN seja rejeitada, as mensagens trocadas durante a tentativa de seu mapeamento são computadas. A diferença da quantidade de mensagens trocadas (Figuras 7(c) e 7(d)) é fruto do fato de S1 possuir VNs com tempo de vida maior e com pouca rotatividade; ao passo que em S2 existe um número grande de VNs entrantes e saíntes com um tempo de vida menor, o que eleva a quantidade de mensagens trocadas.

Neste trabalho, o *overhead* causado pela comunicação inter-D foi aferido apenas quanto ao número de mensagens enviadas/recebidas, mostradas nas Figuras 7(c) e 7(d). Na prática, em redes cabeadas com milhares de roteadores e enlaces, essa troca de mensagens não tem um impacto muito alto em termos de consumo extra de banda. No entanto, pode acarretar em algum atraso no tempo de mapeamento até que sua coordenação seja efetivada entre vários domínios. Ao se analisar o comportamento das diferentes abordagens, percebe-se que em ambos os cenários a abordagem *VE\_d2* gera uma carga mais alta de mensagens trocadas em relação a abordagem *VE\_d1*, pois em *VE\_d1*, o orquestrador já possui uma certa informação, o que implica na realização da consulta em menos domínios. Por outro lado, *VE\_d2* é capaz de prover soluções de melhor qualidade para o mapeamento. Dessa forma, observa-se que uma maior troca de mensagens permite uma melhor coordenação e otimização dos recursos.

## 7. Conclusão

Este trabalho analisou diferentes abordagens para o mapeamento de redes virtuais através de um orquestrador com conhecimento total, parcial e sem conhecimento do SN. Os resultados obtidos indicam que o aumento do gasto de energia e da troca de mensagens, junto à diminuição do lucro acarretados por modelos descentralizados, ainda são desafios a serem superados. De acordo com as abordagens, é possível aumentar o lucro de um domínio ao otimizar o consumo energético, porém com uma maior troca de mensagens. Nos experimentos realizados, a troca de mensagens não tem um impacto alto em termos de redução do número de VNs aceitas. No entanto, em termos práticos, isto pode acarretar atraso no tempo de mapeamento até que sua coordenação seja efetivada entre vários domínios. A heurística de mapeamento GRASPF-RVNS adotada se comportou bem junto ao orquestrador, mesmo perante a dificuldade do problema *online* e MD.

Em trabalhos futuros pretende-se estender e avaliar experimentalmente os modelos propostos para o VNE junto a conceitos de funções de rede e redes definidas por *software*, com a aplicação de conceitos de demanda elástica e variação do consumo dos componentes físicos. Ainda, aperfeiçoar o modelo de energia adotado, aplicar e considerar outros fatores como a variação de tarifação da energia elétrica, ganhos com energias limpas e crédito de carbono além de despesas de operação e expansão.

## Agradecimentos

Os autores agradecem ao CNPq, CAPES e FAPEMIG.

## Referências

Alkmim, G. P., Batista, D. M., and Fonseca, N. L. S. d. (2011). Mapeamento de Redes Virtuais em Substratos de Rede. *XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 45–58.

- Anderson, T., Peterson, L., Shenker, S., and Turner, J. (2005). Overcoming the Internet impasse through virtualization. *Computer*, 38(4):34–41.
- Araújo, S. M. A., Guidoni, D. L., and Souza, F. S. H. D. (2015). A Metaheuristic Approach for the Virtual Network Embedding Problem. In *2015 IEEE 14th International Symposium on Network Computing and Applications*, pages 68–75.
- Bojanova, I. and Voas, J. (2017). Trusting the Internet of Things. *IT Professional*, 19(5):16–19.
- Chiaraviglio, L., Mellia, M., and Neri, F. (2012). Minimizing ISP Network Energy Cost: Formulation and Solutions. *IEEE/ACM Transactions on Networking*, 20(2):463–476.
- Chowdhury, M., Rahman, M. R., and Boutaba, R. (2009). Virtual Network Embedding with Coordinated Node and Link Mapping. In *INFOCOM*, pages 783–791. IEEE.
- Dalla-Costa, A. G., Schimunek, M. A. K., Wickboldt, J. A., Both, C. B., Gasparly, L. P., and Granville, L. Z. (2017). NFV em Redes 5G: Avaliando o Desempenho de Composição de Funções Virtualizadas via Maestro. *XXXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 788–801.
- Eramo, V., Miucci, E., and Ammar, M. (2016). Study of Reconfiguration Cost and Energy Aware VNE Policies in Cycle-Stationary Traffic Scenarios. *IEEE Journal on Selected Areas in Communications*, 34(5):1281–1297.
- Feo, T. A. and Resende, M. G. C. (1995). Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, 6(2):109–133.
- Fischer, A., Botero, J. F., Beck, M. T., de Meer, H., and Hesselbach, X. (2013). Virtual network embedding: A survey. *IEEE Communications Surveys Tutorials*, 15(4):1888–1906.
- Fischer, A. and de Meer, H. (2016). Generating Virtual Network Embedding Problems With Guaranteed Solutions. *IEEE Transactions on Network and Service Management*, 13(3):504–517.
- Mladenović, N. and Hansen, P. (1997). Variable Neighborhood Search. *Computers & Operations Research*, 24(11):1097–1100.
- Nguyen, L. D., Kim, N., Kim, S., and Kim, C.-K. (2017). RT-VNE: A real-time strategy for Virtual Network Embedding towards resource efficiency. In *2017 International Conference on Information Networking (ICOIN)*, pages 185–190.
- Oliveira, R. R., Marcon, D. S., Bays, L. R., Neves, M. C., Gasparly, L. P., Medhi, D., and Barcellos, M. P. (2015). Opportunistic resilience embedding (ORE): Toward cost-efficient resilient virtual networks. *Computer Networks*, 89:59 – 77.
- Samuel, F., Chowdhury, M., and Boutaba, R. (2013). PolyViNE: policy-based virtual network embedding across multiple domains. *Journal of Internet Services*, 4(1):6.
- Waxman, B. M. (1988). Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622.
- Zhu, Y. and Ammar, M. (2006). Algorithms for assigning substrate network resources to virtual network components. *INFOCOM 2006. 25th IEEE International Conference on Computer Communications*, pages 1–12.