

## **Estratégias de Reconfiguração no Mapeamento de Redes Virtuais**

**Samuel Moreira Abreu Araújo**

Departamento de Ciência da Computação  
Universidade Federal de Minas Gerais (UFMG)  
CEP 31270-901 – Belo Horizonte – MG – Brasil  
smaa@ufmg.br

**Fernanda Sumika Hojo de Souza**

Departamento de Ciência da Computação  
Universidade Federal de Sao João del-Rei (UFSJ)  
CEP 36301-360 – Sao João del-Rei – MG – Brasil  
fsumika@ufs.br

### **RESUMO**

O problema de mapeamento de redes virtuais é considerado NP-difícil e consiste em decidir onde requisições de redes virtuais serão hospedadas em uma rede física de forma a compartilhar os recursos existentes. Em sua abordagem estática, uma vez mapeada a requisição virtual permanece no mesmo local físico até que deixe o provedor. Na abordagem dinâmica, admite-se a reconfiguração das requisições ativas, ou seja, a rede virtual pode ter seu mapeamento alterado durante seu tempo de vida. Duas estratégias de reconfiguração reativa são propostas neste trabalho: reativa à rejeição de uma nova VN e reativa à saída de uma VN. Resultados computacionais demonstram que as estratégias de reconfiguração levam ao aumento da taxa de aceitação e redução dos custos de mapeamento para o provedor. A estratégia reativa à rejeição mostra-se melhor para aumentar a taxa de aceitação enquanto a estratégia a cada saída é mais indicada para melhorar a utilização dos recursos físicos e conseqüentemente diminuir os custos.

**PALAVRAS CHAVE.** Reconfiguração, Mapeamento, Virtualização de Redes.

**Tópicos (Otimização Combinatória, Metaheurísticas, Simulação)**

### **ABSTRACT**

The virtual network embedding problem is considered NP-hard and consists in deciding where virtual network requests will be hosted on a physical network in order to share existing resources. In its static approach, once mapped the virtual request remains in the same physical location until leaving the provider. In the dynamic approach, reconfiguration of active requests is allowed, i.e., a virtual network can have its mapping modified during its lifetime. Two reactive reconfiguration strategies are proposed in this paper: reactive to the rejection of a new VN and reactive to the expiration of a VN. Computational results show that the reconfiguration strategies lead to higher acceptance rate and lower mapping costs for the provider. The reactive strategy to rejection is suitable to increase the acceptance rate while the strategy based on the expiration of lifetime shows better to improve the use of physical resources and thereby reduce costs.

**KEYWORDS.** Reconfiguration, Embedding, Network Virtualization.

**Paper Topics (Combinatorial Optimization, Metaheuristics, Simulation)**

## 1. Introdução

A virtualização de redes faz parte de um novo paradigma computacional, que permite que várias redes virtuais (VNs) coexistam no mesmo substrato físico de rede (SN), compartilhando os recursos de forma eficiente. A implementação da técnica de virtualização possibilita que redes virtuais sejam aninhadas sobre uma mesma rede física, levando ao problema denominado Mapeamento de Redes Virtuais (*Virtual Network Embedding - VNE*) (Fischer et al., 2013). Este problema pertence à classe NP-difícil, sendo foco de estudo de diversos trabalhos da literatura.

A internet hoje é constituída de uma estrutura rígida e resistente a mudanças, impedindo que novos protocolos sejam implantados para auxiliarem no desenvolvimento das mais diversas aplicações. Tal rigidez foi denominada “ossificação da internet” (Zhu e Ammar, 2006). A principal causa deste problema se deve à adoção de serviços fim a fim de transferência de dados e uma arquitetura baseada nos protocolos TCP/IP. Apesar do funcionamento da internet ter sido um grande sucesso em sua criação, a expansão de aplicações atuais não foi prevista. Dessa forma, a virtualização de redes vem como uma alternativa para contornar este problema.

No cenário de virtualização são identificados três atores distintos, que são apresentados segundo Zhu et al. (2008) como: os provedores de infraestrutura, que detém a rede física e fazem a gerência dos recursos do SN; os provedores de serviço, que recebem as requisições a serem mapeadas no substrato de rede (contratam serviços dos provedores de infraestrutura); e por fim os provedores de conectividade, que são responsáveis por instanciar as requisições virtuais sobre as redes físicas e proverem uma cobertura de serviços fim a fim.

Neste contexto, o grande desafio para os provedores de serviço consiste em uma resolução adequada do problema de mapeamento de redes virtuais a fim de obter um melhor aproveitamento do substrato físico e garantir disponibilidade, confiabilidade, escalabilidade e segurança no atendimento das requisições dos clientes. Diversas propostas vem sendo apresentadas neste sentido, buscando maximizar as taxas de aceitação das requisições, minimizar os custos de mapeamento, maximizar o balanceamento de carga na rede, entre outros objetivos (Chowdhury et al., 2009; Cheng et al., 2011; Inführ e Raidl, 2011; Araújo e Souza, 2015). As restrições básicas estão relacionadas às capacidades de processamento nos nós e de largura de banda nos enlaces da rede física. Contudo, diversas outras restrições adicionais podem ser agregadas ao problema buscando aproximar de um cenário real, como por exemplo, restrições de localização geográfica preferencial, restrições de atraso para comunicação, restrições de resiliência a falhas, entre outras.

Análises adicionais dos trabalhos (Araújo e Souza, 2015; Araújo et al., 2015; Butt et al., 2010) demonstraram que em certos cenários o número de rejeições das requisições virtuais pode ser elevado, mesmo com folgas nos recursos do substrato e empregando técnicas de otimização adequadas. Por se tratar de um problema de natureza *online*, as requisições virtuais são associadas a um tempo de vida, sendo que após terem sua atividade expirada, os recursos do substrato físico volta a ficar disponível para utilização de outras requisições que venham a chegar ao provedor. Assim, requisições que tem seu tempo de vida finalizado deixam o substrato sendo responsáveis por sua possível fragmentação. Dessa forma, embora hajam recursos disponíveis, tais recursos podem se tornar inacessíveis (isolados) para um mapeamento completo.

Uma alternativa para se mitigar este problema consiste na reconfiguração do substrato físico. Com a reconfiguração é possível prover um novo mapeamento para as VNs ativas de maneira que estas consumam menos recursos e possivelmente proporcionem espaços menos fragmentados para aceitação de novos mapeamentos. A Figura 1 apresenta um cenário no qual a reconfiguração permite a aceitação de uma nova requisição que seria rejeitada.

Devido à natureza *online* do problema e a necessidade de incorporar outras restrições que ajudam a aproximar o problema a uma aplicação real, alguns fatores podem ser apontados como dificultadores de se controverter a situação apresentada. Estes fatores incluem: a chegada das requisições não é conhecida *a priori* pelo provedor de serviço, o tempo de vida das requisições pode ser bastante heterogêneo e também não é conhecido, requisitos de localização preferencial de

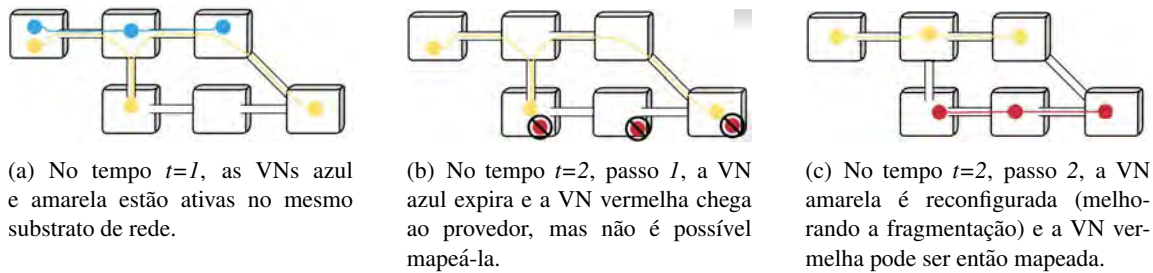


Figura 1: Reconfiguração de VNs no substrato

determinados nós da rede podem ser demandados pelos clientes, requisitos de atraso nos enlaces virtuais precisam ser atendidos, entre outros. Assim, determinados cenários podem levar a uma grande fragmentação do SN, dificultando o bom funcionamento de um algoritmo de mapeamento que não considera a possibilidade de reconfiguração.

Ao considerar a estratégia de reconfiguração, surgem diversas possibilidades em função da razão de sua aplicação. Assim, é possível considerar etapas de reconfiguração que ocorrem de forma temporal ou a partir de algum evento que ocorra na rede. A classificação a seguir apresenta algumas possibilidades de reconfiguração.

**Periódica:** em determinados intervalos de tempo o algoritmo aplica a reconfiguração, independente das VNs que estejam ativas ou do estado de seus componentes físicos.

**Reativa:** a partir de um evento ocorrido na rede, o algoritmo executa a etapa de reconfiguração. Os eventos podem ser a saída de uma requisição, picos no consumo dos recursos, a rejeição de uma requisição por falta de capacidade, entre outros aspectos relevantes.

Neste trabalho, duas estratégias de reconfiguração foram avaliadas, com o objetivo de identificar os cenários em que cada uma delas mostra-se mais adequada. Assim, foram observados o número de vezes em que a reconfiguração ocorreu e qual foi o impacto de sua aplicação em termos de melhorias na taxa de aceitação, receita gerada, e também o custo computacional para sua realização. As estratégias selecionadas são classificadas como reativas e ocorrem frente à ocorrência dos seguintes eventos:

**Rejeição de VN:** nesta estratégia, a cada VN que chega e não consegue ser mapeada (e consequentemente será rejeitada), a etapa de reconfiguração atua sobre o SN, remapeamento as VNs ativas em possíveis novas posições. Com o SN reconfigurado, a VN a ser rejeitada é submetida novamente a uma tentativa de reconfiguração.

**Saída de VN:** nesta estratégia, a cada VN mapeada que tem seu tempo de vida expirado (devolvendo seus recursos ao SN), a etapa de reconfiguração é invocada com o objetivo de melhorar a fragmentação do SN, balanceando a distribuição do uso de seus recursos. Dessa forma, na chegada da próxima VN o algoritmo terá mais chance de sucesso para realizar o mapeamento.

O restante deste artigo está organizado como se segue. A seção 2 apresenta uma breve revisão dos trabalhos da literatura. Na seção 3, o problema de mapeamento de redes virtuais é formalizado, enquanto na seção 4 é descrita a heurística aplicada juntamente com as estratégias de reconfiguração. A seção 5 consiste na análise de desempenho realizada, seguida pelas conclusões na seção 6.

## 2. Trabalhos Relacionados

Os trabalhos relacionados ao problema de mapeamento de redes virtuais podem ser divididos em dois grupos segundo a abordagem dinâmica e estática no tratamento das VNs (Fischer

et al., 2013). Na abordagem estática não contempla a possibilidade de realocar as VNs para melhorar o desempenho do substrato, ou seja, uma vez mapeada, a VN permanece na mesma alocação até o final do seu tempo de vida. Na abordagem dinâmica, é possível reconfigurar as VNs ativas a fim de reorganizar a alocação dos recursos e otimizar sua utilização.

Em trabalhos preliminares (Araújo e Souza, 2015) a versão estática do problema foi estudada, na qual uma heurística GRASP-RVNS foi proposta para resolução do problema. O algoritmo é composto por uma heurística construtiva gulosa que prioriza o mapeamento dos componentes da VN em nós menos saturados e enlaces que proporcionem o caminho mais curto para conexão ou o caminho menos saturado. A busca local é realizada através do RVNS, no qual duas estruturas de vizinhança são adotadas: a primeira é baseada no remapeamento de um enlace virtual da VN e a segunda no remapeamento de um nó virtual e conseqüentemente seus enlaces virtuais. O algoritmo proposto mostrou seu melhor desempenho para o parâmetro alfa do GRASP de 0.6 e mapeamento de enlaces virtuais baseado no caminho mais curto, ao ser comparado com abordagens da literatura.

Em (Araújo et al., 2015), a versão estática do problema foi avaliada em cenários de falha. Para prover resiliência a falha simples de enlaces, foi proposta a alocação de banda através de dois caminhos disjuntos em arestas para cada enlace virtual de uma VN. Resultados computacionais demonstraram que esta reserva adicional de recursos pode comprometer significativamente a taxa de aceitação das requisições.

Yu et al. (2008) propôs um modelo para o mapeamento das redes virtuais baseado em dois estágios, no primeiro é feito o mapeamento de nós, em caso de sucesso é feito posteriormente o mapeamento dos enlaces usando algoritmos de caminhos mais curtos e o de *multicommodity flow*.

Em Sun et al. (2012), foram desenvolvidas variações de algoritmos gulosos denominados VNE2N\_G e VNE2N\_R. A primeira variação é baseada em uma estratégia gulosa para escolher o nó substrato com maior recurso livre para hospedar um nó virtual e a segunda estratégia propõe uma escolha aleatória para selecionar um nó no substrato para hospedar um nó virtual.

Quando uma abordagem dinâmica do problema é considerada, novos métodos passam a ser empregados. Em uma modelagem abordando o problema *online*, Zhu e Ammar (2006) propõem duas abordagens para comparar as melhoras geradas pelo uso da reconfiguração, chamadas de VNA I, sem reconfiguração; e VNA II com reconfiguração. Sua heurística apoia a minimização das regiões críticas em componentes mapeados. Neste caso somente as requisições virtuais mapeadas sobre os componentes físicos mais estressados são reconfiguradas. Para (Zhu e Ammar, 2006) o conceito de *stress* se dá pelo número de mapeamentos que determinado componente físico recebeu. Os autores concluem também que a reconfiguração gera um uso mais igualitário dos recursos.

Outra abordagem, proposta por Yu et al. (2008), consiste no monitoramento periódico do substrato físico da rede, a fim de detectar os enlaces mais sobrecarregados em quantidade de recursos usados. A partir deste ponto é proposta a divisão dos enlaces virtuais sobrecarregados em caminhos físicos distintos para aliviar o estresse do ponto. O problema desta abordagem é não ser viável em um ambiente real de redes, uma vez que um enlace virtual deve ser alocado a um único caminho físico para não ter problema de desempenho em relação a sua demanda. Além disso, o algoritmo não prevê o remapeamento de nós virtuais, o que restringe sua melhora da reconfiguração.

Em Butt et al. (2010), é proposto um mecanismo reativo de reconfiguração, que reconfigura a cada rejeição. Em seu algoritmo são detectados os pontos que estão causando a rejeição, e tenta-se mapear esses nós para outros nós vizinhos aptos. Em outra etapa de seu algoritmo, o problema de fluxo máximo é utilizado para determinar os enlaces que podem estar causando o gargalo.

Na proposta de (Fajjari et al., 2011), a rede virtual não é completamente realocada, e sim alguns pontos chamados de *star*, a fim de minimizar o custo da reconfiguração. Através da reconfiguração, foi possível reduzir as taxas de rejeição em até 68% em relação a seu algoritmo de mapeamento VNE-Greedy.

Em (Ghazisaeedi e Huang, 2015), foi proposta a execução da reconfiguração de enlaces e nós utilizados, a fim de minimizar-se o consumo de energia elétrica usado para manter ativa a

infraestrutura física. No algoritmo heurístico proposto chamado ONL-LLns-F, é feita uma aprendizagem dos momentos de picos de consumo alto e baixo, baseado em mapeamentos anteriores. Em momentos de poucos mapeamentos, as redes são reconfiguradas de maneira a se consumir menos energia, enquanto que em momentos de alto requisição, o mapeamento original é retomado, utilizando mais recursos.

### 3. Definição Formal do Problema

O problema de mapeamento de redes virtuais pode ser modelado através de grafos, onde a rede física é representada por um grafo não direcionado ponderado  $G^P = (N^P, E^P)$ , onde  $N^P$  e  $E^P$  correspondem ao conjunto de nós e enlaces do substrato físico, respectivamente. Cada nó  $n^P \in N^P$  do substrato está associado a uma capacidade de CPU  $c(n^P)$  e sua localização  $l(n^P)$  em um sistema de coordenadas geográficas  $(x,y)$ . Cada enlace do substrato  $e^P \in E^P$  está associado a uma capacidade de largura de banda  $b(e^P)$ .

Da mesma forma, cada requisição de rede virtual é modelada como um grafo  $G^V = (N^V, E^V)$ , com suas respectivas demandas de processamento, largura de banda e localização geográfica preferencial  $(x,y)$ , além de um parâmetro  $D^V$  que indica a distância máxima para o mapeamento de cada nó virtual em relação a sua localização preferencial.

Uma solução para o problema consiste no mapeamento  $f : G^V \rightarrow G^P$ , tal que para todo nó virtual  $n^V \in N^V$ , exista um nó físico  $n^P \in N^P$  com capacidade residual de CPU para hospedá-lo e que esteja dentro da sua área de localização preferencial, e para todo enlace virtual  $e^V \in E^V$ , exista um caminho físico composto por enlaces físicos  $e^P \in E^P$  com capacidade residual de largura de banda suficiente para mapeá-lo.

A Figura 2 exemplifica o mapeamento de duas VNs no mesmo substrato físico. As figuras 2(a) e 2(b) mostram a chegada de duas requisições representadas por seus respectivos grafos. A Figura 2(c) apresenta um exemplo de restrição de localização para um nó da VN1, que possui demanda de 7 unidade de CPU. Este nó tem uma localização preferencial que permite mapeá-lo em três nós dos substrato, de acordo com seu raio. A Figura 2(d) apresenta um mapeamento válido para as duas requisições, que respeita todas as restrições impostas.

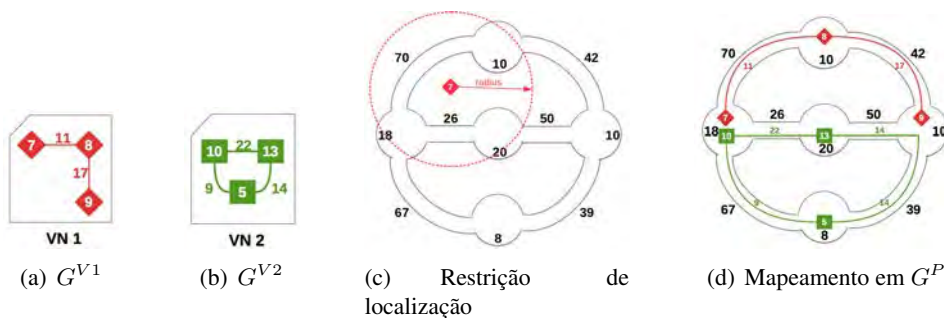


Figura 2: Mapeamento de duas VNs no substrato

O processo de reconfiguração vem como uma estratégia a mais utilizada sobre o problema definido anteriormente, para tentar aumentar a taxa de aceitação em uma abordagem *online* do problema, na qual as requisições virtuais chegam e deixam o substrato ao longo do tempo.

## 4. Heurística de Mapeamento com Estratégias de Reconfiguração

### 4.1. Etapa de Mapeamento

A heurística de mapeamento utilizada neste trabalho foi proposta em (Araújo e Souza, 2015) e consiste em uma combinação GRASP-RVNS (Feo e Resende, 1995; Mladenović e Hansen, 1997). O algoritmo é composto por uma heurística construtiva gulosa e uma busca local composta por duas vizinhanças. A heurística construtiva gulosa é dividida em duas partes: mapeamento dos nós virtuais e mapeamento dos enlaces virtuais. No mapeamento dos nós virtuais, é priorizada a utilização de componentes físicos menos saturados, ou seja, aqueles com maior capacidade residual



de CPU e largura de banda de chegada/saída no nó. No mapeamento de enlaces virtuais, aplica-se um algoritmo de caminho mais curto para geração da rota que irá conectar os nós físicos nos quais os extremos do enlace virtual foram hospedados. A busca local é baseada em uma adaptação RVNS, com a primeira vizinhança baseada no remapeamento de um enlace virtual da VN e a segunda no remapeamento de um nó virtual e consequentemente seus enlaces virtuais.

A versão escolhida para utilização foi a GR\_RVNS\_L6\_PPF, por ter apresentado as maiores taxas de aceitação e receita gerada pelas VNs mapeadas em relação às outras versões avaliadas nos experimentos de (Araújo e Souza, 2015). No entanto, foi aplicada uma pequena modificação em sua implementação, que é destacada a seguir. Nesta versão, a geração de rotas é baseada no caminho mais curto em termos de número de saltos. Assim, como o peso dos enlaces é fixo (unitário), o algoritmo de Dijkstra adotado anteriormente foi substituído por uma busca em largura, visando melhorar seu desempenho.

A etapa de mapeamento ocorre a cada chegada de uma nova VN. Desta forma, quando uma nova requisição chega ao provedor, é feita a tentativa de mapeamento da mesma utilizando a capacidade residual do substrato de rede, ou seja, através dos recursos que ainda encontram-se disponíveis. Caso a etapa de mapeamento não obtenha sucesso, a VN seria então rejeitada por incapacidade de atendimento.

#### 4.2. Etapa de Reconfiguração

A etapa de reconfiguração consiste em alterar os mapeamentos previamente realizados. Desta forma, uma VN já mapeada poderá mudar sua posição física no SN, acarretando a mudança de uso dos recursos disponíveis. Esta etapa atua de forma similar a uma busca local, com a diferença de ser aplicada sob requisições que não seriam mais passíveis de mudança, pois já foram mapeadas quando chegaram ao provedor.

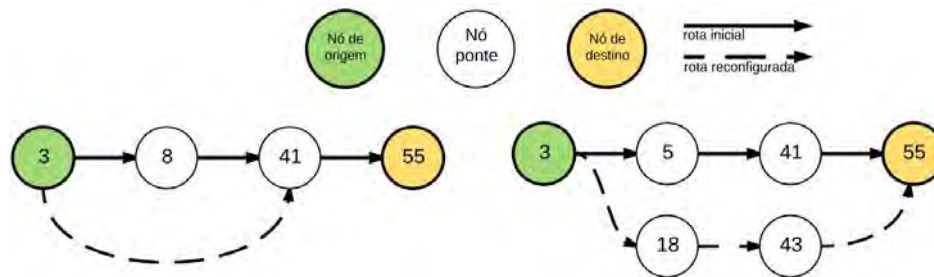


Figura 3: Exemplos de reconfiguração do enlace virtual entre os nós 3 e 55, inicialmente mapeado no caminho físico (3-8-41-55).

A Figura 3 apresenta um exemplo da reconfiguração de um enlace virtual entre os nós 3 e 55, inicialmente mapeado no caminho físico (3-8-41-55). Neste exemplo, apenas o mapeamento de enlace sofre alteração, sendo que o mapeamento de nós permanece inalterado, ou seja, nos mesmos nós físicos. À esquerda, a nova rota física (3-41-55) tem um salto a menos e possui um enlace físico (41-55) em comum com a rota original. Neste caso, a banda antes utilizada entre os nós (3-8) e (8-41) fica livre para ser usada por outro mapeamento. Esta reconfiguração implica na redução do custo de mapeamento, pois o número de enlaces físicos utilizados é reduzido. À direita, a nova rota física (3-18-43-55) corresponde a um caminho físico com o mesmo número de saltos, porém enlaces físicos todos diferentes da rota original. A banda utilizada na rota original é liberada para um novo mapeamento. Neste caso, a nova rota pode prover uma melhoria no balanceamento de carga da rede.

O Algoritmo 1 ilustra a atuação da reconfiguração nas VNs ativas no substrato. O laço das linhas (2)-(4) itera sob todas as VNs ativas no provedor para sua realocação em novas posições através do RVNS caso isto seja benéfico à fragmentação do meio físico. Se pelo menos uma das VNs tiver provido uma melhoria na solução (mapeamento atual), é retornado um resultado verdadeiro

(linha 6). É importante observar que cada VN ativa só é reconfigurada uma vez. Contudo, como são realizadas alterações sequenciais, um novo laço poderia levar a mais melhorias.

---

**Algoritmo 1** Reconfigura()

---

```

1: ret ← 0
2: for i ← 1 to NumAtivas do
3:   ret ← ret + RVNS(V.active[i])
4: end for
5: if ret > 0 then
6:   return true
7: end if
8: return false

```

---

**4.3. Evento “Rejeição de VN”**

Neste cenário, considere a chegada de uma nova VN ao provedor, que passa pela etapa de mapeamento. Quando a etapa de mapeamento não tem sucesso, é gerado um evento de “rejeição de VN”, devido à impossibilidade de aceitação. Neste momento, o gerenciador interrompe sua execução normal para acionar a etapa de reconfiguração das VNs. Após sua finalização é realizada uma nova tentativa de mapeamento da VN que seria rejeitada. Esta estratégia é subdividida em três fases complementares como pode ser observado no Algoritmo 2.

Entre as linhas (1)-(4) do Algoritmo 2, a etapa de reconfiguração e a tentativa de mapeamento da nova VN é realizada por no máximo três vezes. Como mencionado anteriormente, o algoritmo de reconfiguração pode apresentar melhorias se invocado mais de uma vez em sequência. Isto ocorre porque ao alterar a alocação das VNs, é possível que as VNs já alteradas possam prover novas melhorias caso voltem a ser analisadas.

Caso o mapeamento não obtenha sucesso, o algoritmo 2 entra nas linhas (5)-(10), onde a reconfiguração é realizada de forma exaustiva, até que nenhuma VN ativa seja capaz de melhorar a solução. Após, nova tentativa de mapeamento é feita para a nova VN.

Se o mapeamento não tiver sucesso novamente, a VN é finalmente rejeitada (linhas (11)-(13)), retornando o controle ao gerenciador dos eventos.

---

**Algoritmo 2** ReativoAREjeicao (int *contador*)

---

```

1: if (contador < 3) then
2:   Reconfigura()
3:   return true      # Nova tentativa de mapeamento
4: else if (contador == 3) then
5:   while (Reconfigura() == true) do
6:     # Efetua reconfiguração exaustivamente
7:   end while
8:   return true      # Nova tentativa de mapeamento
9: else if (contador == 4) then
10:  return false     # Aceita a rejeição
11: end if

```

---

A divisão em três fases se dá pelo fato da reconfiguração ser extremamente custosa, uma vez que a cada VN movimentada sobre o SN, acaba-se potencialmente liberando recursos para que outras VNs também se movimentem para novas posições. Como não é necessária a movimentação até o ponto de estabilização para que uma VN possa ser aceita, optou-se pela adoção das três fases. Foi observado em testes preliminares que a execução gradual de movimentações tende a ser mais rápida do que a adoção da inicial pela reconfiguração à exaustão. Ainda, é esperado que em

momentos de pouca rejeição esta estratégia faça poucas paradas para a reconfiguração, ao passo que em momentos de alta rejeição gera maior *overhead*.

#### 4.4. Evento “Saída de VN”

Neste cenário, considere que o gerenciador detectou que uma VN ativa teve seu tempo de vida terminado. É gerado então um evento de “saída de VN”, que consiste em liberar os recursos utilizados pela VN em questão, para que estes possam ser utilizados em novos mapeamentos. A estratégia baseada neste evento tem como premissa que este é um bom momento para se efetuar a reconfiguração, porque acarretará na melhoria do mapeamento das VNs ativas, gerando um balanceamento de carga no SN. Esta estratégia é mais simples e é apresentada no algoritmo 3. Basicamente, a cada evento de saída, as VNs ativas são reconfiguradas exaustivamente até que estejam estabilizadas, isto é, não exista mais melhoria associada à alteração das posições.

---

#### Algoritmo 3 ReativoASaida(int *saida*)

---

```

1: if saida = 1 then
2:   while (Reconfigura())== true do
3:     # Efetua reconfiguração exaustivamente
4:   end while
5: end if

```

---

Em momento de poucos mapeamentos, esta estratégia tende a ser rápida e eficaz, melhorando os mapeamentos das VNs que estão ativas e consequentemente reduzindo o custo para o provedor de serviço. Porém, em momentos de alta taxa de aceitação e consequente grande número de saídas, esta estratégia tende a perder desempenho pois executa a reconfiguração várias vezes, podendo até mesmo ser redundante, caso hajam duas saídas sequenciais de VN.

#### 5. Avaliação de Desempenho

Através dos experimentos computacionais realizados, foram comparadas abordagens sem o uso de reconfiguração, e as duas estratégias de reconfiguração propostas. Assim, foi adotado um conjunto de métricas que demonstram as principais diferenças entre as abordagens, listadas a seguir. **Taxa de aceitação:** é dada pela razão entre o número de VNs ativas no momento e o número total de VNS que chegaram ao provedor até o momento.

**Receita das VNs aceitas:** é dada pelo lucro gerado pela VN caso ela seja aceita durante o tempo em que está ativa, sendo proporcional à quantidade de recursos que ela demanda.

**Custo das VNs aceitas:** é expresso em função dos recursos utilizados pela VN durante o tempo em que encontra-se ativa.

**Taxa de utilização de nós:** é dada pela média das razões entre a quantidade de CPU utilizada e a capacidade do nó físico.

**Taxa de utilização de enlaces:** é dada pela média das razões entre a quantidade de largura de banda utilizada e a capacidade do enlace físico.

**Função objetivo:** é definida como o balanceamento de carga na rede, sendo expressa pelo somatório das razões entre a quantidade de largura de banda utilizada e a capacidade de todos os enlaces físicos.

**Tempo computacional:** é dado pelo tempo total de execução do algoritmo, para realizar as tarefas de mapeamento e reconfiguração.

As instâncias utilizadas foram propostas por (Chowdhury et al., 2009) e são as mesmas utilizadas em (Araújo e Souza, 2015). Os testes foram realizados em um computador Intel G870 3.10GHz, com 16gb de RAM DDR3, utilizando o sistema operacional Ubuntu 14.04.2 LTS. Para cada abordagem, foram realizadas 30 execuções, sendo apresentados os resultados médios. Foi utilizado um intervalo de confiança de 95%, onde a baixa dispersão mostra o bom comportamento do algoritmo perante aos cenários avaliados.



A Figura 4 apresenta a média do número absoluto de VNs aceitas ao final da simulação (Figura 4(a)) e o tempo computacional médio (Figura 4(b)). Três cenários são apresentados: o primeiro de baixa densidade, com chegada de 250 VNs; o segundo de média densidade, com chegada de 500 VNs e o terceiro, com a chegada de 2000 VNs. Em todos os cenários, a simulação ocorre durante um intervalo de 50.000 unidades de tempo. É possível observar que a abordagem com estratégia de reconfiguração a cada rejeição apresenta a maior aceitação, mostrando-se superior nos três cenários analisados.



(a) Média do número absoluto de VNs aceitas ao final da simulação (b) Tempo computacional médio ao final da simulação

Figura 4: Cenário de simulação

Observando a Figura 4(a), é possível perceber que quando o foco é maximizar o número de aceitações, a reconfiguração resulta em maior impacto nos cenários com alta rejeição, como o de 2000 VNs. Este fato pode ser explicado devido ao cenário possibilitar à reconfiguração um espaço maior para melhoria do número de aceitações. No cenário de baixa densidade (por exemplo, com a chegada de 250 VNs), como a aceitação já é alta, existe pouca possibilidade da reconfiguração trabalhar em sua melhoria. Na Figura 4(b), observa-se que cenários com maior aceitação (como o de 500 e 250 VNs), a estratégia que efetua a reconfiguração a cada saída, tende a ser mais custosa em tempo computacional do que a que efetua a reconfiguração a cada rejeição. Isto ocorre pois muitas aceitações implicam também um muitas saídas das VNs que tem seu tempo expirado.

A Tabela 1 apresenta as melhorias na taxa de aceitação para o cenário de 2000 requisições, podendo atingir até 10,33%. Vale lembrar que o algoritmo de mapeamento utilizado já possui um bom desempenho se comparado a outras soluções propostas na literatura (Araújo e Souza, 2015). Assim, o espaço para melhoria também fica reduzido. Neste cenário, a estratégia mais indicada é a reconfiguração a cada saída, que obteve um tempo de execução bem inferior ao de reconfiguração a cada rejeição, com melhorias similares na taxa de aceitação.

Na Tabela 2, é apresentado o cenário de 500 VNs, que já possui uma boa taxa de aceitação. Neste cenário, a reconfiguração não é indicada se o objetivo é aumentar a taxa de aceitação, pois não gera grandes melhoras e aumenta o tempo computacional. Por outro lado, se o objetivo for reduzir os custos de mapeamento, a reconfiguração é indicada, pois através da mesma é possível melhorar o posicionamento de cada VN ativa e reduzir o número de componentes estressados, maximizando o balanceamento.

Tabela 1: Relação Aceitação/Tempo (2000 VNs)

Cenário de 2000 VNs	Saída	Rejeição
Melhoria aceitação (%)	9,22%	10,33%
Tempo adicional (%)	207,28%	302,81%

Tabela 2: Relação Aceitação/Tempo (500 VNs)

Cenário de 500 VNs	Saída	Rejeição
Melhoria aceitação (%)	0,63%	3,91%
Tempo adicional (%)	86,4%	80,75%

As Figuras 5 e 6 apresentam a taxa de aceitação no decorrer do tempo de simulação. Pela Figura 5, em um cenário de alta taxa de rejeição, podemos ver que o algoritmo que executa a reconfiguração a cada rejeição, teve um desempenho superior às outras abordagens. O algoritmo que não faz reconfiguração tem uma taxa de aceitação mais baixa e além de prover menos aceitações, ele acaba tendo uma taxa de uso médio de enlaces maior, como será apresentado na

Figura 7. Nos cenário de média densidade (Figura 6) e similarmente no de baixa densidade<sup>1</sup>, pode-se observar que ambas estratégias de reconfiguração tendem a se entrelaçar, onde não é possível distinguir claramente qual é superior. O algoritmo que não provê reconfiguração apresenta um comportamento ligeiramente pior.

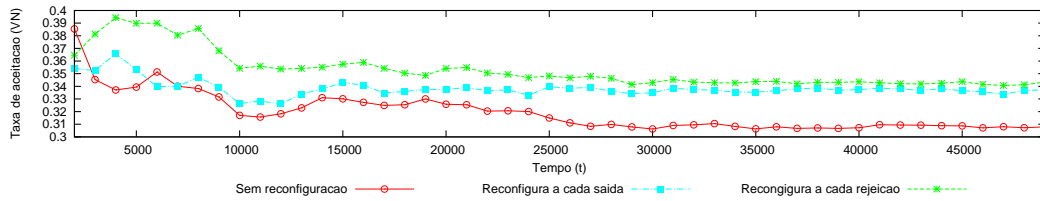


Figura 5: Taxa de aceitação de VNs no cenário com 2000 VNs e tempo de simulação de 50.000 unidades

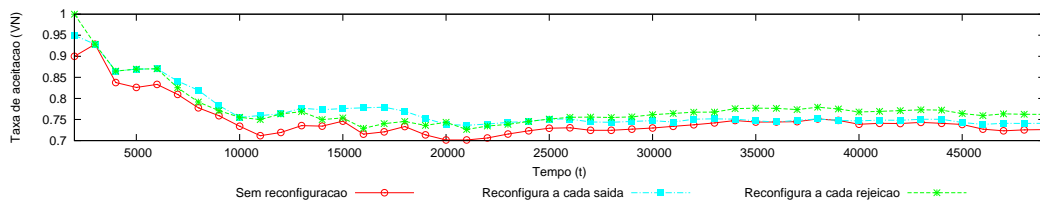


Figura 6: Taxa de aceitação de VNs no cenário com 500 VNs e tempo de simulação de 50.000 unidades

Na análise da taxa média de utilização de enlaces, observa-se que no cenário de alta densidade (Figura 7) a abordagem sem reconfiguração possui uma maior utilização dos enlaces mesmo mapeando menos VNs do que as abordagens com reconfiguração. Este fato demonstra como a reconfiguração proporciona um uso mais eficiente dos recursos físicos, promovendo uma maior taxa de aceitação, juntamente com um melhor balanceamento na carga dos enlaces. A abordagem de reconfiguração a cada saída tem uma ligeira superioridade quanto à economia de recursos de enlace em relação ao que faz a reconfiguração a cada rejeição. No cenário de média densidade (Figura 8), percebe-se que a abordagem de reconfiguração a cada rejeição é levemente mais vantajosa, já que possui uma taxa de aceitação maior.

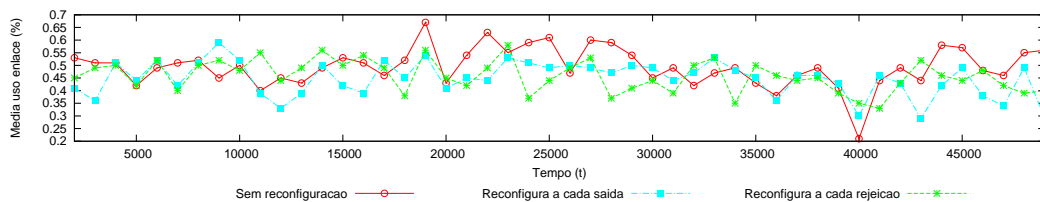


Figura 7: Taxa média de utilização de enlaces do substrato físico no cenário com 2000 VNs e tempo de simulação de 50.000

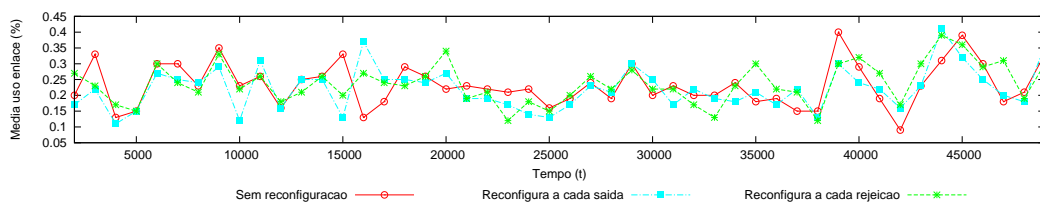


Figura 8: Taxa média de utilização de enlaces do substrato físico no cenário com 500 VNs e tempo de simulação de 50.000

Uma métrica importante para o provedor é a análise do lucro gerado, relacionada diretamente à taxa de aceitação. Quanto mais VNs aceitas, mais o provedor de infraestrutura arrecada. No cenário com 2000 VNs (Figura 9), percebe-se que as abordagens com reconfiguração geram bem mais receita que o modelo sem reconfiguração. Vale lembrar que este aumento no lucro é seguido por um aumento no tempo devido ao *overhead* da reconfiguração. Neste caso, a abordagem reativa

<sup>1</sup>As figuras referentes ao cenário de baixa densidade, com 250 VNs, foram omitidas por falta de espaço.

a cada saída possui a vantagem de gerar menor *overhead* em relação ao reativo a cada rejeição, conforme a Tabela 1. Em cenários de média (Figura 10) e baixa densidade, a métrica receita não é afetado significativamente, uma vez que a taxa de aceitação nas três abordagens já apresenta-se alta, gerando pouca melhoria através da reconfiguração.

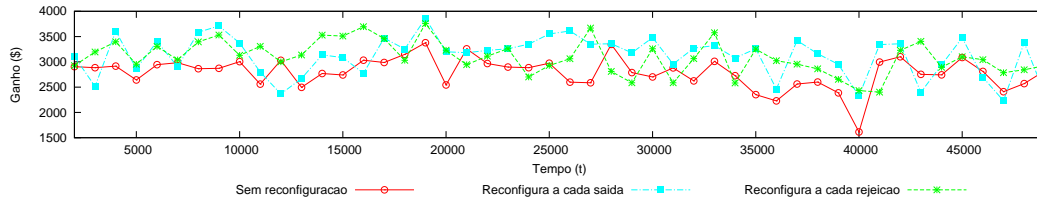


Figura 9: Receita gerada pelas VNs aceitas no cenário com 2000 VNs e tempo de simulação de 50.000

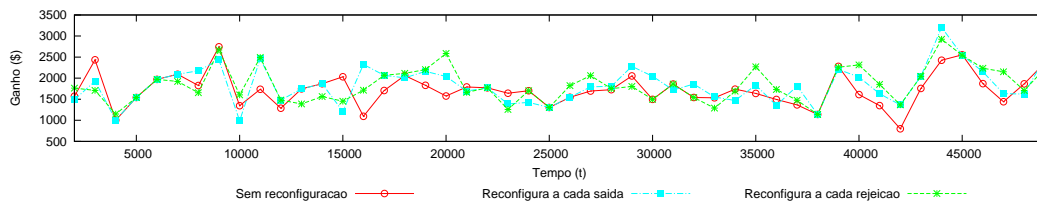


Figura 10: Receita gerada pelas VNs aceitas no cenário com 500 VNs e tempo de simulação de 50.000

Por fim, a análise do custo de se mapear as requisições, nos cenários de maior densidade (Figura 11) e de média densidade (Figura 12), nota-se que a abordagem com reconfiguração a cada saída, possui um custo mais baixo. Se comparada à abordagem sem reconfiguração, mesmo tendo mais aceitações seu custo ainda tende a ser inferior. Isto acontece devido ao fato de que se uma VN que está ativa deixa o provedor, o mapeamento de VNs ativas pode ser melhorado, usando menos enlaces e gerando economia.

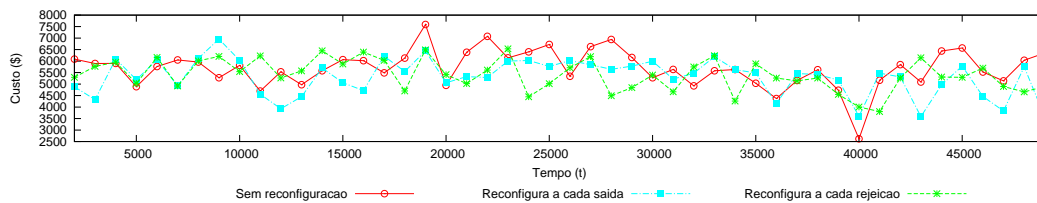


Figura 11: Custo gerado pelas VNs aceitas no cenário com 2000 VNs e tempo de simulação de 50.000

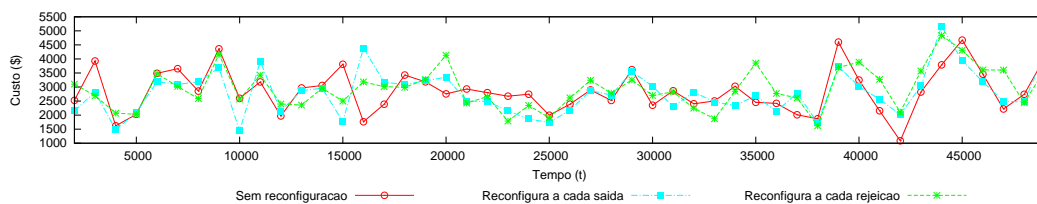


Figura 12: Custo gerado pelas VNs aceitas no cenário com 500 VNs e tempo de simulação de 50.000

## 6. Conclusão

A virtualização de redes é um mecanismo que permite várias redes independentes compartilhando a mesma estrutura física. Devido à natureza *online* do problema, as requisições chegam ao provedor, permanecem por certo período de tempo e tem seu tempo de vida expirado, liberando os recursos por elas utilizados para uso novamente. Contudo, o substrato físico passa a enfrentar o problema de fragmentação, que muitas vezes impede que novas requisições sejam atendidas, mesmo tendo recursos disponíveis.

A reconfiguração é uma estratégia flexível e mostrar-se como uma boa opção para mitigar o problema de gargalos no SN. Com a reconfiguração é possível economizar nos custos, aumentar os lucros do provedor de serviço, além de melhorar a qualidade de conexão para o usuário final. Em contrapartida deve-se ponderar quando e como aplicá-la, visto que sua execução é custosa em tempo

de processamento e pode causar muitas paradas no mapeamento dependendo do momento em que for executada. Nos experimentos realizados, as estratégias de reconfiguração geraram melhoras significativas em termos de taxa de aceitação nos cenários mais densos. Este cenários podem conter uma maior fragmentação, por possuírem uma maior rotatividade de requisições no provedor. Em cenários de menor densidade, a principal melhoria observada nas abordagens com reconfiguração foi a economia de recursos para o provedor, principalmente na estratégia de saída das VNs. Embora a reconfiguração possa prover melhorias no mapeamento de redes virtuais, existem cenários nos quais sua utilização ainda não seja suficiente, devido a restrições de atraso, localização e resiliência, por exemplo. Assim, como trabalhos futuros, pretende-se investigar a possibilidade de redimensionamento da rede física, quando esta for a última alternativa do provedor.

### Referências

- Araújo, S. M. A., Guidoni, D. L., e Souza, F. S. H. (2015). A metaheuristic approach for the virtual network embedding problem. *IEEE Computer Society*, p. 68–75.
- Araújo, S. M. A. e Souza, F. S. H. (2015). Uma heurística grasp-rvns para o problema de mapeamento de redes virtuais. *Anais do XLVII SBPO*, p. 2138–2149.
- Butt, N. F., Chowdhury, M., e Boutaba, R. (2010). Topology-awareness and reoptimization mechanism for virtual network embedding. *ACM SIGCOMM Computer Communication Review*, p. 27–39.
- Cheng, X., Su, S., Zhang, Z., Wang, H., Yang, F., Luo, Y., e Wang, J. (2011). Virtual network embedding through topology-aware node ranking. *SIGCOMM Comput. Commun. Rev.*, 41(2): 38–47. ISSN 0146-4833. URL <http://doi.acm.org/10.1145/1971162.1971168>.
- Chowdhury, N. M. M. K., Rahman, M. R., e Boutaba, R. (2009). Virtual network embedding with coordinated node and link mapping. In *INFOCOM*, p. 783–791. IEEE.
- Fajjari, I., Aitsaadi, N., Pujolle, G., e Zimmermann, H. (2011). Vnr algorithm: A greedy approach for virtual networks reconfigurations. *Globecom 2011 - IEEE Global Communications Conference*, p. 1–6.
- Feo, T. A. e Resende, M. G. (1995). Greedy randomized adaptive search procedure. *Journal of Global Optimization*, 6:109–133.
- Fischer, A., Botero, J. F., Beck, M. T., e de Meer, H. (2013). Virtual network embedding: A survey. *IEEE Communications Surveys & Tutorials*, p. 71–97.
- Ghazisaeedi, E. e Huang, C. C. (2015). Energy-aware node and link reconfiguration for virtualized network environments. *Computer Networks*, p. 460–479.
- Inführ, J. e Raidl, G. R. (2011). Introducing the virtual network mapping problem with delay, routing and location constraints. In Pahl, J., Reiners, T., e Voß, S., editors, *INOC*, volume 6701 of *Lecture Notes in Computer Science*, p. 105–117. Springer. ISBN 978-3-642-21526-1.
- Mladenović, N. e Hansen, P. (1997). Variable neighborhood search. *Computer and Operations Research*, 24(11):1097–1100. ISSN 0305-0548. URL [http://dx.doi.org/10.1016/S0305-0548\(97\)00031-2](http://dx.doi.org/10.1016/S0305-0548(97)00031-2).
- Sun, G., Yu, H., Anand, V., e Li, L. (2012). A cost efficient framework and algorithm for embedding dynamic virtual network requests. *Future Generation Computer Systems*, p. 206–219.
- Yu, M., Yi, Y., Rexford, J., e Chiang, M. (2008). Rethinking virtual network embedding: substrate support for path splitting and migration. *Computer Communication Review*, 38(2):17–29.
- Zhu, Y. e Ammar, M. (2006). Algorithms for assigning substrate network resources to virtual network components. *INFOCOM 2006. 25th IEEE International Conference on Computer Communications*, p. 1–12.
- Zhu, Y., Zhang-Shen, R., Rangarajan, S., e Rexford, J. (2008). Cabernet: Connectivity architecture for better network services. In *Proceedings of the 2008 ACM CoNEXT Conference*, CoNEXT '08, p. 64:1–64:6, New York, NY, USA. ACM. ISBN 978-1-60558-210-8. URL <http://doi.acm.org/10.1145/1544012.1544076>.