

Heurísticas para Mapeamento de Redes Virtuais de Sincronia Híbrida

Rômulo Reis De Oliveira¹, Rasha Ghassan Hasan¹, Fernando Luís Dotti¹

¹Faculdade de Informática
Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre, Brazil – Av. Ipiranga, 6681 Prédio 32 sala 505

romulo.reis@acad.pucrs.br, rasha.hasan@acad.pucrs.br

fernando.dotti@pucrs.br

Resumo. *Mapeamento de redes virtuais é um processo impactado pelos requisitos da aplicação. Recentemente, Redes Virtuais de Sincronia Híbrida (RVSHs) foram propostas e sua novidade é a habilidade de fornecer subconjuntos de recursos que se comportam de maneira síncrona, assim podendo suportar aplicações distribuídas que possam necessitar de sincronia parcial. Mapeamento de RVSHs já foi abordado pela literatura por meio de um modelo matemático para encontrar a solução ótima, mas essa abordagem demonstrou precisar de um tempo computacional considerável. Este trabalho apresenta e discute quatro heurísticas para o mapeamento de RVSHs: Duas heurísticas obtidas da adaptação dos algoritmos clássicos, worst fit e best fit, e duas heurísticas que consideram a topologia e demandas da conectividade durante o mapeamento. Os resultados semi-ótimos foram comparados a solução ótima e demonstraram considerável ganho de duas heurísticas em comparação as outras duas em termos de custo de mapeamento e taxa de aceitação.*

Abstract. *Virtual networks embedding is a process that is impacted by the application requirements. Recently, Hybrid Synchrony Virtual Networks (HSVNs) were proposed and their novelty is the ability to provide subsets of resources that behave synchronously, and thus may support distributed applications that may need partial synchrony. HSVNs embedding has already been tackled in the literature through a mathematical model for optimal solution, but this approach showed to need considerable computational time. This work presents and discusses four heuristics for HSVNs embedding: two heuristics obtained by the adaptation of the classic algorithms, worst fit and best fit, and two heuristics that consider topology and connectivity demands while mapping. The semi-optimal results are compared to the optimal solution and showed considerable gain of two heuristics in comparison to the other two in terms of mapping cost and acceptance ratio.*

1. Introdução

A virtualização de redes tem sido considerada a provável solução da ossificação da Internet [Anderson et al. 2005], [Chowdhury and Boutaba 2010], [Feamster et al. 2007] e [Turner and Taylor 2005], pois permite a coexistência de múltiplas redes virtuais em um mesmo substrato físico compartilhado, o que torna possível criar ambientes para avaliar

novas arquiteturas e protocolos. Além disso, essa tecnologia pode ser integrada às arquiteturas atuais. Neste sentido, Redes Virtuais são oferecidas a usuários que as utilizam para executar suas aplicações distribuídas.

Entre os diferentes requisitos de aplicações distribuídas à infraestrutura subjacente, a sincronia (quando o limite máximo de tempo de comunicação e processamento é conhecido) é um aspecto importante, pois determina o conjunto de algoritmos que podem ser empregados [Schneider 1993]. Ambientes completamente assíncronos tem menos custo, mas deixam maior a complexidade dos algoritmos distribuídos. Por outro lado, ambientes completamente síncronos tem alto custo, mas possibilitam a simplificação dos algoritmos. Naturalmente o uso parcial de sincronia é considerado na literatura [Cristian and Fetzer 1999], [Veríssimo 2006]. Quando um subconjunto de recursos tem comportamento síncrono, estes podem hospedar partes cruciais do sistema distribuído fazendo uso da sincronia e provendo benefícios às outras partes do sistema que podem estar executando em recursos que se comportam de forma assíncrona. Um detector de falhas perfeito é um exemplo de aplicação que se beneficia desse tipo de arquitetura [de Araujo Macedo and Gorender 2009].

Em [Hasan et al. 2014b] são propostas as RVSH, redes virtuais que oferecem abstrações adequadas para a demanda de redes com subconjunto de recursos que se comportam de maneira síncrona. O problema do mapeamento de redes virtuais deste tipo é definido e resolvido de forma ótima, sendo explorado tanto em [Hasan et al. 2014b] quanto em [Hasan et al. 2014a]. Entretanto, devido à potencial escala deste tipo de problema, que é NP-Difícil [Andersen 2002], e da necessidade de solução do mapeamento em curto período tempo, algoritmos heurísticos de mapeamento tem de ser considerados.

Neste artigo são apresentadas quatro abordagens heurísticas adaptadas para solucionar o problema de mapeamento de redes virtuais de sincronia híbrida. A partir dos resultados de um conjunto de experimentos, as soluções heurísticas são comparadas e analisadas com a solução ótima e entre si. Nossos resultados mostram que além do tempo de execução necessário para encontrar uma solução válida ser minimizado pela adoção das heurísticas, algumas destas heurísticas podem obter soluções com o mesmo custo de mapeamento que a solução ótima em alguns cenários.

Este artigo está organizado da seguinte maneira: a Seção 2 revisa trabalhos relacionados, enquanto a Seção 3 apresenta quatro heurísticas para o mapeamento de RVSH. A Seção 4 discute os resultados e por fim é concluído o artigo na Seção 5.

2. Trabalhos Relacionados

Mapeamentos de redes virtuais em redes físicas são extensamente estudados e uma visão geral destes pode ser encontrada em [Fischer et al. 2013]. Os mapeamentos variam em sua definição conforme os objetivos das redes e do mapeamento em questão. Por exemplo, [Zhu and Ammar 2006], [Yu et al. 2008] e [Lu and Turner 2006] consideram questões topológicas no mapeamento, já [Yu et al. 2011], [Rahman and Boutaba 2013], [Chen et al. 2010] e [Oliveira et al. 2013] consideram resiliência, [Bays et al. 2012] segurança e [Hasan et al. 2014b, Hasan et al. 2014a] sincronia. Em algumas propostas, também foram assumidas algumas hipóteses sobre o ambiente para minimizar o espaço do problema. [Zhu and Ammar 2006] e [Lu and Turner 2006], por exemplo, tratam as requisições de redes virtuais de maneira *offline*, onde todas as requisições de redes virtuais já são previamente conhecidas.

O uso de heurísticas para o mapeamento de redes virtuais é importante devido à complexidade computacional para calcular a solução ótima. Neste trabalho, considera-se trabalhos diretamente relacionados as propostas de heurísticas que consideram tempo de atraso. Neste contexto específico enquadra-se [Zhang et al. 2010], onde é proposto um algoritmo heurístico para o mapeamento de redes virtuais orientadas a serviços de *multicast* sujeitos a atrasos e variações de atraso. Nessa arquitetura há um recurso emissor e vários recursos receptores. Jogos online e teleconferência são exemplos de aplicações que se beneficiariam desse tipo de rede virtual. Nessas aplicações é esperado que as mensagens sejam entregues dentro de um período máximo de tempo e que a diferença entre os tempos de entrega nos múltiplos receptores seja mínima. [Zhang et al. 2010] apresenta um modelo matemático cujo o objetivo é minimizar o uso de recursos físicos necessários para efetuar o mapeamento. Também é considerado a capacidade e demanda de CPU e largura de banda dos recursos físicos e virtuais, além do atraso entre o emissor e cada receptor, assim como a variação de atraso entre os receptores. Posteriormente [Zhang et al. 2010] propõe uma heurística com base no modelo matemático, tendo como entrada uma requisição de criação de rede virtual com os seguintes parâmetros: recurso emissor, lista dos recursos receptores, capacidade de CPU dos nodos, capacidade de largura de banda dos enlaces, atraso máximo entre o emissor e receptor e a variação de atraso máximo permitido entre os receptores. Primeiramente são calculados *k shortest paths* entre o emissor e cada receptor, depois são filtrados a fim de localizar o melhor resultado.

O conceito de Redes Virtuais de Sincronia Híbrida (RVSH) foi proposto em [Hasan et al. 2014b, Hasan et al. 2014a], onde RVSHs são redes virtuais que têm subconjuntos de nós e enlaces que obedecem a limites de tempo para o processamento e comunicação. As RVSHs são voltadas basicamente para os sistemas distribuídos que podem se beneficiar de sincronia híbrida [Cristian and Fetzer 1999, Veríssimo 2006]. Em [Hasan et al. 2014b], os autores propuseram um modelo matemático para o mapeamento ótimo na forma de um programa inteiro misto (MIP) e eles mostraram que o modelo de mapeamento proposto foi capaz de otimizar a utilização de recursos físicos síncronos disponíveis na rede de substrato, visto que os recursos síncronos são mais caros. A principal limitação do modelo proposto era o tempo computacional necessário para se obter a solução ótima, chegando a atingir até 24 horas para encontrar uma solução de mapeamento para um experimento considerado pequeno. Neste trabalho, são propostas heurísticas que necessitam de um tempo computacional menor para encontrar uma solução e ao mesmo tempo minimizar o uso dos recursos síncronos.

Este trabalho se difere dos mencionados nos seguintes aspectos: i) O espaço do problema não é minimizado como em [Lu and Turner 2006], pois as requisições são tratadas de maneira *online*, assim como os recursos são considerados limitados, necessitando de controle de admissão; ii) Um subconjunto dos recursos do substrato físico oferece garantia temporal, assim como um subconjunto de recursos das redes virtuais tem como requisito garantia temporal. Diferentemente de [Zhang et al. 2010], onde todos os recursos físicos e virtuais têm garantias ou demandas por garantias temporais; iii) A terceira heurística considera a topologia da rede virtual na etapa do mapeamento dos nodos por meio da construção de uma árvore de alcançabilidade, diferentemente de [Yu et al. 2008] e [Lu and Turner 2006]; iv) A quarta heurística proposta, além de considerar a topologia durante o mapeamento dos nodos virtuais, também considera se os nodos físicos tem enlaces com largura de banda suficiente para mapear os enlaces do nodo virtual.

3. Algoritmos Heurísticos para mapeamento de RVSH

Nesta seção são descritas as notações e definições de variáveis utilizadas nos quatro algoritmos heurísticos para o mapeamento de redes virtuais de sincronia híbrida, também apresentados nesta seção. Sendo esses algoritmos heurísticos: i) centralizados, somente uma entidade executa o processo de mapeamento; ii) estáticos, não há alteração no mapeamento de uma rede virtual após seu mapeamento; e iii) concisos, busca alocar somente os recursos necessários, sem redundâncias [Fischer et al. 2013]. As quatro propostas têm como principais objetivos minimizar o custo de mapeamento dos enlaces e a alocação de recursos síncronos.

Todos algoritmos apresentados efetuam o mapeamento das RVSH conforme elas chegam, seguindo o modelo de fila. Todos também possuem duas etapas, sendo que na primeira etapa é efetuado o mapeamento dos nodos virtuais sobre os nodos físicos. Para um nodo físico ser elegível como hospedeiro, o mesmo deve ter capacidade de CPU disponível suficiente para hospedar um nodo virtual, o qual tem um custo de CPU, além de respeitar os atributos de sincronia. Um nodo virtual síncrono pode ser mapeado somente em nodos físicos síncronos, já os nodos virtuais assíncronos, podem ser mapeados tanto em nodos físicos assíncronos, quanto síncronos. Também não é permitido que mais de um nodo virtual da mesma rede virtual seja mapeado para o mesmo nodo físico, pois isso aumentaria o impacto caso ocorresse um erro ou falha no nodo físico.

Após todos os nodos virtuais serem mapeados, inicia-se a segunda etapa, onde ocorre o mapeamento dos enlaces virtuais sobre os enlaces físicos. Um enlace físico é considerado elegível para hospedar um enlace virtual somente quando o enlace físico tem uma capacidade de largura de banda disponível suficiente para hospedar o enlace virtual, o qual tem um custo de largura de banda. Além de respeitar a sincronia do enlace virtual, pois, os enlaces virtuais síncronos só podem ser mapeados em enlaces físicos síncronos, já enlaces virtuais assíncronos, podem ser mapeados tanto para enlaces físicos síncronos, quanto assíncronos. Como citado anteriormente, um dos objetivos é economizar recursos síncronos (nodos e enlaces), por estes serem mais caros, sendo assim, recursos virtuais assíncronos somente são mapeados em recursos físicos síncronos, caso não haja recursos físicos assíncronos elegíveis disponíveis. Cada enlace virtual pode ser mapeado tanto para um único enlace físico, quanto para um conjunto de enlaces físicos.

3.1. Definição de estruturas de representação

O substrato físico pode ser representado por um grafo não direcional $G_s = (N_s, L_s)$. Este grafo é composto por um conjunto de nodos físicos N_s conectados por meio de um conjunto de enlaces físicos L_s . Cada $n_s \in N_s$ tem uma capacidade limitada de CPU, assim como, cada $l_s \in L_s$ tem uma capacidade limitada de largura de banda.

Uma requisição de criação de redes virtuais de sincronia híbrida, HSVNRequest, é composta por um conjunto de redes virtuais, onde cada rede virtual de sincronia híbrida pode ser representada por um grafo não direcional $G_v^k = (N_v^k, L_v^k)$, onde, N_v^k é um conjunto de nodos virtuais conectados por enlaces virtuais contidos no conjunto L_v^k . O subscrito v indica se tratar de uma rede virtual e o sobrescrito k é o número identificador da rede virtual.

A sincronia dos recursos virtuais pode ser verificada pela função booleana $\text{Sync}(n_v^k)$ para os nodos virtuais e $\text{Sync}(l_v^k)$ para os enlaces virtuais, ambas funções retornam *true* para recursos síncronos e *false* para recursos assíncronos.

3.2. Heurística HSVN-WorstFit

Em outros artigos, como em [Liu et al. 2011] e [Li et al. 2013] é adotado um algoritmo base para que seja possível efetuar comparações com os algoritmos propostos. Este algoritmo base é composto por duas etapas. Na primeira etapa ocorre o mapeamento dos nodos virtuais utilizando o algoritmo clássico Worst Fit, sendo considerado o custo do nodo virtual e a capacidade de CPU dos nodos físicos. Na segunda etapa é efetuado o mapeamento dos enlaces virtuais utilizando o algoritmo do caminho mais curto de Dijkstra, alocando cada enlace virtual em um caminho composto por enlaces físicos com largura de banda disponível suficiente para hospedar o enlace virtual. Esse algoritmo heurístico foi adaptado para suportar RVSH. Um dos principais objetivos desse algoritmo é efetuar um mapeamento balanceado, por meio de uma distribuição mais igualitária dos nodos virtuais, evitando o sobrecarregamento dos nodos físicos.

No algoritmo base de RVSH, HSVN-WorstFit (*Algorithm* 1), é dado como parâmetro de entrada um substrato físico com sincronia parcial $G_s = (N_s, L_s)$ e um conjunto de requisições de criação de redes virtuais de sincronia híbrida HSVNRequest. Para cada $G_v^k \in HSVNRequest$, é chamado o procedimento MapNodesWorstFit(N_v^k, G_s), demonstrado no algoritmo MapNodesWorstFit (*Algorithm* 2).

Algorithm 1 HSVN-WorstFit($G_s, HSVNRequest$)

```

1: for all  $G_v^k = (N_v^k, L_v^k) \in HSVNRequest$  do
2:   if MapNodesWorstFit( $N_v^k, G_s$ ) == rejected then
3:     FreePhysicalResources( $G_v^k, G_s$ );
4:   else
5:     if MapLinks( $L_v^k, G_s$ ) == rejected then
6:       FreePhysicalResources( $G_v^k, G_s$ );
7:   end for

```

Nessa primeira etapa de mapeamento, os nodos virtuais serão alocados de acordo com sua ordem de chegada. O algoritmo MapNodesWorstFit recebe como entrada um conjunto de nodos virtuais de sincronia híbrida e o grafo $G_s = (N_s, L_s)$ que representa o substrato físico. Para cada $n_v^k \in N_v^k$, é verificado a sincronia do mesmo. Caso o nodo virtual seja síncrono, é chamado o procedimento FindSyncWorstFit(n_v^k, G_s), o qual retornará o nodo físico com maior capacidade de CPU disponível, que não tenha hospedado nenhum nodo virtual da mesma rede virtual e que tenha capacidade de CPU suficiente para hospedar n_v^k . Quando for possível localizar um recurso físico elegível, é efetuada sua alocação por meio do procedimento AllocateNode($n_v^k, physicalNode$), o qual é responsável também por subtrair o custo de n_v^k da capacidade de CPU do nodo hospedeiro. Caso não haja um nodo físico que se encaixe nesses requisitos, será retornado a mensagem “*rejected*”.

Quando o nodo virtual for assíncrono, é chamado o procedimento FindAsyncWorstFit(n_v^k, G_s), esse procedimento é semelhante ao FindSyncWorstFit(n_v^k, G_s), porém, nesse caso, será procurado um recurso físico assíncrono. Caso não haja um recurso físico elegível, é chamado o procedimento FindSyncWorstFit(n_v^k, G_s), pois como citado anteriormente, um nodo virtual síncrono pode ser alocado em um nodo físico síncrono. Caso ambos procedimentos não encontrem um candidato elegível, será retornado a mensagem “*rejected*”.

Algorithm 2 MapNodesWorstFit(N_v^k, G_s)

```
1: for all  $n_v^k \in N_v^k$  do
2:    $physicalNode \leftarrow null$ ;
3:   if Sync( $n_v^k$ ) == true then
4:      $physicalNode \leftarrow \mathbf{FindSyncWorstFit}(n_v^k, G_s)$ ;
5:   else
6:      $physicalNode \leftarrow \mathbf{FindAsyncWorstFit}(n_v^k, G_s)$ ;
7:     if  $physicalNode == null$  then
8:        $physicalNode \leftarrow \mathbf{FindSyncWorstFit}(n_v^k, G_s)$ ;
9:     if  $physicalNode == null$  then
10:      return rejected;
11:    else
12:       $\mathbf{AllocateNode}(n_v^k, physicalNode)$ ;
13: end for
14: return accepted;
```

Quando o algoritmo base de RVSH recebe a mensagem “*rejected*” do procedimento MapNodesWorstFit(N_v^k, G_s), significa que não foi possível efetuar o mapeamento de todos os nodos da rede virtual, sendo assim, é chamado o procedimento FreePhysicalResources(G_k^v, G_s). Esse procedimento desaloca os recursos que foram reservados para o mapeamento dessa rede virtual. Entretanto, quando a mensagem recebida for “*accepted*”, será iniciado a segunda etapa do mapeamento pela chamada do procedimento MapLinks(L_v^k, G_s) (Algorithm 3).

O algoritmo MapLinks tem como função mapear os enlaces da rede virtual, esse algoritmo tem como estrada o substrato físico, $G_s = (N_s, L_s)$, um conjunto de nodos virtuais com seus respectivos hospedeiros e um conjunto de enlaces virtuais L_v^k . Para cada enlace é verificado se o mesmo é síncrono ou assíncrono. Quando o enlace for síncrono, será chamada a função FindSyncPath(l_v^k, G_s). Essa retorna o menor caminho físico síncrono, que liga os nodos físicos que hospedam os nodos virtuais de origem e destino desse enlace, com largura de banda disponível suficiente para hospedar o enlace virtual.

Quando o enlace for assíncrono, será chamada a função FindPath(l_v^k, G_s), essa função se difere da função FindSyncPath(l_v^k, G_s), pois, encontra o caminho mais curto, podendo ser esse caminho composto por enlaces físicos tanto síncronos quanto assíncronos. Entretanto, enlaces síncronos possuem um valor maior, sendo assim, o caminho retornado por essa função só vai conter enlaces físicos síncronos caso não haja um caminho assíncrono, ou o caminho assíncrono seja composto por um número de enlaces assíncrono significativamente maior que de enlaces síncronos, tornando-se mais caro.

Após ser localizado um caminho válido, o mesmo será mapeado, sendo seu custo subtraído dos enlaces físicos hospedeiros através da função AllocatePath($l_v^k, path, G_s$). Caso não haja um caminho válido, l_v^k não será mapeado e será retornado a mensagem “*rejected*”.

Quando o algoritmo base de HSVN recebe a mensagem “*rejected*” do procedimento MapLinks(L_s^k, G_s), significa que não foi possível efetuar o mapeamento de todos os enlaces da rede virtual, sendo assim, é chamado o procedimento

Algorithm 3 MapLinks(L_v^k, G_s)

```
1: for all  $l_v \in L_v^k$  do
2:    $path \leftarrow null$ ;
3:   if Sync( $l_v^k$ ) == true then
4:      $path \leftarrow \mathbf{FindSyncPath}(l_v^k, G_s)$ ;
5:   else
6:      $path \leftarrow \mathbf{FindPath}(l_v^k, G_s)$ ;
7:   if  $path == null$  then
8:     return rejected;
9:   else
10:     $\mathbf{AllocatePath}(l_v^k, path, G_s)$ ;
11: end for
12: return accepted;
```

FreePhysicalResources(G_k^v, G_s), esse procedimento desaloca os recursos que foram reservados para o mapeamento dessa rede virtual.

3.3. Heurística HSVN-BestFit

O efeito do algoritmo Best Fit é a concentração dos nodos virtuais alocados em uma quantidade menor de nodos físicos, sendo assim, esse algoritmo também foi adaptado para mapear RVSH. Na heurística HSVN-BestFit, foi trocada a chamada da função MapNodesWorstFit(N_v^k, G_s) para MapNodesBestFit(N_v^k, G_s), linha 2 do *Algorithm 1*. Esse procedimento é similar ao algoritmo MapNodesWorstFit (*Algorithm 2*), porém utiliza a função FindSyncBestFit(n_v^k, G_s) para buscar um nodo físico síncrono com a menor capacidade disponível, mas suficiente para efetuar o mapeamento. Também é utilizado a função FindAsyncBestFit(n_v^k, G_s) para buscar um nodo físico assíncrono com a menor capacidade disponível, mas suficiente para efetuar o mapeamento. Já o mapeamento dos enlaces utiliza o algoritmo MapLinks (*Algorithm 3*).

3.4. Heurística HSVN-Virtual Tree Topology (HSVN-VTT)

Ambos algoritmos apresentados anteriormente não consideravam a topologia ou a distância dos nodos virtuais ao efetuar o mapeamento dos mesmos, o que impacta no custo final do mapeamento (ver Seção 4), já que, nodos virtuais vizinhos na rede virtual podem ser mapeados em nodos físicos distantes, obrigando o mapeamento de um único enlace virtual em vários enlaces físicos, consumindo assim mais recursos.

Por esse motivo, foi desenvolvido a heurística HSVN-VTT, a qual considera a distância entre os nodos virtuais durante o mapeamento dos mesmos. Diversos trabalhos na literatura levam em consideração, no momento do mapeamento dos nodos virtuais para nodos físicos, não somente as demandas aos nodos mas também as demandas da topologia da rede virtual e a possibilidade de seu atendimento na topologia física. [Cheng et al. 2011] e [Liu et al. 2011] consideram a quantidade de enlaces, a largura de banda dos vários enlaces e a CPU de cada nodo, durante a escolha de mapeamento de nodos, para aumentar as chances dos nodos virtuais serem mapeados em nodos físicos, cujos enlaces físicos comportem a demanda dos enlaces virtuais do nodo mapeado em nodos físicos mais próximos, baixando o custo do mapeamento dos enlaces virtuais. Já em [Li et al. 2013], os autores, além de trabalharem com estes elementos, consideram

ainda a distância em número de enlaces intermediários entre dois nodos, para a decisão de mapeamento dos mesmos.

[Li et al. 2013] propõe um algoritmo de mapeamento de redes virtuais também dividido em duas etapas onde no início da primeira etapa é construída uma árvore de alcançabilidade com base na rede virtual. Esta árvore é utilizada para efetuar o mapeamento dos nodos. Esta mesma estratégia foi transposta para o caso de HSVN. Sendo assim, essa heurística tem como base uma parte do modelo apresentado em [Li et al. 2013].

Para cada requisição de criação de rede virtual é construída uma árvore de alcançabilidade, onde a raiz será o nodo virtual com a maior demanda de CPU, sendo seus filhos os nodos diretamente ligados à raiz ordenados de maneira decrescente por demanda de CPU. Os demais nodos são mapeados na árvore de maneira recursiva em profundidade, escolhendo o filho com maior demanda de CPU. A árvore estará completa assim que todos os nodos virtuais fizerem parte dela. A Figura 1 demonstra a árvore obtida a partir de uma requisição de rede virtual.

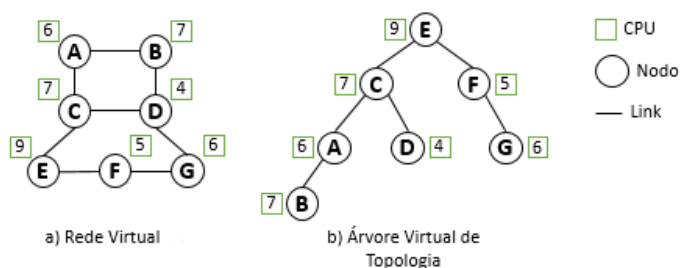


Figura 1. Construção da árvore virtual de topologia a partir de uma rede virtual.
Fonte: [Li et al. 2013]

Após a construção da árvore é iniciado o processo de mapeamento dos nodos virtuais, primeiramente é efetuado o mapeamento do nodo virtual que está na raiz da árvore. O nodo raiz é mapeado de maneira diferenciada, visto que ele não possui pai. Para evitar gargalos, esse nodo virtual é mapeado para o nodo físico com maior capacidade de CPU disponível e suficiente para hospedar o nodo virtual em questão. O requisito de sincronia também é considerado, sendo que, quando o nodo raiz for síncrono o mesmo só poderá ser mapeado em um nodo físico síncrono, mas se o nodo raiz for assíncrono, ele somente será mapeado em um nodo físico síncrono caso não haja um nodo físico assíncrono elegível.

Posteriormente ao mapeamento do nodo raiz, inicia-se o processo de mapeamento dos filhos em pré-ordem, ou seja, será mapeado os nodos virtuais seguindo um percurso em profundidade com base na árvore criada. Para cada nodo virtual a ser mapeado, é calculado o fator NF para cada nodo físico candidato a hospedar o nodo virtual, lembrando que o nodo pai já tem seu mapeamento decidido. A equação do fator de seleção do nodo (Node Selection Factor - NF) não foi alterada com relação a [Li et al. 2013], ele é calculado com base na capacidade máxima de CPU do nodo físico candidato, e na distância entre o hospedeiro do nodo virtual pai, já decidido, e o nodo físico candidato em questão, conforme a equação (1).

$$NF = MaxCPU(n_s) * (1/Dis(Host(Parent(n_v^k)), n_s)); \quad (1)$$

Seja $n_s \in N_s$ um nodo físico candidato a hospedar o nodo virtual $n_v^k \in N_v^k$;

$Dis(n_s, n'_s)$ a distância entre o nodo físico n_s e o nodo físico n'_s dada pelo número de enlaces entre n_s e n'_s ; $Host(n_v^k)$ o nodo físico que hospeda o nodo virtual n_v^k ; $Parent(n_v^k)$ o nodo virtual pai de n_v^k na árvore, a equação (1) resulta em um fator NF diretamente proporcional à capacidade de CPU do nodo físico candidato e inversamente proporcional à distância do nodo físico candidato ao nodo físico que hospeda o pai do nodo virtual em mapeamento.

O nodo virtual é mapeado no nodo físico com o maior NF, mas que tenha capacidade de CPU disponível suficiente para hospedar o nodo virtual. Assim, como no mapeamento do nodo raiz, a sincronia também é considerada. Nodos virtuais síncronos são mapeados somente em nodos síncronos, já os nodos virtuais assíncronos são mapeados em nodos síncronos somente se não houverem nodos físicos elegíveis.

A segunda etapa do mapeamento ocorre somente quando todos os nodos virtuais são mapeados com sucesso, nessa etapa inicia-se o mapeamento dos enlaces virtuais, por meio do algoritmo MapLinks (*Algorithm 3*), já descrito anteriormente. Caso não for encontrado um nodo físico ou um caminho elegível, a rede virtual é rejeitada e os recursos antes reservados para a mesma são liberados.

3.5. Heurística HSVN-Virtual Tree Topology & Bandwidth (HSVN-VTTBW)

As heurísticas apresentadas anteriormente neste artigo têm uma deficiência em comum durante a etapa de mapeamento dos nodos. Elas não consideram para cada nodo físico, sua capacidade total de largura de banda e não fazem uma análise de disponibilidade de banda por enlace físico para hospedar um determinado nodo virtual e as demandas de cada um de seus enlaces virtuais. Com isso, uma maior taxa de rejeição acontece na fase de mapeamento de enlaces, já que o mapeamento dos nodos pode ter considerado nodos físicos com enlaces não capazes de suportar as demandas dos respectivos enlaces virtuais.

Com o objetivo de aumentar a taxa de aceitação do algoritmo heurístico anterior, foi incluído neste algoritmo uma nova regra para que um nodo físico seja considerado elegível. Essa regra define que para que um nodo físico seja elegível para mapear um nodo virtual, a largura de banda disponível nos enlaces deste nodo físico devem ser suficientes para que os enlaces do nodo virtual sejam mapeados.

4. Avaliação de Desempenho

Esta seção apresenta uma avaliação das heurísticas propostas, sendo na Subseção 4.1 descrito as ferramentas e parâmetros utilizados nos experimentos e na Subseção 4.2 a apresentação e discussão dos resultados. Os resultados são avaliados com base nos seguintes critérios: 1) taxa de aceitação, 2) utilização dos recursos síncronos, 3) custo do mapeamento dos enlaces, 4) eficiência do mapeamento e 5) tempo de execução.

4.1. Experimentos

Os dados e o ambiente de execução para os experimentos são os mesmos utilizados em [Hasan et al. 2014b], visando uma comparação mais justa dos resultados. Sendo assim, o substrato físico e as redes virtuais foram randomicamente geradas pela ferramenta BRIT[Medina] seguindo o modelo Waxman[Waxman 1988]. Os experimentos foram executados em um computador com CPU de 4 cores de 1.60 GHz, e 2 GB de memória RAM. Foram executados vinte experimentos, divididos em três grupos, A, B e C. Cada grupo com requisições de redes virtuais com tamanho total de 10, 20 e 30 de nodos respectivamente.

Tabela 1. Parâmetros de configuração dos experimentos

Propriedades dos Experimentos				
Grupo : Tamanho da Rede Virtual	A: 10 nodos B: 20 nodos C: 30 nodos			
Cenário	1	2	3	4
Nodos do Substrato Físico	25			
Banda do Substrato Físico	Uniformemente distribuído entre 1Gbps e 3Gbps			
CPU do Substrato Físico	100			
Banda das Redes Virtuais	Uniformemente distribuído entre 100Mbps e 1Gbps			
CPU das Redes Virtuais	Respectivamente 10, 15 e 25% da capacidade de CPU do substrato físico			
Sincronia do Substrato Físico	30%		100%	
Sincronia das Redes Virtuais	0%	30%	60%	x%

Em todos os experimentos, o substrato físico é composto por 25 nodos, sendo a largura de banda dos enlaces uniformemente distribuída entre 1 e 3 Gbps. Nos cenários 1, 2 e 3 o substrato físico é composto por 30% de recursos síncronos, já no cenário 4, 100% dos recursos são síncronos. As redes virtuais são compostas por 3, 4 ou 5 nodos cada, sendo que cada nodo possui respectivamente 10%, 15% e 25% de CPU. A largura de banda das redes virtuais foi uniformemente distribuída entre 100Mbps e 1Gbps. A sincronia das redes virtuais varia de acordo com o grupo, sendo 0% no cenário 1, 30% no cenário 2, 60% no cenário 3, já no cenário 4 a sincronia é referenciada como x%, pois, neste cenário, o custo de mapeamento será independente da sincronia das redes virtuais, visto que o substrato físico será totalmente síncrono.

4.2. Resultados

O primeiro fator a ser avaliado é a taxa de aceitação, a qual é obtida pela divisão entre o número de requisições de redes virtuais que puderam ser completamente mapeadas e o número total de requisições. A taxa de aceitação pode ser visualizada nos gráficos de custo de largura de banda dos grupos A, B e C (Figura 2). Para diminuir a poluição visual dos gráficos, a taxa de aceitação só é exibida quando a mesma for inferior à 100%.

As taxas de aceitação das soluções ótimas foram 100%, demonstrando que é possível efetuar o mapeamento de todas as requisições deste conjunto de experimentos. A heurística HSVN-BestFit apresentou o melhor desempenho nesse critério, obtendo taxa de aceitação inferior a 100% apenas no experimento C4. Já a heurística HSVN-WorstFit teve a pior taxa de aceitação, tendo mapeado com sucesso todas as requisições de apenas um terço dos experimentos. E embora a heurística HSVN-VTTBW tenha obtido uma taxa de aceitação superior à heurística HSVN-VTT no experimento C2, ambas tiveram taxa de aceitação inferior à 100% no grupo C de experimentos.

O segundo fator avaliado foi a utilização de recursos síncronos, pois um dos objetivos das heurísticas propostas é minimizar o consumo desse tipo de recurso. Visto que não há requisição de redes virtuais com demanda de sincronia no cenário 1 e observando os gráficos do custo de largura de banda dos grupos A, B e C (Figura 2), é possível verificar que as soluções ótimas do cenário 1 não utilizam recursos síncronos, demonstrando que não é necessário a utilização desse tipo de recurso para efetuar o mapeamento. Entretanto, todas as soluções encontradas pela heurística HSVN-WorstFit fazem utilização desnecessária de recursos síncronos, enquanto as demais heurísticas somente utilizam recursos síncronos desnecessários no grupo C, cenário 1, onde as requisições de redes são maiores.

O custo de mapeamento é dado pela soma de CPU e largura de banda utilizada para

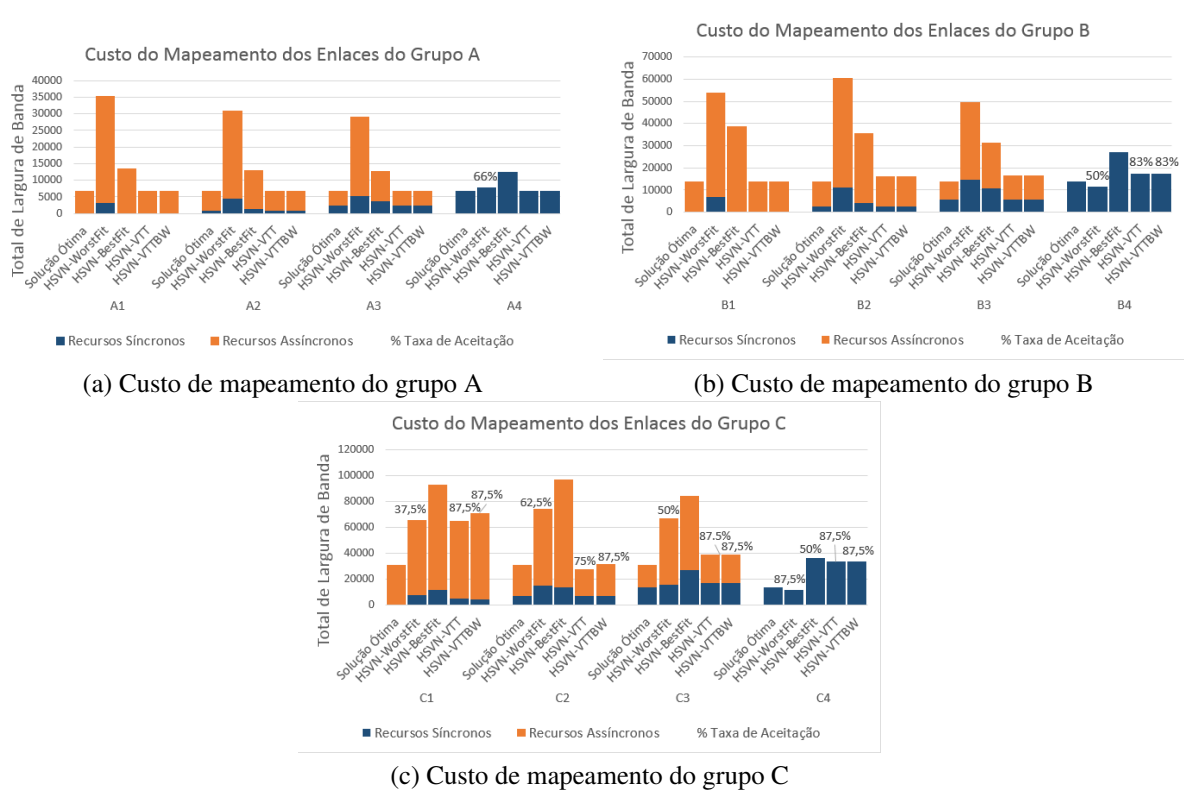


Figura 2. Custo de mapeamento dos enlaces de cada cenário por grupo

efetuar o mapeamento. Entretanto, nas heurísticas propostas, o custo de CPU sempre tem o mesmo valor da solução ótima, pois cada nodo virtual pode ser mapeado somente uma única vez e para um único nodo físico. Por esse motivo foi efetuado somente a análise do custo do mapeamento dos enlaces virtuais. Esse custo está diretamente relacionado ao comprimento dos caminhos que conectam dois nodos virtuais.

A heurística HSVN-WorstFit apresentou os custos mais altos, chegando a custar mais de 500% o custo da solução ótima no experimento A1 com 100% de taxa de aceitação. Já as heurísticas HSVN-Tree e HSVN-TreeBW apresentaram os menores custos, inclusive custos mínimos, equivalentes aos custos da solução ótima, em todos os experimentos do grupo A e no experimento B1. Enquanto a heurística HSVN-BestFit apresentou um custo superior em todos os experimentos, porém a melhor taxa de aceitação. Por isso, uma comparação entre as heurísticas com base somente nos custos de mapeamento, sem considerar a taxa de aceitação, pode ser injusto.

Para fazer uma comparação mais justa entre as heurísticas, foi calculado o fator médio de ineficiência de cada algoritmo por grupo (Figura 3). Esse fator é obtido pela divisão entre a soma de largura de banda dos recursos físicos utilizados no mapeamento e a soma da largura de banda mínima necessária para mapear as redes virtuais que foram mapeadas com sucesso. Nos experimentos deste artigo a solução ótima tem fator de ineficiência 1, que é a menor taxa de ineficiência.

O fator médio de ineficiência da heurística HSVN-BestFit aumentou conforme o aumento das redes virtuais, indicando que tal heurística não é a mais adequada para grandes demandas de requisições de redes virtuais. Já as heurísticas HSVN-VTT e HSVN-VTTBW apresentam no grupo A as mesmas taxas de ineficiências que o modelo ótimo. O

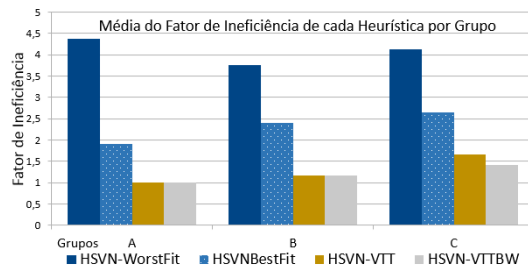


Figura 3. Média de Ineficiência de cada heurística por grupo

que significa que é possível obter-se soluções com custo igual às soluções ótimas com essa heurística. Nos grupos B e C a média de ineficiência apresenta um aumento, mas de qualquer forma ambas as heurísticas obtiveram as menores taxas de ineficiência. No grupo C a heurística HSVN-VTT obteve uma maior ineficiência do que a heurística HSVN-VTTBW, além de uma menor taxa de aceitação, o que indica que o HSVN-VTTBW tende a ter melhor desempenho para efetuar o mapeamento em redes maiores.

Como pode ser verificado na Tabela 2 do tempo de execução, o modelo de [Hasan et al. 2014b] chegou a levar mais de 58 minutos para encontrar uma solução ótima, enquanto todas as heurísticas aqui propostas encontram uma solução em menos de 1 segundo.

Tabela 2. Tempo necessário para obter a solução ótima de mapeamento, em minutos. Fonte: [Hasan et al. 2014b]

Grupo	Exp.1	Exp.2	Exp.3
A	0.85	0.40	0.31
B	37.55	1.64	10.26
C	58.34	27.12	4.47

5. Conclusão e Trabalhos Futuros

O algoritmo HSVN-WorstFit foi adotado por ser o algoritmo base de outros trabalhos similares, e para o caso de RVSH obteve os piores resultados dentre as outras heurísticas propostas. Já a heurística HSVN-BestFit teve a melhor taxa de aceitação para esse cenário de experimentos, mas o custo da largura de banda e a utilização desnecessária de recursos síncronos cresce significativamente com o aumento das redes virtuais.

As heurísticas de mapeamento HSVN-VTT e HSVN-VTTBW encontraram algumas soluções com o mesmo custo que a solução ótima, tendo o máximo de eficiência para redes virtuais pequenas. Porém apresentam uma redução na taxa de aceitação com o aumento das redes virtuais. O custo de largura de banda e a utilização desnecessária de recursos síncronos demonstrou aumentar conforme o aumento das redes virtuais, entretanto, esse aumento continua sendo inferior ao custo das demais heurísticas. A heurística HSVN-VTTBW obteve taxa de aceitação melhor em um experimento e manteve os mesmos custos nos demais experimentos, indicando que essa heurística alcançou o objetivo de aumentar a taxa de aceitação.

Como trabalho futuro, pretende-se estudar o impacto da topologia das redes físicas e virtuais no desempenho das heurísticas propostas neste trabalho. Esse impacto pode ser avaliado em termos de (i) custo de mapeamento, (ii) utilização dos recursos síncronos e

assíncronos, e (iii) taxa de aceitação. Onde [Luizelli et al. 2013] pode ser considerado um trabalho relacionado.

Agradecimentos

Este trabalho é parcialmente financiado pelo projeto FAPERGS-NPRV (Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul - Núcleo de Pesquisa em Redes Virtuais), projeto PRONEM 11/2038-1. Rasha Hasan é parcialmente financiada por projeto FAPERGS desde 2012 no contexto do seu doutoramento.

Referências

- Andersen, D. G. (2002). Theoretical approaches to node assignment.
- Anderson, T., Peterson, L., Shenker, S., and Turner, J. (2005). Overcoming the internet impasse through virtualization. *Computer*, 38(4):34–41.
- Bays, L., Oliveira, R., Buriol, L., Barcellos, M., and Gaspary, L. (2012). Security-aware optimal resource allocation for virtual network embedding. In *Network and service management (cnsm), 2012 8th international conference and 2012 workshop on systems virtualization management (svm)*.
- Chen, Y., Li, J., Wo, T., Hu, C., and Liu, W. (2010). Resilient virtual network service provision in network virtualization environments. In *Parallel and Distributed Systems (ICPADS), 2010 IEEE 16th International Conference on*, pages 51–58.
- Cheng, X., Su, S., Zhang, Z., Wang, H., Yang, F., Luo, Y., and Wang, J. (2011). Virtual network embedding through topology-aware node ranking. *SIGCOMM Comput. Commun. Rev.*, 41(2):38–47.
- Chowdhury, N. M. K. and Boutaba, R. (2010). A survey of network virtualization. *Comput. Netw.*, 54(5):862–876.
- Cristian, F. and Fetzer, C. (1999). The timed asynchronous distributed system model. *IEEE Trans. on Parallel and Distributed Systems*, 10(6).
- de Araujo Macedo, R. and Gorender, S. (2009). Perfect failure detection in the partitioned synchronous distributed system model. In *Proc. of International Conf. on Availability, Reliability and Security (ARES)*, pages 273–280.
- Feamster, N., Gao, L., and Rexford, J. (2007). How to lease the internet in your spare time. *SIGCOMM Comput. Commun. Rev.*, 37(1):61–64.
- Fischer, A., Botero, J., Till Beck, M., de Meer, H., and Hesselbach, X. (2013). Virtual network embedding: A survey. *Communications Surveys Tutorials, IEEE*, 15(4):1888–1906.
- Hasan, R., Machado Mendizabal, O., Reis De Oliveira, R., and Dotti, F. (2014a). A study on substrate network synchrony demands to support hybrid synchrony virtual networks. In *Computer Networks and Distributed Systems (SBRC), 2014 Brazilian Symposium on*, pages 344–352.
- Hasan, R., Mendizabal, O., and Dotti, F. (2014b). Hybrid synchrony virtual networks: Definition and embedding. In *ICN 2014, The Thirteenth International Conference on Networks*.

- Li, X., Guo, C., Wang, H., Li, Z., and Yang, Z. (2013). A constraint optimization based virtual network mapping method. In *2012 International Conference on Graphic and Image Processing*, pages 87683X–87683X. International Society for Optics and Photonics.
- Liu, J., Huang, T., Chen, J.-y., and Liu, Y.-j. (2011). A new algorithm based on the proximity principle for the virtual network embedding problem. *Journal of Zhejiang University SCIENCE C*, 12(11):910–918.
- Lu, J. and Turner, J. (2006). Efficient Mapping of Virtual Networks onto a Shared Substrate. Technical report, Washington University in St. Louis.
- Luizelli, M. C., Bays, L. R., Buriol, L. S., Barcellos, M. P., and Gasparly, L. P. (2013). Characterizing the impact of network substrate topologies on virtual network embedding. In *In Proc. to the 9th CNSM and Workshops (International Conference on Network and Service Management)*, pages 42–50.
- Medina, A.; Lakhina, A. M. I. B. J. Brite: Boston university representative internet topology generator.
- Oliveira, R., Marcon, D., Bays, L., Neves, M., Buriol, L., Gasparly, L., and Barcellos, M. (2013). No more backups: Toward efficient embedding of survivable virtual networks. In *Communications (ICC), 2013 IEEE International Conference on*, pages 2128–2132.
- Rahman, M. and Boutaba, R. (2013). Svne: Survivable virtual network embedding algorithms for network virtualization. *Network and Service Management, IEEE Transactions on*, 10(2):105–118.
- Schneider, F. B. (1993). Distributed systems (2nd ed.). chapter What good are models and what models are good? ACM Press/Addison-Wesley Publishing Co.
- Turner, J. and Taylor, D. (2005). Diversifying the internet. In *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, volume 2, pages 6 pp.–760.
- Veríssimo, P. E. (2006). Travelling through wormholes: a new look at distributed systems models. *ACM SIGACT News*, 37(1).
- Waxman, B. M. (1988). Routing of multipoint connections. *Selected Areas in Communications*, 6(9).
- Yu, H., Anand, V., Qiao, C., and Sun, G. (2011). Cost efficient design of survivable virtual infrastructure to recover from facility node failures. In *Communications (ICC), 2011 IEEE International Conference on*.
- Yu, M., Yi, Y., Rexford, J., and Chiang, M. (2008). Rethinking virtual network embedding: Substrate support for path splitting and migration. *SIGCOMM Comput. Commun. Rev.*, 38(2):17–29.
- Zhang, M., Wu, C., Jiang, M., and Yang, Q. (2010). Mapping multicast service-oriented virtual networks with delay and delay variation constraints. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*.
- Zhu, Y. and Ammar, M. (2006). Algorithms for assigning substrate network resources to virtual network components. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12.