

Mapeamento de Redes Virtuais em Substratos de Rede

Gustavo P. Alkmim, Daniel M. Batista, Nelson L. S. da Fonseca

¹Instituto de Computação

Universidade Estadual de Campinas (UNICAMP) – Campinas, SP – Brasil

{alkmim,daniel}@lrc.ic.unicamp.br, nfonseca@ic.unicamp.br

Abstract. *Network virtualization is a promising technique for building the Future Internet since it allows the introduction of new functionalities at network elements at low costs. To this aim, an open issue is the problem of efficient mapping of virtual network elements to those of the physical (real) network (also called network substrate). Such type of problem is NP-hard and existing solutions ignore a series of real network characteristics in order to solve the problem in reasonable time frames. This paper introduces two algorithm for the problem of mapping virtual networks on network substrates considering a set of network parameters which has not been considered before and which makes the solution more realistic.*

Resumo. *A virtualização de redes é uma técnica promissora para a Internet do futuro, pois facilita a adição de novas funcionalidades nos elementos de rede e diminui os custos das organizações. Uma das questões que ainda precisa ser investigada para a implantação eficiente desta tecnologia é a alocação de recursos físicos para as redes virtuais. Por se tratar de um problema NP-Difícil, os algoritmos propostos, até o momento, desconsideram vários parâmetros, a fim de obter soluções em tempo viável. Este artigo apresenta dois novos algoritmos para o problema de mapeamento de redes virtuais em substratos de rede. Os algoritmos minimizam a utilização dos recursos e consideram diversos parâmetros negligenciados por outros trabalhos na literatura. Experimentos mostram que os algoritmos encontram soluções em tempo viável para diversos cenários de requisição de redes virtuais.*

1. Introdução

Uma das principais características da arquitetura Internet, que facilitou a sua proliferação em escala global, é seu aspecto generalista e minimalista. Dessa forma, a pilha de protocolos TCP/IP pôde ser implementada sobre diferentes tipos de tecnologias e o núcleo da rede foi construído de modo a ser o mais simples possível. No entanto, a diversificação das aplicações e o intenso uso da Internet como infraestrutura global de comunicação acarretou em uma série de adições de protocolos e mecanismos à sua arquitetura, a fim de que se pudesse prover funcionalidades inexistentes na pilha TCP/IP original. A impossibilidade de alterações no núcleo da Internet para inclusão de novas funcionalidades, embora mantenha o núcleo simples, dificulta o desenvolvimento de novas aplicações, motivo pelo qual o termo “ossificação da Internet” é utilizado para representar esse fato.

Para superar essas restrições, novas arquiteturas e mecanismos vêm sendo propostos para promover a evolução da Internet do futuro [Trossen 2009] [Feamster et al. 2007]

[Zhu et al. 2008b] [Zhu et al. 2008a] [He et al. 2008] [Bless et al. 2008]. Várias destas soluções baseiam-se na virtualização da rede. Através da virtualização é possível definir redes virtuais, compostas de roteadores e enlaces virtuais, que fazem uso de roteadores e enlaces da rede física; conjunto de recursos, normalmente, chamado de substrato da rede. A virtualização de redes permite a coexistência de diferentes pilhas e arquiteturas de redes no mesmo núcleo da Internet, sem a necessidade de modificá-lo e sem restringir as características destes protocolos e arquiteturas.

Dentre as diversas questões em aberto na área de virtualização de redes, uma das mais importantes é a busca por mapeamentos eficientes de redes virtuais nos substratos da rede física [Chowdhury et al. 2009] [Yu et al. 2008]. O mapeamento consiste em determinar a alocação de roteadores e enlaces da rede física para os roteadores e enlaces de uma rede virtual. No entanto, mesmo tendo-se o conhecimento prévio de todas as requisições de redes virtuais, o mapeamento ótimo é um problema NP-difícil [Houidi et al. 2008], já que ele pode ser reduzido ao Problema de Separação de Multi-caminhos (*Multipath Separator Problem*) [Andersen 2002], que é NP difícil.

Diversas soluções têm sido propostas para realizar o mapeamento de redes virtuais [Yu et al. 2008] [Chowdhury et al. 2009] [Lu and Turner 2006] [Zhu and Ammar 2006] porém, a maioria destas propostas assume hipóteses restritivas para tornar o problema tratável. Algumas das hipóteses são: (i) considerar que todas as requisições de estabelecimento de redes virtuais são conhecidas antecipadamente [Lu and Turner 2006] [Zhu and Ammar 2006], (ii) assumir que o substrato tem capacidade infinita [Zhu and Ammar 2006] [Fan and Ammar 2006] e (iii) particularizar a topologia da rede virtual [Lu and Turner 2006].

Este artigo apresenta soluções que visam tratar os pontos negligenciados pelos trabalhos anteriores no mapeamento de redes virtuais. As novas propostas para mapeamento de redes deste trabalho consideram a existência de 3 classes de provedores: os provedores de infraestrutura, os provedores de conectividade e os provedores de serviços [Zhu et al. 2008b]. Os provedores de infraestrutura são os responsáveis pela rede física (roteadores, cabeamento, etc), na qual serão instanciadas as redes virtuais. Os provedores de serviços são os responsáveis por fornecer os serviços da Internet para os usuários finais. São estes provedores que solicitam as redes virtuais, para suportar seus serviços. Os provedores de conectividade são responsáveis por instanciar as redes virtuais requisitadas pelos provedores de serviços na infraestrutura física. É papel dos provedores de conectividade mapear as redes virtuais requisitadas pelos provedores de serviços na infraestrutura física fornecida pelos provedores de infraestrutura.

O objetivo deste trabalho é, portanto, propor uma solução eficiente para que os provedores de conectividade possam mapear redes virtuais no substrato da rede, sem restrições nas diversas camadas de rede envolvidas. Em comparação com outros trabalhos propostos na literatura, este trabalho considera um cenário mais realista e processa uma quantidade maior de parâmetros que influenciam diretamente a complexidade da solução do problema. É necessário garantir, portanto, que o tempo de execução do algoritmo de mapeamento seja viável.

Dois algoritmos de mapeamento de redes virtuais em substratos de rede são propostos baseados em uma formulação de programação inteira 0-1 (ILP). O objetivo é mi-

nimizar a quantidade total de largura de banda alocada nos enlaces físicos para uma rede virtual. Ao se minimizar a quantidade de largura de banda alocada para uma rede virtual, tenta-se maximizar a quantidade de recursos disponíveis para as requisições seguintes. A diferença entre os dois algoritmos está no fato de um deles encontrar a solução exata do ILP, enquanto o outro utiliza técnicas de relaxação para diminuir o tempo de execução.

Experimentos mostram que a abordagem introduzida encontra soluções em tempo viável para diversos cenários de requisição de redes virtuais. Os algoritmos fornecem, também, resultados promissores considerando-se o bloqueio e a qualidade da solução.

O artigo está organizado da seguinte forma: a Seção 2 resume os trabalhos relacionados. A Seção 3 apresenta os dois algoritmos propostos. A avaliação de desempenho dos algoritmos é apresentada na Seção 4 e as conclusões e trabalhos futuros são apresentados na Seção 5.

2. Trabalhos Relacionados

Esta seção resume trabalhos relacionados sobre o mapeamento de redes virtuais em substratos de redes. Definições importantes para o entendimento do restante desta seção são a de nós de acesso (nós de borda) e a de nós de núcleo (nó do *backbone*). Os nós de acesso são as origens e os destinos dos fluxos de dados enquanto que os nós do *backbone* são responsáveis exclusivamente pelo roteamento das informações.

Em [Feamster et al. 2007], propõe-se uma arquitetura chamada Cabo na qual os provedores de infraestrutura (PI) são responsáveis pelo controle da camada da rede física e os provedores de serviços (PS) são responsáveis pelo fornecimento dos serviços da rede. Diferentemente, a nossa abordagem considera a existência de uma camada entre o PS e o PI, na qual os algoritmos de mapeamento são executados.

Em [Yu et al. 2008], são apresentadas quatro razões que tornam o problema de mapear redes virtuais tão desafiador. A primeira razão é o grande número de restrições envolvendo os nós da rede, como processamento e memória, bem como envolvendo enlaces, como largura de banda e atraso da rede. Além disso, é necessário mecanismos de controle de admissão na rede, pois os recursos são limitados e, portanto, algumas requisições podem ser potencialmente negadas. O terceiro motivo é o fato de não se conhecer previamente as requisições, e o tempo de vida das redes virtuais. A quarta razão é a diversidade de topologias existentes. Os algoritmos apresentados no presente artigo consideram essas questões colocadas por [Yu et al. 2008] como desafiadores: a disponibilidade dos recursos (roteadores e enlaces) é levada em consideração na tomada de decisões; as redes virtuais especificam os recursos necessários e o seu tempo de vida apenas no momento da requisição e as soluções não são voltadas para uma topologia de redes específica. Embora o nosso trabalho não inclua, explicitamente, a existência de um mecanismo de controle de admissão, ele pode ser facilmente integrado na arquitetura proposta, de modo que apenas as requisições aprovadas por ele sejam repassadas para os algoritmos.

Na maioria das soluções propostas na literatura [Yu et al. 2008] [Chowdhury et al. 2009] [Houidi et al. 2008], [Lu and Turner 2006], os recursos alocáveis são a capacidade dos enlaces físicos e a capacidade de processamento dos nós físicos da rede. No entanto, outros recursos que possuem forte influência na eficiência da alocação, como memória e acesso a disco, não são incluídos na formulação do

problema. Em alguns trabalhos [Padala et al. 2007] esta questão é sugerida como tema de trabalho futuro mas nenhuma solução foi proposta até o momento. A nossa proposta considera diversos parâmetros negligenciados por outros trabalhos; por exemplo, além das capacidades dos enlaces, considera a memória do roteador físico, a quantidade de elementos de processamento e o tempo de *boot* do roteador virtual.

Na abordagem em [Yu et al. 2008], o substrato físico possibilita a divisão de caminhos e a migração de caminho, o que permite que o problema seja resolvido em tempo polinomial. Diferentemente de [Yu et al. 2008], nossos algoritmos processam uma maior quantidade de características do problema, como a presença de imagens que devem ser utilizadas para instanciar os roteadores virtuais.

Em diversos trabalhos na literatura [Yu et al. 2008], separa-se as etapas de mapeamento de enlaces e de mapeamento de nós. Com o objetivo de introduzir uma correlação entre estas duas etapas, dois algoritmos foram propostos em [Chowdhury et al. 2009]: o D-ViNE (*Deterministic VN Embedding*) e o R-ViNE (*Randomized VN Embedding*). Nestes, faz-se um mapeamento dos nós virtuais nos nós do substrato, de forma a facilitar o mapeamento dos enlaces virtuais. Estes algoritmos não consideram todos os parâmetros que nossos algoritmos consideram, como por exemplo a presença de imagens virtuais e o atraso nos enlaces da rede. Nos nossos algoritmos, o mapeamento de nós e de enlaces são também realizados de forma combinada. Desse modo, a probabilidade de soluções serem encontradas aumenta.

Em [Lu and Turner 2006], caracteriza-se uma rede virtual somente pela demanda de transporte entre para origem-destino considerando apenas restrições de tráfego entre os nós definido por um conjunto de restrições genéricas.

Os trabalhos existentes na literatura levam em consideração um conjunto restrito de características dos nós ou dos enlaces da rede. Além disso, não tratam de questões como a presença de imagens na rede física que devem ser utilizadas para instanciar os roteadores virtuais. Na Seção 3, os algoritmos introduzidos incluem todas estas questões, que são importantes para se ter um algoritmo de mapeamento realista.

3. Algoritmos Propostos

Propõem-se dois algoritmos para realizar o mapeamento de redes virtuais em substratos de redes. As principais características dos algoritmos são a inclusão de parâmetros essenciais para garantir um mapeamento realista, bem como a existência de imagens de roteadores na rede utilizadas para instanciar as redes virtuais, dado que a instanciação de um roteador virtual exige a transferência da imagem do *software* de roteador de um repositório até o roteador físico e a inicialização dessa imagem.

Os dois algoritmos são baseados em uma formulação de programação inteira. Para a formulação do problema, considera-se que a rede física é representada por um grafo (N, F) , onde N é o conjunto dos roteadores físicos e F é o conjunto dos enlaces físicos. Analogamente, a rede virtual é representada por um grafo (M, V) , onde M é o conjunto dos roteadores virtuais e V o conjunto dos enlaces virtuais. As entradas para o problema são:

- $N \in \mathbb{Z}$ - Conjunto de roteadores da rede física.
- $F \in \mathbb{Z}$ - Conjunto de enlaces da rede física.

- $M \in \mathbb{Z}$ - Conjunto de roteadores da rede virtual.
- $V \in \mathbb{Z}$ - Conjunto de enlaces da rede virtual.
- $I \in \mathbb{Z}$ - Conjunto de imagens.
- $A \in \mathbb{Z}$ - Quantidade de núcleos disponíveis nos roteadores da rede física.
- $P \in \mathbb{Z}$ - Quantidade de núcleos requisitados pelos roteadores da rede virtual.
- $C \in \mathbb{R}$ - Largura de banda disponível nos enlaces da rede física.
- $Q \in \mathbb{R}$ - Largura de banda requisitada pelos enlaces da rede virtual.
- $D \in \mathbb{R}$ - Atraso nos enlaces da rede física.
- $R \in \mathbb{R}$ - Atraso máximo permitido nos enlaces da rede virtual.
- $L_{n,m} \in \{0, 1\}$ - Define se o roteador virtual m pode ser alocado no roteador físico n . Assume o valor 1 caso a alocação seja permitida e 0 caso contrário.
- $R_{n,i} \in \{0, 1\}$ - Assume o valor 1 caso a imagem i estiver localizada em um repositório conectado diretamente ao roteador físico n , caso contrário assume o valor 0.
- $E_{m,i} \in \{0, 1\}$ - Assume o valor 1 caso a imagem i contiver todos os requisitos necessários pelo roteador virtual m , caso contrário assume o valor 0.
- $B \in \mathbb{R}$ - Quantidade de memória de armazenamento disponível nos roteadores da rede física.
- $G \in \mathbb{R}$ - Quantidade de memória de armazenamento necessária para carregar as imagens.
- $S \in \mathbb{R}$ - Tempo máximo requisitado para a instanciação da rede virtual.
- $T_{n,i} \in \mathbb{R}$ - Tempo necessário para a imagem i ser inicializada no roteador físico n .

A inclusão de $I, D, R, L_{n,m}, R_{n,i}, E_{m,i}, B, G, S$ e $T_{n,i}$ em uma mesma formulação é o diferencial deste trabalho em relação aos demais encontrados na literatura.

A função objetivo minimiza a quantidade total de largura de banda alocada nos enlaces físicos para uma rede virtual, o que implica em maximizar a largura de banda disponível para as próximas requisições, aumentando, assim, a chance de se conseguir recursos para efetuar as próximas requisições. Isso significa que o algoritmo retornará a solução em que a banda alocada seja a menor possível, sem ignorar os requisitos das redes virtuais.

A solução do problema é dada pelos valores das seguintes variáveis:

- $X_{n,m,i}$ - Retorna 1 caso o roteador virtual m tenha sido alocado no roteador físico n utilizando a imagem i . Caso contrário, retorna 0. Esta variável retorna o mapeamento dos nós virtuais nos roteadores físicos e a imagem que foi utilizada para cada roteador virtual. Para facilitar o entendimento, cada tupla (n, m, i) pode ser vista como uma possível alocação para o roteador virtual m .
- $Y_{u,v,w}$ - Retorna 1 caso o enlace físico (u, v) seja alocado para o enlace virtual w . Caso contrário, retorna 0. Pode-se a partir dos valores dessa variável, determinar-se o caminho na rede física que foi alocado para cada enlace virtual.
- $Z_{u,v,n,m,i}$ - Retorna 1 caso o enlace físico (u, v) seja utilizado para transferir para n a imagem i , utilizada pelo roteador virtual m , que foi alocado no roteador físico n . Caso contrário, retorna 0. Esta variável retorna quais enlaces físicos foram utilizados para transferir cada imagem para o roteador físico no qual será instanciada. A partir dos valores obtidos por esta variável, tem-se o caminho na rede física pelo qual a imagem foi transferida até chegar no roteador físico destino.

O problema formulado é apresentado a seguir:

$$\text{Minimize } \sum_{u \in N} \sum_{v \in N} \sum_{w \in V} Y_{u,v,w} \times Q(w)$$

sujeito a

$$\sum_{n \in N} \sum_{i \in I} X_{n,m,i} = 1 \quad \forall m \in M \quad (R1)$$

$$\sum_{m \in M} \sum_{i \in I} P(m) \times X_{n,m,i} \leq A(n) \quad \forall n \in N \quad (R2)$$

$$X_{n,m,i} = 0 \quad \forall n \in N, \forall m \in M, \quad (R3)$$

$$\forall i \in I | L_{n,m} = 0 \text{ ou } E_{m,i} = 0$$

$$\sum_{w' \in V} Y_{u,v,w'} \times Q(w') \leq C(w) \quad \forall w = (u, v) \in F \quad (R4)$$

$$\sum_{u \in N} \sum_{v \in N} Y_{u,v,w} \times D(u, v) \leq R(w) \quad \forall w \in V, (u, v) \in F \quad (R5)$$

$$\sum_{m \in M} \sum_{i \in I} X_{n,m,i} \times G(i) \leq B(n) \quad \forall n \in N \quad (R6)$$

$$\sum_{u \in N} \sum_{v \in N} \sum_{n \in N} \sum_{i \in I} Z_{u,v,n,m,i} \times D(u, v) + \quad \forall m \in M, (u, v) \in F \quad (R7)$$

$$\sum_{u \in N} \sum_{v \in N} \sum_{n \in N} \sum_{i \in I} \frac{Z_{u,v,n,m,i} \times G(i)}{C(u, v)} +$$

$$\sum_{n \in N} \sum_{i \in I} X_{n,m,i} \times T_{n,i} \leq S$$

$$Z_{u,v,n,m,i} \leq X_{n,m,i} \quad \forall u, \forall v, \forall n \in N, \forall m \in M, \forall i \in I \quad (R8)$$

$$Y_{u,v,w}, Z_{u,v,n,m,i} = 0 \quad \forall u, \forall v, \forall n \in N, \forall w \in V, \quad (R9)$$

$$\forall m \in M, \forall i \in I | (u, v) \notin F$$

$$\sum_{f \in N} Y_{n,f,w} - \sum_{f \in N} Y_{f,n,w} = \quad \forall w = (a, b) \in V, \forall n \in N \quad (R10)$$

$$\sum_{i \in I} X_{n,a,i} - \sum_{i \in I} X_{n,b,i}$$

$$\sum_{j \in N} Z_{u,j,n,m,i} - \sum_{j \in N} Z_{j,u,n,m,i} = \quad \forall m \in M, \forall i \in I, \forall n, u \in N, \quad (R11)$$

$$X_{n,m,i} \times R_{u,i} - X_{n,m,i} \times \left(1 - \left\lceil \frac{|u-n|}{\alpha} \right\rceil\right) \quad \alpha = |N|$$

$$X_{n,m,i}, Y_{u,v,w}, Z_{u,v,n,m,i} \in \{0, 1\} \quad \forall u, \forall v, \forall n \in N, \forall m \in M \quad (R12)$$

$$\forall w \in V, \forall i \in I$$

As restrições em (R1) garantem que um roteador virtual seja alocado em um roteador físico e que uma imagem seja utilizada por ele. As restrições em (R8) impedem que a imagem i seja transferida para o roteador físico n , caso ela não seja utilizada. As restrições em (R10) garantem que o conjunto de enlaces físicos alocados para um dado enlace virtual $w = (a, b)$ é um caminho válido na rede física. Para isto, analisa-se o grau de entrada e de saída de cada roteador n da rede física, representados, respectivamente, por $\sum_{f \in N} Y_{f,n,w}$ e $\sum_{f \in N} Y_{n,f,w}$, considerando-se somente os enlaces físicos alocados para o enlace virtual.

As restrições em (R11) garantem que o conjunto de enlaces físicos selecionados para transferir a imagem i , utilizada pelo roteador virtual m , para o roteador físico n é um caminho válido na rede física. Esta restrição é análoga à Restrição 10. Assim como naquela restrição, analisa-se o grau de entrada e de saída de cada roteador u da rede física, representados, respectivamente, por $\sum_{j \in N} Z_{j,u,n,m,i}$ e $\sum_{j \in N} Z_{u,j,n,m,i}$, considerando-se somente os enlaces físicos alocados para a transferência da imagem i . As restrições em (R9) garantem que se o enlace físico (u, v) não existir, então nenhum enlace virtual pode ser alocado utilizando (u, v) . As restrições em (R12) garantem que as variáveis X , Y e Z são binárias.

As restrições (R2) e (R6) referem-se a restrições de limitação dos nós físicos. As restrições em (R2) garantem que o número de núcleos de cada roteador físico é suficiente para atender o requisito de número de núcleos de cada roteador virtual alocado nele. Cada roteador físico deve possuir mais núcleos do que a soma de todos os núcleos de todos os roteadores virtuais que foram alocados nele. As restrições em (R6) garantem que a quantidade de memória de armazenamento disponível em cada roteador físico será suficiente para armazenar as imagens de todos os roteadores virtuais que foram alocados nele.

As restrições em (R3) garantem que os roteadores virtuais só serão instanciados com imagens que atendam todos os seus requisitos de software e em nós físicos que possuam as características permitidas pelo cliente que requisitou a rede virtual.

As restrições (R4) e (R5) referem-se às limitações dos enlaces físicos. As restrições em (R4) garantem que a largura de banda disponível nos enlaces da rede física é suficiente para atender o requisito de largura de banda dos enlaces virtuais. As restrições em (R5) garantem que o atraso total em um caminho físico alocado para um enlace virtual é menor ou igual ao atraso máximo permitido nos enlaces virtuais alocados.

As restrições em (R7) garantem que o tempo total gasto para instanciar a rede virtual seja menor que o tempo requisitado pela rede virtual. O tempo necessário para alocar cada roteador virtual é dado pela soma dos seguintes fatores: atraso para transferir a imagem, tempo gasto para transferir a imagem e o tempo gasto para alocar a imagem no roteador físico. A formulação proposta considera que mais de uma imagem pode ser transmitida pelo mesmo enlace físico simultaneamente.

O primeiro algoritmo proposto (algoritmo ótimo) busca a solução exata do problema. Já o segundo algoritmo proposto (algoritmo relaxado) emprega heurísticas para diminuir o tempo de execução do algoritmo. A heurística empregada é a de relaxação. Nesse caso, as variáveis binárias são consideradas como variáveis reais.

Os algoritmos propostos utilizam o CPLEX [IBM 2010] para resolver o ILP proposto. Algumas etapas realizadas pelo CPLEX ao resolver um problema são: (1) o pré-processamento (CPLEX Presolve), que simplifica e reduz o tamanho do problema, (2) o descobrimento (CPLEX Probe), que analisa as implicações lógicas de se fixar os valores (1 ou 0) das variáveis do problema e (3) a busca pela solução do problema utilizando o método *Branch-and-cut* [IBM 2010]. Na etapa (3) é criada uma árvore de busca de soluções, sendo que no nó raiz são determinadas soluções iniciais para o problema através de heurísticas, que incluem etapas de relaxação [IBM 2010]. A diferença entre os dois algoritmos propostos é que um algoritmo procura pela solução ótima do problema através de todos os nós da árvore de busca e o outro termina sua execução no nó raiz da árvore de busca.

4. Avaliação de Desempenho

Experimentos de simulação foram realizados com o objetivo de avaliar 3 métricas: tempo de execução do algoritmo, quantidade de banda passante do substrato que foi alocada para a requisição de rede virtual e taxa de bloqueio das requisições. Logo, a análise dos resultados busca avaliar o quão úteis para aplicações reais os algoritmos propostos podem ser e quais os aspectos que precisam ser aprimorados em trabalhos futuros. Os algoritmos propostos foram avaliados e comparados em cenários estáticos, nos quais apenas uma requisição de rede virtual é feita, e em cenários dinâmicos, nos quais várias requisições de redes virtuais são feitas ao longo do tempo. Nesse último caso, a disponibilidade dos recursos do substrato varia com o passar do tempo por conta das alocações que vão sendo feitas para as redes virtuais.

Todas as simulações do cenário estático foram realizadas no sistema operacional Debian GNU/Linux Squeeze. O computador utilizado para os experimentos foi um Intel Xeon de 2,27GHz com 2 processadores de 8 núcleos cada, e 6GB de memória RAM. Os dois algoritmos foram implementados em C++ e utilizaram a biblioteca de otimização CPLEX versão 12.0.

As simulações foram realizadas em um simulador desenvolvido em C pelos autores. O simulador recebe como entrada um modelo de rede física e gera eventos de chegada de requisição de redes virtuais. De posse da descrição da rede física e da rede virtual, o simulador invoca os códigos dos algoritmos para que seja devolvida uma alocação e devolve como saída todos os dados de interesse: informação sobre se a requisição foi ou não bloqueada, tempo de execução do algoritmo e banda alocada pelo algoritmo. O mapeamento realizado pelos algoritmos também é devolvido pelo simulador para fins de depuração do seu funcionamento.

4.1. Configuração dos Experimentos

Vários cenários estáticos foram avaliados nos experimentos. Avalia-se nestes experimentos se os tempos de execução do algoritmo relaxado são realmente menores do que os do algoritmo ótimo. Em relação à qualidade das alocações, o resultado esperado é que

o algoritmo ótimo aloque menos largura de banda do que o algoritmo relaxado, mas que o ganho no tempo de execução do algoritmo relaxado seja tal que compense a perda na qualidade da alocação. O conjunto de cenários pode ser obtido através da combinação dos seguintes valores:

- Número de nós do substrato: 5, 7, 10, 12, 15, 17, 20, 22 e 25. A utilização destes valores permite uma análise do desempenho dos algoritmos em função do aumento da rede física. O objetivo com essa faixa de valores é avaliar o desempenho dos algoritmos com substratos de diversos tamanhos. Por conta de limitações do hardware utilizado nos experimentos, a quantidade de nós físicos foi limitada a 25 nós. Simulações com mais de 25 nós físicos terminavam abruptamente por falta de memória.
- Número de imagens na rede: 20% relativo ao número de nós físicos da rede.
- Quantidade de núcleos disponíveis nos roteadores da rede física: 6 núcleos. Valor baseado em roteadores disponíveis no mercado [Cisco Systems 2010b].
- Largura de banda disponível nos enlaces do substrato: uniformemente distribuído entre 1Gbps e 10Gbps. Esta faixa de valores é comum para redes físicas robustas [RNP 2010].
- Quantidade de memória de armazenamento disponível nos roteadores da rede física: 256MB. Este valor baseia-se na quantidade de memória *flash* presente em roteadores reais encontrados no mercado [Cisco Systems 2010a].
- Quantidade de memória de armazenamento necessária para carregar cada imagem: 128MB. Este valor foi baseado na quantidade de memória *flash* recomendada para o *software* definido em [Cisco Systems 2010c], um sistema operacional para roteadores.
- Quantidade de tempo necessário para uma imagem ser instanciada em um roteador físico: 10 segundos. Utilizando os roteadores atuais como referência, o sistema operacional dos mesmos demora pouco tempo para ser iniciado tendo em vista que são sistemas bastante reduzidos.
- Tempo limite para instanciar cada rede virtual: 60 segundos.
- Tipos de requisição: Foram definidas três tipos de requisições de redes virtuais: Tipo 1, Tipo 2 e Tipo 3. As requisições diferem entre si em termos da quantidade de recursos requisitada, de tal forma que as requisições do Tipo 3 requisitam mais recursos do que as do Tipo 2 que, por sua vez, requisitam menos recursos do que as do Tipo 1. A Tabela 1 descreve os requisitos de cada um dos tipos das redes virtuais utilizadas nos experimentos.

Tabela 1. Descrição dos tipos de redes virtuais.

| Tipo | # de nós virtuais | # de núcleos | Largura de banda (uniformemente distribuído) | Probabilidade de alocação de um roteador físico (Restrição de localidade) | Probabilidade de utilização de uma imagem virtual (Restrição de software) |
|------|-------------------|--------------|--|---|---|
| 1 | 5 | 2 | 100Mbps–200Mbps | 100% | 100% |
| 2 | 8 | 3 | 200Mbps–300Mbps | 100% | 100% |
| 3 | 10 | 6 | 300Mbps–400Mbps | 100% | 100% |

Espera-se que a quantidade de redes virtuais do Tipo 1 alocadas pelos algoritmos propostos seja maior do que a quantidade de alocações do Tipo 2, que por sua vez devem ser maiores do que a quantidade de alocações do Tipo 3.

A topologia do substrato da rede e das redes virtuais foram geradas aleatoriamente através da ferramenta BRITE [Medina et al. 2010], utilizando o algoritmo BA-2 [Albert and Barabási 2000]. Para a rede física, os atrasos nos enlaces da rede permitidos são os próprios valores padrão retornados pelo BRITE. Como o atraso permitido nas redes virtuais deve ser maior que o atraso dos enlaces da rede física, o atraso dos enlaces das redes virtuais foi obtido multiplicando o valor retornando pelo BRITE por um valor que é dependente do tipo da requisição de rede virtual. Para as requisições de redes virtuais do Tipo 1, o atraso é o valor retornado pelo BRITE multiplicado por 15 (o que permite, aproximadamente a utilização de 15 enlaces físicos por enlace virtual). Para as requisições do Tipo 2, o atraso é o valor retornado pelo BRITE multiplicado por 10. Para as requisições do Tipo 3, o atraso é o valor retornado pelo BRITE multiplicado por 5.

Para os cenários dinâmicos, o tempo de simulação de cada cenário foi de 10000 segundos. O intervalo de chegada e a duração das requisições foram definidos como um número aleatório exponencialmente distribuído com média de 25 e 1000 segundos, respectivamente. O substrato possui 25 nós físicos e a banda disponível em cada enlace é um valor aleatório uniformemente distribuído entre 1Gbps e 10Gbps. Da mesma forma que no cenário estático, foram definidos os três tipos de requisições, similares aos da Tabela 1. A única diferença é com relação às probabilidades referentes à restrição de localidade e à restrição de software. Ao invés de 100% para as requisições do Tipo 2 e do Tipo 3, esses valores são 80% e 60%, respectivamente. Em todos os tipos de requisições, o número de nós virtuais é um número uniformemente distribuído entre 1 e 5.

Em todos os experimentos realizados cada cenário foi simulado 5 vezes. Quando não especificado, os resultados apresentados a seguir correspondem às médias dos valores encontrados. Os resultados das diversas instâncias foram bastante próximos entre si e por isso os intervalos de confiança não são exibidos nos gráficos para facilitar a visualização. O nível de confiança considerado foi de 95%.

4.2. Resultados e Discussões

A Tabela 2 e o gráfico da Figura 1 apresentam os resultados obtidos nos cenários estáticos.

Tabela 2. Tempo de execução dos algoritmos – Cenário estático.

| Tipo de requisição | Algoritmo | Média (s) |
|--------------------|-----------|-----------|
| 1 | Ótimo | 608,11 |
| 1 | Relaxado | 46,66 |
| 2 | Ótimo | 398,44 |
| 2 | Relaxado | 104,11 |
| 3 | Ótimo | 1202,55 |
| 3 | Relaxado | 882,44 |

A Tabela 2 exhibe a média dos tempos de execução dos algoritmos para cada um dos 3 tipos de requisição. Em média o tempo de execução do algoritmo relaxado é menor do que a do algoritmo ótimo em todos os casos, conforme esperado. Para as requisições do Tipo 1, 2 e 3, o tempo de execução do algoritmo relaxado foi em média 92%, 74% e 27% menor, respectivamente.

Com relação à variação do tempo de execução em função da quantidade de nós, observou-se que eles são diretamente proporcionais. Os gráficos não são exibidos dado

limitação de espaço.

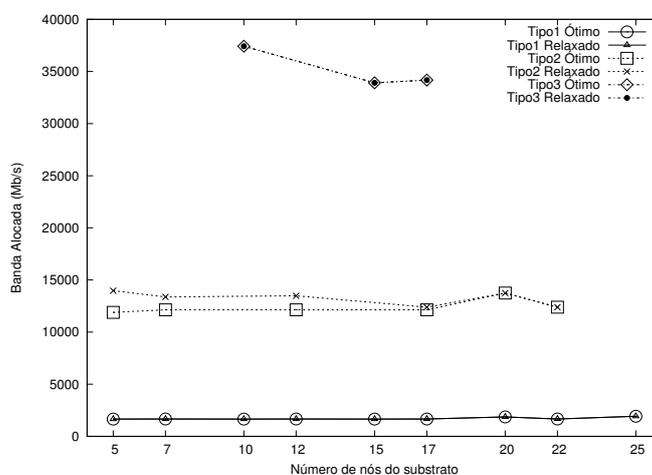


Figura 1. Banda alocada - Cenário estático.

O gráfico da Figura 1 plota a quantidade de largura de banda alocada pelos algoritmos em função do número de nós da rede física. A quantidade alocada pelo algoritmo relaxado foi muito próxima à alocada pelo algoritmo ótimo. Na maioria dos cenários, inclusive, os valores foram iguais. A maior diferença entre as quantidades alocadas pelos dois algoritmos ocorreu para a requisição do Tipo 2 quando haviam 5 nós no substrato, e foi apenas de 15%. Pode-se observar que a medida que as requisições se tornam mais rígidas, a quantidade de pontos apresentados no gráfico diminui. Isso se dá pelo fato de não haver solução para aqueles casos. Por exemplo, para as redes do Tipo 3, em apenas 3 configurações (10, 15 e 17 nós) uma solução foi encontrada.

Os gráficos das figuras 2, 3, 4 e 5 apresentam os resultados obtidos no cenário dinâmico. Os gráficos das figuras 2, 3 e 4 exibem o tempo de execução dos algoritmos para o três tipos de requisição ao longo do tempo (cada gráfico apresenta o resultado de uma única instância simulada. Os resultados de todas as instâncias, para uma mesma configuração, foram bem próximos). O tempo de execução apresentado equivale ao tempo gasto pelos algoritmos para devolver as alocações. Para requisições mais complexas, o tempo de execução do algoritmo ótimo é impraticável mas, de um modo geral, isso não ocorre com o tempo de execução do algoritmo relaxado, como pode ser visto pela curva das requisições do Tipo 3 (Figura 4). O motivo para tal é o fato do algoritmo relaxado parar sua execução no nó raiz da árvore de busca da programação inteira, enquanto o algoritmo ótimo continua realizando a busca até percorrer toda árvore ou até o tempo limite estabelecido ser atingido. Desta forma, para requisições mais complexas ou para redes físicas com um maior número de nós, o número de soluções para o problema aumenta, fazendo com que a árvore de busca aumente também. É possível observar nos gráficos das figuras 2 e 3 que os tempos de execução dos algoritmos diminuiu com o passar do tempo. Isso ocorreu pelo fato da rede estar ocupada com as requisições anteriores. Dessa forma, os algoritmos não encontram solução, por não haverem recursos disponíveis, e devolvem essa informação rapidamente.

O gráfico da Figura 5 mostra o número total de requisições bloqueadas. Da mesma forma, pode-se perceber que o bloqueio produzido dos dois algoritmos são muito seme-

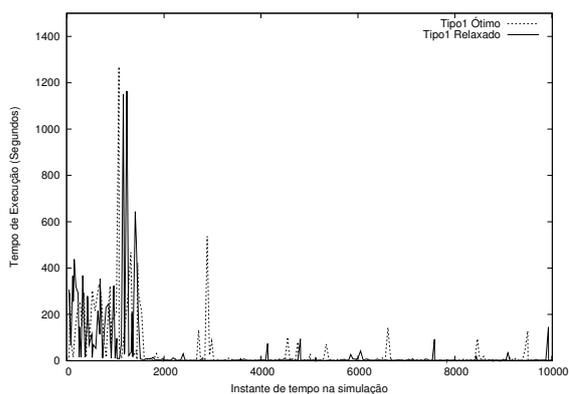


Figura 2. Tempo de execução dos algoritmos (Tipo 1) - Cenário dinâmico

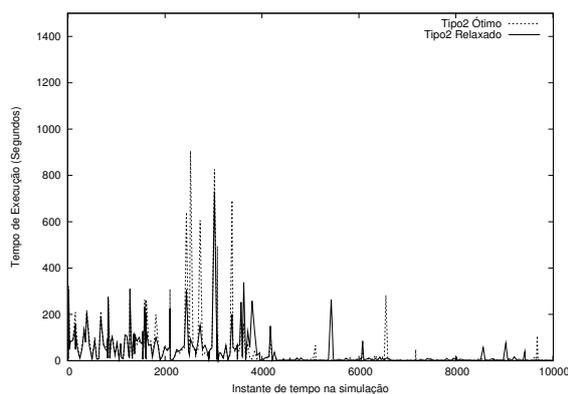


Figura 3. Tempo de execução dos algoritmos (Tipo 2) - Cenário dinâmico

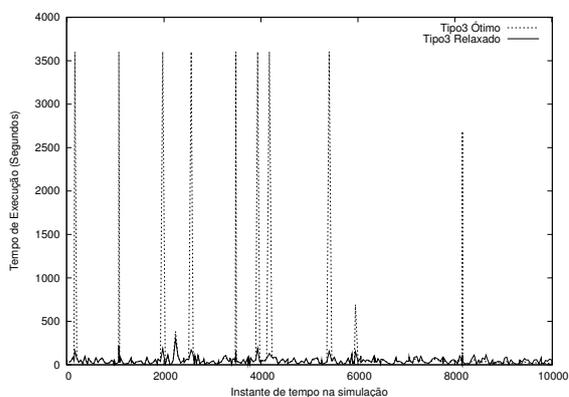


Figura 4. Tempo de execução dos algoritmos (Tipo 3) - Cenário dinâmico

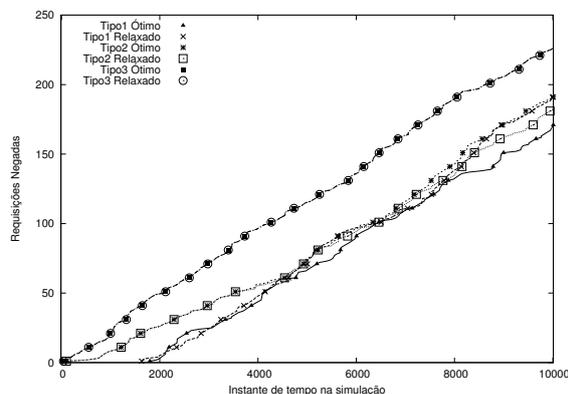


Figura 5. Requisições negadas - Cenário dinâmico

lhantes. O resultado exibido no gráfico da Figura 5 confirma os baixos tempos de execução dos algoritmos nas figuras 2, 3 e 4 a medida que o tempo aumenta. Como a maioria dos recursos está ocupada, a quantidade de requisições bloqueadas aumenta, o bloqueio é influenciado pelo tipo de rede requisitada. Alocações de redes do Tipo 3 são mais bloqueadas do que as do Tipo 2, que por sua vez são mais bloqueadas do que as redes do Tipo 1, como esperado.

5. Conclusões e Trabalhos Futuros

Neste trabalho, foram propostos dois algoritmos para o mapeamento de redes virtuais em um substrato da rede que consideram ambientes realistas. O algoritmo relaxado é capaz de encontrar a solução bem mais rapidamente do que o algoritmo ótimo, enquanto que a banda passante alocada pelos dois algoritmos é bem semelhante. Além disso, o bloqueio produzido pelos algoritmos é semelhante.

Durante as simulações foi constatado que os algoritmos utilizam muita memória RAM para serem executados. Mesmo o algoritmo relaxado pode levar um tempo consi-

deravelmente alto em ambientes com muitos nós físicos e requisições complexas. Desta forma, tem-se como metas futuras a determinação dos valores numéricos do consumo de memória dos algoritmos e o desenvolvimento de um algoritmo que seja capaz de encontrar uma solução satisfatória em um tempo menor do que o dos algoritmos propostos neste trabalho. Além disso, este novo algoritmo deve ser capaz de encontrar soluções utilizando pouca memória RAM.

Como os trabalhos presentes na literatura não levam em consideração a maioria dos parâmetros dos algoritmos propostos neste trabalho, comparações com algoritmos como os apresentados em [Chowdhury et al. 2009] e [Lischka and Karl 2009] exigem adaptações para a entrada e a saída de métricas específicas. Essas comparações serão realizadas pelos autores com o objetivo de se quantificar o impacto decorrente de uma quantidade maior de parâmetros nos algoritmos de mapeamento.

Referências

- Albert, R. and Barabási, A. L. (2000). Topology of Evolving Networks: Local Events and Universality. *Physical Review Letters*, 85(24):5234–5237.
- Andersen, D. G. (2002). Theoretical Approaches to Node Assignment. <http://www.cs.cmu.edu/~dga/papers/andersen-assign.ps>. Último acesso em 20/12/2010.
- Bless, R., Hiibsch, C., Mies, S., and Waldhorst, O. (2008). The Underlay Abstraction in the Spontaneous Virtual Networks (SpoVNet) Architecture. In *Next Generation Internet Networks (NGI 2008)*, pages 115–122.
- Chowdhury, N., Rahman, M., and Boutaba, R. (2009). Virtual Network Embedding with Coordinated Node and Link Mapping. In *IEEE INFOCOM*, pages 783–791.
- Cisco Systems (2010a). Cisco 7200 Series Routers Overview [Cisco 7200 Series Routers]. http://www.cisco.com/en/US/prod/collateral/routers/ps341/product_data_sheet09186a008008872b.html. Último acesso em 20/12/2010.
- Cisco Systems (2010b). Cisco Multiprocessor WAN Application Mode [Cisco Catalyst 6500 Series Switches]. http://www.cisco.com/en/US/prod/collateral/modules/ps5510/product_data_sheet0900aecd800f8965_ps708_Products_Data_Sheet.html. Último acesso em 20/12/2010.
- Cisco Systems (2010c). Download Software. <http://www.cisco.com/cisco/software/release.html?mdfid=278807391&flowid=956&softwareid=280805680&release=12.4.2-XB11&rellifecycle=GD&relind=AVAILABLE&reltype=latest>. Último acesso em 20/12/2010.
- Fan, J. and Ammar, M. H. (2006). Dynamic Topology Configuration in Service Overlay Networks: A Study of Reconfiguration Policies. In *IEEE INFOCOM*, pages 1–12.
- 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.

- He, J., Zhang-Shen, R., Li, Y., Lee, C.-Y., Rexford, J., and Chiang, M. (2008). DaVinci: Dynamically Adaptive Virtual Networks for a Customized Internet. In *ACM CoNEXT '08*, pages 15:1–15:12.
- Houidi, I., Louati, W., and Zeghlache, D. (2008). A Distributed and Autonomic Virtual Network Mapping Framework. In *ICAS '08*, pages 241–247.
- IBM (2010). IBM ILOG CPLEX Optimization Studio V12.2. <http://publib.boulder.ibm.com/infocenter/cosinfoc/v12r2/index.jsp?topic=/ilog.odms.cplex.help/Content/Optimization/Documentation/CPLEX/>. Último acesso em 20/12/2010.
- Lischka, J. and Karl, H. (2009). A Virtual Network Mapping Algorithm Based on Subgraph Isomorphism Detection. In *ACM VISA '09*, pages 81–88.
- Lu, J. and Turner, J. (2006). Efficient Mapping of Virtual Networks onto a Shared Substrate. Technical Report WUCSE-2006-35, Washington University. <http://www.arl.wustl.edu/~jst/pubs/wucse2006-35.pdf>. Último acesso em 20/12/2010.
- Medina, A., Lakhina, A., Matta, I., and Byers, J. (2010). Brite. <http://www.cs.bu.edu/brite/>. Último acesso em 20/12/2010.
- Padala, P., Shin, K. G., Zhu, X., Uysal, M., Wang, Z., Singhal, S., Merchant, A., and Salem, K. (2007). Adaptive Control of Virtualized Resources in Utility Computing Environments. In *ACM EuroSys '07*, pages 289–302.
- RNP (2010). Mapa do backbone RNP. <http://www.rnp.br/backbone/>. Último acesso em 20/12/2010.
- Trossen, D. (2009). Invigorating the Future Internet Debate. *SIGCOMM Comput. Commun. Rev.*, 39(5):44–51.
- 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.
- Zhu, Y. and Ammar, M. (2006). Algorithms for Assigning Substrate Network Resources to Virtual Network Components. In *IEEE INFOCOM*, pages 1–12.
- Zhu, Y., Bavier, A., Feamster, N., Rangarajan, S., and Rexford, J. (2008a). UFO: a resilient layered routing architecture. *SIGCOMM Comput. Commun. Rev.*, 38(5):59–62.
- Zhu, Y., Zhang-Shen, R., Rangarajan, S., and Rexford, J. (2008b). Cabernet: Connectivity Architecture for Better Network Services. In *ACM CoNEXT '08*, pages 64:1–64:6.