

Algoritmo de Engenharia de Tráfego para Aprovisionamento Automático de Serviços Metro Ethernet em Redes PBB-TE

Rafael A. Scaraficci¹, Marcos R. Salvador¹

¹CPqD - Centro de Pesquisa e Desenvolvimento em Telecomunicações
Rodovia Campinas - Mogi-Mirim, km 118,5 – Campinas – SP – Brazil

{raugusto,marcosrs}@cpqd.com.br

Abstract. *This paper describes a traffic engineering algorithm for automatic provisioning of EPL and EVPL Metro Ethernet services in PBB-TE networks. Based on the genetic algorithm technique, the traffic engineering algorithm can calculate working and protection tunnel routes for one or more service requests at once. Besides service requests parameters, the algorithm also provides few configurable parameters for fine tuning, such as load balance between working and protection tunnels over a link. The paper comparatively analyzes the performance of the algorithm against others available in the literature. The results show that the algorithm can cost-effectively meet service requests with low blocking and load balance.*

Resumo. *Este artigo descreve um algoritmo de engenharia de tráfego para o provisionamento automático de serviços Metro Ethernet do tipo EPL e EVPL em uma rede PBB-TE. Baseando-se na técnica de algoritmos genéticos, o algoritmo calcula túneis primário e de proteção para uma ou mais requisições de serviço de uma só vez. Além de parâmetros relacionados às requisições de serviço, o algoritmo permite o ajuste de parâmetros de configuração, como o balanceamento entre túneis primários e de proteção nos enlaces. O artigo analisa comparativamente a performance do algoritmo em relação a outros disponíveis na literatura. O algoritmo mostrou-se capaz de satisfazer as requisições com uma baixa taxa de bloqueio e um bom balanceamento de carga.*

1. Introdução

Atualmente, um dos principais objetivos dos provedores de serviços de internet e telecomunicações é atender as demandas dos clientes por diferentes serviços como, por exemplo, voz, vídeo e dados; e ao mesmo tempo fazer um uso eficiente dos recursos de rede, maximizando o número de clientes atendidos com a mesma quantidade de recursos.

Para satisfazer essa necessidade dos provedores, surgiram tecnologias orientadas à conexão como o MPLS (*Multi Protocol Label Switching*) e, mais recentemente, o PBB-TE (*Provider Backbone Bridge Traffic Engineering*). Essas tecnologias trabalham com rotas fim-a-fim, possibilitando que mecanismos de engenharia de tráfego sejam aplicados eficientemente. Tanto o MPLS como o PBB-TE dependem de um procedimento para computar suas rotas. A maneira como as rotas são escolhidas influencia diretamente nos objetivos dos provedores. Logo, algoritmos para o cálculo de rotas têm sido intensamente pesquisados na literatura [Kuipers et al. 2002, Wang et al. 2008]. No entanto, ainda quase não se encontram soluções voltadas, especificamente, para a tecnologia PBB-TE.

O foco desse trabalho é o cálculo de rotas fim-a-fim para túneis PBB-TE, que serão utilizados para atender requisições de serviço do tipo *e-line* [Metro Ethernet Forum 2004] com demandas específicas de banda por classe de serviço. Devido a requisitos de disponibilidade, as rotas são sempre calculadas com proteção dedicada. Trata-se de um problema de engenharia de tráfego do tipo *online* [Wang et al. 2008], onde uma ou mais requisições podem ser calculadas simultaneamente.

Apresentamos uma solução algorítmica inovadora baseada em algoritmos genéticos. Resultados computacionais mostraram que o nosso algoritmo é uma boa alternativa para se evitar bloqueios, distribuir a carga na rede e ao mesmo tempo garantir proteção dedicada de rotas.

O artigo segue a seguinte estrutura. Na Seção 2 é apresentada uma revisão de literatura. Na Seção 3 é descrito o problema. Na Seção 4 é descrito o modelo de resolução. Na Seção 5 são apresentados os resultados computacionais. Finalmente, na Seção 6 são apresentadas algumas conclusões.

2. Trabalhos Correlatos

Dentre os diferentes tipos de algoritmos de engenharia de tráfego [Wang et al. 2008], estamos interessados nos algoritmos do tipo *online* que consideram restrições de banda. Na sequência apresentamos algumas das estratégias mais comuns para este tipo de problema.

Constrained Shortest Path First (CSPF) é um algoritmo amplamente utilizado. Consiste em remover do grafo, que representa a topologia de rede, as arestas que não atendem a demanda de banda da requisição e utilizar um algoritmo de caminho mínimo no grafo residual. Este algoritmo não balanceia a carga nas arestas, uma vez que sempre seleciona o mesmo caminho mínimo até que ele seja saturado. Essa estratégia tende a criar condições precoces de bloqueios, pois certas arestas podem ser rapidamente saturadas.

Constrained Widest Shortest Path (CWSP) [Guerin et al. 1997] é uma modificação do algoritmo CSPF. Consiste em selecionar entre todos os caminhos mínimos, o caminho de maior banda residual. Assim como o CSPF, o CWSP também não é capaz de selecionar caminhos menos utilizados que sejam maiores do que os caminhos mínimos. Logo, tem-se uma saturação precoce de certas arestas.

Bandwidth Constrained Routing Algorithm (BCRA) [Kotti et al. 2007] é um algoritmo de caminho de custo mínimo que utiliza uma função de custo para cada aresta baseada na sua capacidade, banda reservada e reservável. Inicialmente, o algoritmo remove as arestas que não atendem a demanda de banda da requisição e, em seguida, calcula um caminho de custo mínimo. Esse algoritmo tem um bom desempenho, produzindo resultados melhores do que outros algoritmos de engenharia de tráfego como o MIRA [Kar et al. 2000, Kodialam and Lakshman 2000], que é um algoritmo que busca reduzir o bloqueio de requisições futuras, e DAMOTE [Blanchy et al. 2003], que tenta balancear a carga nas arestas, distribuindo a probabilidade de bloqueio igualmente pela rede.

Os algoritmos CSPF, CWSP e BCRA serão utilizados como referência para comparações. Como esses algoritmos foram propostos para calcular um caminho simples, estendemo-os para também calcularem um caminho de proteção. Para isso adaptamos esses algoritmos ao algoritmo de caminhos disjuntos proposto em [Bhandari 1997].

3. Descrição do Problema

O problema de provisionamento de serviços *e-line* em redes PBB-TE consiste em definir, para um conjunto de requisições de serviços *e-line*, túneis e serviços PBB-TE que garantam uma taxa de transmissão mínima e proteção dedicada de rotas do tipo 1:1¹. Trata-se de um problema de engenharia de tráfego do tipo *online*, onde novos recursos PBB-TE devem ser calculados levando em consideração o estado operacional da rede e os túneis e serviços PBB-TE ativos. A seguir são apresentados em detalhes, os elementos que compõem o problema.

3.1. Topologia de Rede

Considera-se uma topologia de rede em malha, que é modelada como um grafo $G(V, E)$, onde V é o conjunto de todos os nós, representando os *switches*, e E é o conjunto de todas as arestas, representando conectividade entre os nós. Os nós estão particionados em dois conjuntos V_e e V_c , onde V_e é o conjunto de nós de borda, representando *switches* que efetuam o encapsulamento e o desencapsulamento de pacotes PBB-TE, ou seja, são *switches* terminais dos túneis PBB-TE, e V_c é o conjunto de nós de núcleo, representando *switches* que efetuam apenas o encaminhamento de pacotes PBB-TE. Cada aresta $e \in E$ é bidirecional e tem uma capacidade máxima de transmissão C_e .

Os *switches* implementam controle de banda e diferenciação de serviços através de mecanismos de marcação de pacotes e filas de prioridade. A cada classe de serviço s , definida pelo provedor, está associado um fator multiplicativo α_s , sendo $\alpha_s C_e$ a capacidade máxima da aresta e para serviços da classe s .

3.2. Requisições de Serviço *E-Line*

Considera-se os dois tipos de serviços *e-line* definidos pelo *Metro Ethernet Forum*: EPL (*Ethernet Private Line*) e EVPL (*Ethernet Virtual Private Line*). Uma requisição de serviço EPL é modelada como uma trinca (v_s, v_d, b) , onde $v_s, v_d \in V_e$ correspondem aos nós terminais e b corresponde à demanda de banda, a qual estabelece, para cada classe de serviço s , uma taxa de transmissão a ser garantida e uma taxa de transmissão de pico. Uma requisição de serviço EVPL é modelada como um conjunto de quádruplas $(v_s, v_d, b, c-vid)$, onde v_s, v_d e b têm significados análogos aos empregados para serviços EPL e $c-vid$ corresponde a um identificador de *vlan* do cliente. Vale ressaltar que no caso do serviço EPL o fluxo cliente é tratado de maneira agregada, sem distinção de *vlans*; diferentemente do serviço EVPL, onde é possível dar um tratamento distinto para cada *vlan* do cliente. Para maiores informações sobre serviços *e-line*, consulte a especificação para serviços *ethernet* [Metro Ethernet Forum 2004].

É restrição do problema aceitar uma requisição de serviço *e-line* se e somente se for possível garantir a demanda. Isto é

$$\sum_{i \in I_e} \sum_{s \in S} cir_s^i \leq C_e \quad \text{e} \quad \sum_{i \in I_e} cir_s^i \leq \alpha_s C_e, \quad \forall s,$$

onde I_e é o conjunto de serviços *e-line* encaminhados através da aresta e , S é o conjunto de classes de serviço e cir_s^i é a demanda de banda garantida do serviço i para a classe de

¹Neste tipo de proteção, existem duas rotas disjuntas com recursos reservados e no caso de alguma falha na rota que está transmitindo, tem-se a comutação da transmissão para a outra rota.

serviço s . Observe que não se garante a banda de pico dos serviços. O fluxo em excesso, ou seja, acima da banda garantida e abaixo da banda de pico será tratado pelos *switches* como sendo de baixa prioridade e sem nenhuma garantia de entrega, sendo transmitido conforme a disponibilidade de banda.

3.3. Túneis e Serviços PBB-TE

Um túnel PBB-TE define um caminho entre dois nós terminais no grafo G , sendo modelado como uma sequência de nós e arestas distintos $(v_s, e_1, v_1, e_2 \dots v_{n-1}, e_n, v_d)$, onde, $v_s, v_d \in V_e, v_i \in V_c$, para $i = 1, 2, \dots (n-1)$, e $e_j \in E$, para $j = 1, 2, \dots n$. Por definição, cada túnel está associado a um identificador de *vlan*, denominado de *b-vid*. Por limitação dos *switches* considerados, dois túneis em um mesmo elemento de borda têm que ter *b-vids* distintos. Num mesmo túnel PBB-TE pode haver diferentes serviços PBB-TE, os quais são identificados no túnel através de um identificador de serviço, denominado de *i-sid*. Esses identificadores (*b-vid* e *i-sid*) são codificados nos pacotes PBB-TE.

Em uma rede PBB-TE, a proteção contra falhas ao longo do caminho é realizada associando-se dois túneis de maneira a formar um par primário-proteção. A tecnologia PBB-TE provê mecanismos para monitorar os pares de túneis e no caso de falha no caminho do túnel ativo, comuta-se a transmissão dos dados para o outro túnel do par. Para maiores informações sobre a tecnologia PBB-TE, consulte a especificação [IEEE 2007].

Um túnel PBB-TE é modelado como uma quádrupla $(t, b-vid, p, I)$, onde t identifica o túnel como sendo primário ou proteção, *b-vid* identifica a *vlan*, p é um caminho no grafo G e I é um conjunto de serviços PBB-TE. Cada serviço PBB-TE é definido pelo par $(i-sid, b)$, onde *i-sid* é um identificador de serviço PBB-TE e b é a demanda de banda do serviço, conforme definida para as requisições de serviço *e-line*.

É restrição do problema que um túnel PBB-TE esteja sempre associado com um outro túnel PBB-TE, formando um par de túneis primário e proteção. Ademais, esses túneis devem ter caminhos disjuntos e garantir a demanda de banda dos seus serviços. Isto é, a banda de todos os serviços PBB-TE em uma dada aresta e não deve superar a capacidade da aresta C_e e nem a capacidade máxima para cada classe de serviço s na aresta e , definida como $\alpha_s C_e$.

3.4. Relação entre Serviços E-Line e Túneis e Serviços PBB-TE

O *Metro Ethernet Forum* define os serviços EPL e EVPL independentemente da tecnologia utilizada para implementá-los, não estabelecendo regras de mapeamento desses serviços em recursos PBB-TE. Logo, definimos as seguintes regras de mapeamento:

- Um serviço EPL definido como (v_s, v_d, b) é mapeado em um túnel PBB-TE exclusivo com nós terminais em v_s e v_d . Sendo, todo o fluxo do cliente encapsulado em uma única instância de serviço PBB-TE neste túnel.
- Um serviço EVPL definido como $\{(v_{s1}, v_{d1}, b_1, c-vid_1) \dots (v_{sn}, v_{dn}, b_n, c-vid_n)\}$ é mapeado em um ou mais túneis PBB-TE exclusivos. O número de túneis depende do número de pares de nós terminais distintos que compõem o serviço. Cada elemento $(v_{si}, v_{di}, b_i, c-vid_i)$ é mapeado em um túnel PBB-TE com nós terminais v_{si} e v_{di} e numa única instância de serviço PBB-TE. Quando há dois ou mais elementos do mesmo serviço EVPL com os mesmos nós terminais, todos são mapeados no mesmo túnel PBB-TE, mas em instâncias de serviço PBB-TE distintas.

Note que uma requisição de serviço *e-line* é mapeada em um ou mais túneis e serviços PBB-TE, que são exclusivos à requisição. Logo, pelas nossas regras, requisições diferentes não compartilham o mesmo túnel e nem o mesmo serviço PBB-TE.

4. Modelo de Resolução

O algoritmo de engenharia de tráfego é parte de um sistema central de gerência, o qual é responsável por monitorar e configurar toda a rede PBB-TE. Apesar da gerência ser centralizada, mecanismos de redundância podem ser implementados. Considera-se como entrada do algoritmo: uma topologia de rede em malha, definida como um grafo $G(V, E)$, um conjunto de túneis e serviços PBB-TE ativos A , onde $a = (t, b-vid, p, I)$, para $a \in A$, e um conjunto de novas requisições de serviços *e-line* R , sendo que para $r \in R$, tem-se $r = (v_s, v_d, b)$, se o serviço for do tipo EPL, ou $r = \{(v_{s1}, v_{d1}, b_1, c-vid_1) \dots (v_{sn}, v_{dn}, b_n, c-vid_n)\}$, se o serviço for do tipo EVPL. O processo de resolução resume-se em:

1. Calcular recursos de rede disponíveis.
2. Mapear as requisições de serviço *e-line* em requisições de túneis PBB-TE.
3. Calcular um conjunto de novos túneis PBB-TE.
4. Mapear as requisições de serviço *e-line* em serviços PBB-TE nos novos túneis.

As etapas 1, 2 e 4 são resolvidas através de procedimentos determinísticos e a etapa 3 através de um algoritmo genético. Na sequência são descritas as etapas.

4.1. Calculando Recursos de Rede Disponíveis

Uma vez que serviços encontram-se ativos na rede, torna-se necessário calcular os recursos disponíveis. A partir do conjunto A de túneis PBB-TE ativos, computa-se a banda reservável em cada aresta e do grafo G para cada classe de serviço s . Este processo consiste em retirar da capacidade inicial da aresta e a banda garantida, a qual é destinada aos túneis provisionados cujas rotas contêm a aresta e . Ademais, para cada nó de borda $v \in V_e$, computa-se uma lista de identificadores de *vlangs* que não são utilizados em nenhum túnel que tenha uma das extremidades em v . A capacidade reservável em cada aresta e os identificadores disponíveis são utilizados no cálculo de novos recursos PBB-TE, os quais não devem interferir com os recursos PBB-TE ativos.

4.2. Mapeando Requisições de Serviço *E-Line* em Requisições de Túneis PBB-TE

Para cada requisição de serviço EPL em R , descrita como (v_s, v_d, b) , define-se uma requisição de um par de túneis primário e proteção, que é representada pela trinca (v'_s, v'_d, b') , onde $v'_s = v_s$ e $v'_d = v_d$ correspondem aos nós terminais dos dois túneis e $b' = b$ corresponde à demanda de banda de cada um dos túneis, que é igual para o túnel primário e o túnel de proteção.

Para cada requisição de serviço EVPL em R , descrita como $\{(v_{s1}, v_{d1}, b_1, c-vid_1) \dots (v_{sn}, v_{dn}, b_n, c-vid_n)\}$, define-se um ou mais pares de túneis da seguinte maneira. Para cada conjunto de elementos com os mesmos nós terminais v_{si} e v_{di} , define-se uma requisição de um par de túneis (v'_{si}, v'_{di}, b') , onde $v'_{si} = v_{si}$ e $v'_{di} = v_{di}$ correspondem aos nós terminais dos túneis primário e proteção e b' corresponde à soma das demandas de cada elemento da requisição do serviço EVPL com nós terminais em v_{si} e v_{di} .

Aplicando-se essas regras de mapeamento para cada requisição de serviço *e-line* $r \in R$, deriva-se um conjunto T de requisições de pares de túneis primário e proteção.

4.3. Cálculo de Túneis PBB-TE

Para cada requisição de par de túneis primário e proteção (v'_s, v'_d, b') , calcula-se duas rotas de v'_s até v'_d que garantam a demanda de banda b' e sejam disjuntas em relação aos nós. Além disso, define-se um identificador de *vlan* para cada uma das rotas. Por definição, o identificador é um valor inteiro no intervalo $[0; 2^{12}]$, sendo que dois túneis que tenham pelo menos um dos nós terminais em comum precisam ter identificadores de *vlan* distintos. Para se gerar um conjunto de túneis válidos para as requisições de pares de túneis, utilizou-se um algoritmo genético.

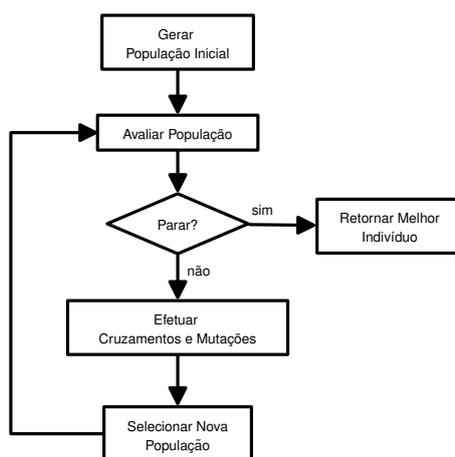


Figura 1. Fluxograma do algoritmo genético

O algoritmo genético [Mitchell 1998] é uma técnica de otimização evolutiva muito estudada no meio acadêmico e utilizada na prática para se resolver problemas NP-Difíceis. Em linhas gerais, uma implementação do algoritmo genético segue o fluxograma da Figura 1, onde, inicialmente, gera-se um conjunto de soluções para o problema, denominado de população inicial. Em seguida, avalia-se as soluções (também denominadas de indivíduos ou de cromossomos da população) e caso o critério de parada seja atingido, retorna-se o melhor indivíduo. Caso contrário, gera-se uma nova população através de cruzamentos e mutações. O cruzamento consiste em gerar novos indivíduos a partir da informação de dois ou mais indivíduos e a mutação consiste em inserir aleatoriamente uma característica no indivíduo. Entre os indivíduos da população corrente e os novos indivíduos, seleciona-se uma nova população para a próxima geração. O processo de evolução continua até que o critério de parada seja atingido. Na seqüência, é apresentado como cada uma dessas etapas foi implementada.

4.3.1. Representação de um Cromossomo (Solução)

Representa-se um cromossomo como uma seqüência de genes, sendo cada gene um par de rotas com um identificador de *vlan* associado a cada uma das rotas. Desta forma, tem-se que cada gene é uma solução para uma requisição de par de túneis $t \in T$. Na Figura 2 é apresentado, em detalhes, um cromossomo e o seu i -ésimo gene, o qual é composto por uma rota primária $(v_s, e_5, v_4, e_6, v_5, e_7, v_d)$ com identificador de *vlan* x e por uma rota de proteção $(v_s, e_1, v_1, e_2, v_2, e_3, v_3, e_4, v_d)$ com identificador de *vlan* y .

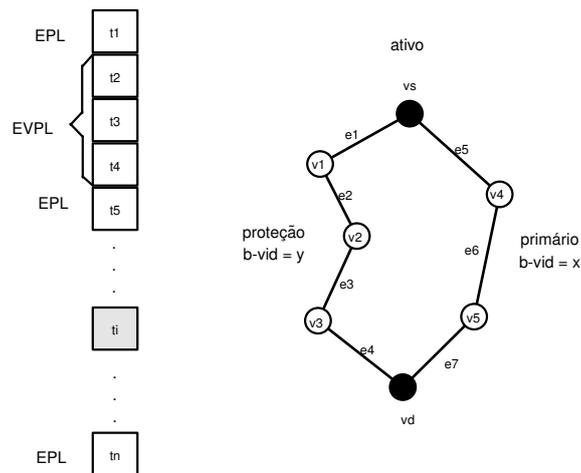


Figura 2. Representação gráfica de um cromossomo

Um gene pode estar no estado ativo ou inativo. O estado ativo significa que a requisição correspondente ao gene será aceita, pois há banda disponível para os túneis primário e proteção e existem identificadores de *vlan* disponíveis para cada um dos túneis. A validação do gene em relação à banda e identificadores é feita durante o procedimento de construção dos cromossomos. Caso o estado do gene seja inativo, tem-se o bloqueio da requisição de serviço correspondente ao respectivo gene. No caso de uma requisição de serviço EVPL composta por mais de um gene, se um dos seus genes está no estado inativo, todos os demais genes deste serviço EVPL também estarão no estado inativo, pois uma requisição de serviço não pode ser parcialmente aceita.

4.3.2. Função de Avaliação

Avalia-se uma solução através de uma função de engenharia de tráfego, que tem como meta: maximizar o número requisições aceitas para aprovisionamento, reduzir o número de bloqueios no futuro, balancear a carga nas arestas entre banda primária e de proteção e distribuir a banda livre adequadamente entre os serviços sem nenhuma garantia de banda. A função de avaliação é descrita pelos seguintes objetivos:

$$Obj_1 : Max \left(\sum_{r \in R^*} 1 \right) \quad e \quad Obj_2 : Min \left(\gamma \sum_{e \in E} \frac{B_e}{C_e - B_e} + \lambda \sum_{e \in E} \frac{D_e}{C_e - D_e} \right),$$

onde R^* é o conjunto de requisições de serviço *e-line* aceitas para aprovisionamento, ou seja, com genes no estado ativo, E é o conjunto de arestas, B_e é a soma da banda com garantia já aprovisionada e aceita para aprovisionamento na aresta e , incluindo tanto banda primária como de proteção, C_e é a capacidade da aresta e e D_e é o módulo da diferença entre banda primária aprovisionada mais banda aceita para aprovisionamento e banda de proteção aprovisionada mais banda aceita para aprovisionamento na aresta e . A variável D_e considera somente banda que deve ser de fato garantida. As constantes γ e λ são, respectivamente, pesos para o primeiro e segundo termo da função Obj_2 .

A função Obj_1 mede o número total de requisições de serviço *e-line* aceitas para aprovisionamento. A função Obj_2 é uma combinação linear de uma medida de atraso

médio nos enlaces e da diferença entre banda primária e de proteção nos enlaces. O uso do atraso médio como função de engenharia de tráfego é sugerido em [Balon et al. 2006] como uma boa métrica para distribuir a carga e reduzir o número de bloqueios, pois consegue balancear o tamanho das rotas com a carga nos enlaces. Isto deve-se ao fato de que rotas longas aumentam o atraso em vários enlaces e rotas com enlaces que tenham pouca banda reservável provocam um grande aumento do atraso. O segundo termo da função Obj_2 foi adicionado para se balancear banda do tipo primário e de proteção, pois isto tende a reduzir o tráfego nos enlaces em condições normais de funcionamento.

Define-se o valor de avaliação de um cromossomo S como $f(S) = (obj_1, obj_2)$, sendo que S é melhor do que um outro cromossomo S' com função de avaliação $f(S') = (obj'_1, obj'_2)$ se e somente se $(obj_1 > obj'_1)$ ou $(obj_1 = obj'_1)$ e $(obj_2 < obj'_2)$. Os dois cromossomos terão a mesma qualidade se e somente se $(obj_1 = obj'_1)$ e $(obj_2 = obj'_2)$.

4.3.3. Geração da População Inicial

Um cromossomo é gerado, construindo-se sequencialmente um gene para cada requisição de par de túneis $t \in T$. Seja S um cromossomo com os i primeiros genes construídos e $t_{i+1} = (v'_s, v'_d, b')$ a próxima requisição para a qual será construído o gene s_{i+1} . Inicialmente, verifica-se a disponibilidade de um par de identificadores de *vlan* em v'_s e v'_d da seguinte maneira. Seja $W_{v'_s}, W_{v'_d}$ os conjuntos de identificadores de *vlan* disponíveis nos nós v'_s e v'_d respectivamente, seleciona-se como identificadores os dois menores valores em $W_{v'_s} \cap W_{v'_d}$. Caso não haja dois identificadores disponíveis, o estado do gene será inativo. Em seguida, remove-se do grafo G as arestas que não são capazes de atender a demanda de banda b da requisição t_{i+1} , gerando um grafo G' . Para se gerar um par de caminhos disjuntos entre os vértices v_s e v_d , utiliza-se uma modificação do algoritmo de caminhos de vértices disjuntos propostos em [Bhandari 1997]. No entanto, ao invés de utilizar a versão do algoritmo de Dijkstra para grafos com arestas negativas, utilizou-se uma busca em profundidade, que foi alterada para limitar o tamanho máximo do caminho e gerar caminhos distintos.

O algoritmo de busca em profundidade foi alterado da seguinte maneira. Para se limitar o número de arestas do caminho, retorna o processo de busca ao vértice anterior se o limite de arestas foi excedido. Para se gerar caminhos distintos, seleciona-se uma aresta não visitada com uma probabilidade inversamente proporcional ao número de vezes que esta aresta já foi selecionada para genes da requisição t_{i+1} . Usando este algoritmo, um par de caminhos disjuntos cuja soma do caminho tenha no máximo h arestas é gerado. O valor de h é escolhido aleatoriamente na sequência $\{h_{min}, h_{min} + 1, \dots, 3h_{min}\}$ com uma probabilidade que decai $\delta\%$ em relação a probabilidade de escolher o valor anterior. O valor de h_{min} corresponde ao número mínimo de arestas para um par de rotas disjuntas com banda suficiente para atender a requisição. O valor h_{min} é calculado antes de se iniciar a geração da população inicial. Se não for possível gerar um par de caminhos com o valor h devido aos cromossomos anteriores, o valor de h vai sendo incrementado até o valor $3h_{min}$. Se mesmo assim não for possível alocar um par de caminhos, coloca-se o gene no estado inativo e calcula-se um par de caminhos de tamanho máximo h , desconsiderando a banda alocada para os i primeiros genes do cromossomo S no estado ativo. No caso em que houver banda disponível, identificadores de *vlan* e o gene não estiver associado a

uma requisição de serviço EVPL com outros genes no estado inativo, ele será colocado no estado ativo, caso contrário inativo. Se for determinado que o estado do gene é inativo e ele estiver associado a uma requisição de serviço EVPL com outros genes no estado ativo, esses genes serão também colocados no estado inativo. Utilizando-se esse procedimento de geração de genes, gera-se uma população de n cromossomos.

4.3.4. Operadores Genéticos: Cruzamentos e Mutações

Desenvolveu-se um operador de cruzamento específico ao problema. Seja as sequências de genes S_1 e S_2 dois cromossomos pais selecionados para cruzamento, onde s_1^i e s_2^i são os i -ésimos genes das sequências S_1 e S_2 respectivamente. Para cada par de genes s_1^i e s_2^i constrói-se um grafo G'' com as rotas que compõem esses dois genes, em seguida, seleciona-se aleatoriamente em G'' dois pares de rotas disjuntas com os mesmos nós terminais das rotas originais. Para a seleção do par de rotas, utiliza-se o mesmo algoritmo utilizado para se gerar genes da população inicial, mas sem a restrição do número máximo de arestas. Esses novos pares de caminhos disjuntos irão ocupar as i -ésimas posições em dois novos cromossomos S'_1 e S'_2 . Uma vez gerada as novas rotas, elas são validadas quanto a disponibilidade de banda. Caso a demanda de banda seja atendida, verifica-se a disponibilidade de identificadores de *vlan*. Seguindo os mesmos critérios da fase de geração da população inicial, os genes dos novos cromossomos são colocados nos estados ativos e inativos. A seleção de indivíduos para cruzamento é feita através do método de torneio binário [Mitchell 1998] e, uma vez selecionado dois indivíduos, aplica-se o cruzamento com uma probabilidade p_c . Utilizando cruzamentos, gera-se n novos indivíduos, onde n é o tamanho da população inicial.

Os genes da população de pais e filhos são selecionados para mutação com uma probabilidade p_m . Uma vez que um gene é selecionado para mutação, esse gene é substituído por um novo gene gerado conforme o procedimento de geração de genes da população inicial, mas desconsiderando a restrição do número máximo de arestas. Seguindo os mesmos critérios da fase de geração da população inicial, os genes que sofreram mutação são colocados nos estados ativos e inativos.

4.3.5. Seleção da Nova Geração

A seleção da nova geração é feita de maneira elitista. Seja P o conjunto de cromossomos pais e P' o conjunto de cromossomos filhos, ambos de tamanho n . Para a próxima geração são selecionados os n melhores cromossomos de $P \cup P'$ de acordo com a função de avaliação.

4.3.6. Critério de Parada

Para o critério de parada, considera-se um número máximo de gerações g_{max} e um número máximo de gerações sem melhora do melhor cromossomo da população g_{sm} . Uma vez alcançado um desses critérios o algoritmo encerra o procedimento de evolução e retorna o melhor indivíduo.

4.4. Mapeando Requisições de Serviço *E-Line* em Serviços PBB-TE

Para cada gene ativo no cromossomo retornado pela etapa de cálculo de rotas, gera-se um ou mais serviços PBB-TE, dependendo do tipo de serviço *e-line* associado ao gene e ao número de componentes de serviço da requisição. Se o serviço for do tipo EPL, define-se um único serviço PBB-TE com identificador inteiro de serviço (*i-sid*) selecionado aleatoriamente no intervalo $[0, 2^{24}]$. Se o serviço for do tipo EVPL, verifica-se o número de componentes de serviço que estão associadas ao gene. Para cada uma das componentes, define-se um serviço PBB-TE com identificador de serviço PBB-TE inteiro selecionado aleatoriamente no intervalo $[0, 2^{24}]$ e que seja distinto dos identificadores das demais componentes.

5. Resultados Computacionais

Nesta seção iremos reportar os resultados para a topologia da Figura 3, que é uma modificação da rede de *backbone* da NSFNET, comumente utilizada para testes de algoritmos de engenharia de tráfego. Modificamos um pouco esta topologia para incluir elementos de borda do tipo PBB-TE e aumentar a capacidade da rede. Não apresentamos os resultados para uma rede PBB-TE real, pois por ser uma tecnologia nova, ainda não existe uma rede operante com as características descritas nesse trabalho, mas acreditamos que qualquer rede em malha possa ser potencialmente uma rede PBB-TE. Na Figura 3, os nós de borda são representados na forma retangular e os nós de núcleo na forma circular. Existem múltiplas arestas entre os nós, sendo a multiplicidade representada pelo número junto das arestas. A capacidade de cada aresta é igual a 10Gb/s.

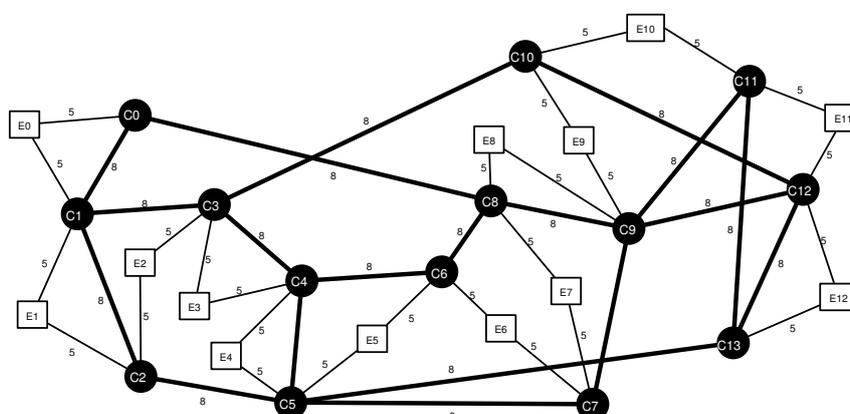


Figura 3. Topologia de teste baseada na topologia da rede NSFNET

Para simulações, considerou-se três classes de serviço, sendo a capacidade máxima das arestas iguais 1Gb/s para classe 1, 3Gb/s para classe 2 e 10Gb/s para a classe 3. Requisições de serviço EPL e EVPL foram geradas aleatoriamente na proporção 3 : 1. A demanda de banda passante de uma requisição de serviço r foi gerada aleatoriamente dentro de certos intervalos. Para a classe 1, definiu-se $cir_1^r = pir_1^r = \{x \text{ Gb/s} \mid x \in \{0; 0,02; 0,04; \dots 0,1\}\}$, para a classe 2, $cir_2^r = \{x \text{ Gb/s} \mid x \in \{0; 0,06; 0,12; \dots 0,3\}\}$ e $pir_2^r = \{x \text{ Gb/s} \mid x \geq cir_2^r \text{ e } x \in \{0; 0,06; 0,12; \dots 0,3\}\}$, e para a classe 3, $cir_3^r = pir_3^r = \{x \text{ Gb/s} \mid x = 1 - pir_1^r - pir_2^r\}$. Os valores cir_s^r e pir_s^r correspondem, respectivamente, aos valores de banda garantida e pico para classe de serviço s na requisição

r . No caso de uma requisição EVPL, gera-se aleatoriamente de 1 à 4 componentes de serviço, sendo a demanda dividida igualmente entre as componentes.

Configurou-se o algoritmo com uma população de tamanho $n = \max(100, 20r)$, onde r é o número de requisições de serviço *e-line*, com um número máximo de gerações $g_{max} = 1000$, com um número máximo de gerações sem melhoras $g_{sm} = \max(50, 5r)$ e com um valor de decaimento do tamanho dos caminhos $\delta = 10\%$. A probabilidade de cruzamento foi configurada com $p_c = 95\%$ e a probabilidade de mutação com $p_m = 1\%$. Na função de avaliação, utilizou-se $\gamma = 1,0$ e $\lambda = 0,1$. Alterar os parâmetros do algoritmo influencia tanto no tempo de execução quanto na qualidade das soluções, por exemplo, diminuindo g_{max} ou g_{sm} é possível reduzir o tempo total de execução, mas por outro lado perde-se na qualidade das soluções, assim como aumentado o valor de λ prioriza o balanceamento entre banda primária e de proteção, mas tende a aumentar a probabilidade de bloqueio. Os valores acima foram obtidos empiricamente através de vários testes que buscaram um equilíbrio entre tempo de execução e qualidade das soluções.

Os resultados obtidos com o algoritmo genético foram comparados com os resultados dos algoritmos citados na Seção 2. Na Figura 4, tem-se o comportamento dos algoritmos para quatro métricas de qualidade. Os resultados correspondem a uma média de 10 execuções de cada algoritmo. Para cada execução, considerou-se um conjunto distinto de requisições, que chegam de acordo com uma distribuição de Poisson de média 1.

No gráfico (a) da Figura 4, tem-se o número médio de bloqueios em função do número de requisições. Observe que os algoritmos CSPF e CWSP apresentam os piores resultados para essa métrica, pois o fato de sempre escolherem a rota de tamanho mínimo leva a condições precoces de bloqueios. O melhor resultado foi obtido pelo algoritmo genético seguido pelo algoritmo BCRA. No gráfico (b) desta mesma figura, tem-se a carga máxima entre todos os enlaces em função do número de requisições. Note que por escolher sistematicamente a rota com o menor número de enlaces, o algoritmo CSPF é o primeiro a bloquear um enlace, por volta da requisição 50. O algoritmo CWSP demora mais para atingir essa mesma condição, por volta da requisição 250. Isto deve-se ao fato do algoritmo CWSP escolher a rota com maior banda entre as rotas com um número mínimo de enlaces. Para essa métrica, o algoritmo genético não tem um bom começo, sendo inicialmente pior do que os algoritmos CWSP e BCRA. No entanto, a medida que o número de requisições aumenta, ele se torna superior aos demais algoritmos.

Com exceção do algoritmo genético, nenhum outro algoritmo trata a diferença entre a quantidade de banda primária e de proteção nos enlaces. Por essa razão, o algoritmo genético apresentou o melhor resultado em relação a essa métrica, como pode ser observado no gráfico (c) da Figura 4, que apresenta a diferença máxima entre banda do tipo primária e proteção em relação a capacidade do enlace. Para essa métrica o pior resultado ficou com o algoritmo CSPF, devido a maneira restrita com que escolhe as rotas. O algoritmo CWSP teve o segundo pior resultado e o algoritmo BCRA foi um pouco melhor para essa métrica.

Apesar de nenhum dos algoritmos tratar diretamente o atendimento da banda sem garantia, eles apresentam comportamentos distintos para essa métrica, conforme apresentado no gráfico (d) da Figura 4, o qual mostra a porcentagem média de banda sem garantia que está sendo atendida. Esse comportamento está relacionado com a distribuição

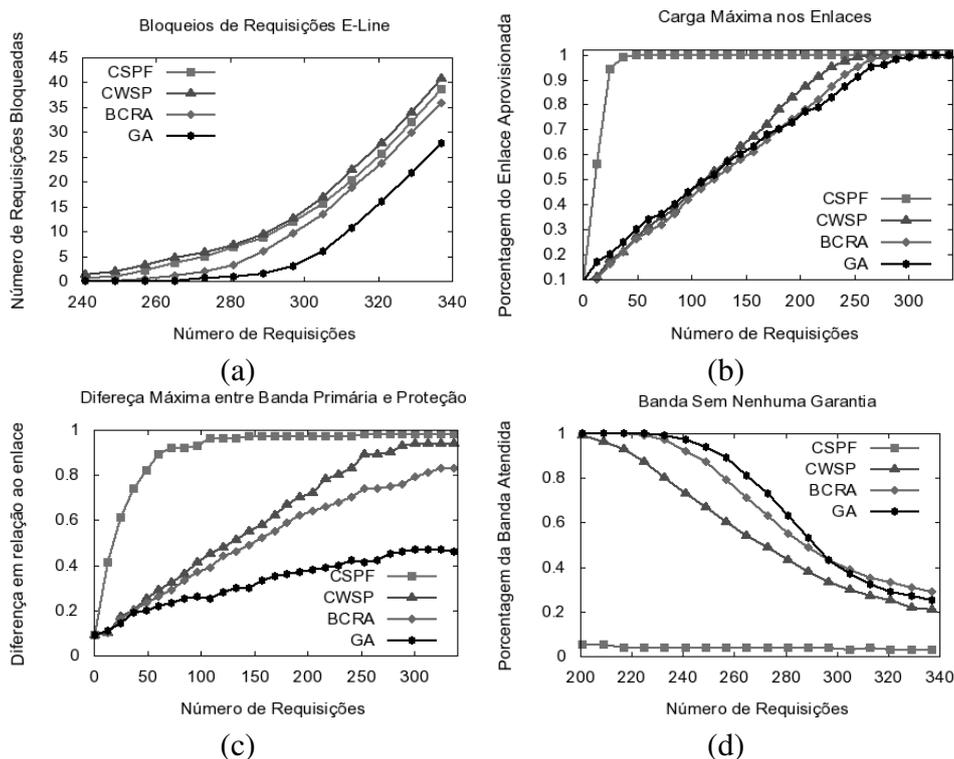


Figura 4. Resultados dos algoritmos (em média 1 requisição por vez)

da carga nos enlaces. Quanto melhor essa distribuição, mais banda livre nos enlaces fica disponível para serviços sem garantia. O pior resultado novamente é do algoritmo CSPF sendo seguido pelo algoritmo CWSP, que é bem melhor. O algoritmo genético tem um resultado superior até por volta da requisição 290, então sendo superado pelo algoritmo BCRA. No entanto, lembre-se que o algoritmo genético bloqueia menos requisições do que os demais algoritmos o que implica numa redução de banda disponível para serviços com e sem garantia.

De maneira análoga testes foram realizados para uma chegada de requisições de serviço de acordo com uma distribuição de Poisson de média 10. Os resultados são apresentados na Figura 5. Os comportamentos dos algoritmos são similares aos descritos anteriormente.

Vale ressaltar que os algoritmos CSPF, CWSP e BCRA tratam as requisições uma-a-uma, apesar de um número maior de requisições poder estar disponível para processamento simultâneo. No entanto, apesar do algoritmo genético ser capaz de tratar todas as requisições disponíveis simultaneamente, não notou-se uma vantagem significativa devido a um maior número de requisições. Além disso, aumentando o número de requisições é necessário um tempo muito maior de processamento. Logo, considerando qualidade e tempo de execução, notou-se que um bom número de requisições de serviço para serem processadas simultaneamente varia entre 1 à 10 requisições. Acreditamos que esse seja um número típico de requisições para uma rede PBB-TE real, mas se houver um número maior de requisições, sugerimos processar essas requisições em blocos de até 10 requisições, o que implica em executar o algoritmo sequencialmente para cada blocos.

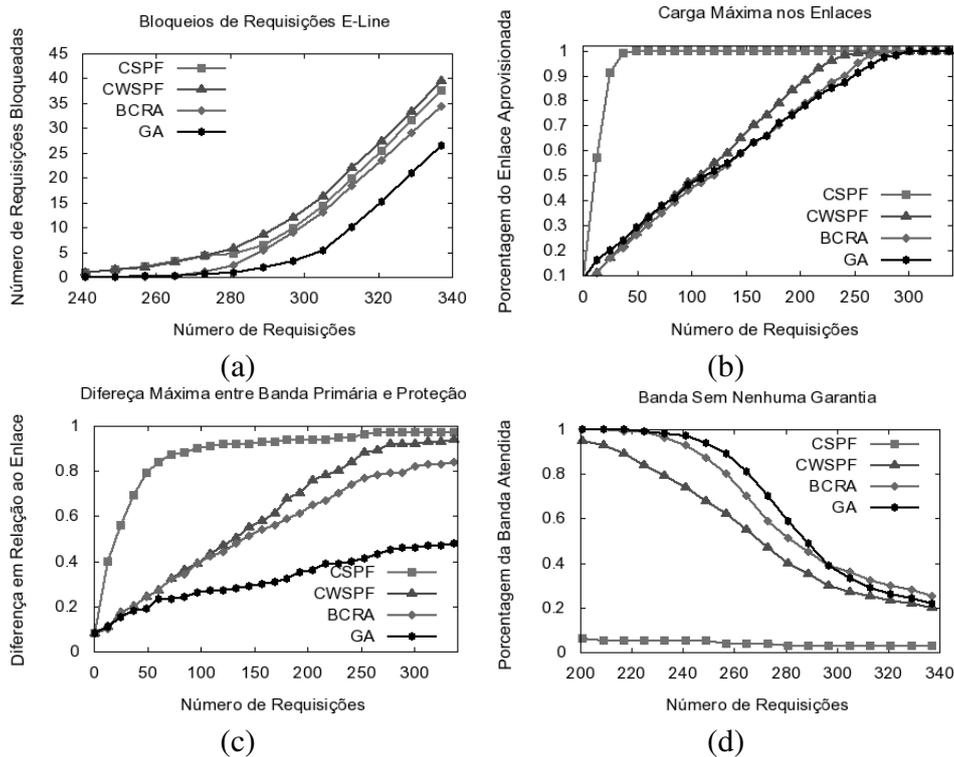


Figura 5. Resultados dos algoritmos (em média 10 requisições por vez)

A complexidade do algoritmo genético é bem maior do que a dos outros algoritmos avaliados, pois enquanto os demais algoritmos calculam apenas uma única solução para uma requisição de serviço, o algoritmo genético calcula um conjunto de soluções para uma mesma requisição e aplica inúmeras operações sobre essas soluções. Por exemplo, os testes aqui relatados foram executados em um Pentium 4 de 2.79GHz com 512MB de memória RAM e o tempo médio de execução do algoritmo genético por requisição de serviço *e-line* foi de aproximadamente 5s, enquanto para os demais algoritmos foi de aproximadamente 2ms. Considerando o fato de que em uma rede PBB-TE real, o algoritmo será executado provavelmente uma ou duas vezes por dia para um número reduzido de requisições, o algoritmo genético mostra-se aplicável.

Testes computacionais foram realizados em outras topologias e desempenhos similares aos apresentados para a topologia de teste da Figura 3 foram observados.

6. Conclusões

Neste artigo, apresentamos uma solução para o problema de provisionamento de serviços *e-line* em uma rede PBB-TE considerando restrições de banda e proteção dedicada de rotas. A solução apresentada é inovadora e considera o processo desde o mapeamento das requisições de serviço EPL e EVPL em túneis e serviços PBB-TE até a definição dos recursos PBB-TE, incluindo o cálculo de rotas com proteção.

Como parte deste trabalho, estendemos alguns algoritmos tradicionais de cálculo de rota (CSPF, CWSP e BCRA) para calcular rotas com proteção dedicada e comparamos os seus resultados com um novo algoritmo genético, proposto neste trabalho. Resultados computacionais mostraram que a nossa solução é uma boa alternativa para se evitar

bloqueios, distribuir a carga na rede e garantir proteção dedicada de rotas.

Dado que o PBB-TE é uma tecnologia nova, muito ainda tende a ser pesquisado e melhorado. No entanto, acreditamos que esse trabalho possa acrescentar novas idéias para o desenvolvimento de novas soluções dentro da tecnologia PBB-TE.

Referências

- Balon, S., Skivée, F., and Leduc, G. (2006). How Well Do Traffic Engineering Objective Functions Meet TE Requirements? In *Proceedings of IFIP Networking 2006, Coimbra*, volume 3976. Springer LNCS.
- Bhandari, R. (1997). Optimal physical diversity algorithms and survivable networks. *Computers and Communications, IEEE Symposium on*, 0.
- Blanchy, F., Mélon, L., and Leduc, G. (2003). An efficient decentralized on-line traffic engineering algorithm for MPLS networks. In Charzinski, J., Lehnert, R., and Tran-Gia, P., editors, *Proc. of 18th International TELETRAFFIC CONGRESS - Providing QoS in Heterogeneous Environments*, volume 5a, pages 451–460, Germany.
- Guerin, R. A., Orda, A., and Williams, D. (1997). Qos routing mechanisms and ospf extensions. In *In Proceedings of IEEE GLOBECOM*, pages 1903–1908.
- IEEE (2007). Ieee p802.1qay/d4.5 - draft standard for local ad metropolitan area networks - virtual bridged local area networks - amendment: Provicer backbone bridge traffic engineering. *IEEE Std P802.1Qay/D4.5*.
- Kar, K., Kodialam, M., and Lakshman, T. (2000). Minimum interference routing of bandwidth guaranteed tunnels with mpls traffic engineering applications. *Selected Areas in Communications, IEEE Journal*, 18(12):2566–2579.
- Kodialam, M. and Lakshman, T. (2000). Minimum interference routing with applications to mpls traffic engineering. *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2:884–893.
- Kotti, A., Hamza, R., and Bouleimen, K. (2007). Bandwidth constrained routing algorithm for mpls traffic engineering. In *ICNS '07: Proceedings of the Third International Conference on Networking and Services*, page 20, Washington, DC, USA. IEEE Computer Society.
- Kuipers, F., Van Mieghem, P., Korkmaz, T., and Krunz, M. (2002). An overview of constraint-based path selection algorithms for qos routing. *Communications Magazine, IEEE*, 40(12):50–55.
- Metro Ethernet Forum (2004). Ethernet services definition - phase 1. Disponível em <http://metroethernetforum.org/PDFs/Standards/MEF6.pdf> (Verificado em dez/2008).
- Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA.
- Wang, N., Ho, K., Pavlou, G., and Howarth, M. (2008). An overview of routing optimization for internet traffic engineering. *Communications Surveys & Tutorials, IEEE*, 10(1):36–56.