

Minimização de Fragmentação de Banda em Algoritmos de Roteamento para Engenharia de Tráfego

Marcelo B. Morgade Cortizo , José A. Suruagy Monteiro

¹NUPERC – Universidade Salvador
R. Ponciano de Oliveira, 126 - 41950-275 Salvador, Ba

morgade@gmail.com, suruagy@unifacs.br

Abstract. *Bandwidth fragmentation can occur in connection-oriented networks such as MPLS and ATM due to the order in which connection requests are processed. This problem causes the reduction of the overall number of accepted requests. Assuming that the set of connection requirements are known in advance, this paper presents a low complexity method to minimize the effects of the bandwidth fragmentation problem. This method is based on the solution of linear diophantine equations and executions of the Extended Euclidean Algorithm, computing a fragmentation potential measure for each link. These measures are used to proportionally increase the link weights therefore reducing the fragmentation caused by routing algorithms. This method was applied to MINHOP, and Su and Chen's modified MIRA algorithms. Simulation results showed the benefits of this generic solution by increasing the overall number of accepted requests.*

Resumo. *Fragmentação de banda pode ocorrer em redes orientadas a conexão como MPLS e ATM devido à ordem em que os pedidos de conexão são processados. Este problema causa a redução do número total de requisições que podem ser atendidas. Assumindo que o conjunto de requisitos de banda associados às conexões pode ser pré-determinado, este artigo apresenta um método de baixa complexidade capaz de minimizar os efeitos do problema de fragmentação de banda. Este método é baseado na solução de equações diofantinas lineares e execuções do Algoritmo de Euclides Estendido, sendo capaz de computar uma medida de potencial de fragmentação para cada enlace da rede. Estas medidas são usadas para aumentar proporcionalmente o peso dos enlaces e, conseqüentemente, reduzir a probabilidade dos algoritmos de roteamento usarem estes enlaces de forma a causar fragmentação. Este método foi aplicado ao algoritmo MINHOP e ao algoritmo de interferência mínima de Su e Chen. Resultados obtidos através de simulações mostraram os benefícios desta solução genérica através do aumento na quantidade total de requisições atendidas.*

1. Introdução

O sucesso das redes IP como solução fácil e de baixo custo se consolidou com sua utilização em larga escala no mundo inteiro. O enorme crescimento desta tecnologia criou uma necessidade de melhoria urgente nas técnicas de controle sobre a real capacidade da

rede e sobre sua eficiência no transporte dos fluxos de dados. Neste contexto, o processo de Engenharia de Tráfego (ET) surgiu como uma ferramenta eficiente para o controle do crescimento da rede. Os estudos sobre Engenharia de Tráfego em Redes IP tomam como base as atividades de otimização e avaliação do desempenho de redes em operação, com o objetivo de torná-las mais eficientes, confiáveis, e capazes de viabilizar valores garantidos para indicadores de qualidade de serviço (*QoS*) [Awduche et al., 2002].

Uma das ferramentas mais usadas pelo processo de Engenharia de Tráfego em redes IP é a tecnologia *MultiProtocol Label Switching* (MPLS). O MPLS trouxe para o mundo IP a possibilidade de utilização de circuitos virtuais de maneira semelhante ao modo de trabalho das redes ATM, possibilitando a criação de um modelo sobreposto de caminhos comutados por rótulos ou *Label Switched Paths* (LSPs) sobre os enlaces físicos da rede. O MPLS permite a criação de LSPs dinamicamente a um baixo custo. Estes canais virtuais podem ser estabelecidos e associados a uma determinada medida de banda a ser reservada nos enlaces através da utilização de alguns protocolos de sinalização auxiliares como o *Constraint-Routing Label Distribution Protocol* (CR-LDP) [Jamoussi et al., 2002] e o *Resource Reservation Protocol - Traffic Engineering* (RSVP-TE) [Awduche et al., 2001]. Os fluxos devem ser mapeados em LSPs levando em conta a distribuição otimizada do tráfego, o balanceamento da carga, e restrições pertinentes à garantia de *QoS*. Assim, o papel do MPLS na Engenharia de Tráfego é estabelecer e manter LSPs com recursos reservados através dos protocolos de sinalização, além de implementar a comutação dos pacotes rotulados.

Um momento crítico do processo de Engenharia de Tráfego é a seleção dos enlaces e nós intermediários sobre os quais cada LSP passará e reservará recursos. As diversas medidas relacionadas ao comportamento de uma ou mais fontes de tráfego é um fator que aumenta a dificuldade no processo de definição dos recursos a serem reservados nos LSPs de modo a acomodar os fluxos que por ele passarão. Para simplificar este problema, pode ser usado o conceito de banda efetiva [Kelly, 1996]. A banda efetiva é uma medida que sumariza todas informações estatísticas e necessidades de *QoS* de uma fonte de tráfego num único valor. Assim, é possível definir como a principal regra para o estabelecimento de um LSP, a necessidade de que cada um dos enlaces que o caminho utiliza satisfaça a medida de banda efetiva associada às fontes dos fluxos de tráfego a serem encaminhados pelo mesmo.

O processo de determinação do conjunto de caminhos otimizado dentro da rede envolve um problema NP-completo [Girish et al., 2000]. Além disso, muitas vezes o processo de computar caminhos acontece à medida em que chegam novas demandas, a intervalos relativamente freqüentes, e com um tempo de resposta que deve ser o mais curto possível. Assim, torna-se necessária a criação de processos de baixa complexidade capazes de encontrar conjuntos de caminhos que satisfaçam as restrições e que maximizem a possibilidade de utilização dos recursos da rede.

Diversos algoritmos vêm sendo propostos para a resolução do problema de seleção de caminhos para LSPs. Apesar dos algoritmos se basearem em diferentes critérios, eles sempre têm o primeiro passo em comum: a eliminação dos enlaces com capacidade inferior à da demanda do LSP. Sobre a rede restante, eles aplicam seus critérios de seleção.

O algoritmo de abordagem mais simples é conhecido como MINHOP, que con-

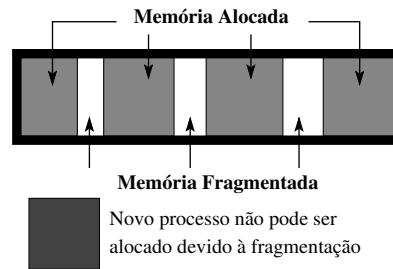


Figura 1: Alocação negada devido à fragmentação da memória

siste apenas na implementação do algoritmo de caminho mais curto. Uma variante deste algoritmo é o *Widest-Shortest Path* (WSP), que usa o caminho com maior capacidade como critério de desempate caso dois ou mais caminhos com o mesmo número de saltos sejam encontrados. O algoritmo *Shortest-Widest Path* (SWP) escolhe o caminho com maior capacidade residual, usando o menor número de saltos como critério de desempate. Estes algoritmos têm a baixa complexidade como vantagem, uma vez que são baseados numa única execução do algoritmo de *Dijkstra* ou de pequenas variações do mesmo [Ma and Steenkiste, 1997].

Existem também algoritmos mais sofisticados, que buscam objetivos diretos como a redução de interferência entre os caminhos. O *Minimum Hop Interference Algorithm* (MIRA) foi o algoritmo que introduziu o conceito de minimização de interferência entre os LSPs [Kodialam and Lakshman, 2000]. O algoritmo apresentado em [Su and Chen, 2002] propôs uma modificação no MIRA de forma a aumentar sua eficiência na determinação de arestas críticas¹. Estes algoritmos apresentam resultados superiores em termos de minimização da quantidade de bloqueios, quando comparados aos algoritmos mais simples. Porém apresentam uma complexidade maior, já que precisam executar procedimentos para cálculo de fluxo máximo. O *Light Minimum Interference Routing* LMIR é uma proposta de algoritmo que consegue resultados semelhantes com menor complexidade computacional [Figueiredo et al., 2004].

Este artigo apresenta o conceito de fragmentação de banda, um fenômeno que pode ocorrer com a sucessiva execução dos algoritmos de roteamento e que diminui a possibilidade de maximizar a utilização dos enlaces. A seção 2 deste artigo apresenta o problema de fragmentação de banda, suas causas e efeitos dentro dos processos de otimização. A seção 3 propõe um método para detecção e minimização do problema de fragmentação de banda em algoritmos de roteamento. A seção 4 mostra resultados de experimentos que mostram os efeitos da aplicação do método de redução da fragmentação em algoritmos de Engenharia de Tráfego.

2. O Problema da Fragmentação de Banda

O problema da fragmentação de banda pode ser comparado com problema de fragmentação de memória existente em sistemas operacionais que não utilizam as técnicas de paginação. A alocação e desalocação de faixas contínuas de memória para processos

¹Este algoritmo não recebeu dos autores nenhum nome ou acrônimo específico. Para facilitar a leitura, este artigo utiliza o termo “SU-CHEN” para se referir ao algoritmo em questão.

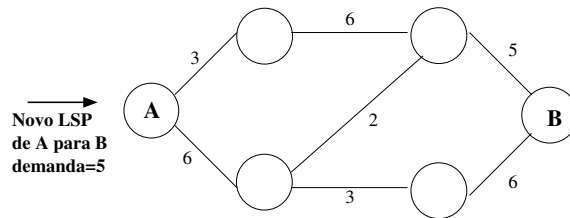


Figura 2: Roteamento prejudicado pela fragmentação

num sistema operacional pode ir criando fragmentos de memória não utilizada. Se um fragmento é pequeno demais para ser alocado a um processo, acaba não tendo utilidade prática dentro do sistema. Quando a memória está repleta destes fragmentos, a porcentagem de memória utilizável cai a um ponto em que o sistema começa a negar pedidos de alocação. Na situação representada na Figura 1, o sistema tem memória livre disponível para o processo, mas ela está tão fragmentada que não pode ser utilizada. Nesta situação, o sistema precisa rodar um algoritmo de desfragmentação da memória.

Uma situação análoga pode ser observada com o problema de roteamento de LSPs. Na topologia mostrada na Figura 2, a capacidade residual dos enlaces mostra que a rede ainda suporta levar um fluxo máximo (*maxflow*) de 8 unidades [Ford and Fulkerson, 1956]. Mas, mesmo com este fluxo total disponível, não há nenhum caminho capaz de suportar o LSP que acaba de chegar com uma demanda associada de 5 unidades de banda. O roteamento não é possível por que este fluxo máximo de 8 unidades disponível passa por enlaces com banda fragmentada que não podem ser usados pelo novo LSP (os enlaces de capacidade residual 2 e 3). Se toda esta banda tivesse outra distribuição pelos enlaces, este novo caminho poderia se estabelecido.

O processo básico de Engenharia de Tráfego definido em [Awduche et al., 2002] propõe um ciclo cuja fase inicial é coletar informações sobre o tráfego de forma a caracterizar os fluxos que passarão pela rede. Após cada execução desta etapa é possível conhecer informações como os possíveis pares origem-destino ou tipos de LSPs que estarão sendo requisitados para a rede. Quando existe o conhecimento prévio sobre o conjunto de demandas de banda associadas aos LSPs que serão estabelecidos na rede durante um dos ciclos do processo de Engenharia de Tráfego é possível perceber os efeitos do fenômeno da fragmentação de banda. Este conjunto de possíveis demandas pode estar associado aos diferentes serviços ou aplicações, por exemplo, que usarão os LSPs da rede.

Para exemplificar a formação de fragmentação em um cenário bastante simples, supõe-se que em uma rede corporativa a fase inicial do processo de Engenharia de Tráfego estudou os fluxos importantes e definiu que os LSPs a serem requisitados servirão para sessões de voz sobre IP usando o codec G.729 (8kbps) e sessões de vídeo em tempo real a uma vazão média de 28kbps. Neste caso, qualquer enlace que tenha banda residual menor que 8kbps será inútil para o estabelecimento de LSPs, pois não atenderá a mais nenhuma demanda. Neste cenário, o somatório das bandas residuais de todos os enlaces com banda residual menor que 8kbps representa o total de banda fragmentada. Este exemplo com LSPs de baixa granularidade é uma forma simples de observar o problema. Porém, ele pode ser detectado em qualquer rede onde o ciclo do processo de Engenharia de Tráfego coleta informações que permitem definir conjuntos de demandas associadas ao LSPs que

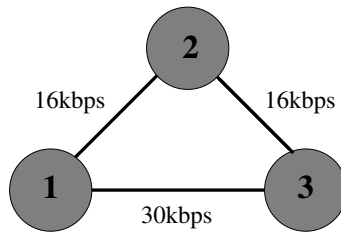


Figura 3: Cenário onde ocorre fragmentação de banda

serão estabelecidos.

Considerando a topologia simples mostrada na Figura 3 onde chegam requisições de 8kbps e 28kbps do nó 1 ao nó 3, e considerando que a primeira requisição é um LSP de 8kbps, é possível observar que todos os algoritmos de Engenharia de Tráfego descritos na seção anterior elegeriam o caminho 1-3 para criar o LSP. Isto aconteceria porque este é o caminho mais curto e com mais banda disponível. Esta escolha, deixaria o enlace 1-3 com uma banda residual de 22kbps.

Com este valor residual, o enlace suportará apenas mais dois LSPs de 8kbps e terminará com 6kbps de banda fragmentada. Além disso, se a segunda requisição for de um LSP de 28kbps, esta já será bloqueada. A rede passará a bloquear todas as requisições entre os pares de entrada e saída quando aproximadamente 87% do fluxo máximo inicial entre estes nós for utilizado.

Porém, se esta primeira requisição utilizasse o caminho 1-2-3, a banda residual dos enlaces 1-2 e 2-3 seria de 8kbps. Estes enlaces poderiam ainda suportar mais um LSP de 8kbps e não gerariam banda fragmentada. A solução que minimizaria a fragmentação e maximizaria a utilização dos enlaces neste caso seria passar dois LSPs de 8kbps pelo caminho 1-2-3 e um LSP de 28kbps pelo caminho 1-3. Essa solução geraria apenas 2kbps de banda fragmentada. Assim a rede começaria a bloquear todas as requisições entre os pares de entrada e saída somente quando aproximadamente 96% do fluxo máximo inicial entre estes nós for utilizado.

Um detalhe importante é que como cada enlace contribui individualmente no total de banda fragmentada, em redes densas o problema tende a se agravar. Isso significa que uma quantidade de banda muito alta pode ficar sem utilização devido à fragmentação nos inúmeros enlaces da rede.

Nos estudos para avaliação do desempenho de algoritmos de roteamento, o problema da fragmentação não é detectado porque muitas vezes as requisições podem receber valores unitários, e acabam consumindo toda a banda dos enlaces. Os experimentos feitos em [Su and Chen, 2002] geraram o gráfico da Figura 4 no qual todo o fluxo da rede é consumido. Isto acontece porque as requisições assumem valores de 1 a 4. Maiores detalhes sobre a reprodução dos experimentos estão na seção 4.

Porém, em casos como o que foi apresentado, nos quais as requisições têm valores de banda maiores, a fragmentação é um fenômeno real que provoca a diminuição da utilização máxima dos recursos. Um exemplo disto é o gráfico da Figura 5. Este gráfico foi gerado reproduzindo o mesmo experimento feito em [Su and Chen, 2002] com os valores de requisições assumindo os valores 300 e 400. Neste caso, é possível notar que

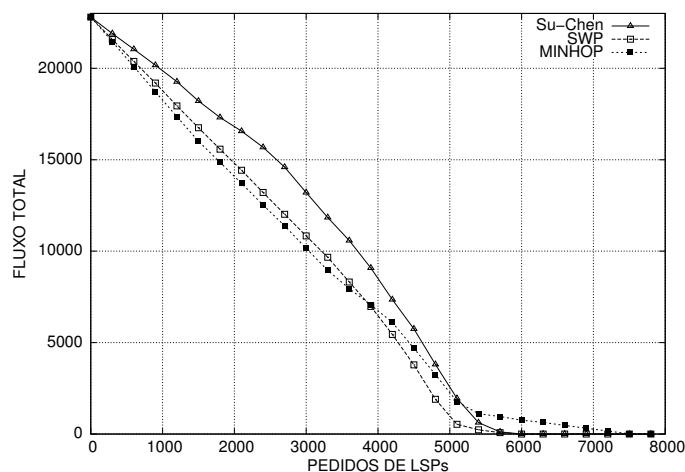


Figura 4: Reprodução do experimento feito em [Su and Chen, 2002]

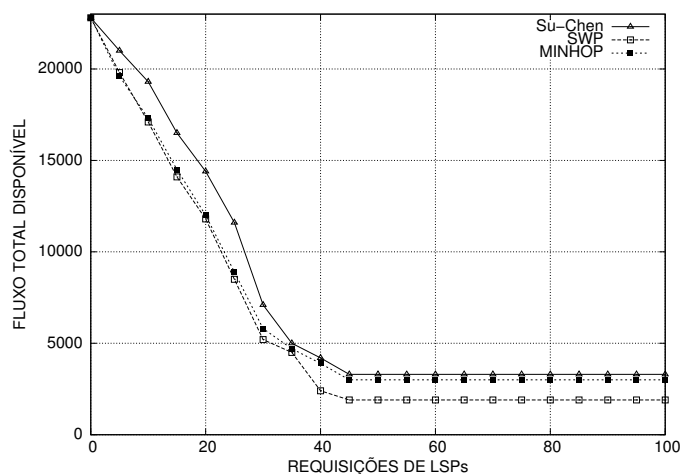


Figura 5: Reprodução do experimento assumindo requisições de 300 e 400.

todas as requisições foram bloqueadas após a requisição 40, mesmo ainda havendo um fluxo total entre 2000 e 4000. Toda esta banda não pôde ser utilizada justamente por estar fragmentada pela rede.

Num sistema operacional, um algoritmo de desfragmentação da memória pode redistribuir as áreas alocadas, possibilitando o acolhimento de novos processos. Porém, o processo análogo em uma rede passaria pelo problema de roteamento, o qual é custoso e por isso é muitas vezes utilizado apenas no caso de falhas em enlaces. Assim, a maneira mais eficiente seria desenvolver um algoritmo que roteasse as requisições já tentando evitar o problema de fragmentação.

Os algoritmos apresentados na seção anterior não podem levar em conta o problema da fragmentação pois não usam o conjunto dos possíveis valores de banda LSPs como entrada.

A próxima seção propõe um método para detecção da probabilidade de fragmentação de banda, minimização do total de banda fragmentada em enlaces e a sua aplicação pelos algoritmos de Engenharia de Tráfego.

3. Detecção e Minimização da Fragmentação de Banda

Aplicar um método para minimização da fragmentação num algoritmo de roteamento pode ser visto como um processo de escolha de caminhos de modo que o conjunto de enlaces e bandas remanescentes tenham o menor potencial de fragmentação após o estabelecimento do LSP. Assim, é necessário desenvolver um método que defina o potencial de fragmentação de um enlace, dado um valor de banda residual e um conjunto de possíveis requisições.

Os valores deste potencial de fragmentação podem ser utilizados como conjunto de pesos utilizados no algoritmo de *Dijkstra*. A vantagem desta abordagem é que como a maioria dos algoritmos de roteamento utilizam o *Dijkstra*, é possível combinar critérios de roteamento como a interferência mínima com o conceito de minimização da fragmentação de banda. O SU-CHEN tem como passo intermediário a determinação de um conjunto de pesos que refletem a criticalidade do enlace. O processo de minimização da fragmentação pode entrar neste ponto e incrementar em valores altos os pesos dos enlaces de acordo com os potenciais de fragmentação calculados para cada um.

O problema do cálculo do potencial de fragmentação em um enlace pode ser instanciado para um caso semelhante ao mostrado na seção 2 da seguinte forma: Dado um enlace com capacidade residual igual a 38kbps e uma requisição de 8 kpbs, é possível esgotar a banda restante somente com LSPs de 8 e 28kbps? Ou seja, é possível utilizar totalmente 30kbps (38kbps - 8kbps) de banda consumindo esta banda em partes de 8 e 28kbps? Expressando em forma matemática, o problema seria descobrir se existe uma ou mais soluções para a equação $8x + 28y = 30$ com x e $y \in N$. Na forma geral, $\exists x_1, x_2, \dots, x_n \in N \mid a_1x_1 + a_2x_2 + \dots + a_nx_n = b$ considerando $a_1, a_2, \dots, a_n \in N$ e $b \in N$.

Problemas que envolvem o cálculo de soluções inteiras para sistemas de equações com uma ou mais incógnitas são estudados pela Álgebra dentro da Teoria dos Números. O problema definido acima é conhecido como equação diofantina linear [Weissten, 2004].

Equações diofantinas lineares podem não ter solução. Um exemplo simples que mostra esta possibilidade é a equação $2x + 4y = 7$. Este é um caso em que não existem combinações de valores inteiros para x e y que tornem a equação verdadeira. Aplicando ao problema de fragmentação de banda, é possível afirmar que enlaces com banda residual igual a 7 estão fadados a deixar banda fragmentada caso a rede aceite apenas LSPs de tamanho 2 e 4. Assim, o problema da detecção do potencial de fragmentação de banda em um enlace deve envolver um método que defina se uma determinada equação diofantina linear tem solução dentro do conjunto de números naturais.

O matemático Diofante, que propôs este tipo de problema, estudou diversos métodos para resolução destas equações dentro do conjunto de números inteiros. Para equações diofantinas lineares com duas incógnitas no formato $ax + by = c$, a Identidade de Bezout mostra que a equação comporta soluções inteiras se e somente se o máximo divisor comum entre a e b divide c . Ou seja, dados $a, b, c \in Z$:

$$\{\exists x, y \in Z \mid ax + by = c\} \iff c \bmod \text{mdc}(a, b) = 0 \quad (1)$$

Como na equação 1, x e y devem representar quantidades de LSPs a serem estabe-

lecidos, somente soluções em N devem ser consideradas. Assim, mesmo que a condição para existência da solução em Z seja verdadeira, é preciso desenvolver um método para encontrar a solução geral e verificar se ela admite pelo menos uma solução onde x e y são naturais.

O algoritmo 1 mostra a implementação do Algoritmo Estendido de Euclides², que é um método eficiente capaz de encontrar uma solução específica para equações diofânticas lineares com duas incógnitas, caso esta exista. A partir desta solução específica, é possível encontrar a solução geral.

Algoritmo 1 Euclides Estendido

Dados $a, b \in N$ onde $a > b$

- 1: $m = a, n = b, r = 1$
 - 2: Definir *passos* como uma pilha
 - 3: **enquanto** $r \neq 0$ **faça**
 - 4: $r = m \bmod n$
 - 5: Empilhar “ $r = m - n(m \text{ div } n)$ ” em *passos*
 - 6: $m = n$
 - 7: $n = r$
 - 8: **fim enquanto**
 - 9: $MDC = m$
 - 10: Desempilhar um elemento em *passos* e descartar
 - 11: Desempilhar um elemento em *passos* e guardar em eq_1
 - 12: **enquanto** houver elemento em *passos* **faça**
 - 13: Desempilhar elemento em *passos* e guardar em eq_2
 - 14: Tomar eq_1 como “ $r_1 = \alpha_1 - \beta_1(\mu_1)$ ” e eq_2 como “ $r_2 = \alpha_2 - \beta_2(\mu_2)$ ”.
 - 15: $\alpha_n = -\alpha_2\mu_1, \beta_n = \beta_2, \mu_n = -(\mu_1\mu_2 + \frac{\alpha_1}{\alpha_2})$
 - 16: Definir eq_1 como “ $r_1 = \alpha_n - \beta_n(\mu_n)$ ”
 - 17: **fim enquanto**
 - 18: $x_0 = \alpha_1/a$
 - 19: $y_0 = -\mu_1$
-

Uma vez encontrados o x_0 e o y_0 com o algoritmo 1, e caso o mdc encontrado divida c , é possível definir $w = \frac{c}{\text{mdc}(a,b)}$. Assim dado $d = \text{mdc}(a, b)$, a solução geral pode ser dada como:

$$\begin{aligned} x &= x_0w + (b/d)t \\ y &= y_0w - (a/d)t \quad t \in Z. \end{aligned} \tag{2}$$

Como esta solução geral admite valores em Z , não pode ser aplicada diretamente ao problema de fragmentação de banda. Porém, com esta solução geral em Z é possível verificar se existem soluções em N . Para isto, basta descobrir o maior valor de t que deixe y positivo. Este t máximo se encontra definindo:

$$t_{max} = \left\lfloor \frac{y_0w}{b/d} \right\rfloor. \tag{3}$$

²Como esta implementação é usada para cálculo de potencial de fragmentação, ela recebe como entrada somente valores de a e $b \in N$ pois refletem valores de demanda de banda que são sempre positivos.

Se este valor de t_{max} aplicado à solução geral de x não resultar num número positivo, não existirá possibilidade de solução em N . Se não existir solução em N , o enlace analisado está fadado a gerar fragmentação caso tenha banda residual igual a c .

O potencial de fragmentação é zero caso exista solução para o problema. Se $c \bmod \text{mdc}(a, b) \neq 0$, o potencial de fragmentação pode ser tomado de forma geral como $c \bmod \text{mdc}(a, b)$. Se $c \bmod \text{mdc}(a, b) = 0$ e a equação diofantina não tem solução em N , o potencial de fragmentação pode ser tomado de forma geral como $\min(a \bmod c, b \bmod c)$. Os potenciais calculados acima de zero não são completamente precisos, pois o algoritmo de Euclides deveria ser rodado novamente para definir o maior valor abaixo de c que não gera fragmentação. Porém, esse aumento no tempo de processamento não valeria a pena pois qualquer valor maior que zero já é suficiente para explodir o peso do enlace e evitar ao máximo que o LSP passe sobre ele.

É interessante ressaltar que a solução geral encontrada com o algoritmo estendido de Euclides define uma proporção entre x e y . Assim, se é conhecida também a proporção entre LSPs de valor a e b que serão estabelecidos, existe a opção de determinar um potencial de fragmentação entre 0 e 1 se a solução geral indicar uma proporção entre x e y muito diferente da prevista.

Este método demonstrado encontra soluções para equações de apenas duas incógnitas. Mas, a solução de equações diofantinas com mais incógnitas também podem ser encontradas. A existência de solução para uma equação diofantina com n variáveis ($n > 2$) pode depender de operações com $n - 1$ destas variáveis.

Numa equação com 3 incógnitas no formato $ax + by + cz = d$ é possível fatorar as duas primeiras incógnitas e definir um termo w usado para reduzir a equação:

$$(a, b) \left(\frac{a}{(a, b)}x + \frac{b}{(a, b)}y \right) + cz = d. \quad (4)$$

$$w = \left(\frac{a}{(a, b)}x + \frac{b}{(a, b)}y \right). \quad (5)$$

$$(a, b)w + cz = d. \quad (6)$$

A solução da equação 6 pode ser verificada com o Euclides Estendido. A equação da solução geral de w deve então ser expandida para os valores originais de x e y , encontrando-se uma nova equação diofantina com duas incógnitas. Se mais incógnitas forem adicionadas, basta repetir a fatoração. O crescimento do número de incógnitas aumenta também o tempo total de definição do potencial de fragmentação. Mas, nem sempre é preciso adicionar todos os possíveis valores de LSPs na equação. Valores de demanda que são múltiplos de outros possíveis valores podem ser descartados pois não farão diferença na minimização da fragmentação, assim como LSPs que aparecem com pouca frequência. Os valores de demanda mais importantes são os menores valores não múltiplos entre si.

Fica clara a importância do estudo do conjunto de demandas baseado no conjunto de serviços ou aplicações que utilizarão a rede. Este estudo deve definir uma equação

diofantina linear modelo para cálculo dos potenciais de fragmentação, ou pode até mesmo decidir que a minimização da fragmentação não poderá ajudar (quando, por exemplo, todos valores de demanda forem múltiplos do menor valor do conjunto). Este estudo deve ser embutido dentro das primeiras fases do ciclo do processo de Engenharia de Tráfego e será a entrada principal para o método de minimização da fragmentação de banda.

O método proposto neste artigo para redução da fragmentação de banda consiste no cálculo do potencial de fragmentação para cada enlace, e posterior utilização de cada valor como fator de multiplicação dos pesos pré-calculados dos enlaces associados que serão utilizados como entrada para o algoritmo de *Dijkstra*.

Algoritmo 2 MINFRAG

Dados $a, b \in N(a > b)$ como os valores de LSP d como a nova demanda a rotear

- 1: Calcular $peso_n$ para os n enlaces a depender do critério de um algoritmo de roteamento
 - 2: $M = \sum peso_i$
 - 3: **para todo** enlace e na rede no qual $banda_e \geq d$ **faça**
 - 4: $\rho = 0, res = d - banda_e$
 - 5: **se** $res < b$ **e** $res > 0$ **então**
 - 6: $\rho = res$
 - 7: **senão**
 - 8: Encontrar $d = \text{mdc}(a, b), x_0, y_0$ chamando Euclides Estendido
 - 9: **se** $res \bmod d \neq 0$ **então**
 - 10: $\rho = (res \bmod d)$
 - 11: **senão**
 - 12: $t_{max} = \lfloor \frac{y_0 w}{b/d} \rfloor$
 - 13: **se** $(x_0 w + (b/d)t_{max}) < 0$ **então**
 - 14: $\rho = \min(a \bmod c, b \bmod c)$
 - 15: **fim se**
 - 16: **fim se**
 - 17: **fim se**
 - 18: $peso_e = peso_e + M\rho$
 - 19: **fim para**
 - 20: Executar o algoritmo de *Dijkstra* usando $peso_n$ para cada enlace n
-

Um algoritmo de interferência mínima pode ser utilizado para encontrar um conjunto de pesos associados à criticalidades dos enlaces, e posteriormente o método de minimização da fragmentação multiplicaria estes pesos por um fator que reflete o potencial de fragmentação de cada enlace depois da hipotética passagem do novo LSP. Assim, o algoritmo de *Dijkstra* encontraria sempre o caminho com menor potencial de fragmentação e menor interferência.

Para o processo de minimização não interferir significativamente no tempo de resposta do roteamento, basta que a sua complexidade seja menor que a complexidade total do algoritmo de roteamento que está sendo usado. A complexidade dos algoritmos de roteamento simples é praticamente a mesma do algoritmo de *Dijkstra*, já que quase todos

estes processos são baseados apenas nele. Definindo que os algoritmos trabalham numa rede que forma um grafo de V vértices e E arestas, pode se afirmar que a complexidade do algoritmo de *Dijkstra* é $O(V^2 + E)$ [Dijkstra, 1959]. Os algoritmos de roteamento com interferência mínima têm na etapa de cálculo do *maxflow*, o maior fator de aumento na complexidade. A depender do algoritmo usado e da relação entre o número de arestas e o número de nós essa complexidade varia, mantendo-se porém sempre acima da complexidade do algoritmo de *Dijkstra*. A implementação do algoritmo de *maxflow* de *Edmonds-Karp* apresenta complexidade de $O(V.E^2)$ [Cormem et al., 1990]. Em redes com uma maior proporção de enlaces em relação aos nós (como a Internet), percebe-se que esta complexidade é bem maior que a do algoritmo de *Dijkstra*. O algoritmo de Goldberg, que é uma implementação mais eficiente no tempo de cálculo do *maxflow*, apresenta uma complexidade $O(\min(V^{\frac{2}{3}}, E^{\frac{1}{2}}).E. \log(\frac{V^2}{E}). \log(U))$ numa rede com capacidades no intervalo $[1, U]$ [Goldberg and Tarjan, 1988]. O algoritmo LMIR é um caso particular que não executa cálculo de *maxflow*, usando um processo de busca de caminhos de menor capacidade que apresenta complexidade $O(V^2)$ [Figueiredo et al., 2004].

A complexidade do MINFRAG está ligada ao cálculo do potencial de fragmentação para cada enlace E da rede. O algoritmo de cálculo do potencial de fragmentação de um enlace tem a mesma complexidade do algoritmo de Euclides Estendido. Em Euclides, para $b \leq a$, o número máximo de divisões que ocorre é $O(\log(b))$ [Cormem et al., 1990]. Quando há n incógnitas na equação diofantina, no pior caso, o algoritmo de Euclides deve ser rodado $n - 1$ vezes para decidir se existe ou não solução para a equação. Assim, pode ser definida a complexidade total do processo como $O(E.(n - 1). \log(b))$. Comparando a complexidade do processo de minimização da fragmentação com a complexidade total dos algoritmos de roteamento, é possível perceber que o MINFRAG não prejudica consideravelmente o tempo total do processo de roteamento. Isso possibilita a real aplicação do processo de minimização da fragmentação em esquemas de roteamento *online*.

A próxima seção apresenta resultados de simulações que mostram o efeito da utilização da redução de fragmentação de banda em alguns algoritmos de roteamento utilizados pela Engenharia de Tráfego.

4. Experimentos

Os experimentos foram feitos usando o simulador *Network Simulator 2* (ns2). Os algoritmos MINHOP, SU-CHEN, e o SWP precisaram ser implementados como extensões do simulador. Para validar estas novas implementações, foi fielmente reproduzido o experimento especificado em [Su and Chen, 2002], obtendo-se o gráfico mostrado na Figura 4. A comparação com o experimento original mostra que os resultados são equivalentes. Foi implementado também o Algoritmo Estendido de Euclides para determinação do potencial de fragmentação de banda em um enlace. As simulações utilizaram a mesma topologia do experimento de validação mostrada na Figura 6, onde cada enlace escuro tem 4800 unidades de banda e cada enlace claro tem 1200 unidades. A figura também mostra os pares de entrada e saída de onde partem e chegam as requisições, e sobre os quais é calculado o fluxo total. Detalhes do cenário podem ser encontrados em [Su and Chen, 2002].

O experimento constou de simulações envolvendo equações diofantinas modelos

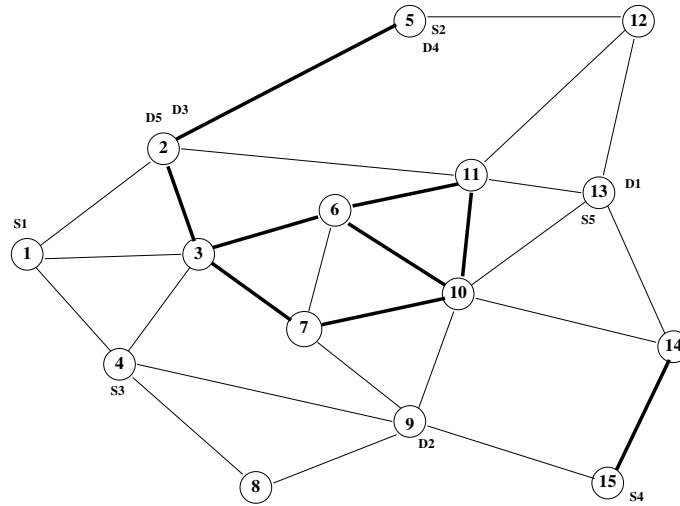


Figura 6: Topologia usada nos experimentos

com 2 e 3 incógnitas. O primeiro experimento utilizou duas incógnitas com requisições de 300 e 400 unidades em distribuição normal, as quais deviam ser roteadas entre os pares de origem-destino escolhidos também em distribuição normal. O segundo experimento seguiu a mesma configuração, porém com as requisições assumindo os valores 300, 400 e 500. Os algoritmos MINHOP e SU-CHEN foram implementados de modo a poderem ou não utilizar o código de redução da fragmentação. Eles foram utilizados para testar os efeitos da aplicação do MINFRAG.

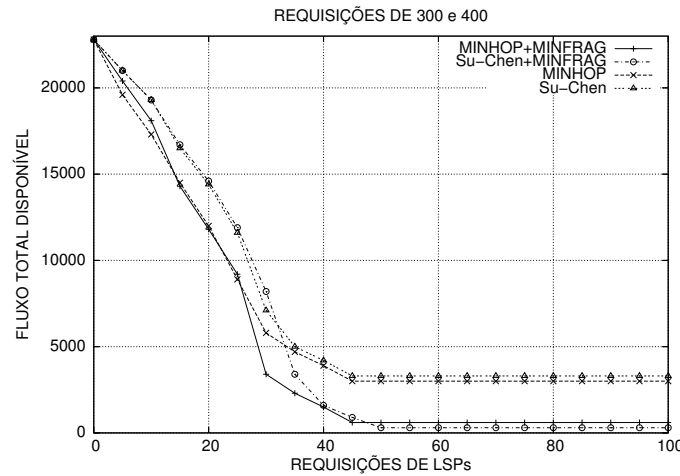


Figura 7: Redução do Fluxo

A Figura 7 mostra a variação do fluxo total entre os pares de entrada-saída com a chegada de requisições quando os algoritmos são executados com e sem a minimização de fragmentação. É possível observar que todos os algoritmos começam a negar todas as requisições mesmo ainda havendo uma boa quantidade de fluxo entre os pares de entrada-saída quando não há redução de fragmentação.

As simulações em que os algoritmos utilizam a minimização de fragmentação mostram que os mesmos conseguem utilizar quase todo o fluxo disponível, aumentando a

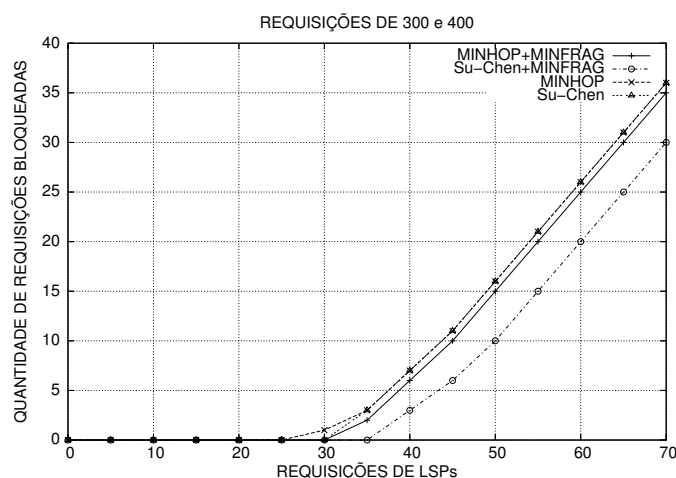


Figura 8: Bloqueios

Tabela 1: Comparativo de bloqueios após 70 requisições (Requisições de 300 e 400)

Algoritmo	Total de Bloqueios	Primeiro bloqueio
MINHOP	41	após 30 requisições
Su-Chen	36	após 35 requisições
MINHOP+MINFRAG	35	após 35 requisições
Su-Chen+MINFRAG	30	após 40 requisições

quantidade de banda total que pôde ser alocada e o número de requisições atendidas.

Uma consequência direta disso é o que mostra a Figura 8. A combinação do SU-CHEN com a minimização da fragmentação produziu o menor número de bloqueios e a maior quantidade de requisições atendidas. Além disso, a Tabela 1 mostra também que o momento do primeiro bloqueio foi atrasado quando houve a utilização do MINFRAG. Outra característica importante a ressaltar é que a minimização da fragmentação só começa a modificar significativamente o comportamento das curvas no momento em que a rede está próxima da saturação, garantindo que o critério inicial do algoritmo de roteamento seja respeitado o máximo possível.

O experimento com requisições 300, 400 e 500 foi realizado para demonstrar a utilização do processo em situações onde a equação diofantina de modelo tem mais de duas incógnitas. Os resultados mostraram também melhorias semelhantes no desempenho dos algoritmos como mostra a Tabela 2.

Tabela 2: Comparativo de bloqueios após 70 requisições (Requisições de 300 e 400 e 500)

Algoritmo	Total de Bloqueios	Primeiro bloqueio
MINHOP	44	após 25 requisições
Su-Chen	43	após 30 requisições
MINHOP+MINFRAG	43	após 30 requisições
Su-Chen+MINFRAG	41	após 35 requisições

5. Conclusões

Este artigo propôs a discussão sobre um problema chamado Fragmentação de Banda, o qual promove uma diminuição da capacidade de utilização total dos recursos de uma rede durante o processo de alocação de banda em enlaces para canais virtuais. Um método para detecção e minimização da fragmentação durante o roteamento também foi proposto baseado na possibilidade de conhecimento do conjunto de possíveis demandas associadas às requisições. Resultados de experimentos mostraram a importância do fenômeno da fragmentação de banda, e como o processo de minimização é capaz de atenuar os efeitos indesejáveis deste problema com uma complexidade computacional relativamente baixa.

Referências

- Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and Xiao, X. (2002). Overview and principles of Internet Traffic Engineering – RFC3272. Relatório técnico, IETF.
- Awduche, D., Gan, D.-H., Li, T., Swallow, G., and Srinivasan, V. (2001). Extensions to RSVP for Traffic Engineering – RFC3209. Relatório técnico, IETF.
- Cormen, T. H., Leiserson, C. E., and Rivest, R. L. (1990). *Introduction to Algorithms*. MIT Press and McGraw Hill.
- Dijkstra, E. W. (1959). A note on two problems in connection with graphs. *Numerische Mathematik*, v. 1, p. 269–271.
- Figueiredo, G. B., da Fonseca, N. L. S., and Monteiro, J. A. S. (2004). Light Minimum Interference Routing. *The IEEE International Conference on Communications (ICC)*.
- Ford, L. R. and Fulkerson, D. R. (1956). Maximal flow through a network. *Canadian Journal of Mathematics*, v. 8, p. 399–404.
- Girish, M. K., Zhou, B., and Hu, J.-Q. (2000). Formulation of the Traffic Engineering Problem. *IEEE ISCC*, p. 214–219.
- Goldberg, A. V. and Tarjan, R. E. (1988). A new approach to the maximum flow problem. *Journal ACM*, v. 35, p. 921–940.
- Jamoussi, B., Andersson, L., Callon, R., Dantu, R., Wu, L., Doolan, P., Worster, T., Feldman, N., Girish, A. F. M., Gray, E., Heinanen, J., Kilty, T., and Malis, A. (2002). Constraint-Based LSP Setup using LDP – RFC3212. Relatório técnico, IETF.
- Kelly, F. (1996). Notes on effective bandwidths. In *Stochastic Networks: Theory and Applications*, p. 141–168. Oxford University Press.
- Kodialam, M. S. and Lakshman, T. V. (2000). Minimum Interference Routing with Applications to MPLS Traffic Engineering. In *INFOCOM (2)*, p. 884–893.
- Ma, Q. and Steenkiste, P. (1997). Quality-of-Service Routing for Traffic with Performance Guarantees. *International Workshop on Quality of Service*.
- Su, B. W. X. and Chen, C. P. (2002). A new bandwidth guaranteed routing algorithm for MPLS Traffic Engineering. In *Proceedings of IEEE International Conference on Communications*, p. 1001–1005.
- Weissten, E. W. (2004). Diophantine Equation. MathWorld – <http://mathworld.wolfram.com/DiophantineEquation.html>.