Algoritmos para Solução do Problema de Dimensionamento de *Buffers* em Roteadores IP

Emilio C. G. Wille[1], **Eduardo Yabcznski**[1], **Clovis R. da Costa Bento**[1]

[1]Universidade Tecnológica Federal do Paraná - UTFPR
Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial - CPGEI
Av. Sete de Setembro 3165, Curitiba (PR), Brasil

ewille@utfpr.edu.br, edu_yab@hotmail.com, cloviscb@gmail.com

Abstract. *Internet routers were widely believed to need large buffers. A widely-used rule-of-thumb states that, because of the dynamics of TCP's congestion control mechanism, a router needs a bandwidth-delay product of buffering, $B = RTT.C$, in order to fully utilize bottleneck links. In this paper we argue that the buffer dimensioning can not be done in isolation, we show the related problem formulation and propose a new solution method.*

Resumo. *Acredita-se, em geral, que os roteadores necessitam de grande espaço para armazenamento de pacotes. Uma conhecida regra de algibeira, oriunda da dinâmica do protocolo TCP, afirma que cada enlace necessita de um buffer de tamanho $B = RTT.C$. Argumenta-se, neste trabalho, que a abordagem tradicional, que envolve o dimensionamento isolado de cada buffer na rede não leva ao melhor projeto. Desta forma, o artigo apresenta uma formulação para o problema, e uma nova metodologia para sua solução. Resultados analíticos são apresentados, e uma verificação é realizada com base em simulações conduzidas com o software ns-2.*

1. Introdução

Os *buffers*, presentes nos roteadores das redes de chaveamento de pacotes, possuem um papel importante. Eles possibilitam uma acomodação das rajadas (*bursts*) de tráfego de modo a impedir ou minimizar as perdas de pacotes e, também, mantêm uma reserva de pacotes de tal forma a manter a vazão dos enlaces (*links*) de saída do roteador.

Atualmente os buffers são dimensionados com base na conhecida regra de "algibeira" que afirma que cada enlace necessita de um buffer de tamanho $B = RTT.C$, onde RTT é o *round-trip time* médio de um fluxo que atravessa o enlace, e C é a capacidade da interface de rede do roteador [Villamizar e Song 1994]. Nota-se que a dimensão do buffer cresce linearmente com a capacidade.

Os operadores solicitam, aos fabricantes de equipamentos de redes, roteadores com 250 ms (ou mais) de armazenamento de pacotes [Bush e Meyer 2003]. Tais buffers, de tamanho tão elevado, impõem grandes desafios para os fabricantes de equipamentos quanto ao projeto e desenvolvimento de roteadores de alta capacidade (pelo uso de numerosas e lentas RAMs dinâmicas) [Appenzeller et al. 2004]. Em segunda análise, buffers grandes são conflitantes com as baixas latências desejadas para aplicações tempo-real tão presentes na Internet.

Esta regra de algibeira surgiu da dinâmica do algoritmo de controle de congestionamento do protocolo TCP. Em particular, um simples fluxo TCP atravessando um enlace congestionado requer um buffer cuja dimensão é igual ao produto capacidade-atraso, de modo a prevenir o enlace de ficar ocioso e então perder vazão.

Em [Appenzeller et al. 2004], os autores argumentam que a regra $RTT.C$ está incorreta e ultrapassada para o dimensionamento dos buffers nos roteadores, propondo a regra $B = RTT.C/\sqrt{N}$ para um enlace alimentado por N fluxos TCP. A regra proposta considera que os múltiplos fluxos TCP (persistentes) que compartilham simultaneamente os enlaces não estão sincronizados. Neste caso o comportamento da janela do protocolo TCP de cada fluxo (modelado por uma função dente-de-serra) é considerado um processo independente. Pelo uso do teorema do limite central a soma destes processos independentes converge a um processo gaussiano. Por conseqüência, espera-se que as necessidades em termos de tamanho de buffer tornem-se menores à medida que aumenta o número de fluxos TCP presentes.

Nestes artigos os modelos empregados consideram um cenário simplificado, onde os fluxos atravessam somente um roteador congestionado. Este não é exatamente o caso em uma rede real, onde agregados de fluxos com diversas destinações, e com diversas necessidades em termos de QoS, compartilham os enlaces. Argumentamos, neste trabalho, que a abordagem tradicional, que envolve o dimensionamento isolado de cada buffer na rede (levando em consideração apenas os valores de RTT e C), não leva ao melhor projeto. A teoria necessita de uma abordagem que seja capaz de considerar toda a rede e distribuir de forma adequada os tamanhos de buffer de forma a satisfazer as necessidades de desempenho dos agregados de fluxos e, ao mesmo tempo, produzir um projeto final de menor custo possível. Este artigo apresenta uma formulação (já proposta anteriormente) para o problema, e uma nova metodologia para sua solução. Resultados analíticos são apresentados, e uma verificação é realizada com base em simulações conduzidas com o software *ns-2*.

2. Conceitos Preliminares

2.1. Metodologia de Projeto de Redes IP

Em [Wille et al. 2004, Wille et al. 2005] os autores propõem uma metodologia de projeto de redes IP que considera a dinâmica das rede de pacotes, assim como os efeitos dos protocolos nas diferentes camadas e na QoS experimentada pelos usuários finais. A Figura 1 mostra o diagrama que representa a metodologia, onde se observam dois grandes blocos (Tradutor de QoS e Procedimentos de Otimização). Como restrições de entrada será considerado, para todo par origem-destino, as especificações de QoS da camada de transporte (latência e vazão). Após passar pelo tradutor de QoS, as restrições de QoS são mapeadas em restrições de camadas inferiores até a camada de rede (atraso médio e probabilidade de perda de pacotes). Este processo de mapeamento é então usado em conjunto com um modelo de tráfego TCP/IP, proposto em [Garetto e Towsley 2003], que produz uma boa estimativa do desempenho da rede sujeita à padrões de tráfego real). Em segundo lugar, o dimensionamento ocorre via processos de otimização, onde se busca minimizar uma função custo e ao mesmo tempo satisfazer uma série de restrições associadas ao desempenho da rede. A Figura mostra sub-blocos referentes à Atribuição de Capacidades (CA) e ao Dimensionamento de Buffers (BA).

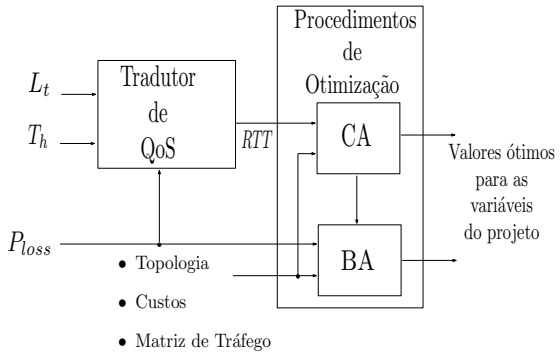


Figura 1. Diagrama de blocos da metodologia empregada no trabalho.

2.2. Modelo de Redes IP

A infraestrutura de rede IP será representada por um grafo direcionado $G = (V, E)$, onde V é um conjunto de nós (com cardinalidade N) e E é um conjunto de arestas (com cardinalidade L). Um caminho é uma seqüência $(v_1, e_1, v_2, e_2, \cdots, v_{n-1}, e_l, v_n)$ de nós e arestas. As extremidades da aresta $e_l \in E$ são os nós $(v_n, v_{n+1}) \in V$. Um nó representa um roteador e uma aresta representa um enlace conectando dois roteadores. Cada interface de saída (com seus respectivos *buffers*) de cada roteador, será modelada como uma fila.

O tráfego para cada par origem(s)-destino(d), é transmitido pela rede através de um único caminho (roteamento não bifurcado), escolhido por um algoritmo de roteamento fixo. Isto permite que o fluxo f_{ij} em cada enlace (i, j), seja facilmente obtido através das tabelas de roteamento e da matriz de tráfego $\widehat{\Gamma}$. O fluxo f_{ij} é definido como a quantidade de informação que trafega (é transmitida) no enlace (i, j). A matriz de tráfego $\widehat{\Gamma} = \{\widehat{\gamma}_{sd}\}$ é obtida do tráfego requisitado entre os nós, onde $\widehat{\gamma}_{sd}$ é a taxa de transmissão do nó de origem s ao nó de destino d.

Tanto o fluxo f_{ij}, quanto a capacidade C_{ij}, definida como a máxima quantidade de informação que pode ser transmitida pelo enlace (i, j), são dados em bits por segundo (bps). Cada *buffer* de cada interface de saída de cada roteador, pode suportar no máximo B_{ij} pacotes e d_{ij} será o comprimento físico do enlace (i, j).

2.3. Modelo de Fila $M_{[X]}/M/1/B$

Para obter uma formulação útil para o problema de projetos de redes é necessário que essa formulação expresse, com a maior precisão possível, as métricas de interesse, e ao mesmo tempo apresente baixa complexidade. Como o modelo clássico de filas $M/M/1/B$ [Kleinrock 1976] não é capaz modelar a correlação existente entre pacotes que chegam nas interfaces dos roteadores em instantes diferentes, é adotado o modelo de filas $M_{[X]}/M/1/B$, que melhor representa o tráfego em rajadas produzido pelo protocolo TCP. Este modelo representa uma fila Markoviana com *chegadas em grupo*. O tamanho do grupo varia entre 1 e W com distribuição $[X]$, onde W é o tamanho máximo (em segmentos) da janela TCP. A distribuição $[X]$ é obtida considerando o número de segmentos que a fonte TCP envia em um RTT. Dada a distribuição de comprimento dos fluxos, é possível calcular a distribuição de tamanho de pacotes $[X]$, através do modelo estocástico do TCP presente em [Garetto e Towsley 2003]. O cálculo do atraso médio dos pacotes

$E[T]$, do número médio de pacotes na fila $E[N]$, e da probabilidade de perda de pacotes, é realizado com base na solução da Cadeia de Markov em Tempo Contínuo (CTMC - *Continuous Time Markov Chain*) que modela a fila [Chao et al. 1999].

O modelo $M_{[X]}/M/1/B$ foi extensivamente analisado [Garetto e Towsley 2003], mostrando-se capaz de prever os indicadores de desempenho de filas alimentadas por tráfego TCP com boa precisão.

3. Formulação do Problema de Dimensionamento de *Buffers*

O problema de atribuição de *buffer* (BA) é baseado no modelo de filas $M_{[X]}/M/1/B$ tendo por objetivo dimensionar os *buffers* dos enlaces da rede. A idéia central corresponde a formular o problema como sendo um problema de otimização onde procura-se dimensionar os *buffers* com o menor tamanho possível, objetivando fazer com que a probabilidade de perda de pacotes em cada rota não ultrapasse o valor PR_{sd} estabelecido pelo projetista. Dados: a topologia, a requisição de tráfego, os fluxos e as capacidades para os enlaces (encontradas *a priori*), o BA será formulado como o seguinte problema de otimização:

$$Z_{BA} = min \sum_{ij} h(B_{ij}) \tag{1}$$

sujeito à

$$\sum_{ij} \delta_{ij}^{sd} p(B_{ij}, C_{ij}, f_{ij}, [X]) \leq PR_{sd} \quad \forall (s,d), \forall (i,j) \tag{2}$$

$$B_{ij} \geq 0 \quad \forall (i,j) \tag{3}$$

A equação (1) é a função objetivo que minimiza o custo total dos *buffers* na rede, onde B_{ij} é o tamanho do *buffer* (em pacotes) no enlace (i,j) e $h(B_{ij})$ é a função custo. Neste trabalho considera-se uma função custo linear, ou seja, $h(B_{ij}) = b_{ij} \cdot B_{ij}$, onde b_{ij} é o custo monetário em \$/pacote/ano. Na equação (2) temos a restrição de perda de pacotes que é representa pela soma das probabilidades de perda em cada enlace do caminho (s,d). $p(B_{ij}, C_{ij}, f_{ij}, [X])$ é a probabilidade de perda de pacotes para a fila $M_{[X]}/M/1/B$. Observa-se que o lado esquerdo da restrição (2) é baseado na consideração de que as perdas nos enlaces são independentes. A equação (3) garante que nenhum *buffer* terá tamanho negativo. É importante lembrar que a performance do TCP é afetada pela perda dos segmentos de dados no envio e pela perda dos pacotes de resposta ACKs no caminho inverso, principalmente para fluxos pequenos. Neste trabalho considerou-se que a probabilidade de perda de ACKs é desprezível.

O problema formulado pode ser classificado com um problema de otimização convexa multi-variável vinculado; a condição convexa garante a existência de uma única solução, ou seja, um ótimo global.

4. Solução Existente

A abordagem utilizada em [Wille et al. 2004], para a solução do problema BA, surge da combinação de dois métodos: Método da Barreira Logarítmica (MBL) [Wright 1992] e Método das Coordenadas Cíclicas (MCC). O MBL opera minimizando uma função composta que reflete a função objetivo original, com a influência dos vínculos. A idéia principal é converter um problema vinculado para um problema não vinculado ou para uma seqüência de subproblemas não vinculados.

Seja o seguinte problema de programação matemática que envolve a minimização de uma função linear (ou não linear) sujeita à um conjunto de vínculos:

$$\text{Minimize} : g_o(x) \tag{4}$$

sujeito à:

$$g_i(x) \leq 0; \quad i = 1, \cdots, m \tag{5}$$

$$x \in R^n \tag{6}$$

onde $g_i(x)$ é uma função convexa e contínua em R.

Os pontos em que todas as funções de restrições (equação (5)) são negativas é denotado por $\Psi(G)$ e definido como:

$$\Psi(G) = \{x : g_i(x) < 0; i = 1, \cdots, m\} \tag{7}$$

Um ponto x em $\Psi(G)$ é dito (estritamente) admissível. A barreira logarítmica é definida como sendo a função:

$$\phi(x) = \begin{cases} -\sum_{i=1}^{m} \log(-g_i(x)); & g_i(x) < 0,\ i = 1, \cdots, m \\ +\infty; & \text{de outro modo} \end{cases} \tag{8}$$

Esta função tende ao infinito quando x se aproxima de $\Psi(G)$. O método da barreira consiste na formulação da função

$$\mathcal{B}(x, t) = g_o(x) + (1/t) \cdot \phi(x) \tag{9}$$

onde $x \in \mathbf{R}^n$, e t é um escalar positivo. Dado um ponto inicial (estritamente) admissível, é possível manter este ponto sendo (estritamente) admissível, minimizando a função $\mathcal{B}(x, t)$. O peso $(1/t)$ no termo pertencente a barreira $\phi(x)$, controla a distância da função em relação a barreira. Quando o peso aumenta, a equação (9) é afastada da barreira, quando diminui, é aproximada. Ao minimizar (9), $(1/t)$ deve ser reduzido com as iterações até próximo de zero, fazendo com que a equação se aproxime da barreira, sem nunca atingí-la, e convergindo para uma solução x^* no limite de $\Psi(G)$. Para resolver o problema e minimizar a função $\mathcal{B}(x, t)$, o MCC é utilizado. Este método consiste em partir de um ponto inicial x_1 e minimizar a função $\mathcal{B}(x, t)$ na direção $d_1 = (1, 0, \cdots, 0)$; encontrando o ponto x_2 que minimiza a função nesta direção; partindo deste ponto na direção $d_2 = (0, 1, \cdots, 0)$, minimizando a função encontrando o ponto x_3, e assim sucessivamente até chegar ao ponto x_{n+1}, onde o ciclo se repete e volta-se a minimizar na direção $d_1 = (1, 0, \cdots, 0)$. O vetor direção d_n, tem o componente n igual a 1 e todos os outros iguais a 0. O processo se repete até alcançar a precisão desejada.

Esta abordagem de solução foi implementada, em [Wille et al. 2004], usando linguagem de programação C. Entretanto o sistema obtido revelou-se de baixa confiabilidade dada a dificuldade de integração entre os dois métodos MBL e MCC. Além disso, dada à característica do método, que exige que as soluções não deixem a região de admissibilidade (i.e., tamanhos de buffer elevados) [Wright 1992], parte do sistema que trata da solução da CMTC pode apresentar instabilidade numérica.

5. Proposta de Solução

Devido aos problemas relatados na seção anterior propõe-se uma nova abordagem de solução. O modelo de filas $M_{[X]}/M/1/B$ não fornece uma fórmula fechada para o cálculo da probabilidade de perda de pacotes, o que dificulta o processo de otimização. A proposta, apresentada neste artigo, está baseada em dois passos: (i) na utilização de uma função analítica aproximada para cálculo da probabilidade de perda (que irá substituir $p(B_{ij}, C_{ij}, f_{ij}, [X])$ na formulação do problema), e (ii) no uso de uma heurística simples (batizada como Heurística da Decomposição) para a obtenção de uma solução para o problema BA.

5.1. Função de Probabilidade de Perda

A função aproximada está representada na equação (10), onde p_{ij} representa a probabilidade de perda de cada enlace (i, j), k é uma constante de ajuste da altura da curva e α ajusta sua inclinação, B é o tamanho do *buffer* e ρ é o fator de utilização do enlace.

$$PP_{ij} = k\frac{(1-\rho_{ij}^{\alpha_{ij}})\rho_{ij}^{\alpha_{ij}B_{ij}}}{1-\rho_{ij}^{\alpha_{ij}(B_{ij}+1)}} \tag{10}$$

Desta forma pode-se através de um processo de ajuste dos parâmetros α e k com os resultados (da solução numérica) da fila $M_{[X]}/M/1/B$ (para um ρ conhecido) calcular os valores da probabilidade de perda de pacotes. Para efetuar o ajuste de curvas, pode ser utilizada uma adaptação do conhecido Método da Seção Áurea (MSA).

Como exemplo, a Figura 2 mostra as curvas de probabilidade de perdas de pacotes produzidas pela fila $M_{[X]}/M/1/B$ e pela equação (10), referentes a um enlace com índice de utilização de 42%, onde observa-se uma boa concordância de valores. Está representado também o valor do parâmetro α utilizado para ajustar as curvas. O valor de κ foi ajustado em 7,1.

5.2. Heurística da Decomposição

A heurística proposta consiste em decompor o problema em $n\times(n-1)$ problemas simples (um para cada caminho (s, d)). Seja I_{sd} um caminho individual entre origem-destino, e seja B_{ij}^{sd} uma variável auxiliar que representa o *buffer* no enlace (i, j) no caminho (s, d). Aplicando o método dos multiplicadores de Lagrange obtem-se:

$$L(\nu) = min\left[\sum_{(i,j)\in I_{sd}} b_{ij}B_{ij}^{sd} + \nu\left(\sum_{(ij)\in I_{sd}} PP_{ij} - PR_{sd}\right)\right] \tag{11}$$

sujeito à

$$B_{ij}^{sd} \geq 0 \quad \forall(i,j), \forall(s,d) \tag{12}$$

As soluções são dadas por:

$$PE_{ij}^{sd} = \frac{b_{ij}\left(1-\rho_{ij}^{\alpha_{ij}(B_{ij}^{sd}+1)}\right)PR_{sd}}{\ln(\rho_{ij})\sum\limits_{(v,w)\in I_{sd}}\dfrac{b_{vw}\left(1-\rho_{vw}^{\alpha_{vw}(B_{vw}^{sd}+1)}\right)}{\ln(\rho_{vw})}} \tag{13}$$

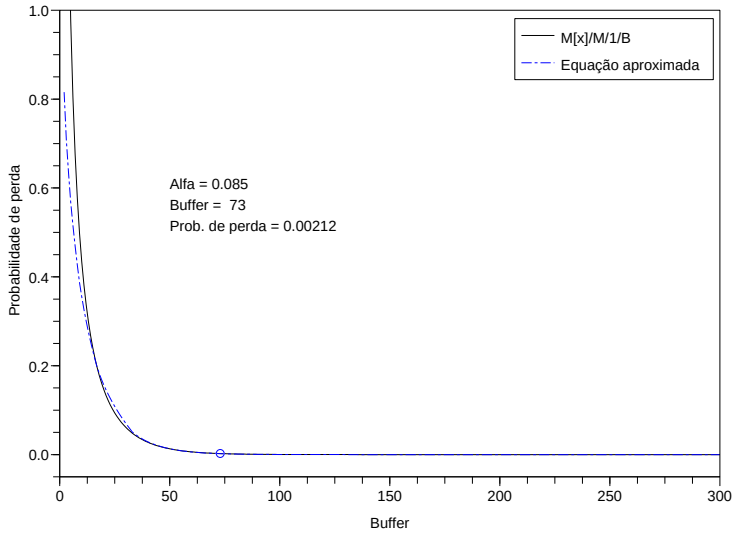


Figura 2. Comparação entre $M_{[X]}/M/1/B$ e equação (10).

Conhecendo os valores de PE_{ij}^{sd} (para cada caminho), obtém-se valores admissíveis para as variáveis PE_{ij}^{*} (no problema BA original) fazendo:

$$PE_{ij}^{*} = min_{sd}(PE_{ij}^{sd}) \quad \forall(s,d), \forall(i,j) \tag{14}$$

Esta última equação seleciona a menor probabilidade de cada enlace considerando sua participação em mais de um caminho. Entretanto, uma dificuldade extra deve ser vencida. Observando-se a equação (13) percebe-se a presença de três grupos de variáveis desconhecidas: α_{ij}, PE_{ij}^{sd} e B_{ij}^{sd}. Na seqüência serão apresentadas duas versões de um algoritmo para a determinação destas quantidades.

Na primeira versão do algoritmo (V1) os valores de PE_{ij}^{sd} e B_{ij}^{sd} são obtidos usando o Método da Iteração Linear; partindo de valores iniciais para PE_{ij}^{sd} e iterando até a convergência. Os valores de α_{ij} são determinados dinamicamente pelo ajustamento da curva dada pela equação (10) e a curva obtida de $M_{[X]}/M/1/B$ (solucionada via *software*) ao longo do processo iterativo. Posteriormente (14) é calculada. Na segunda versão (V2) o algoritmo é dividido em duas fases. Na fase 1 os valores de α_{ij} são determinados pelo ajustamento de curvas, permanecendo então fixos. Na fase 2 os valores de PE_{ij}^{sd} e B_{ij}^{sd} são obtidos pelo processo iterativo, partindo de valores de B_{ij}^{sd} admissíveis, e iterando até a convergência. Posteriormente (14) é calculada.

O algoritmo na versão (V1) pode ser representado pelo pseudo-código a seguir:

```
PROCEDIMENTO BA_V1()
(1)  INÍCIO
(2)      Ler dados de entrada: f_ij, C_ij, ρ_ij, PR_sd;
(3)      REPETIR para cada rota (s, d);
(4)          Atribuir para cada enlace (i, j) ∈ I_sd: PE_ij^0 ← PR_sd/NER, p ← 0;
(5)          ENQUANTO não há convergência FAÇA;
(6)              REPETIR para cada enlace (i, j) ∈ I_sd;
(7)                  Calcular B_ij^p ← f_N^-1(PE_ij^p) ;
(8)                  Calcular α_ij ← f_A^-1(B_ij^p, PE_ij^p);
(9)              FIM REPETIR
(10)             REPETIR para cada enlace (i, j) ∈ I_sd;
(11)                 Calcular PP_ij^p+1, com a eq. 13;
(12)             FIM REPETIR
(13)             Atribuir p ← p + 1;
(14)         FIM ENQUANTO
(15)     FIM REPETIR;
(16)     REPETIR para cada enlace (i, j);
(17)         Calcular PE*_ij, com a eq. 14;
(18)         Calcular B*_ij ← f_N^-1(PE*_ij) ;
(19)     FIM REPETIR
(20) FIM;
```

O algoritmo na versão (V2) pode ser representado pelo pseudo-código a seguir:

```
PROCEDIMENTO BA_V2()
(1)  INÍCIO
(2)      Ler dados de entrada: f_ij, C_ij, ρ_ij, PR_sd;
(3)      Executar ajuste de α_ij para cada enlace (i, j);
(4)      REPETIR para cada rota (s, d);
(5)          Atribuir para cada enlace (i, j) ∈ I_sd: B_ij^0 ← admissível, p← 0;
(6)          ENQUANTO não há convergência FAÇA;
(7)              REPETIR para cada enlace (i, j) ∈ I_sd;
(8)                  Calcular PE_ij^p+1, com a eq. 13;
(9)                  Calcular B_ij^p+1 ← f_N^-1(PE_ij^p+1) ;
(10)             FIM REPETIR
(11)             Atribuir p ← p + 1;
(12)         FIM ENQUANTO
(13)     FIM REPETIR;
(14)     REPETIR para cada enlace (i, j);
(15)         Calcular PE*_ij, com a eq. 14;
(16)         Calcular B*_ij ← f_N^-1(PE*_ij) ;
(17)     FIM REPETIR
(18) FIM;
```

Nota-se que no algoritmo proposto o Método da Seção Áurea (MSA) foi utilizado para o cálculo das inversas das funções f_N e f_A; sendo que f_N calcula a probabilidade de perda do enlace em função do tamanho do *buffer*, fluxo e capacidade (solução numérica da fila $M_x/M/1/B$), e f_A calcula a probabilidade de perda em função do tamanho *buffer*, fluxo, capacidade, fator de utilização e alfa (pela formula aproximada). A quantidade NER é o número de enlaces utilizados em cada rota (s, d), é utilizado para estabelecer valores iniciais para as probabilidades de perda.

5.3. Complexidade

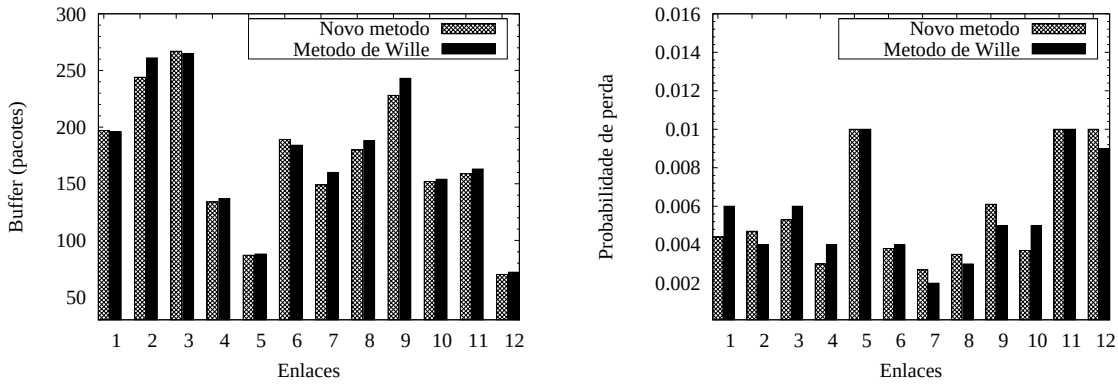
Sendo N_{itr} o número de iterações, N_m o número de restrições e L o número de enlaces, a complexidade do algoritmo de solução para o problema BA é dada por $O(N_m.N_{itr}.L.(\psi_\alpha + \psi_b))$ para a versão V1, e por $O(L.\psi_\alpha + N_m.N_{itr}.L.\psi_b)$ para a versão V2. O processo de Iteração Linear é realizado para cada uma das N_m restrições. Sua convergência requer N_{itr} iterações. A cada iteração, na versão V1, a operação de determinação do valor do buffer na curva $M_{[X]}/M/1/B$, com complexidade ψ_b, e a determinação dos valores do parâmetro α, com complexidade ψ_α, são realizadas para L enlaces (no máximo). Na versão V2 a detreminação de α é realizada separadamente.

6. Simulações e Validação dos Resultados

Para testar os métodos propostos para o dimensionamento de *buffers*, três topologias de redes com tamanhos diferentes foram consideradas. Todas as três topologias foram geradas aleatoriamente. Para as três topologias, a matriz de tráfego foi calculada pela geração de valores aleatórios uniformemente distribuídos para cada par origem-destino. A primeira topologia, composta por 5 nós, 12 enlaces e 11 restrições de atrasos, foi gerada de forma a abranger uma área de 215km x 215km (topologia #1). A segunda topologia é formada por 11 nós, 24 enlaces, 15 restrições de atrasos e abrange uma área de 20km x 20km (topologia #2). Por fim, a terceira topologia é formada por 20 nós, 48 enlaces, 25 restrições e abrange uma área de 20km x 20km (topologia #3).

Usando o procedimento proposto em [Wille et al. 2004, Wille et al. 2005] foram calculadas as capacidades dos enlaces presentes nas diversas topologias. Considerou-se que as mesmas restrições de QoS são impostas para todos os pares origem-destino. São elas: (i) latência inferior a 0.5 s para fluxos TCP com menos de 20 pacotes, e (ii) vazão maior que 512 kbps para fluxos TCP com mais de 20 pacotes. Solucionou-se, então, o problema BA considerando $PR_{sd} = 0.01$.

Para efeito de comparação, os gráficos da Figura3 trazem os resultados do BA obtidos por [Wille et al. 2004] sobre a topologia #1, confrontados com os resultados da versão 1 do algoritmo proposto neste trabalho. Por simplicidade utilizou-se $b_{ij} = 1.0$. Embora os resultados sejam muito próximos, a contribuição está na simplicidade do novo método.



a) Tamanho do *buffer*.

b) Probabilidade de perda.

Figura 3. Comparação entre os resultados de Wille e o novo método.

Na Tabela 1 são mostrados todos os valores calculados referentes a topologia #1.

6.1. Simulações

Para validar os dimensionamentos diversas simulações foram realizadas utilizando o programa *ns-2* [McCanne e Floyd]. Utilizou-se a abordagem tipo *Batch Means* [Taha 2002], com intervalo de confiança de 95%. O teste consiste em simular o comportamento de fluxos TCP que percorrem um determinado caminho na rede (e coletar indicadores de desempenho) entre uma dada origem e um dado destino. As aberturas das conexões TCP seguem processo de Poisson. As taxas de abertura de conexões são determinadas pelo índice de utilização do enlace ρ_{ij}. A quantidade de dados a ser transmitida em cada

Tabela 1. Cálculos para topologia #1, tráfego #1.a.

Enlace	Fluxo	Capacidade	ρ	α	*Buffer*	p
1	16	20	0.8	0.095	140	0.00787
2	11	20	0.55	0.09	76	0.00589
3	21	50	0.42	0.085	72	0.00233
4	21	50	0.42	0.085	73	0.00212
5	2	4	0.5	0.09	61	0.01
6	14	20	0.7	0.095	109	0.00556
7	24	50	0.48	0.09	68	0.00521
8	19	50	0.38	0.085	68	0.00209
9	9	20	0.45	0.085	66	0.00478
10	3	6	0.5	0.09	61	0.01
11	8	10	0.8	0.095	130	0.01
12	21	50	0.42	0.085	65	0.00410

conexão, é expressa em número de pacotes. Um tráfego misto TCP é considerado, onde os tamanhos dos arquivos são obtidos de medidas apresentadas em [Mellia et al. 2002].

Dois casos são considerados. A primeira simulação será sobre um caminho origem-destino com 3 enlaces, pertencente a rede formada por 24 enlaces (topologia #2). A segunda simulação será sobre um caminho origem-destino com 4 enlaces, pertencente a rede formada por 48 enlaces (topologia #3).

As Tabelas 2 e 3 mostram os resultados obtidos. Tanto os fluxos, como as capacidades são dados em [Mpbs]. O tamanho do *buffer* é expresso em pacotes. Os atrasos médios $E[T]$, o número médio de pacotes $E[N]$ e as probabilidades de perdas p_{ij}, obtidos pelas simulações e calculadas com o modelo de filas $M_{[X]}/M/1/B$, também são mostradas. Pode-se observar uma boa concordância entre os valores previstos e aqueles obtidos via simulação em todos os casos. Observa-se que a margem de erro das simulações é de $\pm 15\%$ para $E[T]$ e $E[N]$; e de $\pm 20\%$ para p_{ij}.

Tabela 2. Dados de 3 enlaces da topologia #2.

		$Projeto$			$M_{[X]}/M/1/B$			$NS-2$		
Enlace	Fluxo	ρ	C	B	$E[T]$	$E[N]$	p_{ij}	$E[T]$	$E[N]$	p_{ij}
1	5	0.53	9.37	92	0.020	9.21	0.0017	0.030	12.32	0.0028
2	12	0.78	15.31	149	0.028	29.65	0.0047	0.032	31.60	0.0044
3	10	0.80	14.64	140	0.020	17.98	0.0015	0.027	22.52	0.0020

Tabela 3. Dados de 4 enlaces da topologia #3

		$Projeto$			$M_{[X]}/M/1/B$			$NS-2$		
Enlace	Fluxo	ρ	C	B	$E[T]$	$E[N]$	p_{ij}	$E[T]$	$E[N]$	p_{ij}
1	14	0.76	18.34	167	0.022	27.39	0.0020	0.025	28.65	0.0023
2	14	0.72	19.19	152	0.018	22.67	0.0017	0.019	22.40	0.0020
3	7	0.54	12.78	106	0.015	9.87	0.0010	0.018	10.26	0.0020
4	54	0.89	60.59	275	0.015	73.89	0.0049	0.016	72.03	0.0026

Os gráficos 4 e 5 mostram a latência na transmissão de arquivos de vários tamanhos calculado usando o modelo analítico do protocolo TCP reportado em [Cardwell et al. 2000] (modelo CSA). A restrição de QoS de latência permitida $L_t = 0.5s$, também é mostrada nos gráficos. As observações devem restringir-se à latência na

transferência de arquivos com até 20 pacotes, pois o modelo CSA não é muito preciso quanto à latência na transferência de arquivos grandes [Cardwell et al. 2000]. Observa-se uma boa concordância com os valores obtidos pela simulação do sistema no software *ns-2*. (A margem de erro das simulações varia entre $\pm 15\%$ e $\pm 34\%$ de acordo com o tamanho do fluxo TCP.).

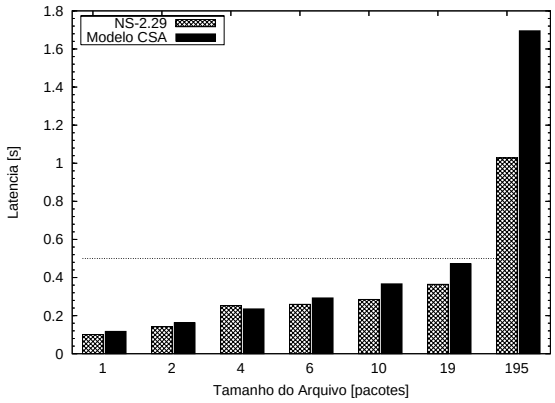


Figura 4. Latência para 3 enlaces, topologia #2.

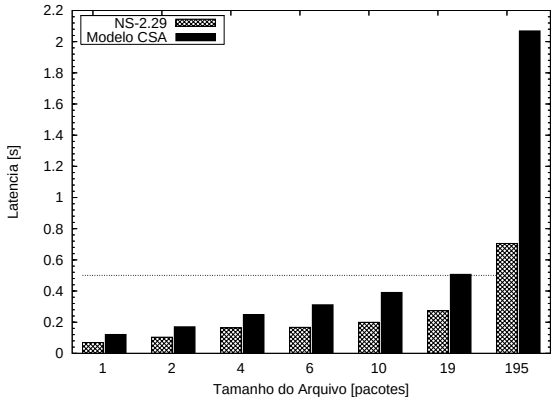


Figura 5. Latência para 4 enlaces, topologia #3, CR

7. Discussão e Conclusões

Uma forma de comparar as abordagens corresponde a encontrar o número equivalente de conexões TCP que produzem o fluxo desejado sobre um roteador (usando a conhecida formula para o throughput de uma conexão TCP [M. Mathis et al. 1997]) e combiná-la com a regra de algibeira. Esta combinação leva à seguinte fórmula para cálculo do tamanho do buffer:

$$B^* = \sqrt{\frac{0,93.C.RTT}{\rho.MSS.\sqrt{p}}} \tag{15}$$

onde MSS é o *maximum segment size* em bits.

Tomando o exemplo da topologia #2 e recalculando os valores de buffer com esta fórmula obtemos a Tabela 4, onde observa-se que os valores de buffer B^* diminuíram o

que levou a um grande aumento das probabilidades de perda (indicando que, na verdade, tais valores de buffer contribuem para uma redução do desempenho do TCP).

Tabela 4. Dados de 3 enlaces da topologia #2.

		$Projeto$			$M_{[X]}/M/1/B$		
Enlace	Fluxo	ρ	C	B^*	$E[T]$	$E[N]$	p_{ij}
1	5	0.53	9.37	49	0.018	7.73	0.0244
2	12	0.78	15.31	40	0.013	12.41	0.0997
3	10	0.80	14.64	54	0.015	12.96	0.0417

Com relação a abordagem proposta, uma última consideração pode ser feita. Se na equação (13) considerarmos que B_{ij}^{sd} é elevado então $\rho_{ij}^{\alpha_{ij}(B_{ij}^{sd}+1)} << 1$. Deste modo pode-se obter a equação simplificada (16) para a distribuição de probabilidade de perdas entre os enlaces (e de conseqüência o dimensionamento de buffers). Esta equação apresenta uma margem de erro em relação aos valores ótimos de $\pm 20\%$ (o que corresponde a uma margem aceitável em termos de engenharia).

$$PS_{ij}^{sd} = \frac{b_{ij}.PR_{sd}}{\ln(\rho_{ij}) \sum\limits_{(v,w)\in I_{sd}} \frac{b_{vw}}{\ln(\rho_{vw})}} \tag{16}$$

Para concluir, poucos artigos sobre o assunto foram encontrados e existem muitas dúvidas ainda sobre como encontrar o melhor caminho para projetar redes complexas levando-se em conta os parâmetros de QoS, principalmente no que diz respeito às ferramentas analíticas. A comprovação dos resultados teóricos com auxílio de simulação, por software e por simuladores de redes (*ns-2*) mostrou-se eficaz. Em relação aos resultados obtidos analiticamente e, paralelamente, com os algoritmos computacionais propostos, conclui-se que o método de aproximação $M_{[X]}/M/1/B$ resulta numa ferramenta simples e eficaz no dimensionamento de *buffers*.

Referências

C. Villamizar e C. Song. “High performance TCP in ANSNET,” *SIGCOMM Comput. Commun. Rev.*, Vol. 24, pp. 45–60, 1994.

R. Bush e D. Meyer. “RFC 3439: Some internet architectural guidelines and philosophy,” dezembro, 2003.

G. Appenzeller, I. Keslassy and N. McKeown. “Sizing Router Buffers ,” *Proceeding of ACM SIGCOMM ’04*, Portland - Oregon, setembro, 2004.

M. Mathis, J. Semke e J. Mahdavi. “The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm,” *Computer Communications Review*, Vol. 27, julho, 1997.

Kleinrock, L. “Queueing Systems, Volume II: Computer Applications,” *Wiley Interscience*, Nova York, 1976.

E. Wille, M. Garetto, M. Mellia, E. Leonardi, M. Ajmone Marsan, “Considering End-to-End QoS in IP Network Design,” *NETWORKS 2004*, Viena, Austria, pp. 13–16, junho, 2004.

E. Wille, M. Garetto, M. Mellia, E. Leonardi, M. Ajmone Marsan, “IP Network Design with End-to-End QoS Constraints: The VPN Case,” *19th International Teletraffic Congress*, Beijing, China, agosto/setembro, 2005.

N. Cardwell, S. Savage, T. Anderson, “Modeling TCP Latency,” *IEEE Infocom 00*, Tel Aviv, Israel, março, 2000.

M. Mellia, A. Carpani, R. Lo Cigno, “Measuring IP and TCP behavior on Edge Nodes,” *Proceedings of IEEE Globecom 2002*, Taipei, Taiwan, novembro, 2002.

X. Chao, M. Miyazawa, M. Pinedo, *Queueing Networks, Customers, Signals and Product Form Solutions*, John Wiley, 1999.

H. A. Taha, *Operations Research: An Introduction*, Prentice Hall, 7a Edição, 2002.

M. Garetto, D. Towsley, “Modeling, Simulation and Measurements of Queuing Delay under Long-tail Internet Traffic,” *ACM SIGMETRICS 2003*, San Diego, CA, junho, 2003.

M. Wright, ”Interior methods for constrained optimization,”*Acta Numerica*, Vol. 1, pp. 341-407, 1992.

S. McCanne, S. Floyd, “NS Network Simulator”, Disponível em http://www.isi.edu/nsnam/ns/.