

# Uma metodologia de dimensionamento com QoS usando cadeias de Markov ocultas

Edmundo de Souza e Silva<sup>1</sup>, Rosa M. M. Leão<sup>1</sup>, Marcos B. Trindade<sup>2</sup>, Ana Paula C. da Silva<sup>1</sup>, Bruno F. Ribeiro<sup>1</sup>, Flavio P. Duarte<sup>1</sup> e Jorge A. Azevedo<sup>1</sup>

**Resumo**—O custo de uma rede de comunicações IP pode ser entendido como o custo financeiro de contratação da banda requerida ou, no caso de uma rede própria, o custo financeiro de manutenção dos cabos, fibras e equipamentos para determinadas capacidades dos enlaces. O objetivo deste trabalho é dimensionar uma rede IP garantindo o menor custo possível mas ao mesmo tempo garantido a qualidade de serviço (QoS) contratada pelos clientes desta rede.

**Abstract**—The cost of an IP network can be seen as the financial cost to acquire the necessary bandwidth or, in the case of a self-owned network, the financial cost to provide maintenance in all equipments given a certain link capacity. The goal of this work is to dimension an IP network assuring the minimum possible cost but, at the same time, providing the agreed quality of service (QoS) to its customers.

## I. INTRODUÇÃO

O objetivo do dimensionamento de uma rede IP é minimizar o “custo da rede” ao mesmo tempo garantindo a qualidade de serviço contratada pelos clientes. Esta qualidade de serviço mínima precisa ser definida pelo projetista da rede. Neste trabalho estaremos dimensionando uma rede a partir do tráfego medido, seja da própria rede a ser projetada ou de outra rede que possua **características** de tráfego similares.

A definição de “custo da rede” apresentada é baseada na capacidade dos enlaces sendo esta definição a mais ampla possível. Podemos entender por custo da rede o custo financeiro de contratação da banda requerida ou, no caso de uma rede própria, o custo financeiro de manutenção dos cabos, fibras e equipamentos para determinadas capacidades dos enlaces. Outras definições também são cabíveis, ficando a cargo do projetista definir qual tipo de **custo** se deseja minimizar. Nos exemplos usados durante este trabalho, usaremos a *função de custo* baseada no custo financeiro de contratação da banda requerida.

Uma vez definida a função que desejamos minimizar, devemos especificar quais são as restrições. Claramente existe um limite para minimizar os custos de uma rede. E este limite é a qualidade de serviço. Mais diretamente, quanto menor a qualidade de serviço, menor é o custo da rede. Esta relação de proporcionalidade será explorada mais adiante e o problema de otimização será solucionado na seção V.

Na seção IV, mapearemos os requisitos de QoS das aplicações para requisitos no nível da rede, de modo que possamos definir as restrições da nossa minimização. Existem dois parâmetros principais para avaliar a qualidade a nível de rede: o atraso fim-a-fim (ou RTT) e a probabilidade de perda.

Para que possamos ponderar os requisitos de qualidade (RTT e probabilidade de perda) na função de custo (que é uma função da capacidade dos enlaces), precisamos definir a relação entre ambos. Sendo assim, devemos especificar um modelo de tráfego que relacione o RTT e a probabilidade de perda com as capacidades dos enlaces. Este modelo precisa ser o mais fiel possível ao tráfego real quando da operação da rede. Isto é necessário para que o projeto represente o mais fielmente possível a rede em operação.

A utilização direta dos dados coletados da rede não é adequada ao dimensionamento pois lhes falta variabilidade estatística. Dessa forma, precisamos de um modelo de tráfego que seja fiel às principais características do tráfego e que não seja complexo nem exageradamente detalhado. Estas duas últimas características tornariam o projeto da rede uma tarefa árdua e pouco automática. Assim, foi escolhida uma modelagem simples e automática usando modelos markovianos ocultos, que têm recebido uma ampla atenção na literatura, conseguindo capturar características importantes do tráfego. A escolha do modelo é uma das etapas mais importantes do processo. A seção III é dedicada à descrição do modelo adotado.

Um modelo de tráfego fiel tem que ser baseado em informações confiáveis do tráfego a ser modelado. Esta é talvez a fase mais crítica de todo o projeto da rede, onde informações erradas podem levar a resultados imprecisos nas outras etapas. Como obter estas informações é objeto de estudo da próxima seção.

Finalmente, na seção VI são apresentados os resultados do método proposto. A seção VII apresenta os trabalhos relacionados e as conclusões e trabalhos futuros são mostradas na seção VIII.

## II. OBTENÇÃO DO TRÁFEGO AGREGADO NOS ENLACES

O objetivo desta seção é descrever como obter o tráfego agregado nos enlaces que será utilizado para parametrizar o modelo analítico adotado neste trabalho. A parametrização deste modelo requer como dado de entrada a quantidade de bits por unidade de tempo que passou no enlace. A escolha da granularidade da coleta e do intervalo de observação foi baseada em trabalhos da literatura [2], [12]. A granularidade usada foi de 1s com um intervalo de observação de 2h.

<sup>1</sup> Universidade Federal do Rio de Janeiro COPPE/Prog. de Eng. de Sistemas e Computação e Depto. de Ciência da Computação do IM. E-mails: {rosam, edmundo, anapaula, bruno, flaviop, allyson}@land.ufrj.br.

<sup>2</sup> Fundação CPqD Centro de Pesquisa e Desenvolvimento em Telecomunicações. Rodovia Campinas - Mogi-Mirim, km 118,5 CEP 13086-902 - Campinas, SP - Brasil E-mail: trindade@cpqd.com.br.

Este trabalho foi parcialmente financiado pelo CNPq e CPqD.

O método mais direto e confiável para obter o tráfego agregado nos enlaces é a medição direta do mesmo. Entretanto, nem sempre há a possibilidade de se medir o tráfego real em todos os enlaces com uma granularidade fina (100ms ou 1s). Neste caso, podemos medir a média do tráfego com granularidade mais grossa, por exemplo de 30 em 30 minutos, e utilizar as características do tráfego obtidas pela metodologia descrita em [2] para estimar o comportamento do tráfego de granularidade fina.

Um ponto a ser ressaltado, é que não é necessário medir o tráfego em todos os enlaces da rede. Diversas metodologias de estimativa do tráfego de toda a rede a partir do tráfego nos roteadores de borda estão disponíveis na literatura (vide [11] para um resumo dos métodos disponíveis).

#### A. Obtendo traces de granularidade fina a partir de estimativas do tráfego

O trabalho [2] mostra como obter a média e a variância do tráfego com granularidade fina (1 segundo) a partir da média, da variância com granularidade grossa (30 minutos) e de um parâmetro,  $b$ , cujo o valor depende do comportamento do tráfego da rede.

O nosso objetivo é obter um *trace* com granularidade fina que servirá como estimativa do tráfego entre dois roteadores  $A$  e  $B$  de uma rede. Iremos supor que é possível estimar os dados necessários para parametrizar o modelo de [2], ou seja, é possível coletar a média e a variância com granularidade de 30 minutos (estes dados podem ser coletados usando o NetFlow, por exemplo). Com relação ao valor do parâmetro  $b$ , dado que não é indicado estimá-lo a partir de dados coletados com granularidade grossa, devemos usar valores de  $b$  calculados para outros fluxos da rede que tenham características semelhantes as do tráfego entre  $A$  e  $B$ , como por exemplo em um enlace de menor capacidade da mesma rede que possa ser medido com granularidade mais fina.

De posse dos parâmetros do modelo de [2], é possível gerar um *trace* com granularidade fina para representar este tráfego. Supondo que este tráfego seja proveniente de diversas fontes independentes, este pode ser aproximado por uma variável aleatória Normal com média e variância obtidas a partir da metodologia descrita acima. Logo, o procedimento para geração do *trace* pode ser resumido como a geração, com granularidade fina, de uma variável aleatória Normal com média e variância que mudam a intervalos de 30 minutos.

A seguir apresentaremos um exemplo com o objetivo de ilustrar o método. Usaremos o tráfego total em Kb/s coletado do roteador do PESC/COPPE com a granularidade de 1 segundo. Denominaremos este tráfego como tráfego medido. Obtivemos a média e a variância para cada intervalo de 30 minutos segundo o modelo de [2] para este mesmo tráfego. Gerou-se um *trace* com granularidade de 1 segundo (denominaremos como tráfego estimado sintético ou somente tráfego estimado) com o objetivo de comparar com o tráfego medido. Na Figura 1 podemos notar a semelhança do tráfego estimado com o tráfego medido (um caminho amostral) para amostras com intervalos de 1 segundo.

Uma vez obtido o tráfego agregado nos enlaces, seja por medições ou baseado em uma estimativa, resta criar um mo-

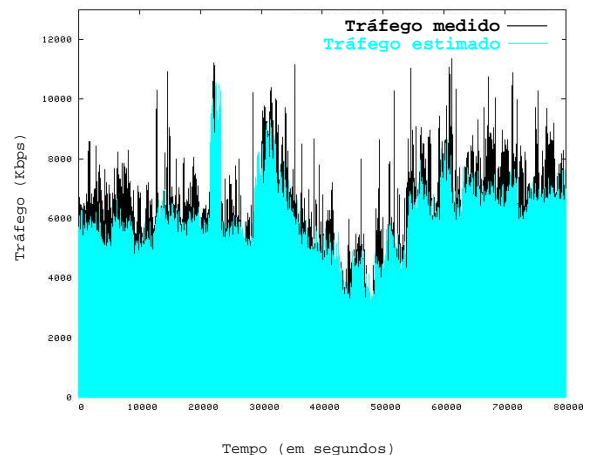


Fig. 1. Tráfego estimado e medido para o roteador do PESC/COPPE

delo que represente as principais características deste tráfego. Este é o assunto da próxima sub-seção.

#### B. Modelo de Tráfego Agregado dos Enlaces

Para cada enlace deve ser obtido um modelo de tráfego agregado. O objetivo é capturar as características mais importantes do tráfego agregado em diversos dias. Para não ficarmos restritos a apenas um caminho amostral, é necessário um modelo analítico preciso para o tráfego.

O modelo analítico estudado neste trabalho é baseado na teoria de cadeias de Markov ocultas (*Hidden Markov Models* - HMM). Estes modelos têm sido utilizados com sucesso para descrever séries temporais, em aplicações que incluem reconhecimento de voz e desenvolvimento de modelos precisos do processo de descarte de pacotes na rede [9], [7]. Este modelo tem como característica capturar correlações temporais que porventura existam no tráfego agregado.

Um modelo HMM que represente o tráfego agregado em um enlace irá alimentar um roteador cujo tempo de serviço é exponencial. Através da resolução analítica deste modelo iremos obter uma função,  $f_i(d_i)$  (a ser usada na equação 2), que mapeia o atraso da fila do roteador  $d_i$  na capacidade do canal  $f_i(d_i)$ . Isso é possível porque estamos trabalhando com a premissa de que a probabilidade de perda de pacotes é pequena. Os detalhes da modelagem de tráfego usando HMM estão descritos na próxima seção.

### III. CADEIAS DE MARKOV OCULTAS

O objetivo desta seção é adicionar variabilidade estatística aos tráfegos medidos (*traces* coletados). A modelagem do tráfego medido deve ser feita de forma automática, de modo que possamos dimensionar grandes redes.

Em [10] os modelos de Markov ocultos ou *Hidden Markov Models* (HMMs) são definidos como um processo estocástico duplamente embutido onde existe um processo estocástico implícito que não é observável. No entanto, este somente pode ser “visto” através de um outro conjunto de processos estocásticos que produzem a seqüência de observações. Nesta seção será apresentada a metodologia utilizada para modelar

os *traces* com HMMs. A sub-seção III-A explica a utilização de um modelo HMM para representar o tráfego de um enlace. Este modelo será posteriormente utilizado para o modelo de dimensionamento.

#### A. Aplicação de HMM para criação de um modelo de tráfego

Iremos usar o modelo de Markov oculto (HMM) para representar o tráfego agregado que passa pelo roteador. Para que a cadeia represente esse tráfego, a mesma deve ser treinada com um tráfego real. Idealmente a cadeia deve ser treinada com o tráfego de vários dias no mesmo horário. Após o treinamento, a cadeia poderá ser utilizada para gerar um tráfego com uma duração maior que o tráfego real usado para treiná-la.

Em um HMM, a cada estado está associada uma probabilidade de emissão para cada símbolo existente. Como o campo a ser modelado (quantidade de bytes observados no enlace) pode assumir um número de valores muito grande, foi necessário utilizar o conceito de níveis, diminuindo, assim, o número de símbolos que cada estado poderia emitir. Nessa abordagem, primeiramente foi verificado a maior quantidade de bytes que atravessaram o enlace em um intervalo de tempo. Esse valor fornece um teto para o cálculo dos níveis. O mapeamento das amostras originais e das amostras discretizadas pelos níveis é obtido pela seguinte fórmula  $n_i = (O_i * K) / \text{MAX}(O_i)$ , onde  $n_i$  é a amostra discretizada em número de níveis,  $O_i$  é a amostra original coletada,  $K$  é a quantidade total de níveis e  $\text{MAX}(O_i)$  é o valor da maior amostra coletada.

A estrutura da cadeia inicial pode ter influência na cadeia obtida após o treinamento. A cadeia escolhida para modelar os *traces* possui uma estrutura do tipo “*Branch-Erlangian*” (semelhante a *phase-type*) com 4 estados (também foi testado um modelo com 6 estados, mas não houve melhora significativa no desempenho do modelo). Essa cadeia foi utilizada em [8] para a previsão de perdas de pacotes de áudio com bons resultados e pode ser vista na Figura 2.

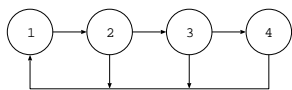


Fig. 2. Cadeia utilizada

Experimentos variando o número de níveis foram realizados com o objetivo de se estudar a influência desse parâmetro no modelo final. Resultados experimentais mostraram que a utilização de 30 níveis apresentou a melhor relação custo benefício para os *traces* do tráfego medido no roteador.

Para validar o modelo, foi adicionada uma fila com o tamanho máximo de 1000 pacotes ao modelo de Markov oculto. Para esse novo modelo, foi calculado o número médio de pacotes na fila, utilizando várias taxas de serviço. Esses resultados foram comparados com simulações equivalentes utilizando o *trace* do tráfego medido. Nessas simulações foi utilizada a ferramenta Tangram-II [4].

A Figura 3 mostra o número médio de pacotes na fila de transmissão de um enlace em função da capacidade do enlace em pacotes por segundo. Nesse gráfico é possível observar que

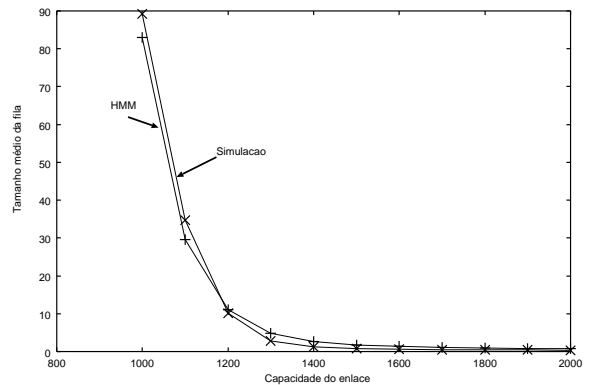


Fig. 3. HMM versus Simulação

os resultados obtidos pelo modelo analítico estão próximos da simulação feita com o *trace* do tráfego medido, indicando que o modelo analítico representa bem este tráfego.

#### B. Função de mapeamento Atraso versus Capacidade

O resultado obtido com o modelo HMM alimentando uma fila é o número médio de pacotes na fila para diversos valores de capacidade do enlace, onde podemos obter facilmente o atraso na fila em função dessas capacidades. O próximo passo é obter uma função que mapeie o tempo de espera na fila em função da capacidade do enlace. Para isso foi feita uma regressão não-linear com objetivo de definir as funções que tenham o menor erro possível, de fácil inversão, definidas em todo o domínio e com derivada primeira negativa. Exemplos de funções que se encaixam nessas definições são a função exponencial e a função potência.

A função que melhor se adequou aos requisitos descritos acima para a maioria dos *traces* foi  $g_l(C_l) = a(C_l - b)^c + d$ , onde  $g_l(C_l)$  é o tempo de espera na fila e  $C_l$  é a capacidade do enlace  $l$ .

O problema de otimização formulado em V exige que a função custo seja expressa em função do atraso em cada enlace pois as restrições são expressas em função desta variável. Iremos então inverter a função  $g_l$  e usaremos a sua inversa no problema de otimização. Definiremos então a função  $f_l(d_l) \equiv g_l^{-1}(d_l)$ . Descobrir as capacidades ideais para uma rede será visto então como um problema de encontrar os valores mínimos de uma função de custo baseada em  $f_l(d_l)$  (função de custo das capacidades) de modo que as restrições de QoS das aplicações sejam atendidas (latência e vazão).

Na próxima seção veremos como traduzir os requisitos de QoS (latência e vazão) das aplicações, em restrições para o problema de otimização.

## IV. MAPEAMENTO DOS PARÂMETROS DE QoS

Ao contratar os serviços de uma empresa para dimensionar suas redes, os consumidores fazem exigências referentes à QoS das aplicações que lhes parecem mais importantes, podendo especificar, por exemplo, o tempo máximo que lhes parece aceitável para trazer uma página *Web* (latência), ou ainda a vazão mínima que deve ser oferecida para a transferência de arquivos maiores, como músicas e vídeos. Assim, é necessário

que exista uma forma de associar tais exigências a parâmetros configuráveis da rede, como capacidade do enlace, o atraso médio e a capacidade de armazenamento das filas do roteador. Estas exigências a nível de rede serão as nossas restrições para a diminuição do custo da rede. O custo somente pode diminuir se a qualidade mínima requerida continuar sendo atendida.

Suponha, por exemplo, que o usuário tenha requisitado que o *download* de arquivos grandes seja feito com uma vazão média  $B$ . A partir de [13] podemos obter uma expressão que associa a média do RTT com a média da vazão:

$$RTT(p) = \text{MIN} \left( \frac{Wmax}{B}, \frac{1}{B \left( \sqrt{\frac{2bp}{3}} + k \text{MIN}(1, 3\sqrt{\frac{3bp}{8}}) p (1 + 32p^2) \right)} \right) \quad (1)$$

onde  $p$  representa a probabilidade de perda,  $b$  representa o número de pacotes confirmados por um ACK,  $B$  representa a vazão,  $T_o = k RTT(p)$  é o tempo de *time out* do TCP e  $Wmax$  o tamanho máximo da janela do TCP. Portanto, a partir da vazão e da probabilidade de perda do enlace é possível determinar o RTT médio.

A latência, por sua vez, é o requisito mais importante para transferência de arquivos pequenos. Em geral, devido ao tamanho do arquivo, o TCP termina de enviá-lo antes de sair da fase de *slow-start*. Estaremos considerando o tempo de transmissão dos pacotes como desprezível em relação ao RTT. Como exemplo, suponha que o arquivo a ser transferido tenha  $1.5 \times 10^4$  bytes e que cada segmento do TCP tenha 1500 bytes. Transmitiremos, portanto, 10 segmentos. Considere a janela inicial do TCP igual a dois segmentos e o RTT médio de 250 ms. Não havendo perda, este arquivo pode ser transferido em aproximadamente 1 segundo.

A hipótese que adotamos é de que não ocorrerão perdas de pacotes na transferência de arquivos pequenos. Esta hipótese é baseada no cenário descrito a seguir. Digamos que a probabilidade de perda de um pacote seja  $10^{-4}$  e as que perdas sejam independentes. Supondo que em média 20 pacotes sejam utilizados para a transmissão do arquivo (contando com o SYN, ACKs, etc), a probabilidade de que o arquivo inteiro seja transferido sem que haja nenhuma perda é de 0.998. Assim, apenas um em cada 500 arquivos sofrerá um atraso extra por causa da perda de um pacote. A contribuição da perda é, portanto, pouco significativa para a média da latência.

Nos exemplos a seguir o tamanho de  $1.5 \times 10^4$  bytes foi utilizado para os arquivos pequenos de acordo com os resultados de [5], [6], onde se vê que aproximadamente 80% dos arquivos *Web* acessados pelos usuários são menores ou iguais a  $1.5 \times 10^4$  bytes.

Para se transferir um arquivo de 15000 bytes, considerando o tamanho da janela inicial do TCP com 2 segmentos, o tamanho dos pacotes de 1500 bytes e desconsiderando a perda de pacotes, são necessários 4 RTTs<sup>1</sup>. Este será considerado o valor máximo permitido de latência para arquivos pequenos encontrados na *Web*. Se a latência deve ser menor ou igual

<sup>1</sup>Se conexões HTTP persistentes forem utilizadas (HTTP 1.1), o tempo médio de latência dos objetos tende a diminuir pois o *3-way hand shaking* do TCP só é executado uma vez para cada página acessada.

a 1.5 segundos, então o RTT deve ser menor ou igual a 375 ms (1.5/4 segundo), sempre lembrando que a probabilidade de perda deve se manter baixa. Para arquivos de tamanho grande o requisito observado será a vazão média  $B$ .

#### A. Tradução dos requisitos de QoS da aplicação para o nível de rede

Esta tradução será feita usando: a rota dos fluxos em estudo; o protocolo de transporte usado pelas aplicações e os requisitos de QoS mais importantes para as aplicações. O primeiro passo é o mapeamento dos requisitos de QoS das aplicações em requisitos da rede. O segundo passo é a obtenção de uma ou mais funções para expressar a restrição que deve ser usada no problema de otimização. Para isso é necessário o conhecimento do caminho entre a fonte e o destino na rede. A partir do caminho usado pela aplicação, do retardo médio para cada enlace da rede e do RTT exigido pela aplicação é possível construir as funções de restrição. No nosso caso a função de restrição pode ser definida como: a soma dos retardos médios dos enlaces por onde trafegam os dados das aplicações deve ser menor que o menor valor de RTT requerido para as aplicações. Se consideramos estas restrições enlace a enlace, elas formam um conjunto de restrições a ser usado no problema de otimização.

De posse das restrições, só nos resta definir a função de custo. Custo aqui será definido como a quantia paga em dinheiro pela banda utilizada. Uma vez que um enlace pode ser comercialmente mais barato que outros, como quando por exemplo um dos enlaces é uma rede local e o outro é uma conexão via satélite, será definida uma constante de custo  $c_i$  para cada enlace  $i$ . Esta constante pode ser generalizada por uma função, conforme veremos posteriormente.

Na próxima seção mostraremos como resolver o problema de otimização mencionado acima.

### V. O PROBLEMA DE OTIMIZAÇÃO PARA CÁLCULO DA CAPACIDADE MÍNIMA

A rede pode ser representada por um grafo direcionado  $G = (V, E)$ , onde  $V$  representa o número de vértices e  $E$  o número de arestas. Neste caso, os vértices representam os roteadores e as arestas representam os enlaces que existem entre eles.

Para cada par origem-destino ( $ab$ ) o tráfego é transmitido em apenas um único caminho na rede. Tais caminhos são determinados pela tabela de roteamento da rede e escolhidos no conjunto de caminhos  $P = p_{ab}$ . O caminho de transporte, considerando o TCP, é representado por  $r_{ab} = p_{ab} \cup p_{ba}$ . Para cada caminho  $r_{ab}$  e enlace  $l \in E$ ,  $\delta_l(r_{ab}) \in \{0, 1\}$  representa a função que tem valor 1 se o enlace  $l$  faz parte do caminho  $r_{ab}$  e valor 0 caso contrário.

Para calcular o custo mínimo da rede é necessário definir uma função de custo do enlace  $l$ ,  $R(\bullet)$ . Essa função é relacionada à capacidade do enlace  $f_l$  (obtida em função do atraso na fila  $d_l$ ) e ao custo de cada enlace  $c_l$ , ou seja:

$$R(d_l) = c_l f_l(d_l) \quad (2)$$

Essa será a função objetivo para o problema de otimização.

A obtenção da função  $f(d)$  está descrita em detalhes na subseção III-B.

O problema de otimização, denominado atribuição de capacidades (AC) pode ser formulado da seguinte forma. Abaixo estaremos omitindo a dependência da função AC nas variáveis  $d_l$  para simplificar a notação.

Minimizar

$$AC = \sum_{l \in E} R(d_l)$$

Sujeito à seguinte restrição:

$$\sum_{l \in E} \delta_l(r_{ab}) d_l < RTT(r_{ab}), \forall a, b \in V \quad (3)$$

onde  $RTT(r_{ab})$  é o menor RTT médio permitido no caminho  $a \rightarrow b \rightarrow a$  de modo que os requisitos de qualidade de serviço estejam garantidos (lembrando que no caminho  $a \rightarrow b \rightarrow a$  pode haver outros roteadores além de  $a$  e  $b$ ).

Para resolver este problema optamos por um algoritmo de otimização não-linear primal-dual de barreira-logarítmica (pontos interiores), pois este não supõe convexidade das funções.

Note que em um roteador o atraso é inversamente proporcional à capacidade (se a probabilidade de perda é baixa). Desta forma, apesar de podermos transformar nosso problema em um problema de otimização convexa, caso o custo dos canais cresça linearmente com a banda ( $c_i$  constante), um algoritmo que não supõe convexidade torna mais livre a nossa escolha por uma função de custo dos canais. Uma característica do problema, pelo fato do atraso ser inversamente proporcional à capacidade, é que a solução ótima será uma solução com o maior atraso possível.

Este algoritmo de otimização necessita de uma solução inicial para obter uma solução ótima. Claramente  $C_i$  necessita ser finito e portanto  $d_l > 0$ . Além disso,  $d_l$  precisa estar contido dentro do conjunto da restrição da equação (3), ou seja,  $d_l < RTT(r_{ab}), \forall l \in r_{ab}$ .

A solução do problema de otimização fornecerá o atraso  $d_l$  em cada um dos enlaces. Através da função  $f_l(d_l)$  podemos calcular a capacidade de cada enlace  $l$  para que ele tenha o atraso de fila  $d_l$ . Desta forma, o planejamento da rede é constituído pelo conjunto das informações das capacidades dos enlaces ( $f_l(d_l), \forall l \in E$ ) da topologia da rede, representada pelo grafo direcionado  $G$ , e do tamanho das filas considerado no modelo analítico (usado para representar o tráfego agregado).

No que segue, veremos os resultados do modelo proposto.

## VI. RESULTADOS

Nesta seção realizamos os seguintes estudos do método proposto aplicado à topologia de rede da Figura 4:

- 1) Verificamos se a rede em estudo ao utilizar as capacidades calculadas oferece a QoS mínima contratada. Para esta análise simulamos a rede usando o simulador NS [1].
- 2) Comparamos o método proposto neste trabalho com o modelo proposto em [2]. O método descrito em [2] nada mais é do que estimar o desvio padrão do tráfego agregado (com granularidade fina) e calcular a capacidade

necessária para que o tráfego agregado esteja 95% do tempo abaixo da capacidade do enlace. Denominaremos este método como “< 95%”.

### A. O modelo de simulação

O modelo de simulação desenvolvido tem como objetivo representar a rede que se deseja dimensionar e, utilizando as capacidades dos enlaces calculadas conforme descrito na seção III, verificar se os requisitos de latência máxima e vazão mínima das aplicações foram atingidos. Para obter os *traces* empregados como tráfego de fundo, foram utilizadas as medições feitas no roteador do PESC/UFRJ entre os dias 23/07/2003 e 24/07/2003. Os *traces* foram divididos em intervalos de 2 horas e cada um deles representou um enlace diferente na nossa análise. A topologia do modelo de simulação é composta por 5 roteadores e 10 enlaces, conforme mostrado na Figura 4.

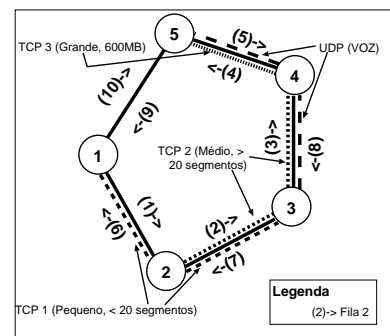


Fig. 4. Topologia usada na simulação

Iremos agora inserir alguns fluxos TCP e UDP e medir se os requisitos de qualidade desses fluxos foram respeitados pelo dimensionamento da rede. Um fluxo TCP de 1500 bytes, representando um acesso a uma página HTML, irá medir a latência. Outros dois fluxos TCP de 600MB e 6MB representarão o download de um vídeo e uma música respectivamente. O objetivo desses dois últimos fluxos é medir a vazão de ambos. Por último, usaremos um fluxo UDP e dele mediremos a perda. O caminho percorrido por esses fluxos pode ser visto na Figura 4. Os requisitos previamente contratados pelos clientes dessa rede são: (a) Latência de transferência de páginas Web inferior a 1.5s. (b) Vazão superior a 512Kbps para arquivos grandes. (c) Probabilidade de perda inferior a  $6 \times 10^{-3}$ .

### B. Resultados obtidos

Executamos a metodologia proposta neste trabalho considerando a topologia ilustrada na Figura 4, e obtivemos para cada enlace as capacidades apresentadas na Tabela I. Na Tabela I, a legenda “Proposto” assinala os valores obtidos pelo método proposto neste trabalho e “< 95%” indica os valores obtidos pelo método descrito em [2]. Podemos notar que como o método descrito em [2] não considera detalhes sobre a QoS, os resultados são, em geral, mais conservadores, o que leva, na maioria dos casos, a escolha de uma capacidade maior do que a necessária para atender os requisitos de QoS definidos.

Enlace	Capacidade (“< 95%”)	Capacidade (Proposto)
Fila1	6.73813Mb	4.518705368Mb
Fila2	11.77320Mb	7.963266735Mb
Fila3	9.69537Mb	11.060235410Mb
Fila4	8.95467Mb	6.659111973Mb
Fila5	10.13850Mb	6.912878224Mb
Fila6	8.50233Mb	5.892622449Mb
Fila7	8.91547Mb	5.679914118Mb
Fila8	5.62828Mb	3.856235399Mb

TABELA I  
CAPACIDADES CALCULADAS PARA OS ENLACES

A Tabela II mostra as estatísticas calculadas para os fluxos TCP da simulação. É possível perceber que tais fluxos conseguiram a QoS desejada. Pode-se observar que o método “< 95%” mostrou-se muito conservador. Isto pode ser visto tanto com relação à latência, que ficou muito abaixo do limite permitido, quanto em relação à vazão, que ficou muito acima do mínimo acordado.

Fluxo	Tamanho	Latência média (s)		Vazão média (Kbps)	
		“< 95%”	Proposto	“< 95%”	Proposto
TCP1	15000 bytes	0.0573232	0.916027	-	-
TCP2	6000KB	-	-	3577.670	1571.880
TCP3	600MB	-	-	2864.166	1376.472

TABELA II  
ESTATÍSTICAS DOS FLUXOS TCP (“< 95%” X PROPOSTO)

Com relação ao fluxo UDP, a porcentagem de pacotes perdidos obtida foi de  $4.28 \times 10^{-3}$  para o método proposto e  $8.3 \times 10^{-4}$  para o método “< 95%”. Mais uma vez o método “< 95%” mostrou-se uma superestimativa, pois a média da perda registrada foi quase 10 vezes menor do que o necessário para se atingir a QoS desejada.

## VII. TRABALHOS RELACIONADOS

O problema do dimensionamento pode ser visto sob a ótica do conhecido problema de “atribuição de capacidade” ou “*capacity assignment problem*” (vide [14] para maiores detalhes). Assim, a literatura na área de dimensionamento é ampla e possui diversos métodos de dimensionamento (vide [3]). Entretanto, a maioria dos artigos presentes na literatura não consideram o comportamento do TCP, não dimensionam considerando os requisitos de QoS e/ou não consideram a rede toda, apenas um roteador “gargalo”.

Em [2] os autores propuseram um método de dimensionamento onde consideram um enlace bem provisionado se a taxa do tráfego agregado fica 95% do tempo abaixo da capacidade máxima. Claramente esta estimativa não leva em consideração parâmetros de QoS e tende a superestimar a capacidade do canal, conforme mostrado na seção anterior.

O trabalho que mais se assemelha ao trabalho apresentado aqui é o [15]. Neste trabalho, os autores também traduzem os requisitos de qualidade de serviço em atraso nas filas dos diversos roteadores e procuram minimizar o custo de acordo com restrições de atraso. Entretanto, a modelagem do tráfego utilizada pelos autores é complexa e pouco automática, além de necessitar de dados complexos como a distribuição do

número de segmentos da janela do TCP no fluxo. Isto faz com que o método seja pouco útil para modelar redes grandes, onde teríamos que coletar e analisar uma enorme quantidade de fluxos.

O método apresentado neste trabalho é mais automático que os métodos presentes na literatura. Os modelos markovianos ocultos (HMMs) foram utilizados para análise de perda de pacotes em [9], [7]. Entretanto, não é do conhecimento dos autores outro trabalho que use HMM para modelar um tráfego agregado. A modelagem proposta neste trabalho, através da utilização de HMMs, mostrou-se mais simples e eficiente para a modelagem do comportamento do tráfego.

## VIII. CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi apresentado um método de dimensionamento de uma rede IP onde se levam em conta fatores como dificuldade de se conseguir medidas do tráfego agregado com granularidade fina, necessidade de um método automático para se dimensionar uma rede grande e necessidade de dimensionar a rede com QoS mas sem superestimar a capacidade dos canais.

O próximo passo deste trabalho é considerar roteadores com políticas ativas de gerenciamento de filas (RED, BLUE, etc) no dimensionamento da rede.

## REFERÊNCIAS

- [1] <http://www.isi.edu/nsnam/ns/doc/index.html>.
- [2] Chadi Barakat, Patrick Thiran, Gianluca Iannaccone, Christophe Diot, and Philippe Owezarski. Modeling Internet backbone traffic at the flow level. *IEEE/ACM Transactions on Signal Processing, Special Issue on networking*, 51(8):2111–2124, 2003.
- [3] Arthur W. Berger and Yaakov Kogan. Dimensioning bandwidth for elastic traffic in high-speed data networks. *IEEE/ACM Transactions on Networking*, 8(5):643–654, 2000.
- [4] R.M.L.R. Carmo, L.R. de Carvalho, E. de Souza e Silva, M.C. Diniz, and R.R. Muntz. Performance/availability modeling with the TANGRAM-II modeling environment. In *Performance Evaluation*, volume 33, pages 45–65. Kluwer, June 1998.
- [5] M. Crovella, M. Taqqu, and A. Bestavros. Heavy-Tailed Probability Distributions in the World Wide Web. In *A Practical Guide To Heavy Tails*, chapter 1, pages 3–26. Chapman & Hall, New York, 1998.
- [6] Allen B. Downey. The structural cause of file size distributions. In *Proc. of IEEE/ACM MASCOTS Symposium*, 2001.
- [7] Flávio P. Duarte, Edmundo de Souza e Silva, and Don Towsley. An adaptive FEC algorithm using hidden Markov chains. In *ACM SIGMETRICS Performance Evaluation Review*, pages 11–13, Setembro 2003.
- [8] Flávio Pimentel Duarte. Algoritmo Adaptativo para Previsão e Recuperação de Perda de Pacotes em Aplicações Multimídias usando Cadeias de Markov Ocultas. Master’s thesis, COPPE/UFRJ, 2003.
- [9] Kavé Salamatian and Sandrine Vatou. Hidden Markov Modeling for Network Communication Channels. In *ACM SIGMETRICS*, 2001.
- [10] Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *IEEE*, pages 257–286, February 1989.
- [11] A. Medina, Nina Taft, K. Salamatian, S. Bhattacharyya, and Christophe Diot. Traffic matrix estimation: Existing techniques and new directions. In *Proc. of the SIGCOMM*, pages 161–174, 2002.
- [12] Luca Muscariello, Marco Mellia, Michela Meo, Marco Ajmone Marsan, and Renato Lo Cigno. An MMPP-Based Hierarchical Model of Internet Traffic. In *to appear in Proc. of IEEE ICC2004*, June 2004.
- [13] J. Padhye, V. Firoiu, D.F. Towsley, and J.F. Kurose. Modeling TCP Reno performance: A simple model and its empirical validation. *IEEE/ACM Transactions on Networking*, 8(2):133–145, April 2000.
- [14] Mischa Schwartz. *Broadband Integrated Networks*. Prentice-Hall, 1998.
- [15] Emilio C. G. Wille, Michele Garetto, Marco Mellia, Emilio Leonardi, and Marco Ajmone Marsan. Considering end-to-end qos constraints in ip network design and planning. Technical report, Politecnico di Torino, 2004.