

Projeto e Otimização de Redes TCP/IP com o Dimensionamento de Enlaces e *Buffers* com Garantia de Qualidade de Serviço

Clovis R. da Costa Bento e Emilio C. Gomes Wille

Resumo—Este trabalho propõe o uso das metaheurísticas GRASP (*Greedy Randomized Adaptive Search Procedure*) e PSO (*Particle Swarm Optimization*) adaptadas para auxiliar no projeto e otimização de redes TCP/IP. Em especial o artigo trata do problema de Atribuição de Fluxos e Capacidades levando em consideração o atraso de pacotes admitido para cada par origem/destino na rede. Propõe-se ainda um procedimento iterativo para a solução do problema de dimensionamento de *buffers*. Cada roteador é modelado por uma fila Markoviana com chegada de pacotes em grupos, de modo a considerar o tráfego em rajadas produzido pelo protocolo TCP. Resultados analíticos, obtidos para algumas topologias dimensionadas, são comparados com resultados de simulação (obtidos com o NS-2) validando a metodologia proposta.

Palavras-Chave—TCP/IP, QoS, Projeto e planejamento de redes de computadores, Metaheurísticas de otimização.

Abstract—This study suggests the use of two metaheuristic methods, the GRASP (*Greedy Randomized Adaptive Search Procedures*) and the PSO (*Particle Swarm Optimization*), customized to help optimize TCP/IP networks. Specially, this article treats the capacity and flow assignment (CFA) problem considering the packet delay allowed within every source/destination pair. This study suggests still a iterative procedure to solve the buffers of the network. Each router is modeled by Markovian queuing with batch arrivals, considering the burst traffic yielded by TCP protocol. Analytic results achieved from studied topologies are faced with simulated results (achieved from NS-2) validating this proposed methodology.

Keywords—TCP/IP, QoS, Telecommunication networks design and planning, Optimization algorithms.

I. INTRODUÇÃO

A elaboração de um projeto de redes, com o dimensionamento dos seus componentes, é uma tarefa complexa diante dos novos cenários da Internet que apresentam demandas de tráfego diversificadas e aplicações emergentes. Tradicionalmente, o projeto de redes de chaveamento de pacotes focava na otimização de custos ou de desempenho pela escolha apropriada das capacidades dos enlaces e de estratégias de roteamento. Dado que os problemas de otimização de capacidades e de roteamento são intimamente interrelacionados, pode ser conveniente resolvê-los em conjunto dando origem a um problema conhecido por Atribuição de Fluxos e Capacidades (*Capacity and Flow Assignment - CFA*) [1]. Existem metaheurísticas tradicionais e já consagradas nas tarefas de otimização, como o GRASP (*Greedy Randomized Adaptive*

Search Procedure) [2], Busca TABU (*TABU Search*) e Algoritmos Genéticos, para a solução de tais problemas. Neste artigo, exploramos o uso do GRASP e da metaheurística PSO (*Particle Swarm Optimization*) [3] com intuito de testar novas alternativas de solução. Neste trabalho, também é proposta uma nova alternativa para o cálculo dos *buffers* da rede. É mostrado também a simulação dos problemas com o Network Simulator afim de se comparar os resultados.

II. ATRIBUIÇÃO DE FLUXOS E CAPACIDADES (CFA)

A infraestrutura de rede IP será representada por um grafo direcionado $G = (V, E)$, onde V é um conjunto de nós e E é um conjunto de arestas. 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. Neste trabalho o CFA tem por objetivo determinar uma rota para o tráfego que flui entre cada par origem/destino (sd) e o conjunto de capacidades para os enlaces (ij) de forma a minimizar o custo da rede sujeito ao atraso de pacotes admitido (selecionado pelo projetista). Denota-se por f_{ij} o fluxo (quantidade de informação que trafega no enlace) e por C_{ij} a capacidade (quantidade máxima que pode ser transmitida pelo enlace), ambos em bps. Os resultados das soluções propostas serão comparados com o método GWFD (*Greedy Weight Flow Deviation*) [1], uma heurística que fornece uma solução subótima para o problema CFA.

A. Modelo matemático

É utilizada a seguinte formulação do CFA, baseada em [1].

Minimizar:

$$Z_{CFA} = \sum_{ij} f(g_{ij}, C_{ij}) \quad (1)$$

sujeito a:

$$K_1 \sum_{ij} \frac{\kappa_{ij}^{sd}}{C_{ij} - f_{ij}} \leq RTT_{sd} - K_2 \sum_{ij} \kappa_{ij}^{sd} g_{ij} \quad \forall (sd) \quad (2)$$

$$f_{ij} = \sum_{sd} \kappa_{ij}^{sd} \gamma_{sd} \quad \forall (ij) \quad (3)$$

$$C_{ij} \geq f_{ij} \geq 0 \quad \forall (ij) \quad (4)$$

$$\kappa_{ij}^{sd} \in \{0, 1\} \quad \forall (ij), \forall (sd) \quad (5)$$

Onde $f(g_{ij}, C_{ij})$, na eq. 1, é a função de custo a ser minimizada e que resulta do produto *Distância* \times *Capacidade* de cada enlace. A eq. 2 é a restrição de QoS, considera o *RTT*

(Round Trip Time) menos o retardo de propagação na rota. O termo $\frac{K_{1,K_2} s^d}{C_{ij} - f_{ij}}$ representa a contribuição do atraso devido ao enfileiramento dos pacotes em cada roteador¹. K_2 converte distância em tempo. As distâncias entre nós é dada pela matriz $D = \{g_{ij}\}$. A eq. 3 corresponde ao fluxo total de cada enlace ij , que está na rota de demanda γ_{sd} . A eq. 4 é a condição de não-negatividade.

B. Proposta de solução com GRASP

O GRASP, neste trabalho, é adaptado para resolver o problema CFA. Na fase de construção, a Lista Restrita de Candidatos (LRC) é construída com a heurística representada na eq. 6. Cada candidato da lista é uma matriz de pesos pré-otimizada.

$$L^{(p+1)} = (g_{ij} \cdot C_{ij})^{(p)} \quad (6)$$

Nesta primeira fase, o conjunto $S = \{L_1, L_2, \dots, L_N\}$, é gerado com N elementos. Cada elemento L_k , é na verdade, uma matriz de pesos gerada e pré-otimizada utilizando-se iterações daquela função. A construção da LRC e do conjunto solução pode ser representada pelo pseudo-código seguinte:

```

PROCEDIMENTO Fase_Construção(semente, Dijkstra( ), CA( ));
(1) INÍCIO
(2)  $S \leftarrow \emptyset$ ;
(3) Entrada de dados:  $N, NI$ ;
(4) ENQUANTO Conjunto  $S < N$  FAÇA;
(5)  $p \leftarrow 0$ ;
(6) Gerar matriz  $L^0$  aleatoriamente;
(7) Computar o Dijkstra para obtenção dos caminhos de menor peso;
(8) Computar os fluxos,  $f^0$ ;
(9) Obtenção das capacidades (CA),  $C^0$ ;
(10) Computar custo total,  $G^0(D, C^0)$ ;
(11) REPETIR;
(12) Computar o peso dos enlaces,  $I_{ij}^{p+1} \leftarrow d_{ij} C_{ij}^p$ ;
(13) Computar o Dijkstra para obtenção dos caminhos de menor peso;
(14) Computar os fluxos,  $f^{p+1}$ ;
(15) Resolver o problema CA para obtenção das capacidades,  $C^{p+1}$ ;
(16) Computar custo total,  $G^{p+1}(D, C^{p+1})$ ;
(17) SE  $G^{p+1}(D, C^{p+1}) < G^p(D, C^p)$  FAÇA;
(18)  $C^* \leftarrow C^{p+1}$ ;  $G^* \leftarrow G^{p+1}$ ;  $L^* \leftarrow L^{p+1}$ ;
(19) SENÃO FIM;
(20)  $p \leftarrow p + 1$ ;
(21) ATÉ Iterações  $\geq NI$ ;
(22) Adicionar  $L^*$  à Lista Restrita de Candidatos(LRC);
(23) Selecionar o elemento  $L_i$  da LRC, aleatoriamente;
(24)  $S \leftarrow S \cup \{L_i\}$ ; Iterações  $\leftarrow 0$ ;
(25) FIM ENQUANTO Retornar  $S$ ;
(26) FIM;
    
```

Onde $G(D, C)$ é a função custo, N é o número de elementos de S e NI as iterações da fase de construção da LRC. Observando-se o pseudo-código, o laço de repetição da linha (11) até a linha (21) gera as iterações que modificam a matriz de pesos, ou seja, a cada passo p a função da eq. 6 cria uma nova matriz de pesos (roteamento). Se esta matriz fornecer um custo menor que a anterior, os valores das capacidades serão armazenados e servirão para guiar a criação da próxima matriz.

A fase de busca local desta proposta baseia-se em gerar uma matriz otimizada L_ξ utilizando partes das matrizes construídas na primeira fase. A princípio a matriz L_ξ assume o valor da melhor matriz da LRC e é melhorada a partir dos elementos

$l_{ij} \in L_k$, para $k = 1, 2, \dots, N$; $\forall (i, j)$. Para selecionar cada elemento l_{ij} a ser adicionado à L_ξ , são feitas permutações entre $l_{ij} \in L_\xi$ e cada $l_{ij} \in L_k$. Na medida em que as permutações ocorrem, o algoritmo reavalia a nova matriz de pesos L_ξ , e a melhor solução é armazenada, de modo que, no final, o resultado é uma matriz L_ξ , tal que $f(L_\xi) < f(L_k)$, $\forall k$. As permutações ocorrem com probabilidade P , de modo que nem todos os elementos serão testados. As permutações, como estratégia de otimização, baseiam-se no fato de se forçar a mudança de rotas a cada mudança dos elementos ij .

```

PROCEDIMENTO Fase_BuscaLocal(Dijkstra( ), CA( ));
(1) INÍCIO
(2) LER  $S$ ;
(3) Selecionar o melhor elemento de  $S$  e obter  $L_\xi$ ;
(4)  $L^0 \leftarrow L_\xi$ ;
(5) Computar o Dijkstra sobre  $L^0$  para os caminhos de menor peso;
(6) Computar os fluxos,  $f^0$ ;
(7) Resolver o CFA para obtenção das capacidades,  $C^0$ ;
(8) Computar custo total,  $G^0(D, C^0)$ ;
(9) Inicializar variáveis:  $p \leftarrow 0$ ;
(10) REPETIR;
(11) PARA  $i \leftarrow 0$  até  $m$  FAÇA;
(12) PARA  $j \leftarrow 0$  até  $n$  FAÇA;
(13) PARA  $k \leftarrow 0$  até  $N$  FAÇA;
(14) SE  $prob = P$  FAÇA;
(15) Trocar um elemento  $l_{ij} \in L^p$  com  $l_{ij} \in L_k$  e gerar  $L^{p+1}$ ;
(16) Computar o Dijkstra para os caminhos de menor peso;
(17) Computar os fluxos,  $f^{p+1}$ ;
(18) Resolver o problema CA para obtenção das capacidades,  $C^{p+1}$ ;
(19) Computar custo total,  $G^{p+1}(D, C^{p+1})$ ;
(20) SE  $G^{p+1}(D, C^{p+1}) < G^p(D, C^p)$  FAÇA;
(21)  $L^p \leftarrow L^{p+1}$ ;  $L_\xi \leftarrow L^{p+1}$ ;
(22) SENÃO FIM;
(23) SENÃO FIM;
(24)  $p \leftarrow p + 1$ ;
(25) FIM PARA;
(26) FIM PARA;
(27) FIM PARA;
(28) ATÉ Critério ser alcançado;
(29) Retornar  $L_\xi$ ;
(30) FIM;
    
```

Onde $prob$ é a probabilidade de haver a permutação do elemento l_{ij} .

C. Proposta de solução com PSO

O PSO foi adaptado para resolver o CFA. A chave do processo foi fazer o PSO trabalhar com partículas em forma de matrizes. As matrizes de pesos são as variáveis (ou partículas) a serem otimizadas. Considera-se cada matriz de pesos como uma partícula multidimensional, onde cada elemento ij é tratado como uma das dimensões. A formulação matemática para o PSO, neste trabalho, considera o modelo canônico, conforme as eqs. 7 e 8 e explicado com detalhes em [3].

$$v_{i,d}(n+1) = \omega v_{i,d}(n) + c_1 \varphi_{1,d} (p_{i,d} - x_{i,d}(n)) + c_2 \varphi_{2,d} (p_{g,d} - x_{i,d}(n)) \quad (7)$$

$$x_{i,d}(n+1) = x_{i,d}(n) + v_{i,d}(n+1) \quad (8)$$

onde, $x_{i,d}$ é um escalar que representa a posição da partícula i na dimensão d , e $v_{i,d}$ a velocidade, em qualquer instante. O ω é o fator de inércia, $p_{i,d}$ é a melhor posição anterior, $p_{g,d}$ é a melhor posição global, entre todas as partículas da vizinhança. As constantes c_1 e c_2 controlam o impacto de cada um dos termos sobre a nova velocidade, φ_1 e φ_2 são variáveis uniformemente distribuídas no intervalo $[0, 1]$, e representam

¹Modelado como uma fila Markoviana, $M_{[X]}/M/1$, com batch arrivals.

um grau de individualidade e sociabilidade, respectivamente. Usualmente usa-se $c_1 = c_2 = 2$, mas alguns experimentos indicam que $c_1 = c_2 = 1$, 49 produzem melhores resultados [4]. Na seqüência, o pseudo-código representa o funcionamento do PSO.

```

PROCEDIMENTO PSO(semente, Dijkstra(), CA());
(1) INÍCIO
(2) GERAR a população de partículas  $P_i$  de tamanho  $S$ ;
(3) PARA  $i \leftarrow 0$  até  $S$  FAÇA;
(4)   Computar o Dijkstra e encontrar os fluxos  $f^n$ ;
(5)   Resolver o problema CA para obtenção das capacidades,  $C^n$ ;
(6)   Computar custo,  $G^n(D, C^n)$ ;
(7)   ENCONTRAR a partícula ( $P_\xi$ ) de menor custo;
(8) FIM PARA;
(9) REPETIR;
(10)  Computar a melhor posição geral informada,  $P_{g,d}$ ;
(11)  Atualizar as velocidades:  $v_{i,d}(n+1)$ ;
(12)  Atualizar as posições:  $x_{i,d}(n+1)$ ;
(13)  Computar o Dijkstra e encontrar os fluxos  $f^{n+1}$ ;
(14)  Resolver o problema CA para obtenção das capacidades,  $C^{n+1}$ ;
(15)  Computar custo total,  $G^{n+1}(D, C^{n+1})$ ;
(16)  SE  $G^{n+1}(D, C^{n+1}) < G^n(D, C^n)$  FAÇA;
(17)     $P_\xi \leftarrow P(n+1)$ ;
(18)  SENÃO FIM;
(19)   $n \leftarrow n + 1$ ;
(20)  ATÉ Satisfazer o critério de parada;
(21)  RETORNAR  $P_\xi$ ;
(22) FIM;
    
```

III. ATRIBUIÇÃO DE BUFFER (BA)

O problema da Atribuição de Buffer (BA), tem o objetivo de dimensionar os buffers com o menor tamanho possível, fazendo com que a probabilidade de perda de pacotes em cada rota não ultrapasse o valor PR_{sd} estabelecido pelo projetista. O BA é formulado através do seguinte problema de otimização:

Minimizar:

$$Z_{BA} = \sum_{i,j} h(B_{ij}) \quad (9)$$

sujeito a:

$$\sum_{ij} \kappa_{ij}^{sd} p(B_{ij}, C_{ij}, f_{ij}, [X]) \leq PR_{sd} \quad \forall (sd), \forall (ij) \quad (10)$$

$$B_{ij} \geq 0 \quad \forall (ij) \quad (11)$$

A eq.9 é a função objetivo que representa o custo total dos buffers na rede. É uma função linear do tipo $h(B_{ij}) = \sum_{ij} b_{ij} \cdot B_{ij}$, onde b_{ij} é o custo monetário e B_{ij} é o buffer do enlace ij . A eq.10 é a restrição de perda de pacotes na rota sd . O parâmetro PR_{sd} indica a máxima perda na rota sd . $p(B_{ij}, C_{ij}, f_{ij}, [X])$ é a probabilidade de perda de pacotes em cada roteador². A eq.11 garante que nenhum buffer terá tamanho negativo.

Em [1] o problema é resolvido numericamente e o processo de otimização utiliza uma combinação dos métodos: Método da Barreira Logarítmica e Método das Coordenadas Cíclicas.

A. Proposta de solução do BA

Nesta proposta, o primeiro passo consiste em substituir $p(B_{ij}, C_{ij}, f_{ij}, [X])$ por uma função analítica aproximada (PP) na formulação do BA (uma vez que aquela não é uma função fechada e contínua). O segundo passo corresponde

²Modelado como uma fila Markoviana, $M_{[X]}/M/1/B$, com batch arrivals.

a usar uma heurística simples (batizada como Heurística da Decomposição) para a obtenção de uma solução para o problema BA. A função PP pode ser ajustada por meio de dois parâmetros de modo a se aproximar a probabilidade de perda obtida da fila $M_{[X]}/M/1/B$. No processo de otimização dos buffers, foi utilizado o algoritmo conhecido como Método das Iterações Lineares (MIL) juntamente com uma adaptação do Método da Seção Áurea (MSA). Este procedimento permite efetuar o cálculo dos buffers e o ajuste das curvas de modo simultâneo. A função PP aproximada está representada a seguir [5]:

$$PP_{ij} = \delta \frac{(1 - \rho_{ij}^{\alpha_{ij}}) \rho_{ij}^{\alpha_{ij} B_{ij}}}{1 - \rho_{ij}^{\alpha_{ij} (B_{ij} + 1)}} \quad (12)$$

onde PP_{ij} representa a probabilidade de perda de cada enlace (i, j) , δ é uma constante de ajuste da altura da curva e α ajusta a sua inclinação, e ρ é o fator de utilização do enlace. Aplicando-se o método dos multiplicadores de Lagrange à rota (sd) , obtém-se:

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

Após o cálculo de PE_{ij}^{sd} para todas as rotas possíveis, obtém-se valores admissíveis para as variáveis PE_{ij}^* , no problema original, fazendo:

$$PE_{ij}^* = \min_{sd} (PE_{ij}^{sd}) \quad \forall (sd), \forall (ij) \quad (14)$$

O procedimento de otimização pode ser representado pelo pseudo-código a seguir:

```

PROCEDIMENTO BA(Aureal(), Aurea2());
(1) INÍCIO
(2) Ler dados de entrada:  $f_{ij}, C_{ij}, \rho_{ij}, PR_{sd}$ ;
(3) REPETIR para cada rota  $(s, d)$ ;
(4)   Atribuir valores iniciais:  $PE_{ij}^0 \leftarrow \frac{PR_{sd}}{NER}, p \leftarrow 0$ ;
(5)   REPETIR para cada enlace  $(i, j)$ ;
(6)     ENQUANTO o buffer  $B_{ij}$  não encontrado FAÇA;
(7)       Executar Aureal( $PE_{ij}^p$ ), e obter  $B_{ij}^p \leftarrow f_N^{-1}(PE_{ij}^p)$ ;
(8)       Executar Aurea2( $PE_{ij}^p$ ), e obter  $\alpha_{ij} \leftarrow f_A^{-1}(B_{ij}^p, PE_{ij}^p)$ ;
(9)       Computar  $PE_{ij}^p$ , com a eq.13;
(10)       $PE_{ij}^{p+1} \leftarrow PE_{ij}^p$ ;
(11)       $p \leftarrow p + 1$ ;
(12)      FIM se o valor de  $B_{ij}$  foi encontrado;
(13)       $p \leftarrow 0$ ;
(14)      FIM REPETIR;
(15)      FIM REPETIR;
(16)      Selecionar o maior valor  $B_{ij}$  entre  $B_{ij}^{sd}$ ;
(17)      RETORNAR  $B_{ij}$ ;
(18) FIM;
    
```

Nota-se que no algoritmo proposto o Método da Seção Áurea 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 (sd) , é utilizado para estabelecer valores iniciais para as probabilidades de perda. No passo 16 do procedimento é selecionado

o maior valor do *buffer* encontrado para o mesmo enlace, isto é o mesmo que selecionar a menor probabilidade de perda de pacotes, uma vez que os parâmetros são inversos (ver eq. 14).

IV. RESULTADOS NUMÉRICOS E SIMULAÇÕES

Para validar os métodos GRASP e PSO, são considerados alguns cenários com topologias de 6 e 20 nós, denominadas neste trabalho de N6 e N20, respectivamente. Para cada topologia os problemas CFA e BA associados foram solucionados utilizando-se os procedimentos propostos neste artigo. Os cálculos consideram os seguintes parâmetros: $RTT = 0,07 s$; $PR_{sd} = 0,01$.

A. Topologia N6

Nesta topologia com 6 nós são utilizados 16 enlaces, 25 rotas, a demanda em cada rota (par origem-destino) varia entre 8 e 32 Mbps (com média em 16 Mbps).

Na tab.I estão representadas a somatória das capacidades, dos *buffers* e dos custos obtidos por cada método. A primeira coluna refere-se aos enlaces, individualmente, na seqüência tem-se as capacidades *C* em Mbps e os *buffers* em número de pacotes. Na parte inferior tem-se a somatória das capacidades e dos *buffers* de todos os enlaces, e no final é mostrado o valor do custo em cada caso. O GWFD está servindo como referência. Ainda, para efeito de comparação, consta na tabela o cálculo das capacidades e *buffers* considerando-se a abordagem CA-P que utiliza uma matriz de pesos fixa para o roteamento.

TABELA I
RESULTADOS PARA A TOPOLOGIA N6.

Enlace	CA-P		GWFD		GRASP		PSO	
	C	Buffer	C	Buffer	C	Buffer	C	Buffer
1	17,41	274	55,34	248	44,46	297	44,46	297
2	43,34	222	35,34	194	51,34	268	51,34	282
3	9,48	157	27,72	246	18,26	251	18,26	251
4	41,72	348	41,72	351	41,72	363	41,72	363
5	31,65	257	22,31	192	11,45	254	11,45	186
6	34,31	274	42,31	320	42,31	330	42,31	328
7	17,45	269	17,45	324	17,45	335	17,45	324
8	9,45	216	27,17	269	37,06	278	37,06	292
9	19,73	189	19,05	212	19,05	212	19,05	212
10	44,11	319	35,05	342	60,62	433	60,62	463
11	5,46	101	31,05	288	31,05	303	31,05	303
12	53,56	323	35,05	342	60,62	433	60,62	463
13	17,55	310	17,55	254	*	*	*	*
14	26,69	295	*	*	*	*	*	*
15	13,51	274	13,51	207	13,51	207	13,51	284
16	52,10	395	17,51	259	17,51	259	17,51	259
Soma	437,51	4185	438,11	4048	466,41	4223	466,41	4307
Custo	155805,99		131446,41		130654,30		130654,30	

Nota: * Indica que o enlace não é utilizado.

Para reforçar a eficácia de $d \times C$ como critério de otimização, observa-se que os menores valores de capacidades não representam, necessariamente, os menores valores de custo na rede (porque não levam em conta o comprimento do enlace). Como exemplo, observa-se que o GRASP e PSO somam capacidades na ordem de 466, 41 (Mbps), porém, geram custos totais na ordem de 130654, 30 (Mbps.km), portanto, menores que os outros métodos considerados.

A representação do tempo gasto pelos algoritmos utiliza o *clock* do processador como métrica. A relação entre *clocks* e tempo (em segundos) é de aproximadamente 100 *clocks*/segundo. Os resultados de cada método estão representados no gráfico da fig. 1.

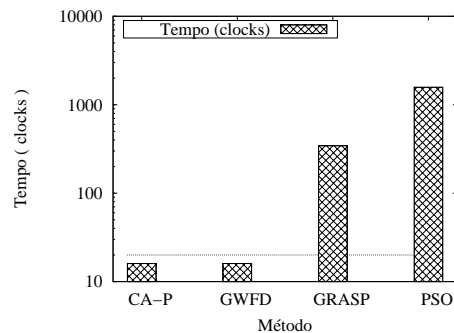


Fig. 1. Tempos de processamento para a topologia N6.

B. Topologia N20

Nesta topologia com 20 nós são utilizados 54 enlaces, 90 rotas, a demanda em cada rota varia entre 5 e 20 Mbps (com média em 10 Mbps). Os enlaces não são mostrados individualmente na tab. II para simplificar a representação. Os tempos de processamento estão mostrados na fig. 2.

TABELA II
RESULTADOS PARA A TOPOLOGIA N20.

	CA-P		GWFD		GRASP		PSO	
	C	Buffer	C	Buffer	C	Buffer	C	Buffer
Soma	4120,77	14403	3659,75	18014	3679,02	17984	3674,68	18455
Custo	1351234,78		703791,01		701685,52		703462,87	

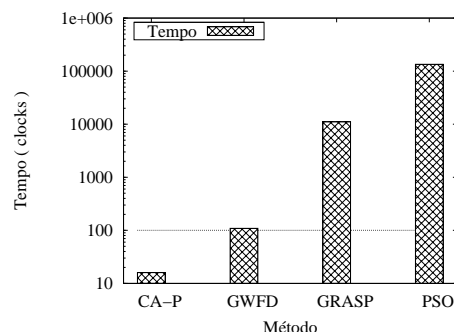


Fig. 2. Tempos de processamento para a topologia N20.

C. Simulações no NS

Nesta seção são apresentadas as simulações executadas no *Network Simulator* versão 2.29 (NS-2.29) para N6 e N20, com o objetivo de verificar se o dimensionamento realizado com os algoritmos se confirma na prática. Nas simulações foi utilizado o *Batch Means Method*, método que divide a tarefa de simulação em diversos blocos e calcula a média entre eles [6]. A dinâmica adotada é a seguinte: diversas conexões TCP (simultâneas) são estabelecidas (segundo um processo de Poisson) entre transmissores e receptores escolhidos de forma aleatória de forma a gerar um tráfego agregado. Cada conexão consiste de uma simples transmissão de um arquivo tipo HTTP, cuja distribuição de tamanhos (1, 2, 4, 6, 10, 19 e 195 pacotes) é aquela mencionada em [7]. Os resultados obtidos na

simulação (níveis de transporte e rede) são comparados com os valores calculados numericamente. Para a realização desta tarefa, são escolhidas rotas com três e quatro enlaces na rede para serem testadas na simulação. Os parâmetros calculados de cada enlace, pela solução do CFA, servem de entrada para o simulador, que fornece os resultados ao nível de rede e de transporte.

A tab. III mostra o dimensionamento para a topologia N6. Resultados de simulação no NS e do modelo $M_{[X]}/M/1/B$ são apresentados na tab. IV. Estão representados os fluxos (f), capacidades (C), buffers (B), atraso médio ($E[T]$), probabilidade de perda (p) e a ocupação média da fila ($E[N]$). O atraso de propagação da rota (causado pelo meio físico) considerado é de $t_p = 0,0039$ s. Observa-se boa aproximação entre os valores previstos e os obtidos pela simulação. A margem de erro das simulações é de $\pm 30\%$ para $E[T]$ e $E[N]$ e de $\pm 15\%$ para p_{ij} .

TABELA III

RESULTADOS DO CFA E BA, ROTA COM TRÊS ENLACES NA REDE.

Enlace	CFA e BA			
	f	C	B	$E[T]$
1	30,00	34,31	274	0,0226
2	38,00	41,72	348	0,0263
3	36,00	43,34	222	0,0133
Soma	104,00	119,37	844	0,0622
$RTT(s)$	0,0700			

TABELA IV

COMPARAÇÃO: MÉTODO NUMÉRICO E NS, ROTA COM TRÊS ENLACES.

Enlace	Simulador NS			$M_{[X]}/M/1/B$		
	p	$E[T]$	$E[N]$	p	$E[T]$	$E[N]$
1	0,002990	0,0357	91,65	0,003183	0,0271	64,05
2	0,003743	0,0414	134,68	0,004460	0,0335	99,06
3	0,002079	0,0200	61,48	0,002350	0,0147	43,45
Soma	0,008812	0,0971	287,81	0,009993	0,0753	206,56
$RTT(s)$	0,0971			0,0753		

O gráfico da fig. 3 mostra a latência na transmissão de arquivos de vários tamanhos calculado usando o modelo analítico do protocolo TCP reportado em [8] (modelo CSA). 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 [8]. 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).

A tab. V mostra o dimensionamento para a topologia N20. Resultados de simulação no NS e do modelo $M_{[X]}/M/1/B$ para N20 são apresentados na tab. VI (com $t_p = 0,0065$ s) e a fig. 4 mostra os resultados da latência em função do tamanho do fluxo TCP em uma rota com quatro enlaces. A margem de erro das simulações é de $\pm 15\%$ e $\pm 30\%$ de acordo com o tamanho do fluxo TCP. Para esta topologia não é mostrada a tabela comparativa.

V. TESTE COMPARATIVO DO GRASP E PSO

Os métodos GRASP e PSO foram testados para se observar o comportamento e a eficiência de cada um, considerando-se a influência dos seus principais parâmetros.

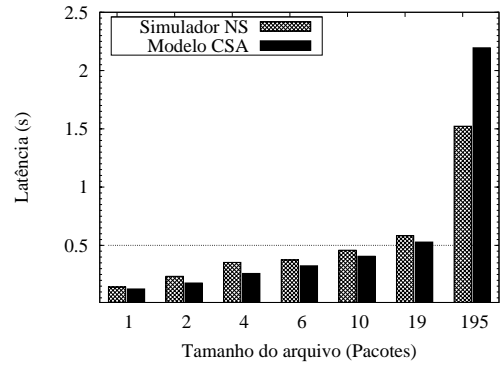


Fig. 3. Latência dos fluxos TCP na rota com três enlaces para rede N6. Comparação entre o método numérico e simulação com o NS.

TABELA V

RESULTADOS DO CFA E BA, ROTA COM QUATRO ENLACES NA REDE.

Enlace	CFA e BA			
	f	C	B	$E[T]$
1	25,00	35,57	145	0,0092
2	19,00	23,75	178	0,0206
3	17,00	21,26	183	0,0229
4	73,00	93,58	206	0,0047
Soma	134,00	174,16	712	0,0574
$RTT(s)$	0,0700			

TABELA VI

COMPARAÇÃO: MÉTODO NUMÉRICO E NS, ROTA COM QUATRO ENLACES.

Enlace	Simulador NS			$M_{[X]}/M/1/B$		
	p	$E[T]$	$E[N]$	p	$E[T]$	$E[N]$
1	0,001092	0,0124	26,64	0,001441	0,0100	19,80
2	0,002345	0,0259	42,20	0,003074	0,0253	34,00
3	0,002203	0,0312	45,38	0,002720	0,0267	34,16
4	0,000985	0,0071	44,47	0,001024	0,0058	31,09
Soma	0,006626	0,0766	158,69	0,008259	0,0678	119,05
$RTT(s)$	0,0766			0,0678		

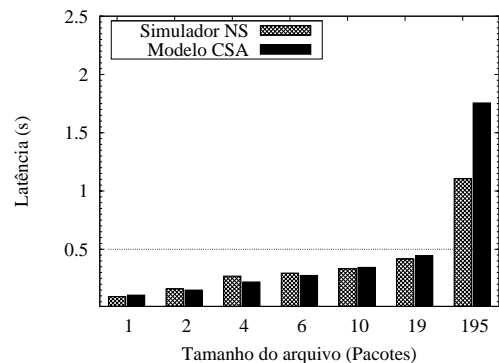


Fig. 4. Latência dos fluxos TCP na rota com quatro enlaces para rede N20.

Foi utilizado um gerador de topologias para efetuar baterias de execuções e a obtenção dos valores médios dos custos e do tempo médio de processamento (\bar{t}) registrado em *clocks*. Os parâmetros de interesse nos testes são aqueles que impactam diretamente no desempenho do algoritmo. No GRASP, os parâmetros considerados são: o tamanho do conjunto solução (S), a probabilidade de troca dos elementos das matrizes de roteamento, no PSO: o tamanho da população. Foram executados testes com diferentes sementes geradoras (para

garantir a geração de diferentes topologias) e averiguar a estabilidade do método e a garantia de convergência para valores otimizados. A rede considerada é a N20 com cinco topologias diferentes geradas automaticamente. O GWFD não participou desta comparação por ser um método que não necessita gerar conjuntos de matrizes (caso das soluções iniciais para o GRASP e das partículas para o PSO).

Na tab. VII estão registrados os valores dos custos e tempo de processamento de acordo com o tamanho do conjunto S do GRASP e da população de partículas do PSO.

TABELA VII

RESULTADOS DOS TESTES COM DIFERENTES TAMANHOS DO CONJUNTO SOLUÇÃO S NO GRASP E DIFERENTES POPULAÇÕES NO PSO.

GRASP			PSO		
Conjunto S	Custo	\bar{t}	População	Custo	\bar{t}
5	822849,57	5790,40	5	855116,10	39215,60
10	807044,79	12506,20	10	839237,14	78390,60
20	783233,54	23428,00	20	802247,73	157818,60
30	782801,87	35687,60	30	791832,52	236681,20
50	785522,20	58653,20	50	775802,20	391437,80

Na tab. VIII está anotado um teste de desempenho do GRASP considerando-se a probabilidade de troca de elementos das matrizes de roteamento durante a fase da busca local, na rede N20. Este parâmetro causa impacto direto no tempo de execução do algoritmo. As curvas de *fitness* dos algoritmos são mostradas na figura 5.

TABELA VIII

RESULTADOS PARA DIFERENTES PROBABILIDADES DE TROCA NA FASE DE BUSCA LOCAL DO GRASP.

GRASP		
Probabilidade	Custo	\bar{t}
0,1	797701,29	11409,20
0,2	788877,83	17781,40
0,5	781565,73	35503,20
0,9	781225,96	60050,00
1,0	780971,83	65656,40

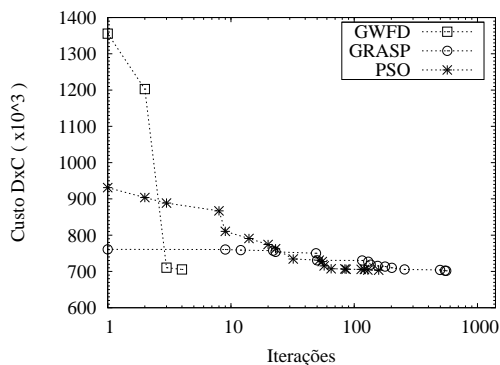


Fig. 5. Gráfico de *fitness* para a topologia N20.

Uma comparação do tempo de processamento entre o GWFD, GRASP e o PSO é mostrada na tab. IX. Observa-se que o tempo cresce mais sensivelmente no PSO.

TABELA IX

COMPARAÇÃO DOS TEMPOS DE PROCESSAMENTO.

Tempo de processamento			
Topologia	GWFD	GRASP	PSO
N6	16	344	1578
N10	16	1031	969
N20	109	11109	134360
N40	1813	2952391	9209047

VI. CONCLUSÕES

Neste artigo foi considerado o dimensionamento de redes de chaveamento de pacotes visando a garantia de QoS. Em especial tratou-se do problema CFA onde capacidades dos enlaces e roteamento de tráfego são as variáveis a serem determinadas. Também apresentou-se um novo método de solução para o problema do dimensionamento de *buffers*.

Observando os resultados numéricos e as simulações, verifica-se a eficácia dos métodos GRASP e PSO propostos. O GRASP e o PSO mostram resultados similares. Porém, o tempo de processamento cresce mais sensivelmente no PSO, como mostra a tab. IX. A razão para o PSO gastar tanto tempo de processamento está na complexidade do algoritmo. A maior parte do peso da execução do PSO está no processamento das partículas, que são geradas no início e atualizadas a cada iteração. Na medida em que uma partícula representa uma matriz de roteamento, a tarefa de otimização torna-se cada vez mais onerosa em função do tamanho da rede. Em relação ao *fitness*, o PSO apresentou uma curva mais comportada, o que representa uma convergência mais suave e maior controle sobre o algoritmo a partir dos seus parâmetros. Por outro lado o GWFD, embora rápido, é particularmente ineficiente quando o problema apresenta muitos mínimos locais.

As simulações com o NS comprovam a eficácia da metodologia proposta. Os gráficos mostram uma boa aproximação entre os valores calculados e simulados. Como observação final, ressalta-se a necessidade de se resolver o problema conjunto e não um simples problema de atribuição de capacidades (com roteamento conhecido). Eximindo-se de considerar o subproblema de otimização de roteamento resulta em soluções mais onerosas (como mostrado na tab. I).

REFERÊNCIAS

- [1] E. C. G. Wille, M. Mellia, E. Leonardi, M. Ajmone Marsan, "Algorithms for IP Networks Design with End-to-End QoS Constraints". Computer Networks, v. 50, n. 8, p. 1086-1103, 2006.
- [2] M. G. C. Resende, "Greedy Randomized Adaptive Search Procedures". AT&T Labs Research Technical Report TD-53RSJY, version 2. Agosto 2002.
- [3] J. Kennedy, R. C. Eberhart, "Particle Swarm Optimization", Proc. IEEE Int. Conf. Neural Networks. Perth, Austrália, p. 1942-1948, Nov. 1995.
- [4] A. Abraham, He Guo, Hongbo Liu, "Swarm Intelligence: Foundations, Perspectives and Applications". Computational Intelligence (SCI), p. 3-25, Berlin, 2006.
- [5] E. Wille, E. Yabcznski, C. R. da Costa Bento, "Algoritmos para Solução do Problema de Dimensionamento de Buffers em Roteadores IP", SBRC 2008, p. 217-229, Rio de Janeiro, Maio 2008.
- [6] H. A. Taha, "Operations Research: An Introduction", Prentice Hall, 7a Edição, 2002.
- [7] E. C. G. Wille, M. Garetto, M. Mellia, E. Leonardi, M. Ajmone Marsan, "Considering End-to-End QoS in IP Network Design". NETWORKS 2004, Vienna, Austria, Junho, 2004.
- [8] N. Cardwell, S. Savage, T. Anderson, "Modeling TCP Latency", IEEE Infocom 00, Tel Aviv, Israel. Março 2000.