ROTEAMENTO DISTRIBUÍDO E ADAPTATIVO PARA REDES DE TELEFONIA BASEADO EM AGENTES AUTÔNOMOS

K. Vittori, A. F. R. Araújo

Departamento de Engenharia Elétrica
Universidade de São Paulo
Av. Dr. Carlos Botelho, 1465
13560-250, São Carlos, SP, Brasil

ABSTRACT

Este artigo apresenta um algoritmo de roteamento distribuído e adaptativo baseado em agentes autônomos, denominado Agentes-Q, que combina três técnicas: aprendizagem-Q, aprendizagem por reforço dual e o método de otimização baseado no comportamento de colônias de formigas. A esta combinação foram adicionados dois mecanismos que aumentam a adaptação do sistema. O algoritmo é constituído por um conjunto de agentes móveis que percorrem a rede em busca dos caminhos menores e menos carregados. Os agentes verificam o estado da rede a cada ponto visitado e utilizam esta informação para atualizar as variáveis empregadas na seleção das rotas, o que ocorre através das regras usadas pelas duas técnicas de aprendizagem selecionadas. O algoritmo proposto foi aplicado a um modelo da rede de telefonia da empresa AT&T e seu desempenho, medido pela porcentagem de chamadas perdidas, foi comparado ao de dois algoritmos baseados no método de otimização utilizado. Os Agentes-Q obtiveram menores perdas e se adaptaram melhor que os demais a variações dos padrões de tráfego, nível de carga e topologia da rede e a condições de falha dos roteadores.

1. INTRODUÇÃO

O roteamento é um dos principais componentes do sistema de gerenciamento das redes de telecomunicações, sendo responsável pelo direcionamento do tráfego entre pontos de origem e destino. A contínua evolução das redes torna necessária a utilização de novos sistemas de roteamento, que sejam capazes de acompanhar de forma rápida e eficaz as diversas situações enfrentadas atualmente pelas mesmas.

Dentro deste contexto, este artigo apresenta um algoritmo de roteamento distribuído e adaptativo, denominado Agentes-Q, baseado em três técnicas: aprendizagem-Q [13][14] e aprendizagem por reforço dual [6], que constituem abordagens de aprendizagem por reforço [7] e o método de otimização baseado no comportamento coletivo de formigas [3][4][5]. Deste modo, o algoritmo proposto é composto por um conjunto de agentes que se movimentam sobre a rede de forma distribuída e independente, buscando os caminhos menores e menos carregados. Durante seu movimento, os agentes avaliam a rota percorrida coletando dados sobre o estado dos pontos visitados e utilizando os mesmos para atualizar as variáveis empregadas na seleção das rotas. Assim, os agentes são capazes de descobrir continuamente os melhores caminhos.

A utilização da aprendizagem-Q [13][14] está relacionada com a capacidade do agente aprender a lidar com o ambiente sem a necessidade de um modelo prévio e completo sobre o seu comportamento, que constitui uma característica do problema de roteamento. Além disso, esta estratégia assegura convergência para uma política de decisão ótima [13], desde que os estados e ações do ambiente sejam continuamente visitados pelo agente. A aprendizagem por reforço dual [6] foi selecionada devido ao aumento da velocidade de adaptação dos agentes por ela proporcionado, uma vez que ela permite que um número maior de variáveis utilizadas na seleção das rotas seja atualizado a cada ponto visitado. A escolha do método de otimização baseado no comportamento de formigas se refere a presença de um conjunto de agentes que se movem sobre o ambiente de modo simultâneo e independente, cooperando entre si na obtenção das melhores soluções. À combinação destas estratégias foram adicionados dois mecanimos que aprimoram a exploração de caminhos alternativos e aumentam a adaptação do algoritmo.

O algoritmo Agentes-Q foi aplicado a um modelo da rede de telefonia da empresa AT&T e seu desempenho, medido pela porcentagem de chamadas perdidas, foi comparado ao dos sistemas ABC (*Ant-Based Control*) [10] e ABC Esperto [1], baseados no comportamento coletivo de formigas. O algoritmo proposto obteve melhor desempenho que os demais sob variações dos padrões de tráfego, nível de carga e topologia da rede, como também sob condições de falha dos roteadores.

Este artigo é organizado da seguinte forma. Na Seção 2 são apresentados os principais aspectos do problema de roteamento em redes de telecomunicações, juntamente com alguns algoritmos baseados nas técnicas utilizadas. Na Seção 3 o algoritmo proposto é explicado, seguido na Seção 4 pelos experimentos realizados. Na Seção 5 são discutidos os resultados obtidos e finalmente, na Seção 6, são sumarizadas as conclusões obtidas e apresentadas as perspectivas de trabalho futuro.

2. ROTEAMENTO EM REDES DE TELECOMUNICAÇÕES

O problema de roteamento em redes de telecomunicações consiste em conduzir o tráfego de informações ao destino

desejado maximizando o desempenho da rede e minimizando seus custos de operação. Ele é representado por um grafo não-orientado $G = (R, T)$, onde os nós pertencentes ao conjunto R representam os roteadores e os laços pertencentes a T representam as linhas de transmissão. Cada nó é caracterizado por uma capacidade de processamento/armazenamento, enquanto cada linha é caracterizada por uma capacidade de transmissão e atraso.

Os diversos algoritmos de roteamento existentes possuem as seguintes funções em comum [11]: (i) a reunião e distribuição de informação sobre o estado da rede; (ii) a seleção das rotas a serem percorridas pelo tráfego e (iii) a sua condução ao destino desejado através dos caminhos selecionados. A escolha das rotas pode ser baseada em diversas métricas, como: comprimento do caminho, atraso, largura de banda e carga. As informações sobre o estado da rede relativas a métrica utilizada são armazenadas em uma estrutura de dados chamada tabela de roteamento, presente em cada nó da rede. O tráfego pode ser enviado sob dois paradigmas: a comutação de circuitos e a comutação de pacotes. Na primeira tecnologia, os recursos da rede são reservados antes da transmissão da informação e permanecem dedicados a uma conexão enquanto existir a comunicação. Assim, toda a informação relativa a um dado par origem-destino percorre a mesma rota. Na segunda tecnologia, as informações transmitidas são divididas em unidades menores, os pacotes, onde cada um deles pode seguir uma rota diferente em direção ao destino. Neste caso, os recursos da rede são utilizados somente durante a transmissão dos pacotes.

Ao aplicar a aprendizagem por reforço ao problema de roteamento, o roteador é considerado um agente e a rede o ambiente sobre o qual ele atua. O roteador percebe o estado da rede s_t a cada instante de tempo t, realiza uma ação a_t, que é a seleção da rota, e recebe da rede um sinal de reforço r_t como consequência da ação realizada. Este sinal é usado para atualizar a estimativa do valor do estado $V(s_t)$ ou do par estado-ação $Q(s_t, a_t)$ com relação a métrica utilizada.

O primeiro algoritmo de roteamento baseado na aprendizagem-Q, Roteamento-Q [2], foi aplicado a redes de comutação de pacotes. Neste sistema, o roteador estima a cada instante o tempo $Q_i(j,d)$ dispendido para enviar um pacote do nó i ao destino d através de seu vizinho j. A seleção da rota a ser percorrida pelo pacote é determinística. Assim, o nó vizinho j que apresenta $Q_i(j,d)$ mínimo é selecionado. Os valores-Q relativos a ação do agente são atualizados a cada nó visitado e o reforço é igual ao atraso sofrido em sua transmissão. O Roteamento-Q obteve menor atraso médio no envio dos pacotes sobre a rede que algoritmos estáticos baseados no menor caminho sob variações no nível de carga da rede.

O Roteamento-Q com Reforço Dual [8], baseado na aprendizagem-Q [13][14] e aprendizagem por reforço dual [6], acrescenta ao Roteamento-Q a atualização dos valores-Q na direção contrária a ação do agente. Esta atualização se baseia no valor-Q armazenado na tabela de roteamento do último nó visitado pelo pacote em direção a sua origem e no atraso a ser sofrido pelo pacote no nó atual. Desta forma, dois valores-Q são atualizados a cada nó percorrido pelo pacote, aumentando o volume de informação apurada sobre o estado atual da rede. Isto permitiu que o sistema se adaptasse mais rapidamente a mudanças no nível de carga e topologia da rede, apresentando melhor desempenho que o algoritmo de roteamento original.

O método de otimização baseado no comportamento coletivo de formigas [3][4][5] utiliza um conjunto de agentes móveis que percorrem o ambiente em busca das melhores soluções, simulando a busca de alimento realizada por colônias de formigas. O primeiro algoritmo de roteamento baseado neste método foi aplicado a uma rede de comutação de circuitos e denominado Controle Baseado em Formigas (ABC – *Ant-Based Control*) [10]. Neste sistema, os agentes percorrem a rede buscando os caminhos de menor comprimento e menos carregados. As variáveis armazenadas nas tabelas de roteamento dos nós representam a probabilidade $P_i(j,d)$ que um agente que se encontra no nó i seja enviado ao nó destino d através de seu nó vizinho j. A cada nó visitado, o agente atualiza o valor da probabilidade relativa ao movimento no sentido contrário a sua trajetória. Assim, um agente lançado do nó o que se encontra no nó j vindo de seu vizinho i, atualiza o valor de $P_j(i,o)$, baseado no tempo decorrido desde o seu lançamento, denominado idade do agente. Esta variável é constituída pelo número de nós percorridos pelo agente somado a um atraso sofrido em cada um deles, relativo a capacidade de processamento ociosa do nó no momento da visita. As probabilidades de escolha dos outros vizinhos do nó j com relação a origem o são normalizadas. As chamadas são enviadas sobre a rede de forma simultânea aos agentes e suas rotas são selecionadas deterministicamente. Assim, o nó vizinho que apresenta maior valor de probabilidade é escolhido.

O desempenho do sistema ABC foi comparado ao de algoritmos estáticos baseados no menor caminho e algoritmos adaptativos baseados em agentes móveis utilizados pela empresa britânica BT (*British Telecom*) [10]. A medida de desempenho considerada foi a porcentagem média de chamadas perdidas pelos algoritmos, situação que ocorre quando um nó selecionado na rota das chamadas não possui capacidade de processamento disponível. O algoritmo ABC obteve menores perdas que os demais sob variações nos padrões de tráfego da rede.

Os bons resultados produzidos conduziram ao desenvolvimento do algoritmo ABC Esperto [1], que aplica ao sistema original um princípio de programação dinâmica [12]. Os agentes deste algoritmo atualizam a cada nó percorrido as probabilidades referentes a escolha do último nó visitado com relação a todos os nós intermediários e não somente ao nó origem. O reforço associado a cada nó visitado é dependente da idade relativa do agente no momento da visita. Deste modo, o algoritmo ABC Esperto obteve melhores resultados que o sistema ABC sob variações nos padrões de tráfego da rede.

3. AGENTES-Q

O algoritmo Agentes-Q combina os melhores aspectos de dois dos algoritmos descritos acima: Roteamento-Q com Reforço Dual [8] e Controle Baseado em Formigas (ABC) [10]. Este sistema foi aplicado a uma rede de comutação de circuitos, sendo composto por um conjunto de agentes que se movimentam

sobre a rede à procura dos caminhos menores e menos carregados. Os Agentes-Q adicionaram aos algoritmos que inspiraram sua criação um mecanismo de reforço negativo na atualização dos valores-Q e a combinação destas variáveis com a situação atual dos nós na seleção das rotas percorridas pelas chamadas. Através destes mecanismos, o algoritmo proposto aperfeiçoa a estratégia utilizada pelos demais sistemas para realizar continuamente o balanceamento de carga sobre a rede. Isto é realizado através da forte redução da chance de um nó ser utilizado na rota de uma chamada, caso ele se encontre altamente carregado no momento da escolha do caminho, mesmo que ele constitua a menor rota. Desta forma, o sistema se torna mais flexível e robusto às mudanças na rede.

Como no algoritmo ABC, um agente é lançado a partir de cada nó da rede em direção a um destino aleatório a cada passo de tempo. Durante seu movimento ao longo da rede, o agente armazena seus nós origem e destino e o tempo decorrido desde o seu lançamento, denominado idade do agente. O próximo nó *j* a ser visitado é selecionado de acordo com os valores de probabilidade armazenados na tabela do nó *i* em que ele se encontra, representados por valores-Q. Os agentes possuem um mecanismo de exploração, que constitui uma pequena probabilidade (5%) de escolha aleatória de um nó. Ao se mover para o nó *j* escolhido, o agente é atrasado de acordo com a capacidade de processamento ociosa deste nó (O_j):

$$atraso_j = 80\, e^{-0.075\, O_j}\ ,\ 0 < O_j < 100 \qquad (1)$$

O agente recebe da rede dois sinais de reforço relativos ao movimento entre os nós *i* e *j*. O primeiro deles (r_i) é função da carga do nó *j* selecionado e o segundo sinal (r_j) é função da idade do agente.

$$r_i = \left(\frac{\sigma}{C_j + \delta} \right), 0 < r_i < 1 \qquad (2)$$

$$r_j = \left(\frac{\beta}{idade\ do\ agente} + \rho \right), 0 < r_j < 1 \qquad (3)$$

onde: C_j = carga do nó *j*, $C_j = 100 - O_j$; e σ, δ, β, ρ = constantes de proporcionalidade obtidas empiricamente.

Após sofrer o atraso, o agente atualiza os seguintes valores-Q: (i) o valor armazenado na tabela de roteamento do nó *i* em direção ao destino *d* e (ii) o valor armazenado na tabela de roteamento do nó *j* em direção a origem *o*.

Atualização na direção do movimento do agente:

$$Q_i(j,d) = \frac{Q_i(j,d) + \Delta Q_i(j,d)}{\sum_v Q_i(v,d)} \qquad (4)$$

Atualização na direção contrária ao movimento do agente:

$$Q_j(i,o) = \frac{Q_j(i,o) + \Delta Q_j(i,o)}{\sum_z Q_j(z,o)} \qquad (5)$$

onde: $\Delta Q_i(j,d) = \alpha_i [r_i + \gamma_i \max_z Q_j(z,d) - Q_i(j,d)]$ (6)

$$\Delta Q_j(i,o) = \alpha_j [r_j + \gamma_j \max_v Q_i(v,o) - Q_j(i,o)] \qquad (7)$$

α_i, α_j = taxas de aprendizado; γ_i, γ_j = fatores de desconto; *v* nó vizinho de *i*; *z* é nó vizinho de *j*; $\sum_v Q_i(v,d)$ e $\sum_z Q_j(z,o)$ são fatores de normalização.

Quando o número de chamadas processadas pelo nó *j* atinge um dado valor determinado empiricamente, neste caso 60% de sua capacidade máxima de processamento, o valor $Q_i(j,d)$ é decrementado por ΔQ, ao invés de ser acrescido. Este reforço negativo é utilizado por algoritmos de aprendizagem por reforço para penalizar o agente quando ele realiza uma ação não satisfatória [12]. Esta penalização objetiva aprimorar a distribuição das chamadas sobre os nós da rede, decrescendo a probabilidade de um nó ser utilizado se o agente que o escolheu percorreu uma rota longa e/ou congestionada.

Quando o agente atinge seu destino, ele é removido do sistema.

As chamadas são introduzidas na rede após um período de tempo denominado inicialização, onde existem somente agentes na rede, como no sistema ABC. Antes de enviar uma chamada, o algoritmo seleciona toda a rota a ser por ela percorrida, levando em consideração os valores-Q dos nós candidatos e sua capacidade ociosa atual, combinando estas informações em uma variável denominada avaliação. O sistema proposto seleciona de modo sequencial e determinístico os nós vizinhos que apresentam maior valor de avaliação, onde a avaliação do nó vizinho *j* de *i* com relação ao destino *d*, $A_i(j,d)$, é calculada da seguinte forma:

$$A_i(j,d) = (1-\omega) Q_i(j,d) + \omega O_j \qquad (8)$$

onde ω é o fator de peso, $0 < \omega < 1$.

Se M_j decresce durante os últimos 50 passos de tempo:

$$\omega = \omega_1 = \frac{1}{1 + \exp[(-k_1 M_j) + k_2]} \qquad (9)$$

Se M_j sofre um acréscimo durante os últimos 50 passos de tempo:

$$\omega = \omega_2 = \frac{1}{1 + \exp[(-k_3 M_j) + k_4]} \qquad (10)$$

M_j é a carga média do nó j; k_1, k_2, k_3, k_4 são constantes de proporcionalidade empiricamente obtidas.

As Eqs. (9) e (10) constituem uma curva de histerese da carga média do nó candidato M_j versus o fator de peso de sua capacidade ociosa ω considerado na sua avaliação. Deste modo, inicialmente na Eq. (8) $\omega \approx 0$ e o nó vizinho que possui maior valor-Q é selecionado, constituindo o menor caminho. À medida que estes caminhos vão sendo utilizados, eles se tornam altamente carregados e propensos ao congestionamento. Assim,

quando a carga do nó atinge um dado nível empírico (93%), ω sofre um rápido acréscimo e a capacidade ociosa deste nó possuirá maior influência na sua escolha que o seu valor-Q. Deste modo, os nós altamente carregados não são selecionados por um período de tempo, quando então rotas alternativas são utilizadas. Após um intervalo de tempo, quando a carga dos nós que constituem os menores caminhos sofre um decréscimo e atinge um dado nível empírico (87%), ω decai rapidamente e estes nós são novamente selecionados. Esta estratégia procura diminuir as discrepâncias entre os valores-Q armazenados na tabela de roteamento dos nós e o seu estado real, decorrente das variações inerentes a rede. Assim, o sistema procura maximizar o uso dos nós que constituem os menores caminhos e simultaneamente evitar seu congestionamento, ou reduzir a duração do mesmo, sob quaisquer situações.

4. EXPERIMENTOS

Os algoritmos Agentes-Q, ABC e ABC Esperto foram aplicados a um modelo da rede de telefonia da empresa AT&T (Fig. 1), com n = 87 nós. Cada nó i possui número de identificação, capacidade máxima de processamento P_i de 40 chamadas, capacidade ociosa O_i e conjunto de vizinhos N_i. Cada nó i possui ainda uma probabilidade de ser um nó terminal de uma chamada, denominada probabilidade de chamada e uma tabela de roteamento com n colunas e linhas definidas por N_i. Cada conexão da rede possui capacidade de processamento ilimitada.

Os parâmetros usados pelo algoritmo proposto foram obtidos empiricamente. Durante o período de inicialização, eles foram estabelecidos como: $\alpha_i = 0.1$, $r_i = 0.005$, $\gamma_i = 1.0$, $\alpha_j = 0.1$, $\beta = 6.0$, $\rho = 0.01$ e $\gamma_j = 1.0$. Após este período: $\alpha_i = 0.01$, $\sigma = 0.05$, $\delta = 0.1$, $\gamma_i = 0.9$, $\alpha_j = 0.06$, , $\beta = 70.0$, $\rho = 0.001$ e $\gamma_j = 0.8$.

Em cada experimento, foram gerados dez blocos de chamadas, onde os nós origem e destino foram selecionados de acordo com uma distribuição de probabilidades aleatória em [0.01, 0.07], sendo normalizadas após sua geração. A cada passo de tempo, 1 chamada em média foi gerada seguindo uma distribuição de Poisson, com duração média de 170 passos de tempo, de acordo com uma distribuição exponencial. A simulação foi dividida em três períodos: inicialização, adaptação (A) e teste (B), com duração de 500, 7500 e 7500 passos de tempo, respectivamente.

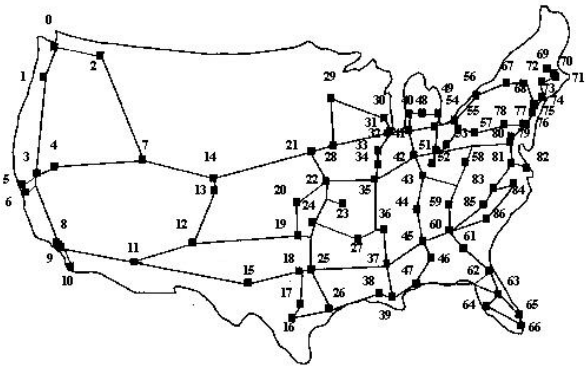

Figura 1. Modelo da rede de telefonia da AT&T.

Neste artigo serão apresentados 8 experimentos realizados envolvendo variações nos padrões de tráfego, nível de carga e topologia da rede, além da simulação de falha temporária nos roteadores, através da introdução de uma probabilidade de escolha aleatória do nó a ser utilizado por uma chamada. No primeiro teste, as probabilidades de chamadas utilizadas no período de adaptação e teste pertenceram ao mesmo conjunto. Assim, os algoritmos utilizaram o subconjunto de probabilidades 1A no período de adaptação e 1B no período de teste, então 2A e 2B e assim sucessivamente, até os subconjuntos 10A e 10B. No segundo teste estes subconjuntos foram modificados, de forma que o período de adaptação envolveu a utilização do subconjunto 1A e o período de teste envolveu a de 2B, até a utilização de 10A e 1B. No terceiro e quarto experimentos, o nível de carga da rede foi modificado para médio e alto, com geração de 1.5 e 2 chamadas, respectivamente, entre os passos de tempo 8000 e 9000. No quinto teste, um nó foi desconectado da rede entre os passos de tempo 8000 e 9000 e no sexto, esta situação ocorreu por todo o período de teste. O nó 60 foi selecionado, devido a sua localização e constante utilização no roteamento. No sétimo e oitavo testes, foi introduzida uma probabilidade de escolha aleatória do nó utilizado pela chamada de 1% e 5%, respectivamente, entre os passos 8000 e 9000.

5. RESULTADOS

Os algoritmos foram analisados pela porcentagem média de chamadas perdidas a cada intervalo de 500 passos de tempo sobre os 10 blocos de chamadas gerados. Os resultados foram submetidos a um teste de hipótese [9], com uma distribuição t de *Student* com nível de significância de 0.05, para verficar diferenças significativas entre os sistemas.

No primeiro teste (Fig. 2), os Agentes-Q produziram inicialmente perdas decrescentes, as quais se estabilizaram após um intervalo de tempo. Os demais agentes apresentaram um acréscimo inicial nas perdas, que decresceram após um período de tempo.

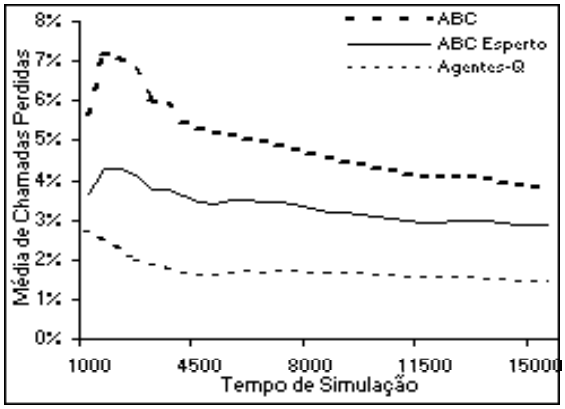

Figura 2. Resultados sob parâmetros invariáveis.

Sob a variação dos padrões de tráfego (Fig. 3), os algoritmos confirmaram as tendências observadas no primeiro teste.

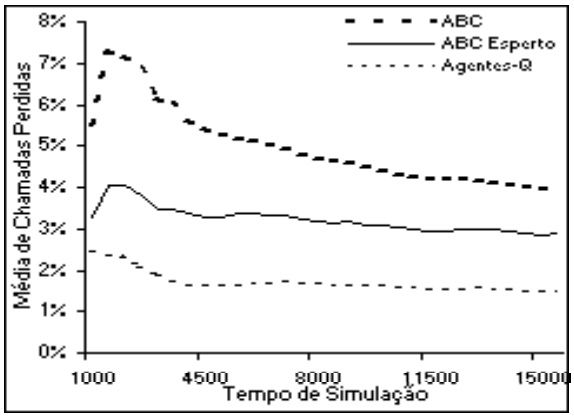

Figura 3. Resultados sob padrões de tráfego variáveis.

As perdas sofreram um grande acréscimo sob variações no nível de carga, superior para o nível alto (Fig. 4) em relação ao nível médio (Fig. 5). Quando a carga foi retornada ao nível baixo, os algoritmos se adaptaram e produziram perdas decrescentes.

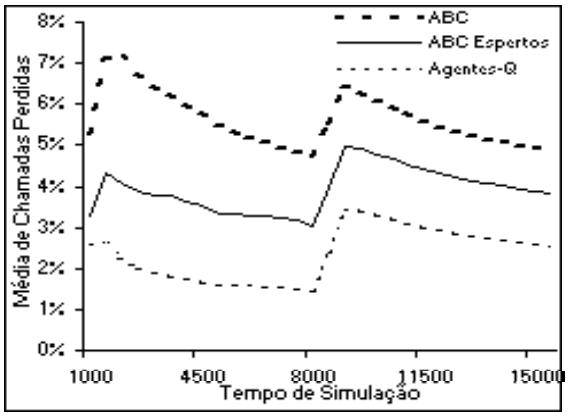

Figura 4. Resultados sob nível de carga médio temporário (entre passos 8000 e 9000).

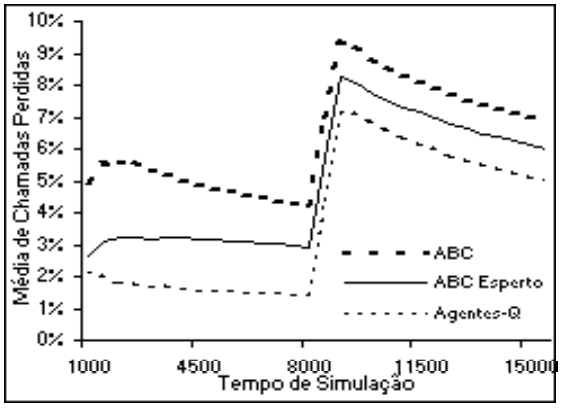

Figura 5. Resultados sob nível de carga alto temporário (entre passos 8000 e 9000).

A desconexão temporária do nó 60 (Fig. 6) provocou perdas crescentes nos três algoritmos, em especial no sistema Agentes-Q. Após a reconexão do nó selecionado, todos os algoritmos obtiveram decréscimo nas perdas, de forma mais acentuada no caso do algoritmo proposto, indicando a adequação dos algoritmos a nova condição da rede.

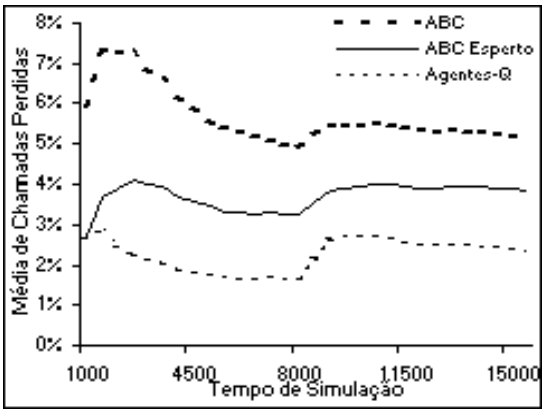

Figura 6. Resultados sob desconexão temporária do nó 60 (entre passos 8000 e 9000).

Quando o nó 60 foi retirado da rede durante todo o período de teste (Fig. 7), todos os algoritmos apresentaram acréscimo nas perdas, porém este foi bem menos acentuado no caso do algoritmo proposto. Estes resultados podem estar relacionados tanto com o desempenho dos algoritmos como com a importância do nó selecionado para o sistema de roteamento.

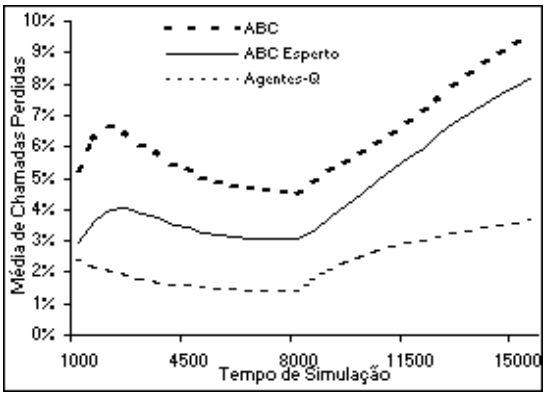

Figura 7. Resultados sob desconexão permanente do nó 60.

A simulação de uma falha nos roteadores por um período de tempo provocou um pequeno acréscimo no volume de chamadas perdidas pelos três algoritmos, tendo sido maior para um ruído de 10% (Fig. 9) do que para 5% (Fig. 8). Após a retirada deste fator, os três algoritmos obtiveram perdas decrescentes. O algoritmo proposto obteve menores perdas que os demais sob os dois valores de ruído considerados, sugerindo melhor adaptação a esta situação da rede.

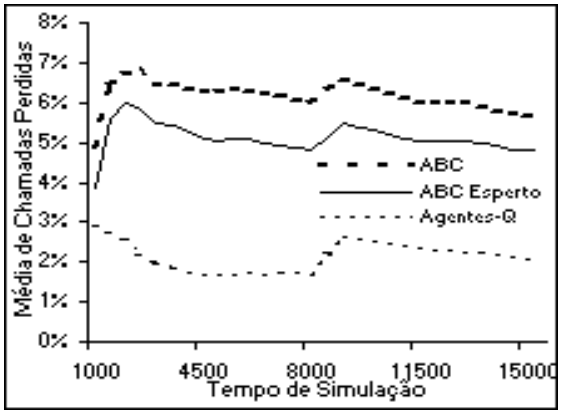

Figura 8. Resultados sob um fator de ruído temporário de 5% (passos 8000 e 9000).

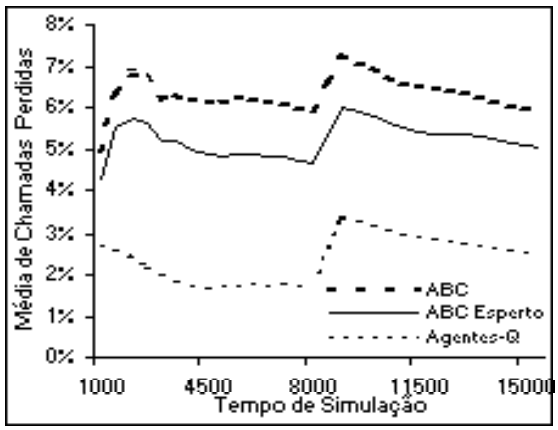

Figura 9. Resultados sob um fator de ruído temporário de 10% (passos 8000 e 9000).

Em todas as situações consideradas, o algoritmo proposto se adaptou de forma mais eficiente as condições da rede, obtendo menores perdas que seus competidores.

6. DISCUSSÃO

Este artigo apresentou um algoritmo de roteamento distribuído e adaptativo, denominado Agentes-Q, baseado em três técnicas: aprendizagem-Q, aprendizagem por reforço dual e otimização baseada no comportamento coletivo de formigas. Isto permitiu ao sistema utilizar um conjunto de agentes autônomos que aprendem continuamente as melhores rotas baseados somente em sua experiência e em informação local sobre o estado da rede, a qual é atualizada em duas direções do ambiente a cada movimento do agente. A combinação de informação presente na tabela de roteamento com o estado atual dos nós, aliada ao reforço negativo na atualização desta informação, possibilitaram que os Agentes-Q aprimorassem o balanceamento de carga sobre a rede sob diversas situações em relação aos seus competidores, produzindo menores perdas sob todos os testes realizados.

Entretanto, os resultados obtidos sugerem que a velocidade de adaptação do algoritmo proposto necessita ser aprimorada. Desta forma, a pesquisa futura envolverá o estudo de novas versões das estratégias utilizadas. Uma outra questão de interesse é a consideração de um número maior de restrições de operação da rede, além da capacidade de processamento dos nós, a fim de aproximar as condições de simulação do problema real de roteamento. Do mesmo modo, seria interessante aplicar o algoritmo proposto a novas topologias de rede e considerar novas variações nos seus parâmetros.

7. REFERÊNCIAS

[1] Bonabeau E., Henaux F., Guérin S., Snyers D., Kunts P. and Théraulaz G. "Routing in telecommunications networks with 'smart' ant-like agents telecommunication applications". *Lectures Notes in AI*, 1437, Springer Verlag, 1998.

[2] Boyan J. A and Littman M. L. "Packet routing in dynamically changing networks: a reinforcement learning approach". *Advance in Neural Information Processing Systems* 6: 671-678. Morgan Kaufmann, San Mateo, CA, 1994.

[3] Colorni A, Dorigo M. and Maniezzo, V. "Distributed optimization by ant colonies". *Proceedings of the European Conference on Artificial Life (ECAL 91)*, Elsevier, 1991, pages 134-142.

[4] Dorigo M. "Optimization, learning and natural algorithms" (in Italian). *Ph.D. thesis*, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy, 1992.

[5] Dorigo M., Maniezzo V. and Colorni A. "The ant system: optimization by a colony of cooperating agents". *IEEE Transactions on Systems, Man and Cybernetics B*, 26:1-13, 1996.

[6] Goetz P., Kumar S. and Miikkulainen R. "On-line adaptation of signal predistorter through dual reinforcement learning." *Proceedings of the Machine Learning: 13th Annual Conference*, 1996.

[7] Kaelbling L. P., Littman M. L. and Moore A. W. "Reinforcement learning: a survey". *Journal of Artificial Intelligence Research*, 4:237-285, 1996.

[8] Kumar S. and Miikkulainen R. "Dual reinforcement Q-routing: an on-line adaptive routing algorithm". *In Proceedings of the Artificial Neural Networks in Engineering*, 1997.

[9] Montgmorey D. C. *Design and Analysis of Experiments*. John Wiley, New York, 1997.

[10] Schoonderwoerd R., Holland O., Bruten J. and Rothkrantz L. "Ant-based load balancing in telecommunications networks". *Adaptive Behavior*, 5(2): 169-207, 1996.

[11] Steenstrup M. E. *Routing in Telecommunications Networks*. Prentice-Hall, 1995.

[12] Sutton R. S. and Barto A. G. *Reinforcement learning: An introduction*. MIT Press, Cambridge, MA 1998.

[13] Watkins C. J. C. H. "Models of delayed reinforcement learning". *Ph.D. thesis*, Psychology Department, Cambridge University, 1989.

[14] Watkins C. J. C. H. and Dayan P. "Q-learning". *Machine Learning*, 8(3): 279-292, 1992.