Gerenciando Múltiplas Falhas Bizantinas em Redes Ópticas Roteadas por Algoritmos baseados em Otimização por Colônia de Formigas

Aldo Ventura da Silva[1], **Gustavo Sousa Pavani**[1]

[1]Universidade Federal do ABC (UFABC)
Av. dos Estados, 5001. Santo André – SP – Brasil. CEP: 09210-580.

{aldo.ventura,gustavo.pavani}@ufabc.edu.br

Abstract. *Routing algorithms based on Ant Colony Optimization (ACO) are especially vulnerable to byzantine failures, where authenticated nodes behave in an arbitrary way and disrupt the routing on the network. In this work, we analyze the use of crankback re-routing extensions associated to the ACO algorithm to tackle byzantine failures that affect many nodes in a wavelength-switched optical network. For the three types of byzantine failures assessed, misdirection of forward ants, dropping of forward ants and dropping of backward ants, simulations demonstrate that the crankback mechanism makes the network resilient to byzantine failures, mitigating the impact on the blocking probability for establishing lightpaths.*

Resumo. *Algoritmos de roteamento baseados em Otimização por Colônia de Formigas (ACO) são especialmente vulneráveis a falhas bizantinas, em que nós autenticados se comportam de maneira arbitrária e comprometem o roteamento da rede. Neste trabalho, analisamos o uso de extensões de rerroteamento* crankback *associados ao algoritmo ACO para gerenciar falhas bizantinas que afetam muitos nós de uma rede óptica comutada por comprimento de onda. Para os três tipos de falha bizantina avaliados, desorientação de formigas no caminho de ida, descarte de formigas no caminho de ida e descarte de formigas no caminho de volta, as simulações demonstram que o mecanismo de* crankback *torna a rede mais resiliente a falhas bizantinas, mitigando o impacto sobre probabilidade de bloqueio no estabelecimento de caminho ópticos.*

1. Introdução

Algoritmos de Otimização por Colônia de Formigas (*Ant Colony Optimization* – ACO) foram aplicados com sucesso em um grande número de problemas difíceis em redes de comunicação [Dorigo and Stützle 2004]. Tais algoritmos foram inspirados no processo de coleta de alimentos das formigas naturais, que é caracterizado por um tipo de comunicação indireta baseada no ambiente para estimular ações subsequentes chamada de estigmergia [Grassé 1959].

A estigmergia é um componente essencial dos algoritmos ACO, sendo responsável pelo comportamento emergente e auto-organizado da colônia de formigas. Com efeito, as formigas artificiais depositam informação nos nós visitados sobre o desempenho do caminho atravessado na rede. Os níveis de feromônio podem ser sentidos localmente pelas formigas, o que reforça boas rotas previamente descobertas.

Algoritmos ACO podem ser usados no plano de controle de redes ópticas, substituindo o protocolo de roteamento enquanto mantém o protocolo de sinalização inalterado [Pavani and Waldman 2010]. O processo de descoberta de boas rotas é totalmente distribuído e baseado em informações locais, sendo resiliente a falhas de enlace ou de nó [Pavani and Waldman 2006b, Pavani and Waldman 2008, Pavani and Waldman 2010].

Entretanto, algoritmos ACO são vulneráveis a um ambiente hostil, no qual um atacante pode explorar os problemas de segurança encontrados em algoritmos baseados em estigmergia [Zhong and Evans 2002]. Por exemplo, não há nenhum mecanismo que garanta a integridade ou a autenticidade da informação carregada pelas formigas. Além disso, existe uma confiança implícita entre os nós da rede [Zhong and Evans 2002].

Embora técnicas de segurança computacional possam ser usadas para mitigar esses problemas e prover um roteamento seguro, tais técnicas não podem evitar os problemas relacionados a um nó autenticado que foi comprometido ou mal configurado [Awerbuch et al. 2002, Awerbuch et al. 2003, Wendlandt et al. 2006].

Assim, uma falha bizantina no protocolo de roteamento acontece quando um nó autenticado exibe um comportamento arbitrário ou com falha que degrada ou mesmo interrompe o serviço de roteamento da rede [Awerbuch et al. 2002, Perlman 1988]. Nesse sentido, uma falha bizantina pode ser considerada como uma vulnerabilidade de segurança do protocolo de roteamento baseado em ACO.

A proposta deste trabalho é gerenciar múltiplas falhas bizantinas em redes ópticas comutadas por comprimento de onda e roteadas por algoritmos baseados em ACO. Através do uso de extensões de rerroteamento *crankback* associadas ao algoritmo ACO [Farrel et al. 2007, Pavani and Waldman 2010], que permitem o cálculo de rotas mesmo na presença de informação de roteamento inexata e/ou desatualizada, demonstramos que é possível mitigar o problema de falhas bizantinas, mesmo que muitos nós na rede sejam afetados por essas falhas.

Com efeito, este trabalho demonstra a robustez do mecanismo de *crankback* para se alcançar a capacidade de sobrevivência de roteamento mesmo com muitos nós exibindo um comportamento bizantino e, consequentemente, afetando o funcionamento do algoritmo ACO, o que era uma importante lacuna de [Pavani et al. 2016]. Ressalta-se que tal grau de robustez do algoritmo de roteamento a falhas bizantinas é raramente tratado na literatura.

O restante deste trabalho está organizado da seguinte forma. A Seção 2 introduz o roteamento baseado em ACO para redes ópticas e o mecanismo de *crankback*. Na Seção 3, é apresentado o modelo de falhas bizantinas no roteamento que é considerado neste trabalho. As simulações utilizadas para avaliar o roteamento ACO em cenários com múltiplas falhas bizantinas são apresentadas na Seção 4. Os resultados obtidos nas simulações são mostrados e discutidos na Seção 5. Por fim, as conclusões são apresentadas na Seção 6.

2. Roteamento baseado em Otimização por Colônia de Formigas

A observação do processo de busca de alimento das formigas é fonte de inspiração para a classe de algoritmos denominada de Otimização por Colônia de Formigas. Algoritmos ACO são baseados em estigmergia [Grassé 1959] artificial, no qual os níveis de feromônios artificiais têm uma realimentação positiva ou negativa de acordo com a quali-

dade da solução vista pelas formigas. Como tais níveis de feromônio contêm informação de soluções prévias do problema, eles podem ser explorados de forma coletiva pelas formigas para melhorar a solução. Embora as formigas sejam agentes muito limitados, a estigmergia permite que a colônia de formigas apresente um comportamento emergente e auto-organizado [Prehofer and Bettstetter 2005].

AntNet [Di Caro and Dorigo 1998] é um algoritmo que pertence a classe ACO, cuja principal aplicação é o roteamento em redes de telecomunicações. O algoritmo AntNet original, que foi projetado para redes comutadas por pacote, foi posteriormente adaptado para ser usado em redes ópticas, como a tecnologia de comutação de pacotes ópticos (*Optical Packet Switching* – OPS) [Pavani and Waldman 2006b] e a comutação de circuitos [Pavani and Waldman 2006a, Pavani and Waldman 2010]. Em vez de usar o atraso introduzido em cada salto como a métrica para roteamento como em [Di Caro and Dorigo 1998], essas adaptações consideram o número de saltos armazenados na memória das formigas. Assim, cada formiga reúne e carrega a identificação de cada nó visitado por ela.

As formigas podem ser implementadas como datagramas convencionais no plano de controle da rede. Elas podem ser vistas como mensagens de controle trocadas pelos nós para coletar o estado atual da rede óptica.

Além das formigas artificiais e sua memória, duas estruturas de dados devem ser mantidas em cada nó: uma tabela de roteamento de feromônio e um modelo estatístico parametrizado. A tabela de roteamento de feromônio é uma matriz que representa o nível de feromônio para um determinado destino e um nó vizinho. O nível de feromônio estima a probabilidade de se atingir um destino através de um nó vizinho como próximo salto. Por sua vez, o modelo estatístico parametrizado mantém estimativas de distância para todos os outros nós da rede, que são calculadas a partir dos caminhos seguidos pelas formigas.

Quando uma formiga vai de um nó origem até um nó destino, ela é denominada *forward ant*, selecionando o próximo salto por uma regra probabilística. Essa regra considera os níveis de feromônio, dado o destino, e informações locais, que consideram a disponibilidade dos comprimentos de onda do enlace conectado ao próximo salto [Pavani and Waldman 2010].

Quando a formiga chega ao seu destino, ela se torna uma *backward ant* e retorna ao nó de origem usando o mesmo caminho que seguiu no caminho de ida. Nesse percurso de volta, a formiga atualiza o modelo estatístico parametrizado e a tabela de roteamento de feromônio para todos os nós referentes ao caminho percorrido e seus subcaminhos estatisticamente relevantes [Di Caro and Dorigo 1998].

Se um nó vizinho estiver no caminho percorrido pela formiga, este receberá um reforço positivo no seu valor de feromônio, que está relacionado a qualidade do caminho encontrado pela formiga. Por outro lado, os outros nós vizinhos receberão um reforço negativo no seu valor de feromônio. Além disso, um mecanismo para evitar a estagnação dos valores do feromônio é empregado, limitando o valor máximo do feromônio para um nó vizinho [Di Caro and Dorigo 1998].

Devido ao emprego de métrica de congestionamento na informação local usada para rotear as formigas no caminho de ida, que é depois refletida nos níveis de feromônios

depositados pelas formigas no retorno, a rede pode usar caminhos mais longos, porém menos congestionados do que os caminhos mais curtos. Assim, ocorre a modificação da representação do caminho percorrido, buscando-se, através de soluções anteriores, a otimização das novas soluções. Como resultado, a rede é capaz de balancear a carga [Pavani and Waldman 2010], o que ajuda a diminuir a probabilidade de bloqueio em toda a rede.

2.1. Estabelecimento de um caminho óptico

A adaptação da AntNet a redes ópticas comutadas por comprimento de onda considera o modelo integrado do padrão *Generalized Multi Protocol Label Switching* (GMPLS) [Mannie 2004] como base da arquitetura de plano de controle.

Os valores locais de feromônio podem ser usados pelas mensagens RSVP-TE [Berger 2003] para estabelecer um caminho óptico como um caminho comutado por rótulo (*Label Switched Path* – LSP) a cada salto. Portanto, a AntNet pode atuar como um protocolo de roteamento, substituindo os algoritmos de estado de enlace comumente encontrados em redes ópticas [Pavani and Waldman 2010].

A maneira mais simples de escolher o próximo salto de um caminho de mensagem RSVP-TE para estabelecimento de um LSP é selecionar o nó vizinho com o nível mais alto de feromônio para um determinado destino [Pavani and Waldman 2006a]. A mensagem `Path` reúne os rótulos disponíveis (comprimentos de onda) dos enlaces percorridos no objeto `Label Set` e, no nó de destino, é escolhido o primeiro rótulo disponível em todos os enlaces *(first-fit)*. Se não houver um rótulo que satisfaça a restrição de continuidade do comprimento de onda ou a mensagem `Path` não possa alcançar o destino devido a um laço, a configuração do LSP é bloqueada e a mensagem RSVP-TE de erro (`PathErr`) apropriada é gerada para informar o problema ao nó origem [Pavani and Waldman 2006a].

Esta abordagem heurística simples e gulosa de roteamento pode ser melhorada com o uso de extensões de *crankback* [Farrel et al. 2007, Pavani and Waldman 2010], no qual um nó pode reeditar o estabelecimento de um LSP para contornar um recurso bloqueado. Nesse caso, um nó pode escolher o nó vizinho com o segundo nível mais alto de feromônio como um próximo salto. O objetivo é desviar de um enlace que fará com que o estabelecimento do LSP falhe, devido a uma violação da restrição de continuidade de comprimento de onda, ou para evitar um laço.

Em um ambiente de roteamento distribuído, como em um típico nas redes ópticas comutadas por comprimento de onda, as informações de estado de recursos anunciadas pelo protocolo de roteamento (ACO) podem estar desatualizadas ou imprecisas [Farrel et al. 2007]. Ao retornar informações sobre uma falha no estabelecimento de um LSP, o mecanismo de *crankback* permite que um grande número de caminhos alternativos sejam pesquisados, enquanto que o esquema guloso apresentado em [Pavani and Waldman 2006a] permite que apenas um único caminho seja usado para se estabelecer um LSP.

O número de tentativas de rerroteamento em cada nó deve ser limitado [Farrel et al. 2007] para evitar, por exemplo, caminhos que são muito longos, ou seja, caminhos que podem usar muitos recursos da rede. A tabela histórica é uma estrutura de dados local para o nó, que é usada para identificar os vizinhos já selecionados para

estabelecer um determinado LSP. Esta seleção segue uma ordem decrescente de valor de feromônio para um determinado destino. Se essas entradas estão totalmente ocupadas, o nó não pode mais ser parte do caminho desse LSP, reduzindo o espaço de busca de soluções de caminhos candidatos. Portanto, o número de tentativas locais de rerroteamento em cada nó para um determinado LSP é limitado pelo número de entradas na tabela histórica [Pavani and Waldman 2010], que é um parâmetro do mecanismo de *crankback*.

Quando o número máximo de tentativas de rerroteamento local é atingido, o nó envia uma mensagem `PathErr` para o nó anterior pertencente à rota já percorrida. O nó que recebe a mensagem encerrará a resposta de erro e reeditará o processo de estabelecimento de LSP por meio de uma mensagem `Path` para encontrar uma rota alternativa, desde que o seu limite local de tentativas não foi excedido. Caso contrário, a mensagem `PathErr` é encaminhada ao nó anterior da rota percorrida. Finalmente, se a mensagem `PathErr` atinge o nó de origem e o seu número de tentativas foi excedido, então o estabelecimento do LSP não pode ser realizado e a requisição será declarada bloqueada [Pavani and Waldman 2010].

Observe que o mecanismo *crankback* será invocado pelo plano de controle durante o processo de descoberta de uma rota factível entre os nós origem e destino sempre que o estabelecimento do LSP falhar, desde que seja permitido novas tentativas de rerroteamento para esse LSP. Maiores detalhes sobre o mecanismo de *crankback* podem ser encontrados em [Pavani and Waldman 2010, Pavani et al. 2016].

3. Modelo de Falhas Bizantinas no Roteamento

Uma rede pode atingir diversos níveis de robustez na presença de falhas [Perlman 1988]. Considera-se que uma rede possui robustez simples quando lida com falhas que tornem inoperantes enlaces ou nós. Lidar com falhas simples é considerado comum na operação de redes e os algoritmos ACO demonstram possuir robustez simples em redes ópticas [Pavani and Waldman 2006b, Pavani and Waldman 2008, Pavani and Waldman 2010].

Diz-se que uma rede possui robustez bizantina quando se comporta de forma correta mesmo possuindo nós com falhas bizantinas. Para se verificar a robustez bizantina do algoritmo AntNet na presença de múltiplas falhas, foram considerados os seguintes tipos de falhas bizantinas, os quais não podem ser tratados por mecanismos de integridade ou de autenticidade [Pavani et al. 2016]:

1 ***Misdirecting Forward (MF) ants***: As formigas no caminho de ida sofrem desorientação e não seguem o roteamento estocástico determinado pelos valores em sua tabela de feromônio e informações de congestionamento local. No nosso caso, consideramos que as formigas são encaminhadas para o nó vizinho com o menor valor de endereço, evitando o retorno ao nó anterior. Um exemplo desta falha pode ser visto na Figura 1.
2 ***Dropping Backward (DB) ants***: O nó afetado descarta as formigas no caminho de volta, exceto quando o nó afetado é o nó de origem da formiga. Tal falha bizantina está exemplificada na Figura 2.
3 ***Dropping Forward (DF) ants***: As formigas no caminho de ida são descartadas pelo nó que se comporta de forma arbitrária, exceto quando o nó afetado é o nó de destino da formiga. Os nós que descartam os pacotes seletivamente também são

referidos como buracos negros [Awerbuch et al. 2002]. Este falha é ilustrada na Figura 3.

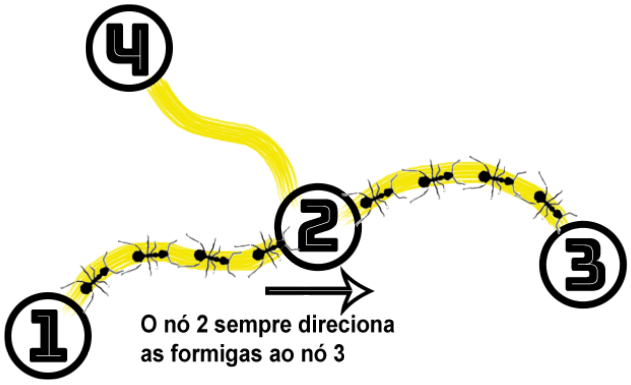


Figura 1. Exemplo de falha de desorientação das formigas no caminho de ida. O nó 2 sempre enviará formigas para o nó 3, independentemente dos níveis de feromônio da tabela de roteamento.

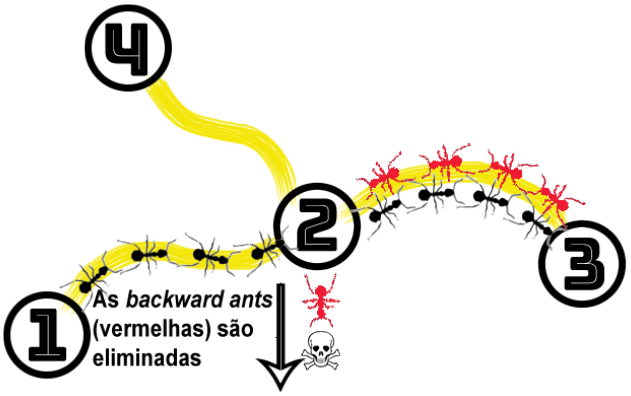


Figura 2. Exemplo de falha de eliminação de formigas no caminho de volta. As formigas atravessaram os nós 1, 2 e 3 no caminho de ida. Ao retornar pelo caminho de volta, elas (formigas vermelhas) são eliminadas pelo nó 2, exceto as que possuem como destino o próprio nó 2.

Percebe-se que todas as falhas bizantinas consideradas afetam o mecanismo de realimentação envolvido no reforço de feromônios do algoritmo AntNet. No primeiro caso, as formigas são redirecionadas para um vizinho fixo, o que tende a aumentar os níveis de feromônio desse determinado caminho e, consequentemente, a chance de que esse vizinho se torne o próximo salto no estabelecimento de um caminho óptico (LSP). Esta situação pode levar a um aumento do congestionamento nesse ponto da rede.

Nos casos em que as formigas são descartadas, há uma perturbação do processo de atualização das estruturas de dados feitas pelas formigas no caminho de volta, o que diminui a probabilidade do nó afetado pela falha bizantina ser considerado como próximo salto para estabelecimento de caminho óptico. O descarte de formigas gera um impacto maior que a simples desorientação das formigas. Uma falha DB apresenta um menor impacto no processo de atualização que uma falha DF, já que pelo menos parte do caminho percorrido pela formiga é atualizado [Pavani et al. 2016].

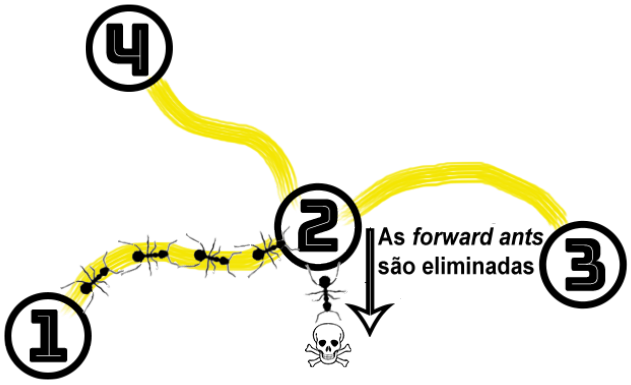


Figura 3. Exemplo de falha de eliminação de formigas no caminho de ida. As formigas visitam os nós 1 e 2. O nó 2 elimina todas as formigas no caminho de ida, exceto as que possuem como destino o próprio nó 2.

Portanto, os níveis de feromônio encontrados nas redes com falhas bizantinas podem não refletir as capacidades de balanceamento de carga comumente encontradas em algoritmos AntNet, o que acaba gerando um aumento na probabilidade de bloqueio da rede.

Vale ressaltar que essas falhas bizantinas são difíceis de detectar, uma vez que são quase indistinguíveis da operação normal do nó. Além disso, uma vez que esse comportamento foi conduzido por nós autenticados, o uso da criptografia pode não resolver completamente esses problemas. Também vale a pena mencionar que as falhas bizantinas podem ocorrer devido a causas não maliciosas. Por exemplo, configurações erradas da rede são bastante comuns e podem causar danos no mecanismo de estabelecimento de rota no plano de controle [Le et al. 2009, Rajendran et al. 2007].

Neste trabalho, assumimos que apenas o protocolo de roteamento é afetado por falhas bizantinas, enquanto o protocolo de sinalização é protegido. Assim, o mecanismo de *crankback* funcionará corretamente através dos nós da rede. Conforme detalhado em [Pavani et al. 2016], o mecanismo de *crankback* busca por uma rota alternativa sem necessitar da detecção dos nós afetados por falha bizantina.

Apesar de não haver garantias de otimalidade ou de completude do mecanismo de *crankback*, ele é bom o suficiente para diminuir a probabilidade de bloqueio sem incorrer em grandes latências ou rotas muito longas no estabelecimento de caminhos ópticos, já que o número de tentativas locais de rerroteamento é limitado para cada requisição de conexão [Farrel et al. 2007, Pavani et al. 2016].

4. Simulação

Para este trabalho, foi desenvolvido um software de simulação orientado a eventos, escrito em Java, para avaliar o algoritmo dinâmico de roteamento e alocação de comprimento de onda (*Routing and Wavelength Assignment* – RWA) baseado em ACO para redes ópticas na presença de falhas bizantinas.

Foram efetuadas diversas simulações com a topologia NSFNet, que é mostrada na Figura 4, sendo que os tempos nos enlaces representam o atraso de comunicação entre os nós. A NSFNet é uma rede com 14 nós e 21 enlaces bidirecionais e é bem equilibrada,

tendo comprimento médio do caminho mais curto entre todos os pares de nós iguais a 2,2 saltos e diâmetro igual a 3 [Di Caro and Dorigo 1998].

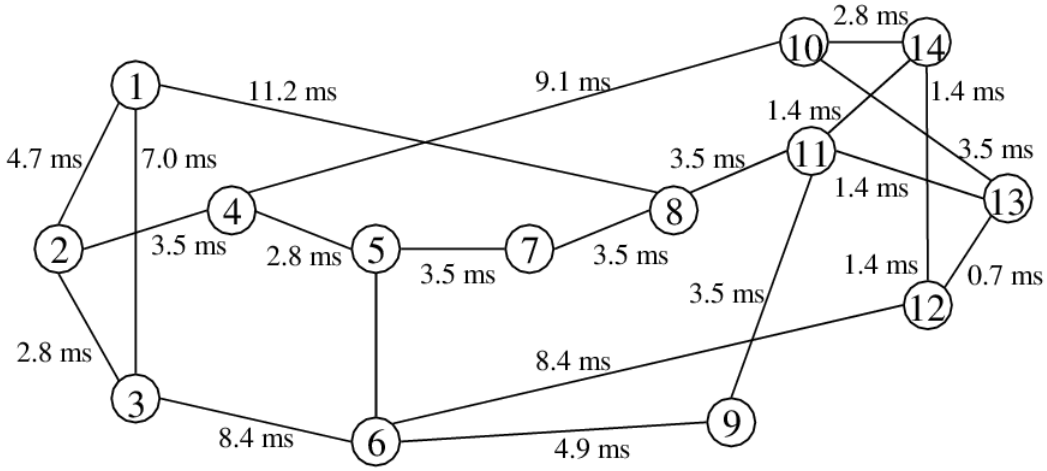


Figura 4. A rede NSFNet.

A Tabela 1 indica o número de simulações necessárias para se considerar todas as possibilidades de falha bizantinas para múltiplos nós. Com efeito, o número de simulações necessárias é igual ao número de combinações possíveis, isto é, $n!/(p!(n-p)!)$, em que n é o número total de nós da rede ($n = 14$) e p é o número de nós bizantinos. Consideramos neste trabalho até um total de 7 nós bizantinos, o que equivale à metade dos nós da rede NSFNet.

Tabela 1. Número de simulações para cada tipo de falha bizantina.

Nós bizantinos	Simulações
2	91
3	364
4	1.001
5	2.002
6	3.003
7	3.432

Assim, dado um determinado tipo de falha bizantina, para cada valor de carga, em que se considera de 2 a 7 nós bizantinos, temos um total de 9.983 simulações. Por esse motivo, este trabalho fica restrito a NSFNet e não leva em consideração a rede NTTNet (*Nippon Telephone and Telegraph*), como em [Pavani et al. 2016], em função do grande esforço computacional necessário para se considerar todas as combinações possíveis de falhas bizantinas envolvendo múltiplos nós.

Assume-se que, para cada tipo de falha bizantina, os nós bizantinos sempre exibem seu comportamento bizantino, como especificado na Seção 3 e desde o tempo de início da simulação, para se maximizar o impacto das falhas bizantinas no roteamento.

Consideramos um tráfego de Poisson, em que cada origem e destino são escolhidos de forma uniforme e com a mesma probabilidade. A duração de cada caminho óptico segue uma distribuição exponencial com um valor médio de 100 s. As simulações foram

realizadas com 8 comprimentos de onda por enlace. Utilizamos a abordagem *first-fit* para o subproblema de atribuição de comprimento de onda [Pavani and Waldman 2010].

O número máximo de saltos permitidos para uma formiga ou uma mensagem de sinalização RSVP-TE é igual a 42. Os demais parâmetros de simulação, incluindo os parâmetros da AntNet, são os mesmos que foram mostrados em [Pavani and Waldman 2010].

5. Resultados

Nos gráficos apresentados como resultados das simulações, a primeira curva representa a média de 10 execuções do algoritmo de roteamento ACO sem falhas bizantinas. As demais curvas representam a média de todas as combinações de simulação com nós apresentando falha bizantina no roteamento, em que se faz uso da seguinte notação: o algarismo representa o número de nós bizantinos e as duas letras seguintes indicam o tipo de falha bizantina, conforme sigla definida na Seção 3.

Cada simulação, seja com ou sem falhas bizantinas, gera 10^4 requisições de caminho óptico. As barras em cada ponto das curvas representam o erro padrão da média. Para facilitar a visualização dos resultados, em (a) se apresentam as curvas em escala linear de probabilidade de bloqueio, enquanto em (b) se apresentam as curvas em escala logarítmica de probabilidade de bloqueio.

A Figura 5 considera o roteamento ACO sem *crankback*, isto é, com a escolha gulosa de próximo salto [Pavani and Waldman 2006b]. A falha MF é a que tem menor impacto sobre a probabilidade de bloqueio, sendo que apenas quando a falha MF atinge dois nós é que não temos um aumento da probabilidade de bloqueio. Note que a natureza estocástica das escolhas de roteamento das formigas traz uma leve melhoria da probabilidade de bloqueio para o caso 2MF em relação ao caso sem falhas bizantinas, quando se considera as cargas mais baixas na rede.

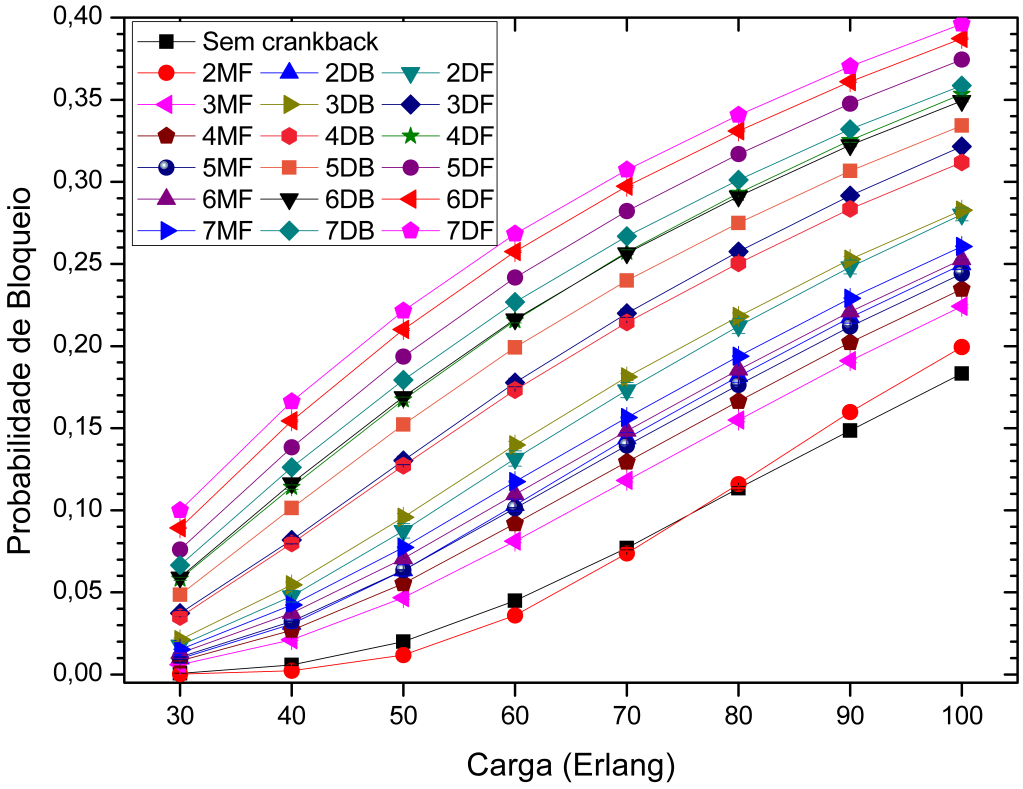
Sete nós com comportamento MF tem um impacto menor do que três nós exibindo o comportamento DB ou dois nós com falhas do tipo DF. Além disso, sete nós com o comportamento DB tem impacto menor na probabilidade de bloqueio do que cinco nós DF.

A Figura 6 considera o roteamento ACO com *crankback* com uma tentativa local de rerroteamento, isto é, com duas entradas na tabela histórica para cada tentativa de estabelecimento de LSP [Pavani and Waldman 2010]. Note que o desempenho, em termos de probabilidade de bloqueio, para a curva sem falhas bizantinas é melhor do que aquele apresentado na Figura 5.

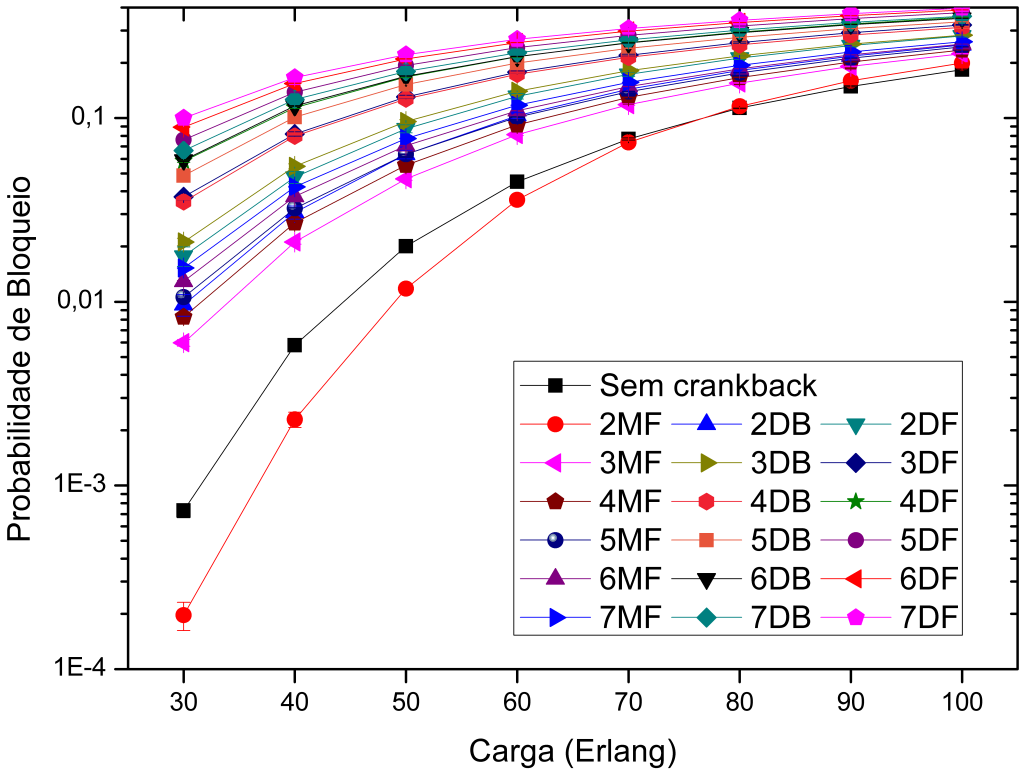
Como ambas as figuras estão na mesma escala, também é possível perceber que o mecanismo de *crankback* mitiga o impacto das falhas bizantinas para os três tipos de falhas avaliados.

Como já visto na Figura 5, sete nós com comportamento MF tem um impacto menor do que três nós exibindo o comportamento DB ou dois nós com falhas do tipo DF. Adicionalmente, sete nós com o comportamento DB tem impacto menor na probabilidade de bloqueio do que cinco nós DF.

A Figura 7 considera o roteamento ACO com *crankback* com duas tentativas lo-

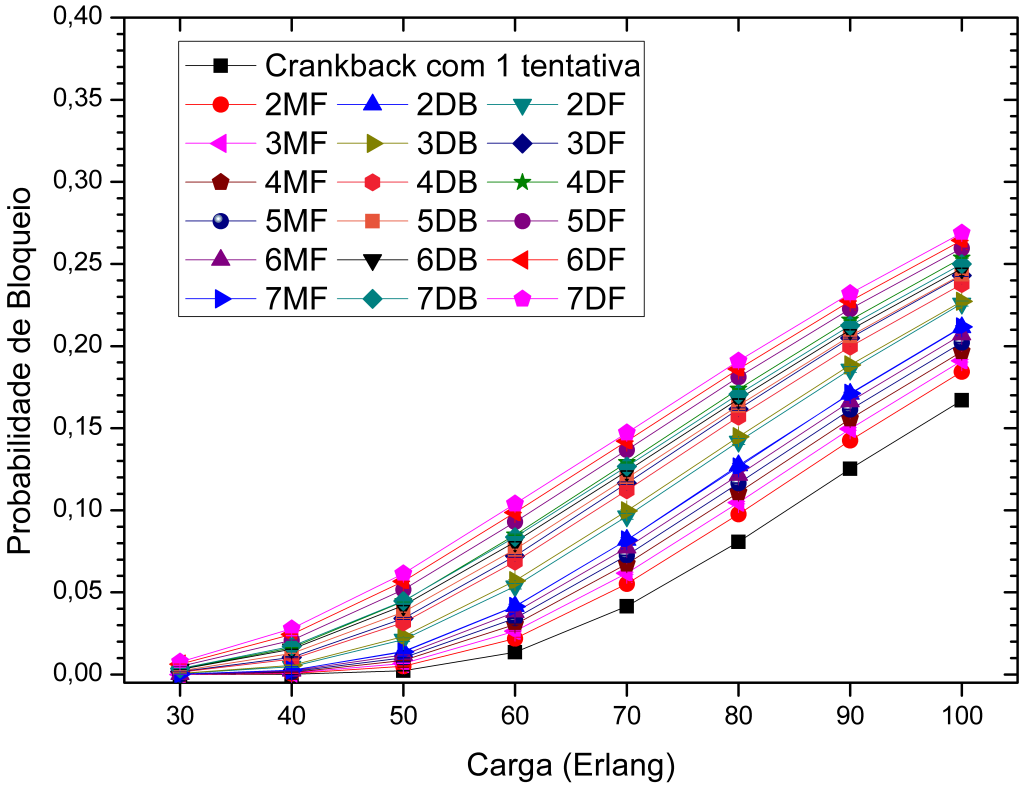


(a) Escala linear.

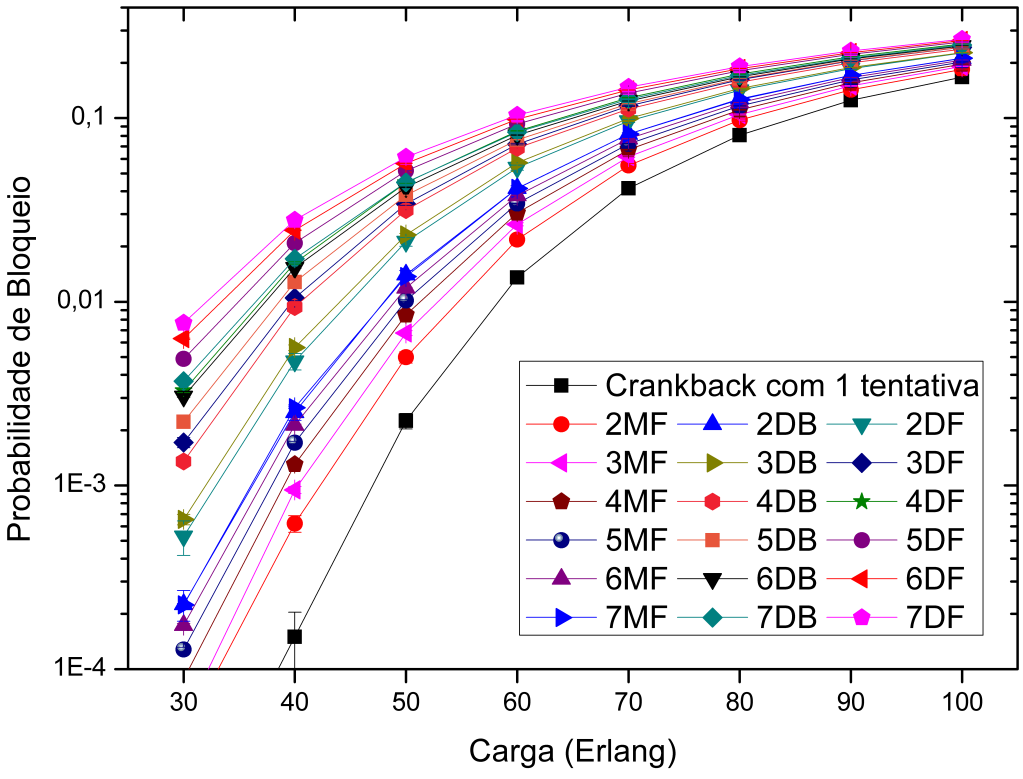


(b) Escala logarítmica.

Figura 5. Sem *Crankback*.

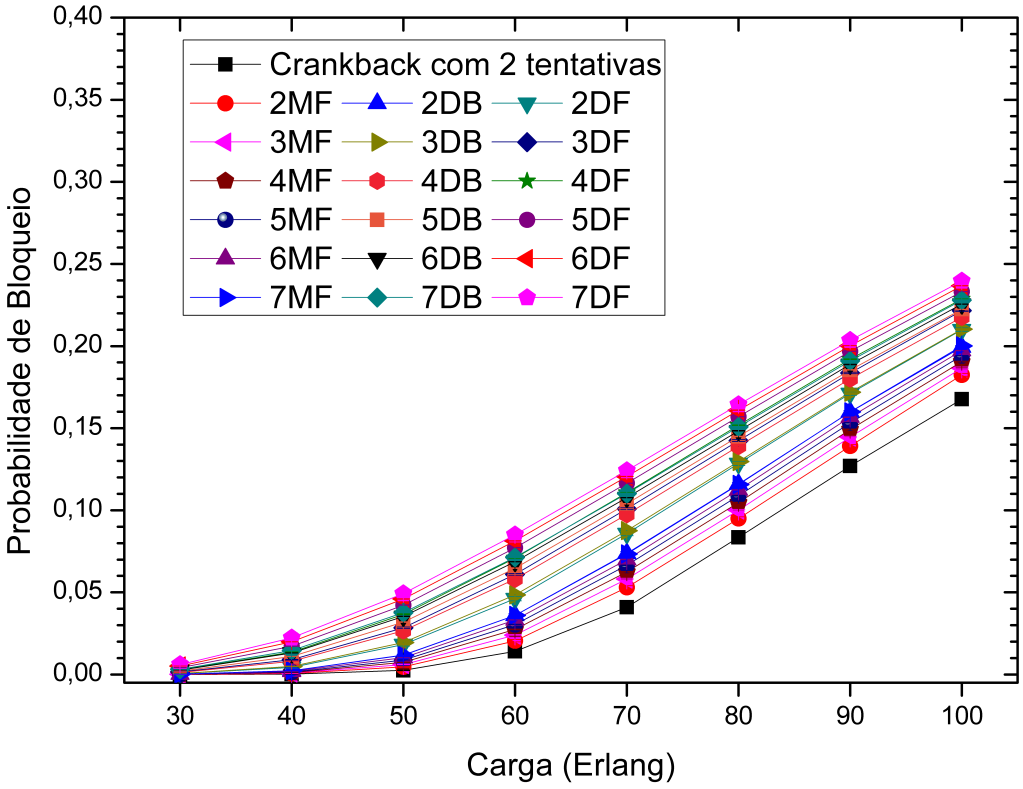


(a) Escala linear.

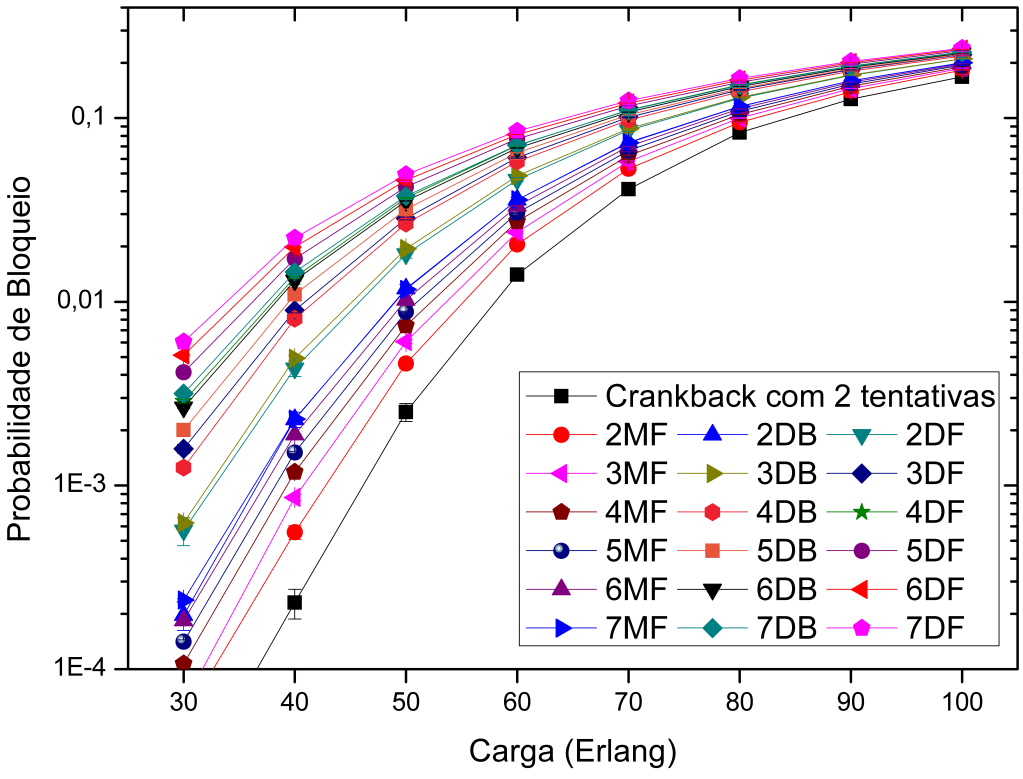


(b) Escala logarítmica.

Figura 6. *Crankback* com uma tentativa local de rerroteamento.



(a) Escala linear.



(b) Escala logarítmica.

Figura 7. *Crankback* com duas tentativas locais de rerroteamento.

cais de rerroteamento, isto é, com três entradas na tabela histórica para cada tentativa de estabelecimento de LSP [Pavani and Waldman 2010].

Como já mostrado na Figura 6, o mecanismo de *crankback* permite uma redução significativa na probabilidade de bloqueio em caso de falhas bizantinas. Com um número maior de tentativas locais de rerroteamento, é possível mitigar ainda mais o impacto das falhas bizantinas no mecanismo de atualização e reforço dos níveis de feromônio distribuídos pela rede, o que diminui a probabilidade de bloqueio.

As relações entre os impactos dos diferentes tipos de falhas continua similar àquelas observadas nas Figuras 5 e 6.

6. Conclusão

Falhas bizantinas podem ter um grande impacto na operação de redes ópticas roteadas por algoritmos baseados em ACO. Esse problema pode ser agravado se muitos nós da rede apresentarem um comportamento bizantino.

Para mitigar tal situação, foi proposto o uso de extensões de rerroteamento *crankback* associadas ao algoritmo ACO. Para três tipos de falhas apresentadas, foi possível demonstrar que o mecanismo *crankback* pode obter robustez bizantina mesmo na presença de muitos nós exibindo um comportamento bizantino, o que fecha uma importante lacuna deixada por [Pavani et al. 2016].

De fato, o mecanismo de *crankback* consegue mitigar o impacto das informações incorretas causadas por diferentes falhas bizantinas no roteamento distribuído da rede, que são armazenadas na forma de níveis de feromônios artificiais pelas formigas, sem a necessidade de se detectar ou localizar as falhas bizantinas, como em propostas anteriores na literatura [Pavani et al. 2016].

Agradecimentos

Os autores gostariam de agradecer o apoio financeiro concedido por meio do processo nº 2015/24341-7, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

Referências

Awerbuch, B., Holmer, D., Nita-Rotaru, C., and Rubens, H. (2002). An on-demand secure routing protocol resilient to byzantine failures. In *1st ACM Workshop on Wireless Security (WiSE 2002)*, pages 21–30.

Awerbuch, B., Holmer, D., and Rubens, H. (2003). Provably Secure Competitive Routing against Proactive Byzantine Adversaries via Reinforcement Learning. Technical report, Johns Hopkins University, Department of Computer Science.

Berger, L. (2003). Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions. RFC 3473 (Proposed Standard).

Di Caro, G. and Dorigo, M. (1998). AntNet: distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, 9:317–365.

Dorigo, M. and Stützle, T. (2004). *Ant Colony Optimization*. MIT Press.

Farrel, A., Satyanarayana, A., Iwata, A., Fujita, N., and Ash, G. (2007). Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE. RFC 4920 (Proposed Standard).

Grassé, P. P. (1959). La reconstruction du nid et les coordinations inter-individuelles chez Bellicoitermes natalenis et Cubitermes sp. La théorie de la stigmergie: Essai d'interprétation des termites constructeurs. *Insectes Sociaux*, 6:41–81.

Le, F., Lee, S., Wong, T., Kim, H. S., and Newcomb, D. (2009). Detecting network-wide and router-specific misconfigurations through data mining. *IEEE/ACM Transactions on Networking*, 17(1):66–79.

Mannie, E. (2004). Generalized Multi-Protocol Label Switching (GMPLS) Architecture. RFC 3945 (Proposed Standard).

Pavani, G. S., de França Queiroz, A., and Pellegrini, J. C. (2016). Analysis of ant colony optimization-based routing in optical networks in the presence of byzantine failures. *Information Sciences*, 340–341:27–40.

Pavani, G. S. and Waldman, H. (2006a). Evaluation of an ant-based architecture for all-optical networks. In *10th Conference on Optical Network Design and Modelling (ONDM'06)*, Copenhagen, Denmark.

Pavani, G. S. and Waldman, H. (2006b). Traffic engineering and restoration in optical packet switching networks by means of ant colony optimization. In *Third International Conference on Broadband Communications, Network and Systems (BroadNets 2006)*, pages 1–9, San Jose, CA.

Pavani, G. S. and Waldman, H. (2008). Restoration in wavelength-routed optical networks by means of ant colony optimization. *Photonic Network Communications*, 16(1):83–91.

Pavani, G. S. and Waldman, H. (2010). Routing and wavelength assignment with crankback re-routing extensions by means of ant colony optimization. *IEEE Journal on Selected Areas in Communications*, 28(4):532–541.

Perlman, R. (1988). *Network Layer Protocols with Byzantine Robustness*. PhD thesis, Massachusetts Institute of Technology.

Prehofer, C. and Bettstetter, C. (2005). Self-organization in communication networks: Principles and design paradigms. *IEEE Communications Magazine*, 43(7):78–85.

Rajendran, R. K., Misra, V., and Rubenstein, D. (2007). Theoretical bounds on control-plane self-monitoring in routing protocols. In *ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 2007)*, pages 325–336.

Wendlandt, D., Avramopoulos, I., Andersen, D., and Rexford, J. (2006). Donť secure routing protocols, secure data delivery. In *5th ACM Workshop on Hot Topics in Networks (HotNets-V)*, pages 7–12.

Zhong, W. and Evans, D. (2002). When ants attack: Security issues for stigmergic systems. Technical Report CS-2002-23, University of Virgina, Department of Computer Science.