# Diagnóstico em Redes de Topologia Arbitrária: Um Algoritmo Baseado em Inundação de Mensagens

Elias Procópio Duarte Jr.
elias@inf.ufpr.br
Universidade Federal do Paraná, Depto. de Informática
Caixa Postal 19081 – Curitiba PR 81531-990 Brasil

Giorgia de Oliveira Mattos
giorgia@pb.cefetpr.br
Centro Federal de Educação Tecnológica do Paraná - Unidade de Pato Branco
Rodovia PR 469, Km 01 - Pato Branco PR 85503-390 Brasil

## Resumo

Considere uma rede de computadores de topologia arbitrária. Um algoritmo de diagnóstico distribuído permite que os nodos sem falha desta rede determinem quais outros nodos da rede são alcançáveis em um determinado momento. Este trabalho apresenta um algoritmo de diagnóstico em redes de topologia arbitrária. Cada nodo da rede testa os seus vizinhos. Quando um nodo detecta a falha de um link este dissemina, em paralelo, para os seus vizinhos, uma pequena mensagem contendo três campos, o identificador dos nodos envolvidos e um contador indicando a falha, avisando-os da falha encontrada. Cada nodo, ao receber a mensagem, atualiza as informações locais de diagnóstico e dissemina a mensagem para seus vizinhos, exceto para aquele do qual a mensagem foi recebida. Quando um nodo já conhece o evento propagado pela mensagem, esta é simplesmente descartada. Experimentos obtidos através de simulação em redes de diferentes topologias confirmam que a latência do algoritmo é a melhor possível, isto é, proporcional ao diâmetro da rede. Os experimentos mostram que o número prático de mensagens redundantes geradas é, na média, menor que o máximo possível, isto é, o dobro do número de links.

## 1 Introdução

O uso das redes de computadores é cada vez maior nas mais diversas áreas de aplicação. Por outro lado, as redes têm se tornado maiores e mais complexas, sendo compostas por equipamentos heterogêneos de diversos tipos. Com isso, tornou-se necessário ferramentas automatizadas que permitam a monitoração e o controle eficientes de uma rede: os sistemas de gerência de redes [6].

Várias abordagens tem sido propostas com o objetivo de solucionar o problema de gerência de falhas em redes de computadores. Considere um sistema composto por nodos conectados por links de comunicação. Um algoritmo distribuído de diagnóstico permite aos nodos sem-falha do sistema determinarem quais nodos estão falhos e quais estão sem falha [12].

O modelo original de diagnóstico foi apresentado por Preparata, Metze e Chien [1] e também em [2] por Hakimi e Amin, nas décadas de 60 e 70. Em um sistema com N nodos, cada nodo testa um subconjunto de seus vizinhos, onde o nodo pode estar *sem-falha* ou *falho*.

Todos os resultados dos testes são enviados a um observador central que é responsável por interpretá-los e completar o diagnóstico de todos os nodos do sistema.

Bagchi e Hakimi, em [3] apresentam um algoritmo para diagnóstico em redes de topologia arbitrária que reduz ao máximo o número de mensagens trocadas entre os nodos. Porém, o algoritmo não é executado on-line; não há um monitoramento dinâmico e contínuo da rede, ou seja, durante a sua execução, um nodo não pode falhar ou recuperar-se.

Em [4], Stahl e outros apresentaram o algoritmo "Adapt". Este é executado on-line, mas emprega um procedimento distribuído que exige grandes quantidades de mensagens.

Rangarajan e outros apresentam em [5] um novo algoritmo para diagnosticar falhas em redes de topologia arbitrária. Este algoritmo é também chamado RDZ, das iniciais dos autores. O algoritmo RDZ é executado on-line, garante o menor número de testes por nodo e apresenta a menor latência de diagnóstico em função do uso de uma estratégia de propagação de mensagens em paralelo. Porém, o RDZ não identifica falhas em certas configurações, a chamada *jellyfish fault configuration* [5].

Em 1997 Duarte e outros em [6, 7], apresentam um algoritmo, para redes ponto-a-ponto de topologia arbitrária que resolve o problema de detecção de falhas na configuração *jellyfish*. Trata-se do algoritmo NBND. Este algoritmo permite o diagnóstico de time-outs de canais de comunicação, calculando a conectividade da rede usando o menor número de testes possível, um por canal, sendo a latência proporcional ao diâmetro da rede.

O algoritmo apresentado em [8] aplica-se a redes de topologia geral e o nodo executando o algoritmo conhece apenas os seus vizinhos, isto é, cada nodo não tem a topologia completa da rede, ao contrário do algoritmo NBND. Em [9] é apresentada uma estratégia de testes alternativa para o NBND, na qual os nodos vizinhos se alternam na execução de testes no link.

O algoritmo de diagnóstico em redes de topologia arbitrária apresentado neste trabalho considera falhas nos links, exigindo que cada nodo tenha conhecimento da topologia de toda a rede. O número de mensagens de disseminação da falha de um link trafegando pela rede é reduzido, devido a um mecanismo de propagação de mensagens em paralelo, onde as mensagens redundantes são descartadas. O tamanho da mensagem de disseminação é o menor possível, apenas três campos. Foram realizados testes de simulação em redes de topologia $D_{1,2}$, hipercubo, RNP e outras topologias específicas. Os experimentos mostram que o número prático de mensagens redundantes geradas é na média bem menor que o máximo possível, isto é, o dobro do número de links.

Este trabalho está estruturado da seguinte forma: na seção 2 o novo algoritmo baseado em inundação de mensagens é descrito; a seção 3 apresenta alguns resultados experimentais obtidos e a seção 4 conclui o trabalho.

# 2 Especificação do Algoritmo

No novo algoritmo apresentado neste trabalho, os nodos da rede testam uns aos outros, de forma que cada nodo testa todos os seus vizinhos. Cada nodo conhece os seus vizinhos através de uma lista de adjacências montada pelo algoritmo no início do seu funcionamento.

Um evento é definido como sendo a mudança de estado de um link, ou seja, o link passa do estado de sem-falha para o estado de falha ou do estado de falha para o estado de não-falha. Um evento de falha é uma transição do estado sem-falha para o estado de falha.

Os testes são realizados intervalos de 30 unidades de tempo. Em um intervalo de testes todos os nodos testam todos os links para seus vizinhos. Se os nodos testadores não detectarem nenhum novo evento, nada ocorre até o próximo intervalo, quando todos os nodos recomeçam a execução de testes.

Quando um evento de falha é detectado, mensagens de disseminação notificando este evento são propagadas através dos nodos vizinhos, de tal forma que, a partir de sua origem, todos os nodos sem falha sejam informados do novo evento. O nodo originador da mensagem a envia para todos os seus vizinhos, os quais, por sua vez, também a enviam para todos os seus vizinhos, exceto para o(s) nodo(s) de onde veio a mensagem, e assim por diante. Este processo de disseminação de mensagens, o qual é usado pelo algoritmo, é chamado inundação ou *flooding*.

A mensagem de disseminação é composta por três campos, o identificador do nodo testador, o identificador do nodo testado e um contador. Este contador, é inicializado em zero, indicando o estado inicial sem-falha. A cada novo evento ocorrido no link, o contador é incrementado de uma unidade, sendo assim possível determinar para cada link o número de eventos sofridos. Um valor par do contador indica que o link está sem falha; quando o valor é ímpar indica o link está falho.

Um mecanismo para descartar mensagens redundantes também é utilizado. Um nodo ao receber uma mensagem de disseminação compara as informações contidas na mensagem com as suas próprias informações. Assim, se as informações recebidas já forem conhecidas pelo nodo que as recebe, o nodo ignora a mensagem, caso contrário ele atualiza as informações locais, e dissemina a mensagem para seus vizinhos.

Em resumo, o algoritmo é como apresentamos a seguir:

```
Início
    Faça para sempre
        Caso teste:
            Para cada link i-j que conecta o nodo i ao nodo j
                Testa link i-j
                Usando o resultado do teste efetuado em j
                Se o estado do link não mudou
                    Dormir até o início do próximo intervalo de testes
                Senão
                    Atualizar informações locais
                    Montar a mensagem de disseminação
                    Iniciar o evento dissemina

        Caso dissemina:
            Para cada vizinho z de i
                Se a informação recebida não é conhecida por z
                    Enviar a mensagem dissemina
Fim.
```

# 3 Resultados Experimentais

As simulações da execução do algoritmo mostrado neste trabalho foram feitas usando a linguagem de simulação de eventos discretos SMPL [10].

O algoritmo foi executado nos mais diversos tipos de topologias, dentre elas a $D_{1,2}$, hipercubo e RNP [11].

A seguir a descrição dos resultados obtidos após a execução do algoritmo em uma topologia especifica, mostrada na figura 1.

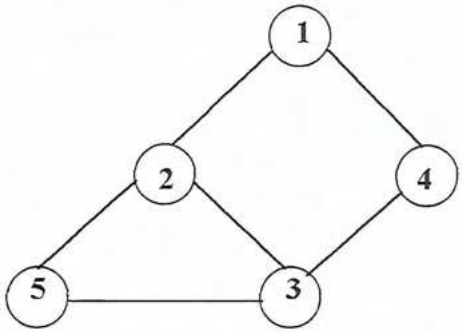


Figura 1: Rede de topologia específica com 5 nodos; link 1-4 torna-se falho em T=40.0

O algoritmo inicia a sua execução em T=0,0. Neste instante de tempo o nodo 1 começa a testar os nodos 2 e 4, o nodo 2 testa os nodos 3 e 5, e assim sucessivamente. Como neste intervalo de testes não houve um novo evento, os testes são novamente iniciados em T=30,0. No instante T=40,0 o link entre os nodos 1 e 4 torna-se falho. Em T=30,0 também não houve um novo evento e os testes são reiniciados em T=60,0. Quando o nodo 1 testa o nodo 4, aquele detecta a falha no link e prepara-se para iniciar o processo de disseminação de mensagens, de acordo com o novo evento detectado. Neste momento a mensagem de disseminação é formada contendo o identificador do nodo testador igual a 1, o identificador do nodo testado igual a 4 e o contador igual a 1. O nodo 1 inicia o evento de disseminação da mensagem para o nodo 2 em T=61,0. O nodo 2 dissemina para os nodos 3 e 5 em T=62,0. O nodo 3 dissemina a mensagem para o nodo 4 e para o nodo 5 (esta redudante) em T=63.0. Neste instante o nodo 5 manda a mensagem (redundante) para o nodo 3. Neste momento o algoritmo é encerrado pois todos os nodos já receberam a mensagem. Neste exemplo o número total de mensagens na rede foi 6, sendo que 4 mensagens foram disseminadas e 2 mensagens foram redundantes, trocadas entre o nodo 3 e o nodo 5.

A execução do algoritmo em uma rede de topologia $D_{1,2}$ com 4 nodos e 6 links, os resultados obtidos foi o total de mensagens igual a 7, sendo 3 mensagens de disseminação e 4 mensagens redundantes. Em uma rede de topologia $D_{1,2}$, mostrada na figura 2, com 8 nodos e 16 links, os resultados obtidos foi o total de mensagens igual a 23, sendo 7 mensagens de disseminação e 16 mensagens redundantes.

De acordo com os resultados apresentados acima, podemos concluir que a latência do algoritmo é a melhor possível, ou seja, ela é proporcional ao diâmetro da rede. Os resultados obtidos através das simulações mostram também que o número prático de mensagens redundantes geradas é, na média, bem menor que o máximo possível, isto é, o dobro do número de links. Por exemplo, na topologia $D_{1,2}$ com 4 nodos e 6 links o número de mensagens redundantes é 4 mensagens, este número é bem menor que o máximo possível, ou seja 12 mensagens. Na mesma topologia, com 8 nodos e 16 links, o número de mensagens

redundantes obtida foi 16 mensagens, menor que o número máximo de mensagens possível, 32.

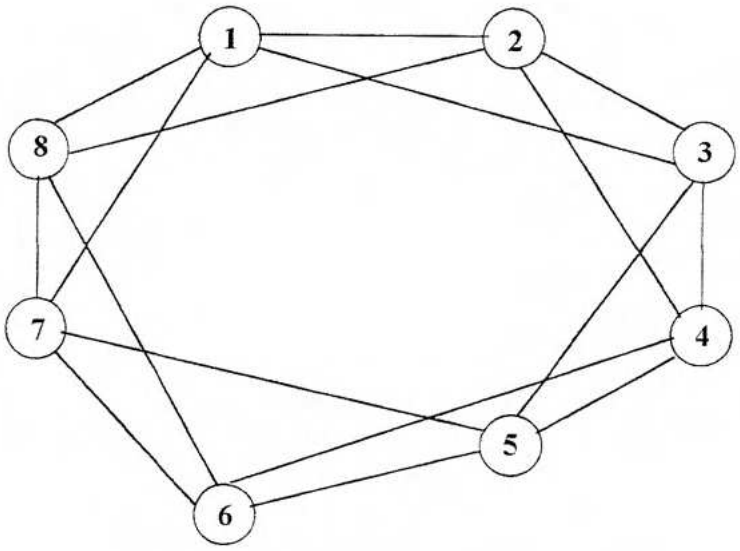


Figura 2: Topologia de rede $D_{1,2}$ com 8 nodos.

# 4 Conclusão

Apresentamos neste artigo um novo algoritmo para diagnóstico em redes de topologia arbitrária utilizando a técnica de disseminação de mensagens conhecida como inundação ou *flooding*.

O algoritmo aqui apresentado, detecta falhas nos links e envia aos nodos alcançáveis da rede uma mensagem de disseminação avisando-os de que um novo evento ocorreu. De acordo com resultados obtidos através de simulações, podemos confirmar que o algoritmo apresenta a melhor latência possível, sendo esta proporcional ao diâmetro da rede. Concluímos ainda que o número de mensagens redundantes geradas é bem menor ao máximo possível ou seja, o dobro do número de links.

Como trabalho futuro, esperamos executar o algoritmo em outras topologias bem como comparar os resultados obtidos com os resultados de outros algoritmos.

# Referências

[1] F. P. Preparata, G. Metze, and R. T. Chien, "On the Connection Assignment Problem of Diagnosable Systems," IEEE Trans. Electrom. Comput., vol. EC-16, pp. 848-854, Dec, 1967.

[2] S. L. Hakimi, and A. T. Amin, "Characterization of Connection Assignment of Diagnosable Systems," IEEE Trans. Comput., vol. C-23, Jan, 1974.

[3] A. Bagchi, and S. L. Hakimi, "An Optimal Algorithm for Distributed System-Level Diagnosis," Proc. 21st Fault Tolerant Computing Symp., June, 1991.

[4] M. Stahl, R. Buskens, and R. Bianchini, "Simulation of the Adapt On-Line Diagnosis Algorithm for General Topology Networks," Proc. IEEE 11th Symp. Reliable Distributed Systems, October, 1992.

[5] S. Rangarajan, A. T. Dahbura, and E. A. Ziegler, "A Distributed System-Level Diagnosis Algorithm for Arbitrary Network Topologies," IEEE Transactions and Computers, Vol. 44, pp. 312-333, 1995.

[6] E. P. Duarte Jr., T. Nanya, G. Mansfield, and S. Nogushi, "Non-Broadcast Network Fault-Monitoring Based on System-Level Diagnosis," Proc. IEEE/IFIP IM'97, 1997.

[7] E. P. Duarte Jr., "Um Algoritmo para Diagnóstico de Redes de Topologia Arbitrária," I Workshop de Tolerância a Falhas da SBC, Porto Alegre, 1998.

[8] J. E. B. Maia, M. A. C. Branco, "Um Algoritmo de Diagnóstico para Redes Considerando Falhas nos Links," XIX Congresso Nacional da Sociedade Brasileira de Computação, Rio de Janeiro, 1999.

[9] J. I. Siqueira, E. Fabris, E. P. Duarte Jr., "A Token Based Testing Strategy for Non-Broadcast Network Diagnosis," 1st IEEE Latin American Test Workshop, pp. 166-171, Rio de Janeiro, 2000.

[10] M. H. MacDougall, *Simulating Computer Systems: Techniques and Tools*, The MIT Press, Cambridge, MA, 1987.

[11] RNP – Rede Nacional de Pesquisa. http://www.rnp.br/backbone/bkb-mapa.html.

[12] P. Jalote, *Fault-Tolerance in Distributed Systems*, Prentice-Hall, 1994.