

# Roteamento Tolerante a Falhas Baseado em Desvios de Alta Conectividade

Rogério Santini<sup>1</sup>, Elias P. Duarte Jr.<sup>1</sup>, Jonatan Schroeder<sup>1</sup>,  
Pedro R. Torres Jr.<sup>2</sup>, Jaime Cohen<sup>3</sup>

<sup>1</sup> Departamento de Informática - Universidade Federal do Paraná  
Centro Politécnico - Jardim da Américas – Caixa Postal 19081  
81531-990 Curitiba, PR E-Mail: {santini, elias, jonatan}@inf.ufpr.br

<sup>2</sup> PoP-PR Ponto de Presença - Paraná - RNP - Rede Nacional de Pesquisa e Ensino  
Centro Politécnico - Jardim da Américas  
81531-990 Curitiba, PR E-Mail: torres@pop-pr.rnp.br

<sup>3</sup> Departamento de Informática Universidade Estadual de Ponta Grossa  
R. C. Cavalcanti, 4748  
84031-900 Ponta Grossa, PR E-Mail: jaime@uol.com.br

Relatorio Tecnico # RT001/2004  
Universidade Federal do Parana'  
Departamento de Informatica  
<http://www.inf.ufpr.br/info/techrep/index.html>

**Abstract.** *Routing protocols present a convergence latency for all routers to update their routing tables after the network topology changes. During this time interval, which may be of up to minutes, packets may be lost before reaching their destinations. In order to allow nodes to continue communicating during the convergence latency interval, we propose the use of alternative routes that are employed to send packets that were lost. These alternative routes are chosen without the knowledge of the corrent topology. In the proposed approach, alternative routes are created through network nodes called detours, that belong to network components that have high connectivity. These components present a larger number of distinct paths, thus the probability that the alternative route created through a detour works correctly is higher. The connectivity criteria used to select detours are presented. An exact algorithm to compute those criteria, as well as an efficient heuristic are also presented. Experimental results were obtained with random graphs generated with the Waxman method, which aims at producing Internet-like topologies. Results show that the fault coverage obtained through the usage of the best detour, computed based on the proposed connectivity criterias, is up to 90%. When the three best detours are considered, the fault coverage is up to 98%. These results confirm the effectiveness of the proposed criteria for selecting detours that avoid unknown faulty elements.*

**Resumo.** *O protocolo de roteamento BGP-4, utilizado no roteamento da Internet, pode levar minutos para alcançar a convergência depois da ocorrência de uma mudança na topologia, como por exemplo uma falha de enlace, gerando potenciais perdas de pacotes e de conexão entre as aplicações que se comunicam através da rede. Com o objetivo*

*de permitir que os nodos da rede continuem a se comunicar durante este período, este trabalho apresenta uma abordagem para roteamento tolerante a falhas. Rotas alternativas são criadas através de nodos chamados desvios que pertencem a componentes de maior conectividade, possuindo grande probabilidade de desviar de uma falha da rede, sem a informação da localização da mesma. Critérios de conectividade para a escolha dos desvios são propostos, assim como um algoritmo exato e uma heurística para cálculo de tais critérios. Resultados experimentais de cobertura de falhas, obtidos a partir de simulação sobre topologias geradas com o método Waxman, comprovam que as rotas alternativas criadas através do melhor desvio possuem probabilidade de até 90% de desviar de uma falha ocorrida na rede, mesmo sem a informação da localização da mesma. Quando os três melhores desvios são utilizados, a cobertura de falhas obtida é de até 98%.*

## 1. Introdução

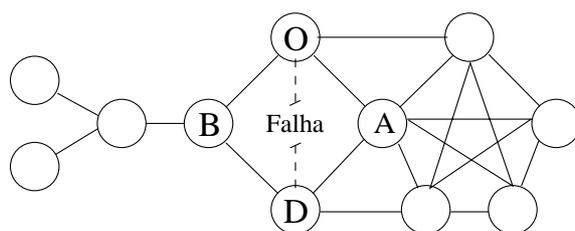
Atualmente, o protocolo de roteamento externo utilizado no roteamento da Internet é o BGP-4, *Border Gateway Protocol* versão 4 [1]. Os protocolos de roteamento necessitam de tempo para atualizar as tabelas de rotas de todos os roteadores, de modo a refletir uma mudança que tenha ocorrido na topologia da rede. Este período de tempo é conhecido como *latência de convergência* ou simplesmente *latência* do protocolo. Quando as tabelas de rotas de todos os roteadores é atualizada, é dito que o protocolo de roteamento convergiu para uma nova tabela de rotas estável, ou seja, alcançou a convergência [2]. A latência média do protocolo BGP, e conseqüentemente da Internet, é de 3 minutos, sendo que foram observados períodos de latência de até 15 minutos [3]. Durante o período de tempo compreendido pela latência do protocolo, alguns roteadores possuem informações inconsistentes nas suas tabelas de rotas, gerando potenciais perdas de pacotes e de conexão entre as aplicações que se comunicam através da rede [3].

Este trabalho apresenta uma abordagem para roteamento tolerante a falhas, que tem como objetivo permitir que os nodos da rede continuem a se comunicar, mesmo durante o período de tempo em que as tabelas de rotas dos roteadores ainda não foram atualizadas para refletir uma falha ocorrida na rede. Os nodos da rede fazem uso de uma *rota alternativa*, que possui grande probabilidade de desviar da falha. A rota alternativa é utilizada antes do protocolo de roteamento alcançar a convergência. Portanto, a rota alternativa é utilizada sem a informação da localização da falha.

Rotas alternativas são criadas através de nodos da rede chamados *desvios*, pertencentes a componentes de maior conectividade. O número de caminhos distintos que atravessam um componente de maior conectividade é maior, se comparado a um componente de menor conectividade. Desta forma, é maior a probabilidade da rota alternativa, criada através de um desvio pertencente a um componente de maior conectividade, ser funcional mesmo na presença de falhas na rede.

A figura 1 apresenta um exemplo de seleção do melhor desvio para os nodos  $O$  e  $D$ , assumindo que a rota entre os mesmos está falha. Neste caso, o nodo  $A$  é escolhido por pertencer a um componente de maior conectividade.

Para a determinação do grau de conectividade de cada nodo da rede são utilizados os critérios de conectividade *número de conectividade*  $\#C(v)$  e *conjunto*  $MCC(v)$ . Estes



**Figura 1: Um exemplo da seleção do melhor desvio para um par de nodos  $O$  e  $D$ .**

critérios são baseados na topologia da rede. Além do algoritmo exato, uma heurística eficiente é apresentada para o cálculo do número de conectividade  $\#C(v)$ .

Para avaliar a eficiência da estratégia de seleção de desvios baseada nos critérios de conectividade propostos, foi realizado o cálculo da cobertura de falhas quando são utilizados o primeiro, segundo e terceiro melhores desvios. Além disso, foi implementado um algoritmo que calcula a diferença entre o  $\#C(v)$  médio obtido através da árvore de corte e o  $\#C(v)$  médio obtido através da heurística, com o objetivo de avaliar a heurística.

Para a obtenção dos resultados experimentais, foram gerados aleatoriamente 2000 grafos utilizando o método Waxman, com tamanhos que variam de 10 a 100 nodos e grau médio dos nodos variando de 3 a 8. O método Waxman foi o escolhido pois o mesmo leva em consideração a distância entre dois vértices no cálculo da probabilidade de existência de uma aresta entre os mesmos, propondo-se desta forma a capturar a característica de localidade existente nas redes de computadores reais, em especial a Internet.

Os resultados experimentais demonstram que a heurística obtém valores de  $\#C(v)$  próximos aos valores obtidos através da árvore de corte. A cobertura de falhas obtida pelo melhor desvio selecionado a partir dos critérios de conectividade propostos variou de 70% a 90%, dependendo do grau médio dos nodos do grafo. Quando os três melhores desvios foram utilizados em conjunto, a cobertura de falhas variou de 81% até 98%, dependendo do grau médio dos nodos do grafo. A cobertura de falhas obtida pelos melhores desvios selecionados a partir dos critérios de conectividade retornados pela heurística apresentou valores similares aos valores apresentados acima. Estes resultados visam comprovar que rotas alternativas, criadas através de desvios selecionados a partir dos critérios de conectividade propostos, possuem boa probabilidade de desviar de uma falha na rede, mesmo sem a informação da localização da falha.

O restante deste trabalho está organizado da seguinte maneira. A seção 2 apresenta a estratégia proposta de roteamento tolerante a falhas, motivada por resultados referentes à avaliação da latência do protocolo BGP, e em seguida os conceitos de desvio e rota alternativa. A seção 3 apresenta os critérios de conectividade e algoritmos para seu cálculo. A seção 4 apresenta os resultados experimentais e por fim a seção 5 apresenta as conclusões do trabalho.

## **2. Roteamento Tolerante a Falhas**

Esta seção apresenta uma abordagem para roteamento tolerante a falhas baseada em desvios de alta conectividade, motivada por resultados referentes à avaliação da latência de convergência do protocolo BGP.

## 2.1. Latência de Convergência de um Protocolo de Roteamento

Entende-se por *latência de convergência* ou simplesmente *latência* de um protocolo de roteamento como sendo o tempo transcorrido desde uma mudança na topologia da rede, até que todos os roteadores que utilizam tal protocolo tenham atualizado suas tabelas de rotas de modo a refletir a mudança. Neste caso, é dito que o protocolo de roteamento convergiu para uma nova tabela de rotas estável, ou seja, alcançou a convergência.

A latência do protocolo BGP-4, utilizado no roteamento da Internet, é influenciada pelos parâmetros *Hold Time* e *MinRouteAdvertisementInterval*, definidos na RFC-1771 [1]. O parâmetro *Hold Time* especifica o tempo máximo que se pode transcorrer entre o recebimento de duas mensagens de um mesmo roteador BGP. O valor do parâmetro *Hold Time* sugerido pela RFC-1771 é 90 segundos. O parâmetro *Hold Time* influencia no tempo de detecção de falhas no enlace utilizado por dois roteadores BGP. Por exemplo, suponha que no instante  $T_0$  um roteador *A* envia uma mensagem a um roteador *B* e que logo em seguida ocorre uma falha no enlace que *A* e *B* utilizam para se comunicar. O roteador *B* pode detectar a falha somente no instante  $T_{90}$ , quando o valor do *Hold Time* expirar.

Já o parâmetro *MinRouteAdvertisementInterval* especifica a quantidade de tempo que deve existir entre dois anúncios de rotas para um determinado destino, enviados a um mesmo roteador BGP. O valor sugerido pela RFC-1771 para o parâmetro *MinRouteAdvertisementInterval* é 30 segundos. O parâmetro *MinRouteAdvertisementInterval* influencia no tempo de propagação de modificações que ocorrem nas tabelas de rotas. Por exemplo, suponha que no instante  $T_0$  um roteador *A* anuncia uma rota *X* a um roteador *B*, e que logo após o roteador *A* recebe uma modificação da rota *X*. Somente no instante  $T_{30}$ , quando o valor do *MinRouteAdvertisementInterval* expirar, que o roteador *A* poderá anunciar a modificação na rota *X* roteador *B*.

Em [4], Craig Labovitz *et al.* apresentam um estudo do impacto da topologia e das políticas de roteamento na latência do protocolo BGP. O autor mostra que a latência ponto a ponto da Internet depende do comprimento do caminho mais longo entre um nodo de origem e um nodo de destino. Na média, cada AS acrescenta um atraso equivalente à metade do valor do parâmetro *MinRouteAdvertisementInterval* no anúncio de uma rota. Utilizando o valor sugerido pela RFC-1771 para o valor do *MinRouteAdvertisementInterval*, temos que, na média, a latência do protocolo BGP entre uma origem *A* e um destino *B* é igual a:

$$\text{MinRouteAdvertisementInterval} / 2 * |p| = 15 * |p| \text{segundos}$$

onde  $|p|$  é a distância, em número de AS's, do caminho mais longo entre *A* e *B*. Ainda temos que o valor limite para a latência do protocolo BGP entre uma origem *A* e um destino *B* é igual a:

$$\text{MinRouteAdvertisementInterval} * |p| = 30 * |p| \text{segundos}$$

onde  $|p|$  é a distância, em número de AS's, do caminho mais longo entre *A* e *B*. Adicionando a este valor, a quantidade máxima de tempo necessária para um roteador detectar uma falha em uma de suas conexões BGP, temos que a latência do protocolo BGP entre uma origem *A* e um destino *B* pode ser igual a:

$$\text{HoldTime} + \text{MinRouteAdvertisementInterval} * |p| = 90 + 30 * |p| \text{segundos}$$

onde  $|p|$  é a distância, em número de AS's, do caminho mais longo entre A e B.

Em [3], Craig Labovitz *et al.* mostram que mudanças na topologia da Internet resultam em uma latência significativa até que o protocolo BGP alcance a convergência. Medições realizadas em 5 ISP's comerciais durante o período de 2 anos demonstram que a latência média do protocolo BGP, e conseqüentemente da Internet é de 3 minutos, sendo que algumas mudanças na topologia apresentaram uma latência de 15 minutos. Este atraso na convergência do protocolo BGP causa perda de pacotes, perda de conexão e atrasos na comunicação fim a fim na Internet. Medições realizadas nos 5 ISP's indicam que, após acontecer uma falha na rede e durante o período compreendido pela latência do protocolo, a perda de pacotes na comunicação fim a fim aumentou 30 vezes e o tempo de resposta na comunicação fim a fim aumentou 4 vezes. Além disso, trabalhos mostram casos de divergência do protocolo BGP [5, 6]. Ou seja, podem existir situações em que, após uma falha, o protocolo BGP nunca alcance a convergência.

## 2.2. Rotas Alternativas Através de Desvios

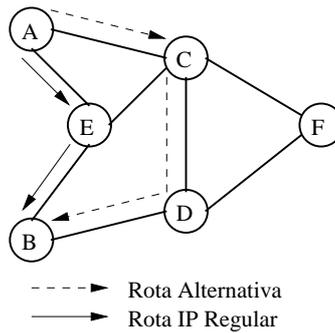
O objetivo da abordagem de roteamento tolerante a falhas apresentado neste trabalho é permitir que os nodos da rede continuem a se comunicar, mesmo durante o período de tempo em que as tabelas de rotas dos roteadores ainda não foram atualizadas de modo a refletir uma mudança ocorrida na rede. Para este propósito, *rotas alternativas* são criadas através de nodos da rede chamados *desvios*.

Um *desvio* é um nodo que possui a capacidade de atuar como uma ponte na comunicação de um nodo origem com um nodo destino. Uma *rota alternativa* de um nodo origem para um nodo destino é uma rota formada por duas rotas IP: a primeira é a rota IP do nodo origem para o desvio, e a segunda é a rota IP do desvio para o nodo destino. Entende-se por rota IP regular de um nodo  $X$  para um nodo  $Y$  como sendo a rota IP que o nodo  $X$  utiliza para comunicar-se com o nodo  $Y$ .

Um exemplo de uma rota alternativa do nodo origem  $A$  para o nodo destino  $B$ , criada através do desvio  $C$ , é apresentado na figura 2. No exemplo, a rota IP regular de  $A$  para  $B$  é dada por  $A - E - B$ . Já a rota alternativa de  $A$  para  $B$  é formada pela rota IP regular de  $A$  para  $C$ , no caso  $A - C$ , e pela rota IP regular de  $C$  para  $B$ , no caso  $C - D - B$ . Ou seja, a rota alternativa do nodo  $A$  para o nodo  $B$ , através do desvio  $C$  é dada por  $A - C - D - B$ . O conceito de desvio já foi apresentado em trabalhos que tratam da localização de rotas para mensagens SNMP no nível de aplicação [7], e da análise de performance de roteadores na Internet utilizando o conceito de tunelamento de pacotes TCP [8].

A rota alternativa é utilizada quando a rota IP de um nodo origem para um nodo destino não está funcional, ou seja, quando existe uma falha em um nodo ou enlace que faz parte da rota IP. Além disso, a rota alternativa é utilizada antes do protocolo de roteamento alcançar a convergência. Ou seja, a rota alternativa é utilizada sem a informação da localização da falha. Desta forma, a rota alternativa deve possuir grande probabilidade de desviar de uma falha ocorrida na rede, sem o conhecimento da localização da falha.

A seleção do desvio a ser utilizado na criação da rota alternativa é realizada levando-se em consideração critérios de conectividade baseados na topologia da rede. Nodos da rede pertencentes a componentes de maior conectividade possuem preferência para serem escolhidos como desvios. O número de caminhos distintos que atravessam um



**Figura 2: Um exemplo de rota alternativa do nodo A para o nodo B.**

componente de maior conectividade é maior, se comparado a um componente de menor conectividade. Desta forma, é maior a probabilidade da rota alternativa, criada através de um desvio pertencente a um componente de maior conectividade, ser funcional mesmo na presença de falhas na rede. Além da conectividade, o comprimento da rota alternativa também é levado em consideração na seleção do desvio.

### 3. Critérios de Conectividade $\#C(v)$ e $MCC(v)$

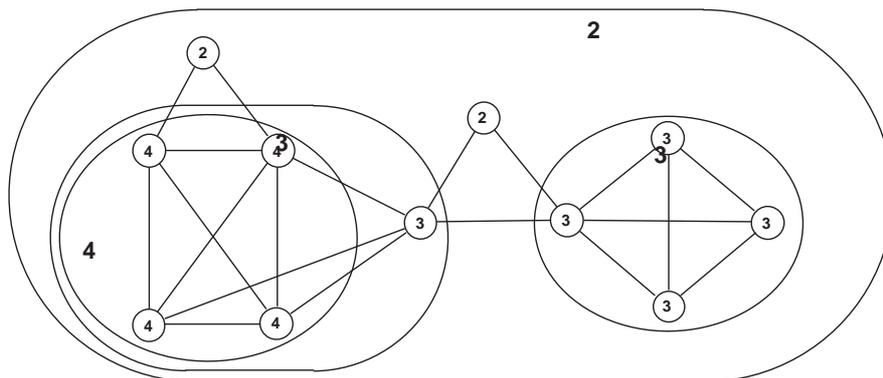
A seguir, são apresentados os critérios de conectividade propostos em [9] usados na seleção do nodo desvio. A topologia física da rede sem a presença de falhas, modelada por um grafo não direcionado, é utilizada no cálculo dos critérios de conectividade. Um grafo não direcionado é um par  $G = (V, E)$ , onde  $V$  é um conjunto finito de elementos chamados vértices e  $E$  é um conjunto finito de elementos chamados arestas. Uma aresta é um conjunto  $\{u, v\}$  onde  $u, v \in V$  [10].

**Definição 3.1** O número de conectividade  $\#C(v)$ ,  $v \in G$ , é definido como a conectividade de arestas de um subconjunto de vértices não unitário de  $G$  contendo  $v$  tal que a cardinalidade de qualquer corte em  $G$  que desconecta vértices deste subgrafo é maximizada.

**Definição 3.2** Seja o conjunto  $MCC(v)$  o maior subconjunto de vértices de  $G$  contendo  $v$ , tal que qualquer par de vértices deste subconjunto não pode ser desconectado por um corte em  $G$  com cardinalidade menor que  $\#C(v)$ .

A figura 3 mostra um exemplo do cálculo do  $\#C(v)$  e  $MCC(v)$  em um grafo  $G$ . Cada vértice  $v$  está rotulado com o seu valor de  $\#C(v)$ . Os conjuntos  $MCC(v)$  estão identificados por círculos rotulados pelo valor  $\#C(v)$  correspondente. Considere, por exemplo os vértices rotulados com o valor 4 e o subconjunto formado por tais vértices. Observa-se que a cardinalidade de qualquer corte que desconecta vértices deste subconjunto é máxima e é maior ou igual a 4. Ou seja, a conectividade de arestas de tal subconjunto de vértices é 4, e portanto o  $\#C(v)$  de todos os vértices pertencentes a este subconjunto é 4. Além disso, tal subconjunto de vértices é o maior subconjunto tal que qualquer par de vértices não pode ser desconectado por um corte com cardinalidade menor que 4, sendo assim um conjunto  $MCC(v)$ .

O seguinte lema é o ponto chave para o cálculo do critério de conectividade  $\#C(v)$ :



**Figura 3: Um exemplo de números de conectividade  $\#C(v)$  e conjuntos  $MCC(v)$ . Cada vértice  $v$  está rotulado com o seu valor  $\#C(v)$ . Os conjuntos  $MCC(v)$  são identificados por círculos.**

**Lema 1** Dado um grafo  $G = (V, E)$  e um vértice  $v \in V$ , o valor do  $\#C(v)$  é igual à cardinalidade máxima entre todos os cortes mínimos separando  $v$  de qualquer outro vértice de  $G$ .

**Prova:** Considere um subconjunto de vértices não unitário  $V'$  de  $G$  contendo  $v$  e com conectividade de arestas igual a  $\#C(v)$ . Pela definição de conectividade de arestas de um subconjunto de vértices  $V'$  de  $G$ ,  $\#C(v)$  é igual à cardinalidade do menor corte mínimo que separa dois vértices de  $V'$ . Pela definição do  $\#C(v)$  a cardinalidade de um corte separando  $v$  de outro vértice  $u \in V'$  é maximizada. Desta forma,  $\#C(v)$  é igual à cardinalidade máxima entre todos os cortes mínimos separando  $v$  de qualquer outro vértice de  $G$ . ■

O lema 1 mostra que o valor do  $\#C(v)$  para  $v \in V$  pode ser obtido através do cálculo do corte mínimo entre  $v$  e todos os outros vértices de  $V$ . O corte mínimo entre cada par de vértices de um grafo pode ser obtido através dos algoritmos de R.E. Gomory e T.C. HU [11] ou D. Gusfield [12]. Ambos os métodos constroem uma árvore de Gomory-Hu ou árvore de corte.

### 3.1. Cálculo dos Critérios de Conectividade Utilizando Árvores de Corte

Uma *árvore de corte*  $T$  de um grafo  $G$  é definida como uma árvore cujas arestas possuem pesos, e além disso:

1. Os vértices de  $G$  correspondem aos vértices da árvore  $T$ ;
2. A cardinalidade de um corte mínimo entre dois vértices de  $G$  é dada pelo valor da aresta de menor peso pertencente ao caminho que liga os dois vértices correspondentes em  $T$ ;
3. Um corte mínimo entre dois vértices de  $G$  é obtido removendo a aresta mencionada acima e considerando os dois conjuntos de vértices induzidos pelas duas árvores formadas pela remoção de tal aresta em  $T$ .

A figura 4 apresenta um exemplo da árvore de corte do grafo da figura 3. Através da árvore de corte, por exemplo, pode-se determinar que a cardinalidade do corte mínimo entre os vértices  $a$  e  $b$  é igual a 4 pois a aresta de menor peso no caminho que liga os vértices  $a$  e  $b$  possui peso 4. Analogamente, a cardinalidade do corte mínimo entre os

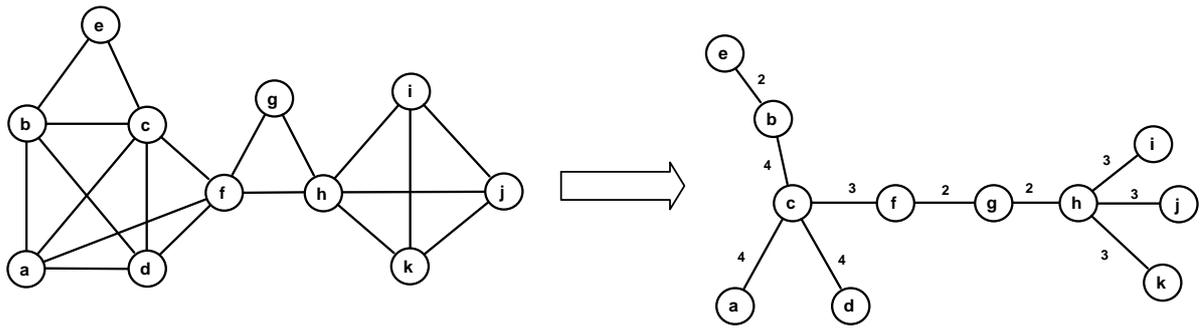


Figura 4: Árvore de corte do grafo da figura 3.

vértices  $c$  e  $j$  é igual a 2 pois a aresta de menor peso no caminho que liga os vértices  $c$  e  $j$  possui peso 2.

Dada uma árvore de corte  $T$  de um grafo  $G$ , os critérios de conectividades  $\#C(v)$  e  $MCC(v)$  podem ser calculados utilizando os seguintes lemas propostos em [9]:

**Lema 2** Para um grafo  $G$ , uma árvore de corte  $T$  de  $G$  e um vértice  $v$ , o valor do  $\#C(v)$  é igual ao valor da aresta de maior peso incidente a  $v$  em  $T$ .

**Prova:** Pela definição, a cardinalidade de um corte mínimo separando  $v$  de outro vértice  $u$  corresponde ao valor da aresta de menor peso no caminho de  $v$  até  $u$  em  $T$ . Este valor deve ser menor ou igual ao peso de uma aresta incidente a  $v$  em  $T$ . Então a cardinalidade máxima de qualquer corte mínimo separando  $v$  de outro vértice deve ser igual ao peso de alguma aresta de  $T$  incidente a  $v$ , mais precisamente, ao peso da aresta de maior valor [9]. ■

O critério de conectividade  $MCC(v)$  também podem ser calculado através da árvore de corte, utilizando o seguinte lema:

**Lema 3** Considere uma árvore de corte  $T$  de um grafo  $G$  e um vértice  $v$ . Aqueles vértices que são alcançáveis na árvore de corte a partir de  $v$ , utilizando somente arestas que possuem peso igual ou superior ao valor do  $\#C(v)$ , pertencem ao conjunto  $MCC(v)$  [9].

**Prova:** Considere um vértice  $v \in G$  e uma árvore de corte  $T$  de  $G$ . Após percorrer a árvore  $T$  a partir de  $v$  e atravessando arestas que possuem peso maior ou igual ao valor de  $\#C(v)$ , todo vértice não alcançado pode ser separado de  $v$  por um corte de tamanho menor que o valor do  $\#C(v)$ . O que significa que o subconjunto de vértices construído é máximo e é o maior subconjunto de vértices que pode ser encontrado. Desta forma, tal subconjunto de vértices é o conjunto  $MCC(v)$  [9]. ■

Como um exemplo do cálculo do conjunto  $MCC(v)$ , considere as figuras 4 e 3. Para o vértice  $b$  temos que  $\#C(b) = 4$ . Percorrendo a árvore de corte da figura 4 a partir de  $b$ , e utilizando somente arestas que possuem peso maior ou igual a 4, é possível alcançar os vértices  $c$ ,  $a$ , e  $d$ . Ou seja,  $MCC(b) = \{a, b, c, d\}$ .

O algoritmo de Gusfield, apresentado na figura 5, foi utilizado na geração da árvore de corte  $T$  de um grafo  $G$ .

A figura 6 apresenta um exemplo de geração da árvore de corte de um subgrafo do grafo apresentado na figura 3. Inicialmente todos os vértices da árvore possuem arestas

GeraÁrvoredeCorte (Grafo G)

- (1) Numerar seqüencialmente os vértices de G com os valores 1 a n;
- (2) Construir uma árvore orientada T, tal que  $p[v] = 1$  para  $v = 2, 3, \dots, n$ ;
- (3) Para  $i = 2$  até n faça:
- (4)     Seja  $j = p[i]$ ;
- (5)     Encontrar um corte mínimo entre os vértices  $i$  e  $j$  no grafo G;
- (6)     Para cada vértice  $x$  tal que  $p[x] = j$  (exceto  $i$ ), faça:
  - // para todos os vértices que apontam para  $j$
- (7)     Se o corte mínimo entre  $i$  e  $j$  não desconecta os vértices  $i$  e  $x$ , então  $p[x] = i$ ;
- (8)     O valor da aresta  $(i, j)$  recebe a cardinalidade do corte mínimo entre  $i$  e  $j$ ;
- (9) Retirar a orientação das arestas e retornar a árvore T.

Figura 5: Algoritmo de Gusfield utilizado na geração da árvore de corte.

orientadas para o vértice  $e$ , como pode ser observado na figura 6(b). Na primeira rodada do algoritmo, apresentada na figura 6(c), é encontrado um corte mínimo entre os vértices  $b$  e  $e$ . Os vértices  $a, c, d, f$  são alcançáveis por  $b$  após o corte. Desta forma, tais vértices passam a apontar para o vértice  $b$ . Como a cardinalidade do corte mínimo é igual a 2, a aresta  $(b, e)$  recebe peso 2. O algoritmo é executado para o restante dos vértices da árvore orientada. Ao final, o direcionamento das arestas é retirado e a árvore de corte é retornada, como apresentado na figura 6(h).

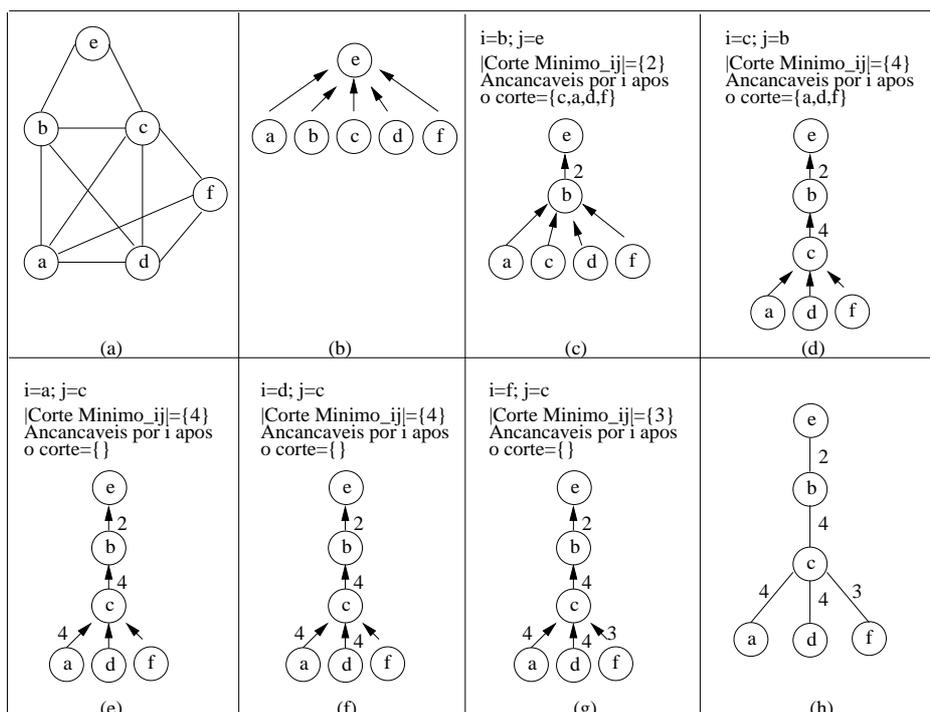


Figura 6: Um exemplo de geração de uma árvore de corte.

Uma simplificação do algoritmo de Ford-Fulkerson descrito em [13] foi utilizado para o cálculo do corte mínimo entre um par de vértices do grafo. O algoritmo de Ford-Fulkerson possui ordem de complexidade de  $\mathcal{O}(V \cdot E)$ , onde  $V$  é o número de vértices e  $E$  é o número de arestas do grafo. O algoritmo de Gusfield executa o algoritmo do corte mínimo entre um par de vértices  $V - 1$  vezes. Desta forma, a ordem de complexidade do algoritmo de Gusfield é dada por  $\mathcal{O}(V^2 \cdot E)$ . A partir da árvore de corte, os critérios de conectividade  $\#C(v)$  e  $MCC(v)$  são obtidos em tempo linear.

### 3.2. Uma Heurística para o Cálculo do $\#C(v)$

Para evitar a ordem de complexidade da construção da árvore de corte, uma heurística pode ser utilizada. Em cada passo, a heurística procura encontrar um conjunto de componentes 2-aresta conexos. Os vértices pertencentes a tais componentes têm a estimativa de seu  $\#C(v)$  incrementada em duas unidades. Já a estimativa do  $\#C(v)$  dos vértices não pertencentes a componentes 2-aresta conexos é incrementada em uma unidade.

O algoritmo da heurística utiliza uma árvore de busca em profundidade (*DFT* - *Depth First Tree*) de um grafo  $G$  e um conjunto de arestas chamadas arestas de cobertura (*cover edges*), que são arestas de retorno, para encontrar o conjunto de componentes 2-aresta conexos. Uma *aresta de cobertura* é uma aresta  $(v_k, v_0) \in G$  e  $(v_k, v_0) \notin DFT$ , tal que  $\langle v_0, v_1, \dots, v_k, v_0 \rangle$  é um ciclo em  $G$  e  $(v_{i-1}, v_i) \in DFT$ ,  $i = 1, \dots, k$ . Neste caso, é dito que a aresta de cobertura  $(v_k, v_0)$  cobre as arestas  $(v_{i-1}, v_i)$ ,  $i = 1, \dots, k$ .

As arestas de cobertura são encontradas utilizando a seguinte heurística: para cada aresta não coberta  $(u, v)$ , encontre a aresta de cobertura  $(x, y)$  que cobre a aresta  $(u, v)$  e que retorna o mais próximo possível da raiz da árvore. Desta forma, a aresta de cobertura tende a cobrir uma maior parte da árvore. Após a tentativa de cobertura de todas as arestas da *DFT*, o valor estimado  $\#C(v)$  é incrementado da seguinte maneira. Vértices  $v$  pertencentes a um ciclo no grafo formado pela  $DFT + \{\text{arestas de cobertura}\}$  fazem parte de um componente 2-aresta conexo e têm o valor estimado de seu  $\#C(v)$  incrementado de 2 unidades. Já vértices  $v$  que não pertencem a um ciclo no grafo formado pela  $DFT + \{\text{arestas de cobertura}\}$  têm o valor estimado de seu  $\#C(v)$  incrementado de 1 unidade. Após o incremento das estimativas do  $\#C(v)$ , as arestas da *DFT* e as arestas de cobertura são removidas de  $G$  e novas *DFT*'s são criadas. Quando o grau de um vértice atinge zero, o vértice também é removido de  $G$ . O algoritmo termina quando todas as arestas são removidas de  $G$ . O algoritmo utilizado no cálculo heurístico do  $\#C(v)$  é apresentado na figura 7.

```

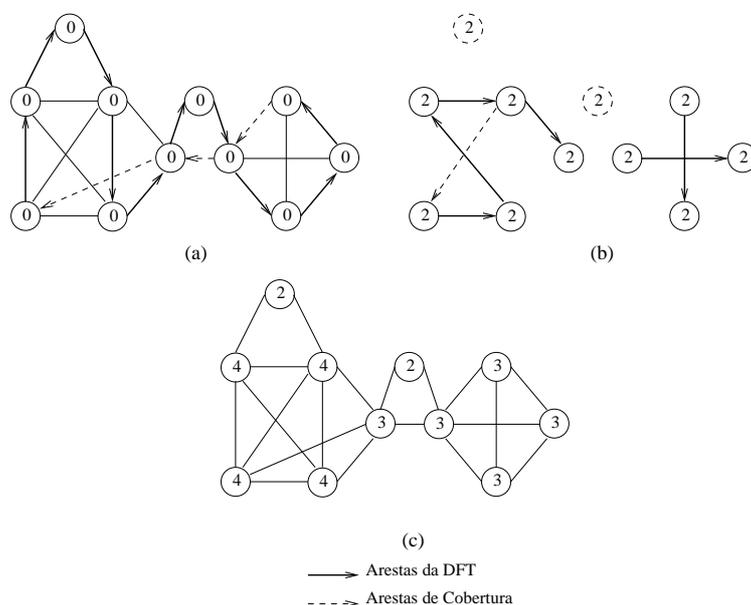
CalculoHeuristico#C(v)(Grafo G)
(1) Para cada vértice v em G faça:
(2)   #C[v] = 0;
(3)   Se grau(v) = 0 então remova v de G;
(5) Enquanto o conjunto de arestas não for vazio faça:
(6)   Para cada componente conexo H de G faça:
(7)     T = uma DFT de H;
(8)     Para cada aresta não coberta 'e' de T faça:
(9)       Encontre uma aresta de cobertura de 'e', se existir;
(10)    Para cada vértice v de H faça:
(11)      Se v pertence a um ciclo em (T + arestas de cobertura)
(12)        então #C[v] = #C[v] + 2;
(13)        senão #C[v] = #C[v] + 1;
(14)    Remova as arestas de T e as arestas de cobertura de G;
(15)    Remova os vértices v tais que grau(v)=0;
Retorne #C();

```

**Figura 7: Algoritmo utilizado no cálculo heurístico do  $\#C(v)$ .**

Um exemplo do cálculo do número de conectividade estimado  $\tilde{\#}C(v)$  dos vértices do grafo da figura 3, utilizando a heurística proposta, é apresentado na figura 8. Os vértices da figura estão rotulados com o valor de seu  $\tilde{\#}C(v)$ . Inicialmente, todos os vértices do grafo possuem  $\tilde{\#}C(v) = 0$ , como é apresentado na figura 8(a). O vértice no canto inferior esquerdo da figura 8(a) é a raiz da primeira *DFT*. As arestas pertencentes à *DFT* são representadas por arestas direcionadas, indicando o caminho per-

corrido pelo algoritmo de construção da  $DFT$ . As arestas de coberturas são representadas por arestas tracejadas e direcionadas, indicando arestas de retorno. Ao final da primeira iteração do algoritmo, todos os vértices da figura 8(a) pertencem a um ciclo em  $DFT + \{arestas\ de\ cobertura\}$  e portanto têm seu  $\tilde{\#}C(v)$  incrementado em duas unidades. As arestas da  $DFT$  e as arestas de cobertura são então removidas do grafo da figura 8(a), resultando no grafo da figura 8(b). Os vértices que possuem grau zero também são removidos do grafo, mas serão mantidos na figura 8(b) representados por círculos tracejados. Novas  $DFT$ 's e arestas de cobertura são encontradas nos componentes da figura 8(b). Todo vértice da figura 8(b) pertencente a um ciclo em  $DFT + \{arestas\ de\ cobertura\}$  tem seu  $\tilde{\#}C(v)$  incrementado em duas unidades. Caso contrário, o vértice tem seu  $\tilde{\#}C(v)$  incrementado em uma unidade. Todas as arestas da figura 8(b) são removidas e o algoritmo termina. A figura 8(c) apresenta os vértices do grafo rotulados com o valor de seu  $\tilde{\#}C(v)$  retornado pela heurística.



**Figura 8: Um exemplo do cálculo do número de conectividade estimado  $\tilde{\#}C(v)$  utilizando a heurística.**

A ordem de complexidade da heurística é de  $\mathcal{O}((E/V) \cdot (V + E))$ . Para grafos esparsos, temos que  $\mathcal{O}(E/V) = \mathcal{O}(1)$ . Grafos esparsos são mais próximos das topologias utilizadas na prática. Portanto, a complexidade esperada do algoritmo é linear e é dada por  $\mathcal{O}(V + E)$  [9].

O critério de conectividade  $MCC(v)$  não é calculado pela heurística. Porém, o  $MCC(v)$  é utilizado como critério de desempate, não sendo estritamente necessário na seleção do melhor desvio para um par de nodos.

### 3.3. Algoritmo para a Seleção do Melhor Desvio

A seguir, o algoritmo utilizado na seleção do melhor desvio para um par de vértices  $a$  e  $b$ , de acordo com os critérios de conectividade  $\#C(v)$  e  $MCC(v)$  é apresentado. As seguintes definições são utilizadas no algoritmo de seleção do melhor desvio:

$new - length(v, a, b)$ : indica o comprimento da rota alternativa do vértice  $a$  para o vértice  $b$ , usando o vértice  $v$  como desvio.

$d - neighborhood(a, b)$ : indica o conjunto de vértices  $v$ , tal que a rota alternativa do vértice  $a$  para o vértice  $b$ , usando o vértice  $v$  como desvio possui comprimento menor ou igual a  $d$ . Ou seja,  $d - neighborhood(a, b) = \{v \in V, v \neq a, b \mid new - length(v, a, b) \leq d\}$ .

O algoritmo procura por desvios dentro de uma vizinhança restrita. O tamanho da vizinhança é incrementado a cada iteração do algoritmo. O objetivo desta limitação é evitar que vértices pertencentes a componente altamente conexos, mas muito distantes dos vértices  $a$  e  $b$  sejam escolhidos como desvios. O incremento do tamanho da vizinhança é proporcional ao comprimento da rota regular existente entre os vértices  $a$  e  $b$ .

O número de conectividade  $\#C(v)$  é utilizado na seleção do melhor desvio. Empates são resolvidos utilizando o tamanho do conjunto  $MCC(v)$ , caso o algoritmo exato esteja sendo utilizado para cálculo do  $\#C(v)$  e  $MCC(v)$ , e utilizando o comprimento da rota alternativa entre  $a$  e  $b$  criada através do desvio. O algoritmo retorna uma lista ordenada por preferência dos melhores desvios para um par de vértices  $a$  e  $b$ . A lista contém todos os vértices do grafo.

```

    EncontreMelhorDesvio(Grafo G=(V,E), vértice a, vértice b)
(1) L <- Lista Vazia;
(2) Para cada vértice v de V faça:
(3)   Encontre o número  $\#C(v)$ , e o tamanho do componente  $MCC(v)$ ;
(4)   Encontre  $new-length(v, a, b)$ ;
(5) P <- rota IP regular usada pelo vértice a para comunicar-se com o vértice b;
(6) d <- |P| * 2;
(7) Enquanto |L| <= |V|-2 faça
(8)   Ordene os vértices v pertencentes à  $V-L$  e à  $d-neighborhood(a, b)$  em ordem
      não-crescente usando a chave  $\langle \#C(v), |MCC(v)|, -new-length(v, a, b) \rangle$ ;
(9)   Insira os vértices ordenados no final de L;
(10)  d = d + |P|; // a vizinhança é incrementada para alcançar mais vértices.
(11) Retorne L;

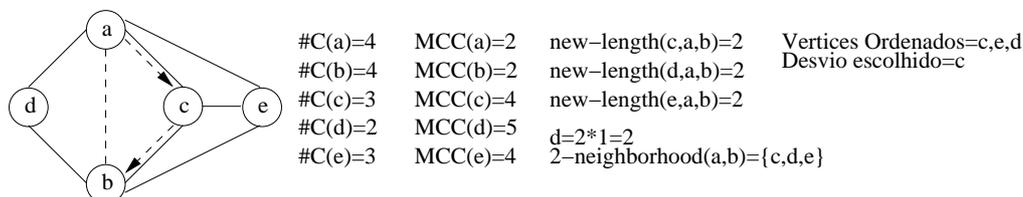
```

**Figura 9: Algoritmo utilizado na seleção do melhor desvio.**

O algoritmo utilizado na seleção do melhor desvio entre um par de vértices  $a$  e  $b$  de um grafo  $G = (V, E)$  é apresentado na figura 9. Inicialmente na linha (1) é criada uma lista vazia  $L$  que irá conter todos os vértices de  $G$  ordenados pelos critérios de conectividade propostos. Na linha (3), os valores  $\#C(v)$  e  $MCC(v)$  são calculados para cada vértice  $v \in V$ . Caso seja utilizado o algoritmo exato, o  $\#C(v)$  e o  $MCC(v)$  podem ser calculados. Caso seja utilizada a heurística, somente o  $\#C(v)$  pode ser calculado. Na linha (4) é realizado o cálculo do comprimento das rotas alternativas do vértice  $a$  para o vértice  $b$ , utilizando cada vértice de  $G$  como desvio. Na linha (5),  $p$  é a rota IP regular entre os vértices  $a$  e  $b$  e na linha (6), a variável  $d$  é inicializada com o dobro do comprimento da rota  $p$ . A variável  $d$  será utilizada para definir uma vizinhança incremental. O laço da linha (7) irá executar até que todos os vértices do grafo sejam ordenados. Na linha (8), os vértices pertencentes à  $d - neighborhood(a, b)$  que ainda não foram inseridos em  $L$  são ordenados inicialmente considerando o seu número de conectividade  $\#C(v)$ , depois o tamanho de seu conjunto  $MCC(v)$ , caso o  $MCC(v)$  tenha sido calculado, e finalmente o comprimento da rota alternativa. Tais vértices são inseridos no final da lista  $L$  na linha (9). Na linha 10, a vizinhança é incrementada para alcançar mais possíveis desvios.

A figura 10 apresenta um exemplo da seleção do melhor desvio para os vértices  $a$  e  $b$ . Os valores do  $\#C(v)$  e  $MCC(v)$  de cada vértice do grafo são apresentados na figura. O comprimento da rota alternativa do vértice  $a$  para o vértice  $b$ , usando o vértice  $c$ ,  $d$  ou

$e$  como desvio, é igual a 2. Como o comprimento da rota IP regular de  $a$  para  $b$  é igual a 1, a variável  $d$  é igual a 2 e  $2 - neighborhood(a, b) = \{c, d, e\}$ . Ordenando os vértices  $\{c, d, e\}$  pelo  $\#C(v)$ ,  $MCC(v)$  e depois pelo comprimento da rota alternativa, temos que o melhor desvio é o vértice  $c$ , que possui  $\#C(v)=3$ .



**Figura 10: Um exemplo de seleção do melhor desvio.**

Caso a árvore de corte seja utilizada no cálculo dos critérios de conectividade  $\#C(v)$  e  $MCC(v)$  a ordem de complexidade do algoritmo é igual a  $\mathcal{O}(V^2 \cdot E) + \mathcal{O}(E \cdot \log(V))$ . Já a ordem de complexidade do algoritmo completo quando a heurística é utilizada é de  $\mathcal{O}(V + E) + \mathcal{O}(E \cdot \log(V))$ , assumindo que a rede é esparsa.

#### 4. Resultados Experimentais

Os resultados experimentais foram obtidos através da implementação dos algoritmos descritos em linguagem Java, utilizando classes do projeto JDigraph [14], que implementam várias operações em grafos. Utilizando o método Waxman [15], foram gerados aleatoriamente grafos com 10, 20, 30, 40, 50, 60, 70 e 100 nodos. Para cada tamanho de grafo, foram gerados 250 grafos com grau médio dos nodos variando de 3 a 8. No total, foram gerados aleatoriamente 2000 grafos. O método Waxman foi o escolhido pois o mesmo leva em consideração a distância entre dois vértices no cálculo da probabilidade de existência de uma aresta entre os mesmos, propondo-se desta forma a capturar a característica de localidade existente nas redes de computadores reais, em especial a Internet.

A cobertura de falhas obtida pela utilização do algoritmo de seleção de desvios em um grafo específico foi calculada da seguinte maneira. Inicialmente foi obtido o caminho mínimo entre cada par de nodos do grafo, com o intuito de modelar uma rede real, onde os nodos comunicam-se através do menor caminho.

Para cada par de nodos  $A$  e  $B$  do grafo, os seguintes procedimentos foram executados:

- Obteve-se a lista ordenada dos melhores desvios tendo o nodo  $A$  como origem e o nodo  $B$  como destino;
- Para cada aresta  $(x, y)$  que não é de corte pertencente ao caminho entre  $A$  e  $B$ , os seguintes procedimentos foram executados:
  - A aresta  $(x, y)$  é marcada como falha e o contador de falhas na rede é incrementado;
  - Verifica-se se a rota alternativa através do melhor desvio  $D1$  contém a aresta  $(x, y)$ . Caso a rota alternativa não contenha a aresta  $(x, y)$ , significa que a mesma é funcional e o contador das falhas cobertas pelo melhor desvio é incrementado. A rota alternativa é composta pela rota regular entre  $A$  e  $D1$  e pela rota regular entre  $D1$  e  $B$ ;

- Repete-se o mesmo procedimento para o segundo melhor desvio;
- Repete-se o mesmo procedimento para o terceiro melhor desvio;

A falha de uma aresta de corte não foi realizada, pois não existiria outro caminho no grafo entre os nodos  $A$  e  $B$ . Nestes casos, a cobertura de falhas utilizando qualquer desvio é impossível.

Após a execução dos procedimentos acima descritos para cada par de nodos do grafo, a cobertura de falhas utilizando o melhor desvio é dada por:

$$\frac{\text{falhas recuperadas pelo melhor desvio}}{\text{falhas na rede}}$$

O cálculo da cobertura de falhas utilizando o segundo e terceiro melhores desvios é análogo.

Além do cálculo da cobertura de falhas utilizando o melhor desvio, os cálculos da cobertura de falhas utilizando o segundo e terceiro melhores desvios foram realizados com o intuito de confirmar se realmente o algoritmo implementado retorna a ordem correta dos nodos que possuem a melhor capacidade de desviar de uma falha na rede. Para cada grafo analisado, também foi calculada a diferença entre o  $\#C(v)$  médio retornado pelo algoritmo exato e o  $\#C(v)$  médio retornado pela heurística, para verificar se a heurística retorna valores próximos aos valores retornados pelo algoritmo exato do  $\#C(v)$ . Os nodos selecionados para desvios utilizando árvores de corte serão chamados *desvios exatos*. Analogamente, os nodos selecionados para desvios utilizando a heurística serão chamados *desvios heurísticos*.

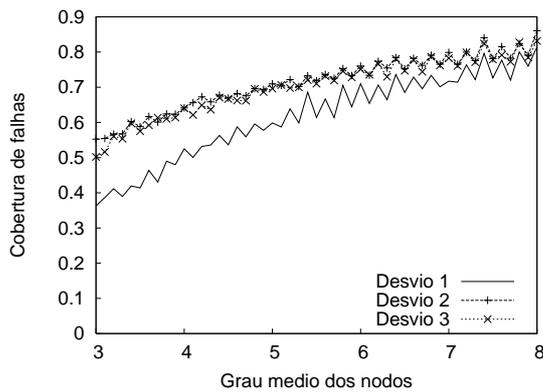
#### 4.1. Avaliação dos Valores de $\#C(v)$ Retornados pela Heurística

A quantidade de arestas de um grafo aumenta com o crescimento do grau médio dos nodos. Desta forma, o valor médio do  $\#C(v)$  exato tende a aumentar com o crescimento do grau médio dos nodos do grafo. Portanto, para uma melhor comparação da variação da diferença média entre o  $\#C(v)$  médio exato e o  $\#C(v)$  médio heurístico, de acordo com a variação do grau médio dos nodos do grafo, as diferenças obtidas foram normalizadas utilizando o grau médio dos nodos. Os resultados da normalização estão descritos na tabela 1.

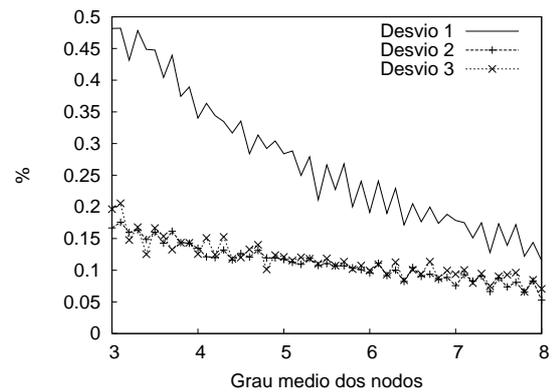
Num. Nodos	Diferença / grau médio dos nodos					Geral
	Grau médio dos nodos no intervalo					
	[3, 4)	[4, 5)	[5, 6)	[6, 7)	[7, 8]	
10	0.153	0.166	0.163	0.163	0.154	0.160
20	0.186	0.199	0.189	0.179	0.170	0.185
30	0.207	0.206	0.197	0.184	0.184	0.196
40	0.200	0.207	0.205	0.195	0.190	0.200
50	0.207	0.211	0.206	0.201	0.192	0.204
60	0.205	0.214	0.210	0.196	0.197	0.204
70	0.207	0.210	0.212	0.198	0.196	0.205
100	0.199	0.216	0.218	0.202	0.194	0.206

**Tabela 1: Diferença média entre o  $\#C(v)$  médio exato e o  $\#C(v)$  médio heurístico, normalizada pelo grau médio dos nodos.**

Analisando isoladamente cada linha da tabela 1, após a normalização não se observa um crescimento da diferença média entre o  $\#C(v)$  médio exato e o  $\#C(v)$



**Figura 11: Cobertura de falhas obtida pelos 3 melhores desvios exatos.**



**Figura 12: Percentagem de vezes em que os melhores desvios exatos pertencem ao caminho utilizado pelos nodos comunicantes.**

médio heurístico de acordo com o crescimento do grau médio dos nodos. Observa-se que as diferenças médias entre o  $\#C(v)$  exato e o  $\#C(v)$  heurístico foram inferiores a  $0.218 * \text{grau médio dos nodos}$ . Ou seja, pode-se concluir que a heurística cumpre sua função em retornar valores de  $\#C(v)$  próximos aos valores retornados pelo algoritmo exato.

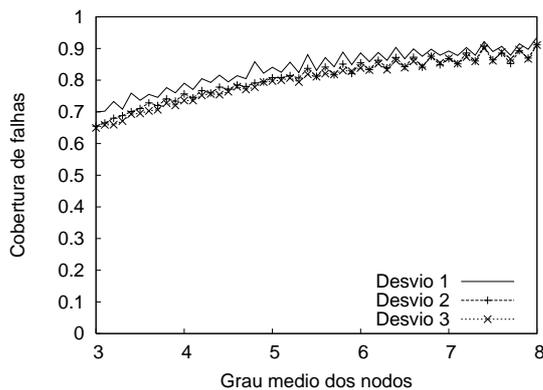
## 4.2. Avaliação da Cobertura de Falhas

Esta subseção apresenta os resultados da cobertura de falhas obtida por desvios selecionados a partir da estratégia proposta de seleção de desvios.

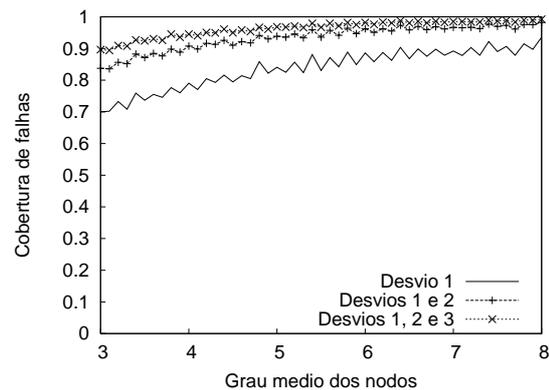
### 4.2.1. Cobertura de Falhas Obtida por Desvios Exatos e por Desvios Heurísticos

A figura 11 apresenta graficamente os resultados da cobertura de falhas obtida pelos 3 melhores desvios exatos em todos os grafos analisados. Observa-se que a cobertura de falhas obtida pelo melhor desvio exato é inferior à cobertura de falhas obtida pelo segundo e pelo terceiro melhores desvios exatos.

Caso o nodo escolhido como desvio pertença ao caminho que o nodo origem utiliza para alcançar o nodo destino, existe a possibilidade da rota alternativa criada através do desvio ser idêntica à rota falha, não sendo possível recuperar a falha da rede. A cobertura de falhas obtida pelo melhor desvio exato é pior que a cobertura de falhas obtida pelo segundo melhor desvio exato, como pode ser observado na figura 11. Tal resultado deve-se ao fato do percentual de vezes em que o melhor desvio exato pertence ao caminho utilizado pelos nodos comunicantes ser maior que o percentual de vezes em que o segundo melhor desvio exato pertence ao caminho utilizado pelos nodos comunicantes, como pode ser observado no gráfico da figura 12. Na figura 12, observa-se também que a diferença entre o percentual de vezes em que o melhor desvio exato e o percentual de vezes em que o segundo melhor desvio exato pertencem ao caminho utilizado pelos nodos comunicantes diminui com o aumento do grau médio dos nodos do grafo. Tal resultado



**Figura 13: Cobertura de falhas obtida pelos 3 melhores desvios exatos e não pertencentes ao caminho.**



**Figura 14: Cobertura de falhas acumulada obtida pelos 3 melhores desvios exatos não pertencentes ao caminho.**

explica o fato da diferença entre a cobertura de falhas alcançada pelo segundo melhor desvio exato e a cobertura de falhas alcançada pelo melhor desvio exato diminuir com o aumento do grau médio dos nodos, como pode ser observado figura 11.

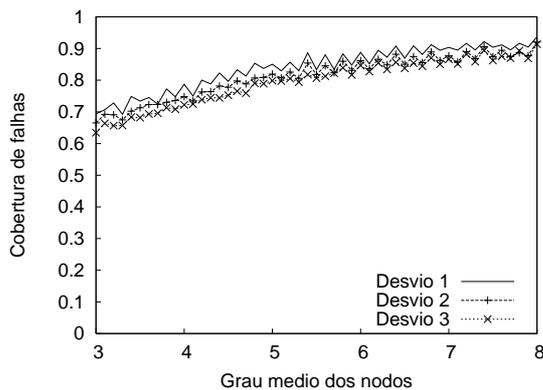
Conclui-se que a estratégia de seleção de desvios baseando-se apenas em critérios de conectividade não obteve o resultado esperado. Foi mostrado que o melhor desvio, ou seja, o nodo que apresenta a maior conectividade, obteve uma cobertura de falhas inferior à cobertura de falhas alcançada pelo segundo melhor desvio, que possui uma menor conectividade.

#### 4.2.2. Cobertura de Falhas Obtida por Desvios Exatos não Pertencentes ao Caminho

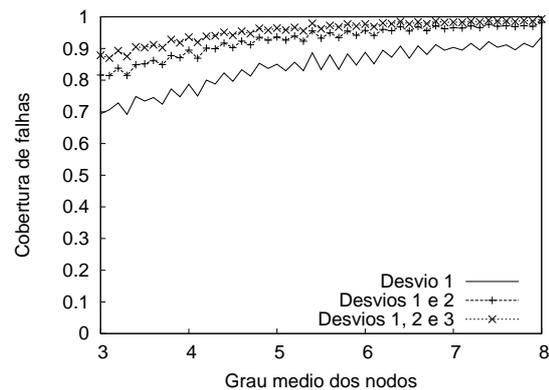
A figura 13 apresenta graficamente os resultados da cobertura de falhas obtida pelos 3 melhores desvios exatos não pertencentes ao caminho. Pode-se observar que a cobertura de falhas obtida pelo melhor desvio é superior à cobertura de falhas obtida pelo segundo melhor desvio, que é superior à cobertura de falhas obtida pelo terceiro melhor desvio.

Comparando os resultados da cobertura de falhas obtida pelos melhores desvios exatos não pertencentes ao caminho (figura 13), com os resultados da cobertura de falhas obtida pelos melhores desvios exatos (figura 11), constata-se que a cobertura de falhas obtida pela utilização dos melhores desvios exatos não pertencentes ao caminho é superior à cobertura de falhas obtida pela utilização dos melhores desvios exatos. Nos experimentos realizados, a percentagem de vezes em que as rotas criadas através de desvios pertencentes ao caminho foram idênticas às rotas falhas foi de 98.7%. Isto explica o fato da seleção de desvios, sem verificar se os mesmos pertencem ou não ao caminho que o nodo origem utiliza para alcançar o nodo destino, levar a resultados piores de cobertura de falhas.

O gráfico apresentado na figura 13, mostra que a cobertura de falhas obtida pelo melhor desvio exato não pertencente ao caminho em grafos que possuem grau médio dos nodos igual a 3 é de 70%. Ou seja, mesmo para grafos com número reduzido de arestas, a cobertura de falhas alcançou um valor alto. Observa-se que, conforme o esperado, a co-



**Figura 15: Cobertura de falhas obtida pelos 3 melhores desvios heurísticos não pertencentes ao caminho.**



**Figura 16: Cobertura de falhas acumulada obtida pelos 3 melhores desvios heurísticos não pertencentes ao caminho.**

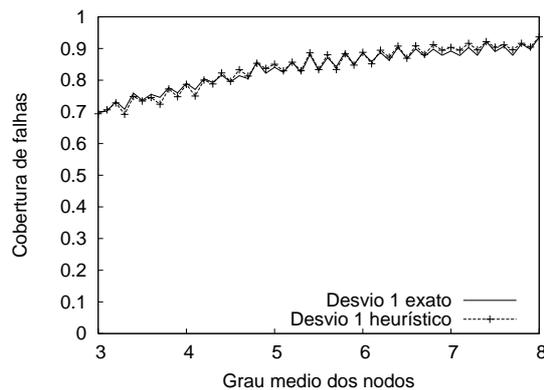
bertura de falhas aumenta de acordo com o aumento no grau médio dos nodos, chegando a 90% em grafos que possuem grau médio dos nodos igual a 8. A figura 14, apresenta a cobertura de falhas acumulada obtida pelo 3 melhores desvios exatos não pertencentes ao caminho. Observa-se que a cobertura de falhas em grafos que possuem grau médio dos nodos igual a 3 salta para 81%, quando são utilizados o melhor e o segundo melhor desvio exato não pertencente ao caminho, e para 87%, quando são utilizados o melhor, o segundo melhor e o terceiro melhor desvio exato não pertencente ao caminho. Além disso, quando são utilizados os três melhores desvios, a cobertura de falhas atinge valores superiores a 95%, nos casos em que o grau médio dos nodos é superior a 4.4, e atinge valores superiores a 98%, nos casos em que o grau médio dos nodos é superior a 6.8.

Conclui-se que a estratégia de seleção de desvios baseada nos critérios de conectividade propostos e que não pertencem ao caminho utilizado pelos nodos comunicantes obteve os resultados esperados de cobertura de falhas.

#### **4.2.3. Cobertura de Falhas Obtida por Desvios Heurísticos não Pertencentes ao Caminho**

A figura 15 apresenta o gráfico da cobertura de falhas obtida pelos 3 melhores desvios heurísticos não pertencentes ao caminho. Já a figura 16 apresenta o gráfico da cobertura de falhas acumulada obtida pelos 3 melhores desvios heurísticos não pertencentes ao caminho. Observa-se que a cobertura de falhas obtida pelos desvios heurísticos não pertencentes ao caminho é similar à cobertura de falhas obtida pelos desvios exatos não pertencentes ao caminho, apresentados nas figuras 13 e 15. A similaridade também pode ser observada na figura 17, que apresenta o gráfico da cobertura de falhas obtida pelo melhor desvio exato não pertencente ao caminho e pelo melhor desvio heurístico não pertencente ao caminho. Observa-se que a cobertura de falhas obtida pelo melhor desvio heurístico é praticamente idêntica à cobertura de falhas obtida pelo melhor desvio exato.

Conclui-se que não existem impeditivos em utilizar a heurística para a seleção dos melhores desvios, visto que os desvios selecionados pela heurística obtiveram cobertura



**Figura 17: Cobertura de falhas obtida pelo melhor desvio exato e pelo melhor desvio heurístico não pertencentes ao caminho.**

de falhas ligeiramente superior à cobertura de falhas obtida pelos desvios retornados pelo algoritmo exato.

## 5. Conclusão

Este trabalho apresentou o uso de rotas alternativas de alta conectividade, com o objetivo de permitir que os nodos da rede continuem a se comunicar, mesmo durante o período compreendido pela latência de convergência do protocolo de roteamento. As rotas alternativas são criadas através de nodos chamados desvios, que pertencem a componentes de alta conectividade, possuindo maior probabilidade de desviar de uma falha da rede, mesmo sem a informação da localização da mesma. Além do algoritmo exato para cálculo dos critérios de conectividade  $\#C(v)$  e  $MCC(v)$  uma heurística eficiente para cálculo do critério  $\#C(v)$  também foi apresentada. Os resultados experimentais demonstram que a heurística obtém valores de  $\#C(v)$  próximos aos valores obtidos pelo algoritmo exato. A cobertura de falhas obtida pelo melhor desvio não pertencente ao caminho pode chegar a 90%. Quando os três melhores desvios não pertencentes ao caminho são utilizados, a cobertura de falhas pode ser superior a 98%. A cobertura de falhas obtida pelos melhores desvios heurísticos não pertencentes ao caminho apresentou valores similares. Estes resultados visam comprovar que rotas alternativas criadas através de desvios selecionados a partir dos critérios de conectividade propostos possuem boa probabilidade de desviar de uma falha ocorrida na rede, mesmo sem a informação da localização da falha. Trabalhos futuros incluem a especificação de um protocolo de roteamento que seleciona a melhor rota para um determinado destino baseando-se nos critérios de conectividade propostos, e sua implementação em uma rede real são também metas para desenvolvimento futuro.

## Referências

- [1] Y. Rekhter. A Border Gateway Protocol 4 (BGP-4), 1995. RFC-1771.
- [2] Christian Huitema. *Routing in the Internet*. Prentice Hall PTR, second edition, 1999.
- [3] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed Internet Routing Convergence. In *SIGCOMM*, pages 175–187, 2000.

- [4] Craig Labovitz, Abha Ahuja, Roger Wattenhofer, and Venkatachary Srinivasan. The Impact of Internet Policy and Topology on Delayed Routing Convergence. In *INFOCOM*, pages 537–546, 2001.
- [5] Timothy Griffin and Gordon T. Wilfong. An Analysis of BGP Convergence Properties. In *SIGCOMM*, pages 277–288, 1999.
- [6] Kannan Varadhan, Ramesh Govindan, and Deborah Estrin. Persistent Route Oscillations in Inter-Domain Routing. *Computer Networks (Amsterdam, Netherlands: 1999)*, 32(1):1–16, 2000.
- [7] Elias Procópio Duarte, G. Mansfield, S. Noguchi, and M. Miyazaki. Fault-Tolerant Network Management. In *ISACC'94*, 1994. Monterrey, Mexico.
- [8] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan. Detour: Informed Internet Routing and Transport. *IEEE Micro*, 1999.
- [9] Jaime Cohen and Elias Procópio Duarte. Fault-Tolerant Routing of TCP/IP PDU's on General Topology Backbones. In *Third International Workshop on Design of Reliable Communication Networks*, 2001.
- [10] F. Harary. *Graph Theory*. Perseus Books Publishing, 1969.
- [11] R. E. Gomory and T. C. Hu. Multi-Terminal Network Flows. *SIAM Journal on Applied Mathematics*, pages 9:551–556, 1961.
- [12] D. Gusfield. Very Simple Method for All Pairs Network Flow Analysis. *SIAM Journal on Computing*, pages 19(1):143–155, 1990.
- [13] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. McGraw-Hill, second edition, 1990.
- [14] D. Walend. JDigraph, 2003. <http://jdigraph.sourceforge.net>, acessado em 20/04/2003.
- [15] Bernard M. Waxman. Routing of Multipoint Connections. In *IEEE Journal of Selected Areas in Communications*/6(9), pages 1617–1622, 1988.