Contornando Falhas em Backbones IP com Caminhos Emergenciais Rápidos

Fernando Barreto[1], Emílio C. G. Wille[1], Luiz Nacamura Junior[1]

[1]Pós-graduação em Engenharia Elétrica e Informática Industrial
Universidade Tecnológica Federal do Paraná (UTFPR)
80.230-901 – Curitiba – PR – Brazil

{fbarreto,ewille}@cpgei.cefetpr.br, nacamura@dainf.cefetpr.br

Abstract. *In general, the routing protocols of high speed backbones, when facing a failure, are not able to obtain a new route in due time. They take from hundreds of milliseconds until tens of seconds in order to converge. This period generates instability, causing erroneous forwarding processes and high rates of packet loss. This study proposes a proactive calculation approach of fast emergency paths aiming to aid routing protocols to bypass failures. An evaluation of these paths was conducted on a representation of a real topology. Results shows the paths are shorter than other approaches and needs a small quantity of extra information added to the forwarding base per router.*

Resumo. *Os protocolos de roteamento, em caso de falhas em backbones mais velozes, geralmente não obtêm nova rota em tempo hábil. Eles precisam desde centenas de milissegundos até vários segundos para convergir. Durante esse tempo, a rede fica instável, realizando um processo de encaminhamento errado, com altas taxas de pacotes perdidos. Este trabalho propõe uma abordagem de cálculo pró-ativo de caminhos emergenciais rápidos para auxiliar os protocolos de roteamento a contornar falhas. Uma avaliação desses caminhos foi realizada em uma representação de topologia real. Nesta avaliação os caminhos apresentam-se menores que as outras abordagens e ocupam menos informações extras adicionadas na base de encaminhamento por roteador.*

1. Introdução

Uma rede IP robusta e confiável é aquela que consegue comportar fluxos de tráfego passantes mesmo na existência de falhas que alteram a infra-estrutura de rede. Na ocorrência de uma falha, a infra-estrutura de rede deve ser capaz de manter uma conectividade suficiente para comportar os tráfegos comprometidos, procurando influenciar o mínimo possível no desempenho desses tráfegos.

Infelizmente, os problemas de falhas em componentes de rede são comumente encontrados no dia a dia. Falhas de enlace ou de roteador (nó) são resultantes de vários eventos como manutenção de software nos nós, falhas nas interfaces de rede dos equipamentos, e ruptura acidental de fibra. A ocorrência desse tipo de falha compromete a estabilidade de parte das rotas existentes, uma vez que ainda encaminham pacotes para o componente falho. Para adequar essas rotas à nova configuração da topologia, há uma reação dos protocolos de roteamento.

Nas redes IP atuais, as falhas de componentes de rede dentro de um sistema autônomo (AS) são tratadas pelos protocolos de roteamento *Interior Gateway Protocol* (IGP) do tipo estado do enlace, como o *Open Shortest Path First* (OSPF) [Moy 1998]. Na presença de falha detectada por um nó em seu componente adjacente, existe primeiramente um período de contenção para possibilitar uma correção por parte da camada física (podendo alcançar em torno de dezenas de milissegundos em redes SONET/SDH atuais [Francois et al. 2005]). Caso esse período de contenção seja ultrapassado, as rotinas de roteamento desse nó comunicam a mudança ocorrida no estado do enlace a todos os outros nós do mesmo AS através de pacotes *Link State Advertisement* (LSA). Isso faz com que todos computem novamente suas árvores de menores caminhos utilizando o algoritmo *Shortest Path First* (SPF) [Dijkstra 1959] e atualizem a *Routing Information Base* (RIB) e a *Forwading Information Base* (FIB). A RIB armazena a tabela de rotas para todos os prefixos de redes divulgados dentro do AS, sendo preenchida pelas rotinas de roteamento. A FIB armazena informações sintetizadas da RIB e são utilizadas pelas rotinas de encaminhamento. O armazenamento da FIB fica diretamente na memória da interface de rede ou, dependendo da arquitetura do roteador, em uma memória comum de acesso a todas as interfaces de rede.

Esse processo reativo de adequação a uma falha, que engloba todos os nós da rede, necessita de um tempo de execução denominado período de convergência. Durante esse período, informações erradas na RIB e FIB no nó detector de falha ocasionam instabilidades na rede que resultam em perdas de pacotes por falta de rotas corretas (pacotes são descartados enquanto não há uma atualização da RIB e FIB), ou por *loops* de roteamento gerados pela não sincronização das RIBs e FIBs nos demais nós.

Uma simulação do período de convergência foi realizada em [Francois et al. 2005] para algumas redes atuais, estimando que, em um melhor caso com *hardware* ideais e configurações otimizadas no tempo de execução das rotinas de roteamento, pode-se atingir um período na ordem de centenas de milisegundos. Outros trabalhos, que utilizam uma configuração padrão, revelam um período na ordem de dezenas de segundos [Paxson 1996] [Alaettinoglu et al. 2000] [Iannaccone et al. 2004].

Mesmo sendo possível reduzir o período de convergência para menos de 1 segundo, esse tempo tende a ser insuficiente, pois conforme a velocidade dos enlaces aumenta maior se torna a taxa de perdas durante esse período. Se um enlace OC-192 (9.6 Gbps) estiver falho por 500 milisegundos, assumindo um tamanho de pacotes de 1024 bytes, são descartados aproximadamente 500 mil pacotes por falta de uma rota atualizada, sem considerar ainda as perdas por *loops* de roteamento. Essa alta taxa de perdas compromete o desempenho de várias aplicações, principalmente VoIP e vídeo.

Outro dado importante sobre as falhas revela que em torno de 50% delas são de característica transitória e com curta duração de tempo: < 1 minuto [Iannaccone et al. 2004] [Markopoulou et al. 2004]. Uma mudança de estado muito rápida gerada pela falha transitória é geralmente evitada com temporizadores em *hardware* (em torno de 10 segundos [Francois et al. 2005]), ou em *software* (em torno de 12 segundos [Markopoulou et al. 2004]). Porém, uma falha transitória pode ocasionar dois ou mais períodos de convergência caso a duração da falha seja maior que os temporizadores, o que gera períodos de instabilidade próximos um do outro aumentando a taxa de perdas.

Esses períodos de instabilidade podem ser amenizados com a utilização de

caminhos de recuperação durante o período de convergência para contornar uma falha, sendo esse o objetivo do artigo. Recentemente, várias abordagens têm surgido para gerar esses caminhos de recuperação, porém algumas questões foram também adicionadas. Dentre elas estão o excesso de informações extras incluídas na FIB para representar um caminho de recuperação e a longa extensão percorrida pelos pacotes quando desviados.

Este trabalho apresenta uma abordagem pró-ativa de cálculo de caminhos de recuperação, que faz parte de um esquema em desenvolvimento para prover reroteamento rápido, local e distribuído: *Esquema de Caminhos Emergenciais Rápidos* (E-CER). A abordagem, denominada CER_pró-ativa fornece caminhos de recuperação na forma de *caminhos emergenciais rápidos* (CER), que são armazenados diretamente na FIB para estarem prontamente disponíveis para uso. Cada CER é classificado em níveis que minimizam o custo e o número de nós dos caminhos utilizados para desviar um tráfego afetado por uma falha, sendo um fator diferencial sobre as demais abordagens pró-ativas relacionadas.

2. Trabalhos Relacionados

As abordagens de caminhos de recuperação podem ser pró-ativas ou reativas. As abordagens pró-ativas fornecem esses caminhos antecipadamente a uma falha, o que possibilita uma correção imediata de falha (principalmente as transitórias) sem necessitar informar imediatamente os demais nós sobre a alteração do estado do enlace [Shand and Bryant 2006]. Essas abordagens pró-ativas são nomeadas abordagens de *reroteamento rápido IP* (IPFRR). Já as abordagens reativas calculam os caminhos após a ocorrência de falha, o que eleva a taxa de pacotes perdidos durante esse cálculo comparado com as abordagens pró-ativas e por isso não sendo interessantes para uma correção rápida de falhas.

O trabalho de [Nucci et al. 2003] utiliza a Busca Tabu para encontrar a distribuição dos pesos aos enlaces de forma a possibilitar o contorno de qualquer falha de enlace utilizando *Equal Cost MultiPath* (ECMP), porém essa abordagem suporta apenas falha de enlace não considerando falha de nó. Os trabalhos de [Zhong et al. 2005], [Atlas and Zinim 2006], [Atlas 2006], [Bryant et al. 2005], [Bryant et al. 2006] e [Kvalbein et al. 2006] suportam tanto falha de enlace quanto de nó. O trabalho de [Zhong et al. 2005] descreve a abordagem *Failure Insensitive Routing Fast Rerouting* (FIFR), que utiliza várias FIBs por interface de rede que são preenchidas antecipadamente pela dedução de quais enlaces podem estar com falha quando um pacote chega por uma interface em que normalmente não chegaria. Porém esse trabalho não permite contornar a falha do último enlace antes do nó destino. O trabalho de [Atlas and Zinim 2006] descreve a abordagem *Loop Free Alternates* (LFA) que fornece um caminho de recuperação simples e com menor uso de recursos baseado em cálculos de distância a partir dos nós vizinhos, porém ela por si só não alcança 100% de cobertura de falhas por ser diretamente dependente da estrutura da topologia. O trabalho de [Atlas 2006] define o *U-Turn* que estende o LFA para considerar os nós adjacentes ao nó vizinho para obter um caminho de recuperação, porém da mesma forma que a abordagem LFA pura não atinge 100% de cobertura de falhas, é dependente da estrutura da topologia e não funciona para topologias em anel. O trabalho de [Bryant et al. 2005] fornece um mecanismo de encapsulamento (*Tunnels*) para desviar os pacotes até um nó capaz de encaminhar os pacotes pelo menor caminho contornando a falha, porém, se-

gundo os próprios autores, não suporta custos de enlaces assimétricos e não é interativo com as demais abordagens IPFRR. Já os trabalhos de [Bryant et al. 2006] e [Kvalbein et al. 2006] conseguem atingir 100% de cobertura de falhas [Hansen et al. 2006].

O trabalho de [Bryant et al. 2006] descreve o NotVia, que consiste na utilização de caminhos pré-calculados a partir de rotas para endereços *not-via*. Cada nó estabelece um endereço *not-via* para representar um componente de rede adjacente a ser contornado, i.e. um enlace ou o nó específico. Esses endereços são divulgados para que todos gerem rotas para cada endereço *not-via* retirando o componente de rede respectivo a esse endereço da topologia. Isso possibilita a criação de um caminho de recuperação para contornar o respectivo componente ao endereço *not-via* considerado. O desvio dos pacotes afetados por uma falha torna-se possível através do encapsulamento dos pacotes com o endereço *not-via*. A escolha de qual endereço *not-via* que será utilizado depende do endereço de destino dos pacotes IP. Caso o endereço destino esteja no nó adjacente, escolhe-se o endereço *not*-via para contornar apenas o enlace adjacente. Caso o endereço destino esteja em um nó após o nó adjacente, deve-se identificar qual o *next-next-hop* que os pacotes devem seguir, para então obter o correto endereço *not-via* situado após o nó adjacente a ser utilizado. Os endereços *not-via* e *next-next-hops* para cada endereço de destino devem estar previamente armazenados na FIB, e dependendo da extensão da topologia, esses dados podem acarretar em problemas de escalabilidade. Além desse problema, alguns caminhos obtidos podem ser mais extensos que o necessário devido à localização estática dos endereços *not-via*. O uso de encapsulamento pode atingir o limite da *Maximum Transmission Unit* dos pacotes, o que gera uma carga adicional de fragmentação (no caso do IPv4) ou um aumento da taxa de perdas com descarte de pacotes (no caso do IPv6) [Deering and Hinden 1998].

Já [Kvalbein et al. 2006] descreve o *Multiple Routing Configurations* (MRC), que consiste da geração de várias sub-topologias a partir da topologia original. Uma heurística isola componentes de rede (enlace ou nó) por sub-topologia, procurando manter total conectividade entre todos os nós. O objetivo é obter um caminho de recuperação em uma sub-topologia que não utilize um componente quando este estiver falho. A heurística possibilita uma política para isolar até um componente de rede por sub-topologia, gera caminhos de recuperação com extensão ideal por considerar apenas um componente de rede como falho. Porém, como cada sub-topologia deve possuir uma RIB e FIB, seriam geradas muitas sub-topologias para cobrir toda a topologia o que prejudica a escalabilidade do MRC. Essa abordagem, então, procura reduzir a quantidade de sub-topologias geradas isolando vários componentes por sub-topologia. Conseqüentemente observa-se que os caminhos de recuperação ficam mais extensos que o necessário para contornar uma falha, já que devem contornar mais componentes por sub-topologia. O MRC utiliza uma representação para cada sub-topologia existente, sendo utilizada como um identificador a ser marcado nos pacotes desviados. A marcação dos pacotes é realizada pelo nó detector de falha ao identificar qual sub-topologia é capaz de isolar essa falha, o que instrui os demais nós a encaminhar os pacotes utilizando a FIB respectiva a essa sub-topologia escolhida. O fato de utilizar marcação de pacotes evita o problema do encapsulamento que existe na abordagem NotVia. O MRC estipula que entre 3 a 5 sub-topologias sejam suficientes para cobrir toda uma topologia, o que representa uma economia de recursos na RIB e FIB.

As abordagens pró-ativas são semelhantes à abordagem *Fast Reroute* disponível

na tecnologia MPLS: *one-to-one backup* e *facility backup* [Pan et al. 2005]. Porém, este trabalho está voltado para redes puramente IP e maiores detalhes das abordagens na tecnologia MPLS são omitidas deste texto.

3. Proposta: CER_pró-ativa

A CER_pró-ativa é uma abordagem para geração de caminhos de recuperação tomando como base o número de nós e a métrica já existente do OSPF: menor soma dos custos dos enlaces do nó origem até os nós destino. O uso dessa métrica torna possível reutilizar as rotas já existentes do OSPF o que diminui a complexidade dessa abordagem. A abordagem proposta assemelha-se em alguns aspectos com a [Bryant et al. 2005], porém permite interação com as demais abordagens de IPFRR (no caso a abordagem ECMP e LFA), independe dos custos dos enlaces serem assimétricos ou não, alcança 100% de cobertura de falhas e não utiliza a técnica de encapsulamento. Cada caminho de recuperação disponibilizado para desviar os pacotes é definido com um ***caminho emergencial rápido*** (CER), que é planejado para ser antecipadamente disponibilizado na FIB de um nó. Ao disponibilizar diretamente na FIB, permite-se desviar imediatamente os tráfegos prejudicados por uma falha adjacente: tanto nó quanto enlace.

A CER_pró-ativa necessita de 2 condições para atingir 100% de cobertura de falha:

- Uma estrutura física de topologia capaz de manter total conectividade durante a ocorrência de uma falha, podendo ser de nó ou de enlace. Para isso, utilizando a Teoria de Grafos, uma topologia modelada em grafo consegue manter total conectividade entre todos os vértices, mesmo após uma falha de um vértice/aresta, se o grafo obedece ao *Teorema de Menger* para ser *k-conexo* [Diestel 2005]. A variável *k* indica que ao remover *k-1* vértices da topologia, a mesma mantém todos os demais vértices interconectados. Portanto, uma topologia mínima deve ser representada por um grafo *2-conexo*, ou seja, suporta total conectividade entre os vértices mesmo removendo no máximo (2-1) vértices do grafo. Esta restrição evita uma alta complexidade nas abordagens de contorno de falha por não tratar múltiplas falhas transitórias, sendo desejável, pois múltiplas falhas transitórias, independentes e simultâneas são raras em topologias reais de *backbone* [Nucci et al. 2003].
- Planejamento antecipado da rede para distribuir os tráfegos na topologia, de forma a utilizar a capacidade dos enlaces em menos de 50%. Essa restrição é observada em *backbones* reais e o seu objetivo é permitir uma acomodação de tráfegos desviados na ocorrência de falhas [Iannaccone et al. 2004]. Para tanto, pode-se utilizar uma matriz de tráfego com uma abordagem *offline* para melhor distribuir os pesos aos enlaces como a descrita em [Fortz and Thorup 2000].

Estas condições descritas podem ser adotadas para qualquer outra abordagem de contorno de falhas. Caso contrário, a topologia pode não ser capaz de comportar os desvios de tráfego em 100% de cobertura de falha na topologia.

3.1. Cálculo dos Caminhos Emergências Rápidos (CERs)

A CER_pró-ativa é projetada para ser executada em cada nó da rede logo após a execução das rotinas de roteamento do OSPF. O objetivo é reutilizar a base de dados atualizada do estado do enlace mantidas pelo OSPF. Um nó que executa a CER_pró-

ativa é denominado de nó_origem. O cálculo dos CERs remove cada um dos componentes de rede adjacentes ao nó_origem, podendo ser um nó junto com o enlace adjacente, ou apenas o enlace adjacente. O objetivo é identificar quais nós ficam inalcançáveis na árvore de menor caminho na ocorrência de uma falha, sendo denominados de nós_destino. Cada nó_destino identificado é utilizado como referência para a CER_pró-ativa gerar os CERs necessários.

A decisão sobre qual componente deve ser retirado (nó ou enlace), é verificada quando o nó_destino de um caminho afetado está localizado exatamente no nó adjacente. Caso essa condição seja verdadeira, seleciona-se apenas o enlace pois o nó adjacente já é o nó final do menor caminho original e, portanto, não pode ser contornado. Caso contrário, seleciona-se sempre o enlace e o nó adjacente, pois um nó, na presença de uma falha real, não possui informações suficientes se a falha é do enlace ou do nó adjacente. Para cada componente retirado (indicando uma falha) realiza-se um cálculo SPF para todos os nós_destino afetados na árvore de menores caminhos. Como apenas uma parte da árvore tende a ser afetada, utiliza-se o *Incremental-SPF* [Narvaez 2000], que melhora consideravelmente o tempo de cálculo. O resultado é um conjunto de menores *caminhos alternativos* para os nós_destino e que conseguem contornar o componente removido. Dessa forma, para cada nó_destino afetado pelo componente removido, existe pelo menos um *caminho alternativo* correspondente.

Cada *caminho alternativo* representa todos os nós, desde o nó_origem até o nó_destino, que devem ser teoricamente seguidos pelos pacotes quando desviados (caminho de recuperação). Além disso, esse caminho é o mesmo caminho do nó_origem até o nó_destino obtido pelo OSPF caso houvesse uma reação à falha desse componente. Devido a essa característica, é possível limitar a extensão do *caminho alternativo* ao identificar um nó especial desse caminho, denominado nó_CER. O nó_CER é identificado quando existir, a partir dele, um menor caminho original do OSPF para o nó_destino e esse caminho não atravesse o nó_origem ou o componente retirado. Dessa forma, um nó_CER possibilita a identificação de um sub-caminho em cada *caminho alternativo* (seqüência: nó_origem até o nó_CER), que é identificado como CER. Para cada *caminho alternativo* para um nó_destino, encontra-se apenas um CER respectivo.

A partir do nó_CER os pacotes são encaminhados reutilizando o encaminhamento OSPF original até o nó_destino, o que completa a seqüência de nós do *caminho alternativo*. Durante a reação do OSPF em um ambiente real, os nós pertencentes ao CER podem ser afetados pelo período de convergência, pois apenas esses nós têm os menores caminhos afetados na árvore SPF a partir do nó_origem até o nó_destino no *caminho alternativo*. O CER tem por objetivo manter o desvio dos pacotes durante esse período, guiando os pacotes para um caminho de recuperação que será o mesmo do OSPF após o término do período de convergência para essa determinada falha. Essa característica permite uma adaptação gradual do caminho de recuperação para o novo caminho OSPF, que é atualizado gradativamente nos nós integrantes do CER. Durante essa atualização, o CER se mantém ativo realizando o desvio separadamente do OSPF por um intervalo suficiente (aproximadamente 10s na configuração padrão OSPF [Iannaccone et al. 2004]) para que todos os nós do CER executem as rotinas do OSPF. Ao término desse intervalo, o nó_origem interrompe o desvio dos pacotes via CER e aciona a CER_pró-ativa para calcular um novo CER de acordo com a base de informações atualizada do estado do enlace. Ao interromper o

desvio, os nós utilizam o encaminhamento OSPF já com o novo caminho configurado em todos os nós, que era a mesma seqüência de nós do *caminho alternativo*. Dessa forma, os CERs servem como uma abordagem auxiliar na redução da taxa de perdas durante o período de convergência do OSPF.

Quando um CER for gerado para um nó_destino, os nós integrantes do CER são armazenados em um vetor denominado: VetorSPF[nó_destino]. Caso existam vários caminhos alternativos para um mesmo nó_destino, então existe mais de um CER para um mesmo nó_destino. Uma seleção dos respectivos CERs é feita de acordo com a localização do nó_CER e é dependente da característica do *caminho alternativo*. Para tanto, a CER_pró-ativa estabelece níveis de classificação: ECMP, LFA e *Signal* (SIG). Para as formulações apresentadas nos níveis, utiliza-se a notação da Tabela 1.

Tabela 1. Notação

Símbolo	Descrição
nó_adjacente	Nó adjacente simulado com falha.
nó_vizinho	Nó adjacente ao nó_origem que é início do *caminho alternativo*/CER
DistânciaSPF(x,y)	Custo do menor caminho de x até y na árvore SPF já existente no OSPF.
NósSPF(x,y)	Conjunto de nós pertencentes ao menor caminho de x até y na árvore SPF já existente no OSPF.
NumNosSPF(x,y)	Número de nós do caminho SPF entre x e y.

3.1.1 CER_pró-ativa: Nível ECMP

Nível ECMP: é o nível mais prioritário, pois busca-se reutilizar caminhos obtidos com *Equal Cost Multipath Protocol* (ECMP) já calculados pelas rotinas de roteamento do OSPF [Moy 1998]. Nesse nível, o *caminho alternativo* obtido possui o mesmo custo total que o caminho original do OSPF, porém não utiliza o componente retirado. Isso implica em ter um *caminho alternativo* que é um dos caminhos originais do OSPF, porém sem utilizar o componente em questão. O nó_CER é identificado já no primeiro nó do *caminho alternativo* (nó_vizinho ao nó_origem). Portanto, o CER de nível ECMP gerado é VetorSPF[nó_destino] ={nó_vizinho}. Para o nível ECMP a formulação (1) deve ser obedecida para identificar caminhos com distâncias iguais sendo limitado pela restrição (1.a), que desconsidera caminhos que utilizem o nó_adjacente como nó integrante do menor caminho quando for diferente do nó_destino.

$$DistânciaSPF(nó\_origem, nó\_destino) = DistânciaSPF(nó\_origem, nó\_vizinho) + DistânciaSPF(nó\_vizinho, nó\_destino) \quad (1)$$

Sujeito a:

$$nó\_adjacente \notin NósSPF(nó\_vizinho, nó\_destino) \text{ se } nó\_adjacente \neq nó\_destino \quad (1.a)$$

Se existir mais de um caminho alternativo deste nível para um mesmo nó_destino, desde que utilize nós_vizinhos distintos, então o CER de cada um deles deve ser considerado. O objetivo é disponibilizar vários CERs desse nível para possibilitar uma divisão dos fluxos com o mecanismo ECMP já existente do OSPF.

3.1.2. CER_pró-ativa: Nível LFA

Nível LFA: é o nível que adapta a abordagem original LFA, sendo utilizado quando o nível ECMP não for possível. O primeiro nó do *caminho alternativo* (nó_vizinho) deve obedecer às restrições de um LFA descritas em [Atlas and Zinin 2006]. Essas restrições verificam se os pacotes, quando desviados, conseguem atingir o nó_destino contornando o componente retirado a partir de um menor caminho do nó_vizinho. O nó_CER é, portanto, identificado no primeiro nó do *caminho alternativo* (nó_vizinho ao nó_origem) e o CER de nível LFA gerado é VetorSPF[nó_destino]={nó_vizinho}.

Caso exista mais de um *caminho alternativo* de mesmo custo para um mesmo nó_destino no nível LFA, deve-se escolher apenas um único CER minimizando o número de nós desse caminho para economia de recursos. A escolha de um único CER deve-se ao fato do OSPF não permitir balanceamento de carga no desvio dos pacotes em caminhos que não são de mesmo custo a partir do nó_origem. Para tanto a formulação descrita em (2) complementa a formulação [Atlas and Zinin 2006], minimizando a soma dos custos dos enlaces do caminho considerando o número de nós. A expressão *DistânciaLFA(x,y)* indica o resultado da abordagem [Atlas and Zinin 2006], que já verifica se o nó_vizinho possui as características para prover LFA. Caso existir mais de um nó_vizinho de nível LFA com um mesmo custo de *caminho alternativo*, então o menor número de nós prevalece no processo de minimização. A restrição (2.a) impede o uso do nó_adjacente no menor caminho quando ele for diferente do nó_destino.

$$\begin{aligned} &\textit{Minimizar (DistânciaLFA(nó\_vizinho, nó\_destino) + NumNósSPF(nó\_vizinho, nó\_destino))} \\ &\textit{Sujeito a:} \end{aligned} \tag{2}$$

$$\textit{nó\_adjacente} \notin \textit{NósSPF(nó\_vizinho, nó\_destino), se nó\_adjacente} \neq \textit{nó\_destino} \tag{2.a}$$

3.1.3. CER_pró-ativa: Nível SIG

Nível SIG: é o nível utilizado quando nenhum dos níveis anteriores for aplicado. Sabe-se que o nó_vizinho não pode ser identificado como nó_CER, pois não é capaz de contornar o componente retirado com sucesso. Verifica-se então qual dos nós seguintes (nó_TMP) ao nó_vizinho na seqüência do *caminho alternativo* pode ser identificado como sendo o nó_CER. O nível SIG cria as formulações (3) e (4) para verificar cada nó_TMP de acordo com a localização do nó_destino.

Se o nó_destino for o nó_adjacente, então o componente retirado é o enlace adjacente e deve obedecer à formulação (3). Essa formulação minimiza a *DistânciaSPF* do nó_TMP até o nó_destino considerando o número de nós dos caminhos possíveis. A restrição (3.a) evita a ocorrência de *loops* de roteamento. Não há a restrição do nó_adjacente (conforme os níveis anteriores), pois o nó_adjacente é o nó final (nó_destino) do *caminho alternativo*.

$$\begin{aligned} &\textit{Minimizar ((DistânciaSPF(nó\_TMP, nó\_destino) + NósSPF(nó\_TMP, nó\_destino))} \\ &\textit{Sujeito a:} \end{aligned} \tag{3}$$

$$\textit{DistânciaSPF(nó\_TMP, nó\_destino)} < \textit{DistânciaSPF(nó\_TMP, nó\_origem)} + \textit{DistânciaSPF(nó\_origem, nó\_destino)} \tag{3.a}$$

Se o nó_destino for diferente do nó_adjacente, então o componente retirado para o cálculo é o nó_adjacente e deve obedecer à formulação (4) para minimizar a

DistânciaSPF do nó_TMP até o nó_destino junto com o número de nós no caminho, além de considerar as restrições: (4.a) para evitar *loops* de roteamento, (4.b) para impedir o uso do nó_adjacente no menor caminho, se ele for diferente do nó_destino.

$$\begin{aligned} &\textit{Minimizar ((DistânciaSPF(nó\_TMP, nó\_destino) + NósSPF(nó\_TMP, nó\_destino))} \\ &\textit{Sujeito a:} \end{aligned} \tag{4}$$

$$\textit{DistânciaSPF(nó\_TMP, nó\_destino)} < \textit{DistânciaSPF(nó\_TMP, nó\_origem)} + \textit{DistânciaSPF(nó\_origem, nó\_destino)} \tag{4.a}$$

$$\textit{nó\_adjacente} \notin \textit{NósSPF(nó\_TMP, nó\_destino), se nó\_adjacente} \neq \textit{nó\_destino} \tag{4.b}$$

O nó_TMP que primeiro satisfazer a formulação (3) ou a formulação (4) de acordo com a localização do nó_destino é identificado como nó_CER. O CER estende-se do nó_vizinho até o nó_CER, e é armazenado em um VetorSPF[nó_destino]= {nó_vizinho, demais_nós_ intermediários, nó_CER}. Se existir mais de um *caminho alternativo* de nível SIG para um mesmo nó_destino, as formulações determinam 1 CER minimizando o custo dos caminhos e o numero de nós para economia de recursos.

Como o nível SIG obtém CERs que se estendem até um nó_CER além do nó_vizinho, necessita-se de um processo de sinalização pró-ativo e simplificado. O objetivo é informar, antecipadamente à falha, os nós integrantes do caminho CER para que realizem um encaminhamento diferenciado dos pacotes desviados (evitar *loops* de roteamento). Esse encaminhamento diferenciado é baseado em uma marcação simplificada nos pacotes (evita os problemas da técnica de encapsulamento), que é previamente acordado entre o nó_origem e os demais nós integrantes do CER, para impedir o encaminhamento baseado no endereço IP de destino. Esse encaminhamento diferenciado apenas é executado quando os pacotes tiverem a marcação acordada, utilizando o mesmo processo de marcação do MRC [Kvalbein et al. 2006]. O processo de sinalização é uma abordagem integrante do esquema E-CER ainda em desenvolvimento e deve ser publicado mais detalhadamente em trabalhos futuros.

3.2. Preenchimento dos VetoresSPF na FIB

Com os VetoresSPF gerados, observando o mesmo conceito adotado por [Bryant et al. 2006] e [Kvalbein et al. 2006], deve-se adiciona-los na FIB como um campo extra para possibilitar uma reação imediata na ocorrência de falha real.

Para economia de espaço na FIB, os VetoresSPF são representados obedecendo a duas regras:

- Todos os prefixos de rede anunciados por um mesmo nó_destino (obtém-se esses dados utilizando as informações da base de dados do estado do enlace) devem referenciar um mesmo VetorSPF[nó_destino].
- Se existir dois ou mais VetoresSPF com nós_destino diferentes porém contendo CERs iguais, utiliza-se apenas uma única referência na FIB, uma vez que o caminho emergencial será igual para esses nós_destino.

4. Exemplificando a CER_pró-ativa

A Figura 1 ilustra os níveis de prioridade descritos na seção 3 considerando uma

topologia. O resultado da CER_pró-ativa é aplicada ao nó *b* (nó_origem), onde os nós_destino *c,d,f,g,k,j* (com coloração mais escura) ficam afetados ao ter uma falha simulada no nó adjacente *a*. Para obter um caminho para o nó_destino *j*, o nível atendido é o ECMP (subseção 3.1.1), pois o custo do caminho OSPF comprometido (a,*d,g,k,j*) é 6, sendo que possui o mesmo custo do *caminho alternativo* (*e,i,j*). Portanto, (*e,i,j*) também é um menor caminho OSPF e o CER é o VetorSPF[j]={*e*}. Para os nós_destino *k* e *g*, o nível LFA é atendido (subseção 3.1.2), pois o caminho alternativo possui, a partir do nó_vizinho (*e*) ao nó_origem, um menor caminho OSPF que alcança os nós_destino *k* e *g* sem utilizar o nó *a*. Portanto, os respectivos CERs são VetorSPF[k]=VetosSPF[g]={*e*}. Para os nós_destino *c*, *d* e *f*, apenas o nível SIG é atendido, pois o *caminho alternativo* não atende aos níveis ECMP ou LFA, encontrando o nó_CER além do nó_vizinho: nó *h*. A partir de *h* o nível SIG é obedecido (subseção 3.1.3) para esses nós e os pacotes, ao serem desviados, conseguem atingir os respectivos nós_destino *c*, *d* e *f*, contornando o nó *a* com sucesso a partir do nó_CER. O CER para esses nós são: VetorSPF[c]=VetorSPF[d]=VetorSPF[f]={*e,h*}.

VetorSPF[c]= **{e,h}** (Nível SIG)
VetorSPF[d]= **{e,h}** (Nível SIG)
VetorSPF[f]= **{e,h}** (Nível SIG)
VetorSPF[g]= **{e}** (Nível LFA)
VetorSPF[j]= **{e}** (Nível ECMP)
VetorSPF[k]= **{e}** (Nível LFA)

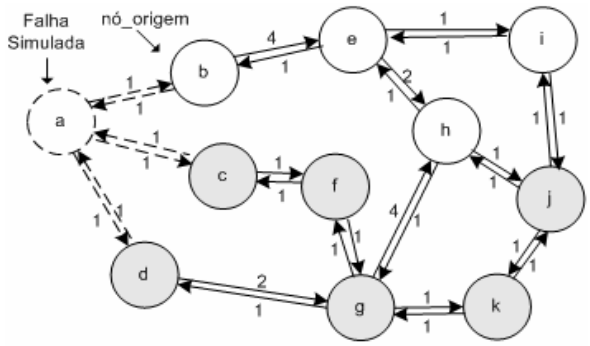

Figura 1. Resultado da CER_pró-ativa no nó *b* (nó_origem) que contorna o nó *a*

Considerando o resultado ilustrado na Figura 1, a Tabela 2 apresenta uma visualização da economia de recursos do nó *b* em relação aos prefixos de rede (com os respectivos nós anunciadores desses prefixos) afetados com a falha simulada em *a*.

Tabela 2. FIB no nó *b* após execução da CER_pró-ativa

Prefixos de rede	OSPF	VetorSPF[ ]
Prefixos_anunciados_por(c)	Próximo nó = a	{e,h}
Prefixos_anunciados_por(d)	Próximo nó = a	
Prefixos_anunciados_por(f)	Próximo nó = a	
Prefixos_anunciados_por(g)	Próximo nó = a	{e}
Prefixos_anunciados_por(j)	Próximo nó = a	
Prefixos_anunciados_por(k)	Próximo nó = a	

5. Avaliação da CER_pró-ativa

Um algoritmo foi desenvolvido para gerar os CERs seguindo a abordagem descrita na seção 3 e implementado em Java. A complexidade do algoritmo obtido é $O(n^3)$ [Barreto

2006], onde *n* é o número de nós da topologia. Porém, esse algoritmo é dependente da complexidade do algoritmo *Incremental-SPF*, que no pior caso (quando toda a árvore SPF for comprometida pela retirada de um componente) tende a ser igual à complexidade do algoritmo SPF de $O(n^2)$, o que não é comum acontecer nesses casos.

Foram implementadas também em Java as abordagens NotVia e MRC, baseando-se na descrição de [Bryan et al. 2006] e de [Kvalbein et al. 2006], respectivamente. O *Incremental-SPF* também é utilizado como algoritmo base do NotVia; já o MRC utiliza um algoritmo próprio que depende do número de nós, enlaces e sub-topologias. Todas as abordagens utilizam como entrada uma matriz de adjacência para representar uma topologia a ser analisada.

Foram geradas 20 topologias artificiais utilizando o gerador BRITE [Medina et al. 2002] tomando como base alguns parâmetros de configuração descritos em [Heckmann et al. 2003]. Foi usado também o *backbone* de uma topologia real GEANT2 *pan-European Research Network* (*http://www.geant2.net*) por tratar-se de uma topologia conhecida e bem documentada. Os pesos dos enlaces foram configurados com um valor inversamente proporcional à capacidade dos enlaces para atender a uma configuração simples recomendada pela Cisco (*Configuring OSPF*) comumente adotada para o OSPF. Para demonstrar dos resultados obtidos, apenas os resultados gerados da GEANT2 são apresentados, pois as 20 demais topologias geraram resultados similares e não foram apresentadas aqui por falta de espaço.

Para analisar a solução provida pela CER_pró-ativa em relação as demais abordagens, compara-se a extensão dos caminhos de recuperação em termos do número de nós utilizados entre todos os pares origem destino [Hansen et al. 2006], pois permite dimensionar qual a extensão a ser percorrida pelos pacotes quando desviados. Quanto maior a extensão, maior é a quantidade de recursos utilizados para realizar o desvio. Foram gerados 647 caminhos de recuperação: 173 obtidos por ECMP, 192 obtidos por LFA e a grande maioria, 283, obtidos por nível SIG da CER_pró-ativa, NotVia ou MRC. Os caminhos ECMP e LFA não são analisados aqui, pois as abordagens NotVia e MRC não possuem um suporte nativo incorporado para esses caminhos, como é o caso da CER_pró-ativa. O NotVia e o MRC apenas indicam a abordagem [Atlas and Zinim 2006] para ser utilizada [Shand and Bryant 2006]. Porém, os resultados obtidos com ECMP e LFA foram considerados iguais para todas as abordagens para simplificar a apresentação.

A Figura 2 ilustra um gráfico comparativo dos 283 caminhos de recuperação gerados para a GEANT2 utilizando as 3 abordagens. O gráfico mostra também o *OSPF normal* (caminhos existentes antes de simular uma falha) e o *OSPF Re* (caminhos existentes após a convergência do OSPF para as falhas simuladas). Os resultados obtidos nesta análise são similares aos obtidos na mesma análise de [Hansen et al. 2006] com outras topologias. O *OSPF normal* possui a maior porcentagem dos caminhos com menor número de nós (62% com 3 e 4 nós) por ter todos os componentes disponíveis na topologia. Quando há uma simulação de falha, a ação das 3 abordagens gera caminhos naturalmente mais extensos que o *OSPF normal* por não terem todos os componentes da topologia. O NotVia apresenta um leve vantagem com um maior percentual dos caminhos com menos nós (43% com 4 e 5 nós) em relação à abordagem MRC (39% com 4 e 5 nós), sendo necessárias 4 sub-topologias para cobrir toda a topologia. Essa pequena diferença ocorre devido ao contorno apenas do componente falho no NotVia,

ao contrário do MRC que contorna mais componentes além do componente falho devido às características das sub-topologias. Já a CER_pró-ativa de nível SIG encontra a maior porcentagem dos caminhos de recuperação com menores caminhos (52% com 4 e 5 nós) que as abordagens NotVia e MRC devido ao processos de minimização utilizados nas formulações. Além disso, os caminhos de recuperação obtidos com a CER_pró-ativa são os mesmos menores caminhos do OSPF se existisse uma reação às falhas simuladas (*OSPF-Re)*, com a vantagem de não depender de um período de convergência.

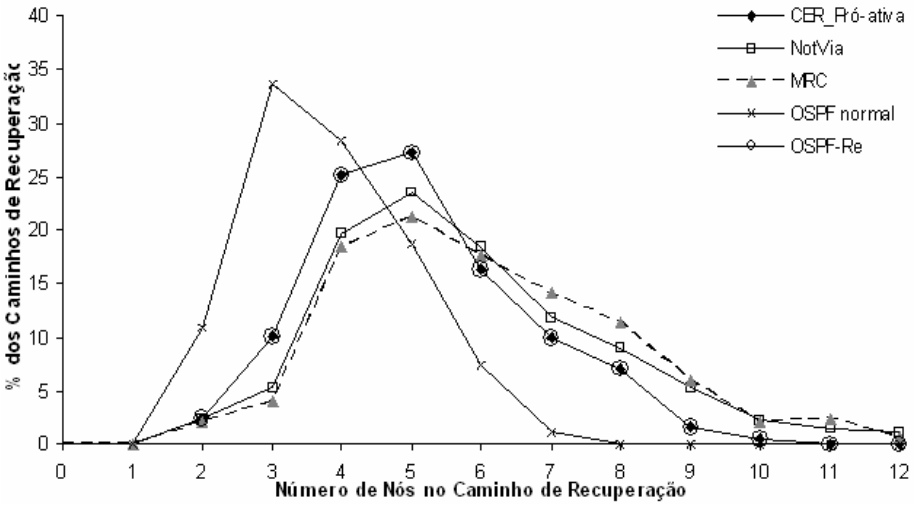

Figura 2. Comparação dos caminhos obtidos com CER_pró-ativa, NotVia e MRC

Os caminhos de recuperação da CER_pró-ativa são os *caminhos alternativos*: do nó_origem até o nó_destino. Porém, o desvio propriamente dito dos pacotes segue apenas os nós apontados nos *caminhos emergenciais rápidos* (CERs), que são sub-caminhos dos *caminhos alternativos*: nó_vizinho até o nó_CER respectivo.

A extensão em número de nós dos CERs é minimizada nas formulações e influenciam diretamente na quantidade de informações que devem ser adicionadas na FIB. Dentre as CERs existentes, a maioria (65%) possui apenas 2 nós, em seguida (16%) 3 nós, (9%) 4 nós e o restante até no máximo 9 nós (10%) que são representadas em forma de vetor na FIB. Esses dados permitem uma análise da quantidade de informações extras de memória inseridas na FIB por cada abordagem. Uma média foi estimada por cada nó da GEANT2, pois não foi possível obter informações mais detalhadas sobre a FIB de cada nó.

A Tabela 3 compara a quantidade aproximada de recursos (memória) inseridos na FIB de cada nó para o correto funcionamento das 3 abordagens em uma rede IPv4. Obedecendo às regras da seção 3.3 para essa topologia, a CER_pró-ativa necessita de 5 VetoresSPF referenciados e distribuídos entre cada entrada na FIB, sendo que a maioria dos vetores possui entre 2 e 4 nós para representar um CER. Como cada nó do OSPF é representado por um *Router ID* de 4bytes [Moy 1998], a CER_pró-ativa necessita de (0,65×5)×(2nós×4) bytes, (0,16×5)×(3nós×4) bytes e (0,09×5)×(4nós×4) bytes aproximadamente. Já a abordagem NotVia necessita de 76 novas entradas para os endereços *not-via* além de 5 *next-next-hops* referenciados e distribuídos entre cada entrada da FIB. Portanto, o NotVia necessita de 5×4 bytes + 76×12 bytes, sendo que os 12 bytes são um valor mínimo aproximado da quantidade de memória necessária para uma nova entrada

na FIB (Endereço de Rede ou IP de destino (4 bytes) + Máscara (4 bytes) + *next-hop* (4 bytes)) [Moy 1998]. A MRC necessita de 4 x FIB bytes para comportar as 4 sub-topologias, sendo facilmente identificável como a abordagem com maior uso de recursos.

Tabela 3. Quantidade média estimada de informações extras na FIB por nó

CER_pró-ativa	NotVia	MRC
5 VetoresSPF com 65% 2 nós , 16% 3 nós e 9% 4 nós por vetor. *Total: 43 bytes*	76 novas entradas na FIB + 5 *next-next-hops*. *Total: 932 bytes*	4 sub-topologias para cobrir toda a topologia *Total: 4×(FIB inteira) bytes*

A CER_pró-ativa demonstra um ganho significativo em termos de economia de recursos necessários na FIB em relação às demais abordagens, sendo aproximadamente 43 bytes, contra 932 bytes do NotVia, e (4 x tamanho da FIB inteira) bytes do MRC.

6. Conclusão

Os protocolos de roteamento, mesmo com um período de convergência menor que 1 segundo, não reagem em tempo hábil a uma falha em *backbones* mais velozes. Isto implica em altas taxas de perdas de pacotes até o final desse período, o que é agravado em casos de falha transitória. Este trabalho apresentou a abordagem CER_pró-ativa para gerar *caminhos emergenciais rápidos*, que são utilizados para auxiliar os protocolos de roteamento a contornar falhas durante esse período. A CER_pró-ativa obteve menores caminhos de recuperação além de utilizar menos memória na FIB para identificar esses caminhos. Cada caminho de recuperação obtido é o mesmo caminho que o OSPF geraria se houvesse uma reação à falha. Essa característica possibilita uma adaptação gradual do caminho de recuperação para o caminho OSPF durante o período de convergência. Um melhor detalhamento dessa adaptação será apresentado em trabalhos futuros bem como uma avaliação da CER_pró-ativa, junto com todo o esquema E-CER, em um simulador de rede e em ambiente real estão sendo desenvolvidos. Uma abordagem mais aprimorada para lidar com múltiplas falhas simultâneas (*Shared Risk Link Group*) e roteamento *Multicast* também será desenvolvida em trabalhos futuros.

References

Alaettinoglu, C., Jacobson, V., and Yu, H. (2000). Towards Mili-second IGP Convergence. In *Internet Draft*. IETF Network-WG.

Atlas, A. (2006). U-Turn Alternate for IP/LDP Fast-Reroute. In *Internet Draft*. IETF Routing-WG.

Atlas, A., and Zinin, A. (2006). Basic Specification for IP Fast-Reroute Loop-Free Alternate. In *Internet Draft*. IETF Routing-WG.

Barreto, F. (2006). CER_pró-ativa Algoritmo. In *Technical Report CER_01*, Universidade Tecnológica Federal do Paraná, Curitiba.

Medina, A., Lakhina, A., Matta, I. and Byers, J. (2002). BRITE Topology Generator.

Bryant, S., Shand, M., and Previdi, S. (2006). Ip Fast Reroute Using Not-via Address. In *Internet Draft*. IETF Routing-WG.

Bryant, S., FilsFils, S., Previdi, S., and Shand, M. (2005). IP fast reroute using tunnels. In *Internet Draft*. IETF Routing-WG.

Deering, S., and Hinden, R. (1998). Internet Protocol, Version 6 (Ipv6) Specification. In *RFC 2460*. IETF Ipv6-WG.

Diestel, R. (2005), Graph Theory, Springer-Velang, 3th edition.

Dijkstra, E. W. (1959). A Note on Two Problems in Connection with Graphs. In *Numerische Mathematik*, pages 269-271.

Fortz, B., and Thorup, M. (2000). Internet Traffic Engineering by optimizing OSPF Weights. In *IEEE INFOCOM Computer Communications*, pages 519-528.

Francois, P., Filfis, C., Evans, C., and Bonaventure, O. (2005). Achieving sub-second IGP convergence in large IP networks. In *ACM SIGCOMM*, pages 34-44.

Hansen, A. F., and Cicic, T., and Gjessing, S. (2006). Alternative Schemes for Proactive IP Recovery. In *Next Generation Internet Desing and Engineering*, pages 1-8.

Heckmann, O., Piringer, M., Shcmitt, J., and Steinmetz, R. (2003). On Realistic Network Topologies for Simulation. In *ACM SIGCOMM*, pages 28-32.

Iannaccone, G., Chuah, C., Bhattacharyya, S., and Diot, C. (2004). Feasibility of IP Restoration in a Tier-1 Backbone. In *IEEE Network Magazine*, pages 13-19.

Kvalbein, A., Hansen, A. F., Cicic, T., Gjessing, S., and Lysne, O. (2006). Fast IP Network Recovery using Multiple Routing Configurations. In *IEEE INFOCOM Computer Communications*.

Markopoulou, A., Iannaccone, G., Bhattacharya, S., Chuah, C., and Diot, C. (2004) "Characterization of failures in an IP backbone" In *IEEE INFOCOM Computer Communications*, pages 406-416.

Moy, J. (1998). OSPF version 2. In *RFC 2328*. IETF Network-WG.

Narvaez, P. (2000). Routing Reconfiguration in IP Networks. Phd Thesis. Massachusetts Institute of Technology.

Nucci, A., Schoroeder, B., Bhattacharyya, S., Taft, N., and Diot, C. (2003). IGP Link Weight Assignment for Transient Link Failures, In *International Teletraffic Congress*.

Pan, P., Swallow, G., and Atlas, A. (2005). Fast Reroute Extensions to RSVP-TE for LSP Tunnels. In *RFC 4090*. IETF Network-WG.

Paxson, V. (1996). End-to-End Routing Behavior in the Internet. In *ACM SIGCOMM*, volume 26, pages 25-38.

Shand, M., and Bryant, S. (2006). IP Fast Reroute Framework. In *Internet Draft*. IETF Routing-WG.

Zhong, Z., Nelakuditi, S., Yu, Y., Lee, S., Wang, J., and Chuah, C. (2005). Failure Inferencing based Fast Rerouting for Handling Transient Link and Node Failures. In *IEEE Global Internet*.