

# O Rerroteamento Parcial Pró-ativo em Redes baseadas em Circuito Virtual

R. de B. Correia, E. L. Cecilio, A.P. M. Dumont, L. F. Rust da C. e Luci Pirmez

Núcleo de Computação Eletrônica - Universidade Federal do Rio de Janeiro  
Tel: 021 2598-3159 - Caixa Postal 2324, Rio de Janeiro, RJ, Brasil

cecilio@ime.eb.br, dumont@unisys.com.br,  
reinaldo@posgrad.nce.ufrj.br, {rust, luci}@nce.ufrj.br

***Abstract.** Flow rerouting in virtual-circuit driven networks, especially in MPLS networks, is being considered a key mechanism for providing IP networks with high degree of availability and resource utilization. We present a rerouting scheme for, proactively, redirecting flows through alternative paths with sufficient resources. The rerouting actions are triggered by external messages, signaling the detection of a QoS fault tendency. Typically, such messages are generated by a proactive performance management system. Our rerouting scheme relies on active technology based infrastructure and Java programming. Test results not only show that it is possible to reroute application flows in a transparent manner but also establish the limits of our proposed rerouting scheme in terms of virtual-circuit extension.*

## 1. Introdução

Este artigo apresenta os resultados dos testes efetuados em um protótipo de uma arquitetura genérica de rerroteamento pró-ativo anteriormente introduzida em [4]. O objetivo da arquitetura proposta é possibilitar o redirecionamento de fluxos de aplicações mediante solicitações externas. Essas solicitações externas são disparadas quando ocorre a detecção de tendência de falha de QoS que, tipicamente, é realizada por algum sistema de gerenciamento pró-ativo [1].

A arquitetura de rerroteamento pró-ativo utiliza os princípios do roteamento baseado em QoS e tem como infra-estrutura uma tecnologia de enlace orientada a conexão e a tecnologia ativa[5]. Os testes foram efetuados sob um protótipo especializado para fluxos com restrição de banda e retardo e para uma infra-estrutura de rede MPLS (*Multiprotocol Label Switching*). Esse protótipo que foi desenvolvido em Java utiliza uma infra-estrutura de mobilidade chamada  $\mu$ code[6]. Os resultados permitiram avaliar o desempenho de seu funcionamento e a sua aplicabilidade no redirecionamento de fluxos de aplicações multimídia. Os resultados dos testes também foram confrontados com os resultados obtidos em [1] para determinar os limites do emprego da arquitetura proposta.

Além desta introdução, o artigo está organizado como segue: a Seção 2 apresenta os principais aspectos do rerroteamento reativo e , principalmente, o pró-ativo; a Seção 3 descreve a arquitetura e a implementação do protótipo; os resultados dos testes assim como a análise destes estão detalhados na Seção 4; os trabalhos correlatos estão resumidos na Seção 5 e a Seção 6 finaliza o artigo com as conclusões e trabalhos futuros.

## 2. Rerroteamento Pró-ativo

Tradicionalmente, as abordagens de rerroteamento são reativas porque as ações são iniciadas depois da ocorrência da falha. Nas abordagens pró-ativas, as ações devem ser disparadas antes que as falhas ocorram, reduzindo a latência de rerroteamento. Existem duas abordagens para o rerroteamento pró-ativo: plena e parcial. O rerroteamento pleno ocorre quando todo o circuito virtual é substituído, enquanto que o parcial é aquele em que o alvo da substituição é apenas um trecho do circuito virtual.

A abordagem parcial de rerroteamento pressupõe, a priori, a existência de um trecho a ser substituído, o trecho crítico. Esse trecho é aquele que contribui com o maior peso percentual do valor fim a fim da métrica de QoS monitorada do circuito virtual. A estratégia no rerroteamento parcial é diminuir a distância entre a origem e destino, restringindo a busca de caminhos alternativos, a fim de reduzir o tempo de processamento. A vantagem de tal abordagem torna-se acentuada quando os circui-

tos virtuais são longos.

### 3. Arquitetura de Rerroteamento Pró-ativo usando Tecnologia Ativa

A Arquitetura de Rerroteamento Pró-ativo é constituída de três agentes: AgenteNóEntrada (ANE), AgenteNóIntermediário (ANI) e AgenteRotaAlternativa (ARA). O agrupamento das funcionalidades necessárias nesses três agentes visa diminuir o tamanho de seus códigos e, conseqüentemente, os tempos de migração, de instalação e de memória nos dispositivos da rede.

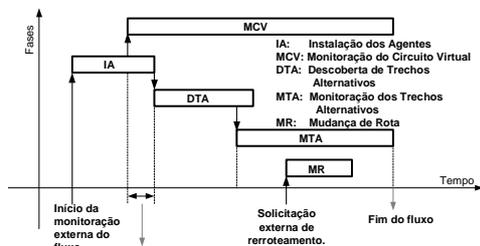


Figura 1 - Fases do rerroteamento

O funcionamento da Arquitetura de Rerroteamento Pró-ativo pode ser explicado estabelecendo cinco fases para as operações necessárias ao rerroteamento de um fluxo: (i) Instalação dos agentes (IA), (ii) Monitoração do Circuito Virtual (MCV), (iii) Descoberta de Trechos Alternativos (DTA), (iv) Monitoração dos Trechos Alternativos (MTA) e (v) Mudança de Rota (MR). O diagrama da Figura 1 apresenta a seqüência de execução das fases, detalhando a relação temporal entre as fases e eventos externos. As ações de rerroteamento são disparadas por dois eventos externos: início da monitoração externa do fluxo e solicitação externa de rerroteamento, indicando a detecção de uma tendência de falha de QoS (fig 1).

instornos: início da monitoração externa do fluxo e solicitação externa de rerroteamento, indicando a detecção de uma tendência de falha de QoS (fig 1).

#### Instalação dos agentes (IA) e Monitoração do Circuito Virtual (MCV)

A Figura 2 apresenta as principais operações executadas durante essas duas fases. As setas tracejadas estão relacionadas à fase MCV. Assim que o agente ANE inicia o seu processamento, cria o agente ANI (operação 1). Uma vez criado, réplicas do agente ANI instalam-se nos nós do circuito virtual (operações 2 a 7), possibilitando o monitoramento do estado local durante a fase MCV (operações 8 a 12). O agente ANE com as informações de estado calcula a posição do trecho crítico e cria o agente ARA. Este migra para o primeiro nó do trecho crítico, encerrando a fase IA.

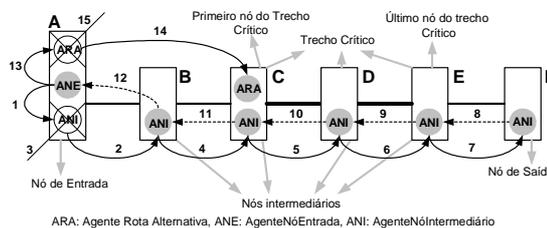


Figura 3 - Operações da fase IA

#### Descoberta de Trechos Alternativos (DTA) e Monitoração dos Trechos Alternativos (MTA)

As operações das fases DTA e MTA estão repre-

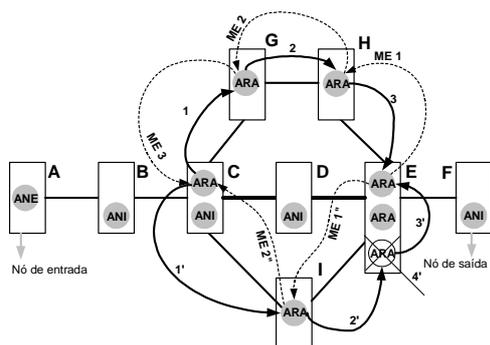


Figura 2 - Fases DTA e MTA

sentadas graficamente na figura 3. A setas tracejadas referem-se às operações da fase MTA. A fase DTA consiste na instalação de réplicas do agente ARA em todos os nós de todos os trechos alternativos em torno do trecho crítico (operações 1", 2" e 1 a 3). A réplica do último nó do trecho crítico envia mensagens periódicas de estado através dos trechos alternativos descobertos (operações ME1", ME2" e ME1 a ME3). O agente ARA do nó C com as informações de estado dos trechos alternativos descobertos é capaz de escolher o melhor trecho alternativo com recursos suficientes na ocasião de solicitação externa de rerroteamento.

#### Mudança de Rota (MR)

As operações envolvidas durante essa fase dependem da abordagem adotada na geração e associação dos identificadores locais do circuito virtual. As abordagens podem ser antecipadas ou sob demanda. Na antecipada, esses identificadores são criados na fase MTA durante a passagem da primeira mensagem de estado pelos nós dos trechos alternativos. A Figura 4 refere-se às duas abordagens. O agente ANE, ao receber uma mensagem externa de rerroteamento, envia uma mensagem de chaveamento (operação 1) ao agente ARA localizado no primeiro nó do trecho crítico (nó C). No caso da aborda-

gem antecipada, o agente ARA do nó C finaliza essa fase, executando a associação (operação 6) entre o identificador de entrada com o de saída. Já no caso da abordagem sob-demanda as operações adicionais 2 a 5 devem obrigatoriamente ser executadas antes da operação 6.

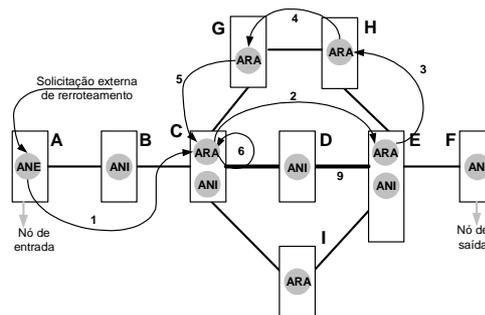


Figura 4 - Fase MR

### 3.3. Protótipo implementado

As estações utilizadas tanto no desenvolvimento dos agentes da arquitetura quanto na fase de testes foram PC's com placas-mãe Asus K6-II. O sistema operacional foi o Linux Red Hat 7.2 com o Kernel 2.4.19 modificado por meio de um *patch* [8] de forma a permitir que as estações funcionassem como roteadores MPLS. Os enlaces que interligam os comutadores MPLS são segmentos Ethernet de 10 Mbit/s dedicados, ou seja, não há compartilhamento. Os agentes foram implementados na forma de threads móveis para oferecer menor sobrecarga computacional ao processador dos comutadores (PC's). A infra-estrutura de mobilidade adotada foi o  $\mu$ Code [6].

### 4. Testes e Análise dos Resultados

O protótipo foi submetido a cinco tipos de teste correspondendo cada um deles às fases descritas na Seção 3. As operações necessárias para cada uma das fases foram agrupadas em subtestes e subfases (tabela 1) com o objetivo de identificar aquelas que são executadas em função dos comprimentos do circuito virtual e do trecho crítico.

Tabela 1: Resultados dos tempos das diversas fases (ms)

Valores	Fase IA			Fase MCV	Fase DTA	Fase MTA		Fase MR				
	1	2	3			1	2	ST 1		ST 2		ST 3
										1	3	A
Tempo médio	846	580	321	33,5	326	27,9	74,7	4	4	90	4,5	122
Desvio Padrão	41	20	4,58	0,35	15	0,71	3,89	0	0	2,87	0,07	3,95
Intervalo de confiança (95%)	11,4	5,44	1,27	0,10	4,24	0,20	1,08	0	0	0,80	0,02	1,09

Tabela 2: Operações das diversas fases

Fase IA	1	Instanciação do agente ANE; instanciação, criação da réplica, serialização do agente ANI, transmissão, remontagem e instalação do agente ANI (operações 1 e 2 - fig. 2) – executada uma única vez.	
	2	Criação da réplica, serialização, transmissão, remontagem, instalação do agente ANI (operações 4 a 7 – fig. 2) – executada em função do comprimento do circuito virtual.	
	3	Obtenção do atraso do enlace <i>downstream</i> ; cálculo do trecho crítico; instanciação, criação da réplica, serialização, transmissão, remontagem e instalação do agente ARA (operações 13 e 14 – fig. 2) – executada uma única vez.	
Fase MCV		Obtenção do atraso do enlace <i>downstream</i> ; montagem, transmissão e tratamento da msg de estado; (operações 8 a 12 – fig. 2) – executada em função do comprimento do circuito virtual (periodicidade: 500ms).	
Fase DTA		Identificação das interfaces ativas; criação da réplica, serialização, transmissão, remontagem e instalação do agente ARA (operações 1 a 3 – fig. 3) – executada em função do comprimento do trecho alternativo.	
Fase MTA	1	Obtenção do atraso e banda do enlace <i>downstream</i> (operações ME 1 a ME 3 – fig. 3) – executada em função do comprimento do trecho alternativo.	
	2	Criação e associação dos rótulos; montagem, transmissão e tratamento da msg de estado (operações 1 a 3 – fig. 3) executada em função do comprimento do trecho alternativo.	
Fase MR	ST1	Transmissão e tratamento da msg de roteamento (solicitação externa de roteamento – fig. 4) – executada uma única vez.	
	ST2	1	Montagem, transmissão e tratamento da msg de chaveamento (operação 1 – fig 4) - executada uma única vez.
		2	Troca de rótulos (operação 6 – fig 3b) – executada uma única vez.
	ST3	A	Transmissão e tratamento da msg de geração e associação de rótulo (operação 2 – fig 4) – executada uma única vez.
		C	Geração e associação de rótulo local; montagem, transmissão e tratamento da msg de fechamento de caminho (operações 3 a 5 – fig 4) – executada em função do comprimento do circuito virtual.

Cada valor de tempo da tabela 1 foi obtido após 50 rodadas de teste, resultando em intervalos de confiança de 95%. A tabela 2 apresenta as principais operações das subfases, relacionando essas operações aos diagramas da seção 3.

As operações referentes à mobilidade dos agentes tiveram uma grande influência nos tempos das fases. A interação dos agentes com os sistemas operacionais dos nós correspondeu de 20% (fase DTA) a 88% (fase MCV) da latência total das fases.

O limite do esquema de roteamento é expresso pelo tempo mínimo necessário para que seja possível efetuar o roteamento de um fluxo ( $t_{mr}$ ). As subfases, cujas latências compõem esse o tempo mínimo, são executadas em função dos comprimentos do circuito virtual ou do trecho alternativo (tabela 2). As expressões para o cálculo de  $t_{mr}$  para a abordagem antecipada e sob-demanda são respectivamente:  $t_{mr} = 613,5c_{lsp} + 428,6c_{ta} + 1587$  e  $t_{mr} = 613,5c_{lsp} + 353,9c_{ta} + 1587$ . Os coeficientes fo-

ram calculados considerando os tempos da tabela 1, a natureza da dependência das fases (tabela 2) e o fato de que o período das mensagens de estado da fase MCV foi arbitrariamente fixado em 500ms. O tempo de execução da fase MR, tempo de chaveamento ( $t_{cv}$ ), totalizou 98ms (soma dos tempos dos subtestes 1 e 2) para a abordagem antecipada. Quando a geração de rótulos sob-demanda é adotada, os tempos das subfases adicionais devem ser considerados. Entretanto, essas fases adicionais tornam o tempo da fase MR ( $t_{cv}$ ) dependente do comprimento do trecho alternativo que é definido pela expressão  $t_{cv} = 122c_{ta} + 102,5$ . Utilizando esta expressão, constata-se que os tempos de chaveamento são inferiores à 834,5 ms para comprimentos de trechos alternativos menores que 6 saltos. Como o GDPA [1] é capaz de detectar tendências de falhas com 5 segundos em média de antecedência à falha, verifica-se que as ações de roteamento são efetuadas de forma transparente às aplicações.

## 5. Trabalhos Relacionados

As abordagens de roteamento adotadas visam reduzir a latência de recuperação da rede após a ocorrência de falhas. Em [7], é desenvolvido um trabalho teórico que se baseia na concatenação de LSP's menores para compor o LSP principal entre o comutador de ingresso e egresso. Este esquema simplifica o roteamento, reduzindo o tempo de recuperação de LSP's.

O esquema proposto em [2] utiliza vários caminhos multiponto/ponto para prover balanceamento de carga e recuperação no caso de falhas de nós. As árvores são construídas de tal forma que o roteamento seja efetuado pelo nó de ingresso. Este esquema acelera o roteamento devido à ausência de sinalização para o redirecionamento do fluxo.

[3] propõe que o roteamento para redes IP sobre WDM seja efetuado tanto na camada IP quanto na WDM. O esquema proposto tenta otimizar as ações de roteamento considerando restrições de topologia, consumo de recurso na rede e características do roteamento nas duas camadas.

## 6. Conclusões e trabalhos futuros

Os resultados dos testes efetuados sob o protótipo evidenciaram que a mobilidade dos componentes da arquitetura juntamente com o esquema adotado de interação desses componentes com os sistemas operacionais dos nós foram os fatores de maior impacto na latência das operações de roteamento. Entretanto, considerando a acentuada discrepância entre as capacidades de processamento das estações utilizadas nos testes e as já existentes no mercado, é possível afirmar que já é factível o uso dessas tecnologias em operações de roteamento. Como sugestões de trabalhos futuros, poderiam ser apontados os estudos da mudança do trecho crítico, da escalabilidade da arquitetura proposta, de outras formas de interação com os sistemas operacionais dos nós, da viabilidade de implementação da arquitetura proposta para outras tecnologias de enlace baseadas em circuito virtual e da extensão da funcionalidade pró-ativa da arquitetura na escolha do trecho alternativo.

## 6. Referências

- [1] Cecilio, E. L., Dumont, A. P., Correia, R. B., Rust, L. F. e Pirmez, L., "Uma Arquitetura de Gerenciamento de Desempenho Pró-ativo Baseado em uma Arquitetura usando Tecnologia Ativa", Anais do 21º SBRC, Natal, RJ, Maio de 2003.
- [2] H. Saito, Y. Miyao e M. Yoshida, "Traffic Engineering using Multiple Multipoint-to-Point LSPs", Info-com'2000.
- [3] Andrea Fumagalli et al, "IP Restoration vs. WDM Protection: Is there an Optimal Choice?", revista IEEE Network, novembro/dezembro de 2002.
- [4] Correia, R. B., Cecilio, E. L., Dumont, A. P., Rust, L. F. e Pirmez, L., "A Viabilidade do Roteamento Pró-ativo em Redes MPLS usando Tecnologia Ativa", Anais do 21º SBRC, Natal, RJ, Maio de 2003.
- [5] Kawamura, R. e Stadler, R., "Active Distributed Management for IP Networks, IEEE Communications Magazine", Abril de 2000.
- [6] Picco, G.P., "µCode: A Lightweight and Flexible Mobile Code Toolkit", Proceedings of the 2nd International Workshop on Mobile Agents 98, setembro de 1998.
- [7] Anat Bremler et al, "Fast Recovery of MPLS Paths", AT&T Labs-Research, 2001.
- [8] James R. Leu, <http://sf.net/projects/mpls-linux/>.