

O Rerroteamento Parcial Pró-ativo em Redes baseadas em Circuito Virtual no Suporte ao Gerenciamento de Desempenho Pró-ativo

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 on virtual-circuit driven networks, in particular MPLS networks, is being considered a key mechanism for providing IP networks with high degree of availability. It also aims at maximizing resource sharing. We present a rerouting scheme to redirect flows proactively through alternative paths with sufficient resources. The rerouting actions are triggered by an external message, signaling the detection of a QoS fault tendency. Typically, such message is generated by a proactive performance management system. Our rerouting scheme stems on active technology based infrastructure and Java. 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 length.*

Resumo. *O rerroteamento de fluxos em redes baseadas em circuitos virtuais, em especial Redes MPLS, vem sendo considerado um eficiente mecanismo para aumentar o grau de disponibilidade e, também, maximizar a utilização de recursos das redes IP. Este artigo apresenta um esquema de rerroteamento com objetivo de redirecionar fluxos por rotas com recursos ociosos pro-ativamente mediante detecção de tendência de falhas de Qos efetuadas externamente por algum sistema de gerenciamento pró-ativo. O esquema utiliza uma infraestrutura baseada em tecnologia ativa e java. Os testes efetuados em um protótipo além de atestar que é possível efetuar o rerroteamento de fluxos de forma transparente às aplicações, possibilitou estabelecer os limites do esquema proposto.*

1. Introdução

Um dos grandes obstáculos na implementação do Gerenciamento de Desempenho Pró-ativo é o fato de que os mecanismos disponíveis em roteadores, comutadores ou estações são implementados de forma monolítica nos sistemas operacionais desses dispositivos. A interação com mecanismos externos é quase inexistente, e quando disponível, é difícil e precária.

Em [1] é proposto o uso da automação das atividades do Gerenciamento de Desempenho Pró-ativo para fazer frente às dificuldades que advêm do porte crescente das redes atuais e, também, dos requisitos funcionais cada vez mais severos que essas redes deverão atender no futuro. Essa automação somente poderá ser materializada se os dispositivos e mecanismos gerenciados oferecerem o suporte ao desencadeamento de ações pelo gerenciamento de desempenho. O ideal é que o gerenciamento de desempenho pró-ativo e os mecanismos dos dispositivos gerenciados interajam de forma flexível, síncrona ou assíncrona e bidirecional.

O Rerroteamento vem sendo considerado um importante componente para dotar as redes IP com um maior grau de tolerância a falhas de enlaces e nós. Apesar de os protocolos IP (OSPF, RIP, BGP) propiciarem uma certa robustez, recalculando as tabelas de roteamento em função de queda de enlaces e nós, os tempos de recuperação de falhas são elevados. Esses tempos podem ser da ordem de dezenas de minutos [2], dependendo do tamanho da rede. Isto inviabiliza o rerroteamento de fluxos de aplicações multimídia em tempo real diante de falhas de QoS. Vários trabalhos têm sido apresentados [3] [4] [5] para minimizar esses tempos de recuperação.

O emprego da tecnologia ativa como base na concepção e implementação dos mecanismos das estações, em especial o rerroteamento, e dos dispositivos de redes disponibiliza o alto grau de flexibilidade necessário na interação com o Sistema de Gerenciamento Pró-ativo. Além disso, ao dotar os nós e estações com recursos computacionais e permitir a execução de programas móveis sob demanda, a tecnologia ativa permite que as tarefas de atualização de código e configuração dos dispositivos sejam efetuadas com maior rapidez.

Este artigo apresenta os resultados dos testes efetuados em um protótipo de uma arquitetura genérica de roteamento pró-ativo anteriormente introduzida em [6]. O objetivo dessa arquitetura é possibilitar o redirecionamento de fluxos de aplicações mediante solicitações externas tais como aquelas geradas pelo Gerenciamento de Desempenho Pró-ativo [1]. Essas solicitações externas são disparadas quando ocorre a detecção de tendência de falha de QoS realizada por esses sistemas externos. A arquitetura de roteamento 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. 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 μ code. 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 conceitos básicos necessários ao entendimento da arquitetura de roteamento e o protótipo; 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. Conceitos básicos

Esta seção apresenta os conceitos básicos que fundamentam a Arquitetura de Rerroteamento Pró-ativo proposta e o desenvolvimento do protótipo: gerenciamento distribuído usando tecnologia ativa, roteamento baseado em QoS, infra-estrutura MPLS e rerroteamento pró-ativo.

2.1. Gerenciamento Distribuído usando Tecnologia Ativa

Com o objetivo de superar as limitações da abordagem centralizada de gerenciamento, vários esquemas distribuídos vêm sendo propostos. A idéia é, ao invés de realizar a coleta das informações para posterior tratamento em um ponto central, levar o processamento para o mais próximo possível dos elementos a serem gerenciados. A hierarquização do gerenciamento [7] [8] [9], dividindo a rede em domínios com cada domínio possuindo uma estação de gerenciamento foi uma das primeiras tentativas. Estudos recentes apontam o uso da tecnologia ativa [10] como um meio eficaz de oferecer maior flexibilidade às tarefas de gerenciamento sem o ônus e complexidade impostas por abordagens que utilizam objetos distribuídos como CORBA e RMI.

A tecnologia ativa [10] é a utilização de pelo menos um entre os paradigmas de agentes móveis, de redes ativas e de redes programáveis tanto na infra-estrutura de rede (roteadores, comutadores

MPLS, Switchers, hubs) quanto nas estações, possibilitando a instalação de programas sob demanda. Esses programas são dotados de capacidade de processamento e realizam o tratamento das informações localmente. Isto diminui o tráfego das mensagens de gerenciamento, economizando banda nos enlaces e propiciando maior rapidez nas ações.

A crescente adoção de serviços multimídia nas redes IP tornou o gerenciamento, em especial, aquele que contempla a área funcional de desempenho [11], um dos componentes mais importantes no provimento de níveis garantidos de QoS. O gerenciamento de desempenho tem a responsabilidade de monitorar os parâmetros de QoS, analisar os dados coletados, desencadear ações quando os níveis dos parâmetros monitorados excedem limites pré-estabelecidos, gerar relatórios, planejar expansão de capacidade instalada em função dos dados coletados e promover o balanceamento de carga para maximizar a utilização de recursos eliminando congestionamentos localizados de enlaces e nós (hot spots). O desencadeamento do processo de adaptação em aplicações multimídia adaptativas [4] e o redirecionamento de um fluxo através de rotas menos congestionadas são exemplos de ações que podem ser disparadas pelo gerenciamento de desempenho.

O gerenciamento de desempenho pode adotar dois tipos de estratégias: a reativa e a pró-ativa. A reativa é aquela em que as ações corretivas são efetuadas após a falha de QoS ter ocorrido. A desvantagem é que as degradações na qualidade das aplicações são sentidas pelos usuários. Na estratégia pró-ativa [13] [14], as ações são efetivadas quando da detecção de tendência de ocorrência de falhas de QoS. O gerenciamento de desempenho pró-ativo é realizado tanto a partir de valores atuais obtidos durante o monitoramento dos parâmetros de QoS, quanto de valores passados que compõem o *baseline* da rede. Com essas informações, valores futuros para os parâmetros monitorados podem ser inferidos por intermédio de algum algoritmo. Caso esses valores futuros sejam maiores que os limites aceitáveis para uma determinada aplicação, um alarme de tendência é gerado. Neste caso, como as ações são realizadas antes da ocorrência da falha de QoS, os efeitos sobre as aplicações podem ser minimizados ou, até mesmo, eliminados. Outra consequência dessa abordagem pró-ativa é que as falhas podem não vir a ocorrer, devido à redução na utilização de recursos em virtude das ações de adaptação de apresentações multimídia ou, então, devido à utilização de rotas alternativas com recursos disponíveis após ações de rerroteamento de fluxos.

2.2. Roteamento

O roteamento IP consiste, basicamente, na execução simultânea de duas tarefas ou processos nos roteadores. A primeira é a construção e atualização da tabela de roteamento. Esta tarefa exige o envio, processamento e recebimento de mensagens de atualização de estado. Essas mensagens contêm as informações necessárias à construção e atualização da tabela de roteamento e são armazenadas em uma base de dados. O algoritmo de roteamento, consultando as informações dessa base de dados, calcula as rotas e constrói ou atualiza a tabela de roteamento. A segunda tarefa é o encaminhamento dos pacotes que chegam ao nó. Assim que um pacote chega, o sistema de roteamento extrai, entre outras informações, o endereço destino para então efetuar a busca na tabela de roteamento e determinar a interface por onde o pacote deve ser enviado.

A primeira tarefa é aquela que distingue o roteamento best-effort daquele baseado em QoS. O tipo best-effort considera no cálculo das rotas apenas o caminho mais curto. O roteamento baseado em QoS calcula os caminhos entre origem e destino considerando uma restrição ou conjunto de restrições de QoS como, por exemplo, retardo, banda e variação do retardo. Dependendo do algoritmo de roteamento utilizado, o caminho obtido pode ser ótimo ou não.

2.3. Comutação de Rótulos - MPLS

A tecnologia MPLS cria um circuito virtual, o Label Switched Path (LSP) para o encaminhamento de pacotes de um fluxo usando apenas a comutação baseada na troca de rótulos. Dessa forma,

é estabelecida uma transmissão orientada à conexão sob uma infra-estrutura IP, que é sem conexão. Uma vez construído o LSP, os pacotes são submetidos à disciplina FIFO de escalonamento de pacotes em todos os roteadores/comutadores (LSR – Label Switch Routers) do LSP, chegando em ordem ao destino.

No esquema IP tradicional, cada roteador toma decisões de encaminhamento independentemente, baseadas apenas nas informações contidas no cabeçalho do pacote IP. Na arquitetura MPLS, os pacotes são tratados de acordo com a posição do comutador no LSP. Os LSR de borda ou Label Edge Routers (LER), em particular, o de entrada, classifica os pacotes de acordo com alguma regra pré-estabelecida, utilizando as informações de nível 3 e/ou de nível 4 do cabeçalho. A classificação consiste em atribuir ao pacote uma classe (FEC – Forwarding Equivalent Class) e associar um número de rótulo que deve ser anexado à frente do cabeçalho IP. A FEC corresponde a um grupo de pacotes que têm o mesmo tratamento nos comutadores pertencentes ao LSP associado a esta FEC.

Os LSR intermediários, ao receberem os pacotes rotulados, realizam o encaminhamento se baseado única e exclusivamente no número do rótulo. É realizada a consulta a uma tabela de rótulos, a Label Information Base (LIB) para determinar a interface de saída em direção ao próximo LSR e também o rótulo de saída que substitui aquele recebido.

O LER de Saída quando recebe um pacote rotulado, retira o rótulo e, em seguida, faz o encaminhamento da forma usual.

O esquema utilizado nas redes MPLS traz diversas vantagens: melhor desempenho no encaminhamento porque os comutadores intermediários não levam em consideração as informações do cabeçalho IP, os pacotes podem ser encaminhados baseados em informações que não estão disponíveis no roteamento IP, garantia de QoS pode ser estabelecida atribuindo prioridades à FEC e suporte à engenharia de tráfego [15] [16].

2.4. Rerroteamento Pró-ativo

A abordagem tradicional adotada no rerroteamento é reativa, uma vez que as ações são realizadas após a ocorrência da falha. No rerroteamento pró-ativo, todas as operações possíveis são realizadas antes de a falha ocorrer. Desta forma, é possível evitar os efeitos negativos produzidos nas aplicações que estão utilizando os serviços de comunicação da rede. No pior caso, esses efeitos são minimizados com tempos de duração reduzidos.

O rerroteamento pró-ativo e o Gerenciamento de Desempenho Pró-ativo (GDPA) [1] podem atuar conjuntamente em prol de uma aplicação multimídia com o objetivo de manter a QoS durante sua apresentação. O GDPA, diante de tendências de falhas de QoS, pode disparar antecipadamente um pedido de rerroteamento do fluxo, corrigindo a falha de QoS no serviço de comunicação, tornando tal evento transparente à aplicação.

O rerroteamento pró-ativo é a ação de substituir um circuito virtual por outro com recursos suficientes para atender os requisitos de QoS do fluxo para o qual foi solicitada a mudança de rota. O caráter pró-ativo da abordagem reside no fato de que todas as ações possíveis são executadas antes de uma solicitação externa de rerroteamento. Essas ações antecipadas reduzem significativamente a latência do rerroteamento, porque no momento da solicitação só resta efetivar a troca de rótulos para desviar o fluxo para a nova rota.

A estratégia básica do rerroteamento pró-ativo é: (i) identificar o maior número possível de caminhos alternativos durante o monitoramento do fluxo; (ii) gerar todos os identificadores locais do circuito virtual nos nós pertencentes aos caminhos alternativos descobertos e (iii) efetuar a troca de rota.

As operações a serem efetuadas dependem da abordagem adotada, que pode ser plena ou 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 roteamento 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. Caso a métrica seja o retardo, o trecho crítico é aquele cujo somatório dos retardos dos seus enlaces tenham um valor tal que, percentualmente, em relação ao retardo fim a fim do circuito virtual, seja significativo.

O uso do roteamento parcial pode ser justificado pelos resultados publicados por G. Apostolopoulos [17], um dos quais apresentando uma variação exponencial do tempo de processamento do cálculo (sob demanda) de rotas em função da distância em saltos entre a origem e destino. A estratégia no roteamento parcial é restringir a busca de caminhos alternativos, a fim de reduzir o tempo de processamento. A vantagem de tal abordagem torna-se acentuada quando os circuitos virtuais são longos. O surgimento de trechos críticos (*hot spot*) em redes IP não é incomum. Eles são o resultado da concentração de tráfego em alguns pontos da rede, que ocorre devido à topologia da rede, ao padrão e à distribuição do tráfego.

O emprego da tecnologia Ativa, dotando os nós com inteligência, flexibilidade e adaptabilidade na forma de agentes móveis, tem o mérito de tornar a tarefa de roteamento de fluxos independente do protocolo de roteamento. Porém, é necessário restringir a difusão de agentes necessários para estabelecimento de caminhos alternativos. A definição do tamanho da área de atuação ou área de busca é um compromisso entre a profundidade desejada para a busca de caminhos alternativos e os recursos consumidos. Uma maior área, se por um lado, aumenta as chances de localizar mais trechos alternativos, por outro, aumenta o consumo de recursos computacionais dos nós e banda dos enlaces da rede.

Outra questão que deve ser apreciada atentamente é a mudança da posição do trecho crítico no circuito virtual durante o tempo de existência do fluxo. Isto pode ocorrer devido à topologia da rede e à variabilidade do tráfego. A mudança da posição do trecho crítico durante o monitoramento do fluxo traz severas complicações na implantação do roteamento parcial. Este fenômeno pode ser visualizado através do movimento do centro da área de busca ao longo do circuito virtual. Este movimento, ao mudar a posição da área de busca, exige o desencadeamento da inundação de agentes em regiões diferentes a todo o momento, provocando um consumo de recursos acentuado.

O roteamento de fluxos de aplicações também apresenta o problema da escalabilidade. Isto ocorre porque, se o número de solicitações de roteamento for elevado, há um consumo excessivo de recursos na rede devido ao aumento de mensagens de controle e dos deslocamentos de agentes para o descobrimento e o estabelecimento de novas rotas.

3. Arquitetura de Roteamento Pró-ativo usando Tecnologia Ativa

Nesta Seção, é proposta uma arquitetura de roteamento parcial pró-ativo de forma a possibilitar o redirecionamento de fluxos de aplicações críticas através de rotas alternativas. A Arquitetura proposta fundamenta-se nos conceitos do roteamento baseado em QoS e pressupõe o uso da tecnologia Ativa e de uma infra-estrutura de rede baseada em circuitos virtuais.

3.1. Descrição Geral da Arquitetura

A Arquitetura de Roteamento Pró-ativo é constituída de três agentes: AgenteNóEntrada (ANE), AgenteNóIntermediário (ANI) e AgenteRotaAlternativa (ARA). O agrupamento das funcionalidades necessárias nesses três agentes visa minimizar o tamanho dos seus códigos e consequentemente os seus tempos de migração e instalação assim como o consumo de memória nos comutadores do circuito virtual ou trechos alternativos.

AgenteNóEntrada (ANE) é o primeiro a iniciar o seu processamento no comutador de entrada do

circuito virtual. Esse componente efetua o gerenciamento do roteamento dos fluxos pertencentes ao mesmo circuito virtual e é o elemento que interage com algum sistema externo.

AgenteNóIntermediário (ANI) possui dois objetivos: (i) identificar os nós pertencentes ao circuito virtual do fluxo a ser roteado; (ii) enviar ao componente ANE hospedado no nó de entrada do circuito virtual os valores das métricas de QoS relevantes ao fluxo dos nós e enlaces do circuito virtual.

AgenteRotaAlternativa (ARA) é especializado na descoberta de trechos alternativos em torno do trecho crítico. O tamanho da região (área de busca) na qual são efetuados os processamentos para a descoberta de trechos alternativos é definido por um parâmetro chamado de raio da área de busca (RAB). O valor desse parâmetro é passado quando criado pelo componente ANE. O componente ARA possui dois outros objetivos: (i) definir o melhor trecho alternativo e (ii) realizar a mudança de rota.

Inicialmente, os três agentes devem ser enviados ao primeiro nó ou comutador do circuito virtual já estabelecido tão logo a monitoração do fluxo da aplicação é iniciada. O primeiro nó é o de entrada do circuito virtual, ou seja, é o primeiro a receber o fluxo proveniente da fonte (servidor de vídeo, etc).

As atividades de roteamento iniciam no momento em que os agentes instalam-se no nó de entrada. Isto caracteriza a abordagem pró-ativa da arquitetura proposta porque essas atividades são realizadas durante a monitoração do fluxo e não após a ocorrência da falha ou tendência de falha de QoS. Quando for sinalizada a falha ou tendência de falha de QoS, a única operação a ser efetuada será a mudança de rota do fluxo através do trecho alternativo previamente descoberto.

O agente ANE é fixado no primeiro nó do circuito virtual visando amenizar o problema de escalabilidade de roteamento de fluxos individuais. A idéia é torná-lo um elemento gerenciador do roteamento de fluxos que pertençam ao mesmo circuito virtual. Desta forma, a migração do agente ANI e a difusão do agente ARA só são efetuadas para a primeira solicitação de roteamento de fluxos que estejam associados a um mesmo circuito virtual.

A arquitetura estabelece que o agente ANE deve implementar uma função de disparo do agente ARA. Nesta função estão configurados os critérios para a criação de novos agentes ARA. Desta forma, é possível evitar ou amenizar os efeitos produzidos devido à mudança da posição do trecho crítico no circuito virtual.

Os agentes ANI foram incluídos com o objetivo de obter as informações dos estados locais dos comutadores do circuito virtual e enviá-las ao agente ANE instalado no nó de entrada. Essas informações de estado são necessárias para que o agente ANE calcule a posição do trecho crítico no circuito virtual.

O roteamento parcial necessita que o trecho crítico pertencente ao circuito virtual seja definido (tarefa efetuada pelo agente ANE) e que trechos alternativos em torno desse trecho crítico sejam descobertos (tarefa efetuada pelo agente ARA). A estratégia de identificação dos trechos alternativos é a difusão limitada de réplicas desse agente na região (área e busca) em torno do trecho crítico. O agente ARA também executa a mudança de rota mediante solicitação do agente ANE.

3.2. Fases do Processo de Roteamento

O funcionamento da Arquitetura de Roteamento Pró-ativo pode ser explicado estabelecendo cinco fases para as operações necessárias ao roteamento 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.

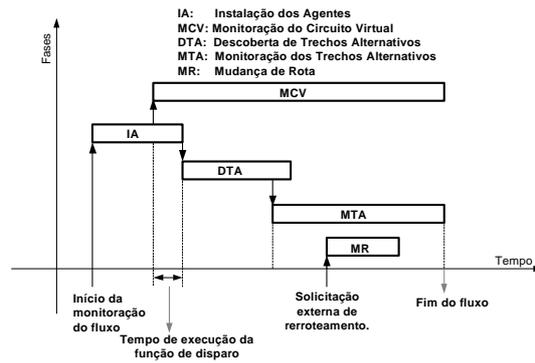


Figura 1 – Fases do rerroteamento

A partir do início do processamento do primeiro agente (ANE) no nó de entrada do circuito virtual, todas as operações referentes a todas as fases, excetuando a fase de mudança de rota (MR), são executadas sequencialmente sem a necessidade de outros eventos externos. Essas fases são chamadas de pró-ativas porque suas operações são concluídas ou iniciadas antes da ocorrência da falha ou tendência de falha de QoS do fluxo. O adiantamento dessas operações é responsável pela redução da latência do rerroteamento. A fase MR é considerada reativa uma vez que sua execução só será efetuada após solicitação de rerroteamento do sistema externo. Apesar de reativa, considerando como referencial o rerroteamento, a fase MR pode ser executada de forma pró-ativa em relação ao fluxo caso a solicitação de rerroteamento externa seja gerada devido à detecção de uma tendência de falha de QoS. Esta situação ocorre quando o sistema externo que está utilizando o rerroteamento tiver funcionalidades pró-ativas como é o caso do Gerenciamento de Desempenho Pró-ativo (GDPA) [6]. A Figura 1 mostra que as fases MCV e MTA, uma vez iniciadas, são executadas durante a existência do fluxo. Essas fases não influenciam no tempo de chaveamento do fluxo.

3.2.1. Instalação dos agentes (IA)

A Figura 2 detalha a seqüência das operações durante a instalação dos agentes de rerroteamento no circuito virtual. O exemplo da Figura 2 contempla um circuito virtual com seis comutadores.

Assim que o agente ANE inicia o seu processamento, cria o agente ANI (operação 1). O agente ANI, recém criado, migra (operação 2) para o próximo nó do circuito virtual (nó B). A operação 3 indica que o agente ANI após deslocar-se para o próximo nó, interrompe o seu processamento naquele nó (nó A). O agente ANI, no nó B (operação 4), gera uma réplica de seu código que migra para o próximo nó (nó C). As operações 5, 6 e 7 são similares à operação 4.

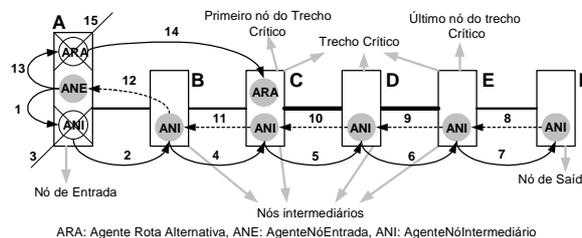


Figura 2 – Seqüência de operações da fase IA

As operações 8 a 12, que não fazem parte da fase de instalação dos agentes, correspondem às mensagens de estado do circuito virtual. Essas operações foram mostradas na Figura 6 porque são os eventos que causam a criação do agente ARA pelo agente ANE. O agente ARA, no nó A, migra (operação 14) imediatamente para o primeiro nó do trecho crítico (nó C). A operação 15 indica que o agente ARA após migrar para o primeiro nó do trecho crítico encerra seu processa-

mento naquele nó (nó A).

3.2.2 Monitoração do Circuito Virtual (MCV)

A fase MCV inicia assim que o agente ANI instala-se no nó de saída do circuito virtual (nó F). O seu término se dá juntamente com o término do fluxo (Figura 1). Durante essa fase, são realizadas as operações necessárias para disponibilizar as informações de estado do circuito virtual ao agente ANE hospedado no nó de entrada do circuito virtual (nó A, Figura 2). Essas informações de estado são utilizadas pelo agente ANE no cálculo da posição do trecho crítico no circuito virtual. O esquema utilizado consiste no envio dessas mensagens a partir do agente ANI do nó de saída até ao agente ANE, passando por todos os agentes ANI instalados nos nós intermediários (operações 8 a 12, Figura 2).

3.2.3. Descoberta de Trechos Alternativos (DTA)

A fase de descoberta de trechos alternativos inicia quando o agente ARA instala-se no primeiro nó do trecho crítico. O encerramento desta fase se dá quando a última cópia do agente ARA instala-se no último nó do trecho crítico, indicando a descoberta do último trecho alternativo. A Figura 3(a) detalha o posicionamento dos agentes nos diversos comutadores e a seqüência de operações da fase DTA em um cenário em que existem dois trechos alternativos na área de busca.

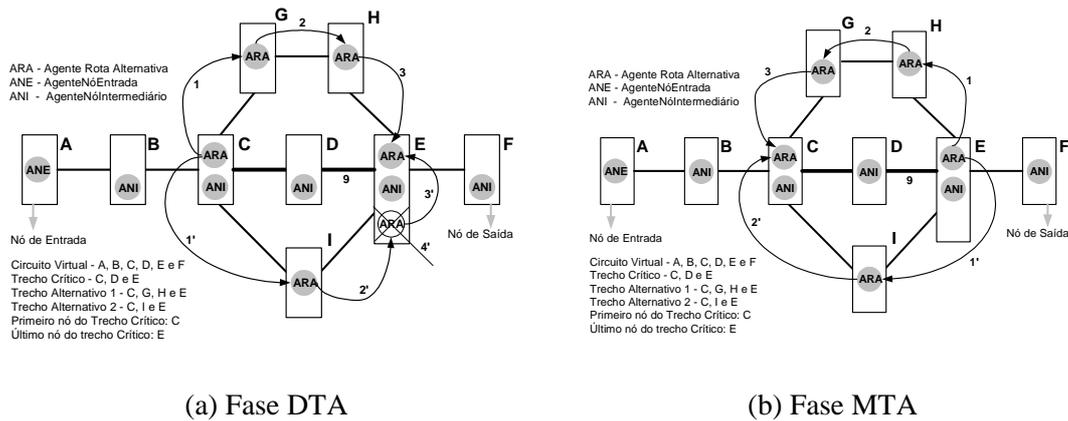


Figura 3 – Seqüência de operações da fase DTA

O agente ARA no primeiro nó do trecho crítico (nó C) cria uma cópia do seu código que migra para o vizinho pertencente ao trecho alternativo 1 (nó G). Esse processo de criação e migração de réplicas para os respectivos nós vizinhos ocorre até que uma das cópias do agente ARA atinja o último nó do trecho alternativo (nó E). Quando a segunda cópia do agente ARA alcança o último nó do trecho crítico (operação 2''), anunciando a descoberta do trecho alternativo 2, envia uma mensagem ao agente ARA já instalado (operação 3''). Essa mensagem contém o número da interface pela qual chegou e o comprimento do trecho alternativo 2. Em seguida, encerra o seu processamento (operação 4'').

3.2.4. Monitoração dos Trechos Alternativos (MTA)

A fase MTA contém as operações necessárias para a obtenção do estado local dos nós dos trechos alternativos e ao envio periódico das mensagens de estado. A origem dessas mensagens é o agente ARA do último nó do trecho crítico (nó E) e o destino é o agente ARA do primeiro nó do trecho crítico (nó C). Essas mensagens são enviadas através de todos os trechos alternativos descobertos, passando por todos os nós desses trechos. As informações de estado contidas nessas mensagens permitem ao agente ARA do último nó do trecho crítico escolher o melhor trecho alternativo com recursos suficientes para atender as restrições de QoS do fluxo a ser rerroteado. A Figura 3 (b) apresenta o esquema adotado para o envio das mensagens durante essa fase. Vale

ressaltar que os identificadores locais dos circuitos virtuais alternativos (trechos alternativos) somente são gerados quando a primeira mensagem é enviada considerando a abordagem sob demanda de geração dos identificadores locais do circuito virtual (rótulos).

3.2.5 Mudança de Rota (MR)

A fase mudança de rota compreende o conjunto de operações executadas entre o recebimento da solicitação externa de roteamento pelo agente ANE e o redirecionamento do fluxo através do trecho alternativo. O redirecionamento é efetuado pelo agente ARA hospedado no primeiro nó do trecho crítico.

As operações envolvidas durante a fase de mudança de rota 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 durante a passagem da primeira mensagem de estado pelos nós dos trechos alternativos. Esta abordagem minimiza o tempo de chaveamento porque a fase MR fica praticamente restrita a uma operação de associação de identificadores. Entretanto, a desvantagem é que impõe um consumo elevado de identificadores locais, podendo tornar a operação dos comutadores crítica.

A Figura 4 (a) refere-se à abordagem antecipada. O agente ANE, ao receber uma mensagem externa de roteamento, envia uma mensagem de chaveamento (operação 1) ao agente ARA localizado no primeiro nó do trecho crítico. Esse agente finalmente executa a associação (operação 2) entre o identificador de entrada com o de saída. O identificador local de saída é aquele que corresponde ao trecho alternativo escolhido.

Para contornar a desvantagem do consumo excessivo de identificadores, embora aumentado o tempo de execução da fase MR, a segunda abordagem pode ser adotada. Nesse caso, a geração e associação dos identificadores locais são feitas sob demanda, ou seja, os nós pertencentes aos trechos alternativos só geram os identificadores quando ocorre a solicitação externa de roteamento. A Figura 4 (b) detalha o desencadeamento das operações para essa abordagem. As operações adicionais (operações 2 a 5) são necessárias para efetuar a geração e associação dos identificadores locais dos nós do trecho alternativo escolhido.

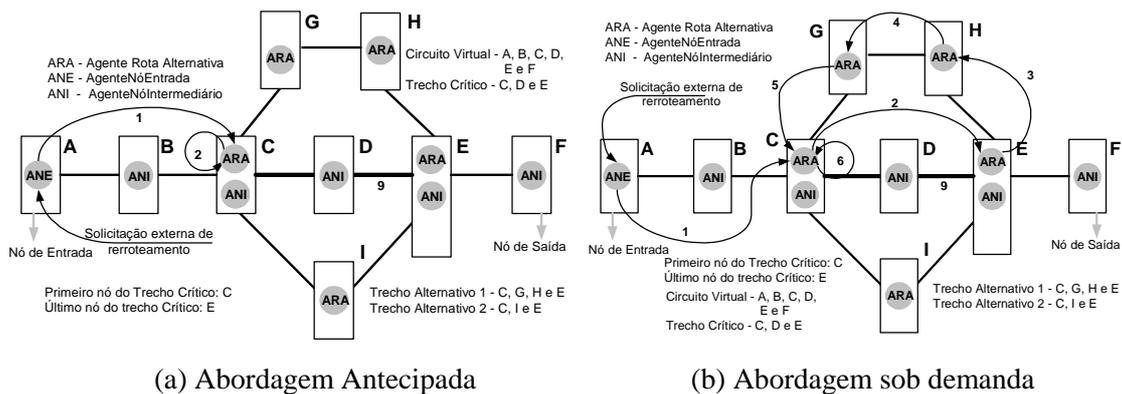


Figura 4 – Sequência de operações da fase MR

3.3. Protótipo implementado

O objetivo deste protótipo é verificar a viabilidade do roteamento de fluxos de aplicações multimídia sobre uma infra-estrutura de rede MPLS mediante solicitação do Sistema de Gerenciamento de Desempenho Pró-ativo [6]. Neste cenário, os três agentes da arquitetura proposta são especializados para o roteamento de fluxos com restrições de banda e retardo.

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.

Os enlaces que interligam os comutadores MPLS são segmentos Ethernet de 10 Mbit/s dedicados, ou seja, não há compartilhamento.

A implementação do protótipo da arquitetura foi feita em Java e seguiu os princípios da orientação a objetos e da orientação a componentes. O pacote JDK versão 1.3.1 foi instalado em todas as estações que compõem a rede de teste.

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 μ Code [18], por possuir características não encontradas em outros ambientes como, por exemplo, o Aglets Software Development Kit (ASDK) [19]. O nível de segurança na movimentação de código, provimento de primitivas básicas de mobilidade, mobilidade gradativa de código e baixa complexidade relativa são características vitais para o desenvolvimento de mecanismos na camada de rede.

4. Testes e Análise dos Resultados

Os agentes de roteamento foram especificamente projetados e implementados para atender fluxos com restrição de banda e retardo. Então, os componentes AgenteNóEntrada (ANE), AgenteNóIntermediário (ANI) e AgenteRotaAlternativa (ARA) passam a ser designados LerEntradaBandaRetardo (LEBR), LsrBandaRetardo (LBR) e RotaAlternativaBandaRetardo (RABR) respectivamente.

Os testes foram realizados com o objetivo de verificar a viabilidade do roteamento de fluxos em função dos resultados dos testes realizados na Arquitetura de Gerenciamento de Desempenho Pró-ativo [1]. Além disso, os testes permitiram que fossem avaliadas as consequências do uso de uma linguagem interpretada como Java executando operações na camada de rede. Finalmente, os testes possibilitaram, através de extrapolações, determinar os limites dentro dos quais é possível, em tempo hábil, concretizar o chaveamento de fluxos.

4.1. Testes efetuados

O protótipo foi submetido a cinco tipos de teste correspondendo cada um deles às fases descritas na Seção 3: (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).

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

Med.	Fase IA			Fase MCV	Fase DTA	Fase MTA		Fase MR					
	1	2	3			1	2	ST			ST		
								1	1	3	A	C	
T. Med.	846	580	321	33,5	326	27,9	74,7	4	4	90	4,5	122	
D. P.	41	20	4,58	0,35	15	0,71	3,89	0	0	2,87	0,07	3,95	
T. Min.	698	544	312	32,5	303	27,0	72,0	4	4	87	4,5	116	
T. Max.	941	632	330	35,0	366	29,0	78,0	4	4	97	5,0	132	
Int. conf. (95%)	11,39	5,44	1,27	0,10	4,24	0,20	1,08	0	0	0,80	0,02	1,09	

As operações necessárias para cada uma das fases foram agrupadas em subfases com objetivo de identificar aquelas que são executadas em função dos comprimentos do LSP e do trecho crítico. Desta forma, foi possível determinar os limites do esquema de roteamento adotado em função desses comprimentos. A Tabela 1 apresenta os tempos médios, mínimos e máximos das subfases

das cinco fases necessárias ao roteamento.

4.2. Análise dos Testes

Nesta seção, os resultados foram analisados considerando três pontos distintos: os diversos fatores que influenciam nos tempos das diversas fases; o impacto da interação dos agentes da arquitetura de roteamento com os sistemas operacionais dos nós e os limites do uso do protótipo implementado.

4.2.1. Tempos de execução das fases

Várias subfases referentes ao encaminhamento IP tradicional devem obrigatoriamente ser realizadas tanto para o envio dos agentes LBR e RABR quanto das mensagens de estado e controle. Entretanto os seus respectivos tempos não foram incluídos na tabela 1 porque a metodologia adotada nos testes não permitiu medir esses tempos. Empregando o utilitário ping para estimar esse tempo com as tabelas de roteamento povoadas com 1000 rotas estáticas, obteve-se um retardo (RTT) da ordem de 1,8 ms para uma distância de um salto. Então, a título de simplificação, os tempos dessas fases foram desconsiderados em virtude de seus valores serem bastante inferiores aos das outras fases (menores que 1 %).

O tamanho das mensagens de estado e controle é um fator que contribui nos tempos das fases MCV e MTA. Adotou-se um tamanho fixo de 87 octetos (pior caso) para simplificar a análise dos resultados e evitar a fragmentação em nível IP. Essa situação simplificada implica em tempos de transmissão das mensagens na ordem de 69,6 μ s no enlace Ethernet e portanto bem inferiores aos tempos obtidos nas outras fases.

A operação de cálculo do trecho crítico da subfase 3 do teste de instalação dos agentes depende, também, do comprimento do LSP, devido ao número de iterações do algoritmo variar em função do número de enlaces e quantidade de métricas envolvidas. Estabeleceu-se que o algoritmo foi sempre executado para o caso no qual existem 10 saltos no LSP, independentemente do comprimento real do LSP do teste.

O tempo de execução da subfase 1 (846 ms) do teste IA (Tabela 1) é cerca de 45,8 % superior ao da subfase 2 (580 ms) do mesmo teste. A razão desta diferença é que, na subfase 1, são efetuadas duas operações (instanciação do agente LBR e RABR) com alto consumo de tempo de CPU enquanto, na subfase 2, é realizada somente uma (remontagem do agente LBR). Entretanto, o impacto da subfase 1 é atenuado porque só é executada uma única vez ao passo que as operações da subfase 2 são efetuadas em função do comprimento do LSP.

4.2.2. Interação com o sistema operacional

Existem basicamente dois tipos de subfases: aquelas envolvidas na migração de agentes e aquelas relacionadas com o envio de mensagens.

No primeiro tipo, a criação dos agentes (instanciação), a remontagem dos agentes no destino e o acesso ao sistema operacional do nó são as operações que mais contribuem para as latências dessas fases. Foi constatado que o tempo de reconstrução das estruturas de dados dos agentes no destino é significativamente maior do que o tempo de serialização e de envio. O método `enviaAgente()` responsável por tal serialização e transmissão apresentou tempo médio de execução igual a 77,5 ms significativamente inferior aos tempos de migração dos agentes LBR e RABR (subfases 1 e 2 da fase IA e fase DTA).

Os métodos que interagem com os sistemas operacionais das estações utilizadas nos testes executam scripts desenvolvidos em Perl para obter informações das tabelas de roteamento e rótulos, das interfaces locais ou para efetuar a geração e associação de rótulos.

As cópias do agente LBR durante as subfases 1 e 2 do teste de instalação dos agentes invocam o

método `proxSaltoCircuitoVirtual()` para descobrir o endereço IP do próximo nó do LSP do fluxo, migrando em seguida. Esse método recebe como parâmetro o endereço IP destino do fluxo monitorado (LER de Entrada) ou o rótulo de entrada (LSR ou LER de Saída). O script associado consulta as tabelas de roteamento e rótulos para depois retornar um vetor contendo o endereço IP da interface de saída local, o endereço IP do próximo nó e o rótulo de saída. Caso o valor do rótulo seja zero, o agente LBR conclui que está no LER de Saída. Considerando o tempo de execução da subfase 2 do teste IA (580ms), esse método cujo tempo médio de execução foi de 61,3 ms tem um peso elevado correspondendo a mais de 10 % do tempo total desta fase. Os 518,7 ms restantes são referentes principalmente a serialização e remontagem do agente LSR.

O método `retornaInterfacesAtivas()` é invocado pelo agente RABR independentemente em que nó esteja instalado. Esse método retorna todas as interfaces ativas do nó corrente em um vetor, utilizando um script que extrai os endereços IP das interfaces locais das informações geradas pelo comando `/sbin/ifconfig -a`. A relação de interfaces locais obtidas por esse método permite ao agente RABR identificar que tipo de nó está instalado (LER de Entrada, primeiro nó do trecho crítico, último nó do trecho crítico ou nó intermediário do trecho crítico) e adequar o seu comportamento. O tempo médio de execução desse método (67 ms) é bastante elevado porque corresponde a 20 % do tempo total da fase DTA (326 ms).

As subfases do segundo tipo também possuem operações que interagem com o sistema operacional. A fase MCV, por exemplo, consiste basicamente na obtenção do retardo do enlace e envio da mensagem. Nessa operação, o método `obtemEstadoLocal()` executa um script para ler os valores de banda ou retardo de um arquivo texto contendo esses valores para todas as interfaces do nó. Esse arquivo foi gerado manualmente de forma que a primeira, segunda e terceira colunas correspondessem respectivamente aos endereços IP das interfaces locais, aos retardos e às bandas disponíveis dos enlaces associados a essas interfaces locais. Em situação normal, processos locais e específicos para medir esses parâmetros seriam os responsáveis pela criação e atualização desse arquivo. A contribuição dessa operação no tempo médio da fase MCV (33,5 ms – Tabela 2) é de 88 %, considerando que o tempo médio para enviar uma mensagem de estado é de 4 ms. Nas subfases dos testes MTA e MR, os percentuais de participação desse tipo de operação são também similares.

4.2.3. Limites do rerroteamento

O tempo mínimo necessário para que seja possível efetuar o rerroteamento de um fluxo é o tempo transcorrido entre o início da execução do agente LBR no LER de Entrada (início da monitoração do fluxo pelo inspetor de desempenho do GDPA[1]) e o registro do primeiro trecho alternativo válido realizado pelo agente RABR instalado no primeiro nó do trecho crítico. Esse tempo corresponde ao somatório da latência da fase de Instalação dos Agentes, do tempo de execução da fase de Descoberta de Trechos Alternativos (DTA), do tempo necessário para o recebimento de três mensagens de estado do circuito virtual na fase MCV (função de disparo) e para o recebimento da primeira mensagem de estado do trecho alternativo na fase MTA (considerando que a primeira mensagem corresponde a um trecho alternativo válido). Esse valor define o tempo de duração mínimo do fluxo para que o rerroteamento possa ser realizado.

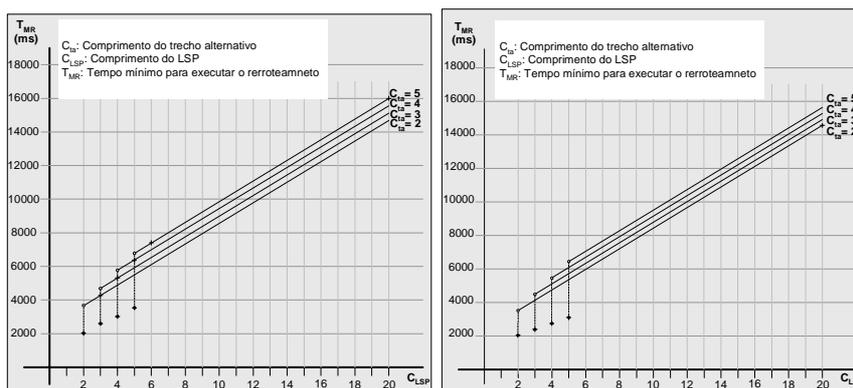
Conforme mencionado anteriormente, algumas subfases são executadas em função dos comprimentos do LSP e trechos alternativos, tornando o tempo mínimo para rerroteamento também dependente desses parâmetros. A expressão $t_{mr} = \sum t_{fi} c_{lsp} + \sum t_{fa} c_{ta} + \sum t_{fi}$ estabelece a relação entre o tempo mínimo para rerroteamento e os comprimentos do LSP e do trecho alternativo onde: (i) t_{mr} é o tempo mínimo para efetuar o rerroteamento; (ii) $\sum t_{fi}$ é o somatório dos tempos das subfases que são executadas em função do comprimento do LSP; (iii) c_{lsp} é o comprimento do LSP; (iv) $\sum t_{fa}$ é o somatório dos tempos das subfases que são executadas em função do comprimento do trecho alternativo; (v) c_{ta} é o comprimento do trecho alternativo e (vi) $\sum t_{fi}$ é o somatório dos tempos das subfases que independem dos comprimentos do LSP e do trecho alternativo.

A Tabela 2 apresenta os valores dos coeficientes da expressão acima de acordo com os resultados dos testes (Tabela 1) e a abordagem de geração de rótulos. Foi também considerado no cálculo desses coeficientes que o agente RABR só é criado pelo agente LEBR após receber três mensagens de estado do circuito virtual consecutivas (função de disparo). O período dessas mensagens de estado foi arbitrariamente fixado em 500 ms.

Tabela 2 – Parâmetros da equação do tempo mínimo

Antecipada			Sob demanda		
Σt_{fi} (ms/salto)	Σt_{fa} (ms/salto)	Σt_{fi} (ms)	Σt_{fi} (ms/salto)	Σt_{fa} (ms/salto)	Σt_{fi} (ms)
613,5	428,6	1587	613,5	353,9	1587

De posse desses coeficientes é possível relacionar o tempo mínimo para o roteamento e os comprimentos do LSP e do trecho alternativo. Então, fixando-se o comprimento do trecho alternativo obtém-se uma equação de reta, tornando o tempo mínimo para roteamento função somente do comprimento do LSP. A Figura 5 (a) apresenta graficamente as equações de retas para diversos valores de comprimento do trecho alternativo, considerando a geração de rótulo antecipada. As retas são paralelas porque os seus coeficientes angulares independem do comprimento do trecho alternativo enquanto os seus coeficientes lineares são função do comprimento do trecho alternativo. Vale ressaltar que os pontos de descontinuidade do gráfico ocorrem porque quando o trecho alternativo coincide com o LSP (o trecho crítico é o próprio LSP) as fases de instalação dos agentes (IA) e de monitoração do caminho crítico (MVC) deixam de existir.



(a) Geração de Rótulos antecipada (b) Geração de Rótulos sob demanda

Figura 5 – Tempo mínimo para roteamento

Nestes casos, o agente RABR assim que é criado pelo agente LEBR no LER de Entrada, inicia imediatamente a fase de descoberta de trechos alternativos (DTA) porque o LER de Entrada passa a ser também o primeiro nó do trecho crítico (roteamento pleno). Observando a Figura 5 (a) é possível determinar qual é o tempo mínimo necessário para efetuar o roteamento ou o tempo mínimo de duração do fluxo que pode ser roteado na abordagem antecipada. Por exemplo, em um LSP de comprimento igual a 14 saltos, somente fluxos acima de 11,0, 11,5, 11,9 e 12,3 s podem ser roteados, considerando comprimentos de trechos alternativos iguais a 2, 3, 4 e 5 saltos respectivamente. A Figura 5.(b) apresenta os gráficos para a abordagem de geração de rótulos sob demanda. Os valores são menores porque a subfase 2 da fase MTA só é executada após solicitação externa e durante a fase MR.

A fase de mudança de rota é executada após a solicitação de roteamento do Gerenciamento de Desempenho Pró-ativo. Esta fase é considerada reativa porque sua execução é efetuada após a falha de QoS ou a detecção de tendência de falha. No caso da geração de rótulo antecipada, o tempo total para esta fase é o somatório dos tempos dos subtestes 1 e 2, totalizando 98 ms.

Quando a geração de rótulos sob demanda é adotada, os tempos das subfases adicionais devem ser considerados. Entretanto, o número de execuções dessas subfases depende do comprimento

do trecho alternativo. A expressão $t_{cv} = \Sigma t_{fa} c_{ta} + \Sigma t_{fi}$ define essa dependência. Os coeficientes são: (i) Σt_{fa} : somatório dos tempos das subfases adicionais que são executadas de acordo com o comprimento do trecho alternativo e (ii) Σt_{fi} somatório dos tempos das subfases adicionais que independem do comprimento do trecho alternativo.

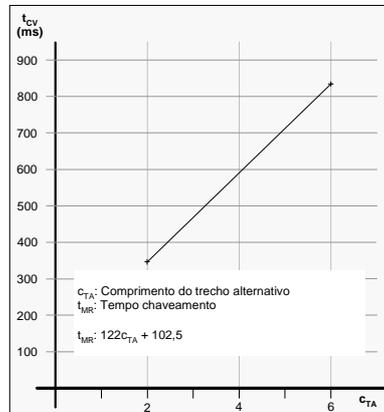


Figura 6 – Tempo de chaveamento (geração de rótulos sob demanda)

Utilizando os valores médios da Tabela 1, obtém-se o gráfico da Figura 6. Neste gráfico, os valores dos tempos de chaveamento são inferiores a 834,5 ms para comprimentos de trechos alternativos menores que 6 saltos, comprovando que é possível o redirecionamento de fluxos mediante solicitação do Gerenciamento de Desempenho Pró-ativo (GDPA) antes da ocorrência de falha de QoS. O GDPA [1] é capaz de detectar tendências de falhas com 5 segundos em média de antecedência à falha e portanto acima dos 834,5 ms necessários ao rerroteamento. Desta forma, é viável que ações de rerroteamento sejam efetuadas de forma transparente às aplicações

4. Trabalhos Relacionados

O uso de agentes móveis em tarefas relacionadas ao roteamento dinâmico é investigado em alguns trabalhos recentes [20] [21] [22]. [23] emprega agentes móveis na construção de rotas multiponto/ponto (árvores) com restrições de QoS para implementar a agregação de FEC's prevista na arquitetura MPLS. [24] confronta o desempenho dos algoritmos de roteamento *shortest path first* (SPF) e Bellman-Ford (BF) com um algoritmo baseado em agentes móveis chamado Ant-Net. Em [25], é proposta uma aplicação utilizando agentes móveis e CORBA para o gerenciamento das operações de roteamento. O objetivo é estabelecer rotas em função da QoS previamente estabelecida em contratos de Serviço (SLA).

O rerroteamento tem sido pesquisado com o objetivo de tornar as redes de comutação de pacotes mais resistentes a falhas de enlaces e nós. As abordagens adotadas visam também reduzir a latência de recuperação da rede após a ocorrência de falhas. Em [26], 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 rerroteamento, reduzindo o tempo de recuperação de LPS's.

O esquema proposto em [3] 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 rerroteamento seja efetuado pelo nó de ingresso. Este esquema acelera o rerroteamento devido à ausência de sinalização para o redirecionamento do fluxo.

Scott [27] apresenta um algoritmo para fornecer serviços robustos com garantias de QoS sobre uma infra-estrutura de rede IP com nós e enlaces suscetíveis à falhas. O algoritmo proposto calcula múltiplos caminhos com o menor número possível de elementos (nós e enlaces) em comum, minimizando assim o impacto das falhas nos serviços.

Em [4], é salientado que o rerroteamento pode ser empregado tanto na camada de rede quanto na camada física e de enlace. O rerroteamento na camada de rede tem a vantagem de poder ser realizado com uma menor granularidade, permitindo que serviços específicos sejam atendidos. A desvantagem é um tempo de recuperação elevado. Nas camadas física e de enlace, em especial as de redes orientadas a circuito, os tempos de recuperação são reduzidos. A contrapartida é que o rerroteamento, quando efetuado, atinge uma grande quantidade de fluxos simultaneamente. [5] propõe que o rerroteamento para redes IP sobre WDM seja efetuado tanto na camada IP quanto na WDM. O esquema proposto tenta otimizar as ações de rerroteamento considerando restrições de topologia, consumo de recurso na rede e características do rerroteamento nas duas camadas.

5. Conclusões e trabalhos futuros

O propósito principal da Arquitetura de Rerroteamento Pró-ativo apresentada neste trabalho é permitir o redirecionamento de fluxos de aplicações críticas mediante solicitações externas. Esses eventos externos podem ser oriundos de detecção de falhas de QoS ou de tendência de falhas. A arquitetura proposta viabiliza o rerroteamento pró-ativo uma vez que todos os procedimentos necessários à descoberta de rotas alternativas podem ser efetuados em tempo de monitoramento do fluxo, ou seja, somente as operações relativas ao chaveamento do fluxo são realizadas após a solicitação de redirecionamento externo. O rerroteamento é realizado sob uma infra-estrutura de rede baseada em circuitos virtuais, de rede com roteamento baseado em QoS e de rede com tecnologia ativa.

As justificativas da adoção do rerroteamento parcial foram apresentadas assim como as suas limitações. Os resultados dos testes efetuados sob o protótipo evidenciaram que a mobilidade dos componentes da arquitetura juntamente com esquema adotado de interação desses componentes com os sistemas operacionais dos nós foram os fatores de maior impacto na latência das operações de rerroteamento. 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 rerroteamento.

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 pelo agente ARA instalado no primeiro nó do trecho crítico, inferindo valores futuros das métricas de QoS desses trechos alternativos descobertos.

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] Labovitz, C. et al, “Delayed internet routing convergence”, In Proc. ACM SIGCOMM '00 pp. 175–187, Estocolmo, Suécia, 2000.
- [3] H. Saito, Y. Miyao e M. Yoshida, “Traffic Engineering using Multiple Multipoint-to-Point LSPs”, Info-com'2000.
- [4] Autenrieth, A. e Kirstdter, “A. Fault-Tolerance and Resilience Issues in IP-Based Networks, Second International Workshop on the Design of Reliable Communication Networks” (DRCN), abril de 2000
- [5] Andrea Fumagalli et al, “IP Restoration vs. WDM Protection: Is there an Optimal Choice?”, revista IEEE Network, novembro/dezembro de 2002.
- [6] Correia, R. B., Cecilio, E. L., Dumont, A. P., Rust, L. F. e Pirmez, L., “A Viabilidade do Rerroteamento Pró-ativo em Redes MPLS usando Tecnologia Ativa”, Anais do 21º SBRC, Natal, RJ, Maio de 2003.
- [7] Schmidt, K. J. e Maura, D., “SNMP Essencial”, Ed Campus, 2001.

- [8] Goldszmidt, G. e Yemini, Y., “Distributed Management by Delegation”, 15a conferência internacional sobre sistemas de computação distribuídos, 1995.
- [9] ITU-T, Rec X.739, “Information Technology – Open Systems Interconnection, Systems Management Functions – Metric Objects and Attributes”, ITU-T, 1992.
- [10] Kawamura, R. e Stadler, R., “Active Distributed Management for IP Networks, IEEE Communications Magazine”, Abril de 2000.
- [11] BRISA (Sociedade Brasileira para Interconexão de Sistemas Abertos), Arquiteturas de Redes de Computadores OSI e TCP/IP, Makron Books, 1994.
- [12] Gomes, R.L., “Autoria e Apresentação de Documentos Multimídia Adaptativos em Redes, Dissertação de Mestrado”, NCE/IM/UFRJ, 2001, disponível em <http://www.nce.ufrj.br/labnet>.
- [13] Data Communications, “Proactive LAN Management, Data Communications Magazine”, março de 1993.
- [14] Franceschi, A.S.M., Rocha, M.A., Weber, H.L. e Westphall, C.B., “Proactive Network Management Using Remote Monitoring and Artificial Intelligence Techniques”, Proceedings of the 2nd IEEE Symposium and Communications (ISCC’97), junho de 1997.
- [15] D. Awduche et al, “Overview and Principles of Internet Traffic Engineering”, RFC-3272, IETF, maio de 2002.
- [16] D. Awduche et al, “Requirements for Traffic Engineering Over MPLS”, RFC-2702, IETF, setembro de 1999.
- [17] G. Apostolopoulos et al, “Intradomain QoS Routing in IP Networks: A Feasibility and Cost/Benefit Analysis”, IEEE Network Magazine, 1999.
- [18] Picco, G.P., μ Code: “A Lightweight and Flexible Mobile Code Toolkit”, Proceedings of the 2nd International Workshop on Mobile Agents 98, setembro de 1998.
- [19] Oshimm, M., Karjoth, G., “Aglets Specification, disponível na INTERNET” via www.url:www.trl.ibm.co/aglets/spec_alpha.html.
- [20] K. Oida e M. Sekido, “ARS: An Efficient Agent-Based Routing System for QoS Guarantees, Comp. Commun”, vol. 23, 2000, pp. 1437-47.
- [21] M. Dorigo e G. Di Caro, “Mobile Agentes for Dynamic Routing”, Proc. 31st Int’l. Conf. Sys. Hawaii, janeiro de 1998.
- [22] N. Minar et al., “Cooperating Mobile Agents for Dynamic Network Routing”, Software Agents for Future Communications Systems, Springer-Verlag, 1999.
- [23] Sergio Gonzalez e Victor C. M. Leung, “QoS Routing for MPLS Networks Employing Mobile Agents”, revista IEEE Network, Maio/Junho de 2002.
- [24] Gianni Di Caro e M. Dorigo, “Mobile Agents for Adaptive Routing”, Proc. 31st Hawaii International Conference on System Sciences, IEEE Computer Society Press, Los Alamitos, CA, pp. 74-83, 1998.
- [25] Angélica Reyes et al, “Routing Management Application Based on Mobile Agents on the Internet2”, EUNICE 2000, setembro 2002.
- [26] Anat Bremler et al, “Fast Recovery of MPLS Paths”, AT&T Labs-Research, 2001.
- [27] Scott Seongwook Lee and Mario Gerla, “Fault tolerance and load balancing in QoS provisioning with multiple MPLS paths”, Lecture Notes in Computer Science, vol. 2092, 2001.