Universidade Federal do Rio Grande do Norte
Centro de Tecnologia
Programa de Pós-graduação em Engenharia Elétrica

ESPECIFICAÇÃO DE UMA REDE MPLS FIM-A-FIM COM DIFERENCIAÇÃO DE SERVIÇOS

EDSON MOREIRA SILVA NETO

Natal
Agosto – 2006

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ESPECIFICAÇÃO DE UMA REDE MPLS FIM-A-FIM COM DIFERENCIAÇÃO DE SERVIÇOS

Edson Moreira Silva Neto

Tese apresentada à Universidade Federal do Rio Grande do Norte, como parte das exigências do Curso de Doutorado em Ciências em Engenharia Elétrica, área de concentração em Sistemas Distribuídos, para obtenção do título de “Doutor em Ciências em Engenharia Elétrica”.

Orientador:

Prof. D. Sc. Sergio Vianna Fialho

Natal
Agosto – 2006

Ao meu Deus e Senhor; e
À minha família:
Sydna, minha princesa amada;
Victor Matheus, presente de Deus para minha vida;
Edson Filho, a bênção dobrada do Senhor.

AGRADECIMENTOS

A Deus. Tenho que começar por Ele. O que seria de mim se não fora o Senhor? É ele quem me concede graciosamente o ar que respiro, o alimento que sacia a minha fome, por Ele existo. Deus tem sido o meu paizinho querido, Ele sempre tem suprido as minhas necessidades e expectativas, muito além daquilo que penso ou peço.

À minha família: Sydna, Victor e Edson Filho. Nestes cinco anos de estudo, muitas foram as dificuldades e os desafios a serem vencidos, mas a compreensão de Sydinha nos momentos de trabalho intenso foi imprescindível, só assim consegui galgar mais esse degrau. Os meus filhos, nascidos durante esta pós-graduação, tornaram-se verdadeiros "anti-oxidantes". Queridos Vitinho e Edinho, hoje vocês ainda não entendem o que agora deixo registrado, mas quantas vezes ao chegar em casa com os "neurônios fumaçando", o sorriso banguelo de vocês aquietaram minha mente e renovaram as minhas energias de forma surpreendente.

Aos meus pais, José Martins e Adna, que desde a minha meninice me encaminharam aos estudos. Sei que vocês abriram mão de benefícios pessoais a fim de custear meus estudos por anos a fio. Papai, Mamãe, este diploma é para vocês. Vocês não tiveram oportunidade de prosseguir nos estudos, mas são doutores em amor, sabedoria e dedicação.

Aos meus irmãos, Sérgio e Martins Filho, bons companheiros de todas as horas. Definitivamente, fui agraciado com uma linda família.

Ao Prof. Sergio Vianna Fialho, meu orientador pela segunda vez, mais que um orientador, um amigo. Poucas pessoas que conheço trabalham com tanta habilidade e dedicação.

Aos membros da banca examinadora, professores João Batista e Luiz Affonso, da UFRN, professora Rossana Andrade, da UFC, e professor George Azevedo do CEFET-RN, pela paciência em ler este trabalho, oferecer sugestões e dar seus comentários. Isto sem dúvida, contribuiu para a minha formação.

Ao amigo, irmão e pai por adoção, Professor José Gilson de Oliveira, do CEFET-RN, um grande incentivador. Caro amigo, seus conselhos me impulsionaram na vida pessoal e acadêmica. Sem seus conselhos, possivelmente não teria enfrentado este desafio.

Aos irmãos, amigos e colegas do PoP-RN, especialmente Mário Sérgio, Hezione e D. Márcia, com quem compartilhei minhas aflições e expectativas, ao longo desses anos.

Edson Moreira Silva Neto

RESUMO

SILVA NETO, Edson Moreira. **Especificação de uma rede MPLS Fim-a-Fim com diferenciação de serviços.** 2006. 173 f. Tese (Doutorado em Ciências em Engenharia Elétrica), Universidade Federal do Rio Grande do Norte, Natal.

Palavras-chave: Redes de computadores, protocolos de comunicação, MPLS, QoS, RSVP, Estelle.

O protocolo proposto nesta Tese, denominado *Resource Reservation Protocol – Switched Virtual Connection* (RSVP-SVC), que consiste numa extensão do RSVP-TE, vai de encontro ao surgimento de novas aplicações multimídia, que usam a Internet como meio de interconexão. Tais aplicações pressionam pelo desenvolvimento de novas tecnologias, tais como: MPLS, DiffServ e RSVP-TE, que introduzem novas e eficientes características ao *backbone* Internet, proporcionando uma significativa diferença no que tange à provisão de QoS (*Quality of Service*). O presente trabalho leva em conta o fato de que para se conseguir uma QoS fim-a-fim verdadeira, não basta implementar tais tecnologias no núcleo da rede, é imprescindível, estender tais melhorias às redes de acesso e quiçá às redes locais. Nesse sentido, muitos trabalhos estão atualmente em desenvolvimento.

É no intuito de contribuir com este processo que este trabalho apresenta a definição de uma UNI MPLS SVC através do RSVP-SVC. Essa extensão dá ao RSVP-TE a capacidade de estabelecer túneis LSP (*Label Switched Path*) a partir de conexões discadas, ampliando portanto o escopo de utilização do MPLS, levando-o até às redes locais através das redes de acesso, e provendo também suporte a uma QoS fim-a-fim verdadeira.

O RSVP-SVC foi especificado em Estelle, que é uma linguagem de especificação formal padronizada pela ISO. A edição, compilação, verificação e simulação do RSVP-SVC foi feita através do programa EDT (*Estelle Development Toolset*).

Ademais, tanto os benefícios quanto às questões mais importantes a serem consideradas quando do uso deste protocolo são apresentados.

ABSTRACT

SILVA NETO, Edson Moreira. ***Specification of an End-to-End MPLS network with differentiated services*.** 2006. 173 p. Thesis (*Degree of Doctor in Eletrical Engineer*), Universidade Federal do Rio Grande do Norte, Natal.

Keywords: Computer networks, commmunication protocols, MPLS, QoS, RSVP, Estelle.

New multimedia applications that use the Internet as a communication media are pressing for the development of new technologies, such as: MPLS (Multiprotocol Label Switching) and DiffServ. These technologies introduce new and powerful features to the Internet backbone, as the provision of QoS (Quality of Service) capabilities. However, to obtain a true end-to-end QoS, it is not enough to implement such technologies in the network core, it becomes indispensable to extend such improvements to the access networks, what is the aim of the several works presently under development.

To contribute to this process, this Thesis presents the RSVP-SVC (Resource Reservation Protocol – Switched Virtual Connection) that consists in an extension of RSVP-TE. The RSVP-SVC is presented herein as a mean to support a true end-to-end QoS, through the extension of MPLS scope. Thus, it is specified a Switched Virtual Connection (SVC) service to be used in the context of a MPLS User-to-Network Interface (MPLS UNI), that is able to efficiently establish and activate Label Switched Paths (LSP), starting from the access routers that satisfy the QoS requirements demanded by the applications.

The RSVP-SVC was specified in Estelle, a Formal Description Technique (FDT) standardized by ISO. The edition, compilation, verification and simulation of RSVP-SVC were made by the EDT (Estelle Development Toolset) software.

The benefits and most important issues to be considered when using the proposed protocol are also included.

SUMÁRIO

Página

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS, SIGLAS E SÍMBOLOS

AR	*Access Router*
ARIS	*Aggregate Route-based IP Switching*
ATM	*Asynchronous Transfer Mode*
BA	*Behaviour Aggregate*
BGP-TE	*Border Gateway Protocol – Traffic Engineering*
CBQ	*Class Based Queuing*
CRC	*Cyclical Redundancy Check*
CR-LDP	*Constraint Route – Label Distribution Protocol*
CR-LSP	*Constraint Route – Label Switched Path*
CSR	*Cell Switching Router*
DiffServ	*Differentiated Services*
DLCI	*Data Link Control Identifier*
DOD	*Downstream On-Demand*
DOU	*Downstream Unsolicited*
DSCP	*DiffServ Code Point*
ECN	*Explicit Congestion Notification*
EDT	*Estelle Development Toolset*
E-LSP	*EXP – Label Switched Path*
ER	*Explicit Route*
FDT	*Formal Description Technique*
FIFO	*First-In First-Out*
FR	*Frame Relay*
GPS	*Generalized Processor Sharing*
IETF	*Internet Engineering Task Force*
IntServ	*Integrated Services*
IP	*Internet Protocol*
IS-IS	*Intermediate-System to Intermediate-System*
ISP	*Internet Service Provider*
LAN	*Local Area Network*
LDP	*Label Distribution Protocol*
LER	*Label Edge Router*
LIB	*Label Information Base*
L-LSP	*Label – Label Switched Path*
LSP	*Label Switched Path*

MF	*Multifield*
MPLS	*Multiprotocol Label Switching*
MPLSoLAN	*MPLS over LAN*
NS	*Network Simulator*
OSPF	*Open Shortest Path First*
OSPF-TE	*Open Shortest Path First – Traffic Engineering*
PHB	*Per Hop Behaviour*
PIM	*Protocol Independent Multicast*
PQ	*Priority Queuing*
PVC	*Permanent Virtual Connection*
QoS	*Quality of Service*
QoS-KN	*Quality of Service – Knowledge Network*
RED	*Random Early Detection*
RIP	*Routing Internet Protocol*
RSVP	*Resource Reservation Protocol*
SF	*Single Classifier*
SLA	*Service Level Agreement*
SVC	*Switched Virtual Connection*
TCP/IP	*Transmission Control Protocol / Internet Protocol*
TE	*Traffic Engineering*
ToS	*Type of Service*
UNI	*User-to-Network Interface*
VCI	*Virtual Channel Identifier*
VoIP	*Voice over IP*
VPI	*Virtual Path Identifier*
VPN	*Virtual Private Network*
WFQ	*Weighted Fair Queuing*
WRED	*Weighted Random Early Detection*
WRR	*Weighted Round Robin*

Capítulo 1
Introdução

Este capítulo apresenta uma introdução ao trabalho desenvolvido. Inicialmente, na Seção 1.1 são tecidos comentários gerais acerca das tendências para as tecnologias de redes Internet, no sentido de contextualizar o trabalho e ressaltar a importância do mesmo. Em seguida, na Seção 1.2 é apresentada uma justificativa para o desenvolvimento do tema assim como os objetivos pretendidos. A organização do trabalho é apresentada na Seção 1.3.

1.1. Tendências Atuais para as Tecnologias de Redes Internet

As exigências das redes atuais em termos de desempenho dificilmente devem ter sido imaginadas quando o protocolo IP (*Internet Protocol*) foi definido pela primeira vez. Atualmente, além de atender ao crescente volume de dados gerados pelos serviços tradicionais da Internet (transferência de arquivos e correio eletrônico, dentre outros), que utilizam o paradigma de encaminhamento de pacotes através do melhor esforço, necessita-se também de capacidade para diferenciar entre várias classes de tráfego, de forma a atender às exigências do tráfego multimídia: dados, voz, música e vídeo, sob demanda, ou mesmo em tempo real. Tudo isto tem levado a uma busca crescente por Qualidade de Serviço (QoS), que se tornou, destarte, um requisito essencial para o sucesso da tendência de convergência no uso do IP, seja no núcleo das redes de longa distância, seja nas redes de acesso, ou ainda nas redes locais corporativas.

Diante da realidade imposta pelas novas aplicações e seus requisitos, muitas alterações têm se revelado necessárias em relação ao modelo inicialmente proposto para a Internet. No que diz respeito ao tradicional roteamento IP *hop-by-hop,* por exemplo, percebe-se que ele está começando a chegar ao seu limite tecnológico, o que implica em uma necessária mudança de paradigma para o processo de encaminhamento de pacotes, uma vez que a fabricação de roteadores maiores, mais rápidos e mais baratos não basta [Ryan98].

Nesse contexto, a tecnologia de comutação por rótulos tem se firmado como uma solução eficiente, que não apenas atende ao aspecto da velocidade,

como também é capaz de distinguir as diferentes classes de tráfego, que exigem características de serviço específicas e que devem ser garantidas por todo o caminho ao longo da rede. O MPLS (*Multi-Protocol Label Switching*), em particular, como proposta padrão definida pelo IETF (*Internet Engineering Task Force)* no que concerne à comutação por rótulos, permite entre outras funcionalidades a criação de caminhos comutados por rótulo (LSP – *Label Switched Path*), com características de serviço diferentes. Essa característica provê melhorias significativas no processo de encaminhamento de pacotes, que passa a ser feito a partir do processamento do cabeçalho de nível 2, dispensando o processamento do cabeçalho do datagrama IP em cada nó do caminho, e criando também um ambiente que dá suporte à QoS em nível de enlace.

Nesse cenário, há, portanto, uma forte tendência para que as redes multi-serviços operem sobre as tecnologias IP/MPLS, devido justamente ao suporte à convergência de serviços e ao oferecimento de novas oportunidades nas áreas de Engenharia de Tráfego (TE – *Traffic Engineering*) e provimento de VPN (*Virtual Private Network*). Leve-se em conta, ainda, o fato de que toda esta melhoria deverá ser provida a um custo menor, o que deve possibilitar inclusive a criação de novas oportunidades de negócios para os provedores de serviços Internet (ISP – *Internet Service Provider*) [Integral02a].

De fato, as principais operadoras mundiais têm construído suas redes em função da necessidade de prover suporte a uma gama de novas aplicações, relacionadas com a rápida adoção, em larga escala, de redes convergentes. Nesse contexto, as grandes operadoras mundiais têm expandido consideravelmente o uso do MPLS graças a aplicações como VoIP (*Voice over IP*) e IPTV.

Em [Physics04] constatou-se que os principais fatores motivadores para adoção do MPLS são:

- Melhor QoS;
- Melhor suporte a serviços IP, como VoIP;
- Conectividade de rede em malha melhor e mais fácil;
- Habilidade de implementar topologias mais flexíveis;
- Uso de um único protocolo de rede: o IP, já conhecido pelos administradores de redes.

A Figura 1.1 apresenta um gráfico que mostra o percentual de respostas, quanto aos motivadores para adoção do MPLS.

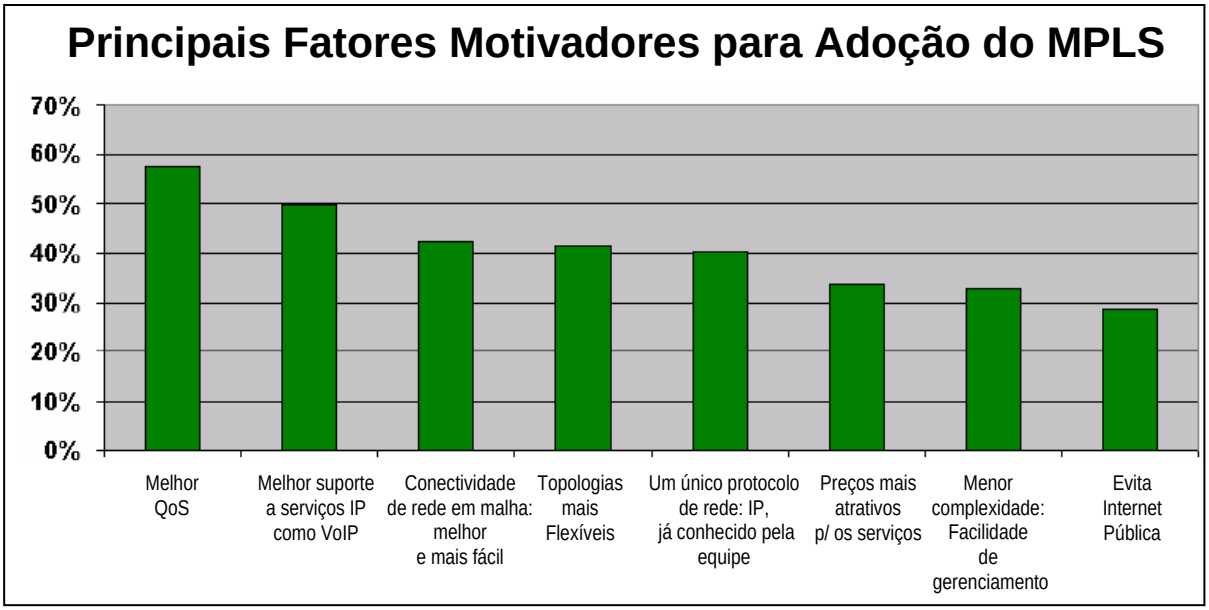

Fig. 1.1 – Fatores Motivadores para Adoção do MPLS

Outra pesquisa [Netscout06], mais recente, confirma esta tendência, onde 48% dos entrevistados acreditam que o emprego do MPLS causa impacto na infra-estrutura de rede. A Tabela 1.1 apresenta os dados referentes à tendência de investimentos em novos projetos para 2007, que causariam impactos em suas corporações.

PROJETO	PERCENTUAL
Implementação de Novas Políticas de QoS	64%
Novo emprego de VoIP	54%
Expansão do uso atual de VoIP	50%
Fornecimento Massivo de VPNs	50%
Emprego do MPLS	48%

Tab. 1.1 – Tendências de Investimentos para Novos Projetos em 2007

Dois aspectos importantes são percebidos nesta pesquisa:

1º) Há uma forte ligação entre QoS e uso de VoIP. Percebe-se que, para suportar VoIP, as empresas estão empregando Políticas de QoS no sentido de garantir a qualidade das chamadas em níveis adequados. Os percentuais obtidos são reveladores: 76% das empresas que estão expandindo o uso de VoIP ou que planejam novos empregos de VoIP, estão, também, expandindo suas políticas de QoS;

2º) O MPLS tem sido a ligação natural entre QoS e VoIP. Uma das maneiras que as empresas estão usando QoS é através do emprego de serviços MPLS, que dão a elas a habilidade de colocar o tráfego de voz, no que os ISPs se referem como sua classe de tráfego de tempo-real.

Esta pesquisa obteve os seguintes números:

- 58% das empresas que estão implementando novos empregos de VoIP também estão implementando o MPLS;
- 54% das empresas que estão expandindo o uso atual de VoIP também estão implementando MPLS;
- 76% das empresas que estão implementando políticas adicionais de QoS também estão implementando MPLS.

Também na implantação de redes metropolitanas, pode-se identificar uma forte tendência no uso do MPLS como solução para o transporte de pacotes, inclusive na arquitetura Metro Ethernet [Raahemi04].

1.2. Justificativa e Objetivos

Não resta dúvida de que o MPLS deve prover uma base sólida para as novas demandas no que diz respeito ao uso dos recursos no núcleo da rede. Contudo, existem indícios que um novo ponto de gargalo deve ser possivelmente encontrado nas redes de acesso [Integral02a].

Assim, tendo como pano de fundo as condições supracitadas, torna-se relevante estudar as questões envolvidas na definição de um ambiente de rede, onde o MPLS esteja presente não apenas no núcleo*,* onde já foi provada a sua eficácia, mas também nas redes de acesso e nas redes locais, ou seja, nos roteadores de acesso, nos *hosts* e *switches*, onde se encontram os usuários finais.

Isto posto, a proposta aqui apresentada objetiva suprir esta lacuna, estendendo gradativamente os benefícios do MPLS até o *desktop*, de forma que se possa estabelecer uma rede MPLS Total. A fim de cumprir com este propósito, essa Tese está sub-dividida em duas partes:

1ª.) Visa estender os benefícios do MPLS até os Roteadores de Acesso, através da especificação de uma UNI (*User-to-Network Interface*) MPLS SVC (*Switched Virtual Connection*), haja vista que já existe um acordo sobre uma UNI MPLS PVC (*Permanent Virtual Connection*). Essa nova especificação pretende fornecer a

capacidade necessária aos roteadores de acesso, para requisitar LSP com parâmetros de QoS específicos, no caso de estabelecimento de conexões chaveadas. Este tipo de conexão é fundamental para provimento de um conjunto de novos serviços, a exemplo de VoIP;

2ª.) Visa estender os benefícios do MPLS até o *desktop* do usuário final, através da especificação de um protocolo MPLSoLAN (MPLS *over LAN*) – que compreende um sub-conjunto das funcionalidades do MPLS original, com novas extensões – que possa ser implementado nos *hosts*, *switches* e roteadores, no domínio das redes locais. Esse protocolo deve disponibilizar um mecanismo de controle de admissão em redes RSVP-*enabled* (*Resource Reservation Protocol*), fornecendo ainda uma facilidade de mapeamento de QoS nível 3 para o nível 2, facilitando assim o gerenciamento e controle de QoS na LAN (*Local Area Network*). O MPLSoLAN deve permitir também, uma distribuição da carga de processamento nos LER (*Label Edge Router*) e nos AR (*Access Router*) relativo à classificação multi-campo de pacotes de ingresso, uma vez que esteja implementado e operacional o serviço SVC. A Figura 1.2 apresenta uma visão geral do escopo desta proposta.

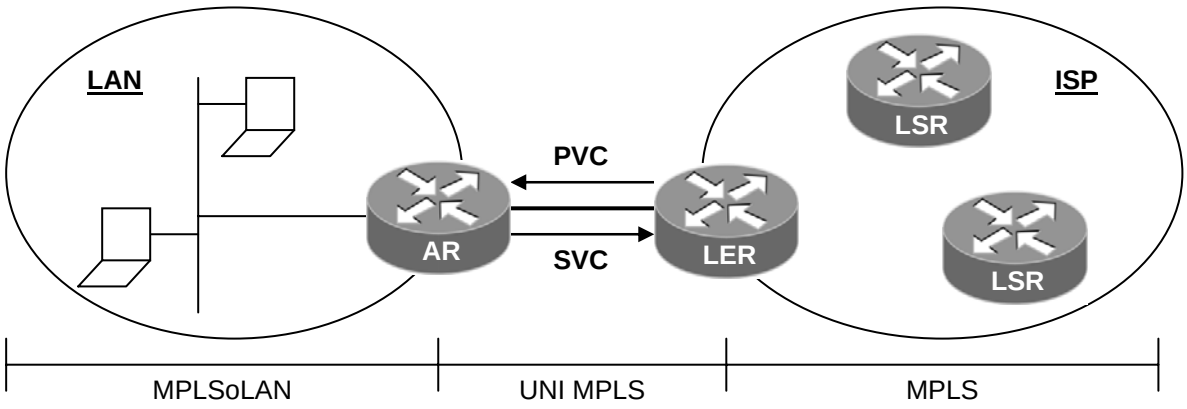

Fig. 1.2 – Escopo das Especificações do MPLS Fim-a-Fim

1.3. Organização do Trabalho

Este trabalho está organizado em 8 capítulos. O presente capítulo apresentou uma introdução, justificando a necessidade e delineando os objetivos dos estudos pretendidos. Os capítulos 2, 3, 4, e 5, apresentam o referencial teórico do trabalho: no Capítulo 2 é dada uma visão geral sobre Qualidade de Serviço, suas necessidades, definições e parâmetros, suas arquiteturas e mecanismos. A

apresentação dessa visão geral foi considerada relevante para esse trabalho, uma vez que o provimento de QoS se constitui num aspecto intrínseco à proposta apresentada e várias referências são feitas aos conceitos envolvidos dessa área.

Já o Capítulo 3 dedica-se a uma revisão da arquitetura MPLS. É apresentada uma visão geral do funcionamento do MPLS e são abordados os principais elementos da arquitetura e os conceitos gerais, tais como: rótulo, FEC, tabelas de encaminhamento, políticas de distribuição de rótulos, entre outros.

O Capítulo 4, por sua vez, apresenta os principais conceitos envolvidos no contexto das Redes Multi-serviço e da Engenharia de Tráfego, procurando colocá-los numa ordem lógica e sistemática. Como o MPLS é uma resposta importante às necessidades de TE (*Traffic Engineering*), duas seções foram dedicadas a fazer a ligação entre os conceitos supra-citados e o MPLS. É mostrado também um comparativo entre o CR-LDP (*Label Distribution Protocol*) e o RSVP-TE, como alternativas de Protocolo de Sinalização para o MPLS, além de uma análise detalhada do funcionamento do RSVP-TE.

A linguagem Estelle e seus conceitos gerais; assim como os principais aspectos relacionados à verificação formal e às boas propriedades de uma especificação, além de uma descrição sucinta da ferramenta EDT (*Estelle Development Toolset*) são apresentadas no Capítulo 5.

Os Capítulos 6 e 7 contêm as principais contribuições deste trabalho e são dedicados à apresentação dos serviços e do protocolo propostos. O Capítulo 6 versa sobre as definições conceituais de uma rede MPLS Fim-a-Fim, que abrange o serviço discado MPLS e o MPLSoLAN. Nele são descritas as premissas básicas, o *modus-operandi*, e as funcionalidades de cada serviço e protocolo relacionado.

Detalhes da especificação do RSVP-SVC, sua verificação formal e os resultados de simulação são apresentados no Capítulo 7.

Finalmente, as conclusões e benefícios de uso do serviço proposto neste trabalho, assim como possibilidades de trabalhos futuros são apresentadas no Capítulo 8.

Capítulo 2
Qualidade de Serviço – Uma Visão Geral

Este capítulo apresenta uma visão geral sobre os principais aspectos envolvidos no provimento de Qualidade de Serviço (QoS) para aplicações que usam a Internet no transporte de suas informações. Para isso, descreve-se na Seção 2.2 os parâmetros de QoS considerados. Já os principais mecanismos usados para controlar estes parâmetros e as arquiteturas atualmente disponíveis para prover QoS em redes IP são apresentados nas Seções 2.3 e 2.4, respectivamente. A Seção 2.5 tece comentários gerais relativos ao Roteamento e QoS.

2.1. Introdução

A arquitetura Internet – também conhecida como arquitetura TCP/IP (*Transmission Control Protocol/Internet Protocol)* – foi concebida, inicialmente, tendo como princípios básicos de projeto a funcionalidade e a simplicidade. Desta forma, ela foi estruturada em camadas e o protocolo Internet (IP) padrão, projetado para prover um serviço de entrega de dados baseado no paradigma do "*melhor-esforço*", ou seja, o tráfego é processado tão rapidamente quanto possível, sem, contudo, prover quaisquer garantias de entrega ou compromissos com o tempo de entrega.

Esta decisão foi importante porque permitiu que a complexidade ficasse nos sistemas finais enquanto a rede continuava relativamente simples, o que é coerente com um princípio de projeto conhecido como argumento fim-a-fim [Saltzer01]. A forma como este modelo foi implementado, além de atender adequadamente o perfil das aplicações à época, proporcionou uma série de outras vantagens [QoSForum99a]:

- Utiliza um conjunto de protocolos aberto e público;
- É independente de equipamento e sistema operacional;
- Implementa um esquema de endereçamento global;
- Provê suporte a protocolos de aplicação que atendem à demanda de serviços distribuídos.

Tais características favoreceram a disseminação deste modelo, tornando-o um padrão *de facto* no mercado. Entretanto, com a evolução das redes de

computadores, as aplicações que eram executadas originalmente na rede, tais como: correio eletrônico e transferência de arquivos, começaram a dividir espaço com telefonia e fax sobre IP, vídeo-conferência, educação à distância, aplicações colaborativas e de grupo, aplicações multimídia e de tempo real, entre outras. Este segundo grupo de aplicações têm exigências mais específicas relacionadas com questões de segurança e de desempenho (Qualidade de Serviço). Estes últimos requisitos se tornaram, portanto, fatores chaves para o sucesso das novas redes, tanto no que diz respeito à ampliação do espectro das aplicações atendidas, quanto às possibilidades de oferta de novos negócios por parte dos provedores de serviços.

Essas novas demandas têm impulsionado o desenvolvimento de novos mecanismos e modelos de serviço, visando atender às exigências de QoS. É nesse cenário que surgem as primeiras propostas para o estabelecimento de padrões relacionados à diferenciação de serviços e à garantia de desempenho para aplicações Internet, haja vista que a rede original não era capaz de dar nenhuma garantia de vazão, nem de entrega de pacotes (descartados ou perdidos), nem ainda de tempo de entrega dos pacotes que nela trafegam.

Estas demandas por QoS podem ser classificadas em dois grandes grupos, como apresentado em [Zhao01]: as motivadas por fatores humanos e as motivadas por fatores comerciais.

- *Demandas Motivadas por Fatores Humanos*: são aquelas devidas a atrasos ou perdas de informação, como por exemplo, nas aplicações de voz interativa, onde um atraso excessivo pode trazer desconforto ou mesmo inviabilizar a aplicação. Portanto, é imprescindível impor limites no atraso de transmissão e/ou na taxa de perda para certas aplicações.
- *Demandas Motivadas por Fatores Comerciais*: muitas aplicações da área comercial e financeira precisam ser completadas dentro de limites estritos de tempo, sob pena de serem abortadas antes da conclusão da transação, o que impõe limitações em certos parâmetros de transmissão.

2.2. Definições e Parâmetros

A Qualidade de Serviço, no contexto das redes de pacotes IP, pode ser definida como um requisito das aplicações para as quais se exige que determinados

parâmetros como atrasos, vazão, perdas, entre outros, estejam dentro de limites bem definidos (valor máximo, valor mínimo e outros) [Martins01].

Vale ressaltar que, entre um cliente e um provedor de serviços de rede, tais valores devem ser claramente explicitados através de uma especificação de nível de serviços em um contrato de prestação de serviços, normalmente designado de SLA (*Service Level Agreement*).

Em geral, tanto o provedor quanto o cliente poderão desejar monitorar o uso e o desempenho dos serviços contratados.

Nesse sentido, os principais parâmetros de QoS que compõem um SLA, conforme encontrado em [QoSForum99e], são explicitados a seguir:

- **Latência ou Atraso**. A rede e os equipamentos usados na comunicação possuem limitações intrínsecas (tempo de processamento no roteador, espera em filas de entrada e saída, atraso de propagação) que proporcionarão, *de per si*, determinado nível de atraso. A latência pode ser entendida, enfim, como o somatório destes atrasos na transmissão dos dados desde a origem até o destino.
- ***Jitter***. Diz respeito à variação no atraso, ou seja, a diferença dos tempos de chegada entre pacotes quando comparados com os intervalos da transmissão original. Algumas aplicações multimídia são particularmente sensíveis a esse parâmetro, a exemplo de transmissões de voz, áudio e vídeo.
- **Banda**. É a capacidade de transmissão de pacotes através da rede por unidade de tempo.
- **Perda de Pacotes**. As perdas de pacote em redes IP se dão, principalmente, em função dos descartes de pacotes nas filas dos roteadores, devido a congestionamento nos enlaces de saída. Podem também ser devidas a erros detectados na camada dois, a exemplo de erros de CRC (*Ciclical Redundancy Check*).
- **Disponibilidade**. É uma medida da execução da aplicação ou serviço ao longo do tempo. Normalmente, é referenciada em termos percentuais.

2.3. Mecanismos

Os diversos mecanismos de QoS devem atuar de forma cooperativa nos equipamentos e camadas de protocolo com o objetivo de controlar o valor dos

parâmetros mencionados de acordo com o desejado. Em geral, nem todos esses mecanismos se encontram implementados nos equipamentos de rede; dependendo do equipamento, sub-conjuntos diferentes estarão disponíveis. Para entender sua função deve-se levar em consideração, entre outras, as seguintes questões: como alocar recursos escassos? Como selecionar o tráfego de pacotes? Como priorizar pacotes? Como descartar pacotes? A seguir são citados os principais mecanismos envolvidos nessa área.

2.3.1. Protocolo de Sinalização

É utilizado pelas aplicações (*hosts*) para informar ou solicitar à rede sua necessidade de QoS. Também permite que os equipamentos de rede possam trocar informações no sentido de cooperarem entre si, visando garantir a QoS aceita pela rede. Podem-se citar como exemplos de protocolos de sinalização: o RSVP (*Resource Reservation Protocol*) e o LDP (*Label Distribution Protocol*).

Associados aos protocolos de sinalização encontram-se, em geral, os processos de controle de admissão e controle de policiamento.

O controle de admissão verifica se o equipamento da rede tem condições de suportar um novo fluxo que demanda serviço, de acordo com as exigências de QoS especificadas no pedido de serviço. Caso os recursos sejam insuficientes, o pedido deve ser rejeitado e a transmissão do fluxo bloqueada.

O controle de policiamento verifica se o usuário tem permissão administrativa para fazer um pedido de reserva de recursos da rede.

2.3.2. Classificação

A classificação de pacotes é um processo que permite a um roteador distinguir entre pacotes de diferentes classes de serviço. Ela acontece em todos os roteadores e, durante o encaminhamento de pacotes de dados, precede todos os outros mecanismos de QoS.

No contexto de uma das arquiteturas de QoS disponíveis atualmente, a arquitetura DiffServ (apresentada na Seção 2.4.3), existem dois tipos de classificadores: um que classifica o fluxo baseado apenas em um único campo (no DiffServ utiliza-se o DS *codepoint* – DSCP) e outro que verifica múltiplos campos no cabeçalho IP. Eles são conhecidos como classificador de comportamento agregado (BA – *Behaviour Agregate*) e classificador multi-campo (MF – *Multifield Classifier*),

respectivamente. Entretanto, como as soluções para provimento de QoS não são excludentes e sim complementares, a tipificação sugerida pelo DiffServ pode ser estendida para outros contextos, uma vez realizadas as adequações necessárias. Assim, é possível sugerir que na arquitetura IntServ (apresentada na Seção 2.4.2), a classificação BA pudesse ser denominada classificação de um único campo (SF – *Single Classifier*), embora a norma não faça referência a esse termo. Isso seria útil para o IPv6, por exemplo, que já contém em um dos campos do cabeçalho de seus datagramas um identificador de fluxo.

- Classificação MF: é o processo de classificar pacotes, baseado no conteúdo dos campos: endereço de origem, endereço de destino, campo ToS (*Type of Service*), identificador do protocolo, portas de origem e de destino, e ocorre geralmente no roteador de entrada de um domínio. Assim, ao receber um pacote, o roteador executa uma classificação MF e determina a classe de serviço que o pacote pode ser associado. Após ser classificado, o pacote pode ser marcado, medido ou simplesmente escalonado para encaminhamento. Vale destacar que todos os pacotes pertencentes a uma mesma classe obtêm o mesmo tratamento do escalonador. É importante ainda lembrar que uma classe é uma abstração que pode ser local a um dado roteador, possibilitando assim que um mesmo pacote seja classificado de forma diferente pelos roteadores ao longo do caminho. Um exemplo clássico é o caso dos roteadores de *backbone* que podem escolher o mapeamento de muitos fluxos em poucas classes agregadas, enquanto que os roteadores periféricos, onde existe menos agregação, podem usar uma classe separada para cada fluxo. Nesse último caso, uma classificação MF se faz necessária.
- Classificação BA: é uma classificação baseada apenas no DSCP e é, em geral, implementada nos roteadores intermediários.

As técnicas seguintes de marcação, medição e moldagem são referidas em seu conjunto como técnicas de condicionamento de tráfego.

2.3.3. Marcação

A marcação dos pacotes é feita, em geral, pela introdução de uma etiqueta ou identificador em um dos campos do cabeçalho do pacote. O valor do identificador introduzido depende da classificação prévia do pacote em um fluxo

específico ou em uma classe de fluxos. Ela pode ocorrer em três pontos distintos, a saber:

- Na origem, os pacotes são marcados diretamente pelas aplicações;
- Nos roteadores de borda, tipicamente após uma Classificação MF; e,
- Nos roteadores do interior de um domínio, em geral, relacionada à agregação de tráfego e/ou tunelamento.

A marcação de pacotes facilita sua classificação no interior da rede, permitindo que se faça uma classificação de único campo.

2.3.4. Medição/Policiamento de Tráfego

Refere-se à fiscalização da taxa de entrada de pacotes em um nó da rede. A idéia é verificar se o tráfego está em conformidade com o especificado no SLA ou não. O algoritmo usado para este fim é o Balde de Fichas (*Token Bucket*), que é definido na RFC 2697 [Heinanen99]. Esse algoritmo oferece um meio de verificar as seguintes características dos fluxos de entrada: tamanho máximo de rajada, taxa de pico e taxa média especificados.

O tráfego que estiver dentro do perfil contratado (*in-profile*) deve ser encaminhado e transmitido de acordo com a QoS contratada. Entretanto, os pacotes do tráfego que exceder os níveis de serviço previstos no SLA (*out-of-profile*) podem ser atrasados (moldagem), descartados, remarcados (para descarte em casos de congestionamento da rede) ou sobretaxados.

Assim, para medir as características do tráfego de entrada, de acordo com o algoritmo do balde de fichas, procede-se da seguinte forma: durante um intervalo fixo de tempo ($\triangle t$) são armazenadas na fila de entrada as rajadas de pacotes entrantes. Se a quantidade de bytes desse tráfego de entrada não atingir o valor aceito contratado (CBS – *Comitted Burst Size*), esses pacotes são transmitidos integralmente, pois considera-se que a taxa média de chegada é inferior ao valor aceito (CIR – *Comitted Information Rate*), já que $CIR=CBS/\triangle t$. Caso a quantidade de bytes do tráfego de entrada ultrapasse o valor do CBS, mas ainda fique abaixo de um valor máximo contratado (EBS – *Excess Burst Size*), os pacotes que ultrapassarem o valor CBS são marcados para um eventual descarte posterior mas, caso a rede não esteja congestionada, ainda podem ser transmitidos integralmente. Finalmente, caso a quantidade de bytes do tráfego de entrada ultrapasse o valor de

EBS, os pacotes em excesso deverão ser descartados. Os pacotes transmitidos são, em geral, enviados a outros processos internos, a exemplo do marcador e/ou módulo de moldagem/descarte dos pacotes.

2.3.5. Moldagem de Tráfego

O objetivo deste mecanismo é adequar o tráfego real ao perfil contratado, quando este excede aquele. O algoritmo comumente usado para esta tarefa é o balde furado (*Leaky-Bucket*) [Akhtar87].

De acordo com o algoritmo *Leaky-Bucket*, o fluxo de entrada no elemento responsável pela moldagem vai sendo armazenado numa fila interna, de onde uma quantidade fixa de bytes é transmitida a cada pulso de relógio, gerando um tráfego de saída a uma taxa constante. Essa fila tem um tamanho máximo que indica a capacidade máxima de armazenamento dos dados entrantes. O tamanho máximo da fila (tamanho do balde furado) e a taxa de transmissão são configuráveis e refletem as características do perfil aceito para o tráfego de entrada.

Todos os pacotes entrantes, que puderem ser armazenados na fila, são aceitos e transmitidos à taxa configurada. Caso um pacote entrante, ou parte dele, causar *overflow* da fila, ele será descartado.

Como a chegada de um fluxo, possivelmente irregular de pacotes (tráfego em rajada), é transformado em um fluxo regular de pacotes de saída par a rede, essa técnica de moldagem de tráfego é capaz de suavizar rajadas, reduzindo significativamente as chances de ocorrência de um congestionamento na rede.

2.3.6. Provisionamento de Recursos

O provisionamento de recursos nos equipamento de rede para garantir a QoS contratada pelos vários clientes, pode incluir mecanismos de uso eficiente dos enlaces de saída, de gerenciamento dos recursos do equipamento e de controle dos parâmetros de QoS.

Como técnica básica, o provisionamento de recursos utiliza um mecanismo de enfileiramento de pacotes, que diz respeito à ação de armazenar pacotes em um local onde eles permaneçam até serem processados. O enfileiramento ocorre, em geral, dentro dos roteadores e em dois momentos distintos:

- Na chegada dos pacotes: quando os pacotes são recebidos pelo processador de interface do dispositivo (fila de entrada);
- Na saída dos pacotes: quando os pacotes são enviados para uma outra interface (fila de saída) do mesmo dispositivo, antes de sua transmissão [Ferguson99].

Em geral, os dispositivos implementam uma única fila de entrada por interface, onde os pacotes são armazenados na ordem em que chegam, e possivelmente várias filas de saída por interface, de forma a poder dispensar tratamentos distintos a diferentes fluxos ou diferentes agregados de fluxos. No DiffServ, por exemplo, implementam-se tantas filas de saída por interface, quantas são as classes de serviço definidas. Pacotes de uma mesma classe são armazenados na fila de saída correspondente, em geral, obedecendo à sua ordem de chegada.

O atendimento diferenciado às várias classes de serviço passa, então, por um processo de escalonamento ou de serviço das várias filas de saída. Esse é um processo de decisão relacionado com a escolha de qual fila será servida no momento da transmissão. Esta escolha dependerá de alguma propriedade da fila e/ou do pacote e pode ser feita de acordo com várias regras ou disciplinas de escalonamento.

Tais regras procuram garantir que os diferentes fluxos de pacotes obterão os recursos que lhe foram alocados em termos de prioridade de processamento e banda. O processo de serviços dessas filas depende de configuração, uma vez que cabe ao administrador determinar que algoritmo, dentre os disponibilizados pelo equipamento, será utilizado.

O mecanismo de enfileiramento básico é o FIFO (*First-In-First-Out*), onde o primeiro pacote armazenado na fila é o primeiro a ser transmitido. Esse é o mecanismo utilizado no encaminhamento de pacotes segundo o serviço de melhor-esforço. Porém, em contextos onde se exige QoS, é necessário a adoção de novos mecanismos que adicionem prioridade. Nesses contextos, prioridade pode ser entendida como um mecanismo que provê diferentes tempos de espera ou tratamento de atraso para o processamento da informação, tanto no que diz respeito ao processamento do pacote quanto na sua transmissão nos enlaces de saída.

São exemplos de algoritmos de escalonamento com tratamento de prioridade: PQ (*Priority Queueing*), WRR (*Weighted Round Robin Queueing*), GPS (*Generalized Processor Sharing*), CBQ (*Class Based Queuing*) e WFQ (*Weighted Fair Queuing*).

2.3.7. Controle de Congestionamento

Quanto a esse tópico há de se considerar três atividades: prevenção, controle e recuperação de congestionamento. A idéia básica é a inibição dos fluxos de pacotes durante o período de congestionamento, de forma que os geradores de fluxos de pacotes IP reduzam a sua carga transmitida sobre a rede. Podem ser entendidos como mecanismos que têm a habilidade de controlar fluxo e retirar o excesso de tráfego em períodos de congestionamento.

Pode-se citar como exemplos de algoritmos de controle de congestionamento: RED (*Random Early Detection*), WRED (*Weighted Random Early Detection*) e ECN (*Explicit Congestion Notification*).

2.4. Arquiteturas de QoS

A implementação da garantia de QoS pela rede implica em atuar nos equipamentos envolvidos na comunicação fim-a-fim, visando o controle dos parâmetros de tráfego. Nesse sentido, várias arquiteturas têm sido propostas. Três das principais soluções encontradas na atualidade são: Super-Dimensionamento de Recursos; Arquitetura de Serviços Integrados ou IntServ, que é um modelo baseado em reserva de recursos da rede; e a Arquitetura de Serviços Diferenciados ou DiffServ, que é um modelo baseado em prioridades no encaminhamento de pacotes [QoSForum99c]. A seguir são apresentadas, de forma resumida, as principais características destes modelos.

2.4.1. Super-Dimensionamento de Recursos

A idéia por trás dessa solução é bastante simples, contudo, nem sempre viável na prática.

Segundo esse enfoque, a rede e seus recursos são dimensionados, antecipadamente, na fase de projeto, a não sofrer congestionamento nos momentos de pico, através do super-dimensionamento da banda e outros recursos de *hardware*, tal como memória nos roteadores da rede. Entretanto, não é difícil perceber que o custo dessa solução é alto, sobretudo nos *backbones*, e, além disso,

não é fácil identificar *a priori* os pontos potenciais de congestionamento, dada a multiplicidade e diversidade de equipamentos utilizados e a própria complexidade das redes.

Um outro aspecto contrário a esta solução é que, nem sempre o serviço de melhor-esforço é suficiente para todas as necessidades de todas as aplicações em todas as condições. Algumas vezes, faz-se necessário quantificar um determinado nível de confiabilidade dos serviços de rede [QoSForum99b].

2.4.2. Arquitetura IntServ ou Serviços Integrados

O IntServ foi definido na RFC 1633 [Braden94], sendo uma arquitetura baseada em fluxos com reserva dinâmica de recursos. Sua filosofia fundamental é que os roteadores da rede precisam reservar recursos para prover QoS quantificável para determinados fluxos de tráfego. Para isso, essa arquitetura propõe o uso do RSVP (*Resource Reservation Protocol*), como protocolo de sinalização para reservar os recursos da rede [Zhao01]. Propõe-se o uso de três classes de serviços [Braden94]:

- **Serviço Garantido**, para aplicações que exigem limites de atraso. Essa classe de serviço garante banda passante e atraso máximo para o fluxo de dados;
- **Serviço de Carga Controlada**, para aplicações que requerem um serviço de melhor-esforço melhorado e confiável. Essa classe prover uma QoS equivalente àquela recebida por um fluxo de dados numa rede sem congestionamento;
- **Serviço de Melhor-Esforço**, para aplicações que não necessitam de garantia de QoS.

É interessante notar que, ao prever a mobilização de recursos em todos os equipamentos de rede envolvidos na transmissão dos dados, este modelo representa uma mudança fundamental no paradigma original da Internet, que estava fundamentado no conceito de que toda a informação de estado relacionada aos fluxos deve estar localizada nos sistemas finais [Xiao99].

O modelo IntServ é implementado através de quatro componentes, conforme apresentado na Figura 2.1.

- **Protocolo de Sinalização** → Aplicações que requerem serviço garantido ou de carga controlada devem estabelecer os caminhos entre o(s) emissor(es) e o

receptor e reservar os recursos necessários antes da transmissão de seus dados. Para isso, usa um protocolo de controle, tipicamente o RSVP;

- **Controle de Admissão** → Processo que decide quando uma requisição por recursos pode ser aceita ou não, levando em conta os recursos disponíveis na rede;
- **Classificador** → É um processo que classifica os pacotes entrantes baseado no conteúdo de seus cabeçalhos, de acordo com regras pré-estabelecidas. Quando um roteador recebe um pacote, o classificador executará algum método de classificação e, baseado no resultado obtido, colocará o pacote em uma fila específica que atende a um determinado fluxo que tenha tido sucesso na reserva de recursos da rede;
- **Escalonador de Pacotes** → É um processo que decide que pacote será atendido primeiro em um sistema de filas múltiplas. Ele procura escalonar os pacotes de forma a atender suas exigências de QoS.

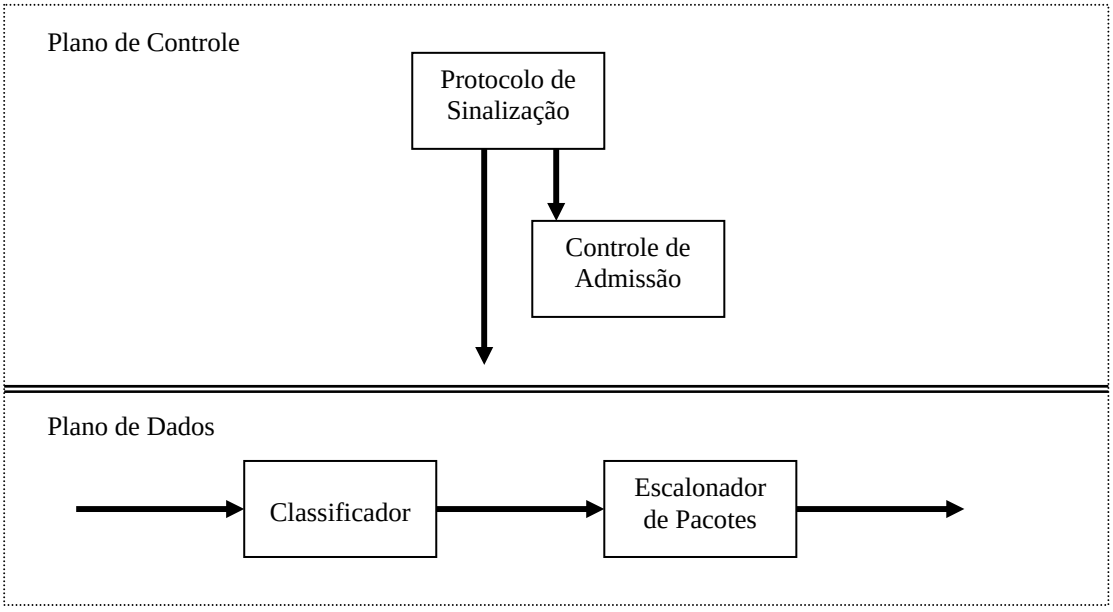

Fig. 2.1 – Componentes da Arquitetura de Serviços Integrados

Assim, uma aplicação que deseje enviar dados com garantias de QoS deve inicialmente utilizar o protocolo de sinalização, a fim de providenciar a reserva dos recursos necessários em cada roteador ao longo do caminho do fluxo, onde é executado um processo de controle de admissão, a fim de aceitar a reserva ou rejeitá-la caso não disponha de recursos suficientes.

A arquitetura IntServ apresenta alguns problemas, notadamente relacionados à questão de escalabilidade, haja vista que a quantidade de informações de estado cresce proporcionalmente ao número de fluxos atendidos. Ademais, ela exige muito da capacidade de processamento dos roteadores, pois todos eles devem implementar o RSVP, controle de admissão, e processos de classificação e escalonamento de pacotes.

2.4.3. Arquitetura DiffServ ou Serviços Diferenciados

A arquitetura DiffServ é, essencialmente, um esquema de prioridades relativas, que foi proposto pelo IETF através da RFC 2475 [Blake98], a fim de solucionar alguns dos problemas associados com a implementação do IntServ e RSVP, principalmente, os relacionados à escalabilidade e à flexibilidade [Zhao01].

A arquitetura DiffServ define:

- Nova semântica para o campo ToS (*Type of Service*) do IPv4 [Nichols98]. O campo ToS é agora chamado de campo DSCP (*DiffServ Code Point*);
- Um comportamento por nó (PHB – *Per Hop Behavior*), que vem a ser um tratamento particular dado aos pacotes nos roteadores durante o encaminhamento.

Para a correta operação deste modelo é necessário o estabelecimento de um SLA entre o cliente e o ISP, o qual pode ser estático ou negociado em bases regulares, tais como: mês/ano; ou ainda dinâmico, onde os valores dos parâmetros de QoS variam de acordo com disponibilidade dos recursos de rede. Neste último caso deve-se usar um protocolo de sinalização como RSVP para requisitar serviços sob demanda.

Chama-se domínio DiffServ a um conjunto de roteadores interligados em rede que estão sujeitos a uma mesma autoridade administrativa, que define regras comuns para os serviços diferenciados providos aos clientes.

Na arquitetura DiffServ, conforme indicado em [Nortel02], há três fases distintas no fluxo de cada pacote através de um determinado domínio: a fase de entrada, de encaminhamento e de saída, cujo funcionamento pode ser esquematizado conforme apresentado na Figura 2.2.

Na entrada da rede do provedor, os pacotes são classificados, marcados, policiados e, possivelmente, moldados, conforme regras derivadas dos SLA.

Vale destacar que os clientes podem marcar o campo DSCP de pacotes individuais para indicar o serviço desejado ou deixar que eles sejam marcados pelos roteadores de borda, baseados numa classificação multi-campo (MF – *Multifield*). De qualquer forma é responsabilidade do roteador de borda de um domínio garantir a marcação dos pacotes entrantes, de acordo com o SLA. Assim, o campo DS pode ser remarcado quando o pacote entra em um novo domínio.

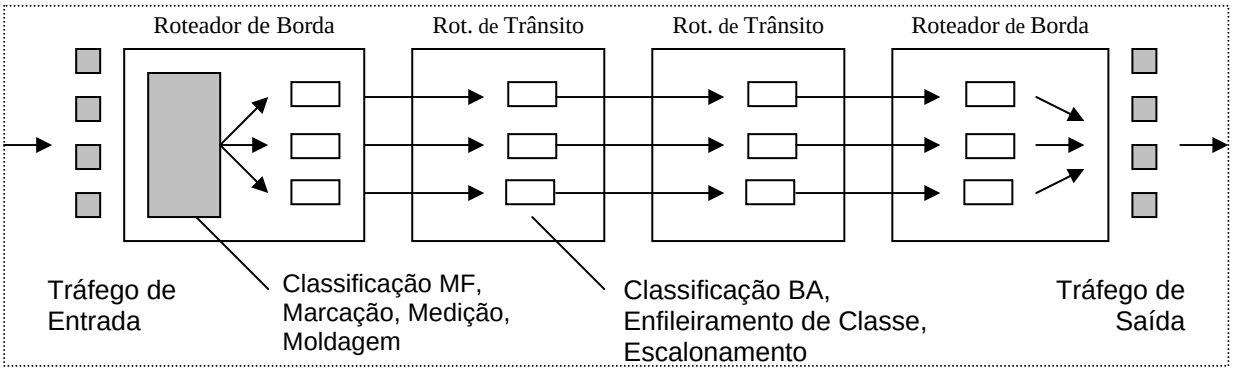

Fig. 2.2 – Funcionamento do DiffServ

Cada domínio DiffServ define quais serviços DiffServ são implementados dentro do domínio e qual a porcentagem relativa de recursos alocada a cada serviço. Comumente, são definidos os seguintes tipos de serviços:

- **Serviço Prêmio ou Serviço de Encaminhamento Urgente**, para aplicações que requerem serviço com baixo atraso, baixo *jitter* e baixa taxa de perdas.
- **Serviço de Encaminhamento Garantido**, equivalente ao serviço de carga controlada do IntServ, fornece valores de atraso próximos ao do serviço melhor-esforço em situações em que a rede não está congestionada. Esse serviço destina-se a aplicações que requerem, entretanto, maior confiabilidade que o serviço de melhor-esforço. Ele comporta quatro classes de prioridades diferentes, cada classe com três níveis de descarte de pacotes. O descarte de pacotes deve acontecer em casos de congestionamento da rede, de acordo com a ordem de precedência indicada. Um exemplo de implementação desse tipo de serviço recebe comumente o nome de Serviço Olímpico (*Olimpic Service*), onde se implementa somente três das quatro classes possíveis, que recebem o nome de classes ouro, prata e bronze.

- **Serviço de Encaminhamento por Seletor de Classes**: definido para manter compatibilidade com a especificação original dos níveis de precedência previstos para o antigo campo ToS (*Type of Service*) no cabeçalho dos datagramas IPv4. Garante, de certa forma, a migração de aplicações legadas que utilizavam a definição original na priorização de pacotes.
- **Serviço de melhor-esforço**, para aplicações que não exigem garantias de QoS.

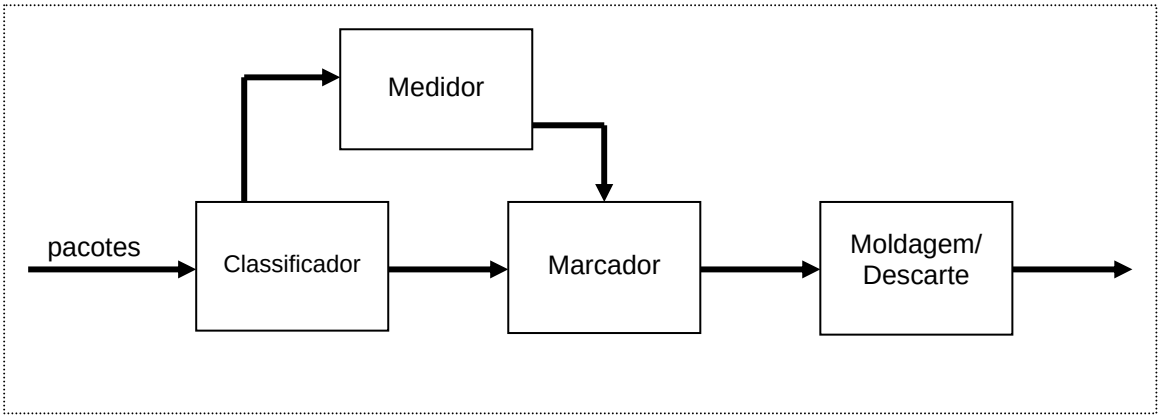

Fig. 2.3 – Blocos Funcionais Básicos da Arquitetura de Serviços Diferenciados

A Figura 2.3 apresenta o diagrama de blocos funcionais do DiffServ:

- **Classificador**: associa os pacotes a classes de tráfego. Há dois tipos de classificadores: Agregado de Comportamento (BA – *Behaviour Aggregate*) que se baseia somente no valor do DSCP; e o Classificador Multicampo que, além do campo DSCP, usa outras informações como: endereço de origem e destino, código de protocolo, número das portas de origem e destino.
- **Medidor/Monitor/Policiador**: mede o tráfego real comparando-o com o perfil contratado, podendo inclusive acumular estatísticas.
- **Marcador**: escolhe o *codepoint* apropriado para o campo DSCP, em função da classificação feita.
- **Moldagem/Descarte**: funciona como um mecanismo complementar de policiamento, ora retardando o pacote para colocá-lo dentro do perfil contratado (presente no SLA); ora, descartando o pacote fora do perfil.

O DiffServ apresenta algumas vantagens em relação ao IntServ, tais como:

- Escalabilidade: o DiffServ é mais escalável, haja vista que ele não trata de fluxos individuais e sim de um número limitado de classes de serviço indicadas pelo

campo DSCP, reduzindo, assim, a quantidade de informações de estado que precisam ser armazenadas. Essa quantidade é agora proporcional ao número de classes e não ao número de fluxos;

- Facilidade de Uso: ele é mais fácil de implementar e usar, pois as complexas operações de classificação multicampo, marcação, policiamento e moldagem são necessárias apenas nas bordas das redes [Xiao99];
- Flexibilidade: as classes de serviço não são derivadas diretamente da arquitetura, o que permite ao administrador do domínio especificar que classes serão consideradas e qual o tratamento dado a cada uma delas.

Contudo, vale destacar, finalmente, que as arquiteturas apresentadas não são excludentes, mas complementares entre si.

2.5. Roteamento e QoS

As atividades de roteamento referem-se à construção das tabelas de encaminhamento (de nível 3 no caso do IP), que são utilizadas na comutação dos pacotes através da rede até seu destino. As atividades de QoS, conforme foi visto anteriormente, procura garantir que nesse processo de encaminhamento de pacotes, determinados parâmetros como banda, atraso, perda e outros, permaneçam dentro de um perfil previamente acordado entre cliente e provedor de serviços de rede. Assim, embora tratem de aspectos distintos, as questões de roteamento e de QoS não devem ser vistas como questões separadas. Conforme descrito em [Zhao01], há evidências que a união de conceitos de roteamento com QoS pode resultar em desempenho superior da rede. É nesse contexto que tem surgido diversos esforços como: Engenharia de Tráfego e Roteamento Baseado em Restrições. Ademais, conforme apresentado no próximo capítulo, o MPLS desponta como um mecanismo poderoso de auxílio aos operadores das redes, no que concerne à TE e, conseqüentemente, nas questões relacionadas com QoS.

Todos os esquemas de QoS que foram apresentados tentam prover serviços diferenciados sob condições de sobrecarga da rede, objetivando sempre a minimização dos atrasos de entrega e das variações de atraso, e o provimento de capacidade constante ou mínima de vazão de dados. Em outras palavras, tenta-se prover algum nível de preditibilidade e controle sobre o serviço de melhor-esforço do IP [QoSForum99b].

De fato, alguns desses serviços diferenciados (os de carga controlada), diferem pouco do serviço de melhor-esforço se a carga na rede é suave.

Quanto à ocorrência de sobrecarga ou congestionamento na rede, ela se deve a, pelo menos, duas razões:

- A demanda de uso excede os recursos de rede disponíveis;
- A ocorrência de uma distribuição irregular do tráfego.

No primeiro caso, pode-se aumentar a capacidade da rede ou limitar o seu uso através de mecanismos de QoS. Entretanto, na segunda situação, uma redistribuição ou balanceamento de carga pode ajudar. É exatamente nessas situações em que a Engenharia de Tráfego pode ser de extrema utilidade, já que ela organiza os fluxos, de modo que congestionamentos causados pela utilização irregular da rede possam ser evitados.

No Capítulo 4 deste trabalho esses assuntos são abordados de forma mais detalhada.

Capítulo 3
Arquitetura MPLS

Este capítulo apresenta a arquitetura MPLS padrão, uma vez que ela é a base para a proposta contida nesse documento. Dessa forma, procura-se descrever todos os conceitos básicos relacionados à comutação por rótulo, os elementos constituintes de uma rede MPLS e também todos os procedimentos de controle envolvidos no estabelecimento de uma conexão virtual através da rede, com o objetivo de enviar um fluxo de dados entre dois equipamentos de usuário. Na Seção 3.1 encontra-se uma visão geral do funcionamento do MPLS, enquanto na Seção 3.2 é apresentado um histórico que mostra a evolução do MPLS. Uma diferenciação entre os conceitos de comutação e roteamento são apresentadas na Seção 3.3. Os elementos da arquitetura MPLS são apresentados na Seção 3.4 e os conceitos básicos tais como: rótulo, FEC, tabelas de encaminhamento, entre outros, na Seção 3.5. A Seção 3.6 trata dos procedimentos básicos para a construção dinâmica de Lsp como políticas de distribuição de rótulos, seu controle, disparo e retenção de rótulos. Finalmente, a Seção 3.7 aborda o papel dos roteadores LER e LSR em um domínio MPLS.

3.1. Visão Geral

A idéia básica por trás do funcionamento do MPLS (*Multi-Protocol Label Switching*) é bastante simples [Rosen01]: consiste na geração e uso de um pequeno rótulo de tamanho fixo, que é usado como argumento para a tomada de decisões de encaminhamento de pacotes. Também com a finalidade de enviar pacotes até seu destino, a metodologia adotada pelo protocolo IP faz uso de um campo no cabeçalho dos pacotes, que contém o endereço para onde eles devem ser encaminhados, sendo que essa informação é processada em todos os roteadores no caminho do pacote através da rede, em um processo conhecido como roteamento *hop-by-hop*.

A Figura 3.1 apresenta um modelo simplificado de um domínio MPLS com seus componentes principais. Um domínio MPLS é constituído por um conjunto de roteadores que implementam as funcionalidades do MPLS e que estão sujeitos a

uma mesma autoridade administrativa, que é responsável pela configuração desses equipamentos. No MPLS, os pacotes IP são encapsulados com estes rótulos pelo roteador de borda do domínio MPLS, denominado LER (*Label Edge Router*), que analisa o conteúdo do cabeçalho IP e seleciona um rótulo apropriado com o qual encapsular o pacote. Esse procedimento de análise e seleção de um rótulo é conhecido como classificação de pacotes. Vale destacar que diferentemente do roteamento IP convencional, esta análise pode ser baseada em vários campos do cabeçalho IP e não apenas no campo de endereço do destinatário. Isto dá grande poder e flexibilidade ao MPLS, principalmente no que concerne à Engenharia de Tráfego (TE).

Em todos os nós subseqüentes dentro da rede, chamados de LSR (*Label Switched Router*), é o rótulo MPLS e não o cabeçalho IP que será usado na tomada da decisão de encaminhamento do pacote. Finalmente, quando os pacotes MPLS deixam a rede, outro roteador de borda remove os rótulos. O caminho por onde os pacotes viajam em um domínio MPLS é chamado de LSP (*Label Switched Path*). Para construir dinamicamente um LSP, utiliza-se um protocolo de distribuição de rótulos, que deve estar disponível em todos os roteadores do domínio MPLS. Uma vez que o LSP tenha sido estabelecido, os pacotes MPLS podem ser encaminhados com base no rótulo inserido no cabeçalho dos pacotes.

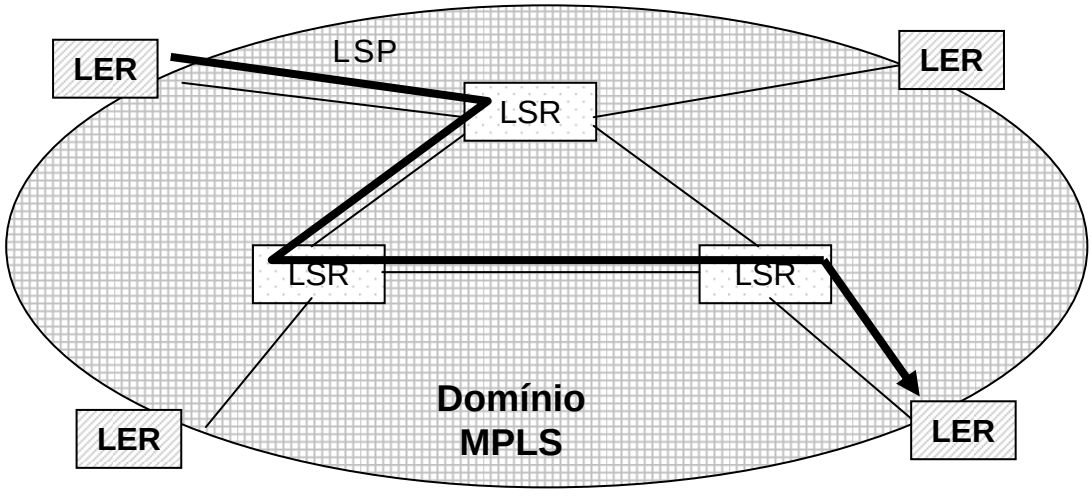

Fig. 3.1 – Modelo Simplificado de uma Rede MPLS

Em função de suas características de projeto, o MPLS tem se mostrado uma solução versátil, não apenas no que tange à melhoria da velocidade de encaminhamento dos pacotes nos roteadores, como também no oferecimento de novas capacidades para redes IP, como, por exemplo: escalabilidade, TE, suporte a QoS e provimento de VPNs [Extreme01, Viswanathan01].

3.2. Histórico

Considerando que o conjunto de protocolos TCP/IP (especialmente o próprio protocolo IP) é, atualmente, o padrão utilizado na maioria das redes públicas e privadas de dados e, que a convergência esperada de redes de voz, dados e vídeo também deve ser largamente baseada nestes mesmos protocolos, constata-se a necessidade de novos esquemas de roteamento e comutação que apresentem maior desempenho. É nesse contexto que a comutação por rótulo ressurgiu, como uma das alternativas ao desafio de se criar redes públicas integradas, a exemplo de redes IP avançadas que suportem distribuição de pacotes em tempo real, integração de IP com protocolos de redes herdadas como ATM e FR, e redes públicas de grande abrangência geográfica.

Vale ressaltar ainda que no contexto dessa solução, deve-se levar em conta aspectos como: demandas recorrentes por aumento de banda, diferenciação de serviços e flexibilidade de uso, entre outros.

O paradigma da comutação por rótulo atende estes requisitos, a partir da combinação do que há de melhor na comutação de nível 2 e no roteamento de nível 3, ou seja: a alta velocidade de operação da comutação em nível dois, com a grande flexibilidade do processo de roteamento da camada de rede. Desta forma, melhora-se a relação preço-por-desempenho do roteamento da camada três, facilita-se a escalabilidade através da agregação de tráfego, e garante-se maior flexibilidade na entrega de novos serviços de roteamento, aumentando, por conseguinte, o potencial de TE, além de se prover suporte a entrega de serviços com garantias de QoS [Hagard].

Até a constituição de um Grupo de Trabalho no IETF com vistas à especificação de um padrão, várias empresas desenvolveram soluções proprietárias. A seguir são citados quatro antecedentes históricos à padronização do MPLS:

a) O **CSR (*Cell-Switching Router*)** que foi desenvolvido pela Toshiba e apresentado à IETF em 1994 [Katsube97].
b) O ***IP Switching***, desenvolvido pela Ipsilon (parte da Nokia atualmente), foi anunciado no início de 1996 e entregue em alguns produtos comerciais. O *IP Switching* caracterizava-se por usar a presença de fluxos de dados para orientar o estabelecimento de rótulos [Newman96].
c) O ***Tag Switching*** é a comutação por rótulo desenvolvida pela Cisco. Em contraste ao CSR e ao IP Switching, o Tag Switching é uma técnica orientada por controle, que não depende da identificação de um fluxo de dados para estimular a montagem das tabelas de encaminhamento de rótulo em um roteador [Rekhter97].
d) O ***Aggregate Route-based IP Switching (ARIS)*** foi a proposta desenvolvida pela IBM. Sua arquitetura é similar ao Tag Switching. O ARIS ligava rótulos a rotas agregadas (grupos de prefixos de endereço) ao invés de fluxos individuais (diferente de CSR e IP Switching). Ligações de rótulo e caminhos comutados por rótulo eram montados em resposta ao tráfego de controle (assim como as atualizações de roteamento) ao invés de fluxo de dados, com o roteador de saída geralmente sendo o iniciador do processo [Feldman97, Viswanathan98].

Uma vez que as múltiplas soluções proprietárias para comutação baseada em rótulo não apontavam para uma direção amplamente aceitável, ficou evidente a necessidade de definição de um novo padrão e de formação de um Grupo de Trabalho IETF, com o objetivo de “integrar o paradigma de encaminhamento de comutação por rótulo com o roteamento da camada de rede”, com foco inicial no IPv4 e IPv6 e que garantisse a interoperabilidade entre os diversos fabricantes. Assim, o MPLS foi proposto na RFC 3031.

3.3. Roteamento x Comutação

Alguns conceitos básicos, que se aplicam a qualquer tecnologia de comutação, precisam ser revistos antes de se descrever o *modus operandi* do MPLS, entre os quais: roteamento e comutação.

a) ***Roteamento Nível 3***

É um termo utilizado para descrever as ações tomadas pela rede para mover pacotes em seu domínio, de forma tão eficiente quanto possível, da origem

ao destino. Assim, os pacotes trafegam pela rede sendo enviados de uma máquina para outra.

Para cumprir esta tarefa, os roteadores, quando recebem um pacote e têm de tomar uma decisão de encaminhamento, pesquisam a tabela de roteamento usando o endereço IP de destino do pacote como um índice, obtendo assim a identificação da máquina que é o próximo nó. A construção da tabela e seu uso para pesquisa na hora do encaminhamento são atividades separadas e independentes. A tabela de roteamento pode ser construída de forma estática ou dinâmica. Entradas estáticas nessa tabela são realizadas através da intervenção manual do administrador do roteador, que insere e retira informações sobre roteamento através de comandos do Sistema Operacional do equipamento.

Entradas dinâmicas na tabela de roteamento, por outro lado, são realizadas através de protocolos de roteamento habilitados pelo administrador do equipamento especificamente para esse fim.

Os protocolos de roteamento, como por exemplo, o RIP (*Routing Intenet Protocol*) e o OSPF (*Open Shortest Path Fisrt*) desempenham papel essencial na construção dinâmica da tabela de roteamento, haja vista que eles são usados para distribuir informação sobre a topologia da rede, de forma a permitir que cada roteador saiba que outra máquina é o próximo nó que um pacote deve tomar para chegar a seu destino.

b) ***Comutação Nível 2***

É um termo geralmente usado para descrever a transferência de dados de uma porta de entrada para uma porta de saída de uma máquina, onde a seleção da porta de saída é baseada em informações contidas no cabeçalho do quadro da camada dois. A comutação por rótulo é uma forma avançada de encaminhamento de pacotes, que substitui o tradicional encaminhamento baseado unicamente no caminho mais curto por um algoritmo de troca de rótulo mais eficiente.

Há três diferenças importantes entre a comutação por rótulo e o roteamento convencional durante a fase de encaminhamento de dados, conforme apresentado na Tabela 3.1:

	Roteamento Convencional	Comutação por Rótulo
Análise completa do Cabeçalho IP	Ocorre em todo nó	Ocorre apenas uma vez na borda da rede quando o rótulo é designado
Suporte *unicast* e *multicast*	Requer múltiplos algoritmos de encaminhamento complexos	Um único algoritmo de encaminhamento é exigido
Decisões de roteamento	Baseada em endereço de destino apenas	Pode ser baseado em quaisquer parâmetros, tais como: endereço destino, parâmetros de QoS e associações de VPN

Tab. 3.1 – Diferenças entre o Roteamento Convencional e a Comutação por Rótulo Durante a Fase de Transferência de Dados

3.4. Elementos da Arquitetura MPLS

Como visto, o MPLS é um protocolo de comutação de pacotes baseado em troca de rótulos, que une a flexibilidade do IP com a estabilidade dos circuitos virtuais. De certa forma, ele adiciona a característica de “orientação à conexão” às redes IP. Ademais, ele é multi-protocolo porque funciona independentemente dos protocolos das camadas dois e três. Conforme apresentado na Figura 3.2, pacotes gerados por qualquer protocolo da Camada 3 podem receber um prefixo contendo o rótulo MPLS e são em seguida encapsulados em quadros, qualquer que seja o protocolo adotado na Camada 2.

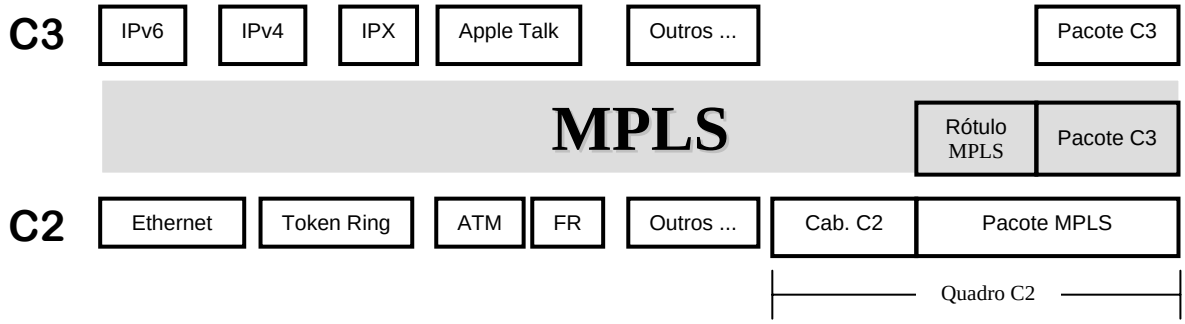

Fig. 3.2 – Independência do MPLS das Camadas 2 e 3

Na arquitetura MPLS dois componentes básicos são definidos: o plano de dados, ou componente de encaminhamento de pacotes e o plano de controle, ou componente de controle, conforme apresentado na Figura 3.3 [Papelnjack00].

É interessante perceber que todos os *switches* e roteadores executam essas duas funções, independente de implementarem protocolos orientados à conexão ou não, de processarem pacotes, quadros ou células. Em um roteador IP,

por exemplo, a função de controle envolve o cálculo da rota quando o roteamento dinâmico está habilitado; para isso, o roteador usa em geral, informação do estado do enlace ou o vetor distância, contida nos protocolos de roteamento, para atualizar suas tabelas de rotas. O encaminhamento de dados, por sua vez, é uma função totalmente separada; o roteador simplesmente analisa cada pacote recebido e, baseado em seu endereço IP destino, consulta a tabela de roteamento para determinar adequadamente qual o próximo nó. Vale ressaltar que, uma vez que o encaminhamento MPLS é baseado em rótulos, é possível separar claramente o plano de encaminhamento (baseado em rótulo) do plano de controle. Assim, cada plano pode ser modificado de forma independente. Com isto, não é preciso mudar o esquema de encaminhamento, por exemplo, para migrar para uma nova estratégia de roteamento ou de construção das tabelas de encaminhamento de nível 2 dentro da rede.

a) O ***Plano de Controle***:

É um componente presente em todo roteador MPLS e suas funções incluem: distribuir a informação de roteamento entre os LSR adjacentes de forma consistente, caso o roteamento nível 3 seja dinâmico, e executar os procedimentos que convertem as informações trocadas, de forma a construir e manter sua própria tabela de encaminhamento baseada em rótulos.

Para cumprir estas funções, este componente inclui, caso o roteador seja capaz de utilizar roteamento dinâmico nível 3, pelo menos um dentre os protocolos de roteamento tradicionais, tais como: OSPF, BGP, RIP, PIM (*Protocol Independent Multicast*), entre outros, que são usados para trocar informação de controle entre os componentes de controle de diferentes roteadores. Vale ressaltar que o componente de controle tem de reagir, quando ocorrem mudanças na rede, como uma falha de conexão, por exemplo. São eles que provêem os roteadores MPLS com o mapeamento entre a FEC e o endereço do próximo nó, para as FEC atualmente cosideradas.

Entretanto, se o roteamento nível 3 for estático, a construção da tabela de roteamento desse nível é feita de forma manual e os roteadores não são capazes de reagir em casos de mudanças na topologia da rede de forma automática.

Independente do tipo de roteamento nível 3 adotado, estático ou dinâmico, e da forma como a tabela de roteamento nível 3 foi construída, a função primordial

do componente de controle está no elemento de Controle de Roteamento MPLS – nível 3 (IP), responsável pela geração da tabela de encaminhamento por rótulo. Para executar essa função o roteador MPLS deve criar uma associação entre informações de encaminhamento de nível 3 (por exemplo, o endereço de destino de um pacote) com rótulos MPLS, de forma a permitir a construção de um caminho comutado a rótulo através da rede.

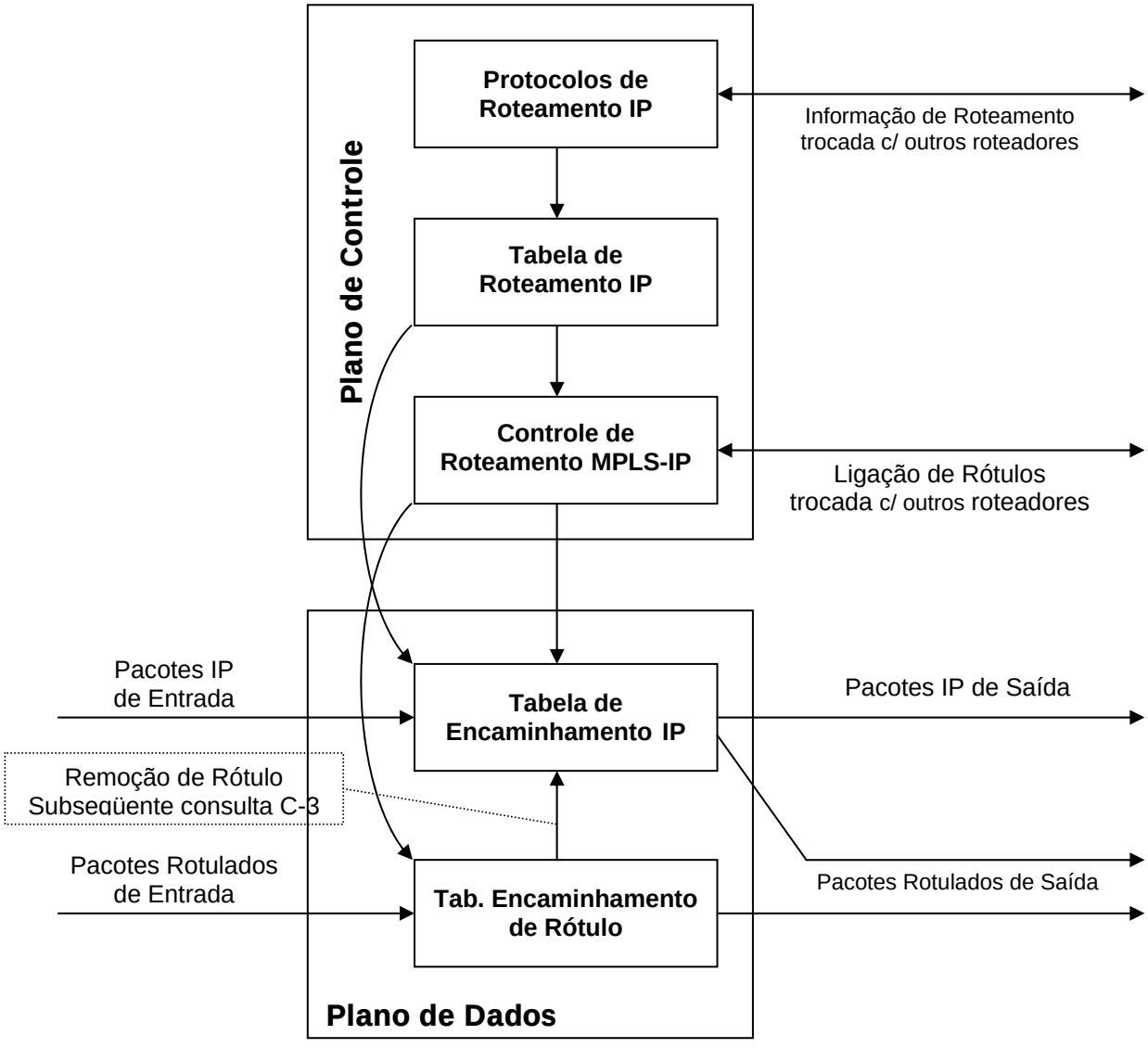

Fig. 3.3 – Arquitetura Básica de um Nó MPLS

Essa associação também pode ser feita de forma estática ou dinâmica. Caso a associação seja feita de forma estática, o administrador do equipamento deve introduzir “manualmente” as informações necessárias para tal associação.

Esse procedimento é claramente restritivo e seria altamente ineficaz em redes de núcleo. Em geral, é a forma dinâmica que é encontrada nos roteadores MPLS. Nesse caso, protocolos de distribuição de rótulos são utilizados para implementar, no elemento de Controle de Roteamento MPLS – nível 3 (IP), a comunicação com outros roteadores MPLS e distribuir as informações relativas aos rótulos.

b) O ***Plano de Dados:***

Este componente, por sua vez, executa o encaminhamento real dos pacotes. Ele usa a informação da tabela de encaminhamento – como mantida pelo roteador, e o rótulo que é transportado pelo próprio pacote, além de um conjunto de procedimentos locais, para tomar decisões de encaminhamento. Em um roteador convencional, um algoritmo que utiliza o critério do caminho mais curto compara o endereço de destino (nível 3) no pacote com entradas na tabela de encaminhamento, até obter a melhor rota disponível. Todo o processo de tomada de decisão completo deve ser repetido em cada nó ao longo do caminho da origem ao destino. Em um LSR, um algoritmo de troca de rótulo usa o rótulo no pacote e uma tabela de encaminhamento baseado em rótulo, para obter um novo rótulo e uma interface de saída para o pacote.

A operação do plano de dados em um domínio MPLS pode ser caracterizada em três momentos distintos: no LER de entrada, nos LSR, e no LER de Saída [Papelnjack00].

- **Encaminhamento no LER de entrada**

Ao chegar um pacote IP, o LER de entrada do domínio MPLS, executa uma consulta-nível-três na tabela de encaminhamento IP, classifica o pacote com base no resultado dessa consulta (em uma FEC) e o marca o pacote com o rótulo de saída correspondente a informação nível 3 obtida (àquela FEC). Finalmente, ele encaminha o pacote para a interface de saída com o rótulo apropriado, segundo informações da tabela.

Uma observação importante é que, no caso da chegada de um pacote IP que será roteado de forma convencional, sem auxílio das funcionalidades do MPLS, a informação de nível 3 corresponde somente à sub-rede destino e a classificação de pacote se resume a uma consulta-nível-três tradicional à tabela de encaminhamento IP.

- **Encaminhamento nos LSR (núcleo)**

Um LSR recebe o pacote rotulado, executa uma consulta-de-rótulo, usa as tabelas de encaminhamento de rótulo para trocar o rótulo de entrada (rótulo do pacote que está chegando) pelo rótulo de saída correspondente e o encaminha para o próximo nó, conforme informações armazenadas na tabela de encaminhamento. No caso de pacotes não rotulados, o encaminhamento é feito de acordo com os algoritmos tradicionais de consulta-nível-três.

- **Encaminhamento no LER de Saída**

O LER recebe o pacote rotulado, executa uma consulta-de-rótulo, remove o rótulo, executa uma consulta-nível-três tradicional no pacote IP resultante e o encaminha para o próximo roteador externo ao domínio MPLS.

3.5. Conceitos Básicos

3.5.1. O Rótulo

A arquitetura MPLS define o rótulo como sendo um identificador não estruturado, de tamanho fixo, e relativamente pequeno, que pode ser usado para auxiliar no processo de encaminhamento, como os DLCI (*Data Link Control Identifier*) usados em redes FR ou os VPI/VCI (*Virtual Path Identifier/Virtual Channel Identifier*) usados em ambientes ATM.

Normalmente, os rótulos MPLS são de significado local, restrito a um simples enlace de dados entre dois roteadores do domínio, sem nenhum significado global, como teria um endereço de rede, por exemplo.

Na Figura 3.4 a seguir é apresentado o formato de um cabeçalho MPLS, chamado de rótulo SHIM, que pode ser utilizado para encapsular um quadro de enlace que não disponibiliza um campo para inserção de rótulo, a exemplo dos quadros Ethernet.

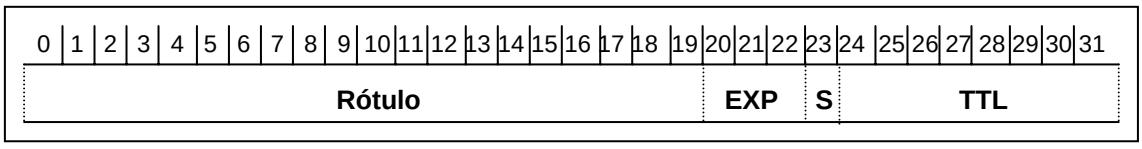

Fig. 3.4 - Formato do Cabeçalho MPLS

Os campos do cabeçalho MPLS apresentam os seguintes significados:

- **Rótulo**: Um identificador local usado para representar uma dada FEC (*Forwarding Equivalence Class*) durante o processo de encaminhamento. É um campo de 20 bits;
- **EXP**: É usado para implementações de QoS. É um campo de 3 bits;
- **S**: Usado para indicar a presença de uma pilha de rótulos. Se o rótulo é único ou o último da pilha, o bit é zero, senão ele é setado em um. Esta possibilidade de inserir múltiplos rótulos dá ao MPLS a possibilidade de suportar roteamento hierárquico. É um campo de 1 bit;
- **TTL**: Provê funcionalidade tempo de vida, semelhante ao IP convencional. É um campo de 8 bits.

Como os rótulos têm significado local, para identificar inequivocamente um caminho único através da rede, basta que os valores dos rótulos do tipo SHIM sejam únicos no escopo do equipamento, ou seja, no mesmo roteador MPLS não se deve usar o mesmo valor de rótulo para indicar mais de um caminho de saída. Entretanto, o mesmo valor pode ser usado em roteadores diferentes para indicar caminhos de saída diferentes, sem que isso venha a causar confusão. Diz-se então que o escopo do espaço de rótulos é por plataforma.

Por outro lado, caso se utilize rótulos do tipo DLCI, em enlaces FR, ou VPI/VCI, em enlaces ATM, basta que o escopo do espaço de rótulos seja por interface do equipamento, não havendo necessidade de unicidade a nível de todo o equipamento.

3.5.2. A FEC

Esse termo foi introduzido pelos padrões MPLS para denotar classes de encaminhamento de pacotes [Hagard]. Desta forma, pode-se definir uma FEC como sendo uma representação de um grupo de pacotes que pode ser tratado de uma maneira equivalente, ou seja, possuem exigências de serviço similares, para propósitos de encaminhamento através da rede [Mplsrc01].

Um exemplo de FEC é o conjunto de pacotes *unicast* cujos endereços destino casam com um dado prefixo de endereço IP. As FEC podem ser definidas com diferentes níveis de granularidade, tais como: por sub-rede de destino, por *host* ou por aplicação, conforme apresentado na Figura 3.5:

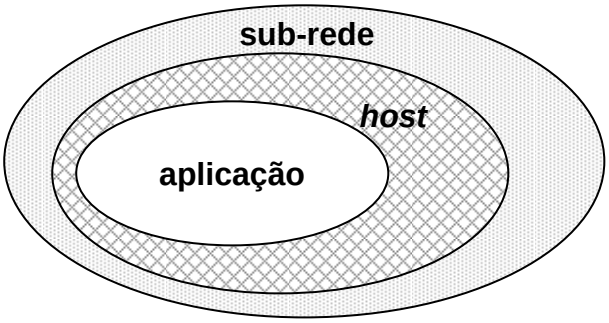

Fig. 3.5 – Granularidade da FEC

Uma FEC pode ser especificada como um conjunto de um ou mais elementos FEC, onde cada elemento identifica um conjunto de pacotes a ser mapeado no LSP correspondente. Os tipos de elementos FEC atualmente considerados na norma são um prefixo de endereço de rede (de qualquer comprimento) e um endereço de rede de um host. Entretanto, outros elementos FEC podem vir a ser considerados, ampliando o escopo para a tomada de decisões de encaminhamento de pacotes na rede, conforme já mencionado.

3.5.3. O Processo de Ligação Rótulo-FEC

Os rótulos são ligados a uma FEC e, conseqüentemente tornam-se significativos para o encaminhamento de pacotes, através de um processo de ligação (*bind*).

A ligação Rótulo-FEC é executada no plano de controle e é fundamental na construção do caminho que o pacote deve seguir através da rede, uma vez que essa ligação permite a geração da tabela de encaminhamento por rótulo em cada roteador.

Um detalhe importante a ser percebido é que no MPLS a decisão de ligar um rótulo **X** a uma FEC **Y** é feita pelo roteador posterior (*downstream*) no sentido da transmissão do pacote em relação à tal ligação e as ligações dos rótulos são distribuídas no sentido contrário à transmissão dos pacotes (*downstream to upstream*). Ademais, o Roteador anterior (*upstream*) – Ru e o posterior (*downstrem*) – Rd devem concordar na ligação rótulo **X –** FEC **Y** para pacotes em trânsito de Ru para Rd. Assim, o rótulo **X**, que é um valor arbitrário e cuja ligação à FEC **Y** é local ao enlace entre **Ru** e **Rd**, torna-se o rótulo de saída de **Ru** representando a FEC **Y** e o rótulo de entrada de **Rd** representando a mesma FEC **Y**.

Esse processo de ligação Rótulo-FEC é disparado por algum evento significativo na rede. Estes eventos podem ser classificados em duas categorias:

a) **Ligações orientadas por dados** (***data-driven***) ocorrem quando o tráfego de pacotes que começa a fluir através de um LER é reconhecido como um fluxo e, portanto candidato à comutação por rótulo. Neste caso, ligações de rótulo são estabelecidas somente quando necessário, resultando em entradas na tabela de roteamento por rótulo somente para fluxos de tráfego IP e não para pacotes isolados.

b) **Ligações orientadas por controle** (***control-driven***) são estabelecidas como resultado de uma atividade do plano de controle e são independentes da existência de fluxos de dados. Estas ligações podem ser estabelecidas em resposta a atualizações de roteamento ou recebimento de mensagens do protocolo de distribuição de rótulos usado, como por exemplo, o LDP. Este tipo de ligação apresenta melhor escalabilidade do que às orientadas por dados e, por esta razão, é a preferida pela arquitetura MPLS.

3.5.4. Tabelas de Encaminhamento por Rótulo

Todos os roteadores MPLS devem ser capazes de, no mínimo, processar pacotes com rótulos de entrada, através de uma matriz de comutação (tabela "*cross-connect*"), ou tabela de encaminhamento. Por exemplo, se existir uma entrada na tabela de encaminhamento associando o rótulo 400 na porta de entrada A ao rótulo 450 na porta de saída B, o roteador MPLS deve ser capaz de realizar essa comutação de rótulos ao tratar os pacotes relacionados.

Essas tabelas são construídas, de forma estática ou dinâmica, e são constituídas por várias entradas chamadas NHLFE (do inglês, Entradas de Encaminhamento por Rótulo ao Próximo Roteador).

Cada NHLFE pode conter as seguintes informações:

- O endereço do próximo roteador (*next hop*) para o pacote;
- A operação a ser feita com a pilha de rótulos, a saber:
  - Substituir o rótulo no topo da pilha por um novo valor dado;
  - Retirar o rótulo do topo da pilha (operação *pop*);
  - Substituir o rótulo no topo da pilha por um novo valor especificado e depois inserir na pilha um ou mais nosovs rótulos especificados (operação *push*);

- Opcionalmente, o tipo de encapsulamento usado para transmitir o pacote no enlace de saída;
- Opcionalmente, a maneira de codificar a pilha de rótulos.

As tabelas construídas nos LER de Entrada são diferentes das tabelas construídas nos outros roteadores, pois destinam-se a diferentes funções.

Um LER de Entrada constrói uma tabela que mapeia FEC em NHLFE, que é chamada de Mapa FTN (FEC-To-NHLFE). Essa tabela é usada para encaminhar pacotes que chegam não rotulados ao LER, mas que precisam sair rotulados ao entrar num domínio MPLS. Isso é feito através de uma operação de *push* no pacote de entrada.

Um LSR constrói uma tabela que mapeia um rótulo em NHLFE, que é chamado de ILM (*Incoming Label Map*, ou Tabela de Rótulos Entrantes). Essa tabela é usada no Plano de Dados, para encaminhar pacotes rotulados. Se o ILM mapear um determinado rótulo de entrada em mais de um NHLFE, o LSR deve selecionar uma única entrada para realizar o encaminhamento do pacote, na fase de transmissão de dados. Esse esquema pode ser útil no balanceamento de carga entre múltiplos caminhos de igual custo. A operação prevista nas NHLFE dessa tabela é geralmente destinada à troca de rótulo (*label swap*), de forma a associar o rótulo de entrada de um pacote que chega através de uma determinada porta de entrada com um rótulo de saída para o mesmo pacote, ao ser transmitido por uma porta de saída do roteador.

Quando se utiliza o LDP, essas tabelas de encaminhamento são chamadas de LIB (*Label Information Base*).

3.6. Procedimentos Básicos para Construção Dinâmica de LSP

Conforme apresentado na Figura 3.1, o LSP é um caminho usado para tráfego de um conjunto específico de pacotes associados a uma FEC, através de uma rede MPLS. O LSP construído para uma FEC é unidirecional por natureza, sendo que o tráfego de retorno, se houver, deve tomar outro LSP.

A RFC 3031 define “um LSP de nível **m**” para um pacote P particular como sendo uma seqüência de roteadores LSR $<R_1, ..., R_n>$ com determinadas propriedades:

- R_1, o LSR de Ingresso, insere um rótulo na pilha de P, resultando numa pilha de profundidade **m**;
- Para todo LSR R_i (1<i<n), P tem uma pilha de profundidade **m** ao ser recebido pelo LSR;
- No trânsito de P entre R_1 e R_{n-1}, nunca P tem uma pilha de profundidade menor que **m**;
- Todo R_i (1<i<n) transmite P para R_{i+1} usando o rótulo no topo da pilha como índice para o mapa ILM;
- Se um sistema S transmite P entre R_i e R_{i+1}, a decisão de encaminhamento de S não se baseia no rótulo de nível **m** (e sim provalvemente em rótulos adicionais **m + k** que tenham sido introduzidos na pilha);
- Um LSP de nível **m** pode então ser visto como uma seqüência de roteadores:
  - Que começa com um LSR (LSR de Ingresso ou LER de Ingresso) que insere um rótulo de nível **m**;
  - Onde todos os LSR intermediários tomam sua decisão de encaminhamento trocando o rótulo de nível **m**;
  - Que termina em um LSR (LSR de Egresso ou LER de Egresso) onde a decisão de encaminhamento é feita por troca de rótulo de nível **m - k** (k > 0) ou através de procedimento não-MPLS.

Um LSP, usado para transmissão de pacotes, deve ser anteriormente construído. Caso se utilize de sinalização dinâmica nessa construção, identifica-se em geral os seguintes procedimentos nor protocolos relacionados:

- Controle da Distribuição dos Rótulos;
- Disparo da Distribuição dos Rótulos;
- Retenção de Rótulo nos Roteadores.

3.6.1. Políticas de Distribuição de Rótulos

Conforme já mencionado, a distribuição de rótulos pode ser feita através de dois métodos [Gallaher02b]:

a) **Programação Estática.**

Similar à maneira como um roteador é programado para roteamento estático. Neste caso, a programação estática elimina a habilidade de re-roteamento dinâmico ou gerenciamento do tráfego.

b) **Sinalização Dinâmica e Distribuição de Rótulo**.

Esta opção leva em conta a realidade das redes contemporâneas que mudam em uma base dinâmica. Há vários protocolos disponíveis para esse fim, como por exemplo: o LDP, desenvolvido especificamente no contexto do MPLS; o RSVP-TE (*RSVP – Traffic Engineering*), uma extensão do RSVP que inclui funcionalidades de Engenharia de Tráfego; além de outros protocolos de roteamento que estenderam suas funcionalidades, como o BGP-TE (*Border Gateway Protocol – Traffic Engineering*) e o OSPF-TE. Cada um apresenta suas vantagens e desvantagens. Contudo, apesar das diferenças, os conceitos básicos da sinalização e distribuição de rótulos permanecem consistentes entre os protocolos. Vale destacar que, de acordo com o protocolo adotado, o rótulo pode ser carregado de duas maneiras:

1ª) **De carona em um protocolo de roteamento.** Nesta solução a informação de ligação de rótulo é adicionada aos protocolos de roteamento tradicionais para distribuição, garantindo a consistência da informação de encaminhamento e evitando a sobrecarga (*overhead*) na implementação de outro protocolo. Entretanto, esta não é uma solução completa, pois nem todas as sub-redes usam roteamento dinâmico e nem todos os protocolos de roteamento são capazes de tratar facilmente de rótulos.

2ª) **Através do uso de um protocolo de distribuição de rótulo específico.** Este método exige o uso de protocolos específicos para essa finalidade, a exemplo do novo protocolo usado unicamente para a distribuição de informação de ligação de rótulo chamado de LDP, ou mesmo a utilização do já bem estabelecido RSVP-TE. A desvantagem deste método de distribuição é que ele aumenta a complexidade do elemento de comutação, uma vez que outro protocolo deve ser suportado, e seu uso precisa ser coordenado com a operação dos protocolos de roteamento associados.

3.6.2. Controle da Distribuição de Rótulos

Há dois métodos para controlar a distribuição de rótulos [Rosen01]:

a) **Controle Independente**. Ocorre quando não há nenhum roteador específico que gerencie ou controle a distribuição dos rótulos. Neste caso, cada roteador tem a capacidade de processar os protocolos de roteamento, montar suas tabelas de comutação e distribuí-las.

b) **Controle Ordenado**. Neste caso, um roteador, tipicamente o LER de saída, é responsável pelo processo de distribuição dos rótulos, que pode ser disparado de forma não solicitada ou sob demanda.

A seguir, é apresentada a Tabela 3.2 comparando os dois métodos:

	Controle Independente	**Controle Ordenado**
Tempo de Convergência	Tempo de convergência de rede mais rápido	Tempo de convergência maior que o independente
Gerenciamento de Rótulos	Qualquer roteador que identifique uma mudança de roteamento pode repassar esta informação para os demais e não se constitui num ponto simples de falha	O gerenciador de rótulos constitui-se em um importante ponto de falha
Controle de Tráfego	Não há um ponto de controle de tráfego, o que torna a Engenharia mais difícil	Melhor TE e controle de rede mais rígido

Tab. 3.2 – Comparação entre os Métodos: Controle Independente x Controle Ordenado

3.6.3. Disparo da Distribuição de Rótulos

Dentro do controle de distribuição de rótulos ordenado, há dois métodos principais para disparar a distribuição de rótulos [Rosen01]:

a) **DOU – *Downstream Unsolicited***: Os rótulos são atribuídos com base nas decisões do roteador gerenciador de rótulos. Os rótulos são enviados pelo gerenciador de rótulos de forma não solicitada. Destarte, o gerenciador de rótulos pode usar como ponto de disparo (gatilho), um intervalo de tempo pré-definido e, assim, enviar rótulos ou mensagens de refrescamento de rótulo a intervalos de tempo regulares. Uma outra possibilidade seria o gerenciador de rótulo usar mudanças na tabela de roteamento padrão como um gatilho, ou seja, quando um roteador mudar uma informação de rota, o gerenciador de rótulos pode enviar atualizações de rótulos a todos os roteadores afetados. A Figura 3.6 abaixo ilustra o mecanismo de funcionamento deste modo.

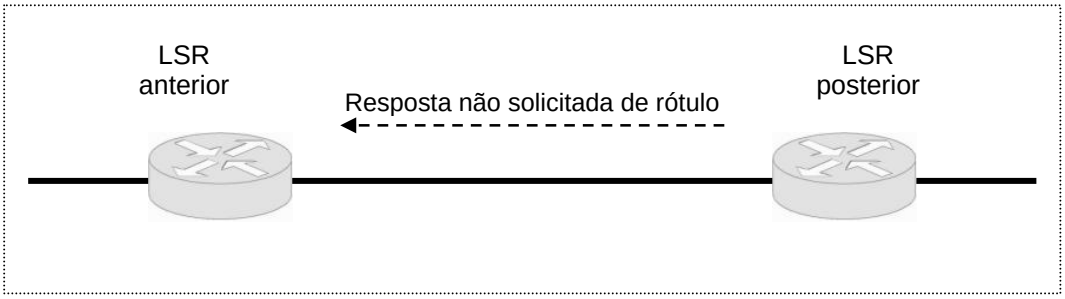

Fig. 3.6 – Modo *Downstream* não-Solicitado

b) **DOD – *Downstream On-Demand***: Nesse método, os rótulos são requisitados pelos roteadores: ou seja, primeiro os rótulos são requisitados e, no segundo passo, os rótulos são enviados, conforme apresentado na Figura 3.7.

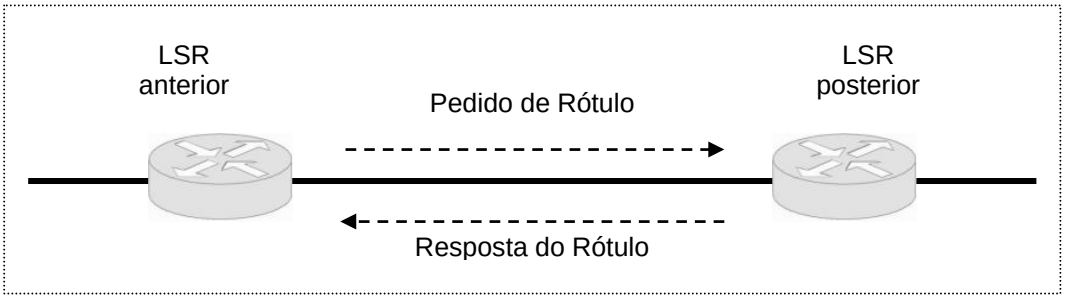

Fig. 3.7 – Modo *Downstream* sob Demanda

O modo DOU está freqüentemente associado à estratégia baseada em topologia, onde os rótulos são associados a entradas na medida em que elas são introduzidas na base de roteamento. Já o modo DOD provê um ambiente mais controlado, beneficiando quanto à prevenção de "*loops*" e garantindo a utilização de FEC consistentes através da rede.

3.6.4. Retenção do Rótulo

A arquitetura MPLS define dois modos para retenção do rótulo, ou seja, para o tratamento de ligações de rótulo de uma dada FEC, provenientes de um LSR posterior (*downstream*) que não é o próximo nó para uma dada FEC:

a) **Conservador**: neste modo as ligações são mantidas apenas se elas forem recebidas de um LSR posterior (*downstream*), que tenha sido selecionado pelo LSR anterior (*upstream*) como sendo o próximo salto para a referida FEC. Nos demais casos, as informações sobre ligações são descartadas. Assim, o LSR mantém poucos rótulos.

b) **Liberal**: neste modo, todas as ligações são mantidas, o que garante uma adaptação mais rápida a alterações na topologia e permitindo o chaveamento do tráfego para outros LSP em caso de mudanças, uma vez que as ligações para caminhos alternativos já foram estabelecidas.

3.7. O Papel dos Roteadores na Arquitetura MPLS

De forma genérica, um roteador MPLS é qualquer equipamento que suporta o componente de controle (protocolos de roteamento, protocolos de distribuição de rótulos e outros) e o componente de encaminhamento baseado na

troca de rótulo. A Figura 3.1 mostra uma rede de comutação por rótulo simplificada e ilustra os LER, provendo as funções de entrada e saída e, os LSR, provendo comutação em alta velocidade. Nota-se que o encaminhamento baseado em rótulo complementa o roteamento tradicional, mas não o substitui completamente.

Agora que os principais conceitos sobre o MPLS foram apresentados, é possível resumir o papel dos roteadores nessa arquitetura da seguinte forma:

3.7.1. O LER

O LER desempenha basicamente três funções principais: classificar o tráfego, e aplicar e remover os rótulos. Como dito anteriormente, os rótulos podem ser indicados com base em outros fatores, além do endereço de destino. Além disso, o LER implementa gerenciamento de acordos e controles de acesso, e executa agregação de tráfego em fluxos maiores quando possível, constituindo-se em um ponto de controle e gerenciamento para os provedores de serviço. É desejável ainda que os LER tenham as seguintes capacidades:

a) **Alta velocidade na classificação de fluxo IP**

Isto permitirá que estes equipamentos indiquem valores de QoS e apliquem rótulos para fluxos IP sem qualquer degradação no desempenho de encaminhamento; e,

b) **Possibilidade de extensão das capacidades de VPN**

Para obter vantagem do MPLS quando da provisão de VPN, estes equipamentos devem ser capazes de rodar múltiplas tabelas de encaminhamento, de forma que os clientes VPN possam ser tratados de forma separada dentro do LER.

Portanto, as capacidades dos LER são chave para o sucesso na implementação de um ambiente totalmente baseado na comutação por rótulo.

3.7.2. O LSR

É um equipamento roteador de alta velocidade localizado no centro de uma rede MPLS, que participa do estabelecimento dos LSP usando os protocolos de sinalização de rótulo apropriados e do chaveamento em alta velocidade do tráfego de dados baseado nos caminhos estabelecidos. Para não comprometer o desempenho de redes com alto volume de tráfego, esse equipamentos precisam ser capazes de processar os pacotes rotulados em altíssima velocidade.

Capítulo 4
O MPLS e as Redes Multi-Serviço

O presente capítulo apresenta os principais conceitos envolvidos no contexto das Redes Multi-serviço e da Engenharia de Tráfego, procurando colocá-los numa ordem lógica e sistemática. Na Seção 4.2 são explicitados os requerimentos para uma rede verdadeiramente multi-serviço. Na Seção 4.3 é formulado um conceito para Engenharia de Tráfego (TE), bem como seu objetivo primário. Questões de roteamento e de reserva de recursos também são abordadas. Como o MPLS é uma resposta importante às necessidades de TE, as seções seguintes dedicam-se a fazer a ligação entre os conceitos supra-citados e o MPLS. Nesse sentido, a Seção 4.4 trata da TE no MPLS, enquanto a Seção 4.5 versa sobre a cooperação do MPLS e do DiffServ como tecnologias de suporte a QoS. A Seção 4.6 apresenta um comparativo entre o CR-LDP e o RSVP-TE, como alternativas de Protocolo de Sinalização para o MPLS. Finalmente, na Seção 4.7 é detalhado o funcionamento do RSVP-TE, por ser o protocolo adotado nesse trabalho.

4.1. Introdução

As principais tecnologias de interconexão usadas até recentemente nos *backbones* das redes Internet, a saber: ATM e FR, apesar de suas reconhecidas qualidades, apresentam uma significativa desvantagem para os ISP, que é a falta de escalabilidade. Destarte, elas não provêem suficiente flexibilidade para alocação dinâmica de banda, de forma a atender a um maior número de clientes comerciais.

Ademais, percebe-se uma tendência, praticamente irreversível, de uma convergência de todos os principais serviços, como telefonia, televisão e vídeo-conferência, serem executados sobre o protocolo IP. Assim, com a maioria do tráfego sendo baseado nesse protocolo, faz sentido tanto do ponto de vista operacional quanto comercial, a tentativa de otimizar o uso dos recursos de rede sobre um núcleo IP [Integral02a]. Portanto, as redes multi-serviço se justificam em função dessa tendência de convergência de todas as mídias sobre o IP.

Nesse contexto, como apresentado em capítulos anteriores, o MPLS surge como uma excelente alternativa, pois une o que há de melhor dos mundos IP e

ATM, com alto grau de escalabilidade, graças às suas capacidades de gerenciamento, que se baseiam em técnicas e princípios de TE. Assim, tipos específicos de tráfego podem receber rótulos diferentes e conseqüentemente características de fluxo, caminhos e prioridades diferentes através da rede. Isto permite uma relação direta entre os rótulos de significado local e o processo de sinalização “externo”, que é de significado em toda a rede. Desta forma, os LSP podem ser criados com atributos específicos, sensíveis aos serviços a eles associados, usando protocolos de sinalização como RSVP-TE, CR-LDP, OSPF-TE, entre outros [Liquidlight01].

4.2. Redes Multi-Serviço

Em função do crescimento exponencial da Internet, já comentado nos capítulos anteriores, tem havido um grande esforço no sentido de aumentar as funcionalidades das redes Internet, através da adição de certas características, tais como: provisão de QoS, suporte a TE e VPN [Integral02a], que favorecerão a implantação de redes multi-serviço.

Pode-se dizer que as redes verdadeiramente multi-serviço requerem [Integral02b]:

- Uma metodologia de **QoS avançada**, que permita que serviços diferenciados operem na infra-estrutura Internet, habilitando entrega garantida e tratamento de tráfego prioritário, aumentando assim as oportunidades de receita para as operadoras;
- **Habilidades no que tange a TE**, para evitar congestionamentos na Internet. Além disso, vale ressaltar que TE também deve estar associada aos esforços de provimento de QoS;
- **Capacidade de garantir a alocação de recursos destinados ao tráfego** de um determinado fluxo de dados, a fim de possibilitar o cumprimento dos acordos firmados através dos SLA.

Vale ressaltar que tais funcionalidades não estão presentes nos roteadores IP herdados.

Neste contexto, como é mostrado nas próximas seções, é possível afirmar que o MPLS atende a estas demandas, pois uma vez que a rede MPLS está estabelecida, protocolos de roteamento adicionais, ou extensões aos protocolos

existentes podem ser usados para ativar as capacidades acima descritas. Ressalta-se, porém, que parte da garantia de alocação de recursos e funcionalidades de TE deverá ser implementada nas camadas inferiores, uma vez que o MPLS, sozinho, não tem como resolver todos os problemas.

Vale destacar ainda que QoS e TE são alguns dos pré-requisitos para entrega de serviço VPN, que atenda a níveis de serviço garantidos e altos níveis de segurança.

4.3. Engenharia de Tráfego

Essa seção propõe uma sistematização dos vários conceitos relacionados à Engenharia de Tráfego, que foram encontrados dispersos em várias publicações da área.

4.3.1. Definição

A Engenharia de Tráfego se refere ao processo de seleção do caminho usado pelo fluxo de dados, com o objetivo de balancear a carga de tráfego nos vários enlaces, roteadores e *switches* da rede. Pode também ser descrita como sendo o processo através do qual um determinado dado é roteado pela rede, de acordo com uma visão de gerenciamento dos recursos disponíveis e sua relação com o tráfego real e o esperado, levando-se em conta ou não a CoS e a QoS exigida para o dado [Brittain00]. Assim, percebe-se claramente que a TE é mais importante em redes onde há múltiplos caminhos ligando seus vários nós.

Uma vez que ela fornece os meios necessários para os operadores fazerem um melhor uso dos recursos disponíveis na rede, distribuindo a carga entre os enlaces da rede, e permitindo que alguns enlaces sejam reservados para certas CoS ou clientes específicos, verifica-se que ela se torna uma ferramenta de suma importância para os ISPs.

4.3.2. Objetivos

O objetivo primário da TE é fazer o melhor uso da infra-estrutura de rede enquanto mantém garantias de QoS. Isto é facilitado pela característica de roteamento explícito do MPLS, que permite o suprimento potencial de muitas falhas associadas ao esquema de roteamento IP atual, tais como: o IP nem sempre faz um bom uso dos recursos de rede disponíveis e não é satisfatório do ponto de vista da

Engenharia de Tráfego, uma vez que os protocolos de roteamento dinâmico associados se limitam a buscar o caminho de menor custo [Aukia02].

Nesse cenário, um aspecto importante a ser ressaltado diz respeito ao controle da TE. Há duas formas principais de controle: manualmente, através da ação dos operadores da rede, ou de forma automatizada, através de processos que reagem a informações providas pelos protocolos de roteamento ou outros meios.

4.3.3. Questões de Roteamento

A TE tem sido geralmente associada ao roteamento *off-line*. Neste tipo de roteamento, assume-se que todos os túneis ou LSP que estão para ser roteados e suas requisições de recursos são conhecidos na hora em que o roteamento é feito. Esse enfoque gera alguns problemas, a exemplo de dificuldades na configuração de novas requisições de recursos, antes que as solicitações anteriores sejam concluídas. Assim, coloca-se a pergunta: como capturar a dinâmica da rede, em termos de recursos e QoS, em tempo real, e efetuar eventuais alterações no que tange ao roteamento? Acomodar novas requisições poderia exigir o re-roteamento de uma grande quantidade de LSP já estabelecidos [Aukia02].

Por outro lado, o roteamento *on-line*, que leva em conta o comportamento da rede em tempo real, para ser prático, pode depender apenas da informação obtida dos protocolos de roteamento, tais como OSPF ou IS-IS (*Intermediate System to Intermediate System*) com extensões apropriadas para disseminação dos parâmetros de QoS, seja a partir de um servidor de rotas centralizado ou distribuído.

A seguir são apresentados outros conceitos relacionados às questões de roteamento, a saber: rotas explícitas e rotas com restrição, roteamento baseado em restrições e roteamento baseado em QoS.

a) Rotas Explícitas e Rotas com Restrição

As Rotas Explícitas (*ER – Explicit Route*) podem ser entendidas como uma seqüência precisa de determinada rota desde o roteador de entrada no domínio MPLS até o de saída. No MPLS, um LSP pode ser configurado para seguir um caminho explícito, de forma total ou parcial. Assim, a rota explícita pode ser do tipo estrita ou "*loose*", respectivamente. Uma rota estrita deve conter apenas os nós explicitados na ER, e deve usá-los de acordo com a ordem especificada. A rota "*loose*", por sua vez, deve incluir todos os nós especificados, mantendo a ordem,

mas podendo incluir nós adicionais, quando necessário, para alcançar os nós especificados [Brittain00].

No contexto do MPLS, o roteamento explícito é particularmente útil para forçar um LSP a seguir por um caminho diferente do oferecido pelo protocolo de roteamento. Assim, ele pode ser usado para diversos fins, como por exemplo: distribuir tráfego em uma rede congestionada, rotear por caminhos que evitem redes com falhas, prover LSP de reserva previamente alocados para proteger de falhas, entre outros.

O conceito de Rotas com Restrição (*Constraint Routes*), por sua vez, é mais abrangente, pois leva em conta o fato de que uma determinada rota, que um LSP pode tomar, pode ser restringida por exigências selecionadas no LER de entrada. A partir desta definição percebe-se que uma Rota Explícita é um caso particular de Rota com Restrição (ou Rota Restrita), onde a restrição é a ordem na qual os LSR serão alcançados e a identidade dos LSR que compõe o caminho dos dados na rede. Outras restrições podem ser impostas, como por exemplo uma descrição do tráfego que deve fluir na rota, que por sua vez pode incluir outras exigências como: valores de banda, atraso e prioridade.

No MPLS, a TE é provida de forma inerente, através do uso de caminhos roteados explicitamente. Para isso, há pelo menos duas abordagens principais possíveis, em termos de protocolo de sinalização, para suprir TE dinâmica e QoS: o CR-LDP e o RSVP-TE, os quais são apresentados na Seção 4.4.

b) Roteamento Baseado em Restrições

Atualmente, na Internet, o roteamento é baseado principalmente na topologia da rede. Ele tenta transferir cada pacote através do caminho mais curto da fonte para o destino. O roteamento baseado em restrições é uma extensão desse modelo básico baseado na topologia. Ele roteia pacotes baseados em múltiplas restrições. As restrições incluem a própria topologia da rede (caminho mais curto), e outros critérios como: disponibilidade de informação de recursos da rede (principalmente a disponibilidade de banda nos enlaces físicos), restrições de QoS ao fluxo e restrições de policiamento. Esta técnica pode ajudar a prover um desempenho melhor e aumentar a utilização da rede. Contudo, essa técnica também

é muito mais complexa, podendo consumir mais recursos da rede e levar a instabilidades de roteamento.

O roteamento baseado em restrições tem como objetivos principais: primeiro, encontrar rotas que atendam os requisitos de QoS e, em segundo lugar, distribuir a carga de tráfego da rede de forma igual ou balanceada. Assim, para encontrar tal rota, o algoritmo de roteamento baseado em restrições leva em conta não apenas o conhecimento da topologia da rede (aspecto relativo à alcançabilidade dos *hosts* de destino), mas também a disponibilidade de recursos nos enlaces (como banda, atraso, entre outros), além das exigências de QoS para fluxos específicos, e/ou questões administrativas que refletem uma política de uso dos recursos da rede, definida pelas próprias operadoras [IEC].

No MPLS, os LSP que são estabelecidos a partir deste tipo de roteamento são chamados de CR-LSP (*Constraint Route* - LSP), onde as limitações ou restrições podem ser, entre outros, nós explícitos ou determinadas exigências de QoS, como por exemplo: que enlaces (*links*) e filas, ou mecanismos de escalonamento serão empregados para o fluxo.

c) Roteamento Baseado em QoS

O roteamento baseado em QoS é um caso particular do roteamento baseado em restrições, onde o processo de selecionar um caminho a ser usado pelos pacotes de um determinado fluxo, a partir de suas exigências de QoS, como banda e atraso [Apostolopoulos98]. Nesse contexto, estudos recentes têm evidenciado que o roteamento baseado em QoS pode prover um aumento da utilização da rede, quando comparados ao roteamento que não é sensível às exigências de QoS do tráfego.

Várias soluções têm sido propostas para esse tipo de roteamento. Dentre elas, pode-se citar a apresentada em [Valenzuela02], que usa uma colônia de agentes estáticos para criar uma rede de conhecimento baseada em QoS, criada e residente em cada nó da rede, com o propósito de monitorar os recursos de rede disponíveis em um comutador DiffServ, para uso posterior na atualização da QoS-KN (*Quality-of-Service with Knowledge Network*). Ainda nessa proposta, uma segunda colônia, de agentes móveis é empregada para a descoberta real de rotas que satisfaçam a QoS exigida.

4.3.4. Questões Relacionadas à Reserva de Recursos

Para assegurar os serviços contratados não é suficiente simplesmente selecionar uma rota que possa prover os recursos adequados. É necessário, também, que estes recursos sejam efetivamente reservados, a fim de garantir que eles não serão compartilhados ou usados indevidamente por outro LSP. Estas exigências de tráfego devem ser enviadas aos LSR durante o estabelecimento do LSP, e são usadas em cada LSR para reservar os recursos requeridos ou recusar o estabelecimento do LSP, caso os recursos não estejam disponíveis.

Ao buscar este objetivo, o processo de TE melhora a utilização total da rede, pois tenta criar uma distribuição uniforme ou diferenciada do tráfego através da rede [IEC01]. Desta forma, um resultado importante deste processo é a prevenção de congestionamento em qualquer caminho da rede.

4.4. Engenharia de Tráfego no MPLS

O paradigma de encaminhamento de pacotes proposto pela arquitetura MPLS possibilita a inclusão de uma série de recursos para atender a questões relacionadas à diferenciação de serviços, dentre os quais a TE é uma das mais significativos [Awduche99]. É importante destacar que a TE tem aplicabilidade em qualquer rede chaveada por rótulo, onde haja pelo menos dois caminhos entre dois nós.

Os objetivos de desempenho associados à TE podem ser classificados em:

a) Objetivos Orientados ao Tráfego

São os objetivos que incluem os aspectos relacionados à melhora dos fluxos de tráfego com QoS, tais como: minimização da perda de pacotes, minimização do atraso, maximização da vazão e refinamento do SLA.

b) Objetivos Orientados aos Recursos

Dizem respeito aos objetivos que incluem os aspectos relativos à otimização da utilização dos recursos de rede, através do gerenciamento eficiente desses recursos. Pode-se citar, como exemplo, o caso da banda passante. Uma vez que a banda é um recurso essencial nas redes atuais, é desejável que se garanta que algumas partes da rede não sejam super-utilizadas e fiquem congestionadas, enquanto outras partes permaneçam sub-utilizadas.

Vale ressaltar, que a minimização do congestionamento é um objetivo de desempenho primário, tanto para o tráfego quanto para os recursos de rede.

Visando alcançar o objetivo principal da TE, o MPLS-TE deve, além do roteamento explícito, apresentar as seguintes características [Mplsrc01]:

- Capacidade para calcular na fonte um caminho levando em conta todas as restrições. Para fazer isto, a fonte deve possuir toda a informação necessária, seja ela disponível localmente ou obtida através dos roteadores da rede.
- Capacidade para distribuir, através da rede, a informação sobre a topologia da rede e os atributos associados aos enlaces. Uma vez que o caminho foi computado, o MPLS precisa de um modo para prover o encaminhamento das ligações FEC-Rótulo ao longo do referido caminho.
- Capacidade para reservar recursos da rede, bem como modificar os atributos do enlace, sempre que novos fluxos de dados adentrem a rede, fazendo uso de certas rotas e consumindo recursos.

A TE provida pelo MPLS pode também incluir um processo de monitoração, onde a informação pode ser roteada através da rede, de acordo com o ponto de vista do gerenciamento global dos recursos disponíveis da rede, além das cargas de tráfego esperadas ou atuais [Extreme01].

Outra alternativa para se conseguir TE no MPLS, pode ser feita através da configuração explícita de múltiplos LSP, usando o protocolo RSVP-TE. Neste caso, níveis específicos de banda especificados através da configuração de parâmetros, como limites de banda mínima e de banda requisitada, podem então ser reservados através de LSP estabelecidos pelo RSVP-TE.

A TE fornecida pelo MPLS também pode ser usada para suportar funções de balanceamento de carga, através do provimento de roteamento de tráfego IP sobre múltiplos LSP configurados explicitamente e que são tratados em uma dada FEC. A habilidade do MPLS de definir de modo flexível LSP explícitos também pode ser usada, juntamente com os recursos disponibilizados pelo roteador e/ou *switch,* para dar suporte a objetivos de gerenciamento de QoS, pela associação de classes específicas de serviço aos LSP, com provisões de banda pré-determinada. As questões de escalonamento são opacas para o MPLS-TE, ou seja, cada roteador e *switch* as implementa à sua maneira. O MPLS só transporta os pedidos de reserva.

4.5. MPLS e DiffServ

A prática tem demonstrado que o Diffserv não pode existir como uma tecnologia única para suporte a QoS; ao contrário, ele deve trabalhar cooperativamente com uma tecnologia de encaminhamento de pacotes que suporte um esquema de agregação, que é inerente à sua natureza de operação, no que diz respeito ao gerenciamento dos fluxos de dados.

Já o MPLS, por sua vez, necessita de mecanismos para garantir QoS aos LSP que estão sendo contratados. Destarte, seria natural a cooperação das duas soluções.

Nesse cenário, considerando que os LSR não processam os cabeçalhos IP, é necessário mapear a informação de classificação no cabeçalho MPLS. Assim, existem dois mecanismos propostos para tal classificação num esquema MPLS-DiffServ [Fineberg02]:

- E-LSP (EXP-LSP): usa o campo EXP do cabeçalho MPLS para diferenciar as várias CoS, que pode ser mapeado diretamente do ToS ou DSCP (encaminhamento por seletor de classes).
- L-LSP (Label-LSP): neste caso, o próprio rótulo transporta a informação de CoS, e o mapeamento de QoS precisa ser mais refinado.

Vale ressaltar que ambos os tipos de mapeamento podem ser tratados pelo LER, baseado nas políticas de rede definidas pelo administrador do domínio MPLS.

4.6. Comparativo entre o CR-LDP e o RSVP-TE

O CR-LDP e o RSVP-TE são protocolos capazes de estabelecer LSP ponto-a-ponto, que atendem às exigências de QoS para o MPLS [Nortel01, Extreme01].

O CR-LDP é uma extensão do protocolo LDP, projetado para dar suporte ao estabelecimento de LSP baseados em restrições, tais como: rotas explícitas e parâmetros de tráfego.

O RSVP-TE, por sua vez, se refere a um conjunto de extensões ao RSVP, que definem procedimentos para a sinalização de requerimentos de QoS e reserva dos recursos necessários para prover o serviço requisitado.

A Tabela 4.1 apresenta um quadro comparativo com as principais características do CR-LDP e do RSVP-TE.

Em função da escolha do RSVP-TE, como protocolo de sinalização para a especificação do serviço SVC, conforme proposto neste trabalho, apresenta-se, na Seção 4.7, alguns detalhes gerais sobre seu funcionamento. A justificativa para essa escolha é apresentada no próximo Capítulo.

CARACTERÍSTICA	CR-LDP	RSVP-TE
Escopo de implementação	Implementado apenas em roteadores	Implementado em roteadores e estações, inclusive com versões para Sistemas Operacionais bem populares como Windows e Linux
Procedimento de sinalização	Estabelecimento e Encerramento de conexão explícitos	Estabelecimento explícito e encerramento implícito através do uso de temporizadores ou explícito através de mensagens *teardown*
Natureza do estado	Protocolo *Hard State*, portanto, não necessita de refrescamentos	Protocolo *Soft-State,* que necessita de refrescamentos periódicos
Distribuição de rótulos	Pode operar no modo DOD ou DOU	Provê distribuição de rótulos no modo DOD
Reserva de recursos	NÃO faz reserva de recursos	Pode fazer reserva de recursos de rede para os LSP
Descoberta de vizinhos	Provê mecanismos de descoberta de vizinhança (protocolo de descoberta de vizinhança) através de mensagens *hello*	A especificação RSVP-TE também prevê mensagens *hello* para descoberta de vizinhos
Tratamento de erro	Não necessita	Usa datagramas IP para transportar mensagens entre pares. Portanto, deve tratar a perda de informação
Uso	Pouco utilizado comercialmente	O RSVP-TE é mais empregado no mercado

Tab. 4.1 – Quadro Comparativo entre o CR-LDP e o RSVP-TE

4.7. RSVP-TE

O RSVP-TE, especificado em [Awduche01], é uma extensão do RSVP. Conforme descrito em [Braden97], o RSVP apresenta as seguintes características gerais:

- Reserva recursos em apenas uma direção, ou seja, opera no modo *simplex*;
- É orientado ao receptor, ou seja, é o receptor que inicia o processo de reserva de recursos para um determinado fluxo de dados de aplicação;

- Não transporta dados de aplicação. É um protocolo de controle Internet, operando no nível dos protocolos ICMP, IGMP ou protocolos de roteamento internos. O RSVP é um protocolo de sinalização, que opera antes da fase de transferência dos dados, reservando recursos da rede nos equipamentos responsáveis pelo encaminhamento futuro dos pacotes de dados.
- Não é um protocolo de roteamento, tendo sido projetado para operar com os protocolos de roteamento *unicast* e *multicast* atuais e futuros.
- Os protocolos de roteamento determinam por onde os pacotes devem seguir, o RSVP relaciona-se apenas com a QoS destes pacotes, que são encaminhados de acordo com a rota definida pelo processo de roteamento.

Além disso, ele diferencia-se de outros protocolos de sinalização pelo fato de poder ser usado não apenas por roteadores, como também pelas estações:

- Nas **estações** ele é usado para requerer qualidade de serviço específica da rede para um determinado fluxo de dados de aplicação.
- Nos **roteadores** é usado para entregar requisições de QoS a todos os nós ao longo do caminho de um fluxo e para estabelecer e manter estados, que garantam o provimento do serviço requisitado.

4.7.1. Operação Básica do RSVP

Anteriormente à operação do RSVP, seguindo procedimentos fora do escopo desse protocolo, um *host* receptor avisa a um *host* emissor que deseja receber dados com garantia de QoS. O RSVP no emissor então inicia sua operação, enviando para o receptor uma mensagem PATH conforme mostrado na Figura 4.1. O objetivo primeiro da transmissão dessa mensagem é identificar através de quais roteadores da rede é possível se construir um caminho entre o emissor e o receptor, uma vez que a reserva de recursos para a transmissão dos dados será feita em cada um dos roteadores identificados. A mensagem PATH também é utilizada pelo emissor para notificar ao receptor qual o perfil do tráfego a ser tranmitido, através de um objeto chamado Tspec, em termos de banda utilizada e outros parâmetros de desempenho.

Para encaminhamento da mensagem PATH através da rede, cada roteador que recebe essa mensagem, utiliza as informações contidas em sua tabela de encaminhamento nível 3 (construída por um protocolo de roteamento), para

determinar o próximo roteador (*next hop*) do caminho em direção ao *host* receptor. Ao transmitir a mensagem PATH para o próximo roteador, cada roteador armazena localmente informações de estado sobre o caminho que está sendo construído (*path state*).

Quando a mensagem PATH chega ao receptor, o RSVP nessa máquina monta então uma mensagem de reserva de recursos RESV, e a envia para o emissor, através do caminho construído, em sentido inverso (*upstream*). A mensagem RESV carrega, dentre outros parâmetros, um objeto que especifica a QoS a ser reservada, chamada de Rspec. Cada roteador que recebe uma mensagem RESV, verifica se dispõe de recursos suficientes para atender a nova reserva (controle de admissão) e, em caso positivo, mobiliza esses recursos, construindo localmente um estado de reserva de recursos (*reservation state*), e retransmite a mensagem RESV para o próximo roteador em direção ao emissor. Quando a mensagem RESV chega ao emissor, o RSVP conclui sua operação de sinalização e a aplicação no *host* emissor inicia a fase de transmissão de dados para o receptor.

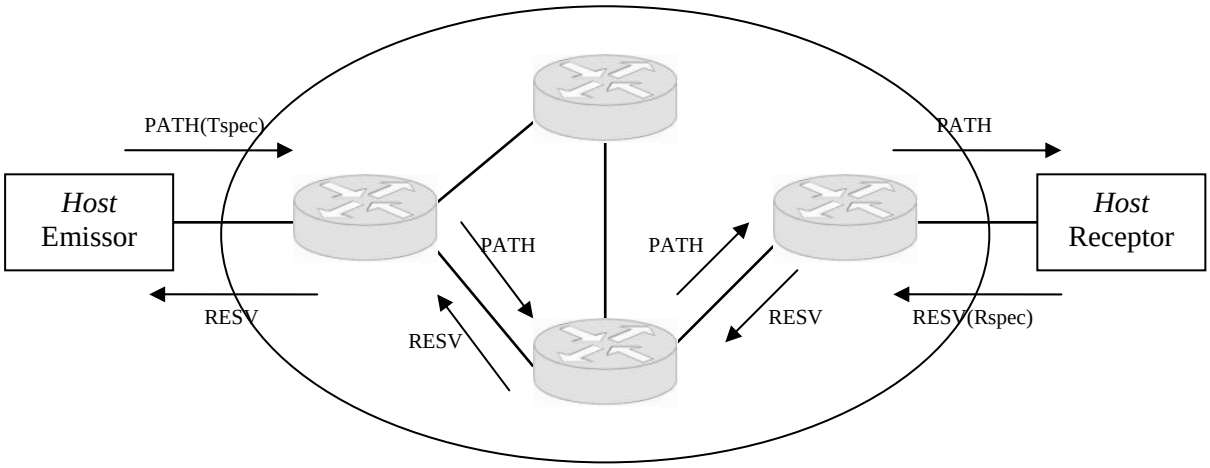

Fig. 4.1 – Operação Básica do RSVP

4.7.2. Mecanismos de QoS

A Figura 4.2 apresenta um esquema básico de inserção do RSVP, associando as funcionalidades disponíveis nos *hosts* e nos roteadores.

A QoS é implementada para um fluxo de dados específico, através de mecanismos coletivamente chamados de “Controle de Tráfego”. Estes mecanismos incluem:

- **Classificador de Pacotes**: determina a QoS para cada pacote;
- **Escalonador de Pacotes**: implementa a QoS desejada;
- **Controle de Admissão e de Policiamento**: usado na fase de sinalização, conforme explicado a seguir.

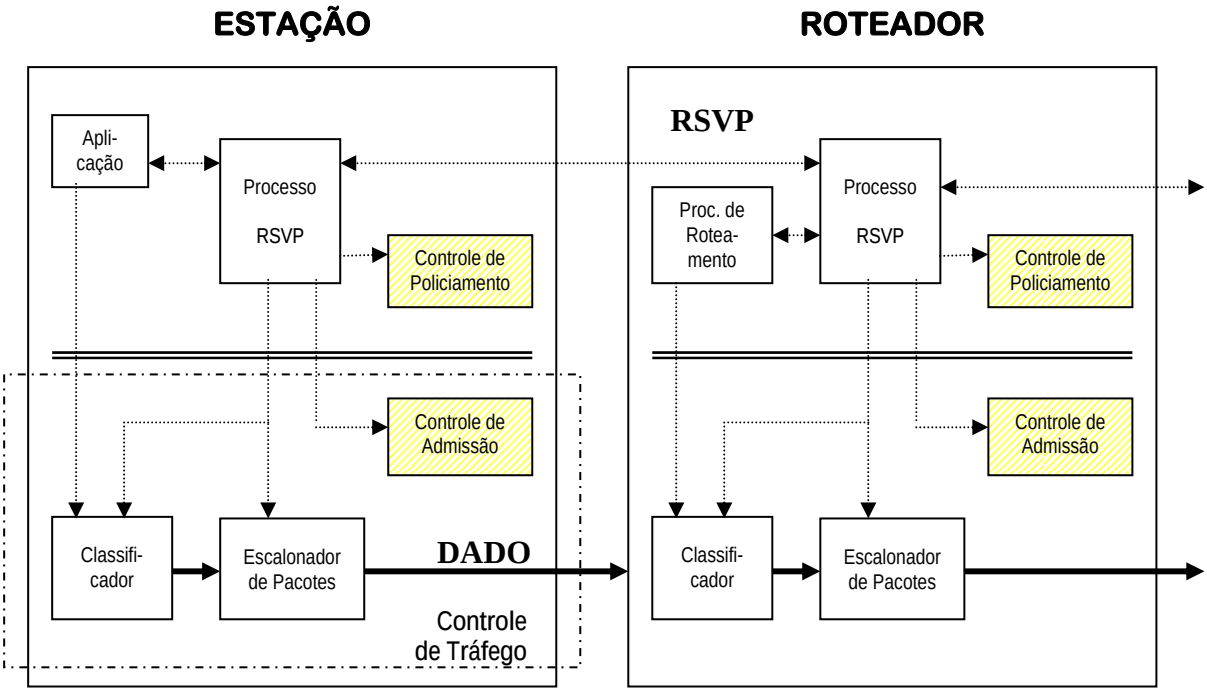

Fig. 4.2 – Esquema Funcional do RSVP nos Elementos da Rede

Durante o estabelecimento da reserva, uma mensagem RSVP de requisição de QoS é processada pelo processo RSVP local, que extrai dela determinados objetos e os repassa para dois módulos de decisão local:

- **Controle de Admissão**: determina se o nó tem recursos disponíveis suficientes para suprir a QoS requisitada.
- **Controle de Policiamento**: determina se o usuário tem permissão administrativa para fazer a reserva.

Se ambos os testes forem bem sucedidos, os parâmetros adequados são configurados (“setados”) no classificador e no escalonador de pacotes (ou outro mecanismo de manipulação da interface da camada de enlace) para obter a QoS

desejada. Caso um dos testes falhe, o elemento retorna uma notificação de erro para o processo de aplicação que originou o pedido (na estação receptora) e a reserva de recursos não é realizada.

Durante a transmissão de dados, o classificador determina a Classe de QoS para cada pacote e o escalonador ordena a transmissão de pacotes de forma a conseguir a QoS desejada para cada fluxo.

4.7.3. Especificação Funcional do RSVP

A seguir são apresentados o formato e os tipos das mensagens RSVP, bem como o formato e os tipos de objetos, tantos os especificados pelo RSVP quanto pelo RSVP-TE.

a) Formato das Mensagens

Uma mensagem RSVP consiste de um cabeçalho comum, seguido por uma relação de objetos, que podem variar quanto à quantidade de tipos de objetos e quanto ao tamanho de cada objeto. Na Figura 4.3 é apresentado o formato das mensagens RSVP.

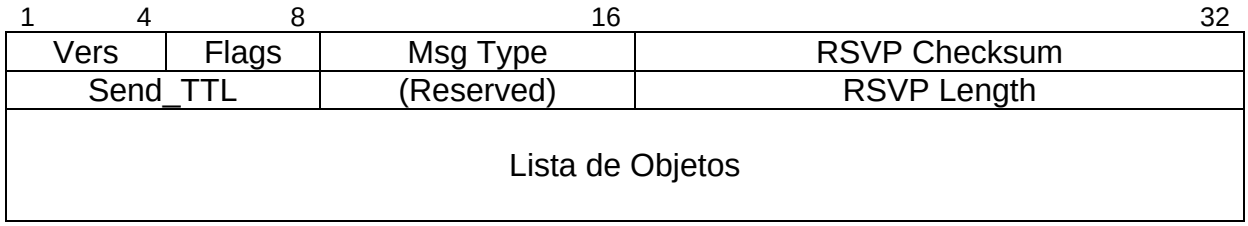

Fig. 4.3 – Formato da Mensagem RSVP

Os campos apresentam os seguintes significados:

- **Vers**: Um campo de tamanho 4 bits, que informa o número da versão do protocolo.
- **Flags**: Um campo de tamanho 4 bits, ainda não definido. Porém, com o valor $(0x01\text{-}0x08)_{16,}$ já reservado para uso futuro.
- **Msg Type**: Um campo de tamanho 8 bits, cujo valor indica o tipo da mensagem, conforme os seguintes valores: 1 = Path; 2 = Resv; 3 = PathErr; 4 = ResvErr; 5 = PathTear; 6 = ResvTear; e, 7 = ResvConf.
- **RSVP Checksum**: Um campo de tamanho 16 bits. É calculado com base no complemento de um da soma do complemento de um da mensagem com o

campo de *checksum* zerado. Se todos os bits forem zero significa que o *checksum* não foi transmitido.

- **Send_TTL**: Um campo de tamanho 8 bits, que é o valor do TTL IP com o qual a mensagem foi enviada.
- **RSVP Length**: Um campo de tamanho 16 bits, que representa o tamanho total da mensagem RSVP em bytes, incluindo o cabeçalho comum e os objetos de tamanho variável que seguem.

b) Tipos de Mensagens

Há dois tipos de mensagens fundamentais, conforme as categorias especificadas no RSVP, a saber: a mensagem PATH e a mensagem RESV.

Há, ainda, as mensagens *Teardown*, que removem o caminho ou o estado de reserva imediatamente. Vale lembrar que o RSVP especifica um mecanismo de refrescamento, de forma que, um determinado estado é apagado se nenhuma mensagem de refrescamento chegar antes do intervalo *cleanup timeout.* Contudo, embora não seja necessário, é recomendável usar as mensagens *Teardown* tão logo uma aplicação termine. Também, neste caso, existem dois tipos de mensagem *Teardown*: PATHTEAR e RESVTEAR.

Quanto às mensagens de erro, duas são especificadas: PATHERR e RESVERR.

Além das mensagens supracitadas, [Braden97] especifica ainda a mensagem RESVCONF, que é uma mensagem de confirmação de reserva para o receptor.

c) Formato dos Objetos

Cada objeto consiste de uma ou mais palavras de 32 bits com um cabeçalho de 1 palavra, conforme o formato apresentado na Figura 4.4, a seguir.

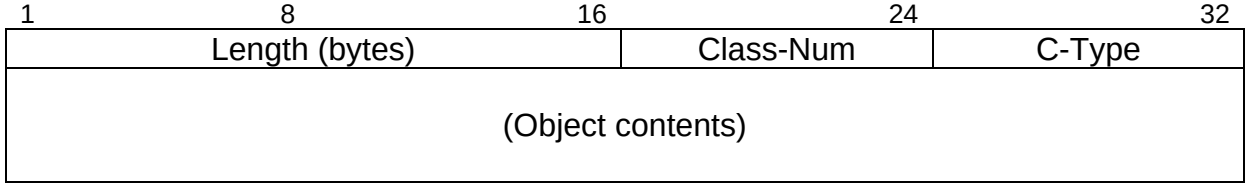

Fig. 4.4 – Formato dos Objetos RSVP

Os campos apresentam os seguintes significados:

- **Length**: Um campo de tamanho 16 bits que contém o tamanho total do objeto em bytes. Deve sempre ser um múltiplo de 4. Deve possuir, no mínimo, 4 bytes.
- **Class-Num:** Identifica a classe do objeto.
- **C-Type**: Indica o Tipo de objeto, seu valor é único dentro de uma Class-Num.

Ressalta-se ainda, que o tamanho de conteúdo máximo de um objeto é de 65528 bytes e, que os campos Class-Num e C-Type podem ser usados conjuntamente formando um número de 16-bits para definir um tipo único para cada objeto. Além disso, os dois bits de maior ordem do campo Class-Num são usados para determinar que ação um nó deve tomar, se ele não reconhecer o Class-Num de um objeto.

d) Tipos de Objetos

Cada classe de objeto pode ser identificada por um nome, os quais são listados na Tabela 4.2, que apresenta ainda uma breve descrição de cada classe.

A RFC 3209 [Awduche01] que especifica o RSVP-TE adiciona novos objetos que proporcionam novas funcionalidades ao RSVP, relacionadas à Engenharia de Tráfego, em ambientes MPLS.

Na referida RFC, extensões primárias adicionam suporte para associação de rótulos e especificação para rotas explícitas, tanto estritas quanto *loose*. Estas extensões são suportadas pela inclusão dos objetos <LABEL_REQUEST> e <EXPLICIT_ROUTE> na mensagem PATH. O destino responde a um <LABEL_REQUEST> pela inclusão de um objeto <LABEL> na sua mensagem RESV. Os rótulos são subseqüentemente associados em cada nó que a mensagem RESV atravessa.

O RSVP-TE inclui, ainda, extensões adicionais que provêem suporte para gravação de rota (<RECORD_ROUTE>), além de opções para detecção de *loop*, operações de re-roteamento, entre outras, através da combinação de diversos objetos.

Classe	Descrição
NULL	O objeto NULL tem uma Class-Num zero, e seu C-Type é ignorado. O seu tamanho deve ser pelo menos 4, mas pode ser um múltiplo de 4. Ele pode aparecer em qualquer posição em uma seqüência de objetos. Ademais, seu conteúdo será ignorado pelo receptor.
SESSION	O objeto SESSION contém o endereço IP do destinatário (DestAddress), o identificador do protocolo IP, e a porta destino, para definir uma sessão específica para os outros objetos que se seguem. É obrigatório em cada mensagem RSVP.
RSVP_HOP	Transporta o endereço IP do nó RSVP que enviou a mensagem e o LIH (*Logical outgoing interface*). Também referido como objeto PHOP (*Previous Hop*) para mensagens *downstream* ou como NHOP (*Next Hop*) para mensagens *upstream*.
TIME_VALUES	Contém o valor para o periodo de refrescamento R usado pelo criador da mensagem. É obrigatório em toda mensagem PATH e RESV.
STYLE	Define o estilo de reserva e a informação style-specific que não está nos objetos FLOWSPEC ou FILTER_SPEC. É requerido em toda mensagem RESV.
FLOWSPEC	Define a QoS desejada, em uma mensagem RESV.
FILTER_SPEC	Define um sub-conjunto de pacotes de dados da sessão que deverão receber a QoS desejada (especificada no objeto FLOWSPEC), em uma mensagem RESV.
SENDER_TEMPLATE	Contém o endereço IP do emissor e possivelmente alguma informação adicional de demultiplexação para identificar o emissor. É requerida em mensagens PATH.
SENDER_TSPEC	Define as características de tráfego de um emissor de fluxos de dados. Requerido em uma mensgam PATH.
ADSPEC	Um objeto que transporta dado OPWA (*One Pass With Advertising*), em uma mensagem PATH, que o receptor pode usar para prever o serviço fim-a-fim.
ERROR_SPEC	Especifica um erro em mensagens PATHERR, RESVERR, ou uma confirmação em uma mensagem RESVCONF.
POLICY_DATA	Transporta informação que permitirá que um módulo de policiamento local decida se uma reserva associada é administrativamente permitida. Pode aparecer em mensagens PATH, RESV, PATHERR ou RESVERR. O uso deste objeto não está completamente especificado.
INTEGRITY	Transporta dado criptografado para autenticar o nó origem e para verificar o conteúdo da mensagem RSVP. O uso do objeto INTEGRITY é descrito em [Baker96].
SCOPE	Transporta uma lista explicita de hosts emissores para os quais a informação na mensagem deve ser encaminhada. Pode aparecer nas mensagens RESV, RESVERR, ou RESVTEAR.
RESV_CONFIRM	Transporta o endereço IP do receptor que solicitou uma confirmação. Pode aparecer em uma mensagem RESV ou RESVCONF.

Tab. 4.2 – Descrição das Classes de Objetos do RSVP

As novas classes de objetos são apresentadas resumidamente na Tabela 4.3. Estes novos objetos são usados na operação do serviço SVC, proposto nesse trabalho. Mais detalhes sobre suas definições e mecanismos de operação no que tange ao serviço proposto são apresentados no Capítulo 6.

Classe	Descrição
LABEL_REQUEST	É uma classe obrigatória, presente na mensagem PATH.
LABEL	É uma classe obrigatória, presente na mensagem RESV.
EXPLICIT_ROUTE	É uma classe de uso opcional, presente na mensagem PATH.
RECORD_ROUTE	É uma classe de uso opcional, presente na mensagem PATH e RESV.
SESSION_ATTRIBUTE	É uma classe de uso opcional, presente na mensagem PATH.

Tab. 4.3 – Descrição das Classes de Objetos do RSVP-TE

Capítulo 5
A Linguagem Estelle e a Ferramenta EDT

Neste Capítulo são apresentados, a guisa de revisão, os conceitos básicos da linguagem de descrição formal Estelle, que foi escolhida como linguagem para a especificação do RSVP-SVC, e também as ferramentas que compõem o EDT (*Estelle Development Toolset*), programa usado para a especificação, compilação, validação e simulação do protocolo proposto.

Na Seção 5.1 é apresentada uma introdução às FDT (*Formal Description Techniques*) e suas ferramentas. Em seguida, na Seção 5.2 são apresentadas as principais características gerais de Estelle, com destaque para seus construtores básicos: os módulos e os canais. A natureza estruturada de Estelle também é abordada.

A Seção 5.3 versa sobre aspectos relacionados à verificação formal tais como: análise léxica, análise sintática, análise semântica e verificação formal das propriedades do sistema.

A descrição do EDT é apresentada na Seção 5.4, onde são descritas todas as ferramentas do programa: o editor gráfico, o tradutor Estelle, o gerador de código C, o compilador Estelle-C, o simulador/debugador, o *browser*, a *pretty printer*, o decompilador, o *splitter*, a ferramenta de visualização e o gerador universal de *test drivers*.

5.1. Introdução

O uso de FDT nos processos de especificação, verificação, análise, implementação, teste e operação dos sistemas distribuídos são necessários em função das crescentes exigências por qualidade. Tais sistemas podem apresentar características de: complexidade; concorrência; de qualidade crítica, onde a confiabilidade da informação é mais importante do que eventuais falhas no sistema, como é o caso das aplicações financeiras, dos protocolos de telecomunicações e dos sistemas operacionais; crítico a falhas, é o caso de sistemas de defesa, medicina, indústria nuclear; de segurança crítica, no qual o sigilo da informação é um requisito importante, como são exemplos sistemas de segurança nacional, privacidade; e, são padronizados [Turner93].

Nesse contexto, conceber especificações corretas em linguagem natural, que por natureza, são imprecisas, torna-se virtualmente impossível. Faz-se mister, portanto, o uso das FDT padronizadas que procuram garantir as seguintes qualidades:

- Especificações não ambíguas, claras e concisas;
- Completude das especificações;
- Consistência das especificações;
- Tratabilidade; e,
- Conformidade entre implementação e especificação.

Ademais, dada a natureza rigorosa das FDT torna-se possível o desenvolvimento de ferramentas que buscam auxiliar na criação, análise e refinamento de descrições formais. Nesse sentido, têm-se algumas opções, tais como: o EDT para linguagem Estelle, o SDT para a linguagem SDL, entre outras.

Isto posto, e considerando o fato de que se teve acesso ao programa EDT, através de uma versão *trial*, disponibilizada via Internet, foi escolhida Estelle como linguagem de especificação formal do protocolo RSVP-SVC. Ademais, as características de Estelle atendem perfeitamente às necessidades do protocolo proposto. As ferramentas disponibilizadas pelo EDT também atenderam as necessidades do trabalho.

5.2. Características Gerais de Estelle

A linguagem Estele foi desenvolvida originalmente para descrição de protocolos e serviços OSI. Ela consiste numa técnica para especificação de sistemas distribuídos, sendo adequada à descrição dos protocolos e serviços oferecidos pelas suas diversas camadas. Ademais, ela é baseada no fato de que os programas de comunicação são freqüentemente descritos e implementados através de um modelo de autômato finito [Turner93].

Uma especificação em Estelle é formada por um conjunto de módulos, implementados através de tipos, que interagem através de pontos de interação. Vale ressaltar que no decorrer de uma especificação podem ser criadas várias instâncias de módulos de um determinado tipo. A criação, destruição e a manipulação de instâncias de módulos são feitas através de primitivas [Fialho92].

Estelle é uma linguagem estruturada cujos construtores básicos são os módulos e os canais. Os módulos comunicam-se uns com os outros através de canais bi-direcionais e podem ser estruturados em sub-módulos.

5.2.1. Módulos Estelle

Os módulos Estelle agem através de transições de um estado para o outro. Assim, um autômato Estelle aceita E (entradas) e produz S (saídas) quando ele faz transições de um estado para o outro. Tais E e S são freqüentemente chamadas de interações e ocorrem através dos IP (*interaction point*).

Os módulos podem ser do tipo:

- Atividade (*activity*). Esse tipo de módulo escolhe um dos módulos-filho e dá a ele a chance de ser executado, de forma não determinística.
- Processo (*process*). Nesse caso, todos os filhos têm a chance de ser executados concomitantemente (em paralelo).

5.2.2. Transições Estelle

Em Estelle, a transição é uma ação atômica e, portanto, indivisível, que consiste em:

- Consumir entradas;
- Mudar um estado; e,
- Produzir uma saída.

As condições que habilitam o disparo de uma transição são:

- O estado de controle em que a máquina se encontra;
- A interação presente no topo de uma fila especificada;
- O valor de um predicado, que é baseado nos valores de variáveis ou parâmetros associados à interação;
- O estado de possíveis temporizadores associados à transição; e,
- O valor de prioridade associado à transição.

Ressalta-se que é possível a existência de transições que não requerem nenhuma E. Neste caso, elas são chamadas de transições espontâneas e, em geral, podem ter requisitos de tempo para disparar sua execução.

Vale destacar ainda que, caso haja um conjunto de transições habilitadas, a escolha de qual será disparada pode ser não-determinística, ficando a cargo do escalonador esta escolha randômica.

5.2.3. Canais Estelle

Como dito anteriormente, os módulos são conectados por canais, que formam ligações de uma saída de um módulo para uma fila associada a outro módulo. Vale ressaltar que não é permitida a ocorrência de interações arbitrárias.

Associados com cada canal existem dois papéis que serão assumidos pelos módulos conectados nas pontas de cada canal.

Os módulos mantêm filas para receber as interações, que pode ser:

- Uma fila individual para cada ponto de interação;
- Ou, uma fila comum, que combinas as entradas de todos os IP, sendo uma por módulo.

5.2.4. Estruturação

Os módulos podem ser estruturados para conter outros módulos. Assim, o Módulo pai pode criar ou destruir dinamicamente módulos filhos, que por sua vez, pode ter seus próprios filhos. A Figura 5.1 mostra um caso onde temos o Módulo A, que é pai dos módulos B e C, e o Módulo B, que é pai de D.

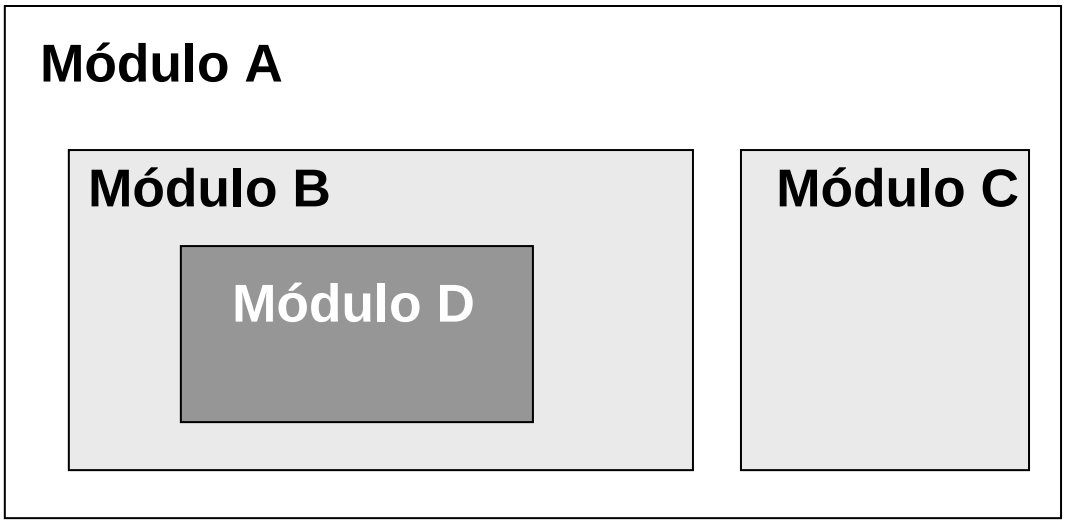

Fig. 5.1 – Estruturação dos Módulos Estelle

Os módulos podem ser conectados (*connect*) ou ligados (*attach*) uns aos outros através das seguintes premissas:

- O módulo pai pode conectar (*connect*) pontos de interação de seus filhos entre si, ou, aos seus pontos de interação internos, portanto, sem visibilidade externa;
- O módulo pai pode ligar (*attach*) seus pontos de interação externos (visíveis) aos pontos de seus filhos. Nesse caso, as interações recebidas são vistas pelo filho.

Ademais, seguem as seguintes características:

- O módulo pai pode partilhar variáveis com os filhos, desde que o filho as exporte;
- O módulo pai sempre executa antes dos filhos;
- A execução de módulos irmãos depende do atributo de classe do pai, assim, filhos de módulo *process* rodam em paralelo, e filhos de módulo *activity* rodam seqüencialmente em ordem randômica;
- Os módulos *activity* só podem conter filhos *activity,* e módulos *process* podem conter filhos *process* ou *activity.*

5.2.5. Primitivas Estelle

As primitivas Estelle permitem realizar:

a) A gestão de instâncias de módulos, através das primitivas:

- *Init*: Cria uma instância de módulo, associando um cabeçalho a um corpo de módulo. As instâncias de objetos associados ao módulo, bem como as instâncias de pontos de interação e variáveis também são criadas;
- *Release*: destrói uma instância de módulo.

b) A gestão de pontos de interação através das primitivas:

- *Connect* e *Disconnect*: conectam e desconectam instâncias de pontos de interação de mesmo nível hierárquico.
- *Attach* e *Dettach*: associam e desassociam instâncias de pontos de interação de níveis hierárquicos adjacentes.

c) O sincronismo entre instâncias de módulos, através das primitivas:

- *Output*: envia uma interação a um ponto de interação;
- *When*: recebe uma interação presente em uma fila de entrada.

5.3. Verificação Formal

O desenvolvimento de uma especificação Estelle envolve vários estágios conforme apresentado na Figura 5.2. Isto posto, uma vez concebida determinada especificação, através de uma descrição informal, faz-se necessária a sua formalização através da edição de uma descrição formal. Em seguida, tendo a descrição formal como código de entrada, procede-se à sua compilação (tradução e geração de Código C).

A fase de análise da compilação é composta por três subfases: análise léxica, análise sintática e análise semântica. Essa fase verifica a existência ou não de erros léxicos, sintáticos e/ou semânticos.

Uma vez que a especificação seja compilada corretamente, isto é, sem apresentar erros, pode-se passar a fase de verificação de seu comportamento, através de simulação ou testes de implementação. A verificação de propriedades de sistemas distribuídos objetiva, em geral, demonstrar que estes sistemas irão funcionar adequadamente após sua implementação, apresentando um comportamento esperado e isento de erros [Fialho92].

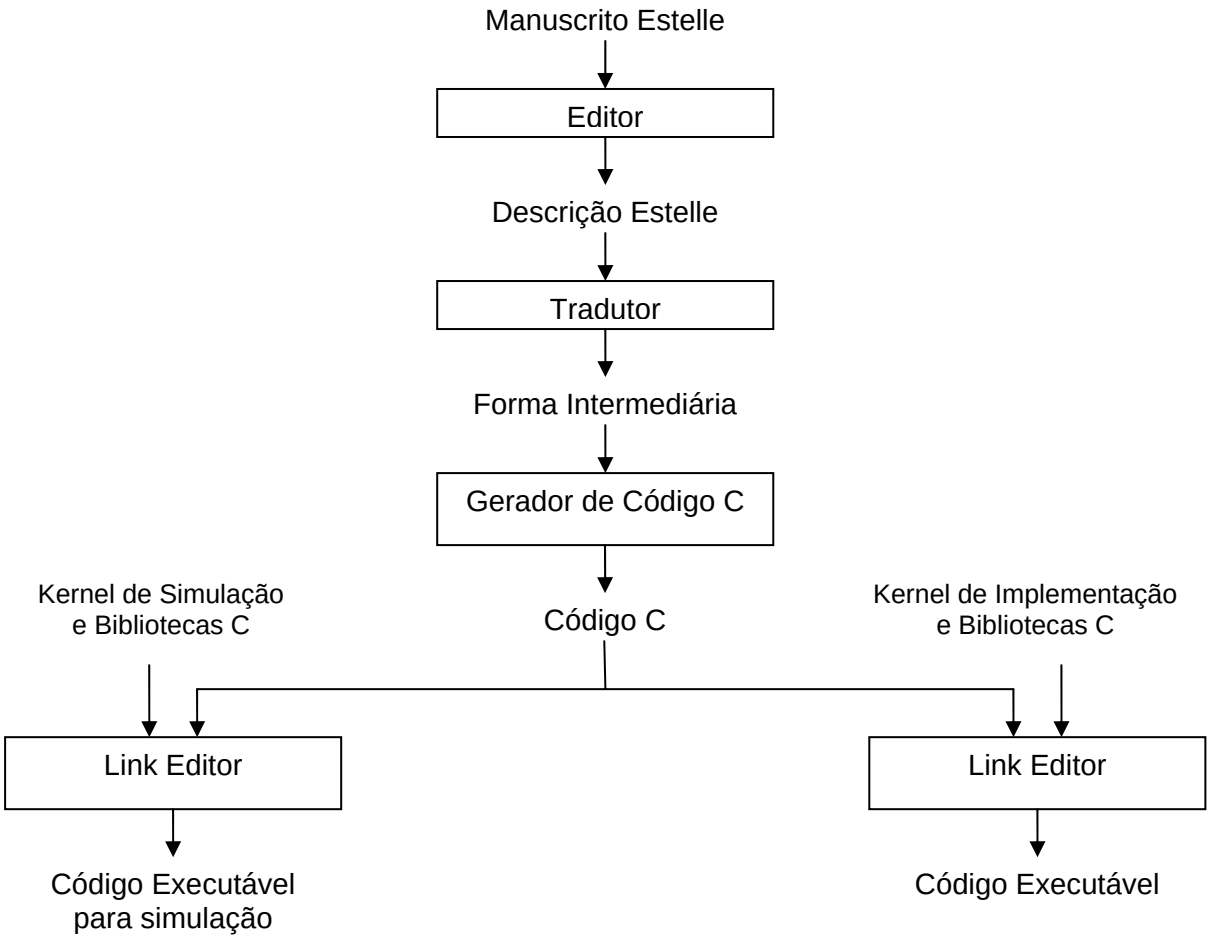

Fig. 5.2 – Estágios de Desenvolvimento em Estelle

5.3.1. Análise Léxica

Nessa fase o manuscrito Estelle é lido caractere a caractere. Símbolos especiais (espaço em branco, símbolos de pontuação e NL – *New Line*) são

utilizados para estabelecer os limites das palavras e os espaços em branco e comentários são eliminados.

Durante a análise léxica, as palavras ou lexemas são guardados na tabela de símbolos e classificados de acordo com a linguagem, em palavras reservadas, comandos, variáveis e tipos básicos. Identificação de erros léxicos que inclui, por exemplo, a identificação da ocorrência de *overflow* de campo numérico e uso de símbolos não pertencentes ao alfabeto da linguagem. Vale salientar que os itens léxicos a serem reconhecidos pelo analisador léxico são determinados pela gramática da linguagem-fonte. Deste modo, caso um item léxico não seja definido por esta gramática, um erro léxico é gerado.

5.3.2. Análise Sintática

É a parte mais importante de um compilador, uma vez que é nessa fase que se verifica se as frases estão escritas corretamente (se as palavras, ou *tokens*, estão na ordem correta). O analisador sintático recebe como entrada a seqüência de *tokens* extraídas do código-fonte, que foi produzido como saída pelo analisador léxico, e analisa a seqüência dessas palavras de acordo com a gramática na qual se baseia o analisador.

O analisador sintático é, portanto, responsável pela verificação da boa formação dos comandos da linguagem, de acordo com as regras especificadas pela gramática da linguagem. Sentenças mal formadas, geralmente, interrompem o processo de compilação e são apresentadas como mensagens de erro. No fim da análise sintática, temos a representação do programa original de forma hierárquica, onde o programa é representado por uma árvore sintática. O responsável pelo agrupamento dos símbolos em unidades sintáticas é o *Parser*.

Vale salientar que os erros de sintaxe são mais freqüentes que os erros léxicos.

5.3.3. Análise Semântica

A análise semântica mais comum consiste na verificação da consistência de tipos dos operandos envolvidos em operações aritméticas ou dos parâmetros passados a procedimentos. O código intermediário deve ser fácil de produzir e fácil de traduzir no programa objeto. Algumas das ações envolvidas na análise semântica são [Neto87]:

- Analisar restrições quanto à utilização dos identificadores: Em função do contexto em que são empregados, os identificadores devem ou não exibir determinados atributos. Cabe ao compilador, através das ações semânticas, efetuar a verificação da coerência de utilização de cada identificador em cada uma das situações em que é encontrado no código-fonte.
- Verificar o escopo dos identificadores: Mediante consulta à informação do escopo em que um identificador está sendo referenciado, o compilador deve executar procedimentos capazes de garantir que todos os identificadores utilizados no código-fonte correspondam a objetos definidos nos pontos dos programas em que seus identificadores ocorreram.
- Identificar declarações contextuais: Algumas linguagens permitem, para alguns tipos de objetos, que a sua declaração seja feita de modo implícito, e não através de construções sintáticas específicas. É outra função das ações semânticas do compilador localizar tais identificadores em seu contexto sintático, e associar-lhes atributos compatíveis com tal contexto.
- Verificar a compatibilidade de tipos: Cabe às ações semânticas efetuar a verificação do uso coerente dos objetos, que representam os dados do programa, nos diversos comandos de que o programa é composto. O mecanismo de passagem por parâmetro também é verificado através dessas ações semânticas.
- Efetuar a tradução do programa: A principal função das ações semânticas é exatamente a de criar, a partir do código-fonte, com base nas informações tabeladas e nas saídas dos outros analisadores, uma interpretação deste código-fonte, expresso em alguma notação adequada. Esta notação não se refere obrigatoriamente a alguma linguagem de máquina, sendo em geral representada por uma linguagem de forma intermediária do compilador.

5.3.4. Verificação das Propriedades de uma Especificação

A verificação de propriedades de sistemas distribuídos objetivo, em geral, demonstrar que estes sistemas irão funcionar adequadamente após sua implementação, apresentando um comportamento esperado e isento de erro. Neste trabalho, as propriedades são verificadas a partir do simulador/debugador da ferramenta EDT.

Segundo o sistema de classificação proposto por Manna [Manna81], as propriedades podem ser separadas em três grupos: propriedades invariantes, propriedades de eventualidade e propriedades de precedência.

a) Propriedades de Invariância

As propriedades desta classe garantem, em geral, que “nada de incorreto ocorrerá durante a execução do sistema” [Lamport77]. Dentre as propriedades mais importantes desta categoria pode-se citar:

- **Correção Parcial**: esta propriedade só tem interesse para sistemas que contém um estado terminal; ou, no caso de sistemas que apresentem um comportamento cíclico, quando estes sistemas possuam um estado que caracterize o término de um ciclo. Se o estado inicial satisfaz uma dada pré-condição, então em todo estado acessível a partir dele, se este estado é um estado terminal, a relação que garante a correção é válida. A pré-condição restringe o conjunto de entradas para o qual se supõe que o funcionamento do sistema esteja correto.
- **Ausência de Bloqueios**: ocorre quando nenhum dos processos que constituem o sistema está habilitado, de forma que o sistema não pode mais evoluir.
- **Exclusão Mútua**: considerando-se dois processo P_1 e P_2 que estão sendo executados em paralelo, é assumido que cada processo contém uma seção crítica C_1 e C_2, que inclui alguma tarefa crítica à cooperação dos processos. Esta propriedade garante que os dois processos nunca executarão suas respectivas seções críticas de forma simultânea.
- **Bom comportamento**: pode-se garantir que, em qualquer estado, teremos uma situação “bem comportada” do sistema, ou seja, uma condição que expressa a execução adequada das ações envolvidas e a geração de nenhuma falha ou erro.
- **Invariantes globais ou locais**: certas propriedades específicas devem ser válidas em todos os estados do sistema, o que caracteriza uma invariância global da propriedade. Outras propriedades específicas são válidas somente num conjunto restrito de estados, sendo chamadas então de invariantes locais.

b) Propriedades de Eventualidade

Esta propriedade expressa que, se uma condição é inicialmente verdadeira então eventualmente uma outra condição também será verificada. Em

geral, estas propriedades garantem que “algo correto acontecerá durante o funcionamento do sistema” [Lamport77].

- **Acessibilidade**: esta propriedade é útil quando se deseja verificar a entrada de um processo numa seção crítica ou numa região de interesse de seu funcionamento.
- **Vivacidade ou ausência de *livelock***: esta propriedade garante que o funcionamento de todo processo particular deve evoluir.
- **Respondimento**: é geralmente de interesse verificar esta propriedade em processos que apresentam um funcionamento cíclico. Dado dois processos P_1 e P_2, se o processo P_1 fizer um pedido de recurso ao processo P_2 então o processo P_2 responde a P_1, alocando o recurso pedido.
- **Correção Total**: esta propriedade só pode ser verificada em sistemas que possuam estados terminais. Ela garante que um processo que satisfaça a pré-condição, com relação aos valores das variáveis de entrada, eventualmente atingirá o estado terminal, e neste estado a condição de correção será verdadeira.
- **Asserções Intermitentes**: esta propriedade que descreve uma noção importante para programas cíclicos, garante que se uma condição C acontece num estado S, então eventualmente uma condição C’ acontecerá num estado S’.

c) Propriedades de Precedência

Esta classe de propriedades normalmente garante que “nada acontecerá de incorreto durante o funcionamento do sistema, até que algo correto aconteça” [Lamport77].

- **Vivacidade Segura**: a vivacidade segura garante que uma propriedade invariante se verifica até que se atinja uma propriedade de eventualidade. Por exemplo, garante-se que não existem bloqueios (ausência de bloqueios) até que se atinja um determinado estado (acessibilidade).
- **Ausência de Resposta Não Solicitada**: um processo não responde a menos que seja explicitamente solicitado.
- **Respondimento Justo**: esta propriedade garante que a precedência de dois eventos ocorre só quando dois eventos anteriores ocorreram na mesma ordem.

O respondimento justo estabelece uma disciplina de “servir primeiro que chegou primeiro”.

5.4. A Ferramenta EDT

O programa utilizado, a saber, o EDT (*Estelle Development Toolset*), encontra-se, atualmente, na sua versão 4.3 e, consiste em um conjunto de ferramentas para especificação e análise de sistemas de comunicação complexos usando a técnica de descrição formal Estelle.

O referido programa disponibiliza as ferramentas tanto para uso individualizado quanto para uso integrado através de uma interface *X-windows*, chamada xEdt. O xEdt faz a integração das seguintes ferramentas [Edt00]:

- Tradutor Estelle;
- Gerador de Código C;
- Compilador Estelle- C (Ec), que integra o tradutor Estelle e o gerador de código C;
- Simulador/Debugador Estelle com gerador de *trace* (Edb);
- *Browser*;
- *Pretty Printer*;
- Decompilador;
- *Splitter* (Gerador de Especificação Distribuída);
- Gerador Universal, que compreende o gerador de *test drivers*, o decompilador e o *Spliter*;
- Ferramenta de Visualização.
- Editor Gráfico.

Todas as ferramentas foram escritas em C, exceto o editor gráfico e o gerador de *trace*, que foram implementados em Tcl/Tk.

A seguir, descreve-se de forma sucinta as funcionalidades de cada ferramenta.

5.4.1. Tradutor Estelle (*Estelle Translator*)

O Tradutor Estelle gera, a partir de uma dada especificação Estelle, uma representação em forma intermediária, adicionada com informações resultantes de uma análise completa (léxica, sintática e semântica) da especificação.

5.4.2. Gerador de Código C (*Estelle Generator*)

O Gerador de Código C retorna o código C da Especificação, a partir de uma representação de uma especificação Estelle em sua forma intermediária.

5.4.3. Compilador Estelle-C (*Estelle to C Compiler*)

Traduz uma especificação escrita na linguagem Estelle em um código fonte em linguagem C. O compilador consiste na integração das duas ferramentas anteriores: o Tradutor e o Gerador de Código C. Ressalta-se que essas duas partes do compilador podem ser usadas separadamente ou de forma encadeada uma após a outra para produzir o código C diretamente do código fonte em Estelle.

O termo compilador, portanto, nesse contexto, refere-se ao uso encadeado das duas ferramentas.

O código gerado pode ser executado sob o controle do Simulador/Debugador ou de um motor de implementação, que serve de interface para um dado sistema operacional. O motor de implementação é um conjunto de rotinas de suporte em tempo de execução pré-compiladas, independente da especificação, a ser ligada (*linked*) ao código gerado para produzir um programa executável.

5.4.4. Simulador/Debugador Estelle (Estelle *Simulator/Debugger* – Edb)

O Edb simula a execução de uma especificação Estelle, de acordo com o modelo semântico definido pelo padrão ISO 9074. O propósito de um Simulador/Debugador é auxiliar o usuário na descoberta e correção de erros que ocorrem durante a execução de uma especificação.

O Simulador/Debugador é uma ferramenta interessante, pois dá ao usuário a capacidade de poder controlar, observar e seguir a execução da Especificação através de comandos de simulação. O Edb é um simulador/debugador simbólico interativo cuja entrada consiste do código fonte da especificação Estelle, de sua forma intermediária e do resultado da compilação do código C gerado.

De acordo com o princípio da atomicidade da transição do modelo semântico de Estelle, a unidade de execução no Edb é uma simples transição.

Vale ressaltar, que todo tipo de não-determinismo inerente a uma descrição Estelle são resolvidos, durante a simulação, através de uma seleção randômica, se tratados sem qualquer intervenção do usuário. Comandos especiais

são oferecidos pelo simulador/debugador para controlar algumas destas escolhas não determinísticas de um modo interativo.

O Edb ofecere ainda meios eficientes, como comandos de macro e observadores, para descrever um cenário de simulação, incluindo a especificação de anomalias a serem detectadas.

Tal ferramenta dá ao usuário a capacidade de poder se concentrar completamente nas propriedades que ele deseja verificar ou detectar.

O Edb fornece ainda a capacidade de, durante uma simulação, o usuário poder requisitar de forma *on-line* a geração de múltiplas visões MSC (*Message Sequence Chart*), individualmente parametrizadas.

5.4.5. *Browser*

É uma ferramenta interativa que gera visões de alto nível chamadas de Visões HLD (*High Level Design*) da especificação, representadas em sua forma intermediária. Tais visões podem ser mostradas na tela e/ou gravadas em arquivo.

A Visão HLD consiste em uma descrição da hierarquia de instâncias dos módulos e dos canais de comunicação entre as instâncias dos módulos (pontos de interação), omitindo-se a descrição de seu comportamento. Ela corresponde, portanto, a arquitetura da especificação, em termos de instâncias dos módulos, até uma certa profundidade definida pelo usuário.

5.4.6. *Pretty Printer*

A ferramenta *Pretty Printer* é usada para dar um formato diferente a especificação, em termos de: tamanho das linhas, tamanho das tabulações, identificadores e palavras-chaves em maiúsculas, minúsculas, entre outros. Permitindo, assim, que o usuário possa aplicar padrões de formatação de forma automática em toda a especificação.

5.4.7. Decompilador (*Decompiler*)

O Decompilador permite gerar, a partir da forma intermediária, a especificação Estelle correspondente.

5.4.8. *Spliter* (*Distributed Specification Generator*)

O *Spliter* permite que o usuário gere, a partir de uma especificação Estelle que contenha “n” sub-sistemas (módulos do tipo “*systemprocess*” ou “*systemactivity*”):

- “n” especificações Estelle, cada uma contendo um sub-sistema e uma inteface especial chamada de *switch board*, ao mundo exteno;
- Um programa supervisor em C.

5.4.9. Gerador Universal (*Universal Generator* – Ug)

Assim como ocorre no Compilador Estelle-C, o Gerador Universal é uma ferramenta que integra outras duas ferramentas, a saber: o *Splitter* (Seção 5.4.5) e o Decompilador (Seção 5.4.6). Cada função do Ug, portanto, pode ser vista como uma ferramenta separada.

O Gerador Universal permite ao usuário:

- Gerar uma especificação modificada, a partir de uma especificação fonte em Estelle, na qual alguns corpos de módulos (*body*) selecionados pelo usuário serão substituídos por corpos universais (*Universal Test Drivers Generator* – Utdg). Esta ferranebta permite a geração de *tests drivers* interativos para qualquer especificação Estelle aberta.
- Gerar especificações menores e mais simples de depurar (*Splitter*);
- Gerar a especificação fonte em Estelle a partir da forma intermediária (Decompilador).

5.4.10. Gerador de Tabela Estado-Evento (*Estelle State/Event Tables Generator* – Edoc)

Essa ferramenta permite a geração automática de tabelas de estado-evento correspondente a um corpo especificado pelo usuário dentro de uma especificação. Estas tabelas são geradas a partir da informação encontrada no arquivo em forma intermediária, que foi gerado pelo Tradutor Estelle a partir de uma especificação Estelle, onde, cada coluna corresponde a um estado *FROM*, um *STATESET* ou *STATELIST* e, cada linha corresponde a um evento que é uma condição de disparo de uma transição. Em cada célula, encontro da coluna e linha, é mostrado o nome da transição, o estado *TO* e as saídas.

5.4.11. Editor Gráfico (*Graphical Editor*)

O editor Estelle/GR é uma interface gráfica baseada em janela para definição, edição e visualização de uma especificação Estelle numa sintaxe gráfica.

O editor provê um ambiente estruturado que separa a estrutura Estelle das definições de comportamento. Ele foi concebido para complementar o EDT. Suporta a importação de especificações textuais que tenham sido compiladas pelo Ec (gerador de código intermediário do EDT) e pode gerar especificações textuais que podem ser usadas como entrada para o EDT.

Capítulo 6

Especificação de uma Rede MPLS Fim-a-Fim com Diferenciação de Serviços

Este capítulo apresenta a proposta central desse trabalho: a especificação de um serviço discado para uma UNI MPLS que pode ser levada até o *host* do usuário, possibilitando a implantação de uma Rede MPLS Total com Diferenciação de Serviços. Assim, o presente capítulo está organizado da seguinte forma: após uma breve introdução que visa dar uma visão geral da proposta, segue-se algumas considerações preliminares na Seção 6.2, que aborda aspectos como: metodologia adotada, terminologia, especificação funcional e classes de serviço. Na Seção 6.3 é apresentada de forma detalhada a proposta de especificação do serviço SVC: as opções de projeto, as funcionalidades propostas, bem como todos os procedimentos envolvidos. O MPLSoLAN é abordado na Seção 6.4, destacando-se os componentes da arquitetura e suas funcionalidades, as vantagens já identificadas da adoção da proposta e o seu *modus operandi*. Finalmente, na Seção 6.5 são descritos aspectos relacionados à compatibilidade com o RSVP-TE e na Seção 6.6 os aspectos co-relacionados ao MPLS Fim-a-Fim, a saber: aspectos relativos ao gerenciamento dos recursos da rede e os relativos à alocação de banda.

6.1. Introdução

É amplamente aceito o fato de que o MPLS provê uma infra-estrutura de rede sólida, ainda que mantendo a simplicidade, para serviços que requerem alta confiabilidade com excelentes mecanismos de QoS, como os apresentados na Seção 2.3. Entretanto, ainda não há um acordo completamente definido para prover serviços discados MPLS nativos via uma UNI MPLS [NwFusion03b].

Assim, como dito no Capítulo 1, o objetivo desse trabalho é estudar as questões envolvidas na definição dos elementos funcionais básicos da UNI MPLS, assim como os requisitos para provê-los: as mensagens de protocolo, seus conteúdos de informação e os procedimentos necessários para uma aplicação acessar a UNI. A definição desses aspectos resulta numa proposta de especificação do serviço SVC (*Switched Virtual Connection*), que dá a capacidade necessária aos dispositivos finais da rede para requisitar LSP com parâmetros de QoS específicos.

Uma diferença marcante entre a proposta do MPLS FORUM e a proposta aqui apresentada é que, nessa última a UNI MPLS SVC é levada até os *hosts* da rede local do usuário e não ao AR, apenas. Com isso, pretende-se não apenas simplificar a alocação de banda nas redes MPLS e incrementar o suporte a chamadas de voz, como aliviar a carga de processamento no AR, conforme mostrado na Figura 6.1, que fornece uma visão geral do MPLS Fim-a-Fim: Serviço SVC e MPLSoLAN.

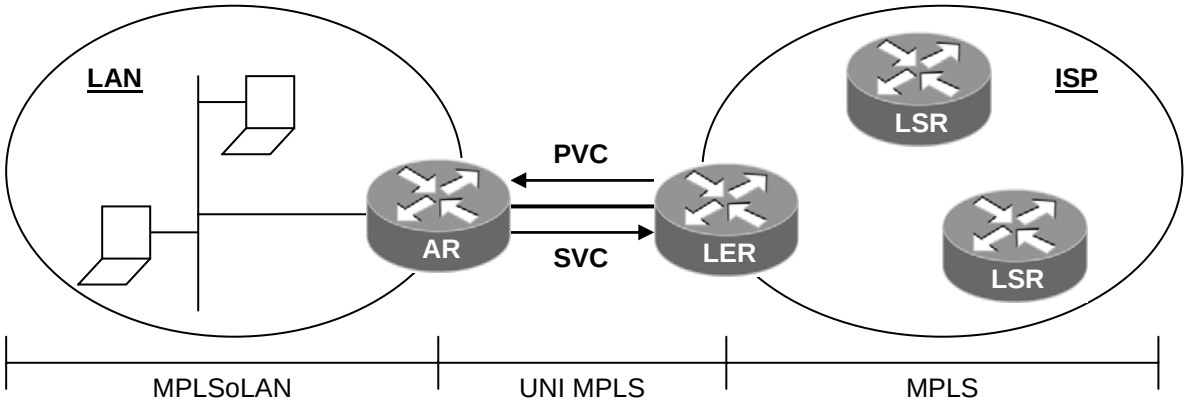

Fig. 6.1 – Escopo das Especificações do MPLS Fim-a-Fim

Prover bons níveis de QoS nesta porção da rede, onde o controle é menos confiável é um desafio e uma exigência indispensável para serviços como VoIP e vídeo-conferência, pois as redes de acesso permanecem com um significativo gargalo potencial [NwFusion03a].

6.2. Considerações Preliminares

Antes de apresentar o Serviço SVC e o MPLSoLAN faz-se necessário tecer algumas considerações preliminares referente à proposta como um todo.

a) Metodologia Adotada

A metodologia usada para a especificação detalhada do Serviço SVC seguiu o mesmo procedimento adotado pela Aliança FR/MPLS no que diz respeito ao Acordo para o Serviço PVC. Nesse sentido, a especificação foi detalhada a partir do protocolo RSVP-TE [Awduche01], destacando o que permanece, o que é desnecessário e o que é alterado, em termos de procedimentos, mensagens e objetos (formato e conteúdo). Adicionalmente, máquinas de estado são

especificadas com o objetivo de possibilitar uma verificação formal do funcionamento do protocolo, no que concerne à validade das trocas de mensagens e objetos.

O MPLSoLAN também é apresentado formalmente seguindo a mesma metodologia do Serviço SVC, a fim de manter a uniformidade do trabalho como um todo.

Vale ressaltar, ainda, que as seções seguintes deste capítulo adicionam comentários em função das modificações necessárias para o estabelecimento dos serviços propostos, ou quando se pretende destacar alguma vantagem ou restrição em relação ao RSVP-TE.

Isto posto, são apresentadas a partir da Seção 6.3.4 as modificações necessárias à especificação do RSVP-TE, visando o provimento do serviço SVC e do MPLSoLAN.

b) Terminologia

Adota-se a mesma terminologia apresentada em [Awduche01]. No glossário é fornecida uma lista dos acrônimos usados nesse trabalho.

c) Especificação Funcional

Quanto aos tipos e formatos das mensagens, nenhuma alteração é necessária em relação à especificação base do RSVP e do RSVP-TE. Elas são suficientes para a extensão proposta neste trabalho, a saber: as mensagens básicas, PATH e RESV; as mensagens de desconexão (*teardown*), PATHTEAR e RESVTEAR; as mensagens de erro, PATHERR e RESVERR; além da mensagem RESVCONF especificada em [Braden97]. A Seção 6.7. apresenta mais informações sobre as referidas mensagens.

Todas as funcionalidades necessárias para a operação do serviço SVC podem ser feitas através do uso dos objetos já previstos em [Awduche01] e dos objetos introduzidos neste trabalho, que, como dito anteriormente, se propõe a oferecer uma extensão ao conjunto de funcionalidades já especificadas pelo RSVP-TE. Uma relação desses novos objetos propostos é apresentada na Seção 6.3.4, Tabela 6.2. Todos os objetos relacionados são formalmente apresentados no decorrer deste Capítulo.

d) Classes de Serviço

Para acomodar a transferência dos vários tipos de mídia e, por conseguinte, poder prestar os vários tipos de serviço, é indispensável uma definição clara das possíveis categorias de serviço. Em princípio seria possível manter as definições apresentadas em [Awduche01], contudo, optou-se, por adotar àquela preconizada pelo IEEE 802.1p e apresentada na Tabela 6.1, que é a indicada na definição conceitual do MPLSoLAN, conforme mostrado na Seção 6.4. De fato, há uma gama variada de possibilidades. Em função da flexibilidade provida pelo MPLS em termos de como associar pacotes aos LSP, cada operadora pode adotar a classificação que melhor lhe apraz.

Além disso, por questões de compatibilidade, é desejável prover suporte ao Serviço *Null* como preconizado em [Awduche01, Bernet00]. O serviço *Null* permite que as aplicações se identifiquem para os agentes de QoS da rede através da sinalização do RSVP além de não requerer que as aplicações especifiquem suas exigências de recursos. Os agentes de política de QoS da rede respondem aplicando políticas de QoS apropriadas para a aplicação como determinadas pelo administrador da rede.

Prioridade de Usuário	Sigla	Classe de Serviço	Descrição
0	BE	Melhor-esforço	Tráfego de dados sem garantias de QoS
1	BK	*Background*	Transferência de grandes volumes de dados e outras atividades que são permitidas na rede, mas que não deveriam impactar o uso da rede por outros usuários e aplicações
2	-	-	Reservado
3	EE	Esforço Excelente	Serviço um pouco melhor do que o melhor-esforço para usuários diferenciados
4	CL	Carga Controlada	Tráfego sujeito a controle de admissão
5	VI	Vídeo, < 100ms latência e *jitter*	Tráfego caracterizado por atraso menor que 100ms
6	VO	Voz, < 10ms latência e *jitter*	Tráfego caracterizado por atraso menor que 10ms
7	NC	Controle de rede	Tráfego destinado a manter e suportar a infraestrutura de rede

Tab. 6.1 – Classes de Serviços 802.1p

Assim, uma possível solução seria mapear esse serviço na classe 802.1p de prioridade 2, que está reservada para uso futuro, através da definição de uma CoS chamada de Sinalização de QoS (SQoS), por exemplo.

Essa escolha se justifica, uma vez que esse serviço deve, de fato, ter prioridade acima do tráfego de *background*, pois se destina a uma função de controle que deve preceder a transferência de dados de configuração de equipamentos e serviços, tipicamente transportados na classe 1, e abaixo dos serviços de transferência de dados com garantias mínimas de QoS já anteriormente estabelecidas.

6.3. Especificação do Serviço SVC

A definição de uma UNI MPLS para um Serviço SVC provê uma interface para conexão discada entre o AR (*Access Router*), e uma rede pública MPLS, como apresentado na Figura 6.2. Esta interface é importante, porque garante que uma infra-estrutura de rede baseada em MPLS possa suportar aplicações de desempenho crítico, incluindo voz e vídeo em tempo real, enquanto preserva o uso e a flexibilidade das tecnologias Internet atuais.

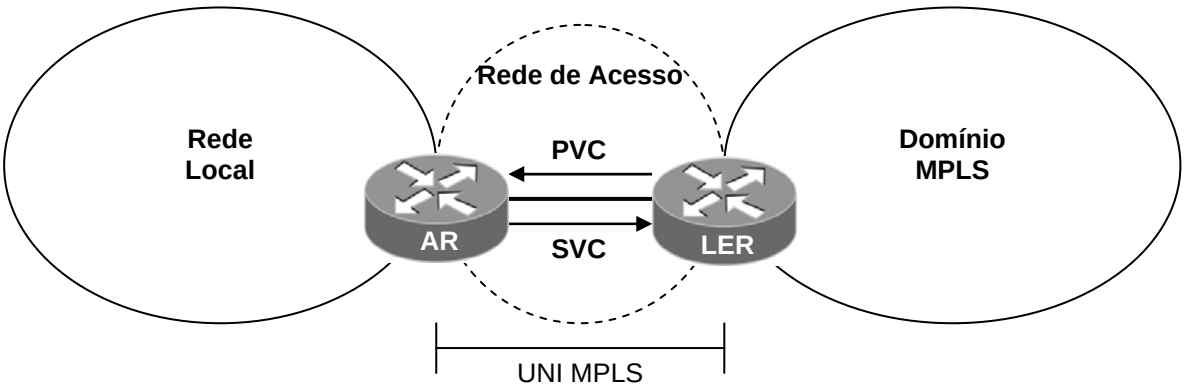

Fig. 6.2 – Interface Usuário-Rede

Nesse contexto, já existe um grupo de trabalho na Aliança MPLS/Frame Relay (MPLS/FR ALLIANCE) que discute o desenvolvimento da referida UNI, no intuito de prover uma QoS fim-a-fim (QoS-e2e) verdadeira sobre uma rede MPLS. Além disso, a UNI MPLS é particularmente importante porque possibilita trazer as ferramentas de gerenciamento de tráfego, atualmente presentes apenas no núcleo do *backbone*, para os pontos finais da rede [NwFusion03a]. O referido grupo já

desenvolveu a primeira versão da UNI MPLS [MPLSForum03c], que contempla apenas uma abordagem do serviço do tipo PVC (*Permanent Virtual Connection*), que é caracterizado pelo provisionamento iniciado pelo provedor de serviços e a operação passiva do AR. Ainda não foi especificado, contudo, o serviço SVC no qual os AR serão capazes de iniciar a sinalização de requisições para estabelecer ou encerrar LSP, bem como modificar parâmetros de tráfego.

6.3.1. Considerações sobre as Opções de Projeto

A proposta de serviço do tipo SVC aqui apresentada procurou atender às seguintes premissas: ser eficiente e também manter as características do MPLS de ser escalável, simples e funcional.

- **Eficiência**: Pretende-se melhorar o desempenho da rede, ao trazer para o roteador de acesso a função de realizar uma classificação mais extensiva do tráfego de entrada, de forma a aliviar a carga de processamento nos roteadores de borda da rede;
- **Escalabilidade**: Revela-se importante trazer para o roteador de acesso, os ganhos devidos à granularidade com que as FEC podem ser definidas no MPLS, permitindo assim a agregação de tráfego, e garantindo uma QoS fim-a-fim individual sem a necessidade de manter o *status* do fluxo em cada equipamento de rede do núcleo (*core*);
- **Simplicidade**: É desejável retomar o paradigma inicial da Internet, conhecido como Princípio Fim-a-Fim (*E2E Principle*) [Saltzer01], ou seja: complexidade no sistema final, simplicidade na rede;
- **Funcionalidade**: É de interesse possibilitar a requisição do estabelecimento de um LSP a partir da rede de acesso, com base em atributos como, por exemplo: endereço destino e exigências de QoS.

Nesse contexto, o principal aspecto relacionado à especificação do serviço SVC diz respeito à escolha do protocolo de sinalização a ser utilizado pelo serviço. Opcionalmente, para melhorar o desempenho total do serviço, um outro aspecto a ser considerado diz respeito à escolha do mecanismo de alocação de banda.

a) Quanto à Escolha do Protocolo de Sinalização

Essa escolha deve levar em conta não apenas os aspectos relacionados à distribuição dos rótulos, como também aspectos relacionados às necessidades de

provimento de QoS, de forma a reservar os recursos da rede de maneira adequada ao longo de um LSP. Nesse sentido, seria possível escolher entre opções, tais como: o CR-LDP (*Constraint Routing – Label Distribution Protocol*) [Andersson01] e o RSVP-TE (*Resource Reservation Protocol – Traffic Engineering*) [Awduche01]. O protocolo de sinalização adotado para o Serviço SVC é chamado de RSVP-SVC, e se constitui numa extensão proposta nesse trabalho ao protocolo RSVP-TE. A escolha do RSVP-TE como base para a extensão proposta foi feita em função das seguintes características:

- **Segurança** – O RSVP [Braden97] pode implementar **autenticação e controle de admissão e de policiamento explícitos**: as mensagens RESV podem carregar um objeto que contém informações sobre a permissão do receptor para reservar os recursos da rede, cujo conteúdo pode ainda ser protegido. Este dado pode ser usado quando do processamento de mensagens de reserva para executar controle de admissão de forma segura;
- **Protocolo *soft-state*** – Este modo de operação captura as mudanças de roteamento automaticamente e, além disso, torna-se um mecanismo eficiente contra perdas de mensagens de controle. O refrescamento ajuda a garantir, por exemplo, que o estado do LSP está devidamente sincronizado entre os nós vizinhos em caso de mudança do LSP. No caso de alteração das exigências de recursos (mudança nos parâmetros de tráfego) de um LSP ou re-roteamento, o RSVP também “responde” mais rápido, uma vez que todo nó mantém informação de estado. Um efeito colateral é a necessidade de mais memória para armazenar todas estas informações;
- **Criação de ER-LSP** (*Explicit Route - LSP*) – Esta característica provê os meios necessários para a implantação das seguintes capacidades: re-roteamento suave, preempção de LSP previamente estabelecidos em função da ativação de outros LSP com maior prioridade e detecção de *loop*;
- **Estabelecimento de Túneis LSP (LSP-Tunnel)** – Uma vantagem do uso do RSVP-TE consiste na sua capacidade de estabelecer Túneis LSP com alocação de recursos ao longo do caminho;
- **Difusão de Uso** – O RSVP-TE tem se tornado um padrão *de facto* do mercado;

- **Abrangência –** Sua execução não se restringe aos roteadores, podendo ser executado também nas estações, o que favorece o estabelecimento de uma política fim-a-fim.

b) **Quanto à Escolha do Mecanismo de Alocação de Banda**

Um outro aspecto importante relacionado ao serviço SVC refere-se à escolha do mecanismo de alocação de banda para cada conexão discada. Entretanto, esse aspecto não precisa necessariamente fazer parte da especificação da UNI desse serviço, ficando fora do escopo desse trabalho.

Mesmo assim, é possível se fazer alguns comentários quanto à escolha desse mecanismo. O IETF TE-WG (*Traffic Engineering – Work Group*) definiu até o presente dois modelos de alocação e restrição de banda, a saber: o *Maximum Allocation Model* (MAM) [LeFaucher03a] e o *Russian Dolls Model* (RDM) [LeFaucher03b]; contudo, vislumbra-se uma outra possibilidade, através da utilização de técnicas de Inteligência Artificial. Neste caso, uma rede neural poderia capturar a dinâmica do perfil do tráfego na rede e, a partir deste aprendizado, tomar decisões, de forma automatizada, sobre a alocação de recursos. Trabalhos futuros poderão desenvolver um estudo comparativo das soluções sugeridas acima, no sentido de indicar que mecanismo usar, sob quais circunstâncias e em que contextos, no que se refere ao Serviço SVC.

6.3.2. Novas Funcionalidades Propostas

O serviço SVC deve ser capaz de prover, no mínimo, as seguintes funções:

a) Estabelecer dinamicamente novos LSP, a partir do assinante, com requisição de QoS (banda passante, taxa de perda de pacote, atraso, *jitter*, entre outros) especificados pelo usuário, além de possibilitar a especificação de parâmetros de tráfego como: vazão média, vazão de pico, duração de pico, entre outros;
b) Ativar os LSP de acordo com um SLA (*Service Level Agreement*). Uma vez que o contrato *on-line* tenha sido assinado, presume-se, obviamente, que ambas as partes concordaram com os termos do SLA;
c) Suspender temporariamente determinado LSP, em função da necessidade de atender fluxos de maior prioridade, ou mesmo por questões administrativas;

d) Suportar modificações de QoS, ou seja, suportar sinalização que permita o AR ou o LER modificar dinamicamente os atributos (parâmetros de QoS) do LSP, desde que não descumpra o previsto no SLA; e,
e) Encerrar LSP ativos, tanto por parte do assinante quanto da rede.

6.3.3. Premissas Básicas

Antes de apresentar o *modus operandi* do Serviço SVC, na Seção 6.3.4, faz-se necessário a apresentação de algumas premissas.

a) Pares RSVP

Em uma UNI MPLS SVC, o AR e o LER atuam como pares RSVP simultaneamente. Quando na fase de sinalização, por exemplo, eles negociam o valor do rótulo seguindo o método DOD, conforme apresentado na Figura 6.3.

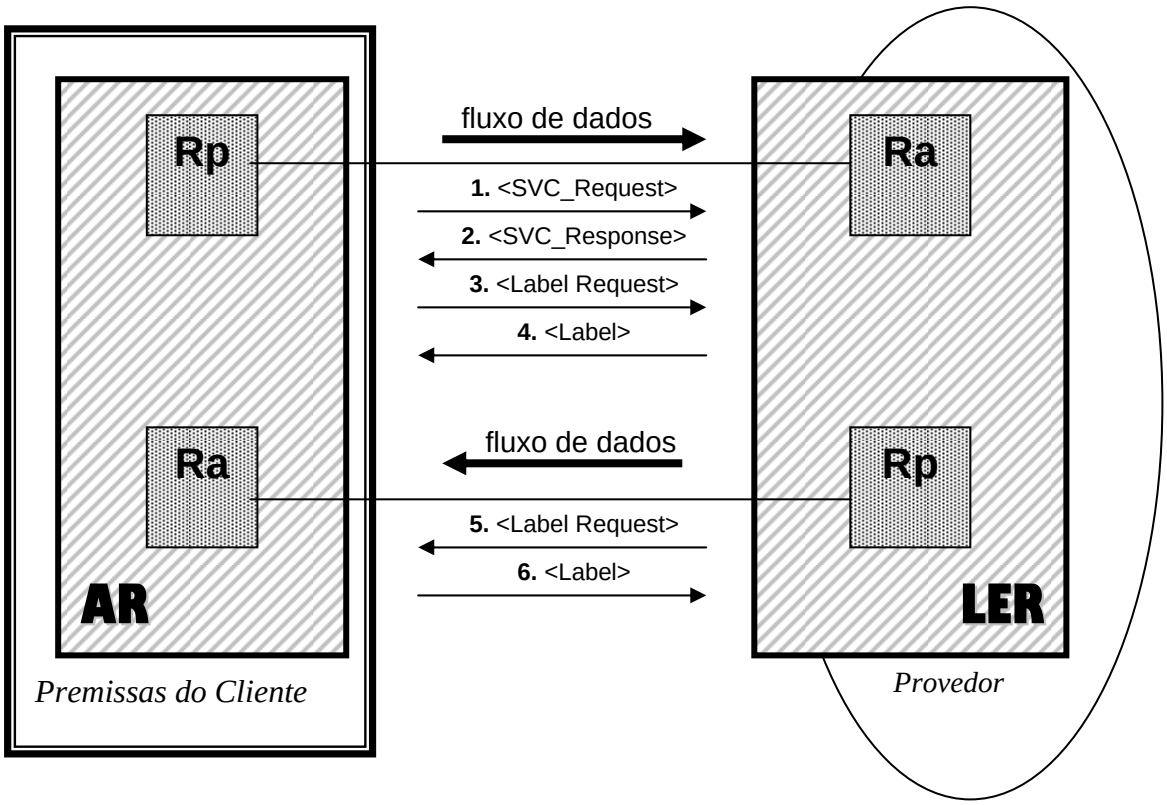

Fig. 6.3 – O AR Negocia a Configuração do Serviço SVC com o LER

b) Modo de Retenção do Rótulo

É usado o modo de retenção conservador cujas características são apresentadas na Seção 3.6.4., por ser este o modo usado pelo RSVP.

c) Modo de Distribuição do Rótulo

Os rótulos são distribuídos usando o método DOD, que é o método adotado pelo RSVP, conforme já apresentado no Capítulo 4 e na Seção 3.6.3, onde encontram-se maiores explicações sobre o referido método.

Assim, uma requisição para ligar rótulos a um determinado Túnel LSP é iniciada pelo AR (nó de entrada) através de uma mensagem RSVP PATH. Para tanto, é utilizado um objeto chamado <LABEL_REQUEST> que é encapsulado nessa mensagem.

Os rótulos são alocados no sentido *downstream* e distribuídos por meio de mensagens RSVP RESV que é propagada no sentido contrário. A mensagem RSVP RESV encapsula um objeto especial chamado <LABEL>, que informa o valor do rótulo alocado localmente.

d) Aspectos Topológicos

Assume-se que a ligação AR-LER é *monolink*, e que o protocolo de rede é o IPv4. Ademais, o AR e o LER devem suportar os protocolos RSVP-SVC (RSVP-TE com a extensão aqui proposta) e o MPLS, a fim de que eles possam associar rótulos a fluxos RSVP. Além de ser uma necessidade óbvia, também mantém a vantagem de, ao combinar o MPLS e o RSVP, permitir que a definição de um fluxo de dados possa ser feita de forma mais flexível, uma vez que o mapeamento rótulo-tráfego de dados pode ser baseado em mais critérios, como já apresentado no Capítulo 4. Assim, um determinado conjunto de pacotes que são designados pelo mesmo valor de rótulo pelo AR pertence a uma mesma FEC e, efetivamente, define um fluxo RSVP.

Quando rótulos são associados a fluxos de tráfego, torna-se possível para um roteador identificar o estado de reserva apropriado para um pacote, baseado no valor do seu rótulo, o que caracteriza uma vantagem adicional do uso do MPLS associado ao RSVP.

e) LSP-SVC

Nesse trabalho convencionou-se adotar o termo LSP-SVC para referir-se aos LSP criados a partir de chamadas discadas.

6.3.4. *Modus Operandi*

Nesta seção serão considerados dois contextos ilustrados na Figura 6.4: um relacionado com o processo de sinalização (Sessão RSVP) e outro relacionado com a fase de transferência de dados propriamente dita, que ocorrerá na camada MPLS.

Todo o processo de sinalização é conduzido pelo RSVP-SVC, que é o RSVP-TE com as adições que provêem suporte ao serviço SVC. Esta sinalização envolve as seguintes atividades:

- Verificação do suporte e operacionalidade do serviço SVC;
- Admissão de um novo LSP-SVC;
- Suspensão de um LSP-SVC;
- Re-ativação de um LSP-SVC;
- Capacidade de Roteamento Explícito em um LSP-SVC;
- Modificação dos parâmetros de tráfego em um LSP-SVC; e,
- Encerramento de um LSP-SVC.

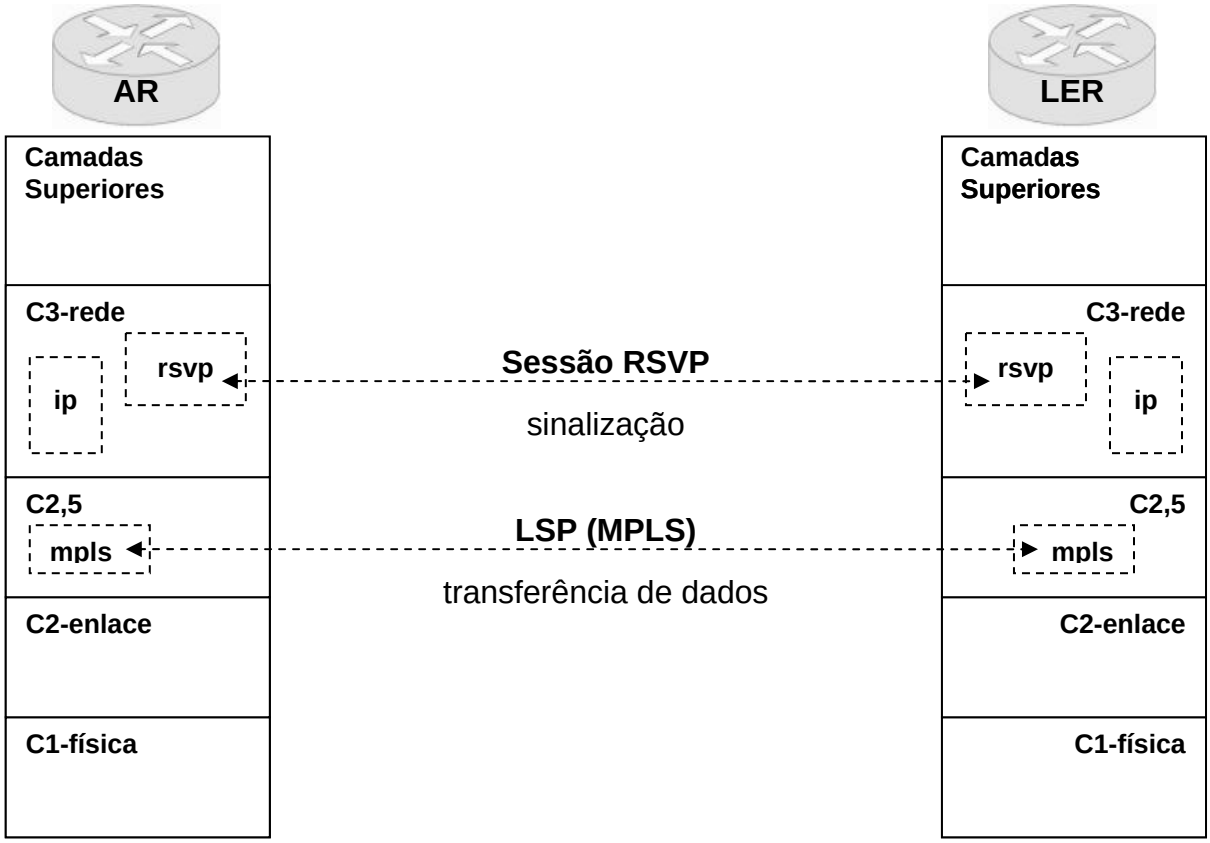

Fig. 6.4 – Contextos do Serviço SVC: Fases de Sinalização e de Transferência de Dados

Vale ressaltar que outras capacidades próprias da operação padrão do RSVP e do RSVP-TE continuam válidas no contexto da extensão proposta e que na Seção 5.5 são abordados aspectos relacionados a questões de compatibilidade.

Os objetos adicionados nesse trabalho são apresentados, de forma resumida, nas Tabelas 6.2 e 6.3. A Tabela 6.2 apresenta os novos objetos e suas classes, conforme adotado pelos padrões do RSVP, destacando-se o número da classe, na coluna Class-Num, bem como os objetos de cada classe e seus respectivos códigos de tipo, coluna C-Type. A Tabela 6.3 fornece uma descrição resumida de cada objeto.

Classe	Objeto	Class-Num	C-Type
SVC_REQUEST_CLASS	SVC_REQUEST	30	1
SVC_RESPONSE_CLASS	SVC_RESPONSE	31	1
LSP_SUSPEND_REQUEST_CLASS	LSP_SUSPEND_REQUEST	32	1
LSP_SUSPEND_RESPONSE_CLASS	LSP_SUSPEND_RESPONSE	33	1
LSP_REACTIVATION_REQUEST_CLASS	LSP_REACTIVATION_REQUEST	34	1
LSP_REACTIVATION_RESPONSE_CLASS	LSP_REACTIVATION_RESPONSE	35	1
LSP_ERO_REQUEST_CLASS	LSP_ERO_REQUEST	36	1
LSP_ERO_RESPONSE_CLASS	LSP_ERO_RESPONSE	37	1

Tab. 6.2 – Relação dos Novos Objetos Definidos pelo RSVP-SVC e seus Respectivos Códigos

Objeto	Descrição Resumida
30	É um objeto obrigatório para o estabelecimento do Serviço SVC, presente na mensagem PATH. É enviado sempre no sentido AR → LER.
31	É um objeto obrigatório para o estabelecimento do Serviço SVC, presente na mensagem RESV. É enviado sempre no sentido LER → AR.
32	Objeto responsável pela requisição de suspensão de um LSP. É encapsulado em mensagens PATH. Pode ser enviado em ambos os sentidos AR ←→LER.
33	Objeto responsável pela suspensão de um LSP, em resposta ao objeto <LSP_SUSPEND_REQUEST>. É encapsulado em mensagens RESV. Pode ser enviado em ambos os sentidos AR ←→ LER.
34	Objeto responsável pela requisição de re-ativação de um LSP. É encapsulado em mensagens PATH. Pode ser enviado em ambos os sentidos AR ←→ LER.
35	Objeto responsável pela re-ativação de um LSP, em resposta ao objeto <LSP_REACTIVATION_REQUEST>. É encapsulado em mensagens RESV. Pode ser enviado em ambos os sentidos AR ←→ LER.
36	Objeto de uso opcional, presente na mensagem PATH, apenas na ligação AR → LER, necessário em casos onde o *host* originador do fluxo deseja explicitar uma determinada rota a seguir.
37	Objeto de uso opcional, presente na mensagem RESV, apenas na ligação LER → AR, enviado como resposta ao objeto <LSP_ERO_REQUEST>.

Tab. 6.3 – Descrição Resumida dos Novos Objetos Propostos para o Serviço SVC

A Figura 6.5 apresenta a arquitetura do serviço SVC em Estelle, onde são indicados os módulos: mSvcTx e mSvcRx, que constituem o Serviço SVC; o módulo mIp, que simula a camada de transporte; e, os módulos mApiAr e mApiLer, que simulam a interface entre as aplicações e o serviço SVC, no AR e no LER, respectivamente. Nesta figura também estão indicados os pontos de interação de cada módulo, assim como os canais que fazem a ligação entre os módulos.

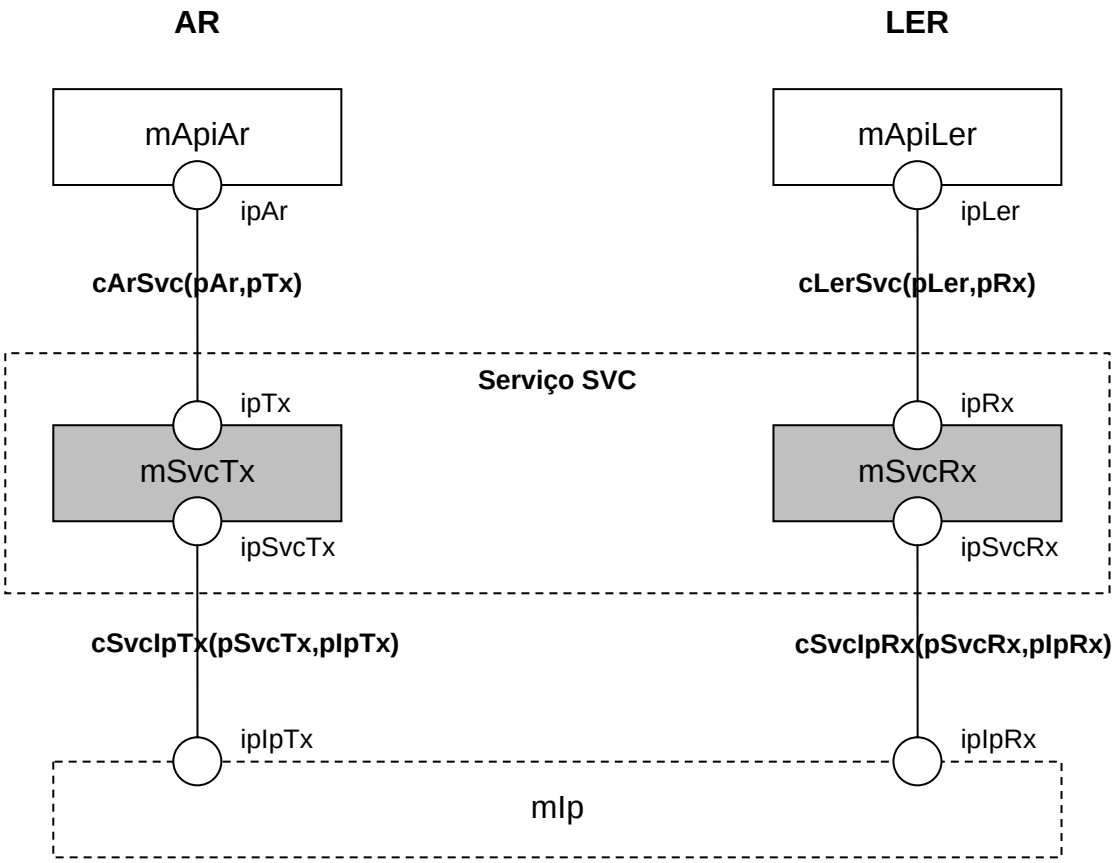

Fig. 6.5 – Arquitetura do Serviço SVC em Estelle

Todas as funcionalidades propostas pelo Serviço SVC são formalmente descritas nas máquinas de estados apresentadas nas Figuras 6.6 e 6.7. Elas apresentam as máquinas de estados do AR e do LER, respectivamente. Para não sobrecarregar estas figuras, as transições das máquinas de estados são etiquetadas de forma simplificada e as Tabelas 6.4 e 6.5, respectivamente, descrevem estas transições de acordo com a seguinte sintaxe:

<transição> ::= <condição> / <ação>.

As expressões das condições e ações podem conter elementos do tipo **?<mensagem>** ou **!<mensagem>**, significando a recepção e o envio de mensagens, respectivamente.

Nas sub-seções seguintes explicações mais detalhadas são apresentadas, com o uso, inclusive, de diagramas de seqüência, visando um maior entendimento.

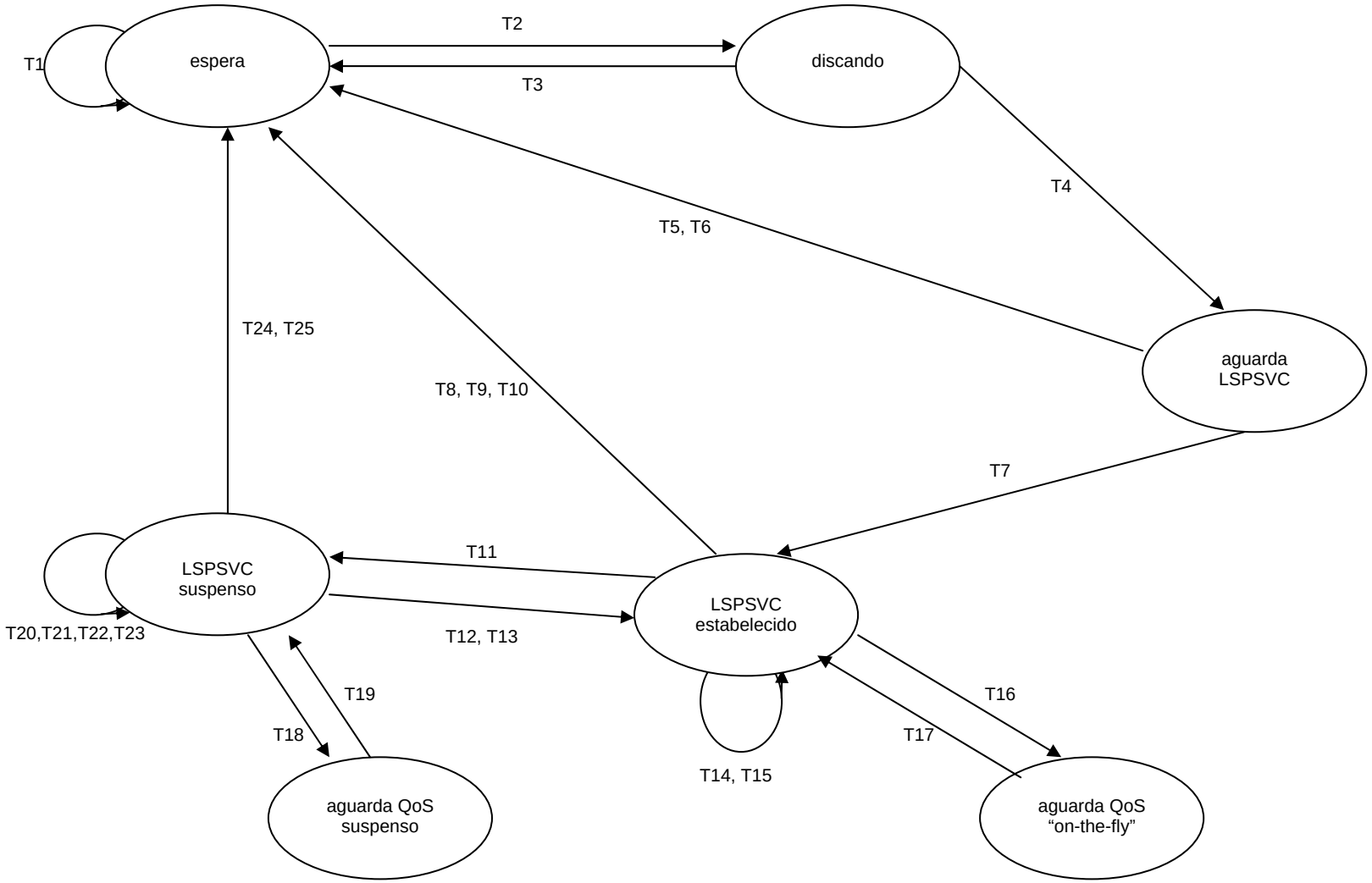

Fig. 6.6 – Máquina de Estados do Serviço SVC-Tx (AR)

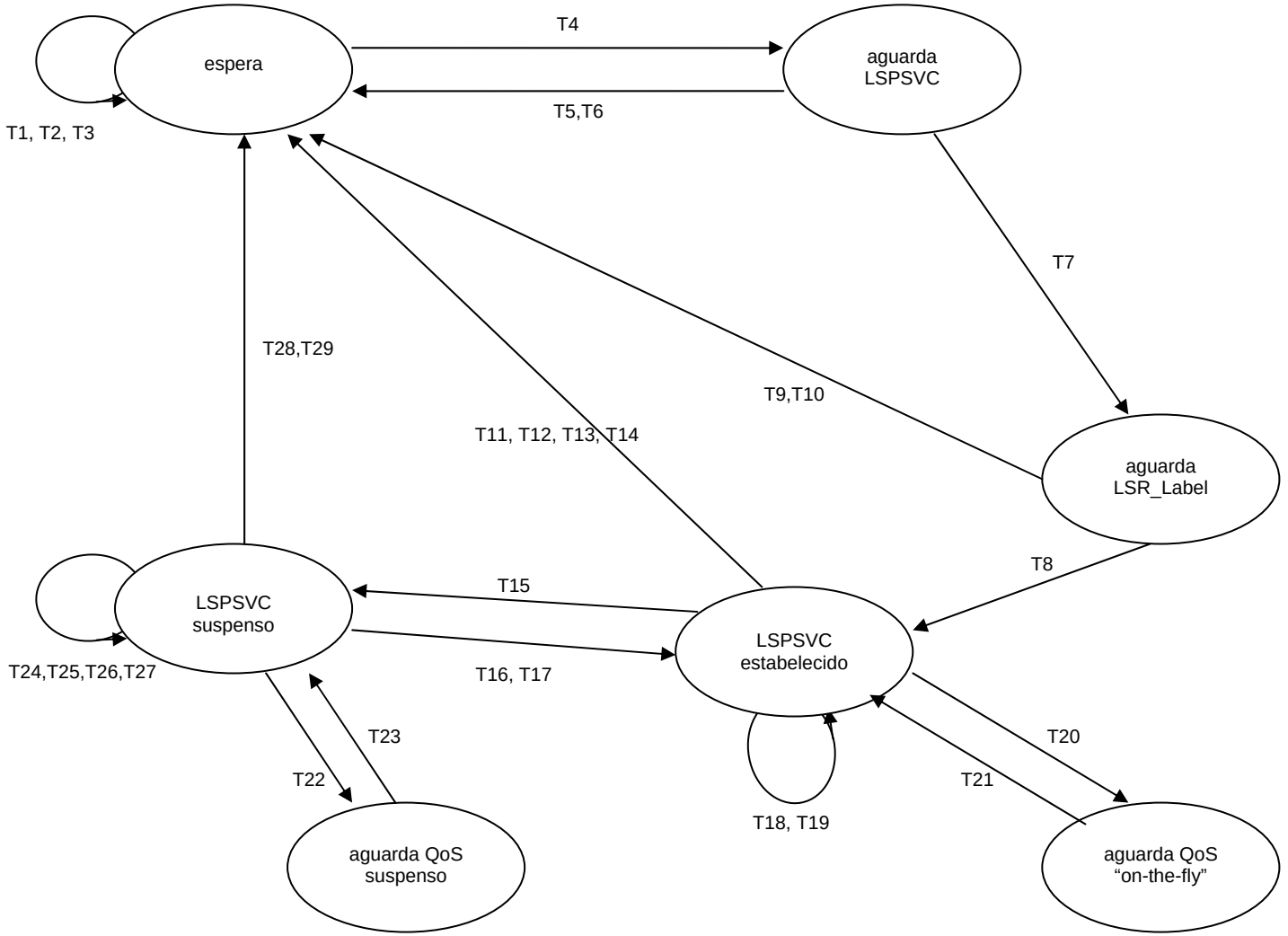

Fig. 6.7 – Máquina de Estados do Serviço SVC-Rx (LER)

A Tabela 6.4 abaixo apresenta a legenda das transições indicadas na Figura 6.6.

T	SIGNIFICADO
1	Se (?resv.LER[<SVC_RP> .AND. NOT objeto_implementado(AR,<SVC_RP>)) / !patherr.LER[erro<13,0>] “Objeto Desconhecido”
2	Se (?lspsvc_req.API-AR) / !path.LER[SVC_RQ]
3	Se (*temp_refresh_esgotado(AR)* / !informe-operador.API-AR[timeout.ind]
4	Se (?resv.LER[<SVC_RP> .AND. objeto_implementado(AR,<SVC_RP>)) / !path.LER[<LABEL_RQ>]
5	Se (*temp_refresh_esgotado(AR)* / !informe-operador.API-AR[timeout.ind]
6	Se (?resverr.LER[erro<26,3>] “Valor QoS inaceitável”) / !informe-operador.API-AR[qosinaceitavel.ind]
7	Se (?resv.LER[<LABEL>] .AND. objeto_implementado(AR,<LABEL>) .AND. tem_recursos(AR,<FILTERSPEC>)) / reserva_recursos(AR,LSP_ID,<FILTERSPEC>) .AND. !informe-operador.API-AR[lspestab.ind,LSP_ID]
8	Se (*temp_refresh_esgotado(AR)*) / deleta_estado(AR,LSP_ID) .AND. !informe-operador.API-AR[timeout.ind,LSP_ID]
9	Se (?pathtear.API-AR[LSP_ID]) / deleta_estado(AR,LSP_ID) .AND. !pathtear.LER[LSP_ID]
10	Se (?pathtear.LER[LSP_ID]) / deleta_estado(AR,LSP_ID) .AND. !informe-operador.API-AR[lspdeletado.ind,LSP_ID]
11	Se (?path.LER[<SUSPEND_RQ>] .AND. lsp_valido(LSP_ID)) / muda_status(AR,LSP_ID) .AND. !resv.LER[<SUSPEND.RP>]
12	Se (?path.LER[<REACTIVATION_RQ>) .AND. lsp_valido(LSP_ID) .AND. tem_recursos(AR,<FILTERSPEC>) / muda_status(AR,LSP_ID) .AND. !resv.LER[<REACTIVATION.RP>]
13	Se (?resv.LER[<REACTIVATION_RP>] .AND. lsp_valido(LSP_ID)) / muda status(AR,LSP_ID)
14	Se (?path.LER[<SUSPEND_RQ>] .AND. NOT lsp_valido(LSP_ID)) / !patherr.LER[erro<26,4>] “LSP_ID inválido”

Continua na próxima página ...

Continuação da página anterior ...

T	SIGNIFICADO
15	Se (?path.LER[<novoFILTERSPEC>] .AND. NOT tem_recursos(AR,<FILTERSPEC>)) / !patherr.LER[erro<26,3>] “Valor QoS inaceitável”
16	Se (?path.API-AR[<novoFILTERSPEC>]) / !path.LER[<novoFILTERSPEC>]
17	Se (?resv.LER[<novoFILTERSPEC>]) / atualiza_valores_qos(AR,LSP_ID,<novoFILTERSPEC>).AND.!resv.API-AR[qosalterado.ind,LSP_ID]
18	Se (?path.API-AR(<novoFILTERSPEC>) .AND. lsp_suspenso(LSP_ID)) / !path.LER[<novoFILTERSPEC>]
19	Se (?resv.LER[<novoFILTERSPEC>] .AND. tem_recursos(AR,<novoFILTERSPEC>)) / !resv.API-AR[<FILTERSPEC>] .AND. ?path.LER[<REACTIVATION_RQ>]
20	Se (?path.LER[<REACTIVATION_RQ>] .AND. NOT lsp_valido(LSP_ID)) / !patherr.LER[erro<26,4>] “LSP_ID inválido”
21	Se (?path.LER[<REACTIVATION_RQ>] .AND. NOT tem_recursos(LER,<FILTERSPEC>) / !resverr.LER[erro<26,5>] “Rede Sem Recursos”
22	Se (?resv.LER[<REACTIVATION_RP>] .AND. NOT lsp_valido(LSP_ID)) / !patherr.LER[<26,4>] “LSP_ID inválido”
23	Se (?path.LER[<SUSPEND_RQ>] .AND. lsp_valido(LSP_ID) / atualiza_temp _suspensão(AR,LSP_ID,novo_valor) .AND. ?resv.LER[<SUSPEND_RP>]
24	Se (temp_susp_esgotado(AR)) / deleta_estado(AR,LSP_ID) .AND. !pathtear.LER[LSP_ID] .AND. !informe-operador.API-AR[temp_susp.ind,LSP_ID]
25	Se (temp_susp_esgotado(AR) .AND. temp_refresh_esgotado(AR)) / deleta_estado(AR,LSP_ID) .AND. !pathtear.LER[LSP_ID]

Tab. 6.4 – Legenda Máquina de Estados do SVC-Tx (AR)

A Tabela 6.5 abaixo apresenta a legenda das transições indicadas na Figura 6.7.

T	SIGNIFICADO
1	Se (?path.AR[<SVC_RQ>] .AND. NOT objeto_implementado(LER,<SVC_RQ>)) / !patherr.AR[erro<13,0>] "Objeto Desconhecido"
2	Se (?path.AR[<SVC_RQ>] .AND. NOT status_operacional(LER,<SVC_RQ>)) / !patherr.AR[erro<26,1>] "Não Operacional"
3	Se (?path.AR[<SVC_RQ>] .AND. NOT chamador_valido(end_chamador)) / !patherr.AR[erro<26,2>] "Não disponível para este assinante"
4	Se (?path.AR[<SVC_RQ>] .AND. objeto_implementado(LER,<SVC_RQ>)) / !resv.AR[<SCV_RP>]
5	Se (*temp_refresh_esgotado(LER))* / !informe-operador.API-LER[timeout.ind]
6	Se (?path.AR[<FILTERSPEC>] .AND. NOT tem_recursos (LER,<FILTERSPEC>)) / !resverr.AR[erro<26,3>] "Valor QoS inaceitável"
7	Se (?path.AR[<LABEL_RQ>] / !path.LSR[<LABEL_RQ>]
8	Se (?resv.LSR[<LABEL>] .AND. tem_recursos(LER,<FILTERSPEC>)) / reserva_recursos(LER,<FILTERSPEC>) .AND. !resv.AR[<LABEL>]
9	Se (?resv.LSR[<LABEL>] .AND. NOT tem_recursos(LER,<FILTERSPEC>)) / !resverr.AR[erro<26,3>] "Valor QoS inaceitável"
10	Se (temp_refresh_esgotado(LER)) / !informe-operador.API-LER[timeout.ind]
11	Se (temp_refresh_esgotado(LER)) / !informe-operador.API-LER[timeout.ind]
12	Se (?pathtear.API-LER[LSP_ID]) / deleta_estado(LER,LSP_ID) .AND. !pathtear.AR[LSP_ID] .AND. !pathtear.LSR[LSP_ID]
13	Se (?pathtear.AR[LSP_ID]) / deleta_estado(LER,LSP_ID) .AND. !pathtear.LSR[LSP_ID] .AND. !pathtear.API-LER[LSP_ID]
14	Se (?pathtear.LSR[LSP_ID]) / deleta_estado(LER,LSP_ID) .AND. !pathtear.AR[LSP_ID] .AND. !pathtear.API-LER[LSP_ID]
15	Se (?path.AR[<SUSPEND_RQ>] .AND. lsp_valido(LSP_ID)) / muda_status(LER,LSP_ID).AND. !resv.AR[<SUSPEND.RP>]

Continua na próxima página ...

Continuação da página anterior ...

T	SIGNIFICADO
16	Se (?path.AR[<REACTIVATION_RQ>] . AND. lsp_valido(LSP_ID) .AND. tem_recursos(LER,<FILTERSPEC>)) / muda_status(LER,LSP_ID) .AND. !resv.AR[<REACTIVATION.RP>]
17	Se (?path.AR[<REACTIVATION_RP>] .AND. lsp_valido(LSP_ID)) / muda_status(LER,LSP_ID)
18	Se (?path.AR[<SUSPEND_RQ>] .AND. NOT lsp_valido(LSP_ID)) / !patherr.AR[erro<26,4>] “LSP_ID inválido”
19	Se (?path.AR[<novoFILTERSPEC>] .AND. NOT tem_recursos(LER,<novoFILTERSPEC>)) / !resverr.AR[erro<26,3>] “Valor QoS inaceitável”
20	Se (?path.AR[<novoFILTERSPEC>]) / !path.LSR[<novoFILTERSPEC>]
21	Se (?resv.LSR[<novoFILTERSPEC>]) / atualiza_valores_qos(LER,LSP_ID,<novoFILTERSPEC>) .AND. !resv.AR[<novoFILTERSPEC>]
22	Se (?path.AR[<novoFILTERSPEC>] .AND. lsp_svc_supenso(LER,LSP_ID)) / !path.LSR[<novoFILTERSPEC>]
23	Se (?resv.LSR[<novoFILTERSPEC>] .AND. tem_recursos(LER,<novoFILTERSPEC>)) / !resv.AR[<FILTERSPEC>] .AND. ?path.AR[<REACTIVATION_RQ>]
24	Se (?path.AR[<REACTIVATION_RQ>] .AND. NOT lsp_valido(LSP_ID)) / !patherr.AR[erro<26,4>] “LSP_ID inválido”
25	Se (?path.AR[<REACTIVATION_RQ>] .AND. NOT tem_recursos(LER,<FILTERSPEC>)) / !resverr.AR[erro<26,5>] “Rede Sem Recursos”
26	Se (?resv.AR[<REACTIVATION_RP>] .AND. NOT lsp_valido(LSP_ID)) / !patherr.AR[<26,4>] “LSP_ID inválido”
27	Se (?path.AR[<SUSPEND_RQ>]) / atualiza_temp_susp(LER,LSP_ID,novovalor) .AND. !resv.AR[<SUSPEND_RP>]
28	Se (temp_susp_esgotado(LER)) / deleta_estado(LER,LSP_ID) .AND. !pathtear.LSR[LSP_ID].AND. !pathtear.AR[LSP_ID].and. !pathtear.API-LER[LSP_ID]
29	Se (temp_susp_esgotado(LER) .AND. temp_refresh_esgotado(LER)) / deleta_estado(LER,LSP_ID) .AND. !pathtear.LSR[LSP_ID] .AND. !pathtear.AR[LSP_ID] .AND. !pathtear.API-LER[LSP_ID]

Tab. 6.5 – Legenda Máquina de Estados do SVC-Rx (LER)

6.3.4.1. Verificação do Suporte e Operacionalidade do Serviço SVC

Os diagramas de seqüência apresentados a seguir adotam a seguinte convenção para identificação das mensagens e objetos enviados: primeiramente, um texto livre em itálico, que indica a ação a ser executada e, após a barra inclinada (*/*), a mensagem RSVP efetivamente usada com o(s) objeto(s) necessário(s) para a consecução da ação.

a) Operação

Inicialmente, o AR deve verificar se o LER provê suporte ao serviço SVC. Caso o LER não esteja habilitado a oferecer o serviço, então ele deve enviar uma mensagem de erro ao AR e a tentativa de uso do serviço SVC é descartada, conforme apresentado na Figura 6.8.

A referida verificação de suporte ao serviço SVC é feita mediante o uso de um novo objeto denominado <SVC_REQUEST>. O objeto <SVC_REQUEST>, além de verificar o suporte ao serviço, ainda verifica sua disponibilidade para o assinante que está requerendo o serviço.

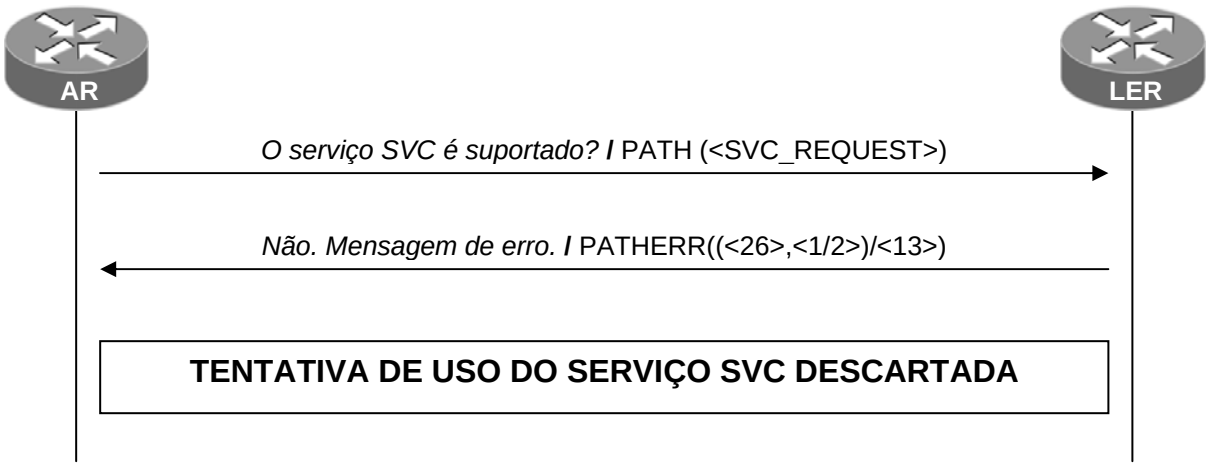

Fig. 6.8 – Serviço SVC não Suportado pelo LER

No caso da indisponibilidade do serviço, o LER responde ao AR, através de uma mensagem de erro PATHERR indicando o tipo e sub-tipo do erro, conforme a seguinte codificação: o código de tipo de erro é **<26>** “Erro de SVC”; e os sub-tipos são: **<1>** “Serviço SVC não operacional”; **<2>** “Serviço SVC não disponível para este assinante”.

No caso do LER não reconhecer o objeto, ele responde com o código de erro **<13>** “Classe de objeto desconhecida”. Vale lembrar que o sub-tipo de erro é opcional.

Caso contrário, quando a operadora é capaz de prover o serviço SVC e o assinante pode requerer o serviço, o LER envia uma resposta positiva usando uma mensagem RESV contendo um objeto chamado <SVC_RESPONSE>, conforme apresentado na Figura 6.9. Opcionalmente, o objeto <SVC_RESPONSE> pode ser usado para indicar valores limites para os parâmetros de tráfego dos LSP a serem admitidos, de forma que os AR possam controlar localmente a quantidade de recursos ainda disponível para sua rede. Assim, os valores a serem requeridos quando da admissão de novos LSP terão uma boa chance de serem aceitos.

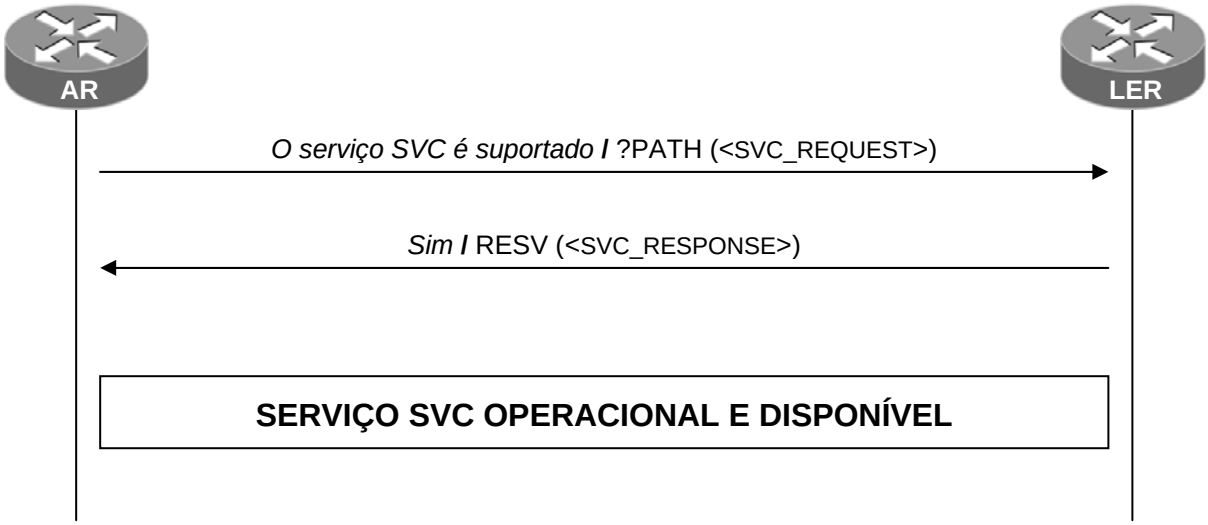

Fig. 6.9 – Serviço SVC Suportado pelo LER

Uma vez que o suporte e a disponibilidade do serviço forem comprovados, o AR pode iniciar o procedimento de admissão de LSP do tipo SVC, que é apresentado na próxima seção.

b) Formato do objeto <SVC_REQUEST>

O objeto <SVC_REQUEST> possui o seguinte formato:

1 – 8	9 – 16	17 – 24	25 – 32
Tamanho (bytes)		*Class-Num*	*C-Type*
IPv4-Address-Chamador			
IPv4-Address-Chamado			

Fig. 6.10 – Formato do Objeto <SVC_REQUEST>

Os campos apresentam os seguintes significados:

- **Tamanho**: Um campo de tamanho de 16 bits que contém o tamanho total do objeto em bytes. Deve sempre ser um múltiplo de 4.
- **Class-Num:** Identifica a classe do objeto, e o seu valor sugerido é 30 (SVC_CLASS).
- **C-Type**: Indica o Tipo de objeto dentro de uma Class-Num, sugere-se que seja 1 o seu valor.
- **IPv4-Address-Chamador**: Endereço IPv4 do Roteador de Acesso.
- **IPv4-Address-Chamado**: End. IPv4 do Roteador de Borda (LER) do Domínio MPLS.

c) Formato do Objeto <SVC_RESPONSE>

O objeto <SVC_RESPONSE> possui o seguinte formato:

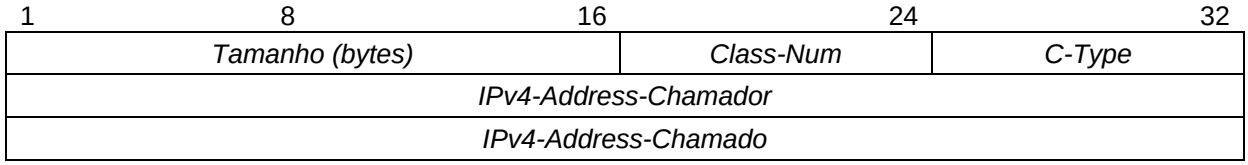

Fig. 6.11 – Formato do Objeto <SVC_RESPONSE>

Os campos apresentam os seguintes significados:

- **Tamanho**: Um campo de tamanho de 16 bits que contém o tamanho total do objeto em bytes. Deve sempre ser um múltiplo de 4.
- **Class-Num:** Identifica a classe do objeto, e o seu valor sugerido é 30 (SVC_CLASS).
- **C-Type**: Indica o Tipo de objeto dentro de uma Class-Num, sugere-se que seja 2 o seu valor.
- **IPv4-Address-Chamador**: Endereço IPv4 do Roteador de Acesso.
- **IPv4-Address-Chamado**: End. IPv4 do Roteador de Borda do Domínio MPLS.

d) Tratamento de Exceção dos Objetos <SVC_REQUEST> e <SVC_RESPONSE>

Em circunstâncias normais, um nó, que não seja o AR ou o LER, jamais receberá os objetos <SVC_REQUEST> e <SVC_RESPONSE> em uma mensagem PATH e RESV, respectivamente.

Apenas os LER podem receber objetos <SVC_REQUEST> de AR diretamente conectados.

Os AR, por sua vez, nunca receberão objetos <SVC_RESPONSE> em mensagens RESV, a não ser que tenham incluído um objeto <SVC_REQUEST> na mensagem PATH correspondente. Se isto ocorrer ele deve descartar tal objeto.

Se o LER não reconhecer o objeto <SVC_REQUEST>, ele envia uma mensagem de erro para o AR: PATHERR com o código de erro **<13>** “classe de objeto desconhecida”.

Se o AR receber uma mensagem com um objeto <SVC_REQUEST> do LER, ele deve desconsiderá-la.

6.3.4.2. Admissão de um Novo LSP-SVC

a) Operação

Uma vez estabelecida uma sessão RSVP entre o AR e o LER e, no caso de transmissão bi-direcional, também entre o LER e o AR, e, tendo sido comprovado o suporte e operacionalidade do serviço SVC, pode-se, então, iniciar o procedimento de Controle de Admissão de um novo LSP-SVC, conforme fluxograma apresentado na Figura 6.12.

Vale ressaltar que a execução do procedimento em apreço é necessária não apenas quando da admissão de um novo LSP, mas também quando da re-ativação de um LSP-SVC que esteja em modo suspenso, como é apresentado na Seção 6.3.4.4. Assim, uma vez que determinado LSP tenha sido admitido, pode-se iniciar a transmissão do fluxo de dados.

De fato, um ponto importante a ser considerado quando do estabelecimento de LSP está relacionado com a definição da estratégia para controle de admissão, cujo objetivo é determinar se um novo LSP pode ser acomodado na rede, de acordo com os parâmetros de desempenho solicitados pelo cliente. Neste caso, uma vez que a rede garanta os recursos necessários para o atendimento da solicitação do cliente, o assinante e a rede devem estabelecer, de forma *on-line,* um contrato de tráfego temporário também chamado de SLA – ou seja, eles devem concordar com um descritor de tráfego, caracterizando o tráfego a ser transportado – que atenderá ao LSP. Esta possibilidade é interessante do ponto de vista dos ISP, pois abre novas possibilidades de negócio e, conseqüentemente, de uso da infra-estrutura de rede instalada. Fica ainda em aberto a questão da contabilização do uso desse serviço, que deve ser definido no contexto de cada LSP

contratado. Do ponto de vista do cliente, essa também é uma possibilidade interessante, pois o mesmo poderá contratar serviços adicionais à medida em que eles sejam realmente necessários.

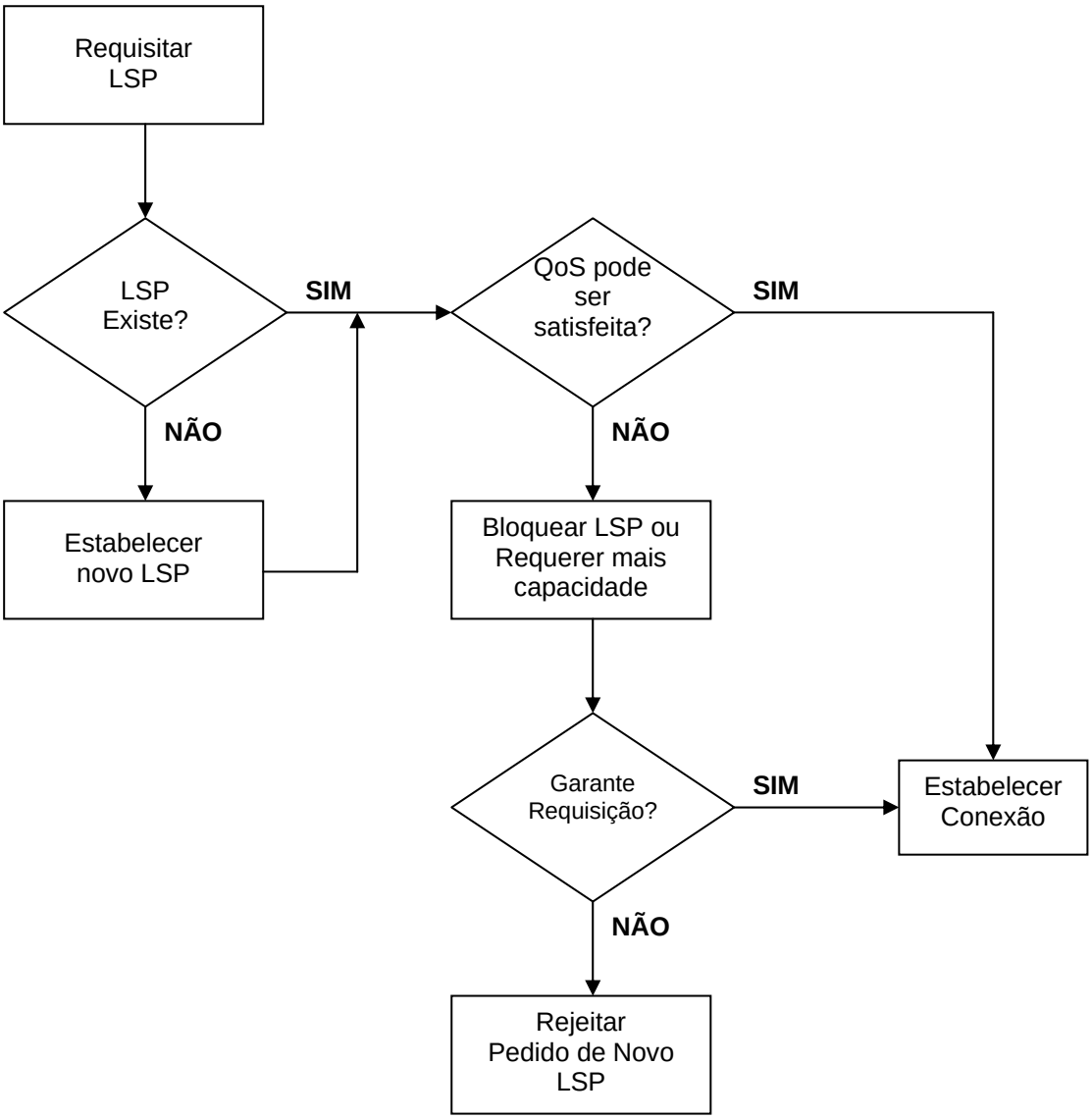

Figura 6.12 – Fluxograma Relativo ao Controle de Admissão de LSP-SVC

Uma vez “assinado” o contrato *on-line*, subentende-se que a rede concorda em suportar o tráfego com as características especificadas e que o assinante concorda em não exceder os limites de desempenho garantidos a ele.

Desta forma, considerando que o serviço SVC está operacional e disponível para determinado assinante então o seu roteador de acesso pode,

quando for necessário, requisitar rótulos para LSP que possam ser provisionados na interface de saída.

O procedimento de sinalização deve ser iniciado pelo AR, que envia uma mensagem RSVP PATH especificando os atributos do LSP no sentido AR → LER. O LER deve encaminhar essa mensagem PATH para a rede e, depois de receber a mensagem RESV do receptor, e verificado que pode atender a solicitação, responde com uma mensagem RESV, informando, “de carona”, o rótulo a ser usado na direção AR → LER. Em seguida, se necessário, o procedimento é repetido na direção oposta (LER → AR). Os dois LSP, apesar de distintos, são correlacionados pela associação dos seus identificadores, resultando em um par de LSP bi-direcionais correlatos. A Figura 6.13 ilustra a seqüência básica de troca de mensagens para o estabelecimento do LSP-SVC.

É importante relembrar que nesta troca de mensagens para o estabelecimento do LSP-SVC, além da negociação do rótulo também pode ser necessária uma negociação dos parâmetros de tráfego, e que o RSVP já traz na sua operação básica procedimentos que possibilitam esse “ajuste fino”.

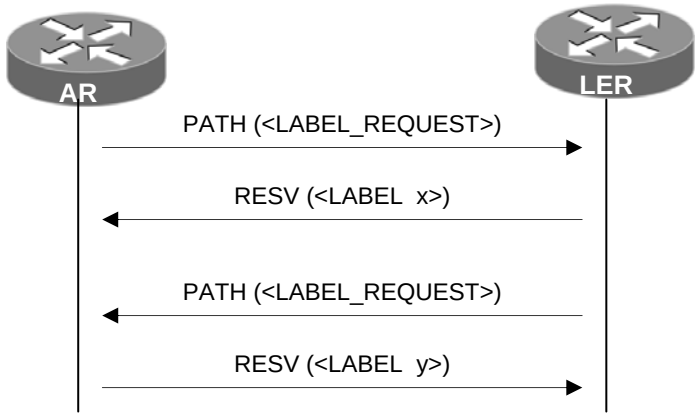

Fig. 6.13 – Troca de Mensagens para o Estabelecimento de um LSP-SVC bi-Direcional

A Figura 6.14 explicita a negociação dos parâmetros de tráfego indicando os objetos que são usados.

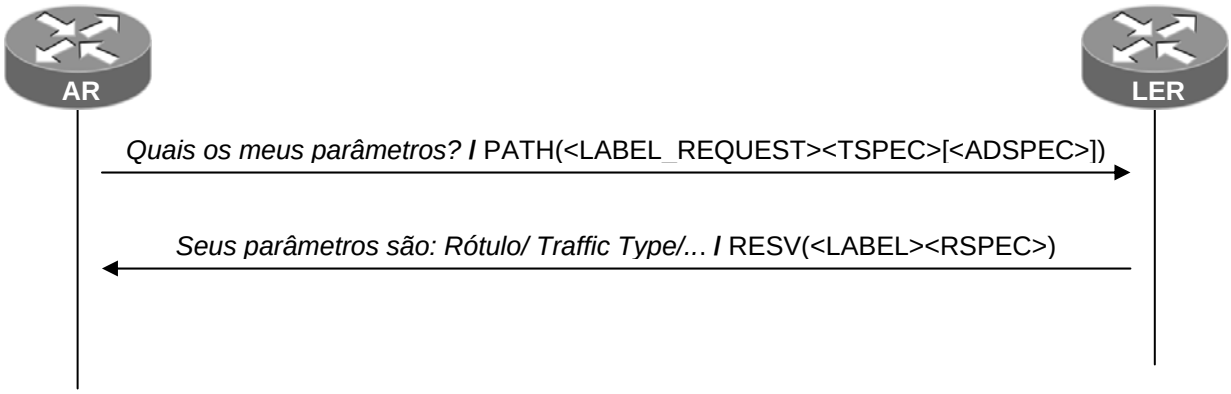

Fig. 6.14 – Negociação dos Parâmetros de Tráfego

Visto que as reservas RSVP são unidirecionais por natureza, então os recursos deverão ser reservados em ambas as direções, se necessário. Assim, o AR inicia o procedimento de reserva de recursos pela transmissão de uma mensagem PATH contendo um objeto <TSPEC>. Como discutido em [Awduche01], o <TSPEC> descreve as características de tráfego da fonte em termos de *Peak Data Rate (PDR)*, *average data rate*, *burst size*, tamanho de pacote máximo e tamanho de pacote mínimo, conforme apresentado na Tabela 6.6. A mensagem PATH também pode conter opcionalmente um objeto <ADSPEC> que é atualizado pelos elementos da rede ao longo do caminho para indicar informações tais como: a disponibilidade para serviços QoS em particular, a banda máxima disponível ao longo do caminho, assim como a latência mínima e a MTU do caminho. O estado é instalado em cada dispositivo atravessado pela mensagem PATH, mas nenhum recurso é, de fato, reservado ainda. Entre outras coisas, estes estados identificam os nós RSVP vizinhos, o que fixa o caminho para reserva. Somente quando o receptor responde a mensagem PATH com uma mensagem RESV é que os recursos são efetivamente reservados.

PARÂMETRO	**DESCRIÇÃO**
PDR (*Peak Data Rate*)	Taxa máxima em que o usuário pretende transmitir pacotes.
ADR (*Average Data Rate*)	Taxa média em que o usuário pretende transmitir pacotes.
BS (*Burst Size*)	Número máximo de pacotes que podem ser enviados, ponta a ponta, na taxa de pico PDR.
Tamanho do Pacote Máximo	Tamanho máximo do pacote em Bytes.
Tamanho de Pacote Mínimo	Tamanho mínimo do pacote em Bytes.

Tab. 6.6 – Parâmetros de Tráfego Negociados entre o AR e o LER

Quanto ao tamanho dos parâmetros, todos são palavras de 32 bits, conforme apresentado na Figura 6.15.

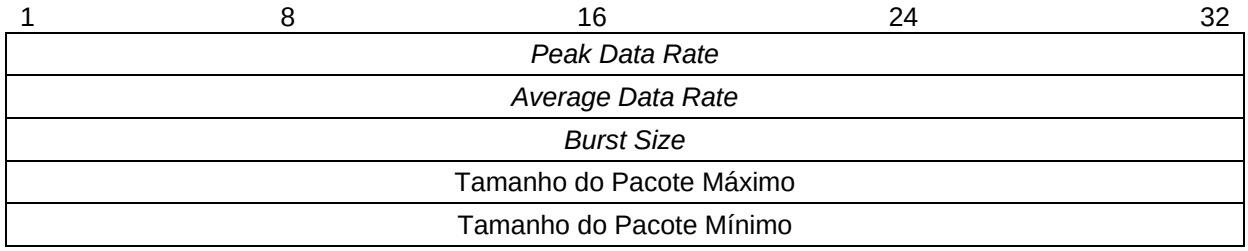

Fig. 6.15 – Tamanhos dos Parâmetros de Tráfego

Após o recebimento de uma mensagem PATH, o destino pode examinar o objeto <TSPEC> do emissor e o <ADSPEC> juntamente com informação sobre a política local, a fim de determinar a especificação de QoS real que deve ser incluída na mensagem RESV. A mensagem RESV simplesmente segue o caminho reverso estabelecido pela mensagem PATH, e os recursos apropriados são reservados em cada nó.

De uma forma mais específica, um pedido de reserva assume a seguinte a sintaxe: *reservation.request* (*flowdescriptor* (*flowspec*; *filterspec*)); onde:

- *flowspec*: especifica a QoS desejada. Este parâmetro será usado para setar os parâmetros no escalonador de pacotes do nó. Ele especifica os atributos da CoS e mais dois conjuntos de parâmetros numéricos, a saber: Rspec e Tspec.
- *filterspec*: define o conjunto de pacotes que receberão a QoS. Ele seta os parâmetros no classificador de pacote ou outro mecanismo da camada 2, como o MPLSoLAN por exemplo. Ademais, vale ressaltar que eles são dependentes do protocolo de rede (IPv4 ou IPv6). Neste trabalho será considerado apenas o baseado no IPv4.

É válido lembrar que em cada nó intermediário, uma requisição de reserva dispara duas ações: primeiro, fazer uma reserva no *link*; e, segundo, encaminhar a requisição para o próximo nó anterior (*upstream*).

No caso de impossibilidade de acordo em relação aos níveis de QoS, uma resposta RESVERR deve ser enviada (do LER para o AR), informando o código do erro: **<26>** “Erro de SVC”, e o seu sub-tipo: **<3>** “Valor de QoS inaceitável”.

Uma vez estabelecido o Túnel LSP, todo o fluxo de dados poderá ser então transmitido.

6.3.4.3. Suspensão de um LSP-SVC

a) Operação

A Figura 6.16 apresenta a seqüência de mensagens trocadas entre o AR e o LER, visando a suspensão de um LSP. Conforme mostrado na figura, tanto o AR (caso A) quanto o LER (caso B) podem tomar a iniciativa desse procedimento, caracterizando um serviço iniciado pelo usuário e outro serviço iniciado pelo provedor. Nesse sentido eles fazem uso do objeto <LSP_SUSPEND_REQUEST>.

Uma vez que o LER tenha procedido com a suspensão do LSP, ele responde ao AR, usando o objeto <LSP_SUSPEND_RESPONSE> no corpo de uma mensagem RESV, conforme descrição a seguir.

Se, em ambos os casos, não chegar um pedido de re-ativação dentro do tempo de suspensão máximo indicado no objeto, então o referido LSP é definitivamente cancelado e uma mensagem PATHTEAR é enviada aos vizinhos RSVP.

É importante considerar ainda que na hipótese do objeto <LSP_SUSPEND_REQUEST> conter algum valor inválido como, por exemplo: o AR requerer a suspensão de determinado LSP cujo LSP_ID não é conhecido pelo LER ou vice-versa, então, o nó que recebeu o objeto com valor desconhecido, responde com uma mensagem PATHERR informando o código do erro: **<26>** “Erro de SVC”, e o seu sub-tipo: **<4>** “Valor de LSP_ID desconhecido ou inexistente”.

Um outro fato a destacar é que, apesar deste mecanismo apresentar a desvantagem de ocupar memória e sobrecargas devido aos refrescamentos nos AR e nos LER, ganha-se por não ter que renegociar os valores dos parâmetros de tráfego posteriormente, quando da re-ativação do LSP suspenso.

Adicionalmente, pode ser necessária a suspensão do LSP correlato, devido mais uma vez, a característica da uni-direcionalidade dos fluxos RSVP.

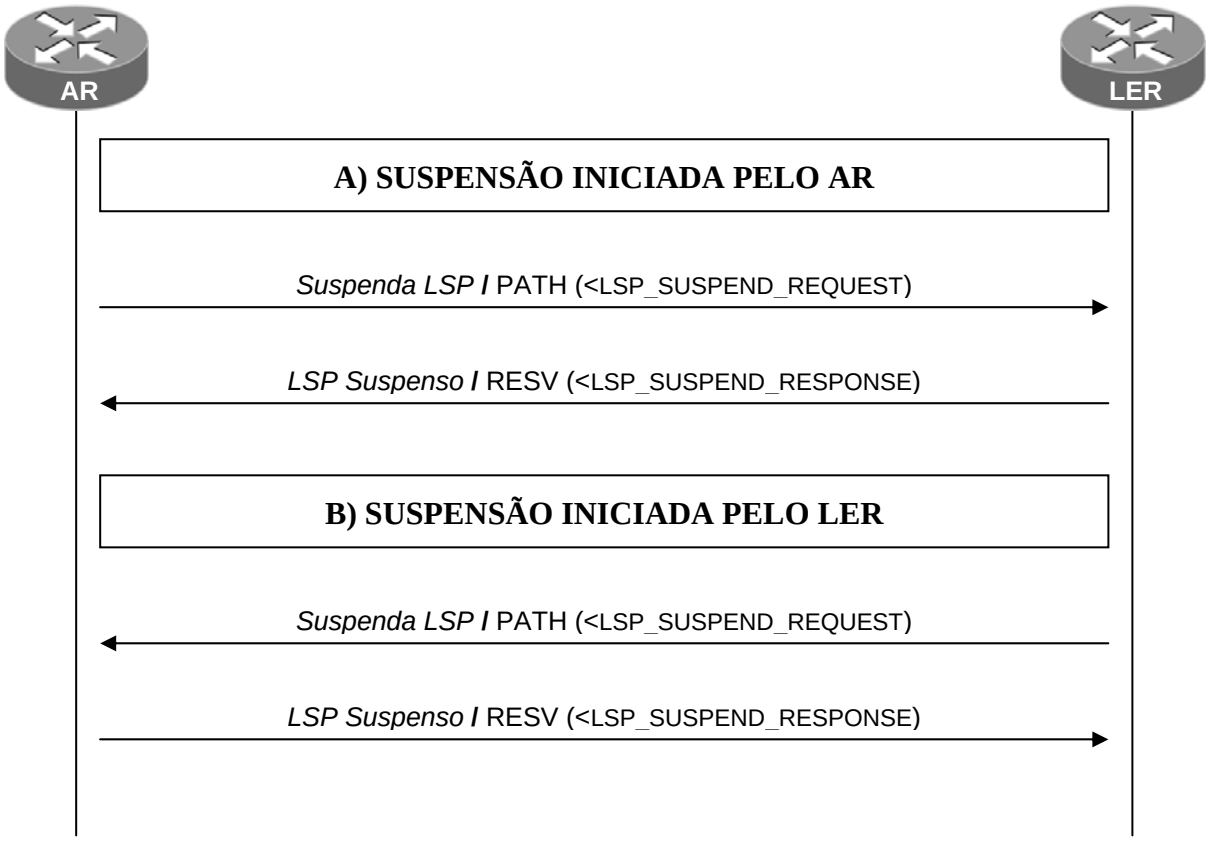

Figura 6.16 – Suspensão de um LSP-SVC

b) Formato do Objeto <LSP_SUSPEND_REQUEST>

O objeto <LSP_SUSPEND_REQUEST> possui o seguinte formato:

1 8 16 24 32		
Tamanho (bytes)	*Class-Num*	*C-Type*
IPv4-Address-Chamador		
IPv4-Address-Chamado		
Temporizador-de-suspensão		
LSP_ID	*Reservado*	

Fig. 6.17 – Formato do Objeto <LSP_SUSPEND_REQUEST>

Os campos apresentam os seguintes significados:

- **Tamanho**: Um campo de tamanho de 16 bits que contém o tamanho total do objeto em bytes. Deve sempre ser um múltiplo de 4.
- **Class-Num:** Identifica a classe do objeto, e o seu valor sugerido é 31 (LSP_SUSPEND_CLASS).

- **C-Type**: Indica o Tipo de objeto dentro de uma Class-Num, sugere-se que seja 1 o seu valor.
- **IPv4-Address-Chamador**: Endereço IPv4 do Roteador de Acesso.
- **IPv4-Address-Chamado**: End. IPv4 do Roteador de Borda do Domínio MPLS.
- **Temporizador-de-Suspensão**: Tempo de aguardo pela re-ativação do LSP em milissegundos.
- **LSP_ID**: Identificador de 16 bits usado no objeto <SESSION> para identificar um túnel. Este valor permanece constante durante a existência do túnel.
- **Reservado**: Trabalhos futuros poderão usar este campo para armazenar o TUNNEL_ID, aproveitando assim uma característica já prevista no RSVP-TE, de ambientes *multi-link*.

c) Formato do Objeto <LSP_SUSPEND_RESPONSE>

O objeto <LSP_SUSPEND_RESPONSE> possui o seguinte formato:

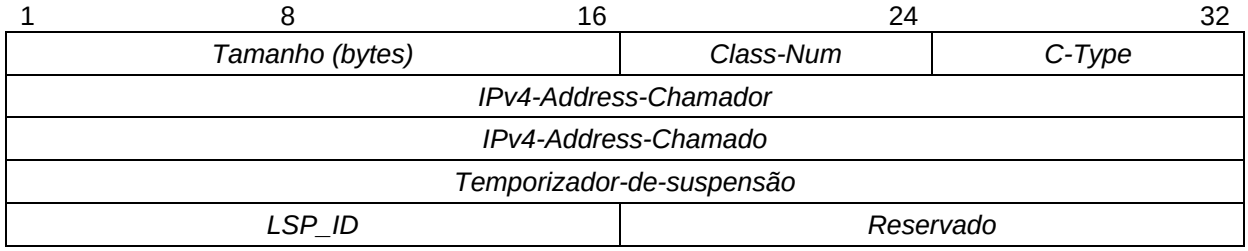

Fig. 6.18 – Formato do Objeto <LSP_SUSPEND_RESPONSE>

Os campos apresentam os seguintes significados:

- **Tamanho**: Um campo de tamanho de 16 bits que contém o tamanho total do objeto em bytes. Deve sempre ser um múltiplo de 4.
- **Class-Num:** Identifica a classe do objeto, e o seu valor sugerido é 31 (LSP_SUSPEND_CLASS).
- **C-Type**: Indica o Tipo de objeto dentro de uma Class-Num, sugere-se que seja 2 o seu valor.
- **IPv4-Address-Chamador**: Endereço IPv4 do Roteador de Acesso.
- **IPv4-Address-Chamado**: End. IPv4 do Roteador de Borda do Domínio MPLS.
- **Temporizador-de-Suspensão**: Tempo de aguardo pela re-ativação do LSP em milissegundos.

- **LSP_ID**: Identificador de 16 bits usado no objeto <SESSION> para identificar um túnel. Este valor permanece constante durante a existência do túnel.
- **Reservado**: Trabalhos futuros poderão usar este campo para armazenar o TUNNEL_ID, aproveitando assim uma característica já prevista no RSVP-TE, de ambientes *multi-link*.

d) Tratamento de Exceção dos Objetos <LSP_SUSPEND_REQUEST> e <LSP_SUSPEND_RESPONSE>

Assim como os objetos <SVC_REQUEST> e <SVC_RESPONSE>, em circunstâncias normais, um nó, que não seja o AR ou o LER jamais receberá os objetos da classe LSP_SUSPEND_CLASS em uma mensagem PATH e RESV.

Apenas os LER e seus AR diretamente conectados podem receber os objetos em apreço. Não é necessário propagar este objeto ao longo de todo o Túnel LSP uma vez que, no núcleo não há maiores problemas de recursos, assim, não será oneroso em termos de processamento para o ISP, reconstruir o túnel quando da reativação do LSP. Portanto, no núcleo tais estados serão deletados por falta de refrescamento ou por intervenção direta do operador através de mensagens *teardown*.

Se o LER e/ou o AR não reconhecerem os objetos <LSP_SUSPEND_REQUEST> e/ou <LSP_SUSPEND_RESPONSE), então eles enviam para o seu par uma mensagem de erro: PATHERR com o código de erro **<13>** “classe de objeto desconhecida”.

6.3.4.4. Reativação de um LSP-SVC

a) Operação

Uma vez que determinado LSP-SVC esteja em modo suspenso, ele pode ser reativado através do objeto <LSP_REACTIVATION_REQUEST>, conforme ilustrado na Figura 6.19.

Vale salientar que a reativação de um LSP pode ser de iniciativa do AR ou do LER, e que pode ser necessário a sua re-ativação em ambos os sentidos, a saber: AR → LER e LER → AR.

Dois objetos foram definidos para esta operação: o <LSP_REACTIVATION_REQUEST> e o <LSP_REACTIVATION_RESPONSE>. O primeiro é responsável por requisitar a re-ativação de um LSP para uso imediato, e é

encapsulado numa mensagem PATH, enquanto o segundo objeto é encarregado de comunicar a resposta ao solicitante, sendo encapsulado em uma mensagem RESV.

Quando a comunicação de dados entre o AR e o LER é bi-direcional, faz-se necessário a re-ativação dos dois LSP correlatos. A Figura 6.16 mostra esse procedimento, onde procurou se otimizar o processo, de forma que a confirmação da reativação do LSP num sentido, provoca automaticamente a confirmação da reativação do LSP correlato, através do envio de uma segunda mensagem RESV contendo um objeto <LSP_REACTIVATION_RESPONSE>.

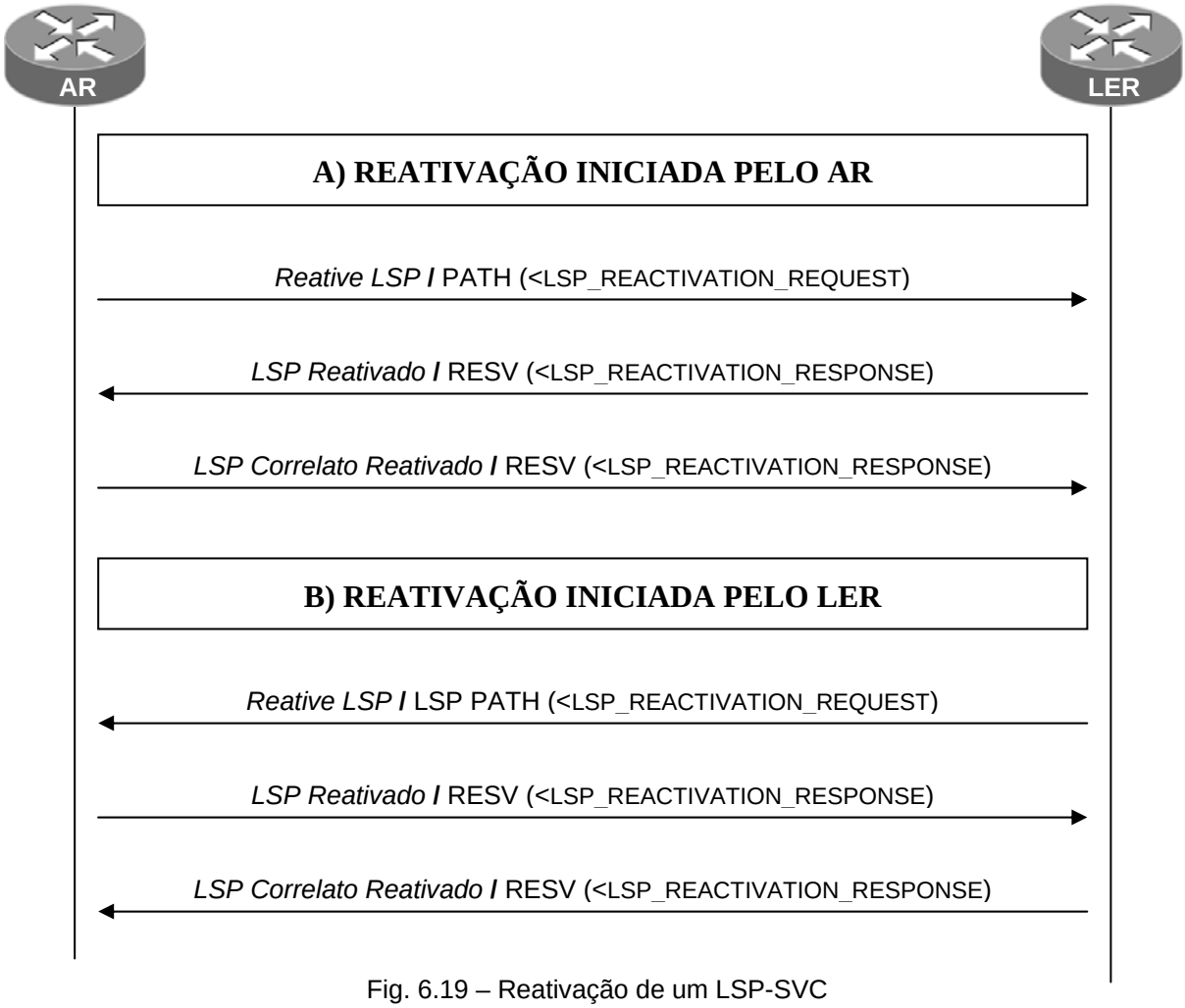

Fig. 6.19 – Reativação de um LSP-SVC

É importante observar que o procedimento de reativação de um LSP iniciado pelo AR também está sujeito ao Controle de Admissão. Quando o LER receber um <LSP_REACTIVATION_REQUEST> ele deve verificar se há recursos na rede, ou seja, ele deve submeter o pedido de reativação ao processo de Controle de Admissão. Se não houver recursos disponíveis na rede no momento, ele deve

recusar o pedido de reativação, informando através de uma mensagem RESVERR o código de erro: **<26>** “Erro de SVC”, e o seu sub-tipo: **<5>** “Rede sem recursos no momento”. Ademais, ele deve retemporizar a suspensão ou cancelar de vez o LSP suspenso, seja diretamente através de uma mensagem de desconexão ou, por não refrescar o estado relacionado.

b) Formato do Objeto <LSP_REACTIVATION_REQUEST>

O objeto <LSP_REACTIVATION_REQUEST> possui o seguinte formato:

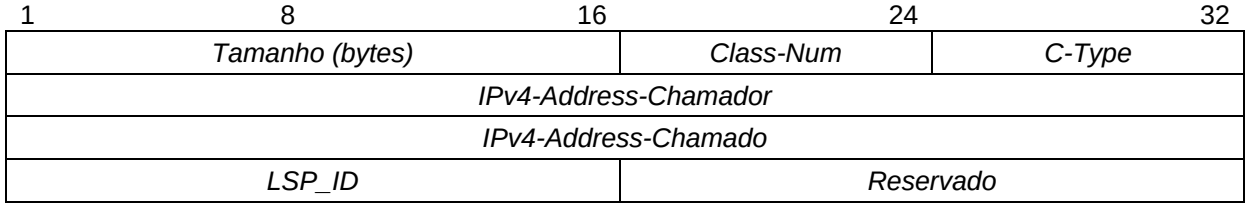

Fig. 6.20 – Formato do Objeto <LSP_REACTIVATION_REQUEST>

Os campos apresentam os seguintes significados:

- **Tamanho**: Um campo de tamanho de 16 bits que contém o tamanho total do objeto em bytes. Deve sempre ser um múltiplo de 4.
- **Class-Num:** Identifica a classe do objeto, e o seu valor sugerido é 32 (LSP_REACTIVATION_CLASS).
- **C-Type**: Indica o Tipo de objeto dentro de uma Class-Num, sugere-se que seja 1 o seu valor.
- **IPv4-Address-Chamador**: Endereço IPv4 do Roteador de Acesso.
- **IPv4-Address-Chamado**: End. IPv4 do Roteador de Borda do Domínio MPLS.
- **LSP_ID**: Identificador de 16 bits usado no objeto <SESSION> para identificar um túnel. Este valor permanece constante durante toda a existência do túnel.
- **Reservado**: Trabalhos futuros poderão usar este campo para armazenar o TUNNEL_ID, aproveitando assim uma característica já prevista no RSVP-TE, de ambientes *multi-link*.

c) Formato do Objeto <LSP_REACTIVATION_RESPONSE>

O objeto <LSP_REACTIVATION_RESPONSE> possui o seguinte formato:

1 8	16	24 32
Tamanho (bytes)	*Class-Num*	*C-Type*
IPv4-Address-Chamador		
IPv4-Address-Chamado		
LSP_ID	*Reservado*	

Fig. 6.21 – Formato do Objeto <LSP_REACTIVATION_RESPONSE>

Os campos apresentam os seguintes significados:

- **Tamanho**: Um campo de tamanho de 16 bits que contém o tamanho total do objeto em bytes. Deve sempre ser um múltiplo de 4.
- **Class-Num:** Identifica a classe do objeto, e o seu valor sugerido é 32 (LSP_REACTIVATION_CLASS).
- **C-Type**: Indica o Tipo de objeto dentro de uma Class-Num, sugere-se que seja 2 o seu valor.
- **IPv4-Address-Chamador**: Endereço IPv4 do Roteador de Acesso.
- **IPv4-Address-Chamado**: End. IPv4 do Roteador de Borda do Domínio MPLS.
- **LSP_ID**: Identificador de 16 bits usado no objeto <SESSION> para identificar um túnel. Este valor permanece constante durante toda a existência do túnel.
- **Reservado**: Trabalhos futuros poderão usar este campo para armazenar o TUNNEL_ID, aproveitando assim uma característica já prevista no RSVP-TE, de ambientes *multi-link*.

d) Tratamento de Exceção dos Objetos <LSP_REACTIVATION_REQUEST> e <LSP_REACTIVATION_RESPONSE>

Assim como os objetos da classe LSP_SUSPEND_CLASS, em circunstâncias normais, um nó, que não seja o AR ou o LER, jamais receberá os objetos da classe LSP_REACTIVATION_CLASS em uma mensagem PATH e RESV.

Apenas os LER e seus AR diretamente conectados podem receber os objetos em apreço. Se o LER ou o AR não reconhecerem os objetos <LSP_REACTIVATION_REQUEST> e/ou <LSP_REACTIVATION_RESPONSE),

então eles enviam para o seu par uma mensagem de erro: PATHERR com o código de erro **<13>** “classe de objeto desconhecida”.

6.3.4.5. Capacidade de Roteamento Explícito em um LSP-SVC

a) Operação

A capacidade de roteamento explícito introduzida pelo RSVP-TE é mantida através do uso do objeto <EXPLICIT_ROUTE> em mensagens PATH, desde que observada as restrições necessárias na fase de negociação. Ou seja, uma vez que o LER “assinou” o SLA com o AR sobre a prestação do serviço SVC, pode ser estabelecida uma rota explícita (ER – *Explicit Route*), quando o LER atuará como procurador do AR. Vale destacar que a ER só faz sentido a partir do LER, uma vez que partimos do pressuposto de que há apenas uma ligação entre o AR e o LER. Trabalhos futuros poderão estudar os aspectos necessários para um contexto, onde existam mais de uma ligação entre esses roteadores.

b) Formato do Objeto <LSP_ERO_REQUEST>

O objeto <LSP_ERO_REQUEST> possui o seguinte formato:

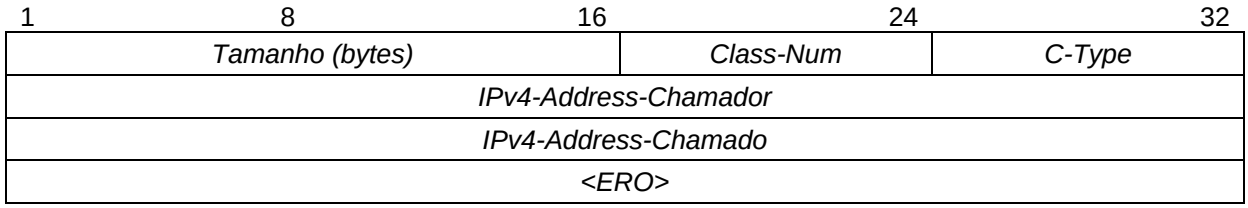

Fig. 6.22 – Formato do Objeto <LSP_ERO_REQUEST>

Os campos apresentam os seguintes significados:

- **Tamanho**: Um campo de tamanho de 16 bits que contém o tamanho total do objeto em bytes. Deve sempre ser um múltiplo de 4.
- **Class-Num:** Identifica a classe do objeto, e o seu valor sugerido é 33 (LSP_ERO_CLASS).
- **C-Type**: Indica o Tipo de objeto dentro de uma Class-Num, sugere-se que seja 1 o seu valor.
- **IPv4-Address-Chamador**: Endereço IPv4 do Roteador de Acesso.
- **IPv4-Address-Chamado**: End. IPv4 do Roteador de Borda do Domínio MPLS.
- **ERO**: Objeto <EXPLICIT_ROUTE> definido em [Awduche01].

c) Formato do Objeto <LSP_ERO_RESPONSE>

O objeto <LSP_ERO_RESPONSE> possui o seguinte formato:

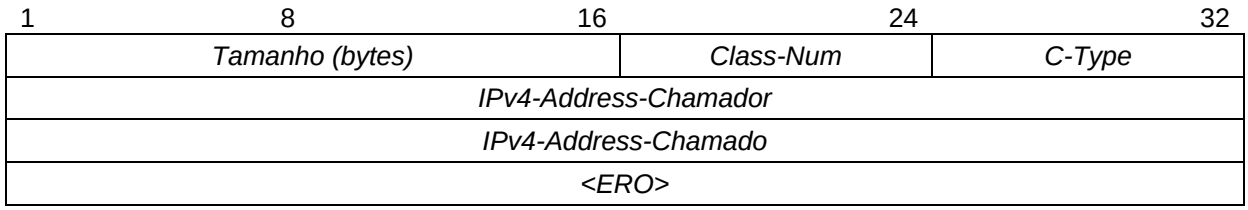

Fig. 6.23 – Formato do Objeto <LSP_ERO_RESPONSE>

Os campos apresentam os seguintes significados:

- **Tamanho**: Um campo de tamanho de 16 bits que contém o tamanho total do objeto em bytes. Deve sempre ser um múltiplo de 4.
- **Class-Num:** Identifica a classe do objeto, e o seu valor sugerido é 33 (LSP_ERO_CLASS).
- **C-Type**: Indica o Tipo de objeto dentro de uma Class-Num, sugere-se que seja 2 o seu valor.
- **IPv4-Address-Chamador**: Endereço IPv4 do Roteador de Acesso.
- **IPv4-Address-Chamado**: End. IPv4 do Roteador de Borda do Domínio MPLS.
- **ERO**: Objeto <EXPLICIT_ROUTE> definido em [Awduche01].

6.3.4.6. Modificação dos Parâmetros de Tráfego em um LSP-SVC

A proposta aqui apresentada relativa a modificação dos parâmetros de tráfego em um LSP-SVC é compatível com a definida pelo RSVP-TE, a saber, uma nova mensagem PATH é transmitida, usando-se o objeto <SESSION>, mantendo-se ainda os mesmos valores para o TUNNEL_ID e também para o EXTENDED_TUNNEL_ID. Entretanto, um novo valor para LSP_ID é usado, a fim de formar um novo <SENDER_TEMPLATE> diferente do original. É, então, criado um novo objeto <EXPLICIT_ROUTE> para a nova rota, se for o caso. A nova mensagem PATH é enviada e, ambas as mensagens PATH, a nova e a velha, serão refrescadas no nó de entrada (AR).

O LER responde com uma mensagem RESV com um descritor de fluxo formatado do seguinte modo: [<FLOWSPEC> <old_FILTER_SPEC> <old_LABEL_OBJECT> <new_FILTER_SPEC> <new_LABEL_OBJECT>].

Quando o AR receber as mensagens RESV relativas ao novo LSP, ele pode começar a usar a nova rota, com os parâmetros de tráfego modificados. Note-se que, em seguida, ele deve enviar uma mensagem PATHTEAR para a velha rota.

6.3.4.7. Encerramento de um LSP-SVC

É usado o procedimento padrão do RSVP base, ou seja, as conexões são encerradas de forma implícita em função do não refrescamento dentro do tempo previsto, ou então, explicitamente através do uso das mensagens de desconexão. Esta proposta adiciona a possibilidade de encerramento em virtude do temporizador-de-suspensão ter esgotado seu tempo.

6.3.4.8. Transferência de Dados em um LSP-SVC

Durante a fase de transferência de dados, segue-se o procedimento padrão do RSVP, a saber: conforme programação dos temporizadores, é necessário enviar mensagens PATH e RESV, a título de refrescamento dos estados nos nós ao longo do caminho. Lembrando que, pode-se optar pelo aumento do tempo dos temporizadores e fazer uso das mensagens de desconexão para diminuir a sobrecarga do protocolo na rede.

Nesta fase de transferência de dados, é possível, conforme previsto nesta proposta do Serviço SVC, ocorrer a suspensão temporária de um determinado LSP, a fim de liberar recursos para LSP de maior prioridade, ou então, modificação nos parâmetros de tráfego acordados inicialmente, conforme detalhamento explicitado nas Seções 6.3.4.3 e 6.3.4.5, anteriores, respectivamente.

6.4. Especificação do Protocolo MPLSoLAN

Os ambientes das redes locais podem variar consideravelmente tendo em vista inúmeros fatores, como: fabricantes e modelos de equipamentos de rede, sistema operacional de rede empregado, entre outros. Assim, uma proposta de solução genérica de QoS deve levar em conta, *a priori*, as tecnologias padronizadas e mais comumente empregadas. Nesse sentido, constata-se que, atualmente, a tecnologia dominante nas redes locais, no que concerne a protocolos de comunicação é o TCP/IP (*Transmission Control Protocol* / *Internet Protocol*) sobre Ethernet, com uma forte presença do RSVP (*Resource Reservation Protocol*) como protocolo de reserva de recursos. Além disso, é inegável a tendência de aumento no uso das redes chaveadas. Essas redes podem, em geral, fazer uso de mecanismos de prioridade, como o IEEE 802.1p.

Isto posto e levando em conta ainda o fato de que o RSVP e os mecanismos de diferenciação de CoS são consideravelmente independentes das tecnologias de enlace, percebe-se a necessidade de definição de um esquema de mapeamento do RSVP e das classes de serviço nível 3 em tecnologias específicas da sub-rede de comunicação. Nesse sentido, muito trabalho tem sido desenvolvido, a exemplo do SBM (*Subnet Bandwidth Manager*) [Yavatkar00], que em cooperação com outros protocolos como IEEE 802.1p, Q e D, pretendem atingir o objetivo supra-citado.

Nesse contexto, a proposta aqui apresentada, denominada MPLSoLAN, tem por objetivo prover mecanismos para um maior controle de QoS na rede local, tanto no que diz respeito ao uso dos recursos, quanto no controle de admissão de pacotes de aplicações e usuários, através do mapeamento das classes de QoS nível três para o nível dois e meio, proporcionando, assim, as condições necessárias para que uma rede IP seja de fato uma rede multi-serviço. O mapeamento da QoS nível dois e meio para o nível dois por sua vez, é direto, uma vez que é sugerido, neste trabalho, o uso do mesmo esquema de prioridades preconizado pelo IEEE 802.1p.

Possibilitará, ainda, o estabelecimento de conexões discadas via uma UNI MPLS, a partir dos *hosts*, distribuindo, com isso os custos de processamento no AR devidos à classificação e marcação de pacotes.

6.4.1. Considerações Iniciais

O MPLSoLAN especifica um sub-conjunto de funcionalidades do MPLS relativas a: capacidades de inserção e remoção de rótulos, bem como propõe o uso do RSVP-SVC como método de sinalização e controle de admissão na rede local e, adicionalmente, também sugere mecanismos de controle de fluxo e tratamento de prioridade para a camada de enlace (Seção 6.6). O MPLSoLAN apresenta as seguintes características principais:

a) É um protocolo de nível 2,5, ou seja, um protocolo que deve operar acima do protocolo MAC da rede local, mas abaixo do IP, conforme apresentado na Figura 6.24;

b) Ele descreve a operação entre *hosts* e roteadores/switches nível-3 capazes de executar o RSVP-SVC, através de dispositivos nível 2, como *switches* 802.1p compatíveis, para suportar reserva de recursos da LAN para fluxos de dados habilitados pelo RSVP. Vale ressaltar que esta necessidade de implementar o RSVP não oferece maiores problemas, uma vez que sistemas operacionais amplamente difundidos como o Windows e o Linux, já o incluem nativamente. Portanto, acrescentar a extensão aqui proposta não deverá representar um grande problema;

c) Considera o uso de redes Ethernet chaveadas, como tecnologia dominante da sub-rede de comunicação. Nesse sentido, o MPLSoLAN deve suportar encapsulamento de camada dois para Ethernet. Contudo, nada impede que trabalhos futuros possam incluir o encapsulamento para outros protocolos.

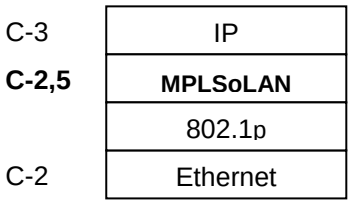

Fig. 6.24 – Contexto de Implementação do Protocolo MPLSoLAN

6.4.2. Descrição Funcional dos Componentes da Arquitetura MPLSoLAN

Além dos *hosts*, *switches* e roteadores, esta proposta prevê ainda a presença de um QoS-Server, que vem a ser um elemento lógico (módulo de *"software"*), capaz de operar como servidor/gerente de QoS em nível 2,5. A Figura 6.25 mostra os componentes da arquitetura MPLSoLAN:

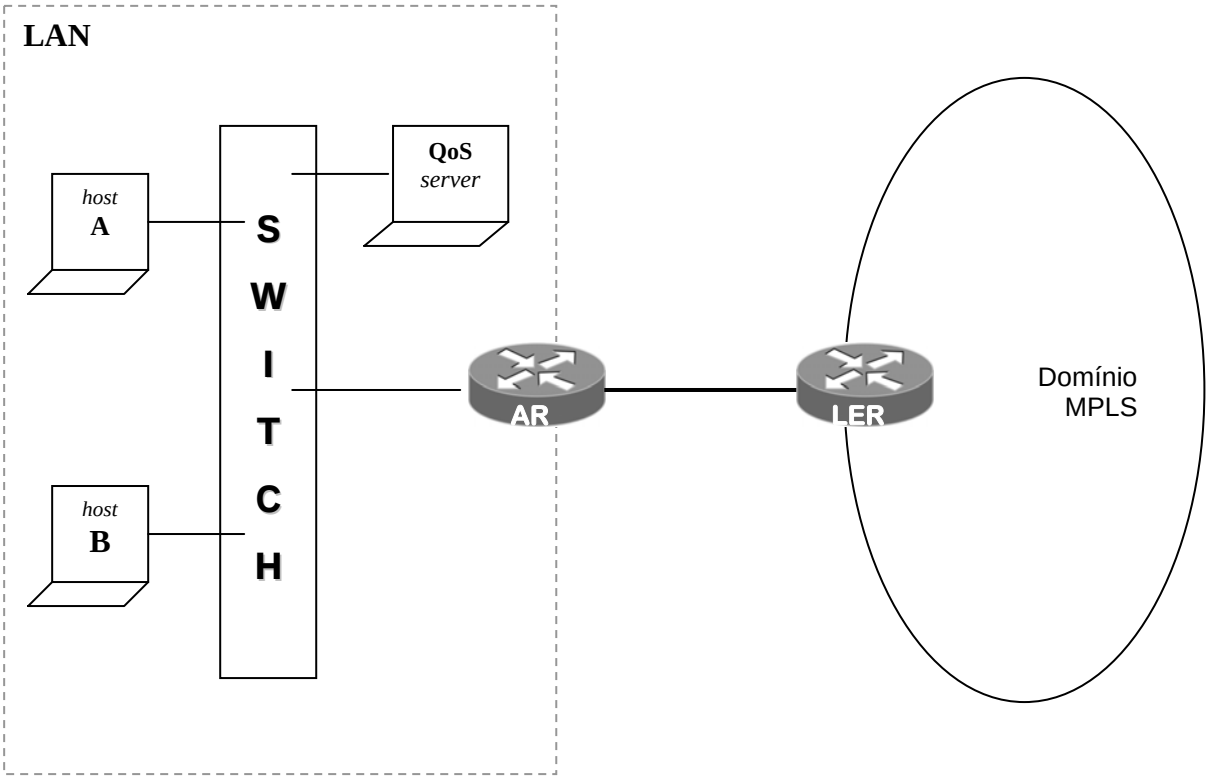

Fig. 6.25 – Componentes do MPLSoLAN

a) Hosts

Em cada *host*, todas as interfaces de rede devem ser configuradas com informações sobre o endereço do QoS-Server, que a título de sugestão pode vir a ser o próprio *gateway* da rede, ou seja, o roteador de acesso (AR).

Além disso, vale ressaltar que durante o processo de inicialização do *host*, ou quando da abertura de uma conexão que requeira níveis específicos de QoS, é necessário verificar a presença do QoS-Server e se o mesmo está ativo e disponível para o *host* solicitante, o que é feito mediante o uso do objeto <SVC_REQUEST> (ver Seção 6.3.4 para maiores informações sobre o objeto <SVC_REQUEST>). Uma vez comprovada a sua presença, os *hosts* podem se comunicar com o mesmo para

fins de controle de admissão de fluxos. Caso contrário, a rede deve seguir o seu comportamento padrão, sem utilizar as funcionalidades do MPLSoLAN.

Destaca-se ainda, que é na entrada da interface dos *hosts* onde devem ocorrer os processos de classificação e marcação; enquanto na saída é feito o escalonamento, conforme apresentado na Figura 6.26.

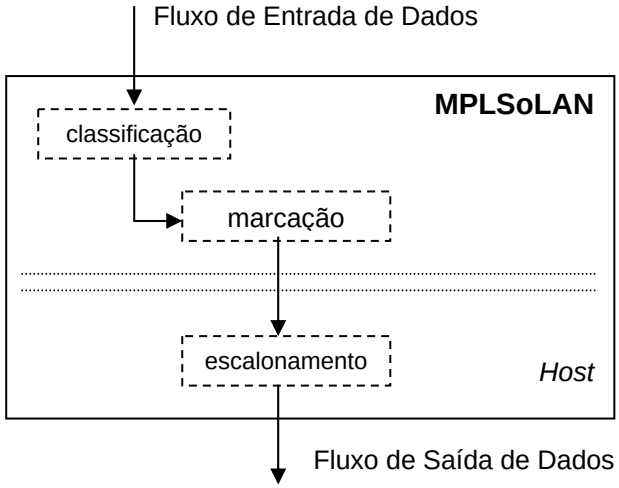

Fig. 6.26 – Procedimentos do MPLSoLAN

b) QoS-Server.

O QoS-Server se constitui num elemento importante nesse trabalho, uma vez que é a partir de suas atribuições que se garantirá um melhor uso dos recursos da rede. É através dele que efetivamente parte da banda pode ser reservada para certas classes de tráfego ou certos usuários em particular. Assim, um aspecto importante a ser considerado diz respeito à sua configuração, ou seja, é preciso informar *a priori* qual o limite de banda reservável para cada CoS, em cada segmento gerenciado sob seu controle. A princípio isto deve ser feito de forma estática, entretanto, mecanismos de configuração dinâmica também são possíveis, ainda que não abordados nesta proposta. Trabalhos futuros podem avaliar a aplicabilidade desse mecanismo, embora na Seção 6.6.2 já seja sugerido um mecanismo capaz de realizar essa configuração dinâmica de banda.

A utilização do QoS-Server proporciona a vantagem de centralizar as informações e requisições de serviços, sem contudo interferir na fase de transferência dos dados. De fato, ele desempenha um papel preponderante apenas nas fases de inicialização e encerramento das conexões. Opcionalmente, ele pode

monitorar a conexão, com o fim de coletar dados estatísticos relativos à real utilização dos recursos, durante a fase de transferência dos dados.

Quanto aos aspectos de gerenciamento, vale destacar que esta solução não implica numa perda do controle da rede, uma vez que apenas a função de classificação é deslocada para dentro da rede local.

Sugere-se que o QoS-Server tenha suas funcionalidades implementadas no roteador de acesso (*gateway*) da rede local, por ser este elemento o ponto de convergência de todo o tráfego da rede. De outra forma, seria acrescentado mais um nó no caminho, aumentando o custo de processamento e de tempo.

c) Roteador de Acesso

Sua operação se dá conforme as funções explicitadas na especificação do serviço SVC para uma UNI MPLS, apresentado na Seção 6.3, adicionado da capacidade para tratamento de requisições de conexões discadas a partir dos *hosts*.

Contudo, ressalta-se aqui, mais uma vez, a recomendação de que as funcionalidades do QoS-Server sejam implementadas no AR, pelas razões já explicitadas em seções anteriores. Além dessas, deve-se considerar a necessidade de haver uma comunicação permanente entre o AR e o QoS-Server, para que o QoS-Server tenha sempre atualizada uma base de dados dos serviços ativos, suspensos ou agendados. A possibilidade de agendamento, inclusive, é um serviço opcional que pode vir a ser uma característica interessante. Além disso, os dados mantidos pelo QoS-Server podem servir de repositório de estatísticas e *logs*, facilitando atividades relacionadas à contabilidade e monitoração dos recursos de rede.

Futuros trabalhos poderão abordar os aspectos de segurança e confiabilidade da informação, a fim de proteger o roteador de acesso de alterações indevidas por parte do usuário.

6.4.3. Funcionalidades do MPLSoLAN

O protocolo MPLSoLAN deve ser capaz de prover, no mínimo, as seguintes funções:

a) Inserir rótulos;
b) Extrair rótulos;

c) Prover mapeamento da QoS nível IP para o nível 2,5, isto é, camada MPLSoLAN, através da devida marcação dos bits do campo EXP, do cabeçalho MPLSoLAN, que é, em termos de formato, idêntico ao cabeçalho MPLS;
d) Atender pedidos de estabelecimento de conexões discadas provenientes de clientes MPLSoLAN (aplicações) dos *hosts*;
e) Negociar parâmetros de QoS entre *hosts* e o QoS-Server, o que é feito pelo Plano de Controle através do RSVP-SVC. O objetivo é negociar os parâmetros de QoS para um fluxo de dados específico, que leve em conta, total ou parcialmente, a tupla (porta de origem, porta de destino, protocolo (no caso o IPv4), endereço de origem, endereço de destino). Uma vez negociados os parâmetros de QoS entre o QoS-Server (AR) e a aplicação, o LER de entrada informaria ao AR o rótulo alocado, e este ao *host* (aplicação) chamador. Assim, os pacotes já sairiam classificados e marcados do *host* de origem, poupando o AR e o LER de entrada, de um domínio MPLS, de tal função;
f) Possibilitar a alteração "*on-the-fly*" de parâmetros de QoS. O QoS-Server, a partir de solicitações dos *hosts*, deve poder negociar com as aplicações originadoras do fluxo uma alteração dos parâmetros de QoS, seja para acomodar novos fluxos de maior prioridade, seja por motivos pró-ativos, no sentido de evitar congestionamentos na rede local, ou no(s) enlace(s) de saída.

O mapeamento da QoS nível 2,5 para o nível 2 (802.1p), quando possível, deve ser feito pelo protocolo da C-2. Trabalhos futuros podem contemplar o desenvolvimento de um módulo de mapeamento para o Ethernet.

6.4.4. Arquitetura do MPLSoLAN

Como apresentado no Capítulo 3, a arquitetura MPLS possui dois componentes básicos bem definidos: o plano de controle e o plano de dados. No caso do MPLSoLAN, como pretendemos dar aos *hosts* a capacidade de inserir e extrair rótulos, é proposto um esquema funcional mais resumido conforme apresentado na Figura 6.27. Notar que as atividades relativas à troca de rótulos são desnecessárias, bem como lidar com a possibilidade de pilha de rótulos, uma vez que o *host* desempenhará basicamente, apenas duas funções: inserir o primeiro rótulo da pilha, no caso de ser o *host* chamador, ou retirar o último rótulo, no caso de ser o *host* chamado.

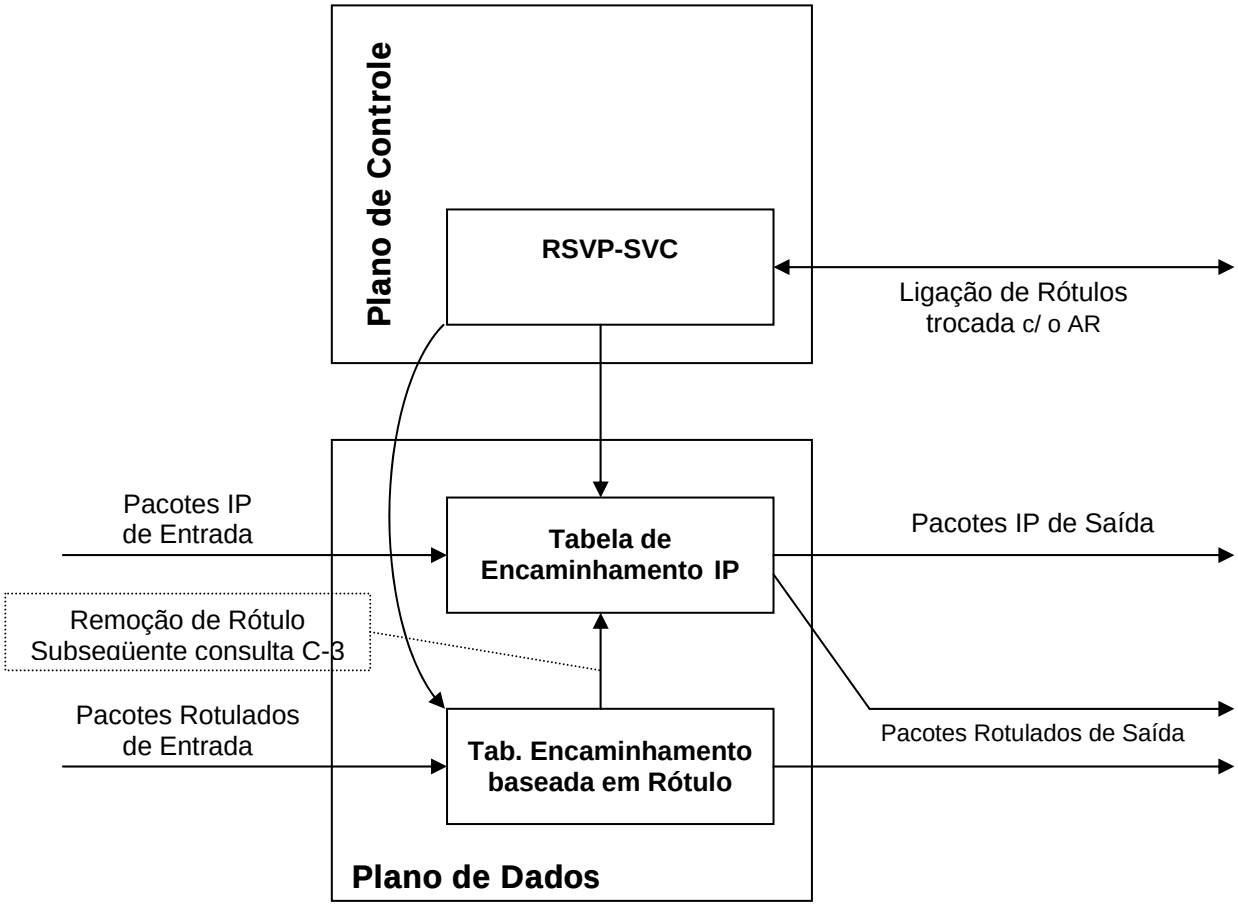

Fig. 6.27 – Arquitetura Básica do Nó MPLSoLAN

O Plano de Controle do MPLSoLAN deve ser capaz de criar as ligações entre os rótulos e as FEC e distribuir estas ligações ao seu par, o QoS-Server. Destaca-se ainda que, no caso do MPLSoLAN, essa atividade não está relacionada com os protocolos de roteamento, mas com o protocolo de sinalização RSVP-SVC, que foi especificado na Seção 6.3.

O Plano de Dados, por sua vez, é responsável pela manutenção de uma tabela de encaminhamento, e pela inserção do cabeçalho MPLS antes da transmissão do pacote. Sua operação pode ser caracterizada em dois momentos distintos: no *host* chamador e no *host* chamado.

a) Operação do MPLSoLAN no *host* chamador

Ao receber um pacote IP, a camada MPLSoLAN do *host* chamador classifica o pacote em uma FEC e o rotula com o rótulo de saída correspondente àquela FEC. Mapeia a QoS nível 3 para campo EXP, se pertinente e, finalmente,

encaminha o pacote para a camada inferior com o rótulo apropriado, segundo informações da tabela de encaminhamento baseada em rótulo.

Uma observação importante é que, no caso da chegada de um pacote IP que será transmitido de forma convencional, sem auxílio das funcionalidades do MPLSoLAN, ele é passado diretamente para a camada inferior sem alterações.

b) Operação do MPLSoLAN no *host* chamado

O *host* chamado recebe o pacote rotulado da C-2, executa uma consulta-de-rótulo, remove o cabeçalho MPLSoLAN e o encaminha para C-3.

6.4.5. *Modus Operandi*

A proposta básica aqui apresentada é que, uma vez negociados os parâmetros de QoS entre a aplicação e o QoS-Server, e estabelecida a conexão discada entre *host* chamador (fonte) e o *host* chamado (destino), os pacotes já partam do *host* chamador do fluxo: classificados, marcados, rotulados e com tamanhos adequados, de forma a não prejudicar um tráfego sensível à QoS, como VoIP, por exemplo. Vale lembrar que, nesta proposta, tanto a classificação multi-campo quanto o controle de fluxo são feitos diretamente na origem, minimizando ao longo do caminho os problemas dele decorrentes, de forma a não comprometer outros fluxos de tráfegos.

É importante destacar ainda que, uma vez que os fluxos de tráfego utilizam o procedimento de controle de admissão baseado no RSVP-SVC (Plano de Controle do MPLSoLAN) para requisitar reserva de recursos antes de enviar qualquer tráfego, esse mecanismo de controle irá restringir a quantidade total de tráfego gerado pelos fluxos que necessitam de QoS, dentro de limites desejados pelo administrador. Além disso, como o tráfego melhor-esforço gerado pelas outras aplicações é *rate-adaptive*, este irá se adaptar de forma a acomodar-se dentro da banda disponível restante.

6.4.5.1. Estabelecimento da Conexão Discada a partir do *host* Chamador

Quando um *host* chamador A desejar abrir uma conexão com um *host* chamado X de uma outra rede, com determinados parâmetros de QoS, ele deve enviar uma solicitação de recursos ao QoS-Server local (AR), conforme preconizado pelo RSVP-SVC. Este procedimento de controle de admissão de conexão na rede local constitui-se na primeira linha de defesa para a rede proteger a si mesma de

uma carga excessiva [Stallings92]. Deste modo, quando um usuário/aplicação requisitar um novo serviço (LSP), faz-se mister a especificação implícita ou explicita das características de tráfego para esta conexão, além de sua característica de uni ou bi-direcionalidade, dependendo da aplicação. A seleção dessas características de tráfego deve ser feita através da escolha de uma das CoS que a rede é capaz de prover. Naturalmente, a rede só aceitará a conexão, se ela puder atender o nível solicitado de tráfego, enquanto mantém a QoS das conexões existentes.

No intuito de requisitar uma reserva de recursos, os clientes MPLSoLAN devem observar as seguintes premissas:

1º) Quando um cliente MPLSoLAN envia uma mensagem RSVP PATH sobre uma interface ligada a um segmento gerenciável (MPLSoLAN *enabled*), ele a envia para o QoS-Server ao invés de fazê-lo para o endereço de destino da sessão RSVP;

2º) O processamento no QoS-Server pode implicar em uma atualização do Adspec (que é usado com o serviço opcional OPWA – *One Pass With Advertisements*), e, com certeza, a construção e manutenção de um estado de PATH para a sessão e o registro do nó C2-C3 que enviou a mensagem PATH;

3º) Após o processamento do pedido de reserva de recursos, o QoS-Server (AR) encaminha o pedido para o LER;

4º) Após a devida negociação de parâmetros de tráfego e as devidas reservas de recursos feitas ao longo do caminho, o *host* chamador A terá condições de fazer a classificação e a marcação do campo EXP, que irá refletir o mapeamento da QoS C-3 na QoS C-2,5.

Vale lembrar ainda que os LSP são controlados de modo distribuído, ou seja, cada nó negocia um rótulo para cada FEC com seu vizinho posterior e anterior ao longo do caminho. Por padrão, o nó anterior aloca um rótulo para uma FEC e informa seu par posterior. Este procedimento de distribuição de rótulos é executado pelo RSVP-SVC. Como resultado, cada roteador constrói uma tabela de informação de rótulos, que mapeia a relação entre o rótulo específico de enlace de cada LSP e a FEC correspondente. Vale ressaltar que sempre que ocorre uma mudança na tabela de informação de encaminhamento, o cliente MPLSoLAN renegocia a ligação rótulo-FEC e atualiza a tabela de rótulos.

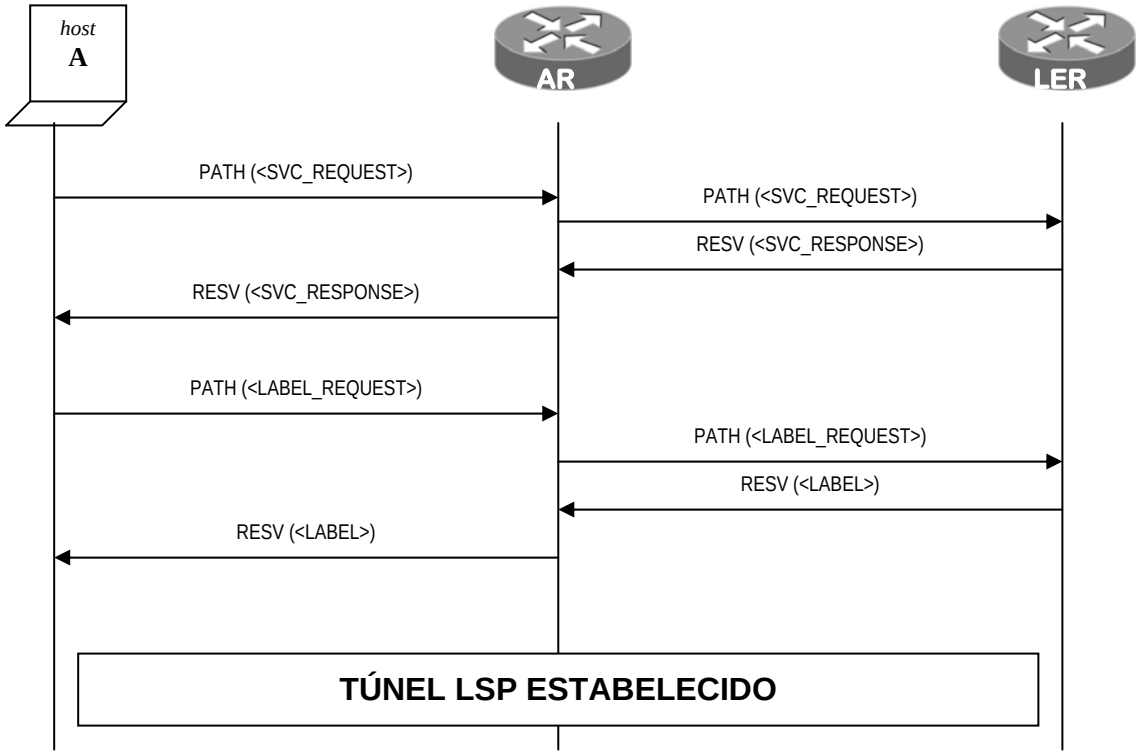

Fig. 6.28 – Estabelecimento com Sucesso de um Túnel LSP através de uma Conexão Discada

Uma vez requerido o serviço, o QoS-Server responderá ao *host* chamador com um <SVC_RESPONSE> ou PATHERR. No caso de uma resposta negativa, o *host* chamador A deverá tentar em outra oportunidade conforme configuração da aplicação cliente, pois não deve haver, no momento, recursos suficientes na rede local e/ou de acesso para atender à sua demanda.

No caso de ser possível o estabelecimento de uma conexão discada, conforme apresentado na Figura 6.28, significa que existem recursos disponíveis na rede local para atender o nível de QoS exigido e que o QoS-Server (AR) negociará com o LER a abertura de uma conexão do tipo SVC ou a utilização de uma PVC já existente e que se adeque às exigências de QoS especificadas pela aplicação em execução no *host* chamador A. Se a operação for realizada com sucesso, o QoS-Server transforma os descritores de tráfego RSVP nos parâmetros correspondentes da camada de enlace e aloca o rótulo antes mensagem ser propagada para o *host* chamador A.

Se não for possível estabelecer o serviço SVC ou não houver uma PVC disponível, então o QoS-Server informa ao *host* chamador A da impossibilidade de atendê-lo, conforme mostrado na Figura 6.29.

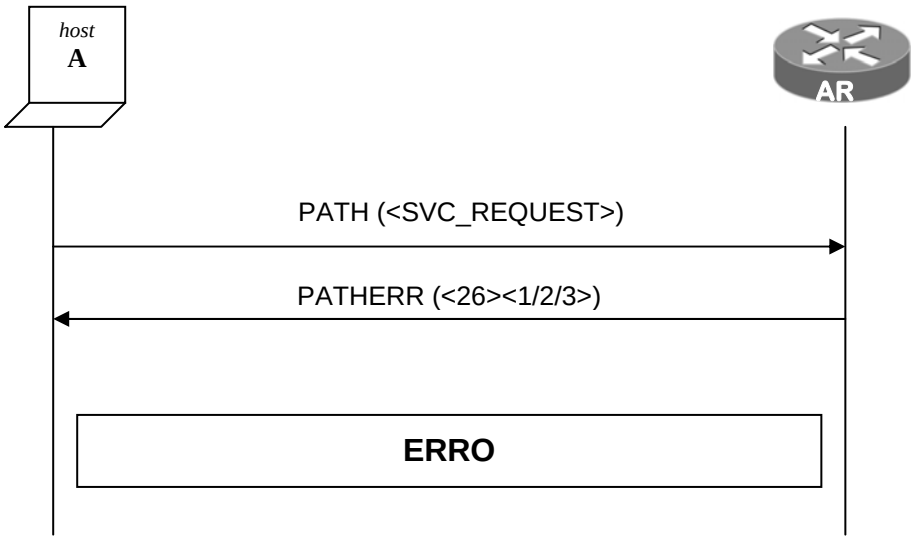

Fig. 6.29 – Erro na Chamada Discada

Vale ressaltar a importância da implementação de um sistema de filas no *host* chamador, que realize um controle de fluxo na origem, a fim de garantir o uso acordado dos recursos da rede para cada aplicação. Trabalhos futuros poderão avaliar qual o melhor algoritmo a ser implementado para o controle dessas filas, dentre as possibilidades mais utilizadas atualmente, incluindo PQ (*Priority Queueing*), WFQ (*Weighted Fair Queueing*), Round-Robin, entre outros. Como ponto de partida, sugere-se o uso do esquema apresentado na Seção 6.6.1.

6.4.5.2. Outros Procedimentos

Vale observar que as funcionalidades relativas a inserção (*push*) e extração (*pop*) de rótulos, assim como a funcionalidade relativa ao mapeamento da QoS nível IP para o nível 2,5 (marcação nos bits EXP, por exemplo), são funcionalidades existentes no padrão MPLS e, portanto, representam o sub-conjunto de funcionalidades a serem implementadas pelo MPLSoLAN.

As demais funcionalidades, como negociar parâmetros de QoS entre *hosts* e o QoS-Server e possibilitar a alteração *on-the-fly* dos parâmetros de QoS, são providas pelo RSVP-SVC, que é o protocolo de sinalização adotado pelo MPLS Total.

6.5. Aspectos Relacionados à Compatibilidade com o RSVP-TE

6.5.1. Túneis LSP e Túneis TE

Mantém-se as definições de Túneis LSP e Túneis TE apresentadas em [Awduche01]. Assim, o rótulo define o fluxo, e o respectivo LSP é chamado de

Túnel LSP (LSP_TUNNEL). O Túnel TE por sua vez é um conjunto de um ou mais Túneis LSP. Nesse contexto, a fim de permitir a associação (agregação) de conjuntos de Túneis LSP, dois identificadores são transportados: um identificador de túnel (Tunnel_ID) e o identificador de LSP (LSP_ID). O Tunnel_ID é parte do objeto SESSION, e define unicamente um Túnel TE. Os objetos SENDER_TEMPLATE e FILTER_SPEC transportam um LSP_ID. Assim, o objeto SENDER_TEMPLATE (ou FILTER_SPEC) juntamente com o objeto SESSION define unicamente um Túnel LSP.

Esta característica é melhor aproveitada em ambientes *multi-link*, devido a sua utilidade em operações de re-roteamento e em operações de distribuição de carga. No contexto do SVC, como proposto neste trabalho, o uso do LSP_ID é suficiente para identificar inequivocamente os LSP e realizar os procedimentos necessários de encaminhamento de dados.

6.5.2. Operação dos Túneis LSP

Todas as capacidades definidas na RFC 3209 [Awduche01] foram mantidas, a saber:

- Capacidade de Estabelecer Túneis LSP com ou sem exigências de QoS;
- Capacidade de Re-rotear dinamicamente um Túnel LSP já estabelecido;
- Capacidade de Observar a rota real seguida por um Túnel LSP estabelecido;
- Capacidade de Identificar e Diagnosticar Túneis LSP;
- Capacidade de executar a preempção de um Túnel LSP através de um controle de policiamento administrativo;
- Capacidade de Executar alocação, distribuição e ligação de rótulo via método DOD.

6.5.3. Estilos de Reserva

Mantém-se as definições de [Awduche01], ou seja, todos os estilos são permitidos, a exceção do WF (*Wildcard Filter*). A possibilidade de diferentes estilos de reserva é uma característica interessante, pois dependendo do estilo adotado, uma sessão RSVP pode resultar em um ou mais LSP, provendo assim uma boa flexibilidade na definição dos referidos caminhos chaveados por rótulos.

Os estilos descritos a seguir foram definidos em [Awduche01]:

a) Estilo *Fixed Filter* (FF)

O estilo de reserva FF cria uma reserva distinta para cada emissor, que não é compartilhada por outros emissores. É um estilo comum para aplicações cujos

tráfegos oriundos de diversos emissores são concorrentes e independentes, como Videoconferência sobre IP, por exemplo. Mantém-se esse estilo, uma vez que sua aplicabilidade é evidente em ambientes de comunicação multimídia e não existe nenhuma incompatibilidade com o uso do RSVP-SVC.

b) Estilo *Wildcard Filter* (WF)

O estilo WF preconiza o uso compartilhado de uma reserva única para todos os emissores de uma sessão. Esse estilo não é utilizado neste trabalho em função da sua não aplicabilidade à TE, pois as regras de mesclagem ("*merging*") do WF impedem o uso dos objetos <EXPLICIT_ROUTE>.

c) Estilo *Shared Explicit* (SE)

Este estilo permite que um receptor especifique explicitamente que emissores poderão compartilhar uma mesma reserva, a exemplo do que acontece numa áudio-conferência usando VoIP, uma vez que os vários emissores não "falam" ao mesmo tempo e assim podem compartilhar de uma única reserva. É necessário manter este estilo em função da possibilidade de re-roteamento, uma vez que o RSVP propicia a solução deste problema de forma elegante, através da combinação do objeto <LSP_TUNNEL_SESSION> e o estilo de reserva SE.

6.5.4. Capacidade de re-Roteamento de Túneis TE

A capacidade de re-rotear túneis já estabelecidos em função de certas condições, geralmente baseadas em políticas administrativas, é uma exigência para a Engenharia de Tráfego. Nesse sentido, o RSVP-TE provê mecanismos que atendem esta exigência, porém, a sua aplicabilidade é restrita no contexto do Serviço SVC, uma vez que, no contexto deste trabalho, assume-se o pressuposto de uma ligação *monolink* entre o AR e o LER.

Todavia, como dito anteriormente, vale destacar que trabalhos futuros poderão analisar esta característica, a fim de atender casos em que existam mais de um enlace fazendo a ligação AR-LER.

6.5.5. Objetos Relacionados ao Túnel LSP

As observações quanto aos rótulos ATM e FR não são consideradas. Trabalhos futuros poderão incluir definições sobre como codificá-los. Ademais, outras observações são destacadas a seguir.

6.5.5.1. Objeto <EXPLICIT_ROUTE>

Este objeto é aplicável com restrição, pois uma vez que o AR não pode interferir no *backbone* do ISP, sendo necessário uma negociação preliminar. Assim, após uma confirmação junto ao LER da possibilidade de usar determinada rota, o AR tem direito a usar o objeto ERO. Nesse sentido, dois novos objetos são definidos: o <LSP_ERO_REQUEST> e o <LSP_ERO_RESPONSE>, cujos formatos e mecanismos de funcionamentos são descritos na Seção 6.3.4.5.

6.5.5.2. Objeto <RECORD_ROUTE>

O RRO (<RECORD_ROUTE> OBJECT) é um objeto que grava rotas e, opcionalmente, grava rótulos.

Quanto à sua aplicabilidade, em [Awduche01] são apresentados três usos possíveis desse objeto, a saber:

- Pode funcionar como um mecanismo de detecção de *loop* a fim de descobrir *loops* no roteamento de nível 3;
- Pode coletar dados atualizados das sessões RSVP, no que concerne a informação de caminhos *hop-by-hop*; e,
- Com mudanças mínimas em sua sintaxe, ele pode ser usado como entrada para o objeto <EXPLICIT-ROUTE>.

Entretanto, apenas os dois últimos casos necessitam ser mantidos no contexto do serviço SVC, pois o seu uso como mecanismo de detecção de *loop* C-3 é desnecessário, já que a ligação direta entre o AR e o LER é *monolink*.

6.5.6. Extensão do *Hello*

Permanece como descrito em [Awduche01], em função da sua utilidade quanto à detecção de falhas de nó (no AR e/ou no LER). Tanto o formato da mensagem HELLO quanto os formatos do objeto <HELLO>, assim como o seu uso, não necessitam de modificação alguma em relação ao Serviço SVC. Contudo, uma restrição é feita quanto à possibilidade de *multi-link*. Neste caso, a referida seção não é aplicável, uma vez que se parte do pressuposto da existência de apenas uma ligação entre o AR e o LER.

6.5.7. Valores de Códigos e Sub-Códigos de Erro Globalmente Definidos

Na Tabela 6.7 é apresentado um resumo dos códigos e sub-códigos de erro de valor global. Os novos códigos e sub-códigos definidos nesse trabalho estão agrupados no valor **<26>**, denominado “Erro de SVC”.

CÓDIGOS DE ERRO	**SIGNIFICADO**	
24	***Routing Problem***	
	Valores de sub-código de erro, globalmente definidos:	
	Valor	**Significado**
	1	*Bad EXPLICIT_ROUTE object*
	2	*Bad strict node*
	3	*Bad loose node*
	4	*Bad initial subobject*
	5	*No route available toward destination*
	6	*Unacceptable label value*
	7	*RRO indicated routing loops*
	8	*MPLS being negotiated, but a non-RSVP-capable router stands in the path*
	9	*MPLS label allocation failure*
	10	*Unsupported L3PID*
25	***Notify Error***	
	Valores de sub-código de erro, globalmente definidos:	
	Valor	**Significado**
	1	*RRO too large for MTU*
	2	*RRO Notification*
	3	*Tunnel locally repaired*
26	***Erro de SVC***	
	Valores de sub-código de erro, globalmente definidos:	
	Valor	**Significado**
	1	*Serviço SVC não-operacional*
	2	*Serviço SVC não disponível para este assinante*
	3	*Valores de QoS inaceitáveis*
	4	*Valor de LSP_ID desconhecido ou inexistente*
	5	*Rede sem recursos no momento*

Tabela 6.7 – Códigos e Sub-Códigos de Erro

6.6. Aspectos co-Relacionados ao MPLS Fim-a-Fim

Esses aspectos não se referem diretamente à especificação dos serviços e protocolos, estando fora do escopo principal do trabalho. Contudo, são apresentadas nas próximas sub-seções, primeiras considerações a seu respeito.

6.6.1. Aspectos Relativos ao Gerenciamento dos Recursos de Rede

Os recursos de rede são alocados de forma que os fluxos de tráfego sejam separados de acordo com as características de serviço. Nesse sentido, as FECs provêem um meio conveniente e flexível de se estabelecer esses agrupamentos.

Considerando que a arquitetura MPLS define um campo EXP, de 3 bits, verifica-se facilmente que é possível definir até 8 classes de serviço, o que pode ser traduzido na implementação de 8 filas por interface de saída.

Uma vez identificado a que CoS o fluxo de dados pertence, os pacotes são encaminhados para a sua fila específica, conforme apresentado na Figura 6.30. Os LSP de determinada CoS serão policiados e moldados através dos conhecidos algoritmos *Token Bucket*, para policiamento, e o *Leaky Bucket* para a moldagem.

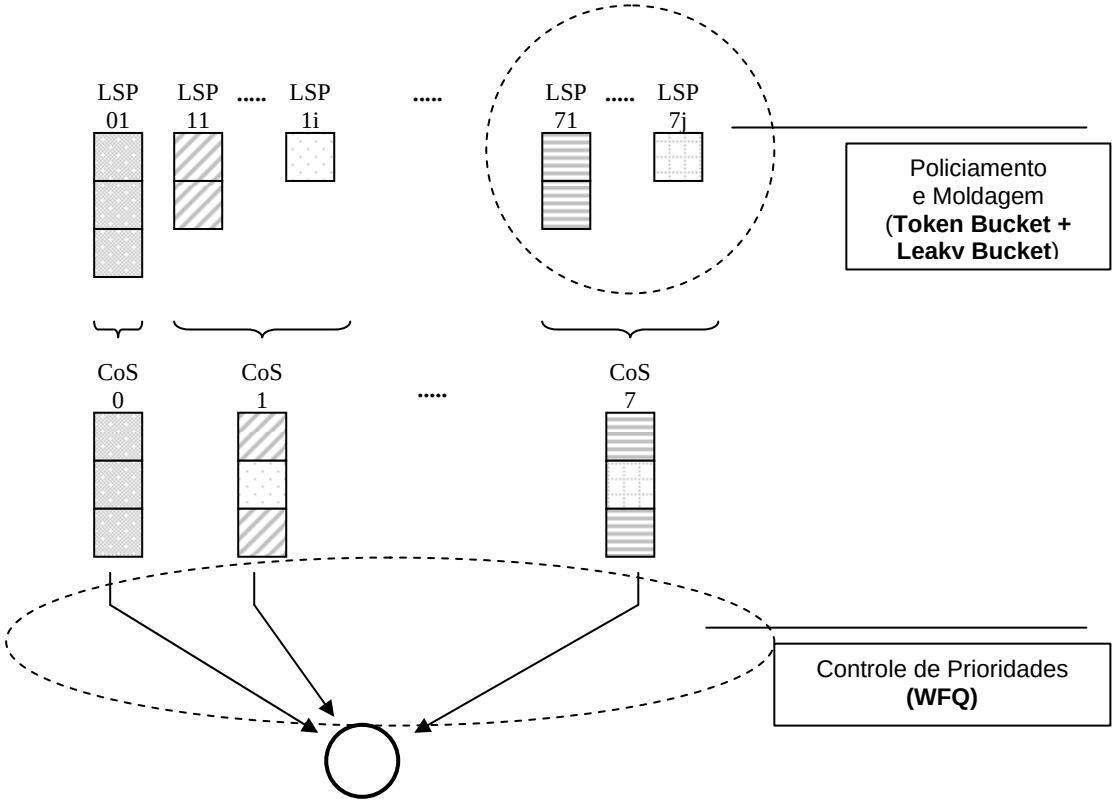

Fig. 6.30 – Enfileiramento de Pacotes Egressos por CoS

Vale lembrar que o policiamento do tráfego é necessário, pois garante que o tráfego oriundo das aplicações está em conformidade com o SLA, além de prover informações que servirão à contabilidade. A moldagem de tráfego, entretanto diz

respeito ao processo de se adequar o fluxo de transmissão aos requisitos definidos pelo SLA.

Soluções como a apresentada acima apresentam como vantagem o provimento à rede IP de características como previsibilidade e determinismo, além de ser uma solução adequada quanto às políticas de gerenciamento.

É importante destacar ainda que os parâmetros de QoS primariamente relacionados com o gerenciamento de recursos da rede são: perda média de pacotes, atraso de transferência de pacote e *jitter* [Stallings92], os quais são afetados pela quantidade de recursos dedicada ao LSP pela rede.

O controle de prioridade por seu turno é necessário quando da ocorrência de descarte de pacotes. Uma vez que a rede está sobrecarregada e que, necessariamente terá que descartar pacotes, então, deve-se dar preferência ao descarte de pacotes de baixa prioridade, a fim de proteger os LSP cujos fluxos de dados são de prioridade maior. Este controle é feito através de filas.

Vários métodos podem ser utilizados para priorizar o tráfego nas interfaces de saída, tais como: *Service Scheduling*, *Weighted Fair Queueing* (WFQ), entre outros.

6.6.2. Aspectos Relativos à Alocação de Banda

Quanto às questões relacionadas à alocação de banda, as seguintes premissas devem ser levadas em consideração: a solução deve ser de fácil gerenciamento e escalável.

Nesse sentido, a banda é reservada não para fluxos em particular, mas para os diferentes conjuntos de Classes de Serviço. Além disso, o procedimento de Controle de Admissão é aplicado em bases *per-classe,* levando em conta, ainda, o originador do fluxo *host*/aplicação/usuário dentro de cada classe. Com isso, obtém-se a vantagem de um melhor controle da vazão e da latência em bases *per host*/aplicação/usuário.

Vale destacar ainda que as seguintes condições devem ser cuidadosamente observadas quando da definição dos critérios sobre alocação de banda:

a) Garantir uma percentual mínimo para cada CoS, em condições de uso pleno;

b) Se a banda estiver sobrando, na medida em que for necessário alocar o que puder para cada CoS;

c) Não se deve reservar todos os recursos de rede para a classe de maior prioridade, tais como aquelas que atendem a tráfego de voz e vídeo interativos, pois faltariam recursos para as demais;

d) Um novo LSP só pode ser estabelecido se a banda residual dentro de sua CoS for maior que a banda requisitada.

As reservas de banda para cada CoS tanto podem ser realizadas estaticamente, através da intervenção do operador, quanto dinamicamente através de programação. Neste último caso, uma outra possibilidade interessante se refere à utilização de uma rede neural, que fosse capaz de modelar o perfil de tráfego na rede e, a partir desse aprendizado, definir os percentuais de alocação de banda para cada CoS. Trabalhos futuros podem considerar essa questão.

Sugere-se o uso de um mecanismo como o apresentado na Figura 6.31, que atende às premissas supracitadas.

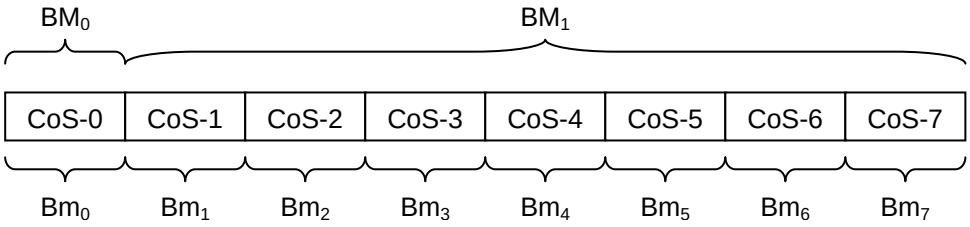

Fig. 6.31 – Esquema Sugestivo de Alocação de Banda

Onde:

- Bm_i = Banda Mínima da CoS_i
- BM = Banda Máxima
- $BM_0 = Bm_0$ = BE → Reserva estrita :: Tráfego *Best Effort*
- Bu_i = Banda em Uso da CoS_i
- $BM_j = 1 - BE - \sum_{i=1}^{7} Bu_i$

Capítulo 7
Verificação e Simulação do RSVP-SVC

O propósito deste capítulo é descrever as principais atividades realizadas em cada estágio de desenvolvimento da especificação, a saber: edição, compilação e simulação.

Este capítulo está organizado da seguinte forma: após uma breve introdução na Seção 7.1, onde é descrito o ambiente de *hardware* e *software* sobre os quais a especificação foi editada e validada, segue-se uma descrição de cada etapa de desenvolvimento. Na Seção 7.2 são mostradas algumas telas do ambiente de edição utilizado e tecidos comentários gerais sobre a ferramenta. Depois, na Seção 7.3, comenta-se sobre o processo de compilação, com destaque para os arquivos gerados. O cenário de simulação está contido na Seção 7.4, que apresenta as seguintes informações: as instâncias criadas, os estados definidos por corpo de módulo (*body*) e as seqüências de testes verificadas, inclusive com a apresentação de diagramas de seqüência que mostram o relacionamento das primitivas de serviço suportadas. Por fim, na Seção 7.5 são feitos comentários sobre as propriedades do sistema que foram verificadas.

7.1. Introdução

O propósito de uma simulação é avaliar, sobretudo, a corretude de uma especificação. Nesse processo, dispor de máquinas robustas e bons programas simuladores é fator por demais importante. Neste trabalho foi possível a utilização de uma máquina padrão PC, processador Pentium 4 3.0 GHz, com 512Mb de memória, com Sistema Operacional Linux Debian Sarge versão 3.01, *kernel* 2.6.8-2.

A ferramenta utilizada em todas as fases de desenvolvimento foi o EDT (*Estelle Development Toolset*), que se encontra atualmente na versão 4.3 e consiste de um conjunto de ferramentas para especificação e análise de sistemas de comunicação complexos, usando a técnica de descrição formal Estelle (vide Seção 5.4 para mais informações).

7.2. Edição

O RSVP-SVC foi editado através do Editor Gráfico orientado à Estelle, ferramenta integrante do EDT (vide Seção 5.4.9). A concepção dessa ferramenta é bastante amigável e intuitiva. Ela armazena a especificação em um formato próprio (arquivos de extensão .GSTL) e disponibiliza a opção de gerar uma especificação equivalente em formato texto (arquivos de extensão .stl). Um aspecto negativo é que ainda não se dispõe de uma opção para, a partir de um arquivo fonte (extensão .stl), gerar o arquivo em formato gráfico. Contudo, se houver um arquivo em sua forma intermediária (extensão .if), que foi traduzido pela ferramenta, ele pode fazer essa importação.

A seguir algumas telas que ilustram o funcionamento desse editor: a Figura 7.1 exibe uma visão da tela inicial do editor; a Figura 7.2 mostra um exemplo de uso das caixas de diálogo para entrada de parâmetros de configuração, no caso, uma caixa de diálogo para edição dos canais e outra para edição das interações; a Figura 7.3 mostra as opções disponibilizadas no editor de transições para a modelagem do comportamento de um dado corpo de módulo; e, a Figura 7.4, exibe uma visão da máquina de estados e transições modelada no editor de transições.

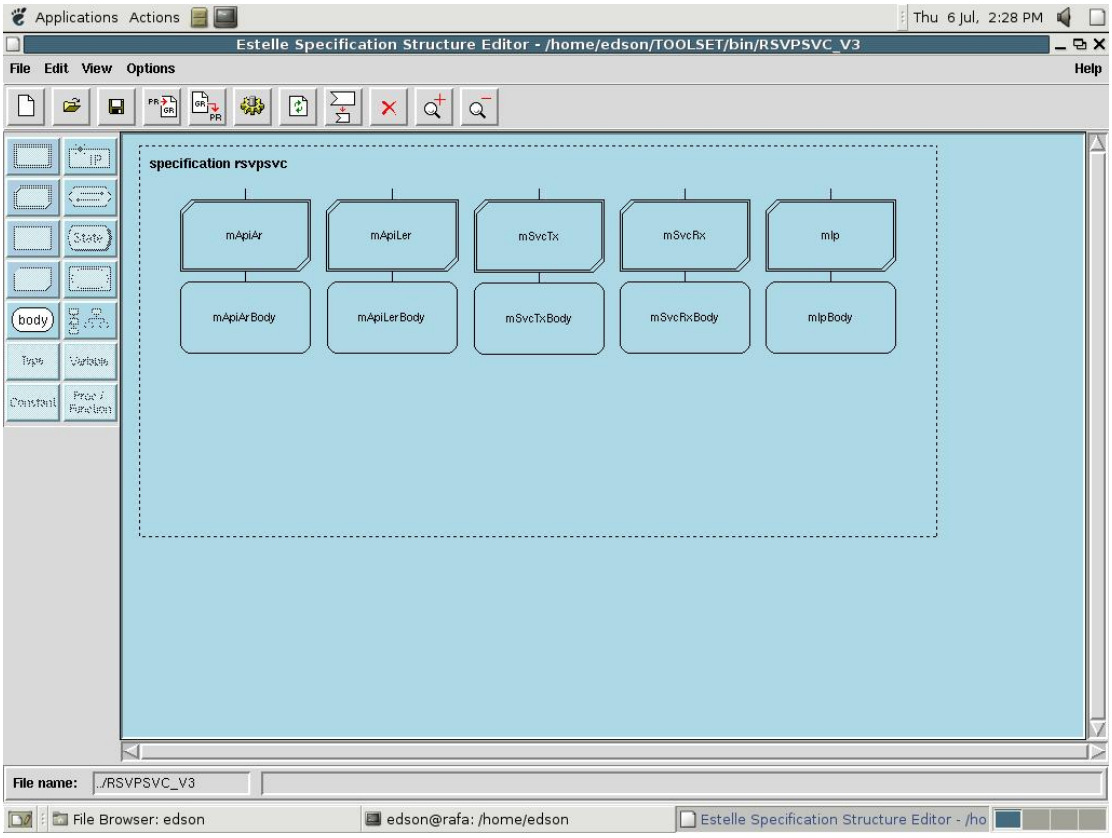

Fig. 7.1 – Tela Inicial do Editor Gráfico orientado a Estelle

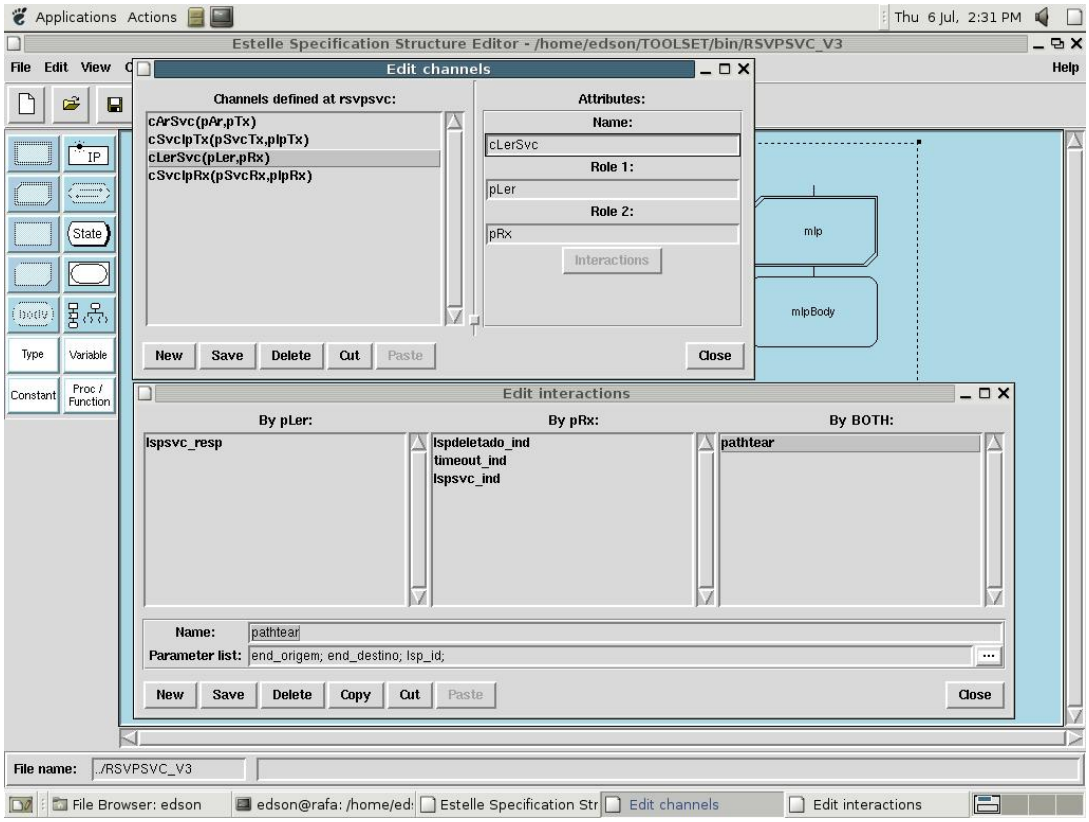

Fig. 7.2 – Caixas de Diálogo para Edição de Canais e Interações

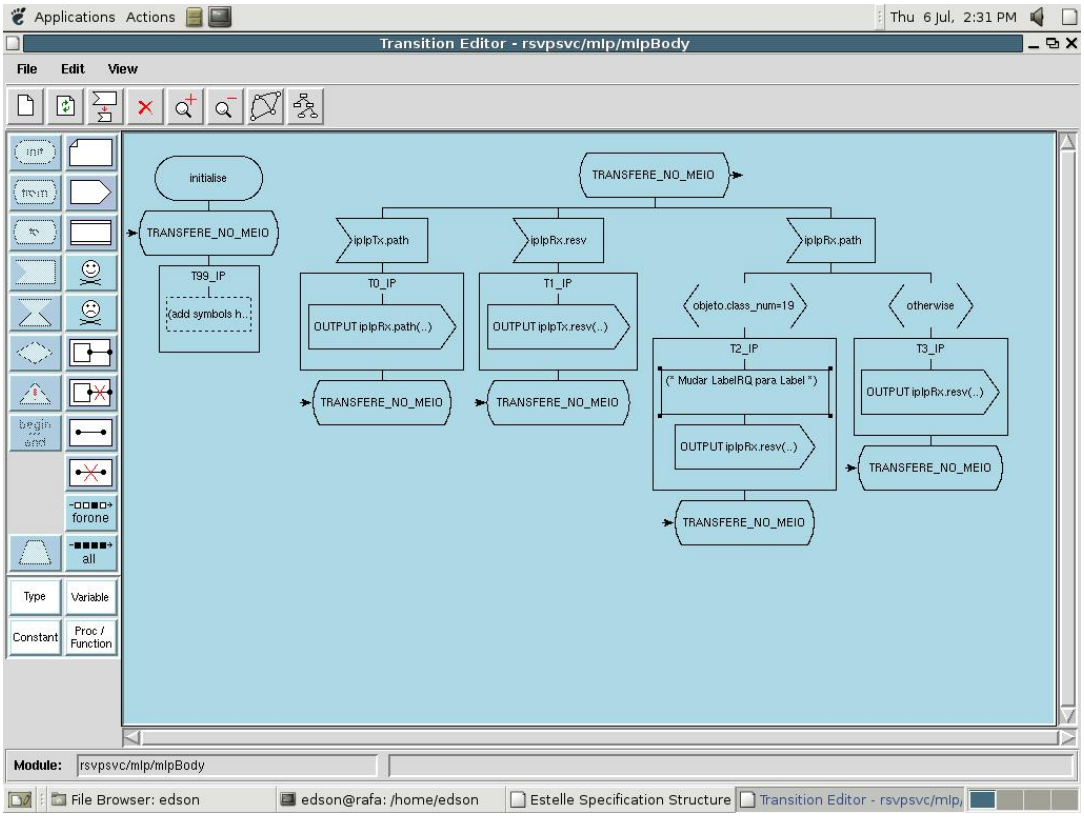

Fig. 7.3 – Editor de Transições

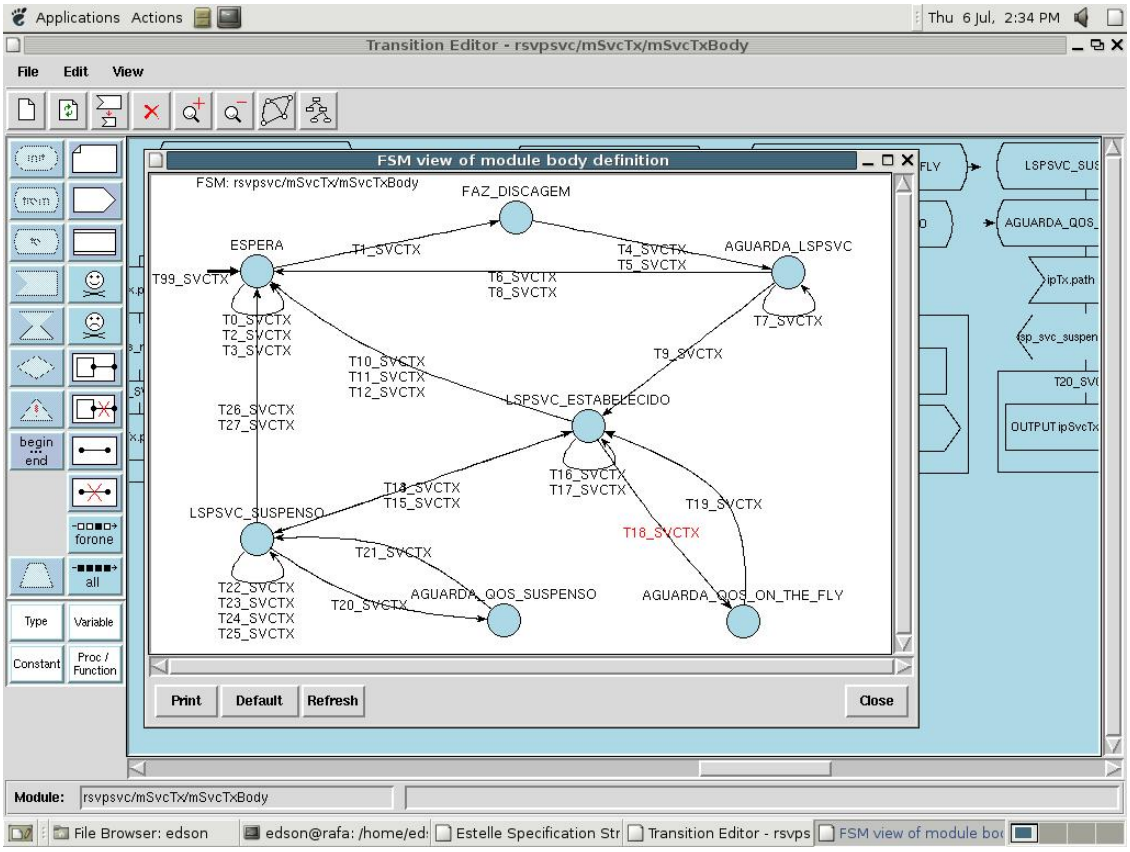

Fig. 7.4 – Visão da Máquina de Estado Modelada no Editor de Transições

7.3. Compilação

Uma vez editada a especificação e gerado o arquivo .stl, pode-se proceder à compilação, a fim de se verificar as ocorrências de erros léxicos, sintáticos e/ou semânticos. Uma vez que a especificação tenha sido completamente compilada sem erros, o compilador gera arquivos em código fonte C (extensão .c) e os arquivos de cabeçalho (extensão .h), para cada corpo (*body*) de módulo presente na especificação. O compilador gera ainda um arquivo com o nome da especificação em sua forma intermediária (extensão .if).

A Figura 7.5 mostra a tela de saída do compilador, no caso, ela indica que o arquivo “rsvpsvc_v3.stl”, que é a especificação proposta nesse trabalho, foi compilado sem erros.

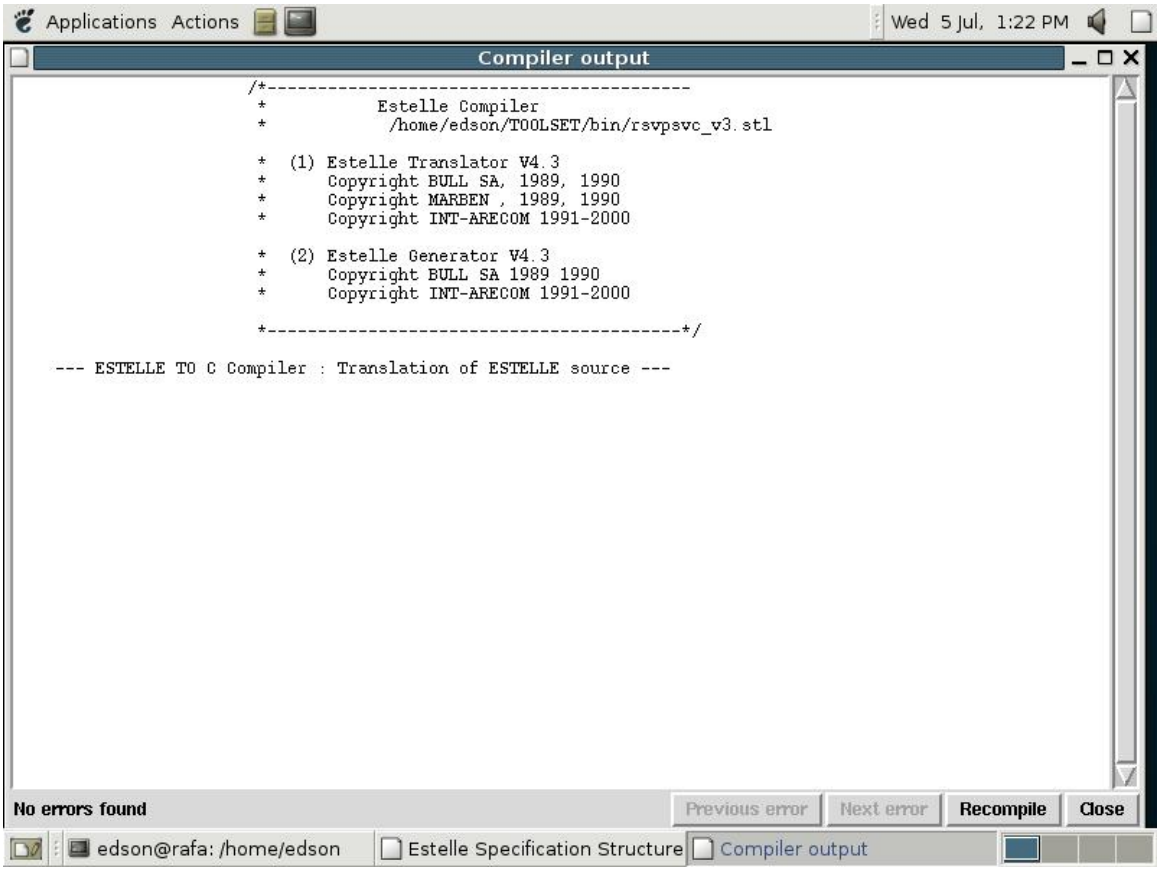

Fig. 7.5 – Saída do Compilador Indicando: Nenhum Erro Encontrado no Arquivo rsvpsvc_v3.stl

7.4. Cenário de Simulação

Visto a impossibilidade de se verificar completamente o sistema, optou-se pela escolha de algumas seqüências, consideradas mais importantes, a fim de proceder com a validação da especificação. Essas seqüências traduzem algumas propriedades da especificação, de acordo com a classificação apresentada na Seção 5.4. As seguintes seqüências de testes foram verificadas e validadas através de simulação: estabelecimento de uma conexão discada sem erro; tentativa de estabelecimento de uma conexão discada com erro devido a serviço SVC não-operacional; tentativa de estabelecimento de uma conexão discada com erro devido a SVC não disponível para assinante; suspensão de um LSP do tipo SVC; re-ativação de um LSP-SVC suspenso e encerramento normal de uma conexão discada.

Nas sub-seções a seguir, todas essas seqüências são apresentadas. Para facilitar o acompanhamento dos diagramas que apresentam a execução das seqüências mencionadas, sumariza-se nas tabelas a seguir as instâncias dos

módulos constantes da especificação (Tabela 7.1) criadas para a simulação assim como os estados componentes de cada módulo instanciado (Tabelas 7.2 a 7.6).

INSTÂNCIA	NOME / (TIPO DA INSTÂNCIA)	ESTADO INICIAL
	DESCRIÇÃO DA INSTÂNCIA	
1	specification-RSVPSVC ()	-
	Módulo principal. A especificação propriamente dita.	
2	MAPIAR-MAPIARBODY (*systemprocess*)	ESPERA
	Módulo que simula possíveis ações do iniciador da chamada como, por exemplo: fazer uma discagem, suspender um LSP, entre outras.	
3	MAPILER-MAPILERBODY (*systemprocess*)	ESPERA
	Módulo que simula possíveis ações do respondedor da chamada.	
4	MSVCTX-MSVCTXBODY (*systemprocess*)	ESPERA
	Módulo que especifica o comportamento do protocolo RSVP-SVC no lado do transmissor (Tx), iniciador da chamada.	
5	MSVCRX-MSVCRXBODY (*systemprocess*)	ESPERA
	Módulo que especifica o comportamento do protocolo RSVP-SVC no lado do operador do serviço (Rx).	
6	MIP-MIPBODY (*systemprocess*)	TRANSFERE_NO_MEIO
	Módulo que simula o meio de transporte, apenas transfere de uma máquina para outra a mensagem.	

Tab. 7.1 – Instâncias Criadas pelo Simulador

A seguir sumariza-se de forma tabular a descrição dos estados de cada módulo.

- Módulo MAPIAR descrito na Tabela 7.2.

NOME	DESCRIÇÃO
ESPERA	Estado em que o módulo é inicializado, aguardando pedidos de conexão dos *hosts*
ABRE_CONEXÃO	Estado que inicializa variáveis e envia requisição de LSP discado para módulo SVCTx
CRIA_LSPSVC	Estado que aguarda a confirmação do serviço discado
AGUARDA_LSPSVC	Estado que aguarda o estabelecimento do Túnel LSP discado
ABERTO	Estado de refrescamento no qual as camadas superiores estão transferindo dados
SUSPENDE_LSPSVC	Estado que monta objeto de suspensão de LSP-SVC
SUSPENSO	Estado de refrescamento no qual as camadas superiores não podem transferir dados, uma vez que o Túnel LSP está suspenso
REATIVA_LSPSVC	Estado que re-estabelece LSP previamente suspenso
ENCERRA_LSPSVC	Estado que monta mensagem de desconexão explícita de um Túnel LSP

Tab. 7.2 – Descrição dos Estados do Módulo MAPIAR

- Módulo MAPI-LER descrito na Tabela 7.3.

NOME	DESCRIÇÃO
ESPERA	Estado em que o módulo de interface das camadas superiores como RSVP-SVC é inicializado

Tab. 7.3 – Descrição dos Estados do Módulo MAPILER

- Módulo MSVCTX descrito na Tabela 7.4.

NOME	DESCRIÇÃO
ESPERA	Estado de inicialização do módulo transmissor do RSVP-SVC
DISCANDO	Estado que monta o objeto <SVC_REQUEST> e o envia para o módulo receptor
AGUARDA_LSPSVC	Estado que aguarda o estabelecimento do Túnel LSP-SVC uma vez que a operacionalidade do serviço SVC foi comprovada
LSPSVC_ESTABELECIDO	Estado de refrescamento no qual as camadas superiores estão transferindo dados
LSPSVC_SUSPENSO	Estado de refrescamento no qual as camadas superiores não podem transferir dados, uma vez que o Túnel LSP está suspenso

Tab. 7.4 – Descrição dos Estados do Módulo MSVCTX

- Módulo MSVCRX descrito na Tabela 7.5.

NOME	DESCRIÇÃO
ESPERA	Estado de inicialização do módulo receptor do RSVP-SVC
AGUARDA_LSPSVC	Estado que aguarda o estabelecimento do Túnel LSP-SVC uma vez que a operacionalidade do serviço SVC foi comprovada
LSPSVC_ESTABELECIDO	Estado de refrescamento no qual as camadas superiores estão transferindo dados
LSPSVC_SUSPENSO	Estado de refrescamento no qual as camadas superiores não podem transferir dados, uma vez que o Túnel LSP está suspenso
AGUARDA_LSR_LABEL	Estado no qual o receptor aguarda o rótulo para determinado LSP

Tab. 7.5 – Descrição dos Estados do Módulo MSVCRX

- Módulo MIP descrito na Tabela 7.6.

NOME	DESCRIÇÃO
TRANSFERE_NO_MEIO	Estado no qual o módulo MIP é inicializado e permanece durante todo o tempo exercendo uma função de repassador de mensagens do transmissor para o receptor e vice-e-versa

Tab. 7.6 – Descrição dos Estados do Módulo MIP

Deve-se levar em conta, ainda, as seguintes observações:

- Todos os módulos são *systemprocess*, assim são executados assíncronamente. Como eles não são refinados em sub-módulos, o comportamento global não mudaria se eles fossem definidos como *systemactivities*;
- Um pedido de conexão discada pode ser solicitado por qualquer *host*, desde que ele implemente o RSVP-SVC e o MPLSoLAN. Ademais, a solicitação é dirigida ao *Access Router* (*QoS-Server*), que efetivamente faz a discagem;
- Ressalta-se também que se fez necessário a definição de alguns procedimentos e funções sumarizados nas Tabelas 7.7 e 7.8, respectivamente, a fim de tornar possível a simulação/verificação do RSVP-SVC. Tais funcionalidades já são implementadas pelo RSVP ou RSVP-TE. Assim, o corpo de execução desses procedimentos apenas simula a atribuição ou retorno de valores, não especificando, portanto, o algoritmo real do RSVP.

NOME	**PARÂMETROS**
DESCRIÇÃO	
reserva_recursos	onde : IPv4_Address lsp_id : Ident_LSP filterspec : FIlterSpec_Type
Simula funcionalidade já implementada pelo RSVP no que concerne à reserva de recursos em cada nó.	
deleta_estado	onde : IPv4_Address lsp_id : Ident_LSP
Simula funcionalidade já implementada pelo RSVP no que concerne a deleção de um estado de PATH ou de RESV.	
atualiza_valores_qos	onde : IPv4_Address lsp_id : Ident_LSP novo_filterspec : FilterSpec_Type
Simula funcionalidade já implementada pelo RSVP no que concerne à alteração dos valores dos parametros de QoS previamente acordados.	

Tab. 7.7 – Procedimentos Definidos para o RSVP-SVC

NOME	PARÂMETROS	RETORNO
DESCRIÇÃO		
objeto_implementado	onde : IPv4_Address class_num : CN_Type class_type : CT_Type	*Boolean*
Função que testa se dado objeto é reconhecido pelo sistema.		
tem_recursos	onde : IPv4_Address filterspec : FilterSpec_Type	*Boolean*
Função executada quando do procedimento de admissão de novos LSP, ou reativação de LSP em estado suspenso, ou quando da alteração dos parâmetros de QoS, a fim de verificar a existência de recursos para atender a solicitação.		
lsp_valido	lsp_id : Ident_LSP	*Boolean*
Verifica se existe algum Túnel LSP com o referido LSP_ID.		
svc_operacional	onde : IPv4_Address	*Boolean*
Função que testa, no módulo SVCRX se o serviço discado está operacional.		
chamador_valido	quem : IPv4_Address	*Boolean*
Função que verifica se o iniciador da chamada tem autoridade e/ou permissão para fazer o chamado de uma conexão discada.		

Tab. 7.8 – Funções Definidas para o RSVP-SVC

7.4.1. Caso 1: Estabelecimento de uma Conexão Discada sem Erro

A seqüência para estabelecimento de uma conexão discada sem erro permite verificar uma propriedade de acessibilidade, pois se partindo do estado inicial do sistema, caso ocorra um pedido de estabelecimento de conexão discada, espera-se que o sistema chegue a um estado onde a conexão foi estabelecida sem erro, caso o sistema suporte o serviço solicitado.

As primitivas de serviço envolvidas no estabelecimento da conexão discada são as apresentadas na Tabela 7.9:

FASE	TIPO	NOME	PARÂMETROS
Discagem	*Request*	Lspsvc_req	SvcRQ
	Indication	Lspsvc_ind	
	Confirmation	Lspsvc_conf	SvcRP
Discagem (Meio)		Path	SvcRQ
		Resv	SvcRP
Estabelecimento Túnel	*Request*	Label_req	LabelRQ
	Indication	Label_ind	
	Confirmation	Label_conf	Label
Estabelecimento Túnel (Meio)		Path	LabelRQ
		Resv	Label

Tab. 7.9 – Primitivas Suportadas pelo Provedor de Serviços

A Figura 7.6 apresenta o Diagrama de Seqüência para o estabelecimento de uma conexão discada sem erro. Os números circulados relacionam o diagrama com as ações descritas (coluna índice) na Tabela 7.10.

Vale lembrar que o Serviço SVC opera entre um par de usuários: o usuário iniciador (tipicamente o AR) e o usuário operador (tipicamente o LER).

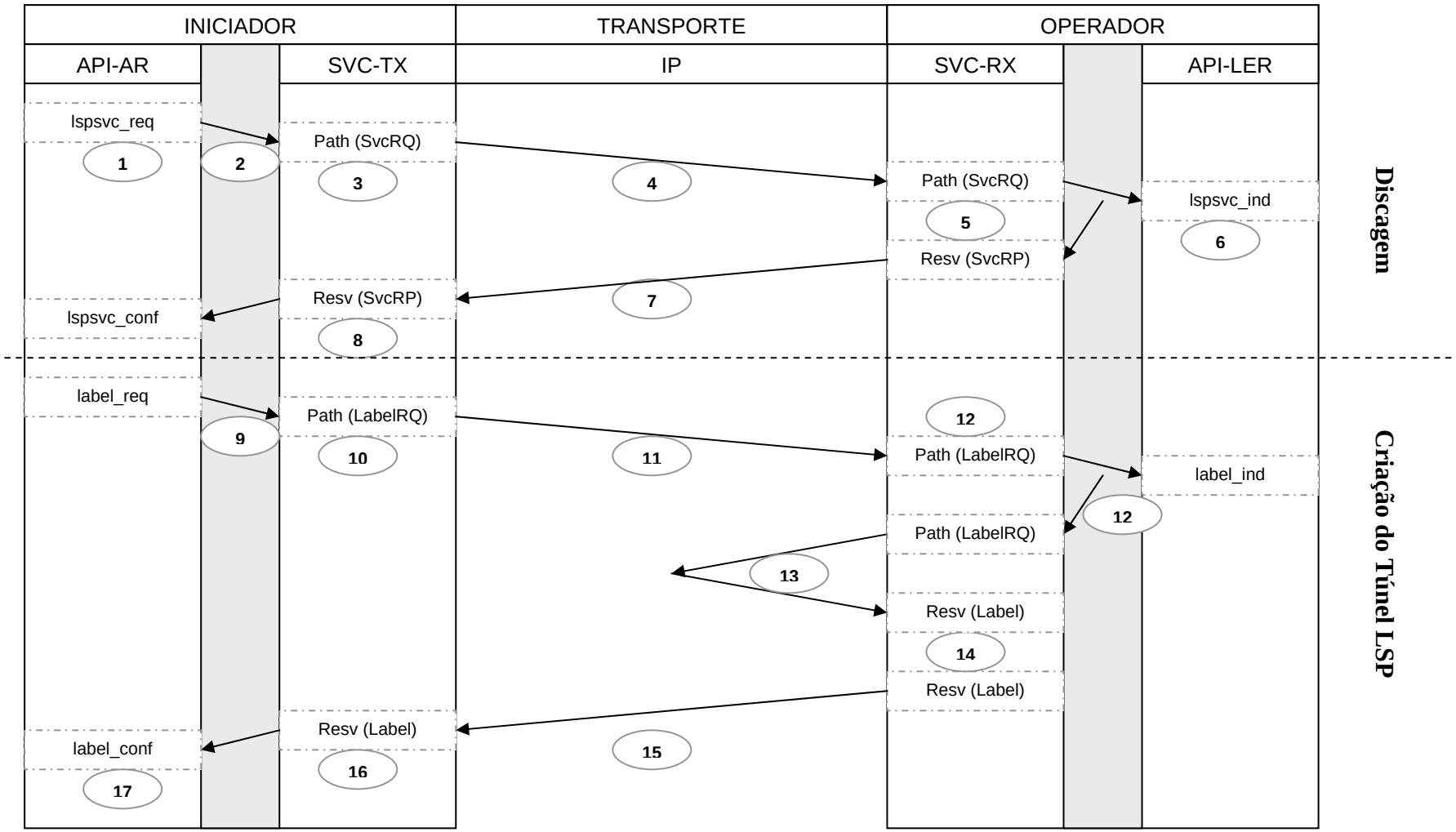

Fig. 7.6 – Diagrama de Seqüência para o Estabelecimento sem Erro de uma Conexão Discada e um LSP Associado

Índice	Instância	Estado *FROM*	Transição	Estado *TO*	Observação
1	2 (AR)	Espera	0 (T0_APIAR)	Abre_conexão	A máquina APIAR é inicializada em ESPERA. Ao chegar um pedido de conexão ela processa o pedido e vai para o estado ABRE_CONEXÃO
2	2 (AR)	Abre_conexão	1 (T1_APIAR)	Cria_lspsvc	Após o módulo APIAR processar o pedido de conexão, ele inicializa variáveis e envia um *lspsvc_req* para o módulo SVCTX e vai para o estado CRIA_LSPSVC
3	4 (TX)	Espera	1 (T1_SVCTX)	discando	O módulo SVCTX é inicializada em ESPERA. Ao chegar um pedido *lspsvc_req*, ela processa-o, monta um PATH(<svc_request>) que é enviado para o módulo MIP e vai para o estado DISCANDO
4	6 (IP)	Transfere_no_meio	0 (T0_IP)	Transfere_no_meio	O módulo MIP recebe PATHs e RESVs e simplesmente transfere para a outra entidade. No caso, recebeu um PATH(<svc_request>) do AR e o enviou para o LER
5	5 (RX)	Espera	3 (T3_SVCRX)	Aguarda_lspsvc	O módulo SVCRX é inicializado em ESPERA. Ao receber uma mensagem PATH(<svc_request>), processa, monta um *lspsvc_ ind* para o módulo APILER, inicializa o objeto SVC_RESPONSE e monta um RESV(<svc_response>) para o SVCTX. Vai para o estado AGUARDA_LSPSVC
6	3 (LER)	Espera	0 (T0_APILER)	Espera	O módulo APILER é incializado em ESPERA. Trata o *lspvc_ ind*, que sinaliza um pedido de conexão discada, descarta-o e permanece em ESPERA
7	6 (IP)	Transfere_no_meio	1 (T1_IP)	Transfere_no_meio	Recebe RESV(<svc_response>) do SVCRX e retransmite-o sem nenhuma modificação para o SVCTX
8	4 (TX)	Discando	4 (T4_SVCTX)	Aguarda_lspsvc	Recebe o RESV(<svc_response>) do LER, confirmando a operacionalidade do serviço e a sua disponibilidade para o assinante iniciador, sinaliza para o módulo APIAR com um *lspsvc_ conf*. Vai para o estado AGUARDA_LSPSVC

Continua na próxima página ...

Continuação da página anterior ...

Índice	Instância	Estado *FROM*	Transição	Estado *TO*	Observação
9	2 (AR)	Cria_lsp	2 (T2_APIAR)	Aguarda_lspsvc	Confirmada a operacionalidade do serviço discado. O módulo APIAR monta um *label_req* e o envia para o módulo SVCTX. Muda para o estado AGUARDA_LSPSVC
10	4 (TX)	Aguarda_lspsvc	9 (T9_SVCTX)	Aguarda_lspsvc	Processa o *label_req* e monta a mensagem PATH(<*label_request*)
11	6 (IP)	Transfere_no_meio	0 (T0_IP)	Transfere_no_meio	Recebe PATH(<*label_request*) do AR e envia-o para o LER
12	5 (RX)	Aguarda_lspsvc	5 (T5_SVCRX)	Aguarda_lsr_label	Recebe PATH(<*label_request*) do meio (MIP) e processa. Monta PATH(<*label_request*>) para a rede e vai para o estado AGUARDA_LSR_LABEL
13*	6 (IP)	Transfere_no_meio	2 (T2_IP)	Transfere_no_meio	Recebe PATH(<*label_request*>) do LER; Simula o envio do PATH(<*label_request*>) pela rede até o destinatário que responde com um RESV(<*label*>). Simula o recebiento do RESV(<*label*>) da rede; Envia RESV(<*label*>) para RX
14*	5 (RX)	Aguarda_lsr_label	6 (T6_SVCRX)	Lspsvc_estabelecido	Processa o RESV(<*label*>) recebido da rede e envia RESV(<*label*>) para o SVCTX. Vai para o LSPSVC_ESTABELECIDO
15	6 (IP)	Transfere_no_meio	1 (T1_IP)	Transfere_no_meio	Recebe RESV(<*label*> do LER e envia RESV(<*label*>) para o AR
16	4 (TX)	Aguarda_lspsvc	11 (T11_SVCTX)	Lspsvc_estabelecido	Recebido o RESV(<*label*>, processa-o e sinaliza para o APIAR
17	2 (AR)	Aguarda_lspsvc	6 (T6_APIAR)	Aberto	Uma vez confirmada a chamada, vai para o estado ABERTO, que é um estado de manutenção dos estados, aguardo de suspensões e términos
* As transições 13 e 14 simulam o envio (LabelRQ) e recebimento (Label) de rótulos pela rede para montagem do Túnel LSP-SVC.					

Tab. 7.10 – Descrição da Seqüência de Transições para o Estabelecimento de uma Conexão Discada e um LSP Associado

Para ilustrar o uso do módulo simulador do EDT, bem como o conjunto de ferramentas que ele disponibiliza, descrevem-se a seguir alguns passos da seqüência de transições disparadas para verificação do caso 1.

A Figura 7.7 mostra a tela inicial do simulador após pedido de execução da especificação rsvpsvc_v3.stl. Pode-se observar no lado esquerdo superior a relação de instâncias (vide Tabela 7.1) e um pouco mais abaixo a transição correntemente pronta para disparo (transição 0 da instância 2, nomeada T0_APIAR). No quadro principal, à direita, tem-se a janela de saída de comandos do simulador (edb>).

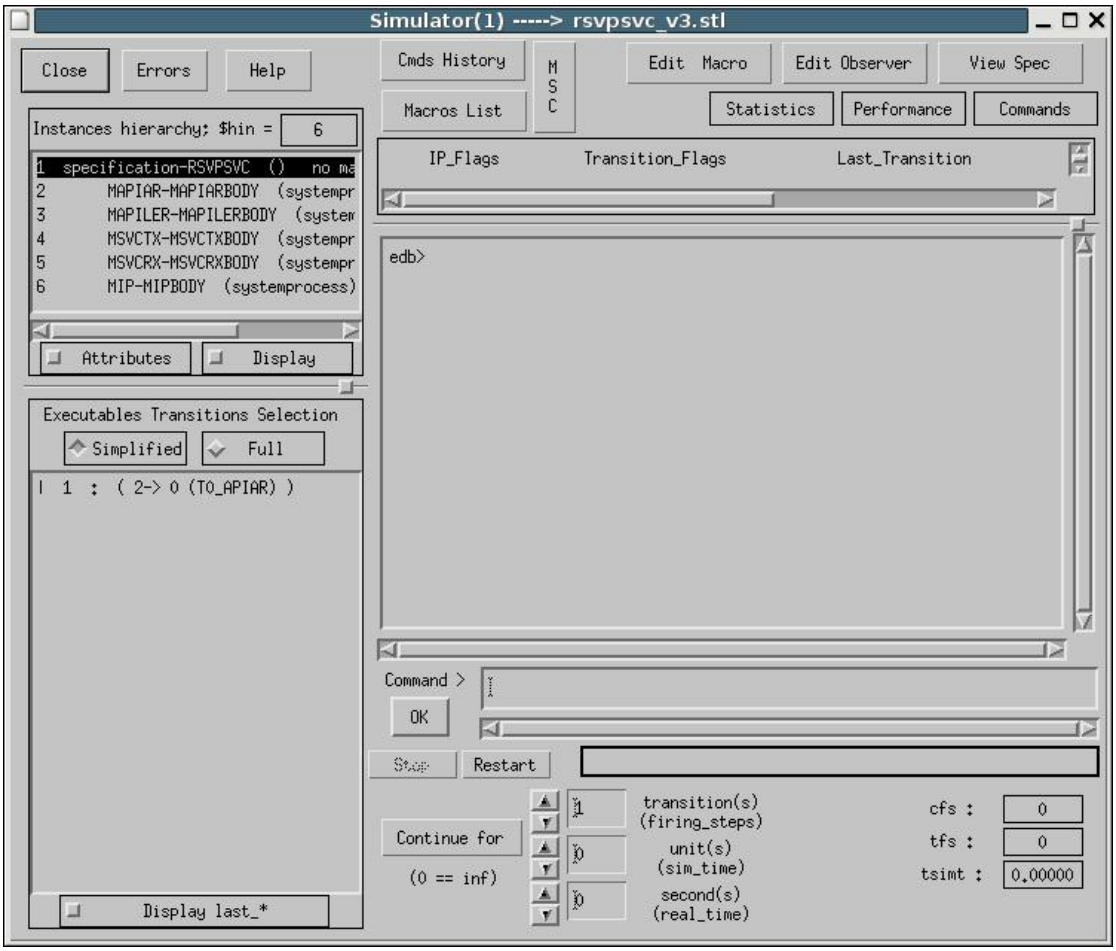

Fig. 7.7 – Tela Inicial da Simulação do rsvpsvc_v3.stl

Pode-se comandar o disparo da transição corrente, clicando sobre ela. O resultado é mostrado na Figura 7.8. Pode-se observar que a transição corrente muda para 1 (T1_APIAR) ainda na instância do módulo 2. No quadro principal é possível identificar que, até o momento, houve uma transição disparada e que a

última transição disparada foi a transição 0 (T0_APIAR) da instância do módulo 2 (MAPIAR MAPIARBODY).

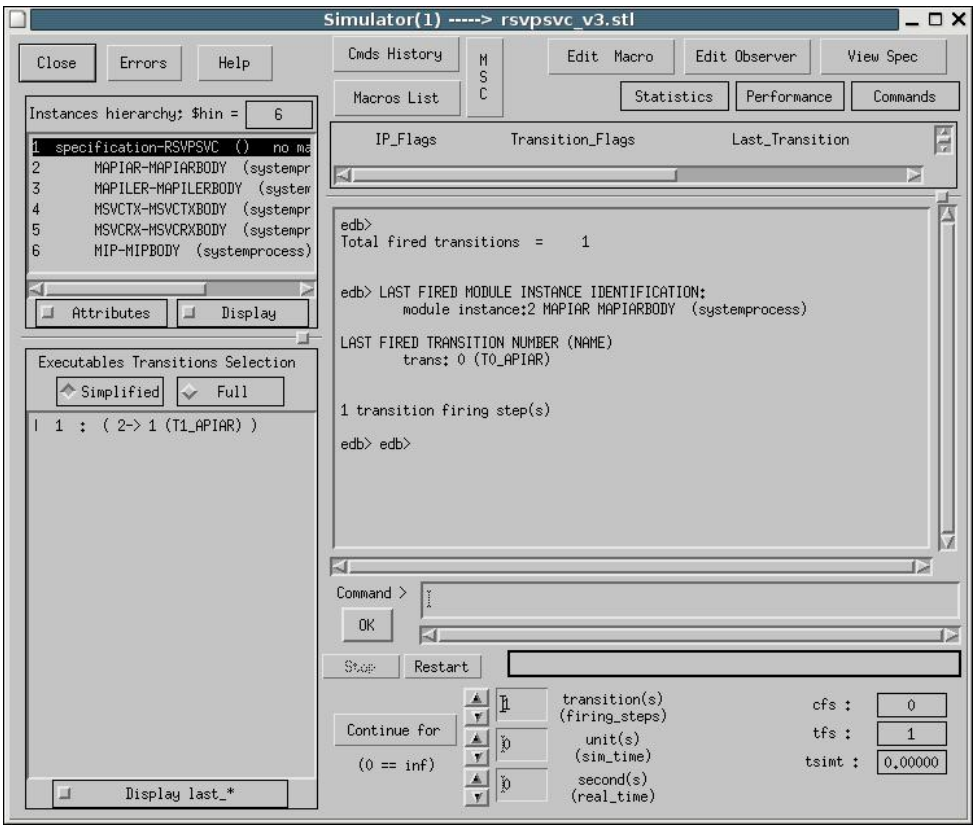

Fig. 7.8 – Disparo da transição 0 (T0_APIAR)

Clicando no botão [*Display last_**], no canto inferior esquerdo, abre-se uma caixa de diálogo com uma série de comandos que disponibilizam informações relativas à última transição disparada, conforme apresentado na Figura 7.9.

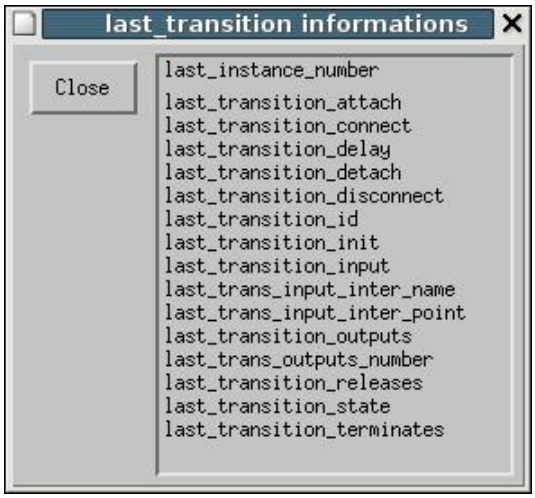

Fig. 7.9 – Caixa de Diálogo *last_transition informations*

A Figura 7.10 abaixo mostra o resultado dos seguintes comandos: $ltrid (*last_transition_id*), $ltro (*last_transition_output*) e $ltrst (*last_transition_state*).

- O comando $ltrid mostra que a última transição disparada foi a transição 1 (T1_APIAR);
- O comando $ltro diz que a saída da última transição foi a primitiva LSPSVC_REQ pela porta IPAR, com os parâmetros SVCRQ.TAMANHO=10, SVCRQ.CLASS_NUM=30, entre outros;
- O comando $ltrst informa que a última transição evoluiu do estado ABRE_CONEXÃO para o estado CRIA_LSPSVC.

A utilização desses comandos na fase de verificação da especificação é grande. Como exemplo, as Tabelas 7.10, 7.12, 7.13, 7.15, 7.17 e 7.19 foram construídas a partir de informações extraídas da execução desses comandos.

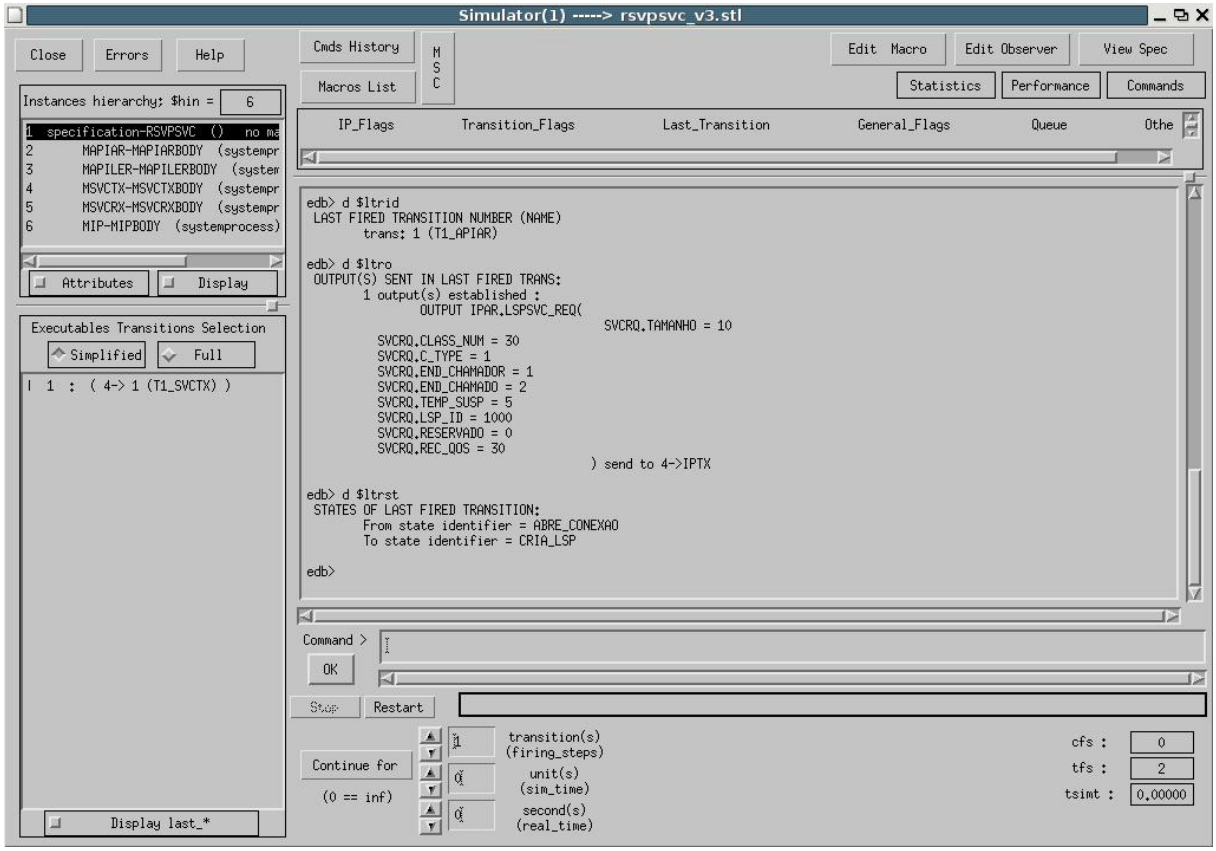

Fig. 7.10 – Exemplo de saída de comandos *last_transition informations*

Esses e outros comandos podem ser digitados diretamente na linha de comando, que se localiza um pouco abaixo da janela de saída dos comandos

(*Command*>), seguidos de um clique no botão [Ok]. Pode-se também parar uma execução (botão [*Stop*]) ou reiniciá-la (botão [*Restart*]). Alterar o passo de disparo e as unidades dos tempos de simulação e real, também são possíveis.

O simulador permite ainda, entre outros comandos, a visualização do hitórico de comandos executados através do botão [*Cmds History*], a edição de macros (botão [*Macro Edit*]), a exibição das macros gravadas (botão [*Macro List*]), a visualização da especificação através do botão [*View Spec*], a visualização de estatísticas (botão [*Statistics*]) tais como: transição mais disparada; quantidade de vezes que determinada transição foi disparada; gerar relação com todas as transições, por módulo, indicando a quantidade de disparos de cada uma delas durante a simulação. Em alguns desses casos é possível, inclusive, a geração de gráficos.

A título de exemplo, a Figura 7.11 mostra o resultado da execução do comando [d $trstat (6)], no 16º. passo da seqüência 1 de testes. Esse comando tem o seguinte significado: exibir na tela o número de vezes que cada transição da instância 6 foi disparada. Pode-se ver que o resultado foi 2, 2, 1, 0 e 0 disparos para as transições 0, 1, 2, 3 e 4, respectivamente, da instância 6 (módulo MIP). O resultado foi gravado no arquivo ./rsvpsvc_v3.trstat_6.gr.

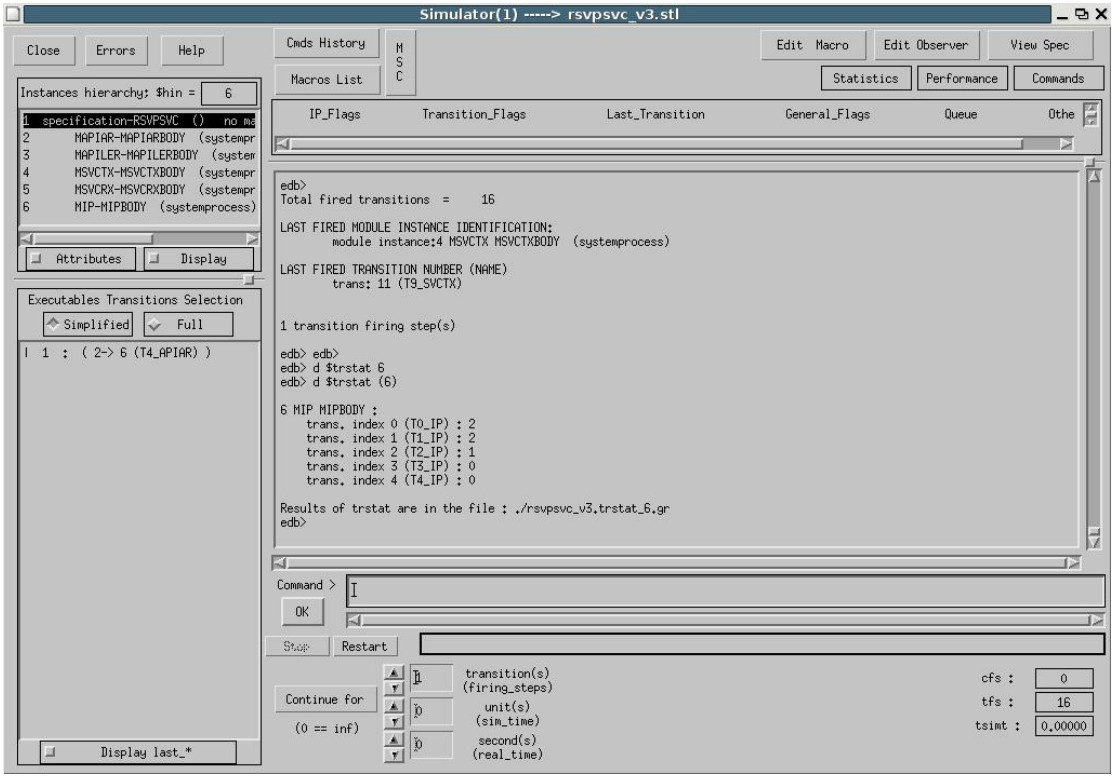

Fig. 7.11 – Exemplo de Saída do Comando [d $trstat (6)]

Uma vez que o resultado foi gravado, pode-se agora selecionar a opção [*plot*] do botão [*Statistics*] para a plotagem do gráfico relativo ao levantamento realizado. Seleciona-se o arquivo desejado (Figura 7.12), clica-se no botão [Ok] e tem-se, na tela, o gráfico correspondente (Figura 7.13).

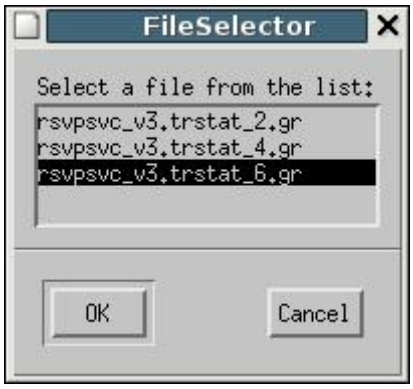

Fig. 7.12 – Seleção de Arquivo para Plotagem

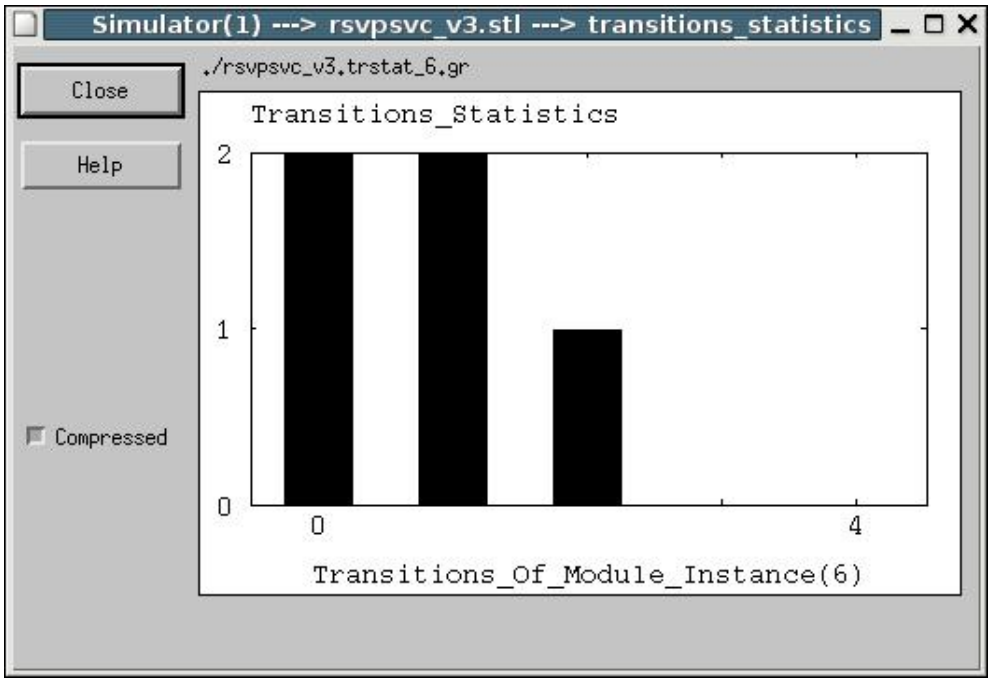

Fig. 7.13 – Gráfico Estatística de Transições da Instância de Módulo 6

O gráfico de barras da Figura 7.13 mostra que as transições 0 e 1 da instância 6, foi disparada 2 vezes, cada. Mostra ainda que a transição 3 foi disparada 1 vez e, que as transições 3 e 4, não foram disparadas na seqüência verificada.

7.4.2. Caso 2: Tentativa de Estabelecimento de uma Conexão Discada com Erro

As seqüências apresentadas nesta sub-seção permite-nos verificar uma propriedade de acessibilidade nos seguintes casos de erro durante a discagem: a) erro devido a serviço SVC não operacional; b) erro devido a serviço não disponível para assinante iniciador.

As primitivas de serviço envolvidas na tentativa de estabelecimento de uma conexão discada que apresenta os erros supra-mencionados são as apresentadas na Tabela 7.11 abaixo:

FASE	TIPO	NOME	PARÂMETROS
Discagem	*Request*	Lspsvc_req	SvcRQ
Discagem (Meio)		Path	SvcRQ
		Resverr	End_origem, End_destino, Tipo_erro, Subtipo_erro
Erro		Servicosvcinoperante_ind	
		Servicosvcindisponivelparaassinante_ind	

Tab. 7.11 – Primitivas Suportadas pelo Provedor de Serviços

A Figura 7.14 apresenta os diagramas de seqüência das duas seqüências: a) erro devido a serviço SVC não operacional; b) erro devido a serviço SVC não disponível para o assinante iniciador. Os números circulados relacionam o diagrama com as ações descritas (coluna índice) na Tabela 7.12, para o primeiro caso, e a Tabela 7.13 para o segundo caso.

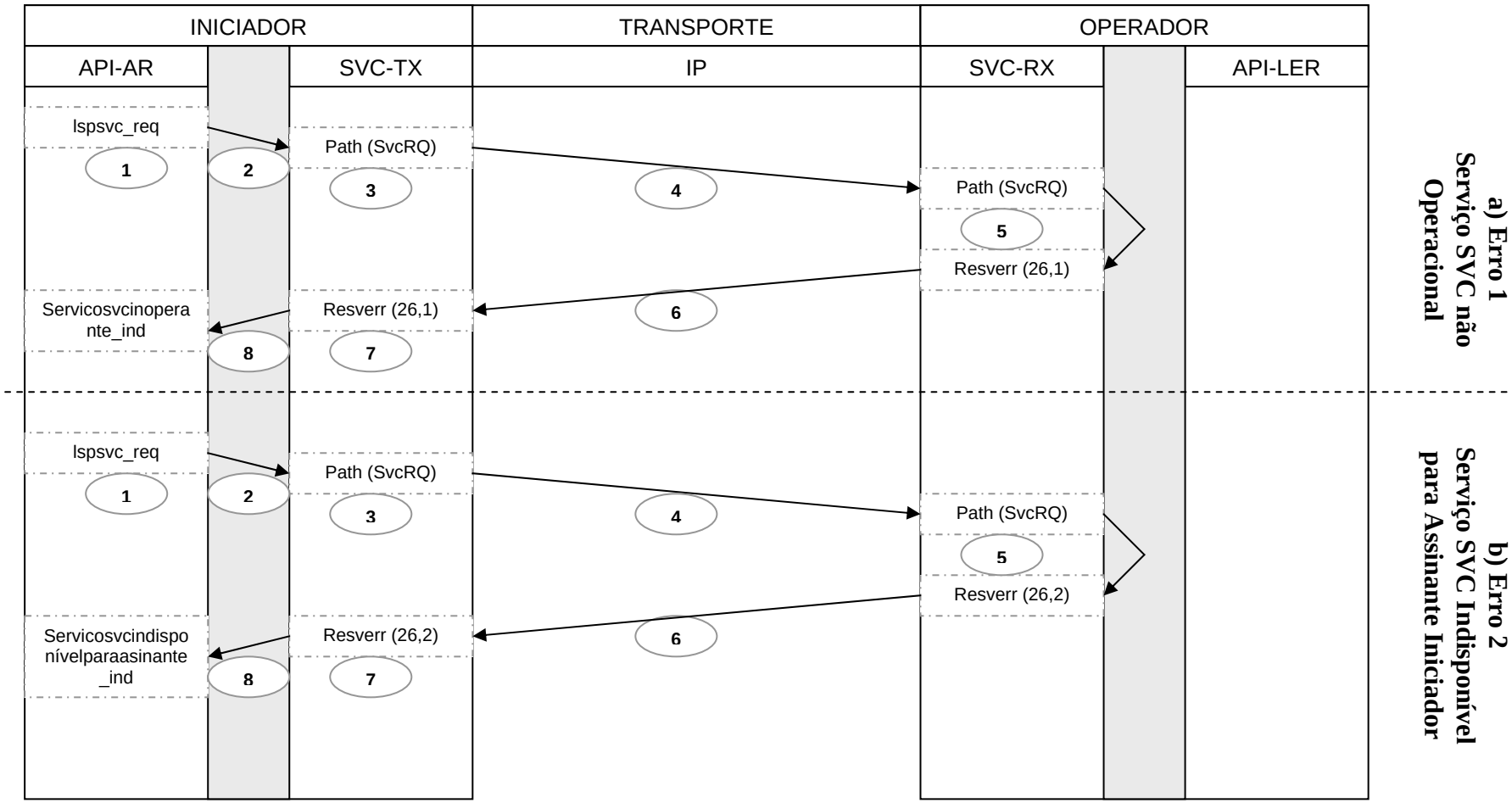

Fig. 7.14 – Diagrama de Seqüência de Tentativa para Estabelecimento de uma Conexão Discada com Erro

Índice	Instância	Estado *FROM*	Transição	Estado *TO*	Observação
1	2 (AR)	Espera	0 (T0_APIAR)	Abre_conexão	A máquina APIAR é inicializada em ESPERA. Ao chegar um pedido de conexão ela processa o pedido e vai para o estado ABRE_CONEXÃO.
2	2 (AR)	Abre_conexão	1 (T1_APIAR)	Cria_lspsvc	Após o módulo APIAR processar o pedido de conexão, ele inicializa variáveis e envia um *lspsvc_req* para o módulo SVCTX e vai para o estado CRIA_LSPSVC.
3	4 (TX)	Espera	1 (T1_SVCTX)	Discando	O módulo SVCTX é inicializada em ESPERA. Ao chegar um pedido *lspsvc_req*, ela processa-o, monta um PATH(<*svc_request*>) que é enviado para o módulo MIP e vai para o estado DISCANDO.
4	6 (IP)	Transfere_no_meio	0 (T0_IP)	Transfere_no_meio	O módulo MIP recebe PATHs e RESVs e simplesmente transfere para a outra entidade. No caso, recebeu um PATH(<*svc_request*>) do AR e o enviou para o LER.
5	5 (RX)	Espera	1 (T1_SVCRX)	Espera	O módulo SVCRX é inicializado em ESPERA. Ao receber uma mensagem PATH(<*svc_request*>), a processa. Ao constatar que o serviço não está operacional monta uma mensagem de erro RESVERR para o módulo SVCTX do AR. Permanece no estado de ESPERA.
6	6 (IP)	Transfere_no_meio	4 (T4_IP)	Transfere_no_meio	Recebe mensagem RESVERR do LER e a envia para o AR.
7	4 (TX)	Discando	6 (T6_SVCTX)	Espera	Ao receber a mensagem de erro RESVERR do LER, identifica o código e sub-código do erro e sinaliza para a interface APIAR a não operacionalidade do serviço discado. O módulo SVCTX do AR volta para o estado ESPERA.
8	2 (AR)	Cria_lspsvc	4 (T4_APIAR)	Espera	Ao receber a sinalização *servicosvcinoperante_ind* processa e volta para o estado de ESPERA.

Tab. 7.12 – Descrição da Seqüência de Transições para o Estabelecimento de uma Conexão Discada com Erro Devido a Serviço SVC não Operacional

Índice	Instância	Estado *FROM*	Transição	Estado *TO*	Observação
1	2 (AR)	Espera	0 (T0_APIAR)	Abre_conexão	A máquina APIAR é inicializada em ESPERA. Ao chegar um pedido de conexão ela processa o pedido e vai para o estado ABRE_CONEXÃO.
2	2 (AR)	Abre_conexão	1 (T1_APIAR)	Cria_lspsvc	Após o módulo APIAR processar o pedido de conexão, ele inicializa variáveis e envia um *lspsvc_req* para o módulo SVCTX e vai para o estado CRIA_LSPSVC.
3	4 (TX)	Espera	1 (T1_SVCTX)	Discando	O módulo SVCTX é inicializada em ESPERA. Ao chegar um pedido *lspsvc_req*, ela processa-o, monta um PATH(<*svc_request*>) que é enviado para o módulo MIP e vai para o estado DISCANDO.
4	6 (IP)	Transfere_no_meio	0 (T0_IP)	Transfere_no_meio	O módulo MIP recebe PATHs e RESVs e simplesmente transfere para a outra entidade. No caso, recebeu um PATH(<*svc_request*>) do AR e o enviou para o LER.
5	5 (RX)	Espera	2 (T2_SVCRX)	Espera	O módulo SVCRX é inicializado em ESPERA. Ao receber uma mensagem PATH(<*svc_request*>), a processa. Ao constatar que o serviço não está operacional monta uma mensagem de erro RESVERR para o módulo SVCTX do AR. Permanece no estado de ESPERA.
6	6 (IP)	Transfere_no_meio	4 (T4_IP)	Transfere_no_meio	Recebe mensagem RESVERR do LER e a envia para o AR.
7	4 (TX)	Discando	7 (T7_SVCTX)	Espera	Ao receber a mensagem de erro RESVERR do LER, identifica o código e sub-código do erro e sinaliza para a interface APIAR a não disponibilidade do serviço para o assinante iniciador. O módulo SVCTX do AR volta para o estado ESPERA.
8	2 (AR)	Cria_lspsvc	5 (T5_APIAR)	Espera	Ao receber a sinalização *servicosvcindisponivelparaassinante_ind* processa e volta para o estado de ESPERA.

Tab. 7.13 – Descrição da Seqüência de Transições para o Estabelecimento de uma Conexão Discada com Erro Devido a Serviço SVC não Disponível para Assinante Iniciador

7.4.3. Caso 3: Suspensão de um LSP-SVC

A seqüência para a suspensão de um LSP-SVC permite verificar uma propriedade de respondimento, pois a partir do estado onde o sistema se encontra aberto (conexão e LSP estabelecidos), a requisição de suspensão de um LSP-SVC é atendida sem erro.

As primitivas de serviço envolvidas na suspensão de uma conexão discada são as apresentadas na Tabela 7.14 abaixo:

FASE	TIPO	NOME	PARÂMETROS
Discagem	*Request*	suspendlspsvc_req	SuspendRQ
	Indication	suspendlspsvc_ind	
	Confirmation	suspendlspsvc_conf	SuspendRP
Discagem (Meio)		Path	SuspendRQ
		Resv	SuspendRP

Tab. 7.14 – Primitivas Suportadas pelo Provedor de Serviços

A Figura 7.15 apresenta o Diagrama de Seqüência para a suspensão de um LSP-SVC. Os números circulados relacionam o diagrama com as ações descritas (coluna índice) na Tabela 7.15.

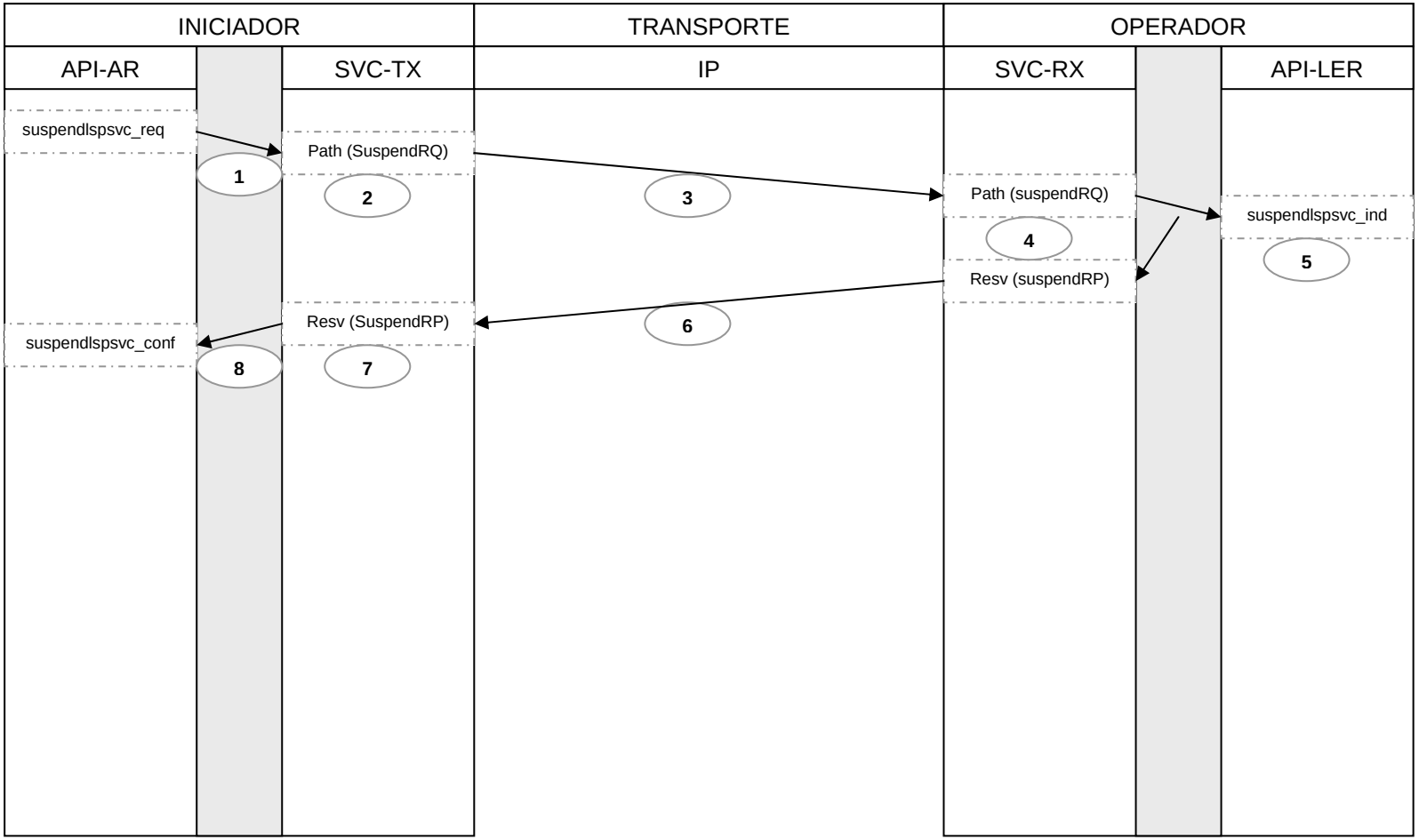

Fig. 7.15 – Diagrama de Seqüência para Suspensão de um LSP-SVC

Índice	Instância	Estado *FROM*	Transição	Estado *TO*	Observação
1	2 (AR)	Aberto	7 (T7_APIAR)	Suspende_lspsvc	Estando no estado ABERTO, o módulo APIAR pode receber um pedido de *suspendlspsvc_req* vindo do usuário iniciador. Nesse caso, ele testa a validade do LSP_ID e sinaliza o módulo SVCTX. Muda para o estado SUSPENDE_LSPSVC.
2	4 (TX)	Lspsvc_estabelecido	15 (T15_SVCTX)	Lspsvc_suspenso	Ao receber um *suspendlspsvc_req*, o módulo SVCTX incializa o objeto <*suspend_request*> e monta uma mensagem PATH(<*suspend_request*>) para o módulo receptor da entidade par. Muda o estado para LSPSVC_SUSPENSO.
3	6 (IP)	Transfere_no_meio	0 (T0_IP)	Transfere_no_meio	Recebe a mensagem PATH(<*suspend_request*>) do AR e a envia para o LER.
4	5 (RX)	Lsp_estabelecido	13 (T13_SVCRX)	Suspenso	Ao receber a mensagem PATH(<*suspend_request*>) do meio (IP), a processa e envia um *suspendlspsvc_ind* para o módulo APILER, incializa o objeto <*suspend_response*> e monta uma mensagem RESV(<suspend_response>) que será enviada para o módulo transmissor do AR.
5	3 (LER)	Espera	1 (T1_APILER)	Espera	Recebe *suspendlspsvc_ind* e permanece em ESPERA.
6	6 (IP)	Transfere_no_meio	1 (T1_IP)	Transfere_no_meio	Recebe RESV(<*suspend_response*>) do LER e envia RESV(<*suspend_response*>) para o AR
7	4 (TX)	Lspsvc_suspenso	26 (T26_SVCTX)	Lspsvc_suspenso	Ao receber o RESV(<*suspend_response*>) do LER, confirmando a suspensão do LSP-SVC, processa a mensagem e sinaliza para APIAR com um *suspendlspsvc_conf*. Permanece no estado LSPSVC_SUSPENSO.
8	2 (AR)	Suspende_lspsvc	9 (T9_APIAR)	Suspenso	Ao receber o *suspendlspsvc_conf,* que confirma a suspensão do LSP-SVC, vai para o estado SUSPENSO.

Tab. 7.15 – Descrição da Seqüência de Transições para a Suspensão de um LSP-SVC

7.4.4. Caso 4: Reativação de um LSP-SVC Suspenso

A seqüência para a reativação de um LSP-SVC permite verificar outra propriedade de respondimento, pois a partir de um estado onde um LSP-SVC se encontra suspenso, quando se solicita a reativação do LSP-SVC, o sistema chega a um estado onde a conexão é reativada ou não.

As primitivas de serviço envolvidas na suspensão de uma conexão discada são as apresentadas na Tabela 7.16 abaixo:

FASE	TIPO	NOME	PARÂMETROS
Discagem	*Request*	reactivationlspsvc_req	ReactivationRQ
	Indication	reactivationlspsvc_ind	
	Confirmation	reactivationlspsvc_conf	ReactivationRP
Discagem (Meio)		Path	ReactivationRQ
		Resv	ReactivationRP

Tab. 7.16 – Primitivas Suportadas pelo Provedor de Serviços

A Figura 7.16 apresenta o Diagrama de Seqüência para a reativação de um LSP-SVC. Os números circulados relacionam o diagrama com as ações descritas (coluna índice) na Tabela 7.17.

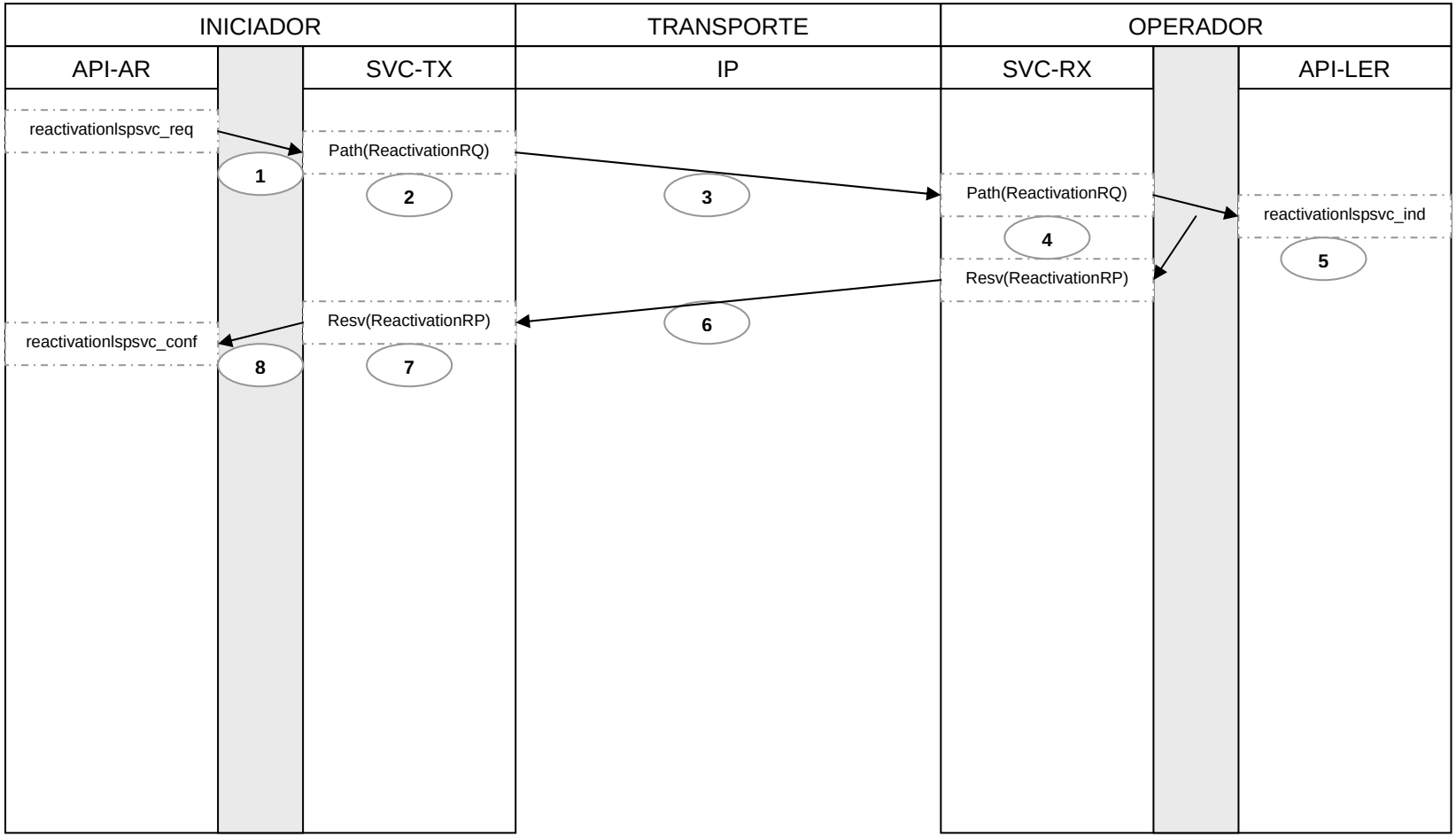

Fig. 7.16 – Diagrama de Seqüência para Reativação de um LSP-SVC

Índice	Instância	Estado *FROM*	Transição	Estado *TO*	Observação
1	2 (AR)	Suspenso	10 (T10_APIAR)	Reativa_lspsvc	Estando no estado SUSPENSO, o módulo APIAR pode receber um pedido de *reactivationlspsvc_req* vindo do usuário iniciador. Nesse caso, ele testa a validade do LSP_ID e sinaliza o módulo SVCTX. Muda para o estado REATIVA_LSPSVC.
2	4 (TX)	Lspsvc_suspenso	27 (T27_SVCTX)	Lspsvc_suspenso	Ao receber um *reactivationlspsvc_req*, o módulo SVCTX incializa o objeto <*reactivation_request*> e monta uma mensagem PATH(<*reactivation_request*>) para o módulo receptor da entidade par. Muda o estado para LSPSVC_SUSPENSO.
3	6 (IP)	Transfere_no_meio	0 (T0_IP)	Transfere_no_meio	Recebe a mensagem PATH(<*reactivation_request*>) do AR e a envia para o LER.
4	5 (RX)	Suspenso	17 (T16_SVCRX)	Lspsvc_estabelecido	Ao receber a mensagem PATH(<*reactivation_request*>) do meio (IP), a processa e envia um *reactivationlspsvc_ind* para o módulo APILER, incializa o objeto <*reactivation_response*> e monta uma mensagem RESV(<*reactivation_response*>) que será enviada para o módulo transmissor do AR.
5	3 (LER)	Espera	2 (T2_APILER)	Espera	Recebe a sinalização de que o LSP-SVC foi reativado através da primitiva *reactivationlspsvc_ind* e permanece em ESPERA.
6	6 (IP)	Transfere_no_meio	1 (T1_IP)	Transfere_no_meio	Recebe a mensagem RESV(<*reactivation_response*>) do LER e a envia para o AR.
7	4 (TX)	Lspsvc_suspenso	24 (T24_SVCTX)	Lspsvc_estabelecido	Ao receber a mensagem RESV(<*reactivation_response*>) do LER, confirmando a reativação do LSP-SVC, processa a mensagem e sinaliza para APIAR com um *reactivationlspsvc_conf*. Muda para o estado LSPSVC_ESTABELECIDO.
8	2 (AR)	Reativa_lspsvc	11 (T8_APIAR1)	Aberto	Ao receber o *reactivationlspsvc_conf,* que confirma a reativação do LSP-SVC, vai para o estado ABERTO.

Tab. 7.17 – Descrição da Seqüência de Transições para a Reativação de um LSP-SVC

7.4.5. Caso 5: Encerramento Normal de uma Conexão Discada

A seqüência para a suspensão de um LSP-SVC permite verificar uma propriedade de acessibilidade, pois a partir do estado onde o sistema se encontra aberto, existe uma seqüência que faz com que o sistema chegue a um estado onde a conexão é suspensa sem erro.

A execução consecutiva das seqüências dos casos 1 e 5 também permitem a verificação de uma propriedade de correção total da especificação, pois garante que partindo do estado inicial, o sistema consegue chegar ao estado final sem erros.

As primitivas de serviço envolvidas no encerramento normal de uma conexão discada são as apresentadas na Tabela 7.18:

FASE	TIPO	NOME	PARÂMETROS
Encerramento	*Request*	releaselspsvc_req	Lsp_id
	Indication	releaselspsvc_ind	Lsp_id
Encerramento (Meio)		Pathtear	
		Resvtear	

Tab. 7.18 – Primitivas Suportadas pelo Provedor de Serviços

A Figura 7.17 apresenta o Diagrama de Seqüência para o encerramento explícito de um LSP-SVC. Os números circulados relacionam o diagrama com as ações descritas (coluna índice) na Tabela 7.19.

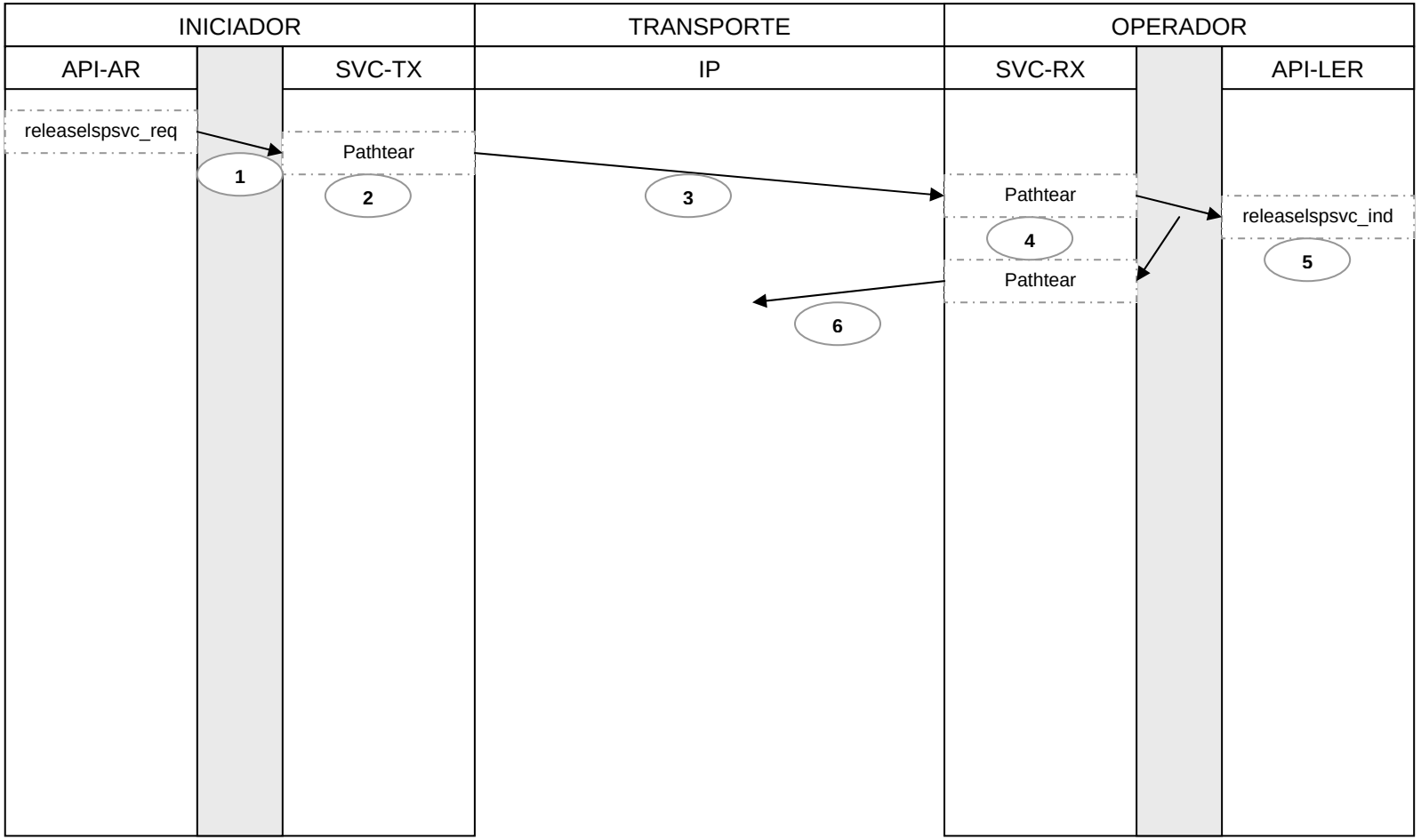

Fig. 7.17 – Diagrama de Seqüência para o Encerramento Explícito de uma Conexão Discada

Índice	Instância	Estado *FROM*	Transição	Estado *TO*	Observação
1	2 (AR)	Aberto	8 (T8_APIAR1)	Espera	Estando no estado ABERTO, o módulo APIAR pode receber um pedido de desconexão explícita através da primitiva *releaselspsvc_req* vindo do usuário iniciador. Nesse caso, ele testa a validade do LSP_ID e sinaliza o módulo SVCTX. Muda para o estado ESPERA.
2	4 (TX)	Lspsvc_estabelecido	16 (T16_SVCTX)	Espera	Ao receber um *releaselspsvc_req*, o módulo SVCTX monta uma mensagem PATHTEAR para o LER e vai para o estado de ESPERA.
3	6 (IP)	Transfere_no_meio	5 (T5_IP)	Transfere_no_meio	Recebe a mensagem PATHTEAR do AR e a envia para o LER.
4	5 (RX)	Lsp_estabelecido	8 (T13_SVCRX)	Espera	Ao receber a mensagem PATHTEAR do meio (IP), a processa e envia um *releaselspsvc_ind* para o módulo APILER e manda uma mensage PATHTEAR para a rede.
5	3 (LER)	Espera	3 (T3_APILER)	Espera	Recebe a sinalização de que o LSP-SVC foi desconectado através da primitiva *releaselspsvc_ind* e permanece em ESPERA.
6	6 (IP)	Transfere_no_meio	6 (T6_IP)	Transfere_no_meio	Recebe a mensagem PATHTEAR do LER e a envia para a rede.

Tab. 7.19 – Descrição da Seqüência de Transições para o Encerramento Explícito de um LSP-SVC

7.5. Propriedades Verificadas

A verificação de propriedades de sistemas distribuídos objetiva, em geral, demonstrar que esses sistemas irão funcionar adequadamente após sua implementação, apresentando um comportamento esperado e isento de erros.

Dentre as propriedades descritas na Seção 5.3.4, considerou-se que as mais relevantes na verificação da especificação desenvolvida são as que se seguem:

a) Propriedades de Invariância

Vale salientar que, não se pode garantir completamente a validade deste conjunto de propriedades, a menos que seja possível a geração completa do grafo de alcançabilidade. Isso não pôde ser feito, uma vez que a ferramenta EDT não conseguiu gerar todas as seqüências possíveis em tempo hábil: foi submetida à ferramenta a execução da geração completa do grafo de alcançabilidade e, depois de 5 dias completos de processamento contínuo, nenhum resultado foi gerado. Esse fato provavelmente deve-se à explosão do espaço de estados gerado e às limitações de *hardware* da máquina utilizada para essa verificação. Entretanto, procurou-se se utilizar de heurísticas, baseadas no conhecimento da operação desejada do sistema, de forma a testar a execução de seqüências selecionadas de eventos. Assim, nas seqüências verificadas, constatou-se as seguintes propriedades:

- Correção Parcial: Em todas as seqüências testadas verificou-se a validade desta propriedade, uma vez que para toda entrada testada, dentro do funcionamento correto do sistema, chegou-se à saída esperada, que caracteriza o término do ciclo;
- Ausência de Bloqueios: Também nesse caso, em todas as seqüências simuladas foi constatada a ausência de *deadlocks,* visto que sempre havia um processo habilitado que permitia a evolução do sistema;
- Bom Comportamento: De fato, tomando-se qualquer estado do sistema especificado, sempre há uma condição que expressa a execução adequada e geração de nenhuma falha ou erro. Percebeu-se a ausência de interações não executáveis e a ausência de recepções não especificadas.

b) Propriedades de Eventualidade

- Acessibilidade: A verificação dessa propriedade foi demonstrada durante a simulação das seqüências apresentadas (vide casos 1, 2 e 5);
- Vivacidade: Em nenhum caso observou-se a presença de *livelock*, de forma que todos os processos evoluíram corretamente;
- Respondimento: Essa propriedade também foi verificada. Em todas as seqüências simuladas sempre que houve uma requisição de serviço ele foi devidamente respondido (vide casos 3 e 4);
- Correção Total: Verificou-se que partindo do estado inicial o sistema chega ao estado final.

c) Propriedades de Precedência

- Vivacidade Segura: Levando-se em conta as seqüências onde se verificou a propriedade de acessibilidade, também foram verificadas as propriedades de ausência de bloqueios e de bom comportamento.
- Ausência de Resposta não Solicitada: Em nenhum caso verificou-se a ocorrência de um processo responder a um outro sem ser solicitado.
- Respondimento Justo: Sempre foi servido primeiro quem chegou primeiro. Salienta-se que não se fez nenhum teste com duas requisições “simultâneas”.

Capítulo 8
Conclusão e Trabalhos Futuros

Este capítulo apresenta as conclusões finais desse trabalho, estando organizado da seguinte forma: a Seção 8.1 tece um comentário geral sobre o que foi proposto; alguns benefícios e vantagens advindos da adoção do protocolo RSVP-SVC são relacionados na Seção 8.2; e finalmente, na Seção 8.3, relacionam-se possíveis estudos e trabalhos futuros já identificados.

8.1. Comentário Geral e Contribuições

Considerando que as demandas por QoS são irreversíveis, com um forte uso do RSVP-TE como protocolo de reserva de recursos e o mecanismo de classificação de tráfego do DiffServ e, considerando ainda, que as redes do *core* estão migrando para uma solução IP/MPLS, constata-se que as redes de acesso têm se tornado o novo gargalo, no que concerne ao provimento de uma QoS fim-a-fim verdadeira. Nesse sentido, a proposição de um serviço do tipo SVC, especificado em uma UNI MPLS, constitui-se numa forma alternativa de se estender o MPLS da borda dos ISP, através dos LER, para os roteadores de acesso das empresas, garantindo assim, uma série de benefícios adicionais não apenas para os ISP quanto para os próprios assinantes (vide Seção 8.2).

A contribuição desta Tese está focada principalmente na especificação e verificação do protocolo RSVP-SVC, a partir da definição de uma extensão ao RSVP-TE. O protocolo especificado foi usado como protocolo de sinalização para um serviço discado MPLS, bem como para o protocolo MPLSoLAN. As funcionalidades do RSVP-SVC permitiram a operacionalização de um serviço discado MPLS a partir do roteador de acesso do usuário, ou mesmo o estabelecimento de um túnel LSP fim-a-fim. Dessa forma, o objetivo de se estender o MPLS para o roteador de acesso e para os *hosts* foi alcançado.

Isto posto, as outras contribuições deste trabalho podem ser resumidas da seguinte forma:

- O serviço discado MPLS foi definido;
- Foi especificada a extensão do MPLS às redes locais através do protocolo MPLSoLAN;

- Foi definido um ambiente MPLS Fim-a-Fim com serviços diferenciados;
- A especificação do RSVP-SVC foi parcialmente verificada formalmente através da ferramenta EDT.

8.2. Benefícios

Alguns benefícios mais evidentes do uso do RSVP-SVC já foram identificados e apresentados em [SilvaNeto03a], a saber:

a) Desempenho superior devido a:

- Possibilidade de executar as aplicações diretamente sobre o MPLS, conforme detalhado em [SilvaNeto03b];
- Diminuição da carga de processamento nos AR e nos LER. Uma vez que o problema de classificação é, com a solução proposta, resolvido primariamente na origem, o processo de classificação nos AR e nos LER é resumido, possivelmente, ao procedimento mais simples de classificação que é a classificação por agregação de comportamento (BA), baseada na análise do campo EXP. No demais, o trabalho do AR seria apenas de encaminhador de pacotes, como os LSR. Ao invés de concentrar todo o processo de classificação multi-campo de pacotes nos roteadores de borda de um domínio MPLS (LER) ou nos AR, o RSVP-SVC distribui esse processo com os *hosts* (aplicações) originadores dos fluxos de dados, onde também poderá ocorrer controle de fluxo, através da implementação de filas e de escalonamento;
- Forma mais flexível e menos onerosa, em termos de processamento bruto, para a solução do problema de classificação. Uma vez que os parâmetros já foram acordados e, ainda, um valor de rótulo já foi negociado para uma dada tupla, bastaria uma consulta a uma tabela local, construída e mantida pelo MPLSoLAN, para a inserção do cabeçalho MPLS com o valor de rótulo indicado.

b) Gerenciamento de QoS por Serviço

No contexto das redes de acesso, esta característica facilita a implantação de uma política de gerenciamento e agendamento dos serviços, provendo assim os meios necessários para uma melhor utilização dos recursos da rede por parte das aplicações e dos usuários.

c) Plano de Controle Homogêneo

Conforme apresentado em [SilvaNeto03b] um plano de controle homogêneo fim-a-fim é a maneira ótima de gerenciar uma rede. Nesse sentido, estender tais mecanismos de controle até as redes de acesso e, mais precisamente aos *hosts*, que são as fontes de fluxos de dados, seria extremamente desejável, uma vez que dá ao administrador da rede local condições de melhor interagir com seu provedor de serviços, a partir de bases de dados comuns.

d) Camada MPLSoLAN como Complemento do 802.1Q

A camada MPLSoLAN constitui-se num excelente complemento ao protocolo 802.1Q, uma vez que possibilita a criação de um novo tipo de domínio de *broadcast*, a saber: *broadcast* a nível de aplicação/serviço, o que é desejável para algumas aplicações multimídia, a exemplo de videoconferência, VoIP, entre outras.

e) Campo EXP como Provimento de QoS na Rede Local

O campo EXP do cabeçalho MPLS também se constitui num complemento interessante ao IEEE 802.1p como esquema de tratamento de prioridade e, por conseguinte, um mecanismo bastante simplificado de provimento de QoS para a rede local.

f) Granularidade Fina no Estabelecimento de LSP a Partir dos *Hosts*

Essa solução provê ainda uma granularidade variável no que diz respeito ao controle de vazão e latência em bases per *host*/usuário/aplicação, uma vez que o QoS-Server (AR), por ter uma visão global dos recursos disponíveis e utilizados pela rede local, pode aproveitar esse conhecimento para realizar um efetivo controle de admissão e de fluxo, desde sua origem.

g) Redução de Alguns Fatores Inibidores da Adoção do MPLS

Na medida em que a especificação aqui apresentada adiciona facilidades no gerenciamento de ambientes com QoS, de migração de uma rede IP convencional para uma sobre o MPLS e, além disso, dá ao usuário uma boa noção quanto ao real benefício do uso do MPLS no seu dia-a-dia, ela vai de encontro aos principais fatores que, atualmente, inibem uma maior expansão do uso do MPLS [Physics04].

h) Flexibilidade no Estabelecimento de Novos Sistemas de Contabilidade

A adoção do MPLS dá flexibilidade às operadoras no que concerne à implementação de novos planos de contabilidade, que se baseiam no comportamento real do assinante.

Assim, do ponto de vista das operadoras, esta proposta pretende fornecer mais um mecanismo através do qual, novos serviços poderão ser introduzidos, com possibilidades de adição de novos assinantes e, principalmente, o estabelecimento de meios efetivos de cobrança, o que redunda em maior lucratividade para as operadoras e maior justiça para os assinantes.

Uma estrutura de contabilidade baseada no uso real pode gerar interesse nos médios e pequenos usuários, que dependem da queda nos preços, uma vez que o modelo atual baseado na cobrança de uma taxa fixa pode ser proibitivo para usuários eventuais ou locais.

No intuito de implantar efetivamente esse modelo é imprescindível o conhecimento de como os usuários estão usando os serviços, bem como de que serviços estão sendo usados sobre suas facilidades, que sejam possíveis de cobrança.

Nesse sentido, as operadoras podem estabelecer um sistema de contabilidade verdadeiramente flexível, no qual o custo pode ser ajustado ao valor que o usuário dá ao serviço, e que leve em conta a localidade e o horário do acesso e a forma como o serviço é usado, dentre outros aspectos.

Os benefícios identificados junto às operadoras podem ser relacionados a seguir:

- Cobrar efetivamente o uso do assinante, mantendo as margens de lucro através dos diversos níveis de preço;
- Aumentar a receita por assinante;
- Capturar a informação requerida na detecção de abuso;
- Oferecer pacotes de serviços personalizados, por cliente;
- Eliminar perda de receita tipicamente associada aos serviços baseado no uso;
- Prover flexibilidade para suportar novos serviços e modelos de negócio se houver mudança na tendência; e,
- Abrir novos mercados a assinantes sensíveis ao preço.

8.3. Trabalhos Futuros

No decorrer do desenvolvimento desse trabalho foi percebida a necessidade de estudos e trabalhos futuros, que podem levar em conta os seguintes aspectos:

- Segurança. Segurança e confiabilidade da informação nos roteadores de acesso a fim de protegê-lo de alterações indevidas por parte do usuário;
- Quantidade de conexões discadas simultâneas. Limitar a opção de conta de assinante que garanta uma quantidade pré-definida de conexões discadas;
- Problema do controle de filas nos *hosts*. Avaliar qual o algoritmo mais indicado a ser implementado para o controle das filas nos *hosts*, quando do uso do MPLSoLAN;
- Uso do ATM e do FR. Avaliar a possibilidade de implementação do serviço discado sobre redes ATM e FR;
- Problema de alocação de banda. No que diz respeito ao problema da alocação de banda pode-se considerar a possibilidade de uso de técnicas de IA (Inteligência Artificial) que automatizem a alocação de recursos;
- Escolha do QoS-Server. A princípio, a indicação do QoS-Server é realizada de forma estática, entretanto mecanismos de configuração dinâmica também seriam possíveis. Futuros trabalhos podem avaliar sua aplicabilidade;
- Implementação do RSVP-SVC. Verificar a possibilidade de implementação do RSVP-SVC em roteadores *Linux-based*, e assim viabilizar a realização de medidas de desempenho;
- Implementação do MPLSoLAN. Verificar possíveis ganhos com a utilização do MPLSoLAN;
- Avaliação de desempenho do MPLS Fim-a-Fim;
- GMPLS. Incluir na especificação o uso do GMPLS, em razão de suas características de interesse para aos casos aqui considerados, a exemplo da construção de LSP bi-direcionais.

Referências Bibliográficas

[Akhtar87] AKHTAR, S. Congestion Control In A Fast Packet Switching Network. Master's thesis, Washington University, 1987.

[Andersson01] ANDERSSON, L., DOOLAN, P., FELDMAN, N., FREDETTE, A., THOMAS, B. LDP Specification. Internet RFC 3036, Janeiro, 2001.

[Apostolopoulos98] APOSTOLOPOULOS, George, GUÉRIN, Roch, KAMAT, Sanjay, TRIPATHI, Satish K. Quality of service based routing: A performance perspective. Extraído em maio de 2002 de http://citeseer.nj.nec.com.

[Aukia02] AUKIA, P., KODIALAM, M., KOPPOL, P. V., LAKSHMAN, T. V., SARIN, H., SUTER, B. RATES: A server for MPLS Traffic Engineering. Extraído em maio de 2002 de http://citeseer.nj.nec.com.

[Awduche01] AWDUCHE, D., BERGER, L., LI, T., SRINIVASAN, V. SWALLOW, G. RSVP-TE: Extensions to RSVP for LSP Tunnels. Internet RFC 3209, Dezembro, 2001.

[Awduche99] AWDUCHE, D., MALCOLM, J., AGOGBUA, J., O'DELL, M., McMANUS, J. Requirements for traffic engineering over MPLS. Internet RFC 2702, Setembro, 1999.

[Bernet00] BERNET, Y. SMITH, A. DAVIE, B. Specification of the Null Service Type. Internet RFC 2997, Novembro, 2000.

[Blake98] BLAKE, S., BLACK, D., CARLSON, M., DAVIES, E., WANG, Z., WEISS, W. An Architecture for Differentiated Services. Internet RFC 2475, Dezembro, 1998.

[Braden94] BRADEN, R., CLARK, D. SHENKER, S. Integrated Services in the Internet Architecture: an Overview. Internet RFC 1633, Junho, 1994.

[Braden97] BRADEN, R., ZHANG, L. BERSON, S. HERZOG, S. JAMIN, S. Resource reservation protocol (RSVP) – version 1 Functional Specification. Internet RFC 2205, Setembro, 1997.

[Brittain00] BRITTAIN, Paul. FARREL, Adrian. MPLS Traffic Engineering: a choice of signaling protocols. Data Connection. White Paper. Extraído em janeiro de 2003 de www.dataconnection.com.

[Edt00] Estelle Development Toolset. Extraído em dezembro de 2005 de ftp://ftpdsr.int-evry.fr/EDT4.3/.

[Extreme01] Leveraging MPLS to enhance network transport capabilities In service provider and enterprise enviroments. Extreme Networks. White Paper. Extraído em março de 2002 de www.extremenetworks.com.

[Feldman01] FELDMAN, Anja. MUTHUKRISHNAN, S. Tradeoffs for Packet Classification.Tradeoffs for Packet Classification. Extraído em outubro de 2001 de http://citeseer.nj.nec.com.

[Feldman97] FELDMAN, N., VISWANATHAN, A. ARIS specification. Internet draft, draft-feldman-aris-spec-00.txt, Março, 1997.

[Ferguson99] FERGUSON, P. HUSTON, G. Quality of Service: Delivering QoS on the Internet and in Corporate Networks. Wiley Computer Publishing. 1999.

[Fialho92] FIALHO, Sergio Vianna. Uma técnica heurística para verificação semi-automatizada de sistemas distribuídas. Tese de Doutorado. UFRJ. Rio de Janeiro, 1982.

[Fineberg02] FINEBERG, Victoria. A practical architecture for implementing end-to-end QoS in an IP network. IEEE communications magazine. Janeiro, 2002.

[Gallaher02a] GALLAHER, Rick. An introduction to MPLS. Extraído em julho de 2002 de www.converdigest.com/tutoriais/mpls2.

[Gallaher02b] GALLAHER, Rick. Introduction to MPLS Label Distribution and Signaling. Extraído em julho de 2002 de www.converdigest.com/tutoriais/mpls2.

[Gallaher02c] GALLAHER, Rick. Advanced MPLS Signaling. Extraído em julho de 2002 de www.converdigest.com/tutoriais/mpls2.

[Gallaher02d] GALLAHER, Rick. MPLS network reliance and recovery. Extraído em julho de 2002 de www.converdigest.com/tutoriais/mpls2.

[Gallaher02e] GALLAHER, Rick. MPLS traffic engineering. Extraído em julho de 2002 de www.converdigest.com/tutoriais/mpls2.

[Hagard] HAGARD, Goran, WOLF, Mikael. Multiprocol Label Switching in ATM networks.

[Heinanen99] HEINANEN, J., FINLAND, Telia, GUERIN, R. A Single Rate Three Color Marker. Internet RFC 2697, Setembro, 1999.

[IEC01] Multiprocol label switching (MPLS) tutorial. Web ProForums. Extraído em setembro de 2001 de www.iec.org/tutorials/mpls/topico01.html.

[IEEE00] MPLS: The magic behind the myths. IEEE communications magazine. Janeiro, 2000.

[Integral02a] The Evolution toward Multi-service IP/MPLS Networks. White Paper. Integral Access. Extraído em maio de 2002 de www.integralaccess.com.

[Integral02b] MPLS and Next Generation Access Network. Integral Access. White Paper. Extraído em maio de 2002 de www.integralaccess.com.

[Katsube97] KATSUBE, Y., NAGAMI, K., MATSUZAWA, S., ESAKI, H. Internetworking based on Cell Switch Router – Architecture and Protocol Overview. Proc of IEEE, Vol 85, Nº 12, Dezembro, 1997.

[Lamport77] LAMPORT, L. Providing the correctness of multiprocess programs. IEEE Transactions on Software Enginneering., v. SE-3, n. 7, p. 125-43, Mar. 1977.

[LeFaucher03] LE FAUCHEUR, F. Requirements for support of Differentiated Services-aware MPLS traffic engineering. Internet RFC 3564, Julho, 2003.

[LeFaucher03a] F. Le Faucher, Maximum Allocation Bandwidth Constraints Model for Diff-Serv-aware MPLS Traffic Engineering, draft-ietf-tewg-diff-te-mam-00.txt, Jun. 2003.

[LeFaucher03b] F. Le Faucher, Russian Dolls Bandwidth Constraints Model for Diff-Serv-aware MPLS Traffic Engineering, draft-ietf-tewg-diff-te-russian-03.txt, Jun. 2003.

[Liquidlight01] MPLS lite. Liquidlight. White paper. Extraído em dezembro de 2001 de http://www.liquidlight.com.

[Manna81] MANNA, Z., PNUELI, A. Verification of concurrent programs: temporal proof principles. In: Workshop on logics of programs, 1981. [Proceedings …] [s.l.]: Springer-Verlag, 1981.

[Martins01] MARTINS, Joberto. Itelcom. Qualidade de Serviço (QoS) em redes IP: princípios básicos, parâmetros e mecanismos. Extraído em setembro de 2001 de http://www.jsmnet.com/Downloads/Downloads.html.

[MPLSForum03a] MPLS Forum tackles. Layer 2 service interworking and MPLS uni enhancements. MPLS forum. Extraído em agosto de 2003 de www.mplsforum.org.

[MPLSForum03b] QoS Support in MPLS Networks. MPLS/Frame Relay Alliance White Paper. Maio, 2003. Extraído em agosto de 2003 de www.mplsforum.org.

[MPLSForum03c] MPLS PVC User to Network Interface – Implementation Agreement. MPLS/Frame Relay Alliance. MPLS/Frame Relay Alliance Technical Committee. Maio, 2003. Extraído em agosto de 2003 de www.mplsforum.org.

[Mplsrc01] FAQ MPLS. Extraído em abril de 2001 de www.mplsrc.org.

[Neto87] NETO, João José. Introdução a compilação. Rio de Janeiro: Livros Técnicos e Científicos, 1987.

[Netplane01] Layer 3 switching Using MPLS. White paper. Extraído em abril de 2001 de www.netplane.com.

[Netscout06] METZIER, Jim. The movement to deploy MPLS. Netscout Systems. IT Impact Brief. Extraído em maio de 2006 de http://www.netscout.com/docs/itimpactbriefs.

[Newman96] NEWMAN, P., EDWARDS, W. L., HINDEN, R., HOFFMAN, E., LIAW, F. C., LYON, T., MINSHALL, G. Ipsilon Flow Management Protocol Specification for Ipv4 Version 1.0. Internet RFC 1953, Maio, 1996.

[Nichols98] NICHOLS, K., BLAKE, S., BAKER, F., BLACK, D. Definition of the Differentiated Services Field (DS Field) in the Ipv4 and Ipv6 Headers. Internet RFC 2474, Dezembro, 1998.

[Nortel01b] MPLS – An introduction to multiprotocol label switching. White paper. Nortel Networks. Extraído em abril de 2001 de www.nortelnetworks.com.

[Nortel02a] IP QoS – A Bold New Network. White Paper. Nortel. Extraído em fevereiro de 2002 de www.nortel.com.

[Nwfusion03a] Work underway to better support VoIP over MPLS. Extraído em agosto de 2003 de www.nwfusion.com/newsletters/converg/2003/0707converge2.html.

[Nwfusion03b] MPLS: the ultimate user-to-network interface. Extraído de em agosto de 2003 de www.nwfusion.com/newsletters/frame/2003/0623fr1.html.

[Papelnjack00] PEPELNJACK, Ivan. GUICHARD, Jim. MPLS and VPN arqchitectures. A pratical guide to understanding, designing and deploying MPLS and MPLS-enabled VPNs. Cisco Press: Oct, 2000.

[Physics04] MPLS Plans e Attitudes. Network Physics. White Paper. Extraído em junho de 2004 de www.webtorials.com.

[QoSForum99a] QoSForum. QoS Protocols & Architectures. Extraído em junho de 2001 de http://citeseer.nj.nec.com/forum99qos.html.

[QoSForum99b] QoSForum. The Need for QoS. Extraído em junho de 2001 de http://citeseer.nj.nec.com/forum99qos.html.

[QoSForum99c] QoSForum. IP QoS FAQ. Setembro, 1999. Extraído em junho de 2001 de http://citeseer.nj.nec.com/forum99qos.html.

[QoSForum99d] QoSForum. Introduction to QoS Policies. Extraído em junho de 2001 de http://citeseer.nj.nec.com/forum99qos.html.

[QoSForum99e] QoSForum. Quality of Service – Glossary of Terms. Extraído em junho de 2001 de http://citeseer.nj.nec.com/forum99qos.html.

[Raahemi04] RAAHEMI, B. CHIRUVOLO, G. GE, A., ALI, M. Quality of Services in Metro Ethernet Networks. International Symposium on Communications Interworking 2004. Ottawa, Canada, Novembro, 2004.

[Rekhter97] REKHTER, Y., DAVIE, B., KATZ, D., ROSEN, E., SWALLOW, G. Cisco systems' tag switching architecture overview. Internet RFC 2105, Fevereiro, 1997.

[Rosen01] ROSEN, E., VISWANATHAN, A., CALLON, R. Multiprocol label switching architecture. Internet RFC 3031, Janeiro, 2001.

[Ryan98] RYAN, Jerry. The Technology guide series: Multiprocol label switching. Extraído em setembro de 2001 de www.techguide.com.

[Saltzer01] SALTZER, J. H., REED, D. P., CLARK, D.D. End-to-end arguments in system design. MIT. Extraído em setembro de 2001 de www.reed.com/papers/endtoend.html.

[Stallings92] STALLINGS, W., ISDN and Broadband ISDN with frame relay and atm. Prentice Hall, 1992. 3rd edition.

[SilvaNeto03a] SILVA NETO, Edson Moreira, FIALHO, Sergio Vianna. A Proposal for the Specification of a SVC Service in a MPLS UNI. WSEAS Transactions on Communications. Issues 2 and 3, Volume 2, Julho, 2003.

[SilvaNeto03b] SILVA NETO, Edson Moreira, FIALHO, Sergio Vianna. MPLSoLAN: an Alternative Protocol for Mapping Level Three QoS into Level Two. WSEAS Transactions on Communications. Issue 4, Volume 2, Outubro, 2003.

[Turner93] TURNER, Kenneth J. Using Formal Description Techniques. An introduction to Estelle, Lotos and SDL. England: John Wiley & Sons Ltd, 1993.

[Valenzuela02] VALENZUELA, Sergio González, LEUNG, Victor C. M. QoS Routing for MPLS networks employing mobile agents. IEEE Network, Maio-Julho, 2002.

[Viswanathan01] VISWANATHAN, A., FELDMAN, N., WANG, Z., CALLON, R. Evolution of Multi-Protocol Label Switching. Extraído em outubro de 2001 de http://citeseer.nj.nec.com.

[Viswanathan98] VISWANATHAN, A., FELDMAN, N., BOIVIE, R. ARIS: aggregate Route-Based IP Switching. IBM Technical Report TR29.2353, Fevereiro, 1998. Extraído em outubro de 2001 de http://citeseer.nj.nec.com.

[Xiao02] XIAO, Xipeng, TELKAMP, Thomas, NI, Lionel M. A pratical approach for providing QoS in the internet backbone. Extraído em fevereiro de 2002 de http://www.cse.msu.edu/~xiaoxipe.

[Xiao99] XIAO, Xipeng, NI, Lionel M. Internet QoS: A Big Picture. IEEE Network, Março/Abril, 1999. Extraído em fevereiro de 2002 de http://www.cse.msu.edu/~xiaoxipe.

[Yavatkar00] YAVATKAR, R., HOFFMAN, D., BERNET, Y., BAKER, F, SPEER, M. SBM (Subnet Bandwidth Manager): A protocol for RSVP-based Admission Control over IEEE 802-style networks. Internet RFC 2814, Maio, 2000.

[Zhao01] ZHAO, Weibin, OLSHEFSKI, David, SCHULZRINNE, Henning. Internet Quality of Service: an Overview. Columbia University. Extraído em outubro de 2001 de http://citeseer.nj.nec.com.

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo