Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Computação e Automação Industrial

Rafael Pasquini

Arquitetura e Prototipação de uma UNI Óptica para a Rede GIGA

Campinas, SP

2006

Rafael Pasquini

Arquitetura e Prototipação de uma UNI Óptica para a Rede GIGA

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: **Engenharia de Computação**.

Orientador: Prof. Dr. Maurício Ferreira Magalhães

Campinas, SP

2006

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Pasquini, Rafael
P265a Arquitetura e prototipação de uma UNI óptica para a rede GIGA / Rafael Pasquini. – Campinas, SP: [s.n.], 2006.

Orientador: Maurício Ferreira Magalhães
Dissertação (Mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Redes de computação - Protocolos. 2. Programação (Computadores). 3. Engenharia de tráfego.
4. Comunicações óticas. I. Magalhães, Maurício Ferreira. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título

Título em Inglês:	Architecture and prototype of an optical UNI for the GIGA network
Palavras-chave em Inglês:	GMPLS, UNI, Overlay model, Signaling protocols, Optical networks
Área de concentração:	Engenharia de Computação
Titulação:	Mestre em Engenharia Elétrica
Banca Examinadora:	Eleri Cardozo, Leonardo de Souza Mendes e Marcos Rogério Salvador
Data da Defesa:	28/07/2006

Rafael Pasquini

Arquitetura e Prototipação de uma UNI Óptica para a Rede GIGA

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: **Engenharia de Computação**.

Banca Examinadora:

Prof. Dr. Maurício Ferreira Magalhães - DCA/FEEC/Unicamp
Prof. Dr. Eleri Cardozo - DCA/FEEC/Unicamp
Prof. Dr. Leonardo de Souza Mendes - DECOM/FEEC/Unicamp
Dr. Marcos Rogério Salvador - CPqD

Campinas, SP

2006

Resumo

Os principais interesses do Projeto GIGA são o desenvolvimento de Tecnologias de Redes Ópticas, Serviços e Aplicações de Rede, Serviços Experimentais de Telecomunicações e Aplicações Científicas. No contexto deste Projeto, nosso grupo de pesquisa tem trabalhado no desenvolvimento de um Plano de Controle IP/WDM para integrar a Rede Cliente IP/MPLS com a Rede Óptica de Transporte WDM, de acordo com as especificações GMPLS e ASON. Especificamente neste trabalho é proposta uma Arquitetura para a UNI (*User Network Interface*) independente de protocolo de sinalização para prover a integralização entre as Redes Cliente e a Rede Óptica de Transporte. A Arquitetura da UNI proposta neste trabalho mantém inalterada a semântica de sinalização das redes cliente e de transporte da rede de testes do Projeto GIGA. Esta dissertação descreve como a Arquitetura desenvolvida suporta a independência quanto ao protocolo de sinalização e apresenta o protótipo implementado para validar a Arquitetura proposta.

Palavras-chave: GMPLS, UNI, Modelo *Overlay*, Protocolos de Sinalização, Redes Ópticas.

Abstract

The main interests of the GIGA Project are the deployment of Optical Network Technology, Network Services and Applications, Experimental Telecommunication Services and Scientific Applications. In the context of this Project, our research group has been working on the design of an IP/WDM Control Plane to integrate the IP/MPLS Client Network with the Optical Transport Network (WDM), according to the GMPLS and ASON specifications. Specifically in this work is proposed a signaling protocol independent UNI (User Network Interface) architecture to integrate the Client and the Optical Transport Networks. The proposed UNI architecture maintains unchanged the client and transport networks signaling semantics for the GIGA testbed network. This dissertation describes how the designed architecture supports the independence of the signaling protocol and presents the prototype implemented to validate the architecture.

Keywords: GMPLS, UNI, Overlay Model, Signaling Protocols, Optical Networks.

Agradecimentos

Ao meu orientador, Professor Dr. Maurício Ferreira Magalhães, agradeço a orientação e a oportunidade oferecida.

Aos colegas, Fábio Luciano Verdi e Luiz Gustavo Zuliani, pela ajuda oferecida durante este período.

Aos demais colegas de pós-graduação, Naur J. J. Junior, Carlos A. Miglinsk, Eduardo N. F. Zagari, Rodrigo C. M. do Prado, Rossano P. Pinto, Marcos A. de Siqueira, Mabia D. Cavalcante, Rafael L. Duarte, Fabrízzio C. de Lacerda, Adriane Bellé, Virgínia M. Cardoso, Giselle C. Cardoso, Jane M. Rondina, Paulo M. S. Bueno, Adler C. G. da Silva, Patrícia R. de Toro e Cláudio A. Vieira, pelas críticas, sugestões e principalmente pelos inúmeros momentos de descontração que tivemos.

Aos amigos, Cláudia A. Tambascia, Nelson V. Augusto, Miriam Von Zuben e Eduardo Trettel pelo primeiro contato com o ambiente da FEEC.

A minha família, pelo imenso apoio, em todos os aspectos, durante esta jornada.

Aos meus pais, irmãos e sobrinhos

Sumário

Lista de Figuras

Lista de Tabelas

Glossário

ABR	Available Bit Rate
ASON	Automatically Switched Optical Network
ATM	Asynchronous Transfer Mode
BUS	Broadcast and Unknown Server
CBQ	Class Based Queuing
CBR	Constant Bit Rate
DLL	Data Link Layer
DRR	Deficit Round Robin
DSCP	DiffServ CodePoints
DWDM	Dense Wavelength Division Multiplexing
ELAN	Emulated Local Area Network
FSC	Fiber Switch Capable
GMPLS	Generalized MultiProtocol Label Switching
IETF	Internet Engineering Task Force
IPCC	IP Control Channel
ISI	Internal Signaling Interface
ITU-T	International Telecommunication Union-Telecommunication Standardization Sector
L2SC	Layer 2 Switch Capable
LAN	Local Area Network
LANE	LAN Emulation
LEC	LAN Emulation Client

LECS	LAN Emulation Configuration Server
LES	LAN Emulation Server
LIS	Logical IP Subnet
LMP	Link Management Protocol
LSA	Link State Advertisement
LSC	Lambda Switch Capable
LSDB	Link State Database
LSP	Label Switched Path
LSR	Label Switching Router
MAC	Medium Access Control
MAN	Metropolitan Area Network
MPC	MPOA Client
MPLS	MultiProtocol Label Switching
MPOA	MultiProtocol Over ATM
MPS	MPOA Server
NBMA	Non-Broadcast Multiple Access
NHRP	Next Hop Resolution Protocol
NHS	Next Hop Server
OIF	Optical Internetworking Forum
OSPF	Open Shortest Path First
OSPF-TE	Open Shortest Path First-Traffic Engineering
OXC	Optical Cross Connect
PHB	Per Hop Behavior
PSC	Packet Switch Capable
PXC	Photonic Cross Connect
QoS	Quality of Service

RFC	Request for Comments
RIP	Routing Information Protocol
RSVP	Resource reSerVation Protocol
RSVP-TE	Resource reSerVation Protocol-Traffic Engineering
RWA	Routing and Wavelength Assignment
SDH	Synchronous Digital Hierarchy
SLA	Service Level Agreement
SONET	Synchronous Optical Networks
SRLG	Shared Risk Link Group
TCA	Traffic Control Agreement
TDM	Time Division Multiplexing
TLV	Type, Length, Value
TNA	Transport Network Assigned Address
TOS	Type of Service
UBR	Unspecified Bit Rate
UNI	User Network Interface
VBR	Variable Bit Rate
VLAN	Virtual Local Area Network
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing
WFQ	Weighted Fair Queuing
WRR	Weighted Round Robin

Trabalhos Publicados Pelo Autor

1. R. Pasquini, F. L. Verdi, L. G. Zuliani, M. Magalhães, S. Rossi. “An Optical UNI Architecture for the GIGA Project Testbed Network”. *VI International Telecommunications Symposium - ITS2006*. Fortaleza - CE, Brasil. 3-6 de Setembro de 2006.

2. L. G. Zuliani, M. Savasini, G. S. Pavani, R. Pasquini, F. L. Verdi, M. Magalhães. “An implementation of an OSPF-TE to support GMPLS-controlled All-Optical WDM Networks”. *VI International Telecommunications Symposium - ITS2006*. Fortaleza - CE, Brasil. 3-6 de Setembro de 2006.

3. M. Siqueira, R. Pasquini, F. L. Verdi, R. Duarte, F. C. de Lacerda, M. Magalhães e E. Madeira. “Um Mecanismo Baseado em Web services para Divulgação de Topologias Virtuais Inter-Domínios em Redes GMPLS”. *X Workshop de Gerência e Operação de Redes e Serviços (WGRS 2005) - SBRC 2005*. Fortaleza - CE, Brasil. 9-13 de Maio de 2005.

Capítulo 1

Introdução

Este capítulo introdutório apresenta o Projeto GIGA, quais são seus objetivos e como ele foi subdividido entre grupos de pesquisa de todo o país. Também apresenta a motivação que levou ao desenvolvimento de uma UNI (*User Network Interface*) para o Projeto GIGA e quais são as suas contribuições.

1.1 O Projeto GIGA

Desde 1995 temos presenciado um grande crescimento no tráfego e na oferta de serviços e aplicações relacionadas à Internet. Atualmente, é possível afirmar que o tráfego IP constitui a maior fração de todo volume de tráfego existente. Por exemplo, em outubro de 2002, o IDC (*International Data Corporation*) previu que o tráfego na Internet iria aproximadamente dobrar a cada ano durante os próximos 5 anos [Papadimitriou and Verchere, 2005] e, em abril de 2003, a RHK (*Ryan Hankin Kent Research*) previu que o IP poderia aumentar sua fatia de 50% para aproximadamente 80% do volume total do tráfego gerado até 2006 [Papadimitriou and Verchere, 2005]. Este cenário de forte crescimento no tráfego e na oferta de novos serviços, juntamente com o interesse em firmar conhecimentos na área de redes ópticas, foram os fatores que conduziram à implementação, em maio de 2004, da rede do Projeto GIGA.

O GIGA é um projeto desenvolvido em parceria pela Fundação CPqD (Centro de Pesquisa e Desenvolvimento) e pela RNP (Rede Nacional de Ensino e Pesquisa), com recursos financeiros do FUNTTEL (Fundo para o Desenvolvimento Tecnológico das Telecomunicações) e apoio da FINEP (Financiadora de Estudos e Projetos). A missão do Projeto GIGA é desenvolver tecnologias de Redes e de Serviços de Telecomunicações voltadas para IP/WDM (*Wavelength Division Multiplexing*) e Serviços/Aplicações de Banda Larga. A tecnologia WDM [Agrawal, 2002] utilizada no Projeto Giga associa sinais ópticos a diferentes freqüências de luz (comprimentos de onda ou lambdas) o que

permite separar, dentro de um mesmo meio físico (a fibra óptica), canais diversos para tráfego de dados. O Projeto tem por objetivo capacitar empresas nacionais em tecnologias competitivas, de forma associada com instituições de P&D (Pesquisa e Desenvolvimento), além de fomentar a oferta de novos produtos, protocolos e serviços de telecomunicações à sociedade brasileira, desenvolvendo componentes, dispositivos, equipamentos e soluções para redes ópticas.

Um dos objetivos já alcançados do Projeto é a implantação de uma Rede Experimental (Rede GIGA), utilizando tecnologias ópticas, para transportar quadros Gigabit Ethernet pelos seus enlaces, interligando entidades de pesquisa nas cidades de Campinas, São Paulo, São José dos Campos, Cachoeira Paulista, Rio de Janeiro e Petrópolis numa primeira fase. Outras cidades poderão fazer parte da Rede GIGA em fases posteriores. Em suma, a rede possui 735Km de extensão e interconecta 17 universidades e centros de pesquisa. A Figura 1.1 apresenta a topologia inicial resumida da rede experimental do Projeto GIGA.

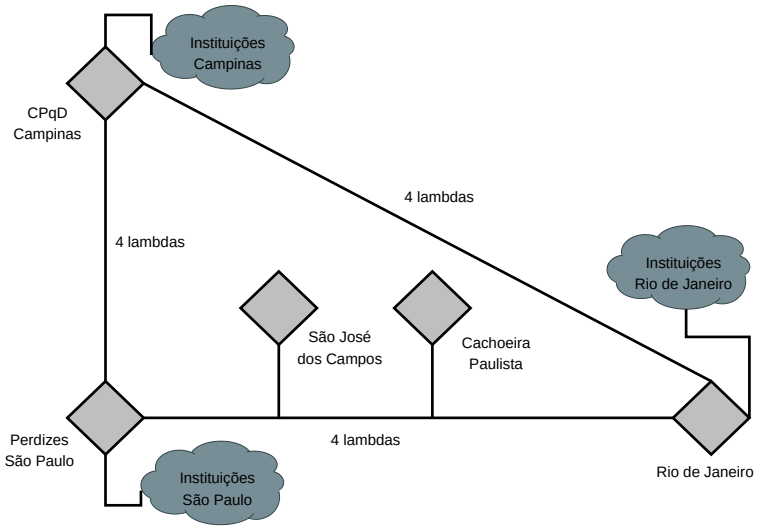

Fig. 1.1: *Backbone* da Rede GIGA na primeira fase.

Desta forma, o Projeto GIGA desenvolve P&D, gerando produtos e serviços em forma de protótipos e irá transformar tecnologia em valor para empresas nacionais. Os produtos e serviços desenvolvidos serão validados na Rede GIGA, que é na verdade uma grande plataforma de testes. O Projeto GIGA é gerenciado através de uma Coordenação Executiva, da qual participam representantes do CPqD e da RNP e através de quatro Coordenações Temáticas nas áreas de:

- Redes Ópticas
- Serviços Experimentais de Telecomunicações
- Protocolos e Serviços de Rede

- Serviços e Aplicações Científicas

As duas primeiras Coordenações estão sob a responsabilidade do CPqD e as duas últimas sob a responsabilidade da RNP. Estas áreas temáticas foram subdivididas entre diversos grupos de pesquisa. Nosso grupo trabalha, em colaboração com outras instituições (Instituto Atlântico e CIN/UFPE), com a Coordenação Temática de Redes Ópticas (CTRO) no desenvolvimento de um Plano de Controle IP sobre WDM para automatizar a rede experimental. Este Plano de Controle deve prover ações como o provisionamento dinâmico de caminhos ópticos e a proteção/restauração automática de caminhos ópticos.

1.2 Motivações

Um dos principais interesses do Projeto GIGA é o desenvolvimento de tecnologias de redes ópticas não proprietárias, ou seja, independentes de plataforma ou fabricante. A maneira de se obter esta independência é através da utilização de protocolos especificados por órgãos de padronização como OIF (*Optical Internetworking Forum*), IETF (*Internet Engineering Task Force*) e ITU-T (*International Telecommunication Union-Telecommunication Standardization Sector*). Entretanto, existem algumas limitações nos elementos da rede cliente do Projeto GIGA que tornam necessárias a utilização de algumas soluções proprietárias para prover a interação entre a rede cliente e a rede óptica de transporte. As limitações dos elementos da rede cliente (equipamentos comerciais) dizem respeito à impossibilidade de introduzir novos módulos/funcionalidades nestes equipamentos, tal como a impossibilidade de implementar uma UNI (*User Network Interface*) diretamente nestes equipamentos.

A tecnologia mais promissora capaz de atender aos requisitos do Projeto GIGA é o conjunto de protocolos GMPLS (*Generalized MultiProtocol Label Switching*) [Mannie, 2004] definido pelo IETF. O GMPLS generaliza o paradigma de comutação por rótulos introduzido pelo MPLS (*MultiProtocol Label Switching*), atendendo elementos de rede que fazem encaminhamento baseado em tecnologias de comutação por comprimento de onda, por fibra e por tempo. O MPLS, originalmente introduzido em redes IP para aumentar o desempenho do encaminhamento de pacotes, introduziu ainda conceitos como QoS (*Quality of Service*) e Engenharia de Tráfego. O GMPLS é um conjunto de protocolos que especifica as funcionalidades a serem atendidas pela sinalização, roteamento e gerência de enlaces. Os protocolos que constituem o Plano de Controle GMPLS são:

- Sinalização: RSVP-TE (*Resource reSerVation Protocol-Traffic Engineering*) [Berger, 2003b];

- Roteamento: OSPF-TE (*Open Shortest Path First-Traffic Engineering*) [Katz et al., 2003];

- Gerência de Enlaces: LMP (*Link Management Protocol*) [Lang, 2005]

Além destes protocolos, uma arquitetura funcional faz-se necessária. O ITU-T desenvolveu uma arquitetura denominada ASON (*Automatically Switched Optical Network*) [ASON, 2001] que especifica quais os componentes necessários para um plano de controle e quais os requisitos funcionais para uma rede óptica de transporte. Esta arquitetura utiliza o modelo *Overlay* na interconexão entre a rede cliente e a rede de transporte. Neste modelo, o cliente não tem nenhuma visibilidade do interior da rede e vice-versa, podendo-se afirmar que a rede de transporte é uma "caixa preta" para a rede cliente. Como conseqüência desta opacidade, o modelo Overlay necessita de uma interface (a UNI - *User Network Interface*) para interconectar a rede cliente com a rede de transporte.

Atualmente existem duas especificações para a UNI, uma feita pelo OIF [UNI Common Part, 2004] e outra feita pelo IETF [Swallow et al., 2005]. Uma das principais características da UNI é a sua independência quanto ao protocolo de sinalização devido a especificação de mensagens abstratas para efetuar suas ações de sinalização. Esta característica possibilita a existência de diferentes protocolos de sinalização dentro e fora da rede de transporte. Outra importante característica da UNI é o fato dela ser dividida em dois módulos, UNI-C (UNI Cliente) e UNI-N (UNI Rede), que são implementados pelos equipamentos de borda da rede cliente e da rede de transporte respectivamente.

O Projeto GIGA utiliza duas versões do RSVP (*Resource reSerVation Protocol*) [Braden et al., 1997] como protocolo de sinalização. Os equipamentos da rede cliente implementam o MPLS RSVP-TE [Awduche et al., 2001a] e o plano de controle da rede óptica de transporte utiliza o GMPLS RSVP-TE [Berger, 2003a, Berger, 2003b]. Como citado anteriormente, não é possível introduzir uma UNI nos equipamentos que constituem a rede cliente do Projeto GIGA. Devido a esta limitação estamos especificando uma nova Arquitetura para a UNI, pois, nas especificações existentes da UNI, para que a rede cliente possa solicitar conexões para a rede óptica de transporte, ela deve estar ciente da existência da UNI e, conseqüentemente, conhecer a API oferecida por ela.

1.3 Contribuições

O principal requisito da arquitetura apresentada neste trabalho é prover a comunicação entre a rede cliente e a rede de transporte de forma transparente, mantendo sempre a independência e a semântica dos protocolos de sinalização das redes envolvidas. Atendendo a este requisito, a principal contribuição deste trabalho é a Arquitetura de implementação proposta para a UNI do Projeto GIGA. Entretanto, este trabalho também possui outras contribuições que são rapidamente apresentadas nesta seção juntamente com alguns dos requisitos impostos ao desenvolvimento da arquitetura.

Atender à Rede GIGA significa definir uma arquitetura que seja capaz de solucionar as restrições impostas pelos equipamentos utilizados na rede do Projeto GIGA, principalmente a impossibilidade de adicionar novas funcionalidades (UNI) nestes equipamentos. Embora estas restrições dificultem o desenvolvimento da arquitetura, é preciso desenvolver uma arquitetura para a UNI que utiliza o menor número possível de soluções proprietárias, ou seja, seguindo ao máximo as especificações feitas pelos órgãos de padronização OIF e IETF.

As duas versões para a UNI (UNI OIF e UIN IETF, Capítulo 3) foram profundamente estudadas para que fosse possível apontar qual é a mais adequada ao Projeto GIGA e, até mesmo, quais são as extensões necessárias para atender às restrições do GIGA. Este estudo é uma das contribuições deste trabalho.

A arquitetura desenvolvida neste trabalho pode ser utilizada em outros ambientes que utilizam a UNI, pois a arquitetura proposta pode ser utilizada em casos onde os equipamentos que constituem a rede permitem ou não a inserção da UNI diretamente neles.

1.4 Organização do Texto

O Capítulo 2 introduz os conceitos básicos relacionados a este trabalho. Ele apresenta algumas das soluções que foram criadas por órgãos como IETF, ITU-T e ATM-Fórum para atender à crescente demanda por largura de banda devido ao surgimento constante de novas aplicações. Neste capítulo são apresentados os modelos existentes de interconexão de redes (*Overlay, Peer e Augmented*) e os protocolos que estão relacionados ao plano de controle da rede óptica do Projeto GIGA.

O Capítulo 3 tem como objetivo apresentar as especificações existentes da UNI feitas pelo OIF e IETF. Neste capítulo, as características e os mecanismos da UNI são descritos da seguinte maneira: a UNI do OIF e a UNI do IETF possuem muitas características e funcionalidades em comum, conseqüentemente, na seção onde a UNI do OIF é apresentada, todas as características e procedimentos da UNI são descritos. A seguir, há uma seção que introduz a UNI do IETF. Nesta seção, apenas as extensões feitas à UNI do OIF por parte do IETF são apresentadas, ou seja, todo o restante das características da UNI do IETF são idênticas às da UNI do OIF apresentadas na primeira seção do capítulo. Para finalizar o capítulo 3, há uma seção onde é feita uma comparação entre as duas especificações da UNI.

O Capítulo 4 traz a Arquitetura da UNI proposta para a rede óptica do Projeto GIGA e apresenta em detalhes cada um dos módulos da arquitetura. Neste capítulo são apresentadas as restrições impostas pelo Projeto GIGA que levaram à especificação de uma UNI própria para o projeto. Este capítulo possui ainda uma seção que detalha a utilização da arquitetura da UNI proposta no cenário real da rede do Projeto GIGA.

Um protótipo foi desenvolvido para verificar o funcionamento da arquitetura proposta. O Capítulo 5 apresenta este protótipo e detalha o desenvolvimento de cada um de seus módulos. Traz ainda, os resultados das simulações onde são verificados o funcionamento da arquitetura e alguns testes de desempenho que foram obtidos através de simulações no protótipo.

O Capítulo 6 apresenta as conclusões do trabalho realizado e possíveis trabalhos futuros. Logo após estão as referências utilizadas.

Dois apêndices finalizam este trabalho. O Apêndice A apresenta os diagramas de seqüência que descrevem o estabelecimento e a remoção de conexões na Arquitetura. O Apêndice B apresenta os diagramas de classes dos módulos implementados no protótipo.

Capítulo 2

Conceitos Fundamentais

Este capítulo faz um resgate dos conceitos fundamentais relacionados a este trabalho. O capítulo está dividido em três seções e o seu principal objetivo é apresentar a evolução dos mecanismos desenvolvidos para prover QoS e Engenharia de Tráfego em redes.

A primeira seção traz algumas das soluções que foram criadas por instituições como IETF, ITU-T e ATM-Fórum no início dos anos 90 a fim de prover QoS em redes IP e redes ATM (*Asynchronous Transfer Mode*) e apresenta ainda o MPLS e o GMPLS, que são duas especificações utilizadas no Projeto GIGA.

A segunda seção faz um breve detalhamento dos protocolos especificados pelo GMPLS e que constituem o plano de controle da rede do Projeto GIGA.

A terceira seção apresenta os três modelos de interconexão existentes (*Overlay, Peer e Augmented*) na literatura. Dentre estes modelos de interconexão, o utilizado no Projeto GIGA é o *Overlay*.

2.1 Soluções de Qualidade de Serviço (QoS)

A Internet teve início em meados dos anos 80 com a oficialização do TCP/IP como única pilha de protocolo a ser utilizado pelas redes existentes naquela época. Em 1990 já haviam 3 mil redes com aproximadamente 200 mil computadores conectados e, em 1992, o milionésimo computador foi conectado à rede [Tanenbaum, 1997]. Com tamanho crescimento, foi fundada em 1992 a Internet *Society*, com a finalidade de promover o uso da Internet.

A Internet é uma rede de melhor esforço que em seu início possuía quatro aplicações principais:

- Correio eletrônico (*e-mail*);
- Notícias (*News*);

- *Login* remoto;
- Transferência de arquivos (FTP).

Com a proliferação do WWW e o aparecimento dos primeiros navegadores a Internet se popularizou, deixando de ser utilizada apenas pelos meios científicos e governamentais. Com esta popularização e o surgimento de novos serviços e aplicações a Internet passou a enfrentar problemas como congestionamento. Ser uma rede que oferecia apenas um serviço de melhor esforço não era mais suficiente. Era necessário criar soluções que mantivessem a rede operacional e escalável, privilegiando novos serviços antes não suportados. De fato, várias soluções foram criadas. A seguir são tratadas algumas destas soluções para prover QoS na rede.

2.1.1 A Arquitetura de Serviços Integrados (IntServ)

A Arquitetura de Serviços Integrados, definida pelo IETF em [Braden et al., 1994], buscava privilegiar serviços de tempo real que não recebiam tratamento adequado em função das variações de atraso (*jitter*) presentes na Internet, que ocorrem devido ao armazenamento temporário de pacotes nos roteadores e as perdas causadas por congestionamentos. A arquitetura tinha como objetivo estender a arquitetura original da Internet através da introdução de novos componentes e mecanismos que iriam complementar a arquitetura original sem, no entanto, implicar a substituição do serviço IP original. Basicamente, o IntServ busca prover um controle fim-a-fim aos atrasos experimentados pelos pacotes, utilizando-se de mecanismos que visam controlar a banda-passante dos enlaces através da divisão deste tráfego em algumas classes administrativas e atribuindo a cada uma destas classes um percentual mínimo da banda passante do enlace. Este mecanismo é chamado de compartilhamento controlado do enlace (*controlled link-sharing*). A Arquitetura de Serviços Integrados conceitualmente estende o modelo original da Internet através de 2 elementos:

- Modelo de Serviços Integrados (*Integrated Service*);
- *Framework* de Implementação.

O Modelo de Serviço Integrado (IS) define o comportamento visível externamente e inclui duas classes de serviços: serviço garantido e serviço de carga controlada. Deve ainda operar tanto nos contextos *multicast* e *unicast*. A introdução do modelo IS cria a necessidade de roteadores capazes de reservar recursos a fim de prover QoS a fluxos de pacotes específicos, ou seja, é necessária a utilização de mecanismos de reserva capazes de fazer a manutenção dos estados associados a estes fluxos em todos os roteadores. Este novo cenário representa uma grande mudança no modelo original da Internet, onde todos os estados associados aos fluxos são mantidos nos sistemas finais (*end systems*),

tornando a Internet muito robusta [Magalhães and Cardozo, 2003b]. A maneira de se preservar esta robustez, mesmo com a necessidade de manter os estados dos fluxos nos roteadores, é a utilização do conceito *soft state* [Braden et al., 1997] empregado pelo protocolo de sinalização RSVP.

O *Framework* de implementação procura definir a organização dos programas que suportam o modelo IS através de quatro componentes básicos à implementação do IntServ:

1. Escalonador de Pacotes: gerencia o encaminhamento dos diferentes fluxos de pacotes através de um conjunto de filas e temporizadores; possui um estimador responsável por medir o tráfego de saída e gerar estatísticas de avaliação comportamental do escalonador e do controle de admissão;

2. Controle de Admissão: utilizado nos roteadores e nós terminais para determinar se o nível de QoS requisitado pode ser concedido a um novo fluxo sem que haja o comprometimento dos outros fluxos existentes. Efetua ainda ações administrativas como a autenticação das solicitações de reserva e funções de tarifação. Não devemos confundi-lo com o policiamento de tráfego que é realizado nas bordas da rede a fim de garantir que os nós terminais não estão violando o perfil de tráfego ao qual haviam se comprometido a gerar;

3. Classificador: atua com base nos cabeçalhos dos pacotes. Sua função é separar os fluxos pertencentes à mesma classe de serviço e colocá-los na fila correta. Uma classe corresponde a uma abstração que pode ser local a um determinado roteador. Por exemplo, em um *backbone* os roteadores podem agregar vários fluxos em algumas poucas classes, enquanto os roteadores próximos à periferia podem utilizar uma classe para cada fluxo;

4. Protocolo para Estabelecimento de Reservas: é necessário para a criação e manutenção do estado de cada fluxo nos roteadores e nós terminais ao longo do caminho percorrido pelo fluxo. A Arquitetura de Serviços Integrados utiliza o RSVP que será discutido na Seção 2.2.1.

A Figura 2.1 ilustra estes componentes em um roteador IP que suporta IntServ. A figura tem uma divisão funcional representada pelo encaminhamento na parte inferior da figura, e pelas atividades de *background* representadas na parte superior da figura.

2.1.2 A Arquitetura de Serviços Diferenciados (DiffServ)

A Arquitetura de Serviços Diferenciados, proposta pelo IETF em [Blake et al., 1998], constitui-se numa solução mais simples que a Arquitetura de Serviços Integrados. O que motivou o surgimento do DiffServ foi o receio de que o IntServ não seria escalável em *backbones* onde poderiam existir

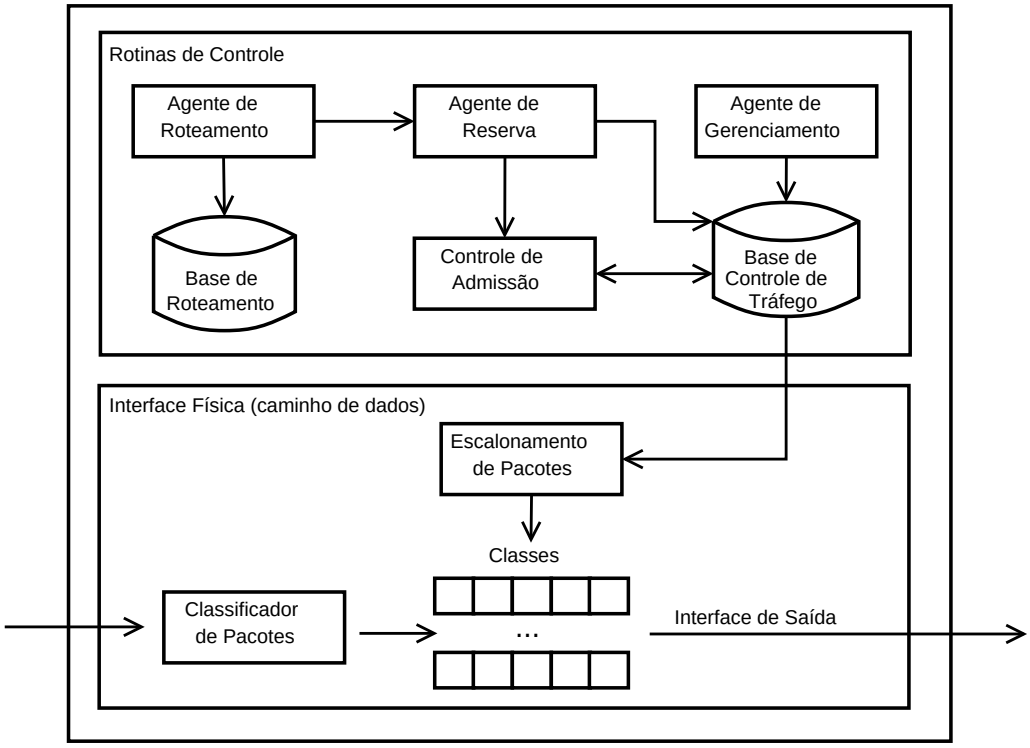

Fig. 2.1: Estrutura do Roteador na Arquitetura de Serviços Integrados.

milhares, ou até mesmo milhões, de fluxos, devido à necessidade do IntServ de manter estados para todos os fluxos.

O DiffServ possui alta escalabilidade pois agrega fluxos em umas poucas classes de serviço utilizando a classificação como mecanismo para prover QoS. Utiliza campos já existentes do cabeçalho IP, campo TOS (*Type of Service*) do datagrama IPv4 ou o campo Classe de Tráfego do datagrama IPv6 para classificar os pacotes. Emprega um mecanismo de priorização denominado "Comportamento por *Hop*" (PHB). PHBs definem padrões de comportamento para que os roteadores efetuem o encaminhamento de datagramas. Se todos os roteadores do caminho implementam o DiffServ, o encaminhamento de um datagrama marcado com um determinado PHB será uniforme, independente do fabricante ou modelo do roteador. Um PHB agrega fluxos gerados por diferentes aplicações, definindo assim uma classe de serviço. Esta definição é flexível, sendo função de grandezas como:

- recursos (banda, *buffer*);
- parâmetros de QoS (atraso, *jitter*, taxa de perda);
- presença de outros PHBs.

PHBs só fazem sentido em conjunto, ou seja, um PHB que utiliza X% da banda de um enlace requer ao menos outro PHB que defina como o restante da banda será utilizada. O DiffServ utiliza DSCPs (DiffServ *codepoints*) para identificar PHBs. O DSCP utiliza 6 bits do campo TOS (IPv4) ou do campo Classe de Tráfego (IPv6) que é compatível com as especificações feitas pelo IETF nas RFCs (*Request for Comments*) 1122 [Braden, 1989] e 1349 [Almquist, 1992]. A Figura 2.2 ilustra este campo.

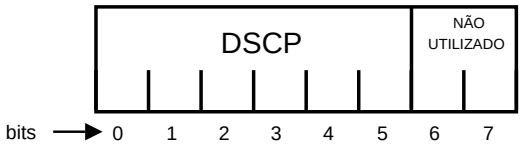

Fig. 2.2: Campo DSCP do DiffServ.

A Figura 2.3 apresenta a Arquitetura de Serviços Diferenciados quanto as suas funcionalidades. Classificadores selecionam pacotes em função de informações contidas no cabeçalho e de regras de classificação, podendo ser de dois tipos:

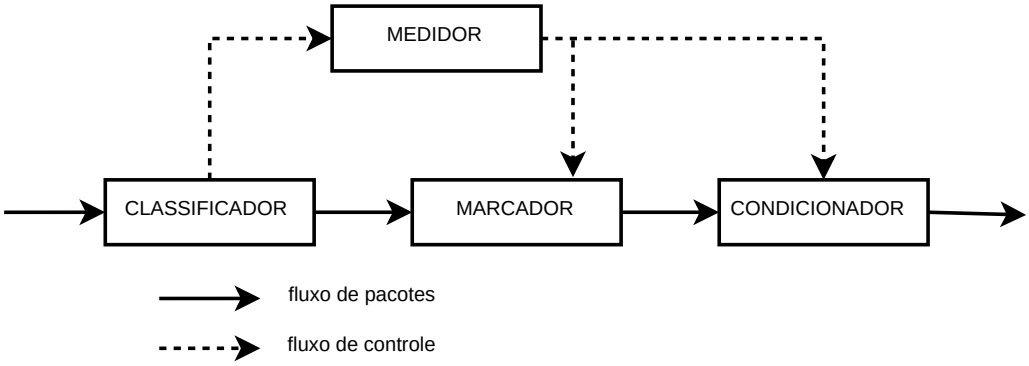

Fig. 2.3: Arquitetura Funcional do DiffServ.

1. Agregador (BA - *Behavior Aggregate*): classifica datagramas baseando-se apenas em *codepoints*.

2. Múltiplo (MF - *Multi Field*): classifica datagramas com base em um ou mais campos do cabeçalho (*codepoint*, endereços IP, protocolo e *ports* de transporte); bem como pela interface que recebeu o datagrama e hora do dia.

O Classificador é responsável por encaminhar os datagramas recebidos em função do PHB ao qual estão associados. Ele respeita o SLA/TCA (*Service Level Agreement/Traffic Control Agreement*)

vigente. Não estando associado a nenhum PHB, o datagrama receberá um PHB padrão referente ao serviço de melhor esforço.

O Medidor afere as propriedades temporais do tráfego, comparando com as especificações definidas no SLA/TCA para cada classe de serviço. Geralmente, implementam algoritmos do tipo *token bucket*. Os resultados das medições podem ser armazenados em MIBs (*Management Information Bases*) para acompanhamento da gerência da rede.

O Marcador atribui um *codepoint* ao datagrama em função do estado informado pelo medidor podendo, inclusive, ter o *codepoint* alterado. Por exemplo, os datagramas excedentes ao tráfego de uma determinada classe poderão ser marcados com o *codepoint* associado ao serviço de melhor esforço.

O Condicionador altera as características temporais de um tráfego não conforme. A conformidade é obtida através da introdução de atrasos no tráfego excedente. Não sendo possível efetuar o condicionamento total do tráfego, ele passa a descartar pacotes.

Os PHBs são especificados pelo IETF através de RFCs baseando-se em critérios presentes na RFC 2475 [Blake et al., 1998]. A seguir, temos alguns destes critérios que estipulam que a criação de PHBs ou grupos de PHBs deve definir precisamente:

- os *codepoints* empregados pelo PHB ou grupo;
- o tipo de serviço que o PHB ou grupo atende;
- as restrições para a implementação do PHB e as conseqüências caso sejam violadas;
- informações de configuração, gerência e segurança dos equipamentos que implementam o PHB;
- se, e em que condições, é permitido remarcar datagramas.

Existem três grupos de PHBs especificados pelo IETF em RFCs:

1. Seletor de Classe: especificado na RFC 2474 [Nichols et al., 1998], destina-se a manter a compatibilidade com o sub-campo Precedência que ocupa os três primeiros bits do campo TOS do IPv4 preservando os esquemas de roteamento e encaminhamento que levam este sub-campo em consideração. A RFC define que o valor 000000 é reservado para o serviço de melhor esforço (BE - *Best Effort*); o valor 110000 e 111000 representam datagramas de controle e devem receber tratamento preferencial; define ainda que quanto maior o valor do *codepoint* mais prioritário é o tráfego associado a ele.

2. Encaminhamento Garantido (AF - *Assured Forward*): especificado na RFC 2597 [Heinanen et al., 1999], destina-se a oferecer um serviço "melhor que melhor esforço". Não define qualquer limite para atraso e *jitter* mas apresenta parâmetros de QoS superiores ao serviço de melhor esforço. Basicamente, ele habilita um provedor a oferecer classes de serviços de encaminhamento diferenciadas, cada qual com três níveis de descarte. A RFC sugere ainda o uso do algoritmo RED (*Random Early Drop*) para efetuar o controle de congestionamento. A Tabela 2.1 apresenta um exemplo do grupo AF com quatro classes de serviço e três níveis de descarte cada uma.

3. Encaminhamento Expresso (EF - *Expedited Forwarding*): especificado na RFC 2598 [Jacobson et al., 1999], destina-se à implementação de serviços com baixas taxas de perda, latência e *jitter*, além de banda garantida. Pode ser chamado de "linha privada virtual" ou serviço *premium*. Entretanto, este modelo é severo com o tráfego não conforme: simplesmente o descarta. Pode implementar diversas políticas de escalonamento como: DRR, WRR, WFQ e CBQ. Detalhes sobre estas e outras políticas de escalonamento podem ser encontradas em [Keshav, 1997]. O *codepoint* utilizado pela classe EF é 101110.

Tab. 2.1: *Codepoints* para o grupo de PHBs AF.

Prob. Descarte	**Classe 1**	**Classe 2**	**Classe 3**	**Classe 4**
Baixa	001010	010010	011010	100010
Média	001100	010100	011100	100100
Alta	001110	010110	011110	100110

2.1.3 Modelos IP sobre ATM

Em meados dos anos 90 houve um forte crescimento na Internet. Todo este crescimento levou ao surgimento de novas tecnologias com maiores capacidades de transmissão a fim de atender a crescente demanda por banda e suportar as novas aplicações. Uma tecnologia que se destacou neste cenário foi o ATM, utilizada em *backbones* de alta velocidade para suportar o tráfego de dados entre redes locais e no transporte de dados em redes de longa distância.

Uma infra-estrutura de chaves ATM, denominada nuvem ATM, deve ser entendida como um ambiente sobre o qual vários serviços são oferecidos, serviços estes que correspondem ao tráfego de dados, vídeo e voz. É possível afirmar que nesta infra-estrutura são utilizados equipamentos ATM de diversos fabricantes. Tendo isto em vista, instituições como o IETF, ITU-T e o ATM-Fórum empenharam-se no desenvolvimento de padrões voltados à utilização da tecnologia ATM na interconexão de redes.

O ATM é caracterizado como uma tecnologia orientada a conexão, incapaz de fazer *broadcast* (NBMA - *Non-Broadcast Multiple Access*) e que utiliza células de tamanho fixo na sua transmissão de dados. Todas estas características dificultam a utilização do IP sobre uma rede ATM, demandando a utilização de mecanismos para encapsular datagramas IP em células ATM (AAL - *ATM Adaptation Layer*) e para fazer a resolução de endereços IP em endereços ATM (ATMARP - *ATM Address Resolution Protocol*). O grupo de trabalho IP sobre ATM do IETF definiu um protocolo para fazer a resolução automática de endereços IP denominado IP Clássico sobre ATM [Laubach, 1994].

O propósito desta especificação é permitir implementações compatíveis e interoperáveis para a transmissão de datagramas IP e a resolução de endereços sobre a camada de adaptação ATM AAL5. Proposto para ambientes onde um grupo de computadores usa uma rede ATM ao invés de uma rede local, forma uma sub-rede lógica IP (LIS). Permite ainda a definição de múltiplas LISs. A Figura 2.4 ilustra uma rede ATM onde foram definidas duas LISs.

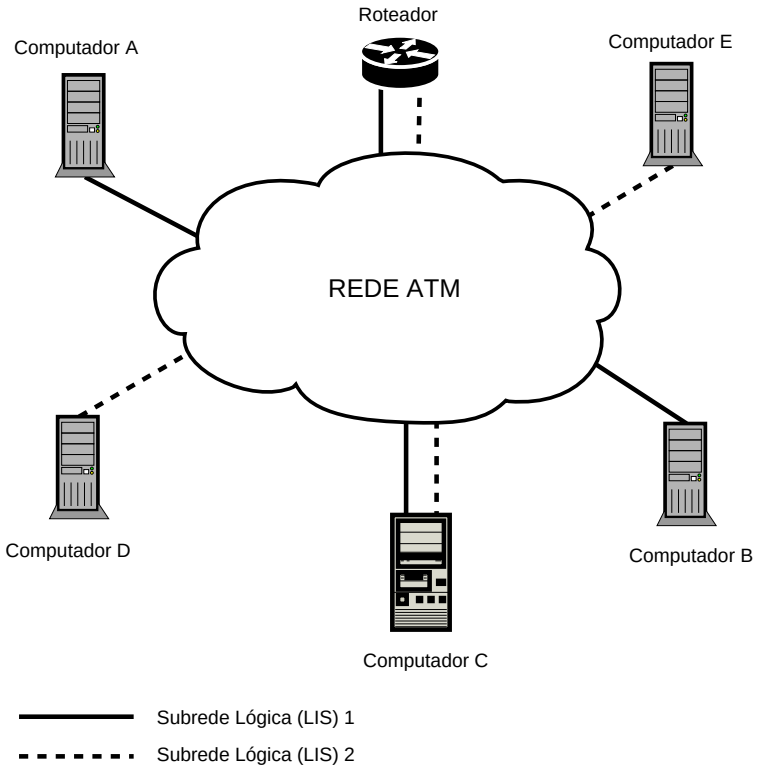

Fig. 2.4: Exemplo do IP Clássico sobre ATM.

Todos os computadores estão conectados na mesma rede ATM, sendo os computadores A, B e C participantes da LIS 1 e os computadores C, D e E participantes da LIS 2. Cada LIS forma conceitualmente uma rede separada, compartilhando um mesmo prefixo de sub-rede IP. Computadores que participam de uma mesma LIS podem estabelecer circuitos virtuais entre eles

para troca de datagramas. Computadores que pertencem a LISs diferentes não podem se conectar diretamente, a comunicação entre eles é feita através de um roteador que participa de ambas as LISs. Cada estação IP deve conhecer seu próprio endereço ATM e o endereço do servidor ATMARP.

A resolução de endereço dentro de uma LIS faz o uso de um protocolo de resolução de endereço ATM (ATMARP), bem como do protocolo de resolução de endereço ATM inverso (InATMARP). O ATMARP e o InATMARP são respectivamente o protocolo ARP e o InARP com as extensões necessárias para suportar um ambiente ATM (*unicast*).

Apesar desta estrutura desenvolvida, a operação do IP Clássico sobre ATM é muito simples, levando-o a muitas limitações. Não possibilita que um datagrama destinado à outra LIS seja entregue diretamente a seu destino sem antes passar por um roteador padrão, o que gera uma ineficiência na rede devido aos gargalos formados pelos roteadores. O Modelo Clássico também não se preocupa com a questão da latência no estabelecimento de conexões virtuais. E por último não suporta *Multicast*.

O IETF fez propostas para contornar estas restrições. Definiu uma arquitetura para possibilitar o *Multicasting* em redes ATM através da utilização de um servidor MARS (*Multicast Address Resolution Server*). Especificou ainda o protocolo NHRP (*Next Hop Resolution Protocol*) que faz resolução de endereços e permite a entrega de datagramas diretamente a destinos vinculados a redes lógicas diferentes. Detalhes sobre estas especificações podem ser encontradas em [Talpade and Ammar, 1997] e [Luciani, 1998] respectivamente.

2.1.4 Modelo de Emulação de LAN (LANE)

Este modelo foi proposto pelo ATM-Fórum. Seu requisito básico é preservar os protocolos atuais que rodam no nível de rede inalterados caso sejam executados em uma rede *Ethernet* ou *Token-Ring*, ou mesmo, caso sejam executados sobre uma infraestrutura ATM (nuvem ATM).

A maior dificuldade em oferecer um serviço de emulação de LAN sobre uma nuvem ATM está no fato de que as redes locais tradicionais e a tecnologia ATM possuem diferenças marcantes. A principal diferença consiste no fato de que as LANs não são orientadas à conexão, enquanto as redes ATM oferecem serviços orientados à conexão. Além disso, o serviço de emulação de LAN tem que ser capaz de suportar o uso de endereços MAC *unicast*, *broadcast* e *multicast*, pois são características nativas das LANs. Porém, o fornecimento destes serviços em uma rede baseada em chaves com enlaces ponto a ponto não é trivial (NBMA).

Uma ELAN suporta a comunicação de quadros de dados similar a uma LAN física convencional. É possível ainda ter mais de uma ELAN em uma mesma rede ATM, sendo cada uma das ELANs independentes, ou seja, as estações não podem se comunicar diretamente através das ELANs sem o uso de um roteador. Cada ELAN é composta de clientes de emulação de LAN (LEC) e um serviço de

emulação de LAN. Este serviço por sua vez é composto de um servidor de configuração LE (LECS), um servidor de LAN emulada (LES) e um servidor de *Broadcast* (BUS). A Figura 2.5 ilustra estes componentes que são tratados a seguir:

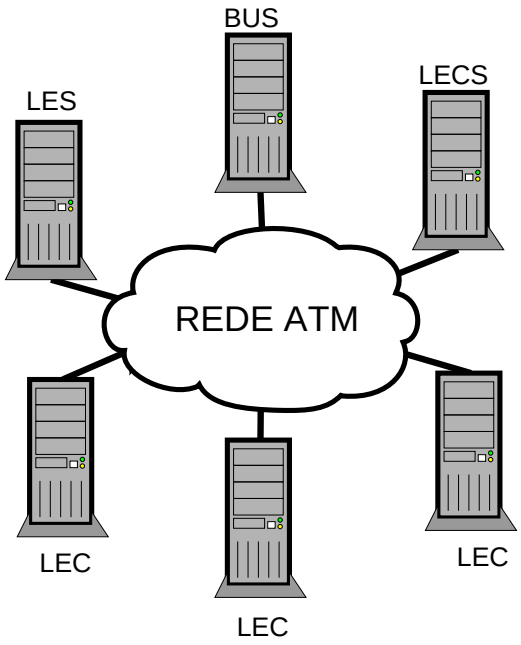

Fig. 2.5: Componentes de uma ELAN.

- Cliente de Emulação de LAN (LEC): ele é responsável por resolver endereços, enviar dados, executar funções de controle para um sistema final em uma única ELAN e prover uma interface de serviço LAN para as entidades de camada mais alta. Um sistema final que se encontra conectado a mais de uma ELAN deve ter um LEC em cada uma. É identificado por um endereço ATM único que é associado a um ou mais endereços MAC. Por exemplo, uma placa de rede possui apenas um endereço MAC e um *switch* possui um endereço MAC em cada uma de suas portas. O LEC pode gerar mensagens de controle ou mensagens de dados. As mensagens de dados podem ser entregues por conexão direta entre dois LECs (sessões *unicast*) ou entregues ao BUS (sessões *broadcast* e *multicast*).

- Servidor de Emulação de LAN (LES): implementa funções de controle para uma ELAN. Responsável pelo registro e resolução de endereços para LECs, existe apenas um LES por ELAN que é identificado por um endereço ATM único. Os LECs devem registrar junto ao LES o seu endereço MAC e as LANs que eles representam. O LES é consultado quando um LEC necessitar resolver um endereço MAC em um endereço ATM e/ou um descritor de rota em um endereço ATM.

- Servidor de *Broadcast* (BUS): as redes locais são baseadas em envio *broadcast* e é necessário prover este serviço na ELAN. Cada LEC está ligado ao BUS através de uma conexão ponto-a-ponto. O BUS envia mensagens *broadcast*, tais como mensagens de resolução de endereços IP, através de uma conexão ponto-multiponto na qual o BUS é a raiz e os LECs são folhas. Esta estrutura permite aos LECs enviarem quadros antes que o endereço ATM do LEC de destino seja resolvido e a conexão direta entre LECs seja estabelecida. Na Figura 2.6, é possível verificar o modo de operação do BUS. Cada LEC está associado a um único BUS em cada ELAN, o LEC obtém o endereço do BUS ao qual ele deve associar-se no LES.

- Servidor de Configuração de Emulação de LAN (LECS): é responsável pela associação dinâmica de diferentes LECs a diferentes ELANs e pela manutenção de uma base de dados contendo as associações resultantes. Existe apenas um LECS por domínio administrativo que serve a todas as ELANs existentes no domínio.

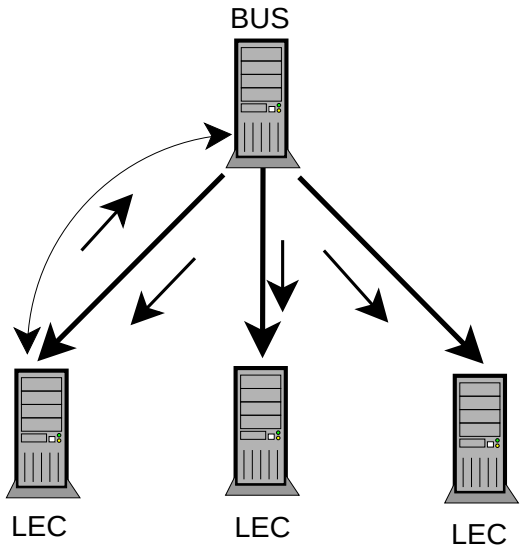

Fig. 2.6: Mensagens *broadcast* gerenciadas pelo BUS.

A transferência de dados em uma ELAN é feita através de conexões diretas entre os LECs ou através da conexão dos LECs com o BUS. Para que um LEC possa estabelecer uma conexão direta com outro LEC ele precisa resolver, via LES, o endereço ATM do LEC ao qual deseja se conectar. Em paralelo a esta resolução de endereços e o estabelecimento do canal direto entre os LECs é possível transmitir dados, via BUS, em *broadcast*. Para evitar o uso abusivo da função de *broadcast* desempenhada pelo BUS é estabelecido um limite de quadros aos LECs por unidade de tempo. Uma vez resolvido o endereço do LEC de destino e estabelecida a conexão ATM entre eles, o tráfego

de dados passa a ser feito diretamente entre os LECs envolvidos. Mas, nos casos de transmissões *multicast*, é realmente necessária a utilização do BUS.

O Modelo LANE é capaz de atender as necessidades das redes conectadas à rede ATM através do fornecimento de um serviço de *broadcast* e um ambiente semelhante ao de uma LAN tradicional. A estrutura é responsável pelo estabelecimento das conexões entre os LECs. Desta forma a camada ATM é transparente para a camada de rede.

Basicamente, existem dois aspectos ineficientes na proposta do LANE. O primeiro diz respeito à transferência de dados entre LECs que pertencem a ELANs diferentes. Para efetuar esta transferência é obrigatório o uso de roteadores entre ELANs, gerando novamente um gargalo nos pontos de interconexão de ELANs e tornando a estrutura de transferência mais rígida. O segundo aspecto negativo é o não aproveitamento das classes de serviço (UBR, ABR, VBR, CBR) oferecidas pelo ATM que garantem a qualidade de serviço.

2.1.5 MPOA

Os modelos apresentados anteriormente (IP/ATM e LANE), assim como o MPOA, utilizam uma nuvem ATM como rede de transporte para o tráfego das redes locais vinculadas a nuvem. O modelo IP/ATM possui aspectos ineficientes que de certa forma foram resolvidos pelo modelo LANE. Entretanto, o LANE não aproveita totalmente a estrutura fornecida pelo ATM. Neste contexto temos o MPOA (*MultiProtocol Over ATM*) como um modelo mais complexo que aproveita a estrutura oferecida pelo ATM.

O MPOA consiste em uma estrutura de comunicação que permite a sobreposição de protocolos de rede (camada 3) sobre uma base ATM através da utilização de um protocolo de emulação de rede local utilizado para estabelecer VLANs sobre a nuvem ATM e para comunicação com as redes locais tradicionais. Utiliza ainda o protocolo NHRP [Luciani et al., 1998] para a resolução de endereços e, conseqüentemente, otimizar a comunicação entre *hosts* ATM e demais clientes MPOA através do estabelecimento de conexões virtuais chamadas de atalhos (*shortcuts*). Estas conexões eliminam o gargalo causado pelo uso de roteadores na interconexão entre VLANs, característica esta não resolvida pelos outros modelos (IP/ATM e LANE).

O MPOA implementa o conceito de "Roteador Virtual"como ferramenta de otimização do uso de caminhos virtuais e redução nos tempos de atraso. Esta funcionalidade separa a tarefa de encaminhamento de pacotes da tarefa de roteamento. O MPOA procura resolver qualquer protocolo de camada 3 especificamente sobre o ATM ao contrário das propostas feitas pelo IETF que visam resolver o IP sobre redes NBMA. O serviço MPOA é constituído de dois módulos básicos: Cliente MPOA (MPC) e Servidor MPOA (MPS) que são apresentados na Figura 2.7.

O Cliente MPOA pode ser implementado em dois tipos de dispositivos: *hosts* ATM e dispositivos

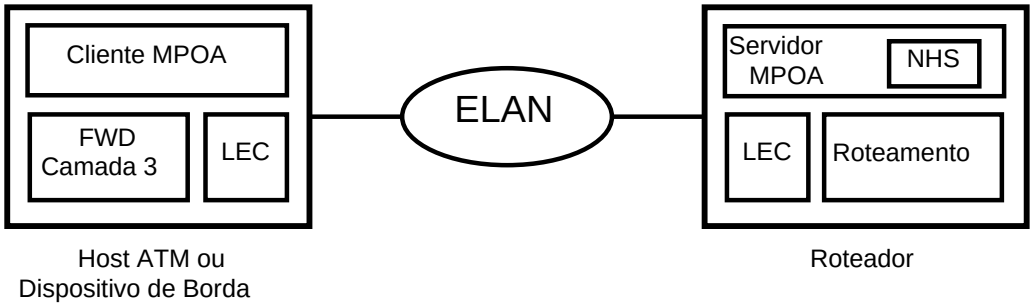

Fig. 2.7: Componentes do Sistema MPOA.

de borda utilizados na conexão de redes locais legadas (*legacy* LANs). O Servidor MPOA é implementado nos roteadores e tem como co-residente o servidor NHS (*Next Hop Server*) [Luciani et al., 1998].

A principal função do MPC é enviar e receber informações nos atalhos (*shortcuts*) estabelecidos entre dois clientes MPOA. O MPC não é responsável pela execução de protocolos de roteamento. Ao detectar na entrada da nuvem ATM um fluxo de pacotes que deve ser encaminhado através do serviço LANE, ele solicita ao NHRP presente no MPS ao qual está conectado, para que determine um atalho para o destino daquele fluxo. Estabelecido o atalho, os pacotes do cliente passam a ser encaminhados diretamente. Do outro lado da nuvem ATM, na saída do fluxo de informações que atravessou o sistema MPOA, temos um MPC recebendo estes pacotes de dados. Os pacotes recebidos pelo MPC são acrescidos do DLL (*Data Link Layer*) adequado e enviados para a interface LAN de destino daquele pacote (cliente destino). A informação a ser utilizada, relativamente ao DLL adequado, é fornecida pelo MPS ao MPC e armazenada no *cache* deste último.

O MPS é um componente lógico presente nos roteadores que tem como objetivo fornecer informações de encaminhamento no nível da camada 3 para os MPCs. O MPS interage com o NHS e o roteador no momento da consulta realizada pelo MPC para um fluxo de entrada, e fornece informações sobre o encapsulamento DLL ao MPC relativo ao fluxo de saída.

O conceito de estabelecimento de atalhos (*shortcuts*) utilizado pelo MPOA pode ser considerado como um de seus principais atrativos. Através dos atalhos é possível otimizar a entrega de informações entre clientes MPOA mesmo quando os clientes pertençam a diferentes ELANs. Esta função elimina os gargalos gerados pelos roteadores em modelos como IP/ATM e LANE tornando a transferência de informações menos rígida. Na Figura 2.8 é possível verificar como ocorre a transmissão de dados por um caminho padrão e por um atalho.

Outro conceito utilizado pelo MPOA é o conceito de Fluxo. Os pacotes de informações que não caracterizam um fluxo são transmitidos de forma normal pelo MPOA, ou seja, um serviço semelhante ao oferecido pelo modelo LANE. Somente após a caracterização de um fluxo que o MPC solicita

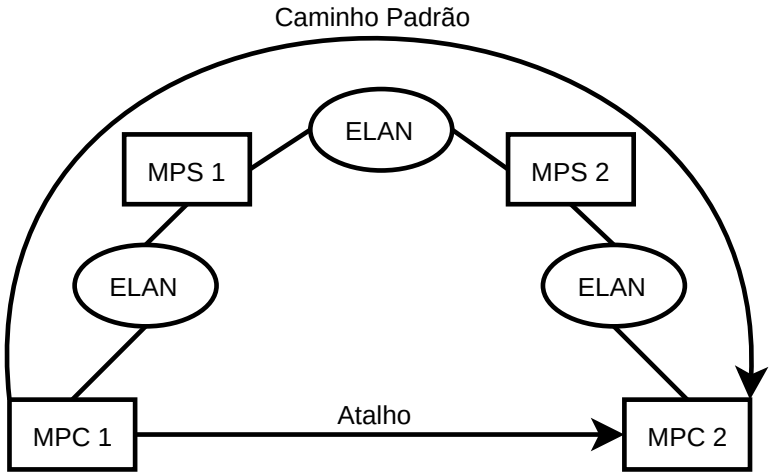

Fig. 2.8: MPOA: Atalho x Caminho Padrão.

ao NHRP o estabelecimento de um atalho. O provisionamento destes atalhos demandam ações de sinalização. Entretanto, existe uma dificuldade em classificar um conjunto de pacotes como sendo um fluxo, pois não é possível padronizar um método que faça este tipo de classificação devido a enorme variedade de aplicações. Por exemplo, podemos ter uma aplicação de vídeo gerando um tráfego constante de pacotes mas que tenha curta duração. Por outro lado, podemos ter uma transferência de arquivos (FTP) experimentando uma grande variação de atraso (*jitter*), o que torna o tráfego dos pacotes inconstante, mas que tenha uma longa duração.

2.1.6 MPLS

Produtos como IP *Switching* [Newman et al., 1998] e *Tag Switching* [Rekhter et al., 1997] surgiram na primeira metade dos anos noventa. Estas tecnologias utilizavam a velocidade oferecida pelo *hardware* ATM e substituíam os protocolos de sinalização do ATM por protocolos de "sinalização IP", mais integrados com o roteamento e o endereçamento IP. Obviamente, estas tecnologias, todas proprietárias, eram incapazes de interoperarem, apontando para a necessidade de um modelo padrão de comutação IP por rótulos e de protocolos de sinalização IP.

Em dezembro de 1996 foi formado um grupo de trabalho no IETF que tinha como foco a integração do roteamento e da comutação através da definição de um padrão que englobasse os principais aspectos das tecnologias baseadas no modelo Par (*Peer*) [Magalhães and Cardozo, 2003a]. Surgia assim o grupo de trabalho MPLS com o objetivo de integrar o roteamento no nível de camada de rede e a comutação por rótulos em uma única solução.

O MPLS é uma tecnologia que oferece as características de gerenciamento de tráfego, QoS, engenharia de tráfego e velocidade no processo de encaminhamento de pacotes encontradas nas redes

ATM. Porém, mantendo a flexibilidade e simplicidade das redes IP [Siqueira, 2002].

Quando pacotes da terceira camada do modelo OSI são transmitidos através de uma rede não orientada a conexão, cada roteador deve tomar decisões de encaminhamento a cada pacote que ele recebe. Isto é, cada roteador deve rodar um algoritmo de roteamento a fim de preencher uma tabela de rotas para que, com base na análise desta tabela e do cabeçalho dos pacotes, seja escolhido o próximo nó para o qual cada pacote deve ser encaminhado.

No MPLS, a análise do cabeçalho de nível três, visando mapear os pacotes para determinadas classes, denominadas FEC (*Forwarding Equivalence Classes*), é feito somente no ingresso da rede. Feito este mapeamento, cada pacote é rotulado com um código associado a sua respectiva FEC. O pacote rotulado é enviado ao próximo nó, que o encaminhará com base neste rótulo e em uma tabela que associa interface/rótulo de entrada a interface/rótulo de saída. Desta maneira, não é mais necessária a análise das informações contidas no cabeçalho de nível três dos pacotes. As vantagens deste paradigma são descritas a seguir:

- O encaminhamento MPLS pode ser feito por comutadores baseados em hardware (ASICs - *Application-Specific Integrated Circuit*) e não somente por roteadores baseados em software.
- O mapeamento de pacotes para FECs no roteador de ingresso pode ser feito com base em informações não contidas no cabeçalho de nível três como, por exemplo, a interface de entrada do pacote ou regras de engenharia de tráfego.
- As técnicas utilizadas no MPLS possibilitam o roteamento explícito sem que cada pacote precise carregar as informações da rota a ser percorrida (*source routing*).

O MPLS pode ser aplicado a qualquer protocolo de nível três, embora sua arquitetura definida na RFC 3031 [Rosen et al., 2001] seja focada no IP. Roteadores que implementam o MPLS são denominados LSRs (*Label Switching Routers*). Cada LSR de núcleo possui uma tabela denominada NHLFE (*Next Hop Label Forwarding Entry*) que guarda informações do tipo: (rótulo-interface de entrada, rótulo-interface de saída) para tomada de decisões de encaminhamento. Na Figura 2.9 é possível visualizar os elementos fundamentais da arquitetura MPLS.

O conceito de rótulos usado pelo MPLS se assemelha aos campos identificadores de circuito virtual do ATM (VPI e VCI) e possibilitam a criação de circuitos virtuais sobre uma rede de pacotes. No caso do MPLS, estes circuitos são denominados LSPs (*Label Switched Path*). A introdução dos LSPs muda as características atuais de melhor esforço das redes IP permitindo a elas incorporarem características como: engenharia de tráfego, tratamento diferenciado para fluxos com diferentes classes de serviços, QoS e VPNs.

Para serem utilizados de forma eficaz, os rótulos devem ser posicionados nos quadros da camada de enlace para permitir a comutação de pacotes sem que haja a necessidade de processamento na

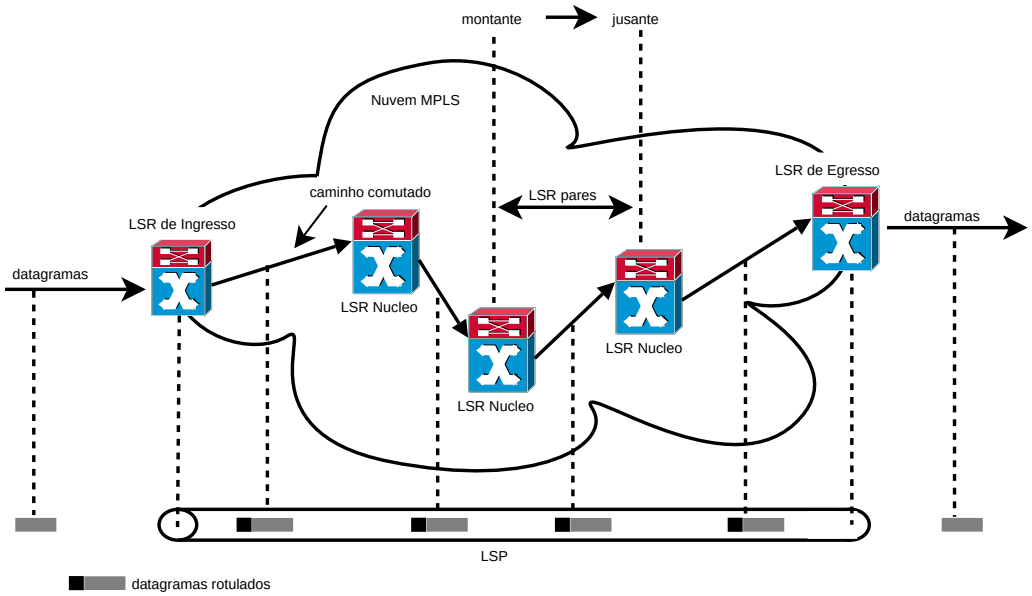

Fig. 2.9: Elementos fundamentais da arquitetura MPLS.

camada de rede. Idealmente, os rótulos devem coincidir com os identificadores de conexões dos enlaces. Entretanto, muitas tecnologias de enlace não empregam identificadores de conexão, como é o caso dos enlaces *Ethernet*. Para estas situações, o MPLS define uma estratégia de rotulação denominada encapsulamento genérico. Uma estrutura denominada *Shim Header* é posicionada entre o cabeçalho de enlace e o cabeçalho de rede. É possível verificar o posicionamento do *Shim Header* na Figura 2.10.

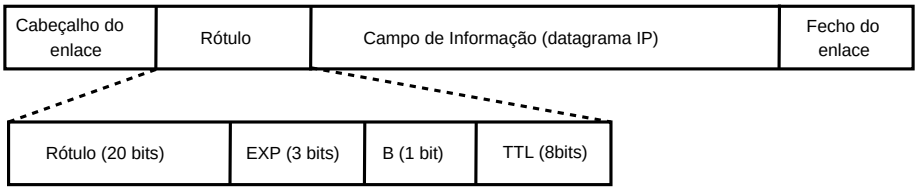

Fig. 2.10: *Shim Header* posicionado entre os cabeçalhos de enlace e rede.

A arquitetura MPLS permite que um pacote carregue vários rótulos organizados na forma de uma pilha. A questão fundamental envolvendo a pilha de rótulos é que a decisão de encaminhamento realizada nos LSRs é sempre baseada no rótulo que se encontra no topo da pilha. Um dos aspectos fundamentais da pilha de rótulos consiste na possibilidade de definir uma hierarquia de rótulos. Esta hierarquização de rótulos permite o tunelamento de informações através de múltiplos domínios, cabendo aos LSRs situados nas bordas dos domínios o controle da pilha de rótulos.

As redes IP tradicionais, conhecidas por seu serviço de melhor esforço e o roteamento do tipo *hop-by-hop* oferecido a seus pacotes de dados, contrapõem-se ao MPLS, o qual oferece o roteamento

explícito de pacotes e a capacidade de agregar tráfego como mecanismos de engenharia de tráfego em redes IP. Geralmente, o LSR de ingresso é responsável por especificar todos ou alguns dos LSRs por onde os pacotes irão passar. A agregação de tráfego é o processo que permite dar o mesmo tratamento a uma determinada classe de tráfego. Por exemplo, independentemente do endereço de destino, um LSR de origem pode utilizar um único LSP para todo o tráfego que possui o mesmo ponto de saída da nuvem MPLS.

2.1.7 GMPLS

O diferencial do GMPLS [Mannie, 2004] em relação ao MPLS é o suporte a múltiplos tipos de comutação que são oferecidos pelo GMPLS. Por exemplo, comutação por tempo (TDM), comprimento de onda (*lambda*) e por fibra. O suporte a estes tipos adicionais de comutação levaram o GMPLS a estender algumas funcionalidades presentes no MPLS e, em alguns casos, adicionar novas funcionalidades. Algumas destas funcionalidades dizem respeito à requisição e distribuição de rótulos (*labels*), propagação de erros e a propagação de informações de sincronização entre os comutadores de ingresso e de egresso.

O GMPLS não restringe o modelo de interconexão utilizado entre as redes. É possível utilizar o modelo Sobreposto (*Overlay*), Aumentado (*Augmented*) e Par (*Peer*). Detalhes do funcionamento destes modelos são apresentados na Seção 2.3. Outra característica do GMPLS é a separação entre o plano de controle e o plano de encaminhamento. Ele ainda divide o plano de controle em plano de sinalização e plano de roteamento. Os tipos de interfaces suportados pelo GMPLS são subdivididas nas seguintes classes:

1. Interfaces que comutam pacotes (PSC): encaminham dados baseadas em informações presentes nos cabeçalhos dos pacotes. Por exemplo, roteamento de pacotes baseados no cabeçalho IP ou comutação baseada no conteúdo do *shim header* MPLS.

2. Interfaces que comutam na camada 2 (L2SC): comutam células ou quadros baseadas no conteúdo de seus cabeçalhos. Por exemplo, interfaces de *switches Ethernet* que comutam com base no cabeçalho MAC e comutadores ATM que utilizam o VPI/VCI para comutar.

3. Interfaces que comutam por intervalos de tempo (TDM): são interfaces capazes de comutar dados baseadas em intervalos de tempos cíclicos. Um exemplo deste tipo de comutação é o SONET/SDH.

4. Interfaces que comutam por comprimento de onda (LSC): baseiam-se no comprimento de onda no qual os dados foram recebidos para tomar suas decisões de comutação. Como exemplos de

equipamentos que comutam comprimentos de onda temos o PXC (*Photonic Cross Connect*) e o OXC (*Optical Cross Connect*).

5. Interfaces que comutam por fibra (FSC): são interfaces que utilizam o posicionamento físico dos dados (no mundo real) para comutar. Exemplos desta interface são PXCs e OXCs que operaram no nível de uma ou múltiplas fibras.

Extensões aos protocolos e algoritmos de roteamento tradicionais são necessárias de forma a codificar e carregar uniformemente as informações dos enlaces. Informações explícitas de roteamento são necessárias à sinalização. Além das informações explícitas de roteamento, a sinalização deve ser capaz de transportar parâmetros como largura de banda, tipo de sinal, proteção desejada, que são requisitados para os caminhos. Os novos mecanismos de sinalização definidos pelo GMPLS otimizam o estabelecimento de caminhos e possibilitam o estabelecimento de caminhos bidirecionais, antes não suportados pelo MPLS. Maiores detalhes destes mecanismos podem ser encontrados em [Mannie, 2004].

O uso de tecnologias como DWDM (*Dense Wavelength Division Multiplexing*) possibilitam o uso de um elevado número de enlaces entre nós adjacentes (milhares de comprimentos de onda caso múltiplas fibras forem utilizadas), tornando o gerenciamento destes enlaces complexo. Para isso, um novo protocolo (LMP) foi definido para viabilizar este novo cenário. O LMP é apresentado na Seção 2.2.3.

A capacidade de operar com equipamentos ópticos que comutam comprimentos de onda somada a toda infraestrutura padronizada pelo IETF, fizeram do GMPLS a escolha natural para compor o núcleo da rede óptica do Projeto GIGA.

2.2 Protocolos do Plano de Controle GMPLS

A Seção 2.1 apresenta uma breve descrição de algumas soluções que foram desenvolvidas a fim de prover uma estrutura que suportasse a demanda gerada por novos serviços e aplicações que surgem a todo o momento. Os modelos gerados por estas soluções, além do ganho em capacidade de transmissão, apresentam mecanismos de automação para redes que são responsáveis por ações de sinalização, roteamento e gerenciamento de redes, entre outros, e visam sempre oferecer aos clientes (redes de acesso) das redes de transporte (*backbones*) uma forma mais transparente e dinâmica de utilização dos recursos existentes.

Como citado anteriormente, a solução escolhida para compor o núcleo óptico de transmissão da rede GIGA foi o GMPLS por oferecer um conjunto de funcionalidades capazes de automatizar uma rede óptica como a utilizada no Projeto GIGA. O IETF padronizou protocolos para desempenhar

funções de sinalização (RSVP-TE, CR-LDP), funções de roteamento (OSPF-TE, IS-IS) e gerência de enlaces (LMP). O IETF desenvolveu ainda uma especificação da UNI que será apresentada em detalhes na Seção 3.2.

O protocolo de sinalização escolhido para a rede GIGA é o RSVP-TE e o protocolo de roteamento é o OSPF-TE. Junto ao OSPF-TE é utilizado um algoritmo para atribuir rotas e comprimentos de onda (RWA) a fim de prover uma melhor utilização dos recursos da rede óptica do Projeto GIGA. As próximas seções descrevem estes protocolos.

2.2.1 RSVP

Em 1991, o Departamento de Ciências da Computação da Universidade da Califórnia em Los Angeles (UCLA) desenvolvia pesquisas na área de IP *multicasting* e teleconferência na rede DARTNET (*DARPA Textbed Network*) sobre a supervisão da professora Lixia Zhang. Neste contexto era necessário um protocolo capaz de efetuar a configuração de estados no cenário *multicast*. Ali nascia o RSVP, sem a mínima pretensão de que ele viesse a ser tão difundido [Durham and Yavatkar, 1999].

Talvez a semântica do RSVP seja responsável por sua sobrevivência e readaptação a diversos cenários. Ele trabalha com a troca de mensagens compostas por objetos de tipos e tamanhos variados (TLVs) e utiliza o conceito do *softstate*. As principais mensagens são as de *Path* e *Resv*, responsáveis pelo estabelecimento das reservas para um determinado fluxo de dados. O *softstate*, talvez sua mais importante característica, determina que de tempos em tempos seja feita a troca de mensagens de *refresh* para a manutenção dos estados de reserva dos fluxos de dados nos roteadores pertencentes à rota dos dados, ou seja, caso a rota para um determinado fluxo venha a ser alterada, as mensagens de sinalização passam a seguir pela nova rota. Na prática, isto significa a efetivação da reserva dos recursos pela nova rota e, também, a liberação dos recursos consumidos pelos roteadores que pertenciam a rota, devido ao *time-out* gerado pela ausência de mensagens de *refresh*.

Existem três grandes marcos de evolução na história do RSVP. Em setembro de 1997 o IETF publicou seis RFCs onde especificava o funcionamento dos mecanismos do protocolo. Neste primeiro instante o RSVP era utilizado no contexto dos Serviços Integrados (Seção 2.1.1) como protocolo responsável pelo estabelecimento de reservas. As RFCs publicadas nesta época foram: RFC 2205 [Braden et al., 1997], RFC 2206 [Baker et al., 1997], RFC 2207 [Berger and O'Malley, 1997], RFC 2208 [Mankin et al., 1997], RFC 2209 [Braden and Zhang, 1997] e RFC 2210 [Wroclawski, 1997]. O *overhead* que o RSVP poderia causar em redes de grande porte que implementassem o IntServ foi uma das principais causas da não difusão do IntServ.

Em dezembro de 2001 o IETF publicou duas RFCs onde o RSVP era estendido para suportar o estabelecimento de caminhos comutados por rótulos (LSPs) no contexto do MPLS (Seção 2.1.6).

Neste segundo instante ele recebe o nome de RSVP-TE, pois agora suporta funções de engenharia de tráfego. Dentre estas funções, é possível estabelecer rotas por caminhos alternativos (rotas explícitas na origem), ou seja, rotas não mais baseadas apenas em algoritmos de roteamento do tipo caminho mais curto primeiro. Além disso, o RSVP-TE recebeu novos objetos para efetuar a distribuição de rótulos do MPLS. As RFCs que definem estas extensões são: RFC 3209 [Awduche et al., 2001a] e RFC 3210 [Awduche et al., 2001b]. A efetiva implementação do MPLS, leia-se conceito de comutação por rótulos, foi adiada devido ao aparecimento dos *GigaRouters*. Roteadores estes capazes de rotear uma quantidade altíssima de pacotes. Em outras palavras, não havia mais a necessidade nem a justificativa para tamanho investimento na implantação do MPLS.

No primeiro trimestre de 2003 foram publicadas cinco RFCs pelo IETF estendendo o RSVP novamente. Agora o RSVP-TE recebe alguns novos objetos e uma nova mensagem para adequar-se às necessidades do GMPLS (Seção 2.1.7). Além dos objetos e da mensagem, o protocolo sofreu algumas mudanças na sua semântica, pois no GMPLS temos uma total separação entre o plano de controle e o plano de encaminhamento. Até então, as mensagens de sinalização percorriam o mesmo caminho dos dados. A maioria dos novos objetos e a nova mensagem visam otimizar o funcionamento do protocolo. Entre a lista de novas funcionalidades estão: a possibilidade de notificar falhas diretamente a um determinado elemento de rede; a sugestão de rótulos para otimizar o estabelecimento de LSPs e a possibilidade de estabelecer LSPs bidirecionais. As RFCs onde o comportamento de RSVP-TE para GMPLS é descrito são: RFC 3471 [Berger, 2003a], RFC 3473 [Berger, 2003b], RFC 3474 [Lin and Pendarakis, 2003], RFC 3476 [Rajagopalan, 2003] e RFC 3477 [Kompella and Rekhter, 2003].

Estes são apenas os principais cenários onde o RSVP pode ser utilizado. Existem outras RFCs definidas pelo IETF onde outros objetos, e até mesmo outras mensagens, são especificados para que o RSVP efetue as mais variadas funcionalidades, caracterizando sua portabilidade.

2.2.2 OSPF

O desenvolvimento do OSPF teve início em 1987 e em 1991 sua primeira especificação foi publicada, RFC-1247 [Moy, 1991]. As motivações que levaram ao surgimento do OSPF foram as falhas de seu predecessor, o protocolo RIP. Naquele tempo, quando o tamanho dos ASs começaram a aumentar em um ritmo acelerado, o tempo de convergência do RIP e a banda consumida no processo começaram a ser inaceitáveis. O RIP é um protocolo de roteamento baseado em vetores de distância, onde a métrica usada para calcular as rotas é o custo até as outras sub-redes. Por outro lado, o OSPF é um protocolo baseado no estado dos enlaces, o qual utiliza uma métrica mais flexível. Um custo é atribuído a cada enlace, normalmente relacionado à capacidade (largura de banda) do enlace. Além de resolver os problemas do RIP, o OSPF introduziu novas funcionalidades, como hierarquia

de roteamento, separação de rotas internas e externas e melhorias na segurança.

Cada roteador OSPF divulga o estado de seus enlaces na forma de LSAs. LSAs são distribuídos pela rede através de um mecanismo complexo chamado *"reliable flooding"*. Este mecanismo garante que todos os roteadores de uma rede (ou área OSPF) terão o mesmo conjunto de LSAs. Este conjunto é chamado de LSDB. LSAs são transportados entre roteadores através de pacotes OSPF. O endereço IP de destino é sempre o endereço IP dos vizinhos ou um endereço *multicast*. Existem cinco tipos de pacotes OSPF: o primeiro (Tipo 1) é o pacote *Hello*, utilizado para descobrir e manter a relação de vizinhança entre roteadores; o segundo (Tipo 2) é o pacote *Database Description*; o terceiro (Tipo 3) é o pacote *Link State Request*; o quarto (Tipo 4) é o pacote *Link State Update* e o quinto (Tipo 5) é o pacote *Link State Acknowledgment*. Os pacotes dos tipos 2 à 5 são utilizados para manter a sincronização da LSDB.

A tabela de roteamento de cada roteador é calculada aplicando-se um algoritmo do tipo caminho mais curto primeiro (SPF) em uma árvore de alcançabilidade construída a partir da LSDB. Este processo resulta em caminhos para cada uma das sub-redes conhecidas, e a identidade do próximo nó para cada destino é inserida na tabela de roteamento. A partir do momento que rotas para sub-redes são calculadas, todo o tráfego relacionado a uma determinada sub-rede irá utilizar a rota estabelecida àquela sub-rede, sem se preocupar se os enlaces que compõem a rota têm ou não recursos suficientes para atender o tráfego em questão, mesmo que existam outros enlaces com recursos disponíveis que também alcancem o mesmo destino. Isto ocorre porque é impossível efetuar engenharia de tráfego em uma rede usando apenas o custo dos enlaces como parâmetro, outros parâmetros precisam ser levados em consideração.

No final dos anos noventa o IETF, através do grupo de trabalho IETF MPLS, propôs melhorias ao OSPF que permitiam ao protocolo não apenas carregar o custo dos enlaces, mas também outras propriedades de engenharia de tráfego, como largura de banda, endereços IP locais e remotos e classes de grupos administrativos. O protocolo OSPF com estas extensões é chamado OSPF-TE, RFC-3630 [Katz et al., 2003].

Com o advento do GMPLS, novas extensões foram propostas ao OSPF para suportar enlaces ópticos e enlaces que fazem multiplexação por tempo (TDM), RFC-4203 [Kompella and Rekhter, 2005]. Recebeu quatro novas sub-TLVs para a sua TLV de enlace. São elas:

- TLV com os identificadores locais e remotos dos enlaces;
- TLV com o tipo de proteção oferecido pelos enlaces;
- TLV descrevendo a capacidade de comutação das interfaces;
- TLV descrevendo o grupo de enlaces que compartilham o mesmo grupo de risco (SRLG).

2.2.3 LMP

No contexto de redes ópticas e do GMPLS, é possível que entre um par de nós exista milhares de interconexões, onde cada uma destas interconexões pode ser composta de vários enlaces de dados. Por motivos de escalabilidade é interessante que estes múltiplos enlaces de dados possam ser combinados em um único enlace TE. Além disso, um canal de controle entre os nós do plano de controle é necessário para prover ações de roteamento, sinalização e gerência de enlaces. É possível que este canal de controle utilize um meio físico diferente do utilizado pelo canal de dados.

Todas estas características dificultam o gerenciamento dos enlaces que compõem a rede de transporte. Para solucionar estes problemas, o IETF especificou o protocolo de gerência de enlaces LMP (*Link Management Protocol*) [Lang, 2005]. O LMP oferece, basicamente, quatro funcionalidades que são desempenhadas através de vinte mensagens definidas por ele. Dentre suas funcionalidades, duas são opcionais e duas são obrigatórias. Sendo elas:

- verificação de enlace (opcional);
- localização de falha (opcional);
- gerenciamento do canal de controle (obrigatória);
- correlação de propriedades dos enlaces (obrigatória).

As duas funcionalidades opcionais do LMP, verificação de enlace e localização de falha, são úteis em ambientes onde o plano de controle é fisicamente disjunto do plano de encaminhamento, como é o caso das redes ópticas que implementam o GMPLS. A verificação de enlaces é utilizada na descoberta de vizinhança do plano de encaminhamento, ou seja, para verificar a conectividade física dos enlaces. Esta tarefa é feita através do envio de mensagens de *Test* sob os enlaces de dados e o retorno de mensagens de *TestStatus* sob o canal de controle. Para isto, é preciso que os equipamentos que constituem a rede óptica suportem a conversão de sinais ópticos em sinais eletrônicos, durante a fase de descoberta de vizinhança, para poder recuperar a mensagem de *Test* recebida em todas as suas interfaces. A localização de falhas é obtida através da troca de mensagens de *ChannelStatus* entre elementos de rede que são adjacentes. Através desta troca de mensagens é possível suprimir a emissão de alarmes pelos nós *downstream* e localizar onde a falha ocorreu para que ações de proteção/restauração sejam tomadas [Lang, 2005].

As duas principais funcionalidades do LMP são: gerenciamento do canal de controle e correlação das propriedades dos enlaces. A primeira é utilizada para estabelecer e manter canais de controle entre elementos de rede adjacentes através da troca de mensagens de *Config* e um mecanismo de *keep-alive* entre os nós. A segunda tem como objetivo realizar a agregação de múltiplos enlaces de

dados em um enlace TE e sincronizar as propriedades deste enlace TE através da troca de mensagens de *LinkSummary* [Lang, 2005].

No final de 2005 e início de 2006, o IETF publicou cinco RFCs definindo o LMP e sua implementação em redes do tipo DWDM ou SONET/SDH. As RFCs são: RFC 4204 [Lang, 2005], RFC 4207 [Lang and Papadimitriou, 2005], RFC 4209 [Fredette and Lang, 2005], RFC 4327 [Dubuc et al., 2006] e RFC 4394 [Fedyk et al., 2006].

A única mensagem do LMP que é transmitida pelo canal de dados é a mensagem de *Test*. Como mencionado anteriormente, sua função (opcional) é verificar a conectividade física dos enlaces entre elementos de rede adjacentes. Os comutadores ópticos desenvolvidos para o Projeto GIGA não desempenham esta função. Desta forma, os procedimentos de verificação de conectividade dos enlaces e o preenchimento da tabela de correlação são feitos manualmente no Projeto GIGA.

2.2.4 RWA

A grande capacidade de transmissão oferecida pelas redes ópticas servirá a diferentes aplicações associadas a uma enorme variedade de redes clientes. Uma conexão de rede óptica consiste em uma comunicação entre um nó de origem e um nó de destino que atravessa vários enlaces formando, assim, um caminho de luz (*lightpath*). A maioria dos nós que constituem o plano de encaminhamento das redes ópticas não são capazes de fazer a conversão de comprimentos de onda. Devido a esta restrição, um caminho de luz precisa ter o mesmo comprimento de onda em todos os enlaces que constituem a rota.

Um caminho de luz será definido por um protocolo de atribuição de rotas e comprimentos de onda (RWA - *Routing and Wavelength Assignment*) [Chlamtac et al., 1992, Ramaswami and Sivarajan, 1995, Chlamtac et al., 1996]. As redes clientes irão requisitar conexões entre dois nós da rede óptica de transporte e uma rota e um comprimento de onda serão atribuídos ao caminho de luz, respeitando as restrições que são impostas, como, continuidade do comprimento de onda, limitações de potência na transmissão ou algum outro impedimento físico.

O algoritmo de RWA deve conhecer a topologia física da rede óptica. O objetivo é acomodar todas as requisições minimizando o uso de comprimentos de onda (recursos escassos). O resultado gerado pelo algoritmo pode variar com diferentes disposições dos dispositivos de rede.

O tráfego das redes pode ser definido como estático ou dinâmico. Para o primeiro caso, todas as requisições de conexões são previamente conhecidas. A solução ótima é calculada *offline*, antes que algum caminho de luz seja estabelecido. Entretanto, a solução ótima é praticamente possível somente para redes com tamanho limitado. No caso dinâmico, as requisições devem ser atendidas sem efetuar qualquer mudança na configuração dos caminhos de luz já estabelecidos. A rota e o comprimento de onda são escolhidos levando em consideração o estado momentâneo da rede óptica. Os objetivos são

similares aos do caso estático: aceitar o maior número de requisições futuras e minimizar o uso dos comprimentos de onda. Mas, neste segundo caso, o RWA também precisa ter informações sobre os caminhos de luz que já estão estabelecidos. Uma solução para o problema de disseminação do estado atual da rede óptica é utilizar o protocolo OSPF (Seção 2.2.2).

Dependendo do tamanho da rede óptica, o RWA dinâmico pode levar muito tempo para gerar uma solução. Para reduzir esta complexidade computacional, o problema RWA pode ser dividido em dois sub-problemas: a atribuição de uma rota e a atribuição de um comprimento de onda. As três estratégias mais comuns de roteamento do RWA são:

1. Roteamento Fixo - apenas uma rota, geralmente a mais curta, pode ser utilizada para conectar os nós de origem e destino. Se não houver nenhum comprimento de onda disponível naquela rota, a requisição é bloqueada. A rota para todos os pares origem-destino são pré-calculadas.

2. Roteamento Fixo-Alternado - uma lista de caminhos é atribuída a um par de nós origem-destino. Se a primeira rota da lista não acomodar a requisição, a próxima rota deve ser testada, até o final da lista. Caso todas as rotas não acomodem a requisição, ela será bloqueada.

3. Roteamento Adaptativo - a rota será selecionada levando em consideração a configuração da rede. Normalmente, o menor caminho disponível naquele instante será escolhido.

2.3 Modelos de Interconexão do Plano de Controle

É indispensável que haja uma interação eficiente entre as redes que compõem um determinado ambiente. Por exemplo, é necessária a interação das redes clientes (MPLS) com a rede óptica (GMPLS) do Projeto GIGA. Atualmente, existem três modelos de interação definidos pelo IETF, cada um deles com diferentes níveis de cooperação. Temos o Modelo Overlay e o Modelo Peer como extremos e o Modelo Aumentado como um modelo intermediário. Nesta seção, estes modelos de interconexão do plano de controle são apresentados.

2.3.1 Modelo Sobreposto (*Overlay*)

Este modelo foi desenvolvido para um cenário comercial no qual operadoras com uma grande capacidade de transmissão instalada fornecem seus recursos e facilidades de rede à redes de acesso (clientes), visando atender o forte crescimento no tráfego de dados (Internet) e a proliferação de redes privadas virtuais (VPNs). Neste modelo é assumido uma grande independência (autonomia) entre as partes envolvidas apresentando uma total separação entre o plano de controle da rede cliente e o plano de controle da rede de transporte (núcleo). Este modelo limita a troca de informações

de sinalização, ou seja, os protocolos de sinalização e roteamento da rede cliente são totalmente independentes dos protocolos da rede núcleo (*backbone*). A interação da sinalização entre os dois planos de controle distintos ocorre através da UNI, definindo uma relação cliente-servidor entre as redes. Como conseqüência este é o modelo mais opaco, possui poucas características e menos flexibilidade que os modelos *Peer* e Aumentado.

Um outro problema deste modelo é o *"full mesh"*. Nos protocolos de roteamento baseados no estado de enlace como o OSPF, a tabela de roteamento é derivada em cada roteador a partir de uma árvore de caminhos mais curtos. Esta árvore é calculada aplicando-se algoritmos de "caminho mais curto primeiro" (usualmente Dijkstra) em uma base de estados de enlaces (LS-DB). Esta base é formada através da troca de informações de estado de enlace (LSA) entre todos os roteadores da rede onde uma relação de vizinhança foi previamente estabelecida. Este mecanismo é denominado "disseminação confiável". No modelo *Overlay*, para que exista esta relação de vizinhança entre os roteadores da rede cliente situados na borda da rede de transporte (n roteadores), são necessárias a cada roteador n-1 conexões através da rede de transporte, ou seja, o número de conexões na rede de transporte cresce proporcionalmente ao quadrado do número de nós da rede cliente, criando um problema de escalabilidade. Este problema ocorre devido à separação, obrigatória no modelo *Overlay*, dos protocolos de roteamento da rede cliente e da rede de transporte. A Figura 2.11 ilustra um cenário que implementa o modelo *Overlay*. Note a existência de diferentes planos de controle, indicando a sobreposição de funções como roteamento e sinalização.

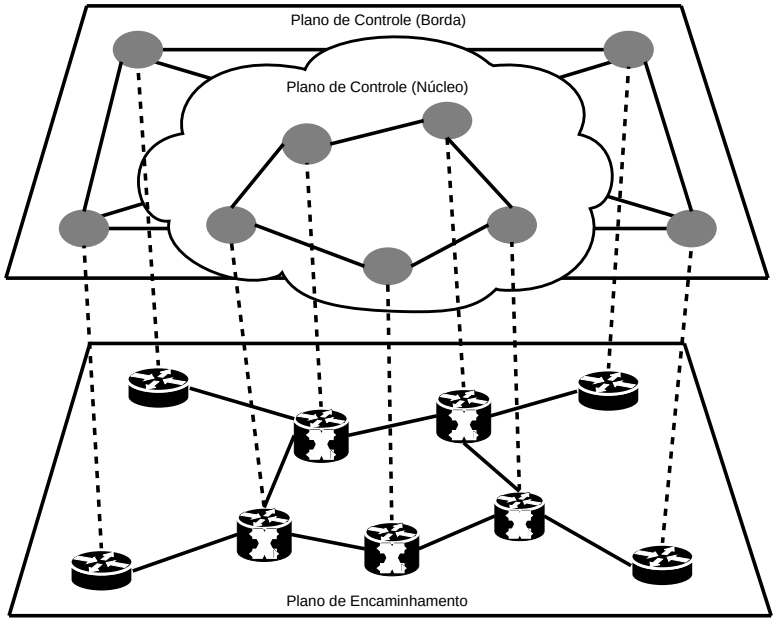

Fig. 2.11: Planos de controle no modelo Sobreposto (*Overlay*).

Apesar de todos os problemas apresentados acima, este modelo é muito utilizado, principalmente

pela opacidade oferecida, o que permite às operadoras "esconder" de seus clientes qual a topologia física de seu *backbone*. Diversas soluções foram propostas para os problemas relatados anteriormente, por exemplo, a criação de áreas de roteamento que resolvem, ou pelo menos minimizam, o problema de escalabilidade. Em uma rede temos um Sistema Autônomo de Roteamento (AS), e este AS possui uma ou mais Áreas de Roteamento. Dentro de cada uma destas áreas a troca das informações de estado de enlace é feita como descrito anteriormente nesta seção. A diferença é que cada uma destas áreas elege um roteador para fazer a troca de informações entre as áreas.

2.3.2 Modelo Par (*Peer*)

Pode ser visto como dual ao modelo *Overlay*, para o qual é assumida uma relação de total confiança entre os domínios. Um exemplo é uma operadora que oferece ambos os serviços, o serviço de transporte e o serviço de acesso à Internet, e deseja otimizar o alinhamento topológico de sua rede de transporte com as operações de serviço de sua rede de acesso.

No modelo *Peer*, o plano de controle da rede de acesso é par (*Peer*) do plano de controle da rede de transporte, ou seja, uma única instância do plano de controle atua sobre as redes de acesso e de transporte, desta forma, não é necessária a utilização da UNI. Em outras palavras, as redes que fazem parte deste plano único de controle trocam informações completas de roteamento. Além disso, uma única instância dos protocolos que constituem o plano de controle é necessária, ou seja, não há sobreposição de funções. A Figura 2.12 ilustra um ambiente que implementa o modelo *Peer*.

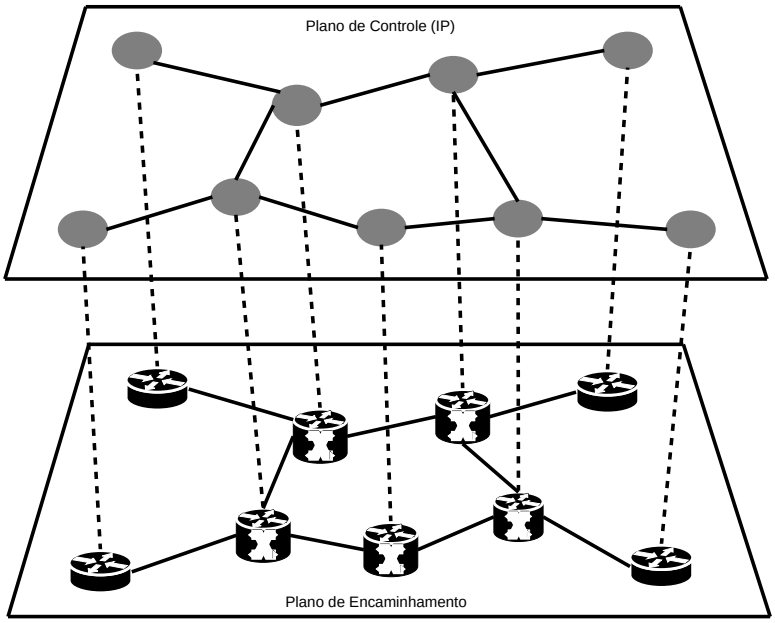

Fig. 2.12: Unificação do plano de controle no modelo Par (*Peer*).

É importante destacar que por considerar todos os elementos de rede como adjacentes, a escalabilidade pode ser afetada em uma rede que implemente o modelo *Peer* caso exista um número elevado de elementos de rede e exista somente um sistema autônomo de roteamento. Outro fator negativo deste modelo é que dificilmente seria implementado entre operadoras diferentes devido ao fato de oferecer visibilidade total às redes que compartilham o mesmo plano de controle. Geralmente, a maioria das operadoras não admite a possibilidade de mostrar sua topologia real a outras operadoras.

2.3.3 Modelo Aumentado (*Augmented*)

Em termos de níveis de confiabilidade, o modelo Aumentado pode ser considerado como um modelo intermediário entre os modelos *Overlay* e *Peer*. Este modelo utiliza a UNI e possui algumas características a mais que o modelo *Overlay*. Permite a troca de uma quantidade limitada de informações de roteamento entre a rede cliente e a rede de transporte. Esta troca limitada de informações de roteamento é justamente a característica que também o difere do modelo *Peer*, no qual informações completas de roteamento são trocadas entre as redes cliente e de transporte.

O modelo Aumentado basicamente permite a troca de informações de alcançabilidade entre elementos da rede cliente situados na borda da rede de transporte. Outra diferença entre os modelos é o esquema de endereçamento empregado. No modelo *Peer* os elementos da rede de transporte são elementos de rede IP endereçáveis. Em outras palavras, todos os elementos de rede compartilham o mesmo espaço de endereçamento. Já o modelo Aumentado permite, mas não obriga, a utilização do mesmo espaço de endereçamento, permitindo uma opacidade total ou parcial da topologia da rede de transporte.

Quando aplicado sem o esquema de endereçamento comum entre a rede cliente e a rede de transporte, o modelo Aumentado resolve apenas o problema de "Resolução de Endereço" (através da troca de informações de alcançabilidade) e mantém uma certa opacidade entre as redes envolvidas. Quando aplicado com o esquema de endereçamento comum entre as redes, é necessária uma estreita integração entre as instâncias de roteamento, de tal forma que se divulgue apenas as informações topológicas desejadas, mantendo assim o nível de opacidade desejado.

2.4 Resumo

Este capítulo fez uma breve apresentação sobre algumas das arquiteturas (IntServ, DiffServ, IP/ATM, LANE, MPOA, MPLS e GMPLS) desenvolvidas com o intuito de oferecer mecanismos capazes de prover QoS e Engenharia de Tráfego em redes que utilizam principalmente o modelo *Overlay*, utilizado pelo Projeto GIGA, e apresentou ainda os outros modelos existentes (*Peer* e

Aumentado).

A segunda seção deste capítulo apresentou os protocolos da família GMPLS que estão diretamente relacionados ao plano de controle da rede do Projeto GIGA (RSVP, OSPF, LMP e RWA) sem, no entanto, apresentar a UNI que será discutida em profundidade no próximo capítulo.

Capítulo 3

Comparação entre UNIs

Este capítulo apresenta as duas especificações existentes da UNI desenvolvidas pelo OIF e IETF, ou seja, é o resultado dos estudos que foram feitos sobre as especificações da UNI. Este capítulo tem o objetivo de introduzir os conceitos e funcionalidades relacionadas à operação da UNI.

Ambas as especificações possuem características em comum. Tendo isto em vista, neste capítulo, a seção que trata a UNI do OIF apresenta todos os conceitos e mecanismos de uma UNI, tais como os procedimentos opcionais e obrigatórios, mensagens de sinalização e regras que definem por exemplo o espaço de endereçamento utilizado na UNI. Após a apresentação da UNI do OIF, são apresentadas as extensões efetuadas pelo IETF à UNI do OIF sem, no entanto, repetir as características compartilhadas pelas duas especificações.

Para finalizar o capítulo, há uma seção onde o estudo das especificações é apresentado através de uma breve comparação entre as duas especificações da UNI.

3.1 UNI OIF (UNI Pública)

A UNI Pública, especificada pelo OIF [UNI Common Part, 2004], foi desenvolvida para trabalhar em redes que implementam o modelo *Overlay* (Seção 2.3.1). Possui cinco funcionalidades principais, cada uma delas possui um lado cliente e um lado servidor que são desempenhados pela UNI-C e pela UNI-N, respectivamente. As funcionalidades são:

1. Estabelecimento de conexões (sinalização)
2. Remoção de conexões (sinalização)
3. Verificação do estado de conexões (sinalização)
4. Descoberta automática de serviço e vizinhança (sinalização)

5. Uso da conexão (tráfego)

O serviço mais básico oferecido pela UNI Pública é a capacidade de criar e remover conexões sob-demanda na rede de transporte. No contexto UNI Pública, uma conexão pode ser definida como um caminho através da rede de transporte que possui parâmetros de tráfego (ex.: largura de banda, atraso) fixos e pode ser unidirecional ou bi-direcional. Através da UNI Pública é possível ainda obter o estado atual das conexões estabelecidas na rede de transporte. Entretanto, ela não possui mecanismos que permitam a modificação das conexões já estabelecidas. Para isso, é necessário remover e estabelecer uma nova conexão com as características desejadas. Uma vez estabelecida a conexão através da rede de transporte, o cliente pode utilizá-la para transmitir seus pacotes de dados. Para o cliente, esta conexão é vista como um enlace qualquer entre dois nós adjacentes.

A descoberta automática de serviço e de vizinhança são funcionalidades opcionais. A primeira função permite à rede cliente descobrir quais serviços são oferecidos pela rede de transporte e, ainda, à rede de transporte descobrir informações (ex.: protocolo de sinalização, tipo de quadro) da rede cliente. A segunda função é fundamental no mapeamento das interfaces que conectam os elementos da rede cliente aos elementos da rede de transporte. Possibilita ainda o estabelecimento do canal de controle entre UNI-Cs e UNI-Ns. A Tabela 3.1 apresenta os procedimentos da UNI Pública.

Tab. 3.1: Procedimentos obrigatórios e opcionais da UNI Pública (UNI OIF).

Procedimento	**Estado**	**Opções**
Sinalização	Obrigatória	RSVP-TE ou LDP
Canal de Controle	Obrigatório	*In-fiber* ou *Out-of-fiber* (rede IP) ou *Out-of-fiber* (rede dedicada)
Manutenção do Canal de Controle	Obrigatória	LMP *Hello* ou RSVP-TE *Hello*
Descoberta de Vizinhança	Opcional	Processo LMP ou Configuração Manual
Descoberta de Serviço	Opcional	Processo LMP ou Configuração Manual

3.1.1 Modelos de Invocação de Serviço

Existem dois modelos de invocação de serviço definidos pela UNI Pública. Um chamado de modelo de invocação direta e outro chamado de modelo de invocação indireta. Cada um deles possui duas variações conforme ilustra a Figura 3.1. É possível usar mais do que um modelo de invocação em uma mesma rede de transporte.

- Modelo de Invocação Direta: Neste modelo, a UNI-C é implementada pelo cliente da rede de transporte. Existem duas configurações para este modelo, itens 1a e 1b na Figura 3.1. A diferença entre eles é que no item 1a a UNI-N é implementada diretamente no elemento da

rede de transporte e no item 1b a UNI-N é um *proxy*. No segundo caso, não há nenhuma limitação quanto à localização da UNI-N. Uma interface de sinalização interna (ISI) é utilizada para carregar as mensagens de sinalização entre a UNI-N e o elemento de rede (TNE). O plano de controle IP (IPCC) entre a UNI-C e a UNI-N pode ser *in-fiber* ou *out-of-fiber*. Maiores detalhes são apresentados na Seção 3.1.2.

A ação que induz o cliente a solicitar serviços da rede de transporte pode partir de um sistema de gerência presente na rede cliente ou de decisões de engenharia de tráfego tomadas pelo próprio cliente. No interior da rede de transporte, os serviços requisitados através da UNI podem ser providos através de um serviço centralizado de provisionamento de conexões ou através de protocolos de sinalização distribuídos.

- Modelo de Invocação Indireta: Neste modelo, os clientes invocam serviços da rede de transporte através de um *proxy* UNI-C. Existe uma ISI entre os clientes e a UNI-C. As duas variações relativas à UNI-N são apresentadas nos itens 2a e 2b da Figura 3.1. Neste modelo:

  1. O cliente não é capaz de implementar o mecanismo de descoberta de vizinhança automática. As informações de vizinhança necessárias à UNI-N e UNI-C devem ser configuradas manualmente;
  2. A UNI-C efetua a sinalização em nome de clientes que são representados por ela;
  3. A comunicação entre a UNI-C e o cliente (ISI) pode ocorrer através de um protocolo proprietário;
  4. O plano de controle entre a UNI-C e a UNI-N é *out-of-fiber*;
  5. É permitido à UNI-C representar vários clientes;
  6. A UNI-N deve possuir um registro dos clientes que contenha informações dos clientes que podem ser representados por uma dada UNI-C ligada a ela;

3.1.2 Modelos de Implementação do Plano de Controle

Conforme apresentado na Figura 3.1, é necessário que haja um canal de controle IP (IPCC) entre a UNI-C e a UNI-N para transportar as mensagens de sinalização. A UNI Pública define as seguintes configurações para a realização de um IPCC:

- *In-fiber*: As mensagens de sinalização são transmitidas sobre um canal de comunicação específico (*lambda* de controle) que está embutido no enlace óptico entre o cliente e a rede de transporte. Aplicável apenas ao modelo 1a apresentado na Figura 3.1.

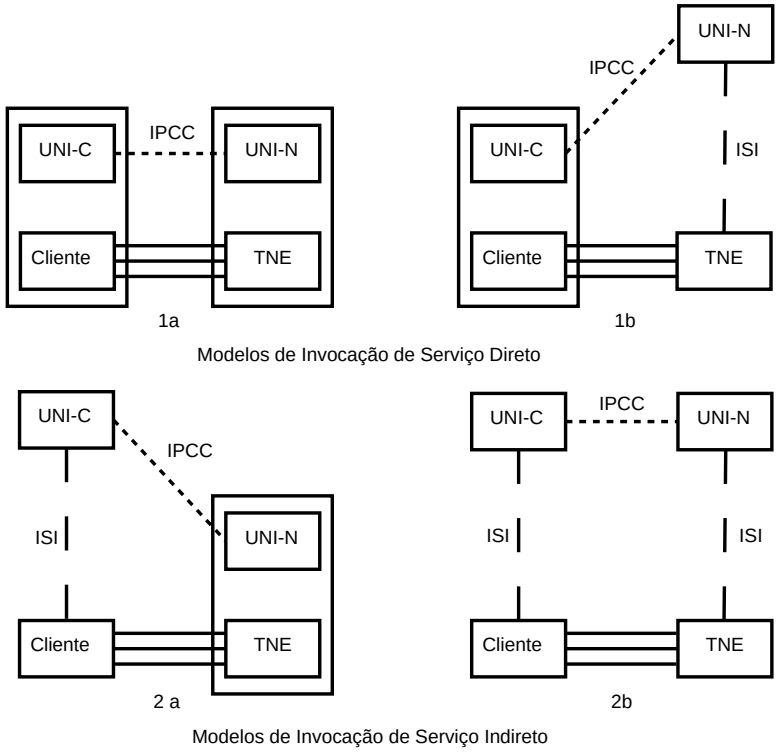

Fig. 3.1: Configurações para Invocação de Serviço na UNI Pública.

- *Out-of-fiber*: As mensagens de sinalização são transmitidas sobre um enlace à parte, dedicado à comunicação entre UNI-C e UNI-N. Por exemplo, é possível utilizar uma rede *Ethernet* na realização do IPCC. Aplicável aos modelos 1b, 2a e 2b apresentados na Figura 3.1.

Entre a UNI-C e a UNI-N pode haver diversos canais de controle estabelecidos. Por exemplo, oferecer redundância para o caso de falhas. O conjunto de todos os IPCCs é denominado Canal de Sinalização. A falha do canal de sinalização pode ser o resultado da queda de todos os IPCCs. Esta queda pode ser caracterizada como uma falha do enlace físico ou mesmo dos módulos UNI-C e UNI-N. A Figura 3.2 exemplifica configurações do IPCC e apresenta o Canal de Controle.

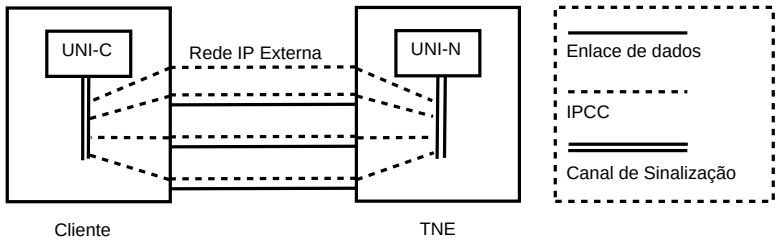

Fig. 3.2: IPCCs entre UNI-C e UNI-N e Canal de Sinalização.

3.1.3 Estrutura de Endereçamento

Espaços de endereçamento relevantes à UNI Pública:

1. Endereços internos da rede de transporte: são os endereços dos nós localizados no interior da rede de transporte. Utilizados nas ações de roteamento, provisionamento de conexões e ações de gerência de rede efetuadas na rede de transporte. São os endereços TNE A e TNE B da Figura 3.3. Estes endereços não são visíveis a partir das redes clientes e não são padronizados pela UNI Pública (OIF).

2. Identificador de nó da UNI-C e UNI-N (*Node ID*): é o endereço IP que identifica a terminação da conexão do canal de controle em uma UNI-C ou uma UNI-N. O *Node ID* permanece inalterado mesmo ocorrendo alguma mudança no estado ou no endereço dos IPCCs que o constituem. A UNI Pública define a utilização de um endereço IPv4 único dentro do domínio de comunicação da UNI-C e UNI-N. Embora não necessite ser um endereço IP válido globalmente, este requisito seria facilmente satisfeito caso o fosse. Este endereço é atribuído pela operadora responsável pelos nós. São representados na figura pelos endereços UNI-C A, UNI-C B, UNI-C C, UNI-N A e UNI-N B.

3. Identificador (IP) de canal de controle (CCID): as pontas dos canais de controle (IPCCs) devem ser identificados para possibilitar ações como a descoberta de vizinhança, descoberta de serviço e manutenção do canal de controle. Um IPCC pode ser numerado, neste caso é atribuído um endereço IPv4 para cada uma de suas pontas. Outra possibilidade é ser não-numerado, neste caso é identificado pela combinação do identificador de nó (*Node ID*) e o índice atribuído à interface. O CCID possui 32 bits, é único a cada nó e independente dos endereços das pontas do IPCC. Existe a possibilidade de haver apenas um IPCC entre uma UNI-C e uma UNI-N, ou seja, apenas uma interface de controle, neste caso os identificadores de canal de controle são os mesmos que identificam os nós da UNI-C e da UNI-N (item 2). Entretanto, é possível que exista diversos canais de controle entre uma UNI-C e uma UNI-N (várias interfaces), neste caso o identificador de canal de controle é o endereço de 32 bits descrito neste item e é diferente do identificador de nó.

4. Endereço atribuído pela rede de transporte (TNA): as pontas das conexões estabelecidas através da UNI são identificadas por um endereço que é atribuído pela rede de transporte (TNA). Cada TNA é um endereço IP, válido globalmente, atribuído a um ou mais enlaces de dados que conectam a UNI-N à UNI-C. Fica a cargo da rede de transporte a distribuição destes endereços com base em suas políticas de gerenciamento.

5. Endereços da rede cliente: estes endereços são definidos pelas redes clientes e não estão envolvidos no processo de sinalização da UNI. Isto ocorre devido ao modelo *Overlay*, ou seja, existe uma total separação entre os endereços das redes cliente e de transporte. Portanto, os formatos destes endereços não são especificados pela UNI Pública (OIF). São denotados como C_A, C_B e C_C na Figura 3.3.

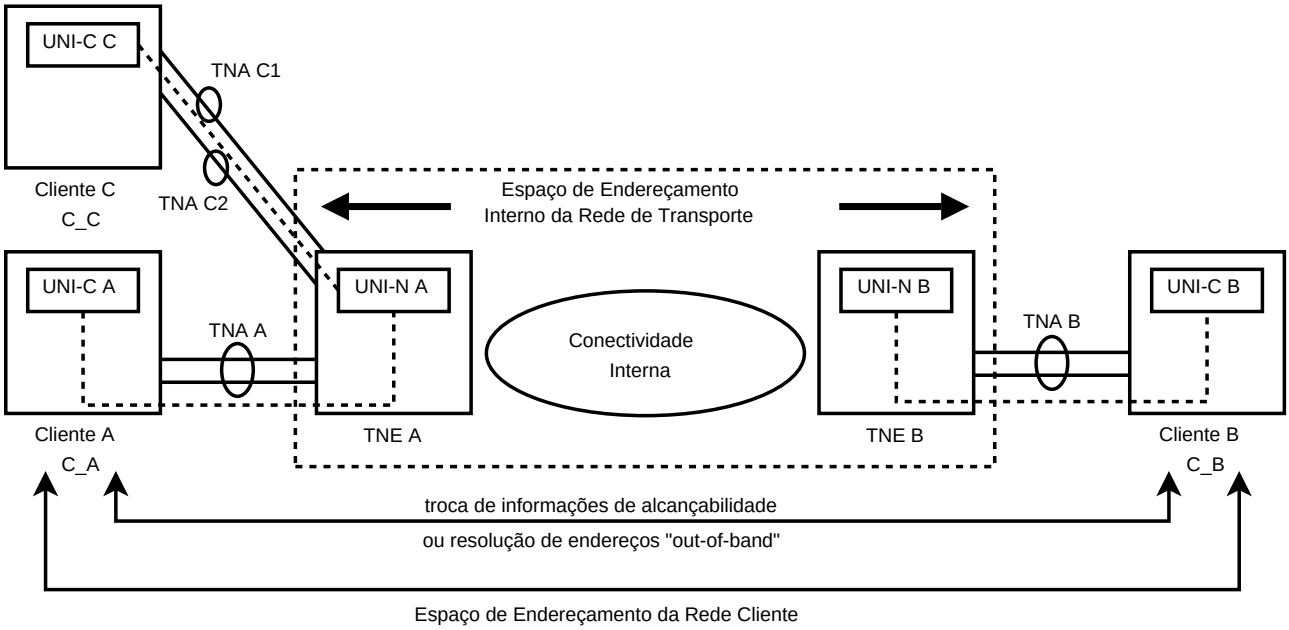

Fig. 3.3: Espaços de endereçamento relevantes à UNI Pública.

Conforme descrito anteriormente, um endereço TNA pode identificar um ou mais enlaces de dados entre a UNI-N e a UNI-C. Em um grupo de enlaces compartilhando o mesmo TNA, um enlace de dados individual é identificado por um Identificador Lógico de Porta em cada uma de suas pontas. A decisão de qual identificador atribuir a estes enlaces é feita localmente a cada nó. Conseqüentemente, cada identificador lógico corresponde a uma porta física. Portanto, é possível utilizar formatos de identificadores diferentes para identificar portas lógicas e portas físicas. Desta forma, um operador de rede poderia expor a seus clientes os identificadores lógicos de suas interfaces sem, no entanto, expor seu espaço de endereçamento interno. A UNI Pública sugere a utilização de um número inteiro de 32 bits como identificador lógico.

O correto funcionamento da UNI está diretamente associado ao mapeamento dos identificadores lógicos da rede cliente com os da rede de transporte pois eles são utilizados no provisionamento de conexões. A UNI Pública permite a configuração manual desta tabela de correlação embora defina um mecanismo automático para efetuar a descoberta de vizinhança (Seção 3.1.4).

3.1.4 Mecanismo de Descoberta de Vizinhança e Manutenção do Canal de Controle

O procedimento de descoberta de vizinhança permite aos elementos das redes cliente e de transporte (TNEs), diretamente conectados, determinarem a identidade um do outro e quais portas remotas estão ligadas às suas portas locais. A manutenção do canal de controle permite aos TNEs e clientes a contínua monitoração e manutenção dos canais de controle IP disponíveis entre eles.

Ambos os procedimentos são definidos como opcionais na especificação da UNI Pública. Se a descoberta de vizinhança não é implementada, a identidade dos vizinhos e das portas remotas devem ser configuradas manualmente nas UNI-Cs e UNI-Ns correspondentes. Se a manutenção do canal de controle não é implementada como definida em [UNI Common Part, 2004], cabe ao protocolo de sinalização utilizado na implementação da UNI o desempenho deste procedimento. Um TNE pode estar ligado a múltiplos clientes e vice-versa. É possível que exista um ou mais IPCCs estabelecidos entre um cliente e um TNE.

A descoberta de vizinhança requer a habilidade por parte dos TNEs e equipamentos clientes de enviar e receber algumas mensagens *in-fiber* nos enlaces de dados que existem entre eles. O protocolo definido pela UNI Pública em [UNI Common Part, 2004] para gerar de forma automática a tabela de correlação de portas (descoberta de vizinhança) e a manutenção do canal de controle é o LMP [Lang, 2005] que é definido pelo IETF.

A Tabela 3.2 exemplifica a tabela de correlação de portas resultante do processo de descoberta de vizinhança. Cada nó possui em sua tabela o *Node ID* (normalmente um endereço IPv4), os identificadores de interface que podem ser um IPv4 (interface numerada) ou um identificador qualquer local (interface não-numerada) e outras informações referentes às portas (velocidade da porta, características de transmissão, etc) para cada uma das suas portas locais e respectivas portas remotas.

Tab. 3.2: Exemplo de Tabela de Correlação de Portas.

***Node ID* local**	***Interface ID* local**	**Outras informações**	***Node ID* remoto**	***Interface ID* remoto**	**Outras informações**
192.168.0.1	1	*Gigabit Ethernet*	192.168.1.1	5	*Gigabit Ethernet*
192.168.0.1	2	*Gigabit Ethernet*	192.168.1.1	3	*Gigabit Ethernet*
192.168.0.1	3	*Gigabit Ethernet*	192.168.1.1	8	*Gigabit Ethernet*
192.168.0.1	4	*SONET/SDH*	192.168.1.2	5	*SONET/SDH*
192.168.0.1	5	*DWDM*	192.168.1.3	2	*DWDM*

A partir do momento que um IPCC é ativado entre um TNE e um cliente (UNI-C e UNI-N) eles passam a trocar periodicamente mensagens de *Hello* para manter a conectividade. A periodicidade das mensagens de *Hello* é estabelecida durante a fase de configuração a partir da troca de um parâmetro

(*HelloInterval*) nas mensagens do LMP [Lang, 2005]. Se um nó envia mensagens de *Hello* mas não recebe *Hellos* em um intervalo definido como *HelloDeadInterval*, o IPCC correspondente deve ser declarado como *down*.

3.1.5 Mecanismo de Descoberta de Serviço

A descoberta de serviço é o procedimento pelo qual a UNI-C indica à UNI-N quais as características dos clientes que ela representa e, também, é o processo no qual a UNI-C obtém informações da UNI-N referentes aos serviços que a rede de transporte oferece. Apesar de ser um processo opcional, quando ele não é implementado é necessária a configuração manual das informações que seriam trocadas automaticamente. Existem quatro atributos de serviços definidos em [UNI Common Part, 2004] que utilizam a troca de mensagens LMP [Lang, 2005] como mecanismo:

1. Versão da UNI e dos protocolos de sinalização suportados: define quais versões da UNI e quais protocolos de sinalização (Por exemplo, RSVP ou LDP) são suportados.

2. Atributos de serviço das interfaces do cliente: descreve o tipo do enlace, os tipos de sinais, os níveis de transparência e os tipos de concatenação suportados sobre cada interface do cliente.

3. Suporte à transparência: descreve o nível de transparência suportado pela rede de transporte.

4. Suporte à diversidade de roteamento: define se a rede de transporte é capaz de gerar rotas que englobam nós, enlaces ou grupos de risco compartilhado (SRLG) diferentes ou não.

A descoberta de serviço é inicializada logo após o término do processo de descoberta de vizinhança. O processo consiste na troca de uma seqüência de mensagens LMP entre a UNI-C e a UNI-N conforme ilustra a Figura 3.4. As mensagens são trocados sobre um IPCC.

A Tabela 3.3 apresenta os objetos que constituem as três mensagens de *ServiceConfig* trocadas entre a UNI-C e a UNI-N. No processo de descoberta de serviço podem ser trocadas várias mensagens de *ServiceConfig* 2 e a correspondente *ServiceConfig Ack/Nack*. Cada uma destas mensagens corresponde a um enlace de dados estabelecido entre a UNI-C e a UNI-N. A mensagem de *ServiceConfig* 3 é enviada pela UNI-N à UNI-C somente após a configuração de todos os enlaces entre elas. Os eventos que podem disparar a descoberta de serviço são:

- a reinicialização do IPCC;

- a inserção de um novo enlace de dados;

- a alteração de algum atributo dos enlaces de dados;

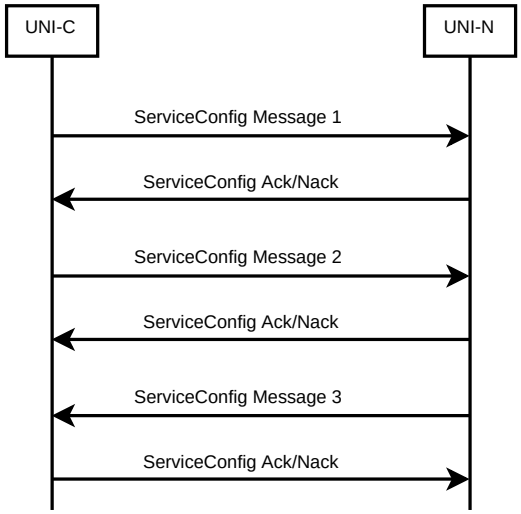

Fig. 3.4: Seqüência de mensagens na descoberta de serviço.

Tab. 3.3: Mensagens de *ServiceConfig*.

Número da mensagem de *Service Config*	**Objeto da mensagem de *Service Config***	**Direção da mensagem**
1	Protocolo de sinalização suportado	UNI-C → UNI-N
2	Atributos de serviço das interfaces do cliente	UNI-C → UNI-N
3	Transparência e diversidade da rede de transporte	UNI-N → UNI-C

- mudanças na versão da UNI ou do protocolo de sinalização;
- no instante em que a descoberta de serviço é habilitada;
- quando a verificação de enlaces é completada e são descobertas mudanças no mapeamento dos enlaces.

3.1.6 Mensagens Abstratas

As mensagens de sinalização especificadas pela UNI Pública em [UNI Common Part, 2004] são denominadas abstratas pois sua efetiva implementação depende do protocolo de sinalização utilizado. É descrito em [UNI RSVP Extensions, 2004] a implementação das mensagens abstratas da UNI com o protocolo de sinalização RSVP-TE. A Tabela 3.4 apresenta as mensagens abstratas da UNI Pública e a Tabela 3.5 traz o mapeamento entre as mensagens abstratas da UNI e as mensagens do RSVP.

O OIF não especifica a maneira pela qual as mensagens abstratas da UNI Pública são mapeadas em ações e nem qual o protocolo de sinalização deve ser usado no interior da rede de transporte. As mensagens abstratas da UNI Pública são descritas a seguir:

1. *Connection Create Request*: utilizada por um cliente para requisitar uma conexão através da

Tab. 3.4: Mensagens Abstratas da UNI Pública (UNI OIF).

Número da mensagem	**Mensagem Abstrata**	**Direção da mensagem**
1	*Connection Create Request*	UNI-C → UNI-N e UNI-N → UNI-C
2	*Connection Create Response*	UNI-N → UNI-C e UNI-C → UNI-N
3	*Connection Create Confirmation*	UNI-C → UNI-N e UNI-N → UNI-C
4	*Connection Delete Request*	UNI-C → UNI-N e UNI-N → UNI-C
5	*Connection Delete Response*	UNI-N → UNI-C e UNI-C → UNI-N
6	*Connection Status Enquiry*	UNI-C → UNI-N e UNI-N → UNI-C
7	*Connection Status Response*	UNI-N → UNI-C e UNI-N → UNI-C
8	*Notification*	UNI-N → UNI-C

rede de transporte entre uma origem e um destino específicos. Adicionalmente, define os atributos que descrevem os requisitos de tráfego.

2. *Connection Create Response*: indica o estabelecimento de uma conexão à UNI-C que iniciou o processo de estabelecimento. Após o recebimento desta mensagem o cliente pode iniciar a transmissão de seus dados.

3. *Connection Create Confirmation*: utilizada para indicar à UNI-C onde a conexão termina e do sucesso no estabelecimento da conexão. Utilizada em um processo de estabelecimento em três vias.

4. *Connection Delete Request*: inicia a remoção de uma conexão na rede de transporte. Durante a remoção a UNI-C onde a conexão inicia-se deve manter os estados de controle da conexão e o cliente correspondente deve manter o plano de dados até que a remoção seja confirmada. Este processo previne que o cliente do outro lado da rede de transporte emita alarmes.

5. *Connection Delete Response*: sinaliza a completa remoção de uma conexão. A UNI-C que iniciou o processo de remoção pode, ao receber esta mensagem, remover os estados de controle referentes a conexão em questão.

6. *Connection Status Enquiry*: é utilizada para pesquisar o estado e os atributos de uma conexão estabelecida na rede de transporte.

7. *Connection Status Response*: é a resposta à *Connection Status Enquiry*. Possui os atributos e o estado atual da conexão pesquisada.

8. *Notification*: enviada autonomamente pela UNI-N à UNI-C para indicar a mudança no estado de uma conexão (por exemplo, uma falha não restaurável).

Tab. 3.5: Mapeamento entre Mensagens Abstratas da UNI Pública e Mensagens RSVP-TE.

Número da mensagem	Mensagem Abstrata	Mensagem RSVP
1	*Connection Create Request*	*Path*
2	*Connection Create Response*	*Resv, PathErr*
3	*Connection Create Confirmation*	*ResvConf*
4	*Connection Delete Request*	*Path* ou *Resv* (bit *Deletion in Progress* setado)
5	*Connection Delete Response*	*PathErr* (*Path_State_removed flag*), *PathTear*
6	*Connection Status Enquiry*	não existe mapeamento
7	*Connection Status Response*	não existe mapeamento
8	*Notification*	*PathErr, ResvErr*

3.2 UNI IETF (UNI Privada)

A UNI Privada, especificada pelo IETF [Swallow et al., 2005], foi desenvolvida para operar em ambientes onde os elementos de rede que trocam mensagens de sinalização pertencem ao mesmo domínio administrativo. Neste ambiente, as interfaces podem ser utilizadas dentro de um mesmo sistema autônomo (AS) de roteamento ou entre múltiplos ASs. Apesar de trabalhar com o modelo *Overlay* (Seção 2.3.1) como a UNI Pública (Seção 3.1), a UNI Privada é ideal para ambientes que implementam o modelo Aumentado.

A UNI IETF é definida como uma interface de sinalização baseada no protocolo RSVP-TE [Braden et al., 1997, Awduche et al., 2001a, Berger, 2003a] que provê basicamente dois serviços: provisionamento de conexões e recuperação de falhas em múltiplas camadas. A primeira inclui o estabelecimento, remoção, modificação e consulta de estado de LSPs. A segunda inclui, entre outras, a notificação de falhas e o estabelecimento de caminhos alternativos em reposta a falhas.

É totalmente compatível com a UNI Pública definida pelo OIF, podendo inclusive operar da mesma forma que a UNI Pública opera. Por este motivo, esta seção apresenta somente as extensões à UNI Pública que são definidas pelo IETF, de tal forma que a UNI Privada seja totalmente compatível com o GMPLS [Mannie, 2004] e, também, opere em ambientes que implementam o Modelo Aumentado.

- Endereçamento: entre os nós onde a UNI-C e a UNI-N estão implementadas o espaço de endereçamento deve ser o mesmo. A UNI Privada define que o espaço de endereçamento usado pelos nós de núcleo da rede de transporte pode, mas não necessita, ser compartilhado entre os nós que implementam a UNI-C e a UNI-N. O nível de compartilhamento do espaço de endereçamento define o modelo de interconexão (*Overlay* ou Aumentado).

- Rota Explícita: a UNI Privada estabelece apenas uma sessão fim-a-fim entre a UNI-C de ingresso e a UNI-C de egresso. A Figura 3.5 ilustra esta característica da UNI Privada. Devido

ao compartilhamento do espaço de endereçamento que existe entre a UNI-C e a UNI-N, é possível que a UNI-C de ingresso inclua um objeto que contenha uma rota explícita fraca (ERO [Awduche et al., 2001a, Berger, 2003b]) com os endereços da UNI-N de ingresso, UNI-N de egresso e UNI-C de egresso por onde a UNI-C de ingresso deseja que a rota seja estabelecida. Conforme a visibilidade que a UNI-C de ingresso tem do interior da rede de transporte aumente (modelo Aumentado), é possível que ela introduza mais nós à rota explícita.

- Recuperação de falhas fim-a-fim: a UNI Privada permite o uso do objeto de proteção (*Protection Object* [Berger, 2003b]). Este objeto permite aos clientes solicitarem explicitamente qual o tipo de proteção fim-a-fim desejada para seus respectivos LSPs.

- Notificação de falhas: o RSVP-TE [Berger, 2003a] possui uma mensagem de notificação (*Notify Message*) que provê um mecanismo mais eficiente para notificar nós não adjacentes sobre eventos (geralmente falhas) relacionados aos LSPs. A mensagem de notificação é gerada somente quando explicitamente solicitada no momento do estabelecimento dos LSPs através da inclusão do objeto *Notify_Request* nas mensagens de *Path* e *Resv*. A mensagem de notificação é mais eficiente do que as mensagens de *PathError* e *ResvError* pois é entregue diretamente, em caso de falha, ao nó cujo endereço foi especificado no momento do estabelecimento do LSP.

Na Seção 3.3 é apresentada uma comparação entre a UNI Pública (OIF) e a UNI Privada (IETF). Cabe aqui uma nota sobre a UNI Privada que será detalhada em maior profundidade na Seção 3.3. A UNI Privada utiliza o RSVP-TE GMPLS [Berger, 2003a, Berger, 2003b] como protocolo de sinalização e assume que este mesmo protocolo é utilizado no interior da rede de transporte. Neste cenário ideal, somente uma sessão (sessão 1 da Figura 3.5) é necessária no estabelecimento de um LSP através da rede de transporte. A Seção 3.3 apresenta o cenário onde não temos o RSVP-TE GMPLS no interior da rede de transporte.

Ainda na Figura 3.5 existe uma segunda sessão (sessão 2) que compreende a sessão fim-a-fim entre os elementos de origem e destino da rede cliente. Neste cenário, o protocolo de sinalização utilizado é o RSVP-TE MPLS [Awduche et al., 2001a]. A partir do momento que um LSP é estabelecido através da rede de transporte, os nós onde as UNI-Cs de ingresso e egresso do LSP estão implementadas passam a ser vizinhos, ou seja, a rede cliente enxerga este LSP como um enlace qualquer entre a origem e o destino fim-a-fim por onde sua sinalização, seja ela qual for, passa como se estivesse passando por um enlace entre dois nós quaisquer.

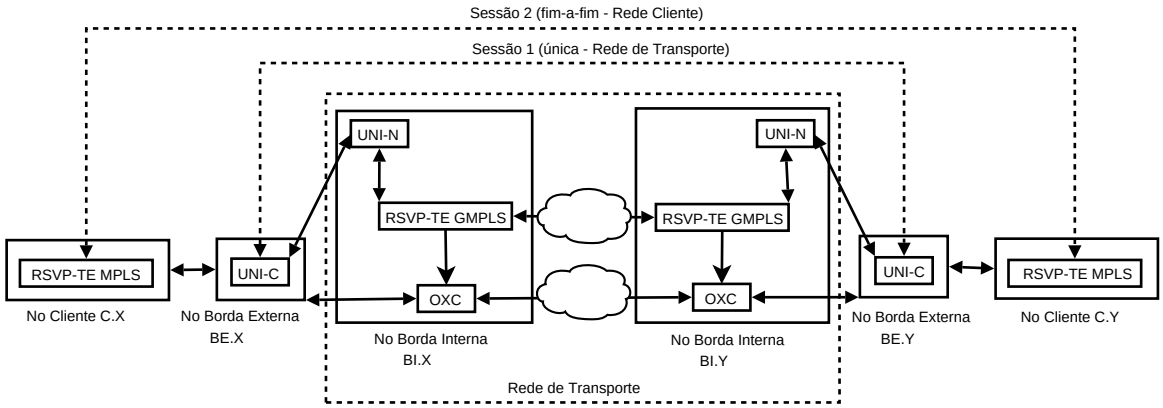

Fig. 3.5: Sessões estabelecidas pela UNI Privada.

3.3 UNI OIF x UNI IETF

Conforme apresentado nas seções 3.1 e 3.2, existem duas especificações da UNI. Uma desenvolvida pelo OIF também conhecida como UNI Pública [UNI Common Part, 2004] e outra desenvolvida pelo IETF chamada de UNI Privada [Swallow et al., 2005]. Ambas baseiam-se na sinalização GMPLS. A primeira foi especificada para atuar no modelo *Overlay* e é ideal para ambientes com domínios administrativos diferentes. A segunda alinha-se ao modelo Aumentado e é ideal para ambientes com o mesmo domínio administrativo. A UNI Privada oferece mais funcionalidades e possibilidades de extensão em relação a UNI Pública, podendo inclusive trabalhar no modelo *Overlay* da mesma forma que a UNI Pública.

A UNI Pública assume total separação entre os planos de controle das redes cliente e de transporte. Pela característica de não confiabilidade entre os domínios administrativos inerente ao Modelo Overlay, ela não faz troca de qualquer informação de alcançabilidade ou topologia. Define apenas mensagens para o provisionamento de caminhos através da rede de transporte e para verificar o estado destes. As requisições feitas à rede de transporte por parte da UNI Pública seguem um Acordo de Nível de Serviço (SLA) pré-estabelecido.

Esta separação dos planos de controle é ao mesmo tempo sua melhor e sua pior característica. De um lado, este fator limita a quantidade de informações de controle que precisam ser trocadas entre os domínios e, por outro lado, como o cliente vê a rede de transporte como uma “caixa preta”, pouca coordenação entre as redes é obtida. Desta forma, é necessária uma sobreposição de funções. Por exemplo, sobreposição de sinalização, roteamento e ações de engenharia de tráfego. Na prática, isto significa uma grande dificuldade em utilizar de forma otimizada os recursos oferecidos pela rede de transporte, especialmente quando existe uma reconfiguração muito intensa nos caminhos ópticos estabelecidos devido à dinâmica de tráfego ou mesmo à ocorrência de falhas.

Embora a UNI Pública seja baseada na sinalização GMPLS, ela define algumas extensões

específicas do OIF que não são compatíveis com o GMPLS. Por exemplo, o objeto *Generalized UNI*. Para manter a total independência entre as redes cliente e de transporte, a UNI Pública trabalha com o estabelecimento de três sessões na rede de transporte que são apresentadas na Figura 3.6. A primeira sessão (sessão 1) é estabelecida entre o cliente e a rede de transporte no ingresso da rede; a segunda sessão (sessão 2) é estabelecida entre os elementos ópticos da rede de transporte; a terceira sessão (sessão 3) é estabelecida entre a rede de transporte e o cliente no egresso da rede. Uma sessão é composta do endereço IP de destino mais o identificador (ID) do caminho criado. Na UNI Pública os endereços das três sessões são diferentes do endereço da sessão do cliente (sessão 4). Esta diferença nos endereços das sessões, somada à falta de informações de alcançabilidade por parte da UNI-C, torna impossível a resolução do endereço do elemento da rede de transporte (TNE) onde o caminho óptico irá terminar. O objeto *Generalized UNI* resolve este problema guardando o endereço da sessão cliente (sessão 4) durante todas as fases do estabelecimento do caminho através da rede de transporte (sessões 1,2 e 3) possibilitando a resolução do endereço no interior da rede de transporte.

A vantagem do estabelecimento de três sessões distintas na rede de transporte é a total independência quanto ao protocolo de sinalização utilizado em cada uma das sessões. É possível que uma sessão utilize o RSVP-TE e outra sessão utilize o CR-LDP ou mesmo um protocolo de sinalização proprietário. Necessita-se apenas a correta tradução do objeto *Generalized UNI*.

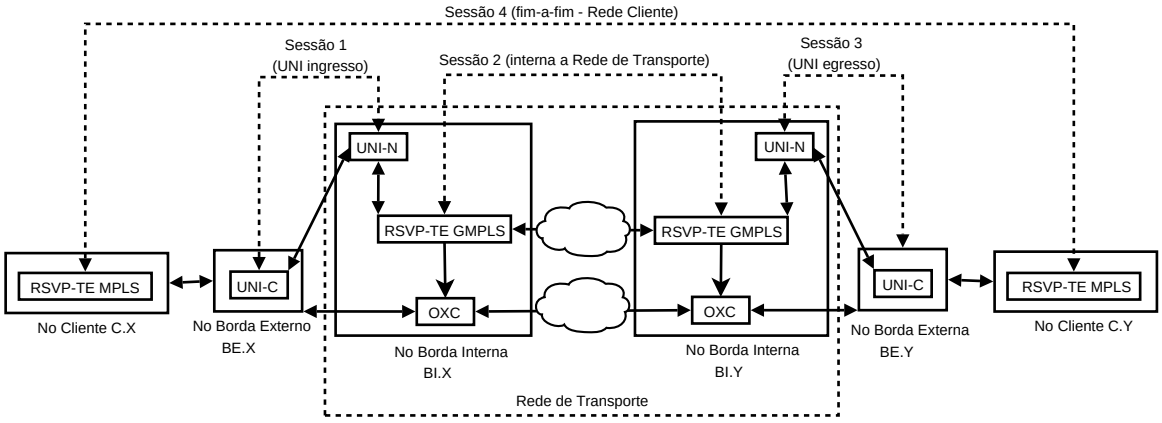

Fig. 3.6: Sessões estabelecidas pela UNI Pública.

A UNI Pública precisaria sofrer mais modificações para poder atuar em cenários onde há uma certa confiança entre as redes cliente e de transporte. Tendo isto em vista, o IETF resolveu fazer sua própria especificação totalmente compatível com o GMPLS (UNI Privada). Esta nova interface oferece uma maior integração entre os planos de controle pois trabalha com os protocolos da família GMPLS (RSVP-TE, OSPF-TE, LMP). Foi desenvolvida para ambientes com o mesmo domínio administrativo mas também pode operar em domínios administrativos diferentes como faz a UNI Pública.

Como apresentado na Seção 3.2, a UNI Privada permite, mas não obriga, que seja utilizado o mesmo espaço de endereçamento presente no interior da rede de transporte entre a UNI-C e a UNI-N. Este fator regula o grau de visibilidade que o cliente tem da rede de transporte e permite a utilização de novos recursos como rotas explícitas que contenham maiores ou menores detalhes.

Em um cenário ideal, onde o interior da rede de transporte implementa o RSVP-TE GMPLS [Berger, 2003a, Berger, 2003b] como protocolo de sinalização, somente uma sessão é necessária como ilustrado na Figura 3.5. Entretanto, caso a rede de transporte implemente outro protocolo de sinalização, serão necessárias o estabelecimento de três sessões como a UNI Pública faz e apresentado na Figura 3.6.

3.4 Resumo

Este capítulo apresentou as duas especificações existentes da UNI (UNI OIF e UNI IETF) através de uma descrição das funcionalidades comuns à ambas especificações na seção onde foi tratada a UNI do OIF e, em seguida, foram apresentadas as extensões feitas à UNI pelo IETF.

Finalizando este capítulo, foi feito um estudo comparativo entre as especificações da UNI a fim de facilitar a comparação entre elas e, também, com a UNI proposta para o Projeto GIGA no próximo capítulo.

Capítulo 4

Proposta de Arquitetura da UNI

Este capítulo apresenta a Arquitetura da UNI proposta, a principal contribuição deste trabalho. Ele está dividido em duas seções principais. A primeira seção traz o detalhamento da Arquitetura proposta de uma forma generalizada e apresenta todos os seus módulos e características que a tornam diferente das outras especificações da UNI, ou seja, apresenta a proposta elaborada com base nos estudos das especificações do OIF e IETF. A segunda parte do capítulo traz o detalhamento da utilização da Arquitetura proposta no cenário da rede óptica do Projeto GIGA. Nesta segunda parte existe um detalhamento da especialização dos módulos da Arquitetura para que eles possam atuar em conjunto com os protocolos presentes na rede do Projeto GIGA, mantendo sempre inalterada a semântica dos protocolos envolvidos, e solucionar as restrições impostas pelos equipamentos do GIGA.

4.1 Componentes da Arquitetura Proposta para a UNI

Conforme apresentado no capítulo 3, a função da UNI é prover a interação entre a rede cliente e a rede de transporte. Entretanto, para que a rede cliente possa efetuar estas requisições, ela precisa estar ciente da existência da UNI. Normalmente, estar ciente da existência da UNI significa que a rede cliente a implementa.

A rede cliente do Projeto GIGA é constituída por equipamentos comerciais que não implementam a UNI. Além disso, não existe a possibilidade de efetuar qualquer tipo de atualização no software destes equipamentos ou mesmo inserir novas funcionalidades. Para somar a estas restrições impostas pela rede cliente do Projeto GIGA, não existe uma especificação de como deve ocorrer a interação entres os protocolos de sinalização presentes nas redes cliente e de transporte com os módulos UNI-C e UNI-N. Segundo o OIF, é necessária a utilização de soluções proprietárias no desempenho desta função.

Considerando todos estes fatores, este trabalho propõe uma Arquitetura de implementação para

a UNI, visando solucionar o problema de interação entre a rede cliente e a rede de transporte de forma totalmente transparente, ou seja, todos os equipamentos pertencentes à rede cliente e à rede de transporte não precisam sofrer qualquer modificação e, principalmente, a arquitetura proposta preserva a semântica dos protocolos de sinalização existentes nestas redes.

Partindo do princípio de que nada nas redes envolvidas pode ser alterado, é preciso encontrar nelas um mecanismo que possa ser utilizado como gatilho para disparar ações na UNI. O mecanismo ideal é o protocolo de sinalização presente no plano de controle destas redes. Entretanto, é comum que entre diversas redes clientes, ou mesmo entre a rede cliente e a rede de transporte, existam diferentes tipos de protocolos de sinalização, tornando complexa a implementação da UNI. Para solucionar este problema, dois novos módulos foram adicionados à UNI para tratar a sinalização das redes clientes e da rede de transporte, chamados de SOP (*Signaling Overlaying Point*) e SMP (*Signaling Mapping Point*), respectivamente. Estes módulos são detalhados nas seções 4.1.1 e 4.1.4.

O capítulo 3 apresentou em detalhes as duas especificações existentes da UNI (UNI OIF e UNI IETF). A UNI proposta neste trabalho é baseada nestas especificações. Entretanto, devido às imposições do cenário real, algumas características da arquitetura proposta são diferentes das especificações existentes. Conseqüentemente, apenas as características que a diferem das especificações do OIF e IETF são apresentadas neste capítulo.

A Figura 3.1 apresenta as quatro possibilidades de invocação de serviço oferecidas pela UNI do OIF. Em todos os quatro modelos, é necessário que tanto os equipamentos da rede cliente quanto os equipamentos da rede de transporte saibam como invocar a UNI, seja a invocação através de uma interface proprietária ou mesmo através de uma interface que implementa a UNI. A arquitetura proposta se assemelha, mas não é exatamente igual, ao modelo 2b da Figura 3.1, pois nem a UNI-C nem a UNI-N são implementadas pelos equipamentos que constituem as redes. O diferencial está no fato de que além de não implementarem a UNI, os equipamentos não precisam conhecer nenhuma API para pode ter acesso a ela. Esta tarefa, na arquitetura proposta, é responsabilidade dos módulos SOP e SMP.

Outra característica muito importante da arquitetura proposta, que também a difere das outras propostas, é referente às sessões que são estabelecidas internamente à rede de transporte. As figuras 3.5 e 3.6 ilustram as sessões que são estabelecidas pelas especificações da UNI feitas pelo IETF e OIF respectivamente. Devido ao nível de portabilidade oferecido pela arquitetura proposta, é possível garantir que somente duas sessões são estabelecidas no interior da rede de transporte, independentemente do protocolo de sinalização utilizado. A Figura 4.1 apresenta como são estabelecidas as sessões na arquitetura proposta para a UNI.

A sessão 1 é estabelecida pelo protocolo de sinalização presente no plano de controle da rede de transporte. A sessão 2 é estabelecida entre a UNI-C de ingresso e a UNI-C de egresso. O endereço da

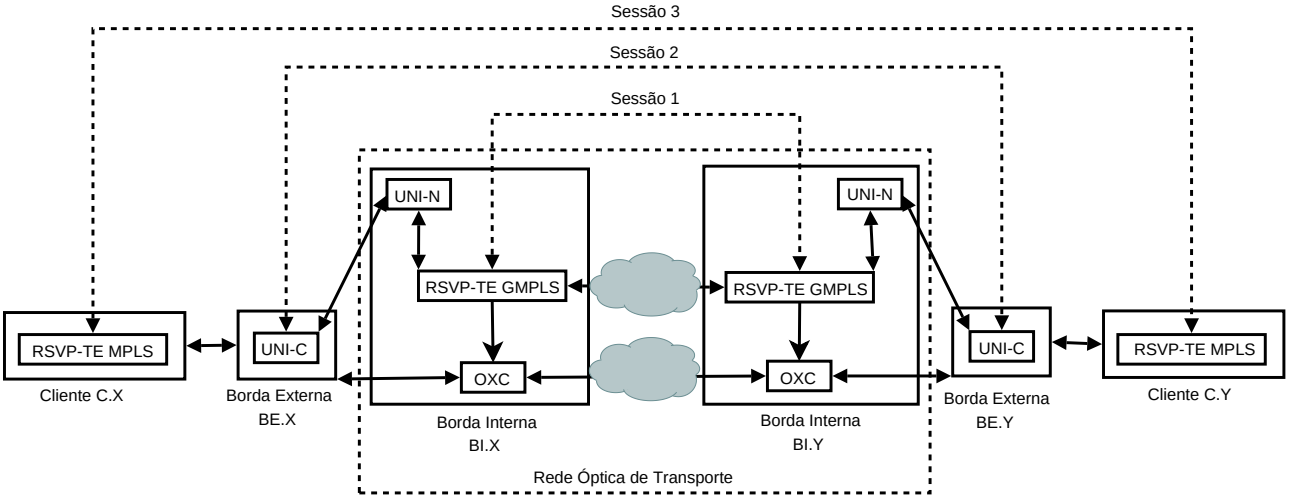

Fig. 4.1: Sessões estabelecidas em um ambiente utilizando a Arquitetura proposta.

UNI-C de egresso é obtido através de uma consulta à tabela de alcançabilidade localizada na UNI-C de ingresso. Para que esta sessão seja estabelecida, é preciso que o protocolo de sinalização da rede de transporte carregue, de forma transparente, o objeto *Generalized UNI* até a UNI-N de egresso. A sessão 3 representa a sessão da rede cliente. Os módulos SOP e SMP são responsáveis por manter a semântica dos protocolos de sinalização envolvidos nas sessões 1 e 3. Os protocolos utilizados na figura (MPLS RSVP-TE e GMPLS RSVP-TE) são apenas ilustrativos.

Para a arquitetura da UNI proposta, elegeu-se o GMPLS RSVP-TE como protocolo de sinalização para implementar as mensagens abstratas da UNI, devido ao fato de ser um protocolo de sinalização não proprietário, atender aos requisitos funcionais da UNI e, também, existir um documento elaborado pelo OIF [UNI RSVP Extensions, 2004] onde é feito o mapeamento das mensagens abstratas da UNI em mensagens do protocolo GMPLS RSVP-TE. O canal de controle entre a UNI-C e UNI-N se dá através de um comprimento de onda específico, ou seja, é *Out-of-band In-fiber*. A arquitetura proposta é apresentada na Figura 4.2 e as seções a seguir detalham seus módulos.

4.1.1 SOP

O SOP (*Signaling Overlaying Point*) é o módulo responsável por interceptar e traduzir a sinalização da rede cliente, seja ela qual for, para uma invocação à API da UNI. Desta forma, é possível que a rede cliente utilize os recursos oferecidos pela rede de transporte, sem que haja a necessidade da rede cliente sofrer qualquer tipo de adaptação para poder se comunicar com a UNI. O SOP, basicamente, atua como um adaptador de protocolos de sinalização.

Para que o SOP consiga interceptar e traduzir a sinalização da rede cliente em uma invocação correspondente à API da UNI, é necessário que ele conheça o protocolo de sinalização que a rede

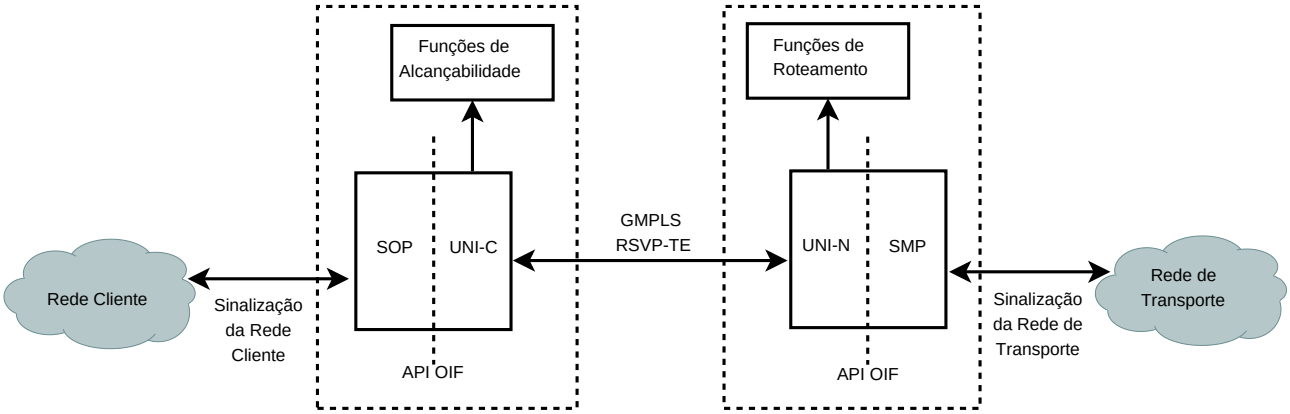

Fig. 4.2: Arquitetura de Implementação da UNI.

cliente implementa, pois ele precisa preservar a semântica da rede cliente durante o período em que o caminho de luz é estabelecido através da rede de transporte. Em contrapartida, geralmente os protocolos de sinalização possuem um comportamento bem definido para desempenhar suas funções, ou seja, existe toda uma especificação onde as mensagens de sinalização e a semântica empregada no tratamento delas é detalhada.

É possível que existam várias redes clientes solicitando conexões à rede de transporte e que estas diferentes redes clientes implementem vários protocolos de sinalização. Para resolver este problema, a arquitetura permite que vários SOPs sejam instanciados e associados a uma mesma UNI-C, cada um deles tratando um determinado protocolo de sinalização.

4.1.2 UNI-C

Após a tradução da mensagem por parte do SOP, ele invoca a API da UNI-C solicitando o serviço indicado pela rede cliente. A UNI-C é responsável pelas ações de controle de admissão. São elas:

- Autenticação: verificar se a rede cliente, indicada nas requisições, está habilitada a solicitar serviços à rede de transporte;
- Disponibilidade de recursos: a UNI-C verifica se a quantidade de recursos solicitados pela rede cliente podem: 1) ser oferecidos à rede cliente com base em contratos (SLAs), 2) se a rede de transporte oferece aquele tipo de recurso e 3) se existem os recursos solicitados para atender a requisição da rede cliente.
- Alcançabilidade: a UNI-C verifica se o destino requisitado pela rede cliente pode ser alcançado através da rede de transporte. Através da função de alcançabilidade, a UNI-C consegue descobrir qual o endereço do elemento de rede que pertence à rede de transporte onde a

conexão da rede de transporte irá terminar. O desempenho desta função pela UNI-C é um dos diferenciais da arquitetura proposta e será detalhado na Seção 4.1.5.

- Registro de conexões: a UNI-C mantém em seu interior um registro das conexões que estão estabelecidas na rede de transporte. Este registro possui o estado atual destas conexões e permite que os clientes verifiquem o estado de suas conexões.

A UNI possui algumas funcionalidades de controle que são desempenhadas entre a UNI-C e a UNI-N. Estas atividades da UNI-C e da UNI-N foram detalhadas no Capítulo 3. São exemplos destas atividades:

- Descoberta de vizinhança;
- Descoberta de serviço;
- Manutenção do canal de controle;

4.1.3 UNI-N

Uma vez terminados os procedimentos da UNI-C, ela encaminha as requisições da rede cliente à UNI-N através de mensagens implementadas pelo protocolo GMPLS RSVP-TE. Uma das funções da UNI-N é solicitar ao protoloco de roteamento presente na rede de transporte a atribuição de uma rota para o endereço de destino que foi resolvido na UNI-C através da tabela de alcançabilidade. Para desempenhar esta função, a UNI-N precisa implementar a API do protocolo de roteamento. Assim que ela recebe o retorno do protocolo de roteamento, é necessário disparar a sinalização da rede de transporte para efetivar o estabelecimento da conexão através dela. No entanto, é possível que a rede de transporte implemente outro protocolo de sinalização que não seja o mesmo utilizado na implementação da UNI. Além disso, não é interessante para a arquitetura estar atrelada ao protocolo de sinalização que o plano de controle da rede de transporte implementa. Para resolver estes problemas, a arquitetura possui um módulo chamado SMP.

No instante em que a UNI-N recebe a resposta do protocolo de roteamento, ela passa a conhecer o endereço do elemento de rede (de egresso) da rede de transporte onde a conexão de transporte irá terminar. Ao obter esta informação, a UNI-N é capaz de preencher o objeto *Generalized UNI*, ou seja, a UNI-N possui o identificador de sessão da UNI e do identificador de sessão da conexão da rede de transporte. O objeto *Generalized UNI* deve ser carregado pelo protocolo de sinalização presente no interior da rede de transporte de forma transparente, para quando a sinalização atingir a UNI-N no egresso da rede de transporte, esta última consiga recuperar o identificador de sessão da UNI. Para disparar a sinalização na rede de transporte a UNI-N utiliza o SMP. O SMP faz a

tradução da sinalização da UNI para o protocolo de sinalização da rede de transporte, ou seja, dispara o estabelecimento da conexão na rede de transporte.

4.1.4 SMP

O SMP (*Signaling Mapping Point*) provê portabilidade à arquitetura referente ao protocolo de sinalização utilizado pelo plano de controle da rede de transporte. Através da utilização deste módulo, a semântica dos protocolos de sinalização da UNI e do plano de controle da rede de transporte são preservadas, ou seja, o SMP é capaz de prover a comunicação entre a UNI-N e a rede de transporte sem que haja a necessidade de ambas as entidades implementarem a API uma da outra.

Similarmente ao SOP, é possível que existam diferentes SMPs associados a uma mesma UNI-N. Esta característica permite que a arquitetura proposta seja utilizada em diversas redes de transporte sem que haja a necessidade da UNI sofrer adaptações referentes ao protocolo de sinalização utilizado por estas redes.

Conforme mencionado anteriormente, uma importante característica da arquitetura diz respeito às sessões que são estabelecidas no interior da rede de transporte. Esta característica está diretamente associada à existência do módulo SMP, pois a rede de transporte precisa carregar de forma transparente o objeto *Generalized UNI* e o SMP é o módulo responsável pela tradução das mensagens de sinalização entre a UNI e o plano de controle da rede de transporte. A correta manutenção dos identificadores de sessão que são carregados pelo objeto *Generalized UNI* é fundamental para o correto estabelecimento das conexões no interior da rede de transporte.

4.1.5 Funções de Alcançabilidade

A consulta à tabela de alcançabilidade realizada pela UNI-C é um dos diferenciais da arquitetura proposta. Na especificação do OIF esta função é realizada pela UNI-N. A localização da função de alcançabilidade está relacionada às sessões que são estabelecidas pela UNI. No caso da UNI do OIF, o desempenho desta função pela UNI-N está amparada pelo fato de serem estabelecidas três sessões independentes no interior da rede de transporte. A Figura 3.6 indica as sessões que são estabelecidas pela UNI do OIF. A sessão 1 da Figura 3.6 corresponde ao momento do recebimento da requisição da rede cliente pela rede de transporte. O identificador desta sessão é o endereço IP da UNI-N à qual a UNI-C está ligada via canal de controle, ou seja, a UNI-C já possui este endereço, pois ele foi descoberto durante o processo de descoberta de vizinhança. Conseqüentemente, a consulta à tabela de alcançabilidade para resolver o endereço do elemento de rede onde a conexão na rede de transporte irá terminar será efetuada somente quando a requisição alcançar a UNI-N de ingresso.

Na arquitetura proposta é estabelecida apenas uma sessão entre a UNI-C de ingresso e a UNI-C de

egresso e esta sessão é totalmente independente da sessão estabelecida pelo protocolo de sinalização presente no plano de controle da rede de transporte. Na Figura 4.1 é possível verificar as sessões estabelecidas pela arquitetura proposta. A sessão 2 é estabelecida pela UNI e o identificador desta sessão é o endereço IP do elemento de rede onde a UNI-C de egresso está instalada, ou seja, BE.Y como ilustrado na figura. Como a sessão inicia na UNI-C de ingresso, cabe a ela resolver o endereço da UNI-C de egresso, por este motivo a função de alcançabilidade está localizada na UNI-C e não na UNI-N.

4.1.6 Funções de Roteamento

Esta é uma função muito importante da arquitetura proposta, pois visa prover um melhor aproveitamento dos recursos disponíveis na rede de transporte e, ao mesmo tempo, oferecer QoS à rede cliente. A UNI-N ao receber a mensagem de sinalização vinda da UNI-C, no ingresso da rede de transporte, faz uma consulta à função de roteamento solicitando uma rota e um comprimento de onda no qual a conexão será estabelecida através da rede de transporte. O retorno desta consulta, uma rota e um comprimento de onda, é passado ao SMP que por sua vez traduz estes parâmetros para o formato do protocolo de sinalização presente no interior da rede de transporte. A UNI-N é o módulo da arquitetura responsável pela interação com o mecanismo de roteamento existente na rede de transporte. Existe em nosso grupo uma pesquisa voltada às funções de roteamento para a rede do Projeto GIGA [Zuliani, 2006].

4.1.7 Considerações sobre a Arquitetura proposta

O mecanismo que dispara o funcionamento da arquitetura proposta no lado da rede cliente é o protocolo de sinalização implementado pela rede cliente, ou seja, ao interceptar uma mensagem de sinalização, vinda da rede cliente, o SOP dispara o processo de sinalização da UNI. Entretanto, é possível que existam cenários onde as redes clientes não possuam nenhum protocolo de sinalização. Neste caso, é considerado que existe uma entidade de gerência de rede responsável pela manutenção das conexões que são estabelecidas na rede de transporte.

Para que seja disparado o processo de sinalização da UNI neste caso, fica a cargo desta entidade de gerência de redes implementar a API da UNI-C para que o gerente da rede possa solicitar o provisionamento de conexões diretamente à UNI, como é feito pelo módulo SOP da arquitetura proposta.

4.2 Utilização da Arquitetura Proposta no Projeto GIGA

A utilização da arquitetura proposta em outras redes dependerá do conhecimento dos mecanismos, hardware e software, oferecidos pela rede para que seja possível definir o equipamento onde a UNI será implementada e especializar os módulos SOP e SMP para atuar com os protocolos de sinalização disponíveis na rede. A rede do Projeto GIGA, em sua porção cliente, é constituída por roteadores comerciais que implementam o MPLS e o protocolo de sinalização existente nestes equipamentos é o MPLS RSVP-TE. Outra restrição dos equipamentos comerciais utilizados é o fato de não ser possível a implantação de novas funcionalidades nestes equipamentos. Conseqüentemente, a UNI precisa ser implementada em um outro equipamento. A rede de transporte do Projeto GIGA é uma rede óptica WDM constituída por OxCs que também estão sendo desenvolvidos em um dos sub-projetos do Projeto GIGA. O plano de controle destes equipamentos (GMPLS) será implementado em computadores de alto-desempenho e cada OxC terá anexado a ele um destes computadores.

Para solucionar a impossibilidade de introduzir a UNI nos equipamentos que constituem a rede cliente do Projeto GIGA, serão utilizados computadores de alto-desempenho na borda da rede de transporte onde a UNI será implementada. Estes computadores implementarão as funcionalidades da parte cliente da UNI, ou seja, UNI-C, SOP e Tabela de Alcançabilidade e estarão no espaço de endereçamento da rede cliente, ou seja, eles serão vizinhos dos equipamentos que constituem a rede cliente. Conseqüentemente, outras funcionalidades como o OSPF e o MPLS RSVP-TE estarão presentes nestes computadores para que eles se integrem à rede cliente adequadamente. A Figura 4.3 ilustra o posicionamento dos equipamentos utilizados na rede do Projeto GIGA.

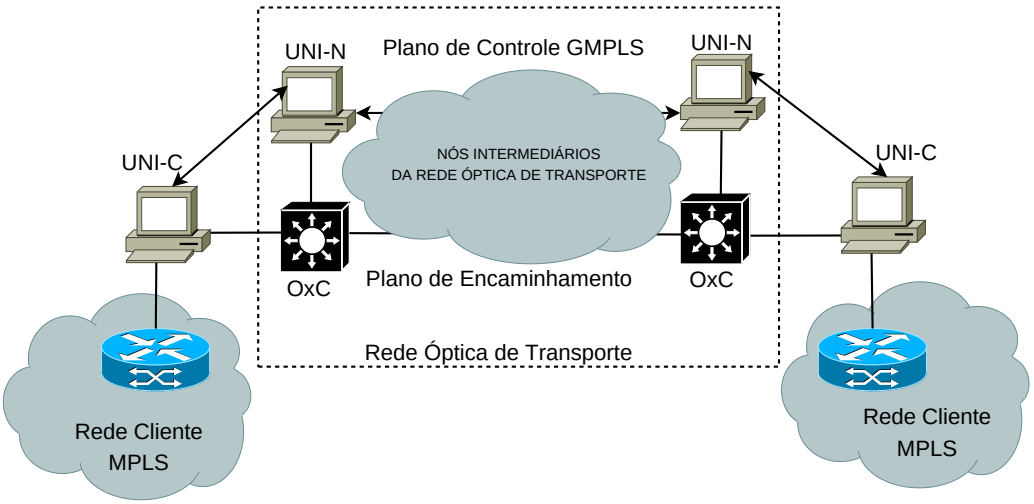

Fig. 4.3: Posicionamento dos elementos de rede no Projeto GIGA.

Um dos principais interesses do Projeto é o desenvolvimento de tecnologias independentes de plataforma e fabricantes. Levando estes fatores em consideração, o plano de controle da rede de

transporte do Projeto GIGA utiliza protocolos especificados por órgãos de padronização como o IETF. A Figura 4.4 apresenta como a arquitetura proposta para a UNI é instanciada no Projeto GIGA.

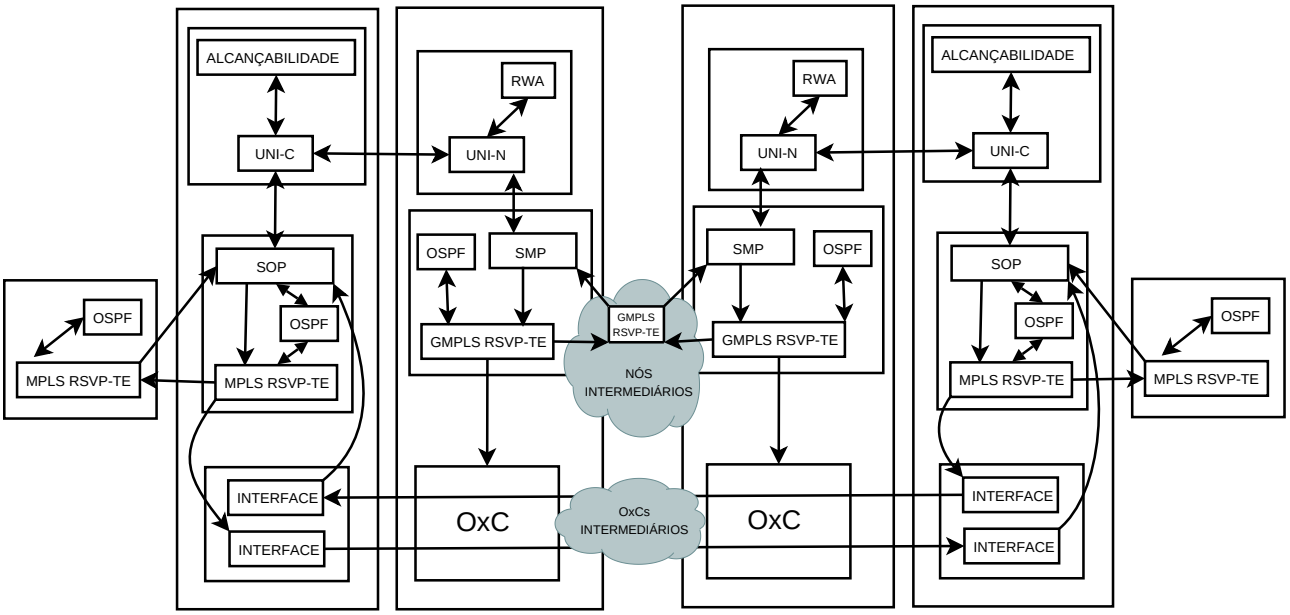

Fig. 4.4: Utilização da Arquitetura Proposta no Projeto GIGA.

A partir da figura é possível verificar quais são os protocolos utilizados na rede GIGA. Os equipamentos da rede cliente, além de implementarem o protocolo de sinalização MPLS RSVP-TE, possuem o protocolo de roteamento OSPF. O plano de controle da rede de transporte utiliza o protocolo de sinalização GMPLS RSVP-TE e o protocolo de roteamento OSPF-TE que trabalha em conjunto com o algoritmo de atribuição de rotas e comprimentos de onda RWA. A seguir são apresentados detalhes da instanciação dos módulos da arquitetura proposta no cenário da rede do Projeto GIGA. Os diagramas de seqüência que descrevem as ações desempenhadas pela arquitetura, especialmente pelo SOP e pelo SMP, durante o estabelecimento e a remoção de LSPs são apresentados no Apêndice A.

4.2.1 O SOP e a Sinalização da Rede Cliente do Projeto GIGA

A funcionalidade do SOP é prover um mecanismo de invocação à UNI para a rede cliente sem que esta última sofra qualquer tipo de adaptação. No caso do Projeto GIGA, a rede cliente implementa o MPLS RSVP-TE. Neste caso, o SOP precisa conhecer o MPLS RSVP-TE para que possa entender a sinalização da rede cliente e ser capaz de traduzi-la para uma invocação da API da UNI. Desta forma, o SOP mantém preservada a semântica do MPLS RSVP-TE. Caso a rede cliente do Projeto GIGA venha a implementar outro protocolo, é necessário implementar um novo SOP para tratar este novo

protocolo de sinalização.

O RSVP foi brevemente introduzido na Seção 2.2.1. A seguir é explicado como o SOP faz a interação entre as mensagens do MPLS RSVP-TE e a UNI-C no estabelecimento de conexões, remoção de conexões e ocorrência de falhas.

- Estabelecimento de Conexões: Conforme mencionado anteriormente, os computadores onde as funcionalidades da UNI cliente estarão implementadas fazem parte do espaço de endereçamento da rede cliente. Logo, estes computadores são vistos como próximos *hops* na direção do destino que a rede cliente pretende alcançar, ou seja, eles fazem parte do sistema autônomo de roteamento da rede cliente. No caso do Projeto GIGA, o SOP é implementado de forma a atuar como se fosse o MPLS RSVP-TE presente neste computador de borda, ou seja, ele recebe as mensagens de sinalização e as intercepta em nome do MPLS RSVP-TE. O RSVP é um protocolo geralmente utilizado diretamente sobre o IP mas que também pode ser utilizado sobre o UDP.

  No caso do estabelecimento de novas conexões, o SOP interceptará uma mensagem de PATH contendo os parâmetros referentes à nova conexão que a rede cliente deseja estabelecer com um determinado destino. Neste instante, o SOP efetua as seguintes ações:

  1. Faz uma consulta à Tabela de Roteamento presente no computador onde está instalado para descobrir qual o próximo nó da direção do destino. Como se trata de um novo estabelecimento de conexão, não haverá um próximo nó. No caso do Projeto GIGA, a tabela de roteamento é gerada pelo protocolo OSPF presente na rede cliente.
  2. Como não é retornado nenhum endereço pela tabela de roteamento, o SOP mapeia a mensagem de PATH da rede cliente em uma invocação à API da UNI-C, ou seja, *Connection Create Request*, armazena a mensagem original de PATH da rede cliente em seu interior e aguarda o retorno da UNI-C.
  3. Ao receber o retorno da UNI-C, *Connection Create Response*, o SOP recupera a mensagem de PATH da rede cliente e a envia ao MPLS RSVP-TE que está implementado no computador junto a ele, ou seja, o real destino da mensagem que foi interceptada por ele. Todo este processo significa que um novo caminho de luz foi estabelecido através da rede de transporte e, conseqüentemente, os dois nós da rede cliente que estão situados nas bordas da rede de transporte passam a ser vizinhos ligados por este novo caminho de luz (enlace TE) e trocam mensagens se sinalização normalmente. O resultado final deste processo traduz-se em transparência à sinalização da rede cliente.

Devido ao mecanismo de atualização do estado das conexões do RSVP (*SoftState*) em intervalos de tempo definidos, mensagens de PATH e RESV são trocadas para atualizar o estado dos caminhos estabelecidos. Neste caso, ao consultar a tabela de roteamento o SOP receberá um retorno, ou seja, ele não precisa efetuar nenhuma outra ação além de entregar a mensagem interceptada ao MPLS RSVP-TE local e deixar que ele efetue suas ações normalmente.

- Remoção de Conexões: A remoção de conexões do RSVP ocorre através da troca de mensagens de PATHTEAR e RESVTEAR, pois ela pode ser iniciada tanto pela origem quanto pelo destino da conexão. Ao interceptar uma destas mensagens, o SOP age da seguinte maneira:

  1. Mapeia a mensagem recebida em uma invocação à API da UNI-C, *Connection Delete Request*. Este procedimento irá disparar a remoção do caminho de luz estabelecido através da rede de transporte.
  2. Paralelamente, o SOP envia esta mensagem ao MPLS RSVP-TE presente no nó onde está implementado para que este, por sua vez, continue a remover a conexão da rede cliente. É desejável que este procedimento ocorra ao mesmo tempo em que é feita a invocação à API da UNI, pois a mensagem da rede cliente precisa passar pelo caminho de luz ainda estabelecido para que possa atingir seu nó vizinho situado do outro lado da rede de transporte. Caso ocorra uma falha, ou seja, o caminho de luz seja removido primeiro, a conexão será removido mesmo assim devido ao *time-out* do protocolo RSVP.

- Ocorrência de Falhas: Este é o caso mais simples do funcionamento do SOP, onde ele atua apenas como um tradutor de mensagens. Um exemplo de uma falha que pode ocorrer é no momento em que a UNI-C efetua o controle de admissão. Suponha que não existam recursos disponíveis na rede de transporte ou que o cliente requisitante não está autorizado a solicitar recursos à rede de transporte. Nestes casos, a UNI-C irá retornar uma resposta, *Connection Create Response*, indicando o erro e o SOP por sua irá mapear esta resposta em mensagens de PATHERROR ou RESVERROR dependendo de onde estas falhas ocorrem, e contendo os devidos códigos de erro.

A partir dos cenários descritos acima, é possível averiguar que o SOP proporciona uma solução de automatização para integrar a rede cliente e a rede de transporte de tal forma que a transparência entre elas é preservada. Além disso, a semântica do protocolo de sinalização da rede cliente é mantida.

4.2.2 O SMP e a Sinalização da Rede de Transporte do Projeto GIGA

Basicamente, o modo de operação do SMP é idêntico ao modo de operação do SOP. Ele está implementado nos elementos de rede localizados na borda interna da rede de transporte e intercepta as

mensagens de sinalização da rede de transporte para poder traduzi-las em invocações à API da UNI-N. Ele também é responsável pela tradução das invocações da UNI-N em mensagens de sinalização do protocolo que existe no plano de controle da rede de transporte.

O protocolo de sinalização que constitui o plano de controle da rede de transporte do Projeto GIGA é o GMPLS RSVP-TE. O trabalho do SMP neste caso é muito similar ao do SOP devido à similaridade dos protocolos de sinalização. A seguir é descrito o funcionamento do SMP no estabelecimento de conexões, remoção de conexões e ocorrência de falhas. No caso do estabelecimento e remoção de conexões é preciso ressaltar que o SMP terá comportamento diferente no ingresso e no egresso da rede de transporte.

- Estabelecimento de Conexões (ingresso da rede de transporte): depois de efetuadas as ações da UNI-N, ela invoca o SMP para que ele possa disparar o estabelecimento da conexão através da rede de transporte. Neste caso, ele apenas mapeia a requisição da UNI-N, *Connection Create Request*, em uma mensagem de PATH do GMPLS RSVP-TE e a entrega ao módulo GMPLS RSVP-TE que existe no nó onde ele está implementado, não cabendo a ele armazenar qualquer mensagem em seu interior. Quando a mensagem de RESV retornar da rede de transporte confirmando o estabelecimento do caminho de luz, ele mapeia a mensagem de RESV em uma invocação à API da UNI-N, *Connection Create Response*, e ao mesmo tempo ele entrega a mensagem de RESV ao módulo GMPLS RSVP-TE implementado localmente.

- Estabelecimento de Conexões (egresso da rede de transporte): Ao interceptar uma mensagem de PATH no egresso da rede de transporte, ele a armazena em seu interior e mapeia esta mensagem em uma invocação à API da UNI-N, *Connection Create Request*. Ao receber a resposta da UNI-N, ele recupera a mensagem de PATH armazenada em seu interior e a entrega ao módulo GMPLS RSVP-TE local para que este último módulo processe a mensagem de PATH e devolva a mensagem de RESV confirmando o estabelecimento da conexão na rede óptica de transporte.

- Remoção de Conexões (ingresso da rede de transporte): Neste caso, o SMP irá ser invocado pela UNI-N, *Connection Delete Request*, e precisará mapear esta mensagem em uma mensagem de PATHTEAR do GMPLS RSVP-TE para que a rede de transporte possa efetuar a remoção do caminho de luz estabelecido.

- Remoção de Conexões (egresso da rede de transporte): Neste outro caso, o SMP irá interceptar a mensagem de PATHTEAR vinda da rede de transporte e traduzi-la em uma invocação à API da UNI-N, *Connection Create Request*. Paralelamente, o SMP irá entregá-la ao módulo GMPLS RSVP-TE local para que a remoção do caminho de luz na rede de transporte seja finalizada.

- Ocorrência de falhas: Neste caso, é possível dizer que o SMP também atua de forma diferente no ingresso e no egresso da rede de transporte. Mas esta é uma função, basicamente, de tradução de mensagens de erro da rede de transporte em invocações à API da UNI-N e vice-versa. Similar ao funcionamento do SOP.

Apenas como esclarecimento, a maneira pela qual o SMP consegue identificar-se como nó de ingresso ou nó de egresso é através dos campos IP de Origem contido no objeto *Sender Template* e IP de Destino contido no objeto *Session* respectivamente, ambos objetos do RSVP.

4.2.3 A UNI-C e a Tabela de Alcançabilidade do Projeto GIGA

Em um primeiro momento, a tabela de alcançabilidade do Projeto GIGA será preenchida manualmente, ou seja, será simplesmente uma tabela que contém o endereço da UNI-C de egresso onde o caminho de luz a ser estabelecido através da rede de transporte terminará. Este endereço será obtido a partir do endereço de destino informado na mensagem de PATH da rede cliente.

A utilização de protocolos para automatizar o preenchimento desta tabela será discutida em uma próxima fase do Projeto GIGA. Somente em caráter especulativo, talvez o BGP venha a ser a solução utilizada.

4.2.4 A UNI-N e o Mecanismo de Roteamento do Projeto GIGA

Terminadas as ações da UNI-C, ela encaminha a solicitação de estabelecimento do caminho óptico à UNI-N. Esta última, por sua vez, é responsável por solicitar ao mecanismo de roteamento da rede de transporte do Projeto GIGA a atribuição de uma rota e um comprimento de onda no qual o novo caminho de luz será estabelecido. No caso do Projeto GIGA, o mecanismo de roteamento utilizado compreende o protocolo de roteamento OSPF-TE, o algoritmo de definição de rotas e o algoritmo de atribuição de comprimento de onda RWA. Maiores detalhes deste mecanismo são encontrados em [Zuliani, 2006].

Apenas para ilustrar, os comutadores ópticos utilizados na rede de transporte do Projeto GIGA não são capazes de efetuar a conversão de comprimento de onda, ou seja, é necessária a utilização do mesmo comprimento de onda em todos os enlaces pertencentes à rota. Este fator aumenta a complexidade do RWA.

4.2.5 Considerações sobre a Arquitetura no Projeto GIGA

Embora o protocolo de sinalização utilizado no plano de controle da rede de transporte do Projeto GIGA (GMPLS RSVP-TE) seja o mesmo protocolo de sinalização utilizado na arquitetura proposta

da UNI, duas sessões são estabelecidas conforme ilustrado na Figura 4.1. Mesmo sendo a rede do Projeto GIGA a motivação inicial para o desenvolvimento da arquitetura proposta, é preciso considerar outros casos onde os protocolos de sinalização não serão os mesmos e, além disso, é um requisito da arquitetura que a mesma seja portável a outros ambientes.

4.3 Resumo

Neste capítulo foi apresentada a principal contribuição deste trabalho, através de um detalhamento, em um primeiro instante generalizado, da arquitetura proposta. A seguir, foi feito um detalhamento voltado à utilização da arquitetura na rede do Projeto GIGA. O próximo capítulo apresenta como foi feita a validação desta arquitetura através de um protótipo.

Capítulo 5

Protótipo Implementado

Este capítulo traz detalhes da implementação do protótipo utilizado na verificação do funcionamento da Arquitetura proposta para a UNI. Basicamente está dividido em duas partes. Na primeira parte há um detalhamento da implementação dos módulos que formam a Arquitetura e os módulos adicionais que precisaram ser desenvolvidos para prover um ambiente eficaz de testes para a Arquitetura. A segunda parte deste capítulo traz os resultados das simulações feitas com o protótipo, divididos em duas partes. A primeira parte traz a comparação entre uma parte do *log* gerado nas simulações e o respectivo diagrama de seqüência, a fim de comprovar o funcionamento da Arquitetura. A segunda parte apresenta testes de desempenho da Arquitetura, onde é possível verificar os tempos médios de execução do protótipo durante as fases de estabelecimento e remoção de conexões. A partir destes tempos coletados, extraímos os tempos de funcionamento da UNI proposta com o intuito de que estes tempos venham a ser utilizados como referência a uma implementação real da UNI.

5.1 Descrição do Protótipo

Após o desenvolvimento da arquitetura proposta para a UNI e sua adequação às necessidades do Projeto GIGA, um protótipo foi implementado para que se pudesse avaliar seu funcionamento e, caso necessário, efetuar alterações na arquitetura com base nos resultados das simulações. A função principal do protótipo é verificar o funcionamento da arquitetura em casos onde a rede cliente solicita o estabelecimento e a remoção de caminhos ópticos através da rede de transporte e qual o seu comportamento quando erros ocorrem.

Inicialmente, foram implementados os módulos que constituem a arquitetura proposta (SOP, SMP, UNI-C E UNI-N) respeitando as decisões de implementação do Projeto GIGA. A seguir são apresentados alguns detalhes do desenvolvimento destes módulos:

- SOP: as redes clientes do GIGA utilizam equipamentos que implementam o protocolo de sinalização MPLS RSVP-TE, ou seja, o SOP foi implementado para interagir com este protocolo e tornar a interação com a UNI transparente às redes clientes.
- SMP: o plano de controle da rede óptica de transporte utiliza o protocolo de sinalização GMPLS RSVP-TE. Conseqüentemente, o SMP foi implementado para interagir com o GMPLS RSVP-TE e, também, manter transparente a interação entre a rede de transporte e a UNI.
- UNI-C e UNI-N: o Projeto GIGA define a utilização do protocolo de sinalização GMPLS RSVP-TE na interação entre a UNI-C e a UNI-N. Logo, as mensagens destes módulos foram implementadas utilizando o formato (PDU) do GMPLS RSVP-TE.

Em uma segunda fase, foram implementados módulos adicionais para que fosse possível estabelecer um ambiente de testes onde as ações da arquitetura fossem verificadas. Estes módulos simulam a sinalização das redes clientes (MPLS RSVP-TE), a sinalização da rede óptica de transporte (GMPLS RSVP-TE), os caminhos ópticos estabelecidos (Emulador de Interfaces) e os protocolos de roteamento que existem na rede cliente e na rede de transporte (Servidor de Rotas). Todos os módulos do protótipo foram implementados em JAVA e o Apêndice B traz os respectivos diagramas de classes. A seguir são apresentadas algumas características do protótipo que serão detalhadas nesta seção.

1. Características gerais de protótipo: a utilização da arquitetura está associada à existência de uma rede óptica de transporte e, pelo menos, duas redes clientes. Sendo uma rede cliente de origem e uma de destino que são separadas pela rede de transporte. Entretanto, é desejável que o protótipo suporte diferentes topologias, ou seja, topologias que possuam um número variado de redes clientes ou que utilizem um número diferente de nós no interior da rede óptica de transporte. Conseqüentemente, um ambiente de testes adequado envolve a utilização de diversas instâncias dos módulos do protótipo para simular o estabelecimento e a remoção de conexões. Logo, é recomendável que o ambiente de simulação possa ser distribuído entre vários computadores e, também, que ele seja capaz de tratar requisições em paralelo. Para atender a estas características, todos os módulos do protótipo são servidores *Socket MultiThread*.
2. Fidelidade às especificações existentes: não havia nenhuma implementação dos protocolos de sinalização utilizados na arquitetura (GMPLS RSVP-TE e MPLS RSVP-TE) disponível para utilização no protótipo. Devido a este fato, estes dois protocolos tiveram de ser implementados e suas implementações respeitaram as especificações feitas pelo IETF, embora não implementem todos os objetos definidos. Entretanto, a não implementação de todos os objetos definidos não prejudica o seu funcionamento pois apenas os objetos opcionais deixaram de ser implementados. Uma outra importante decisão de implementação dos protocolos diz

respeito à utilização do XML nas mensagens de sinalização devido à facilidade de utilização fornecida pela linguagem XML e pelo ferramental associado a ela.

3. Facilitar a execução das simulações e a coleta de resultados: é necessário o desenvolvimento de uma ferramenta para controlar ações do protótipo como a inicialização ou a finalização dos módulos do protótipo (distribuídos em diversos computadores) automaticamente, disparar o estabelecimento ou a remoção de conexões e unificar o *log* gerado para facilitar a análise do funcionamento da arquitetura.

Embora seja possível definir qualquer topologia com o protótipo, uma topologia foi estabelecida para padronizar os testes da arquitetura. Esta topologia é apresentada na Figura 5.1 e é suficiente para efetuar os testes de funcionamento necessários à arquitetura. Foram utilizados 4 computadores Intel(R) Pentium(R) IV HT 3.0 GHz com 1 GB de memória Ram equipados com 4 interfaces de rede *Fast Ethernet* e cada um destes computadores é responsável pela execução de diversos módulos do protótipo. A distribuição dos módulos envolvidos nas simulações em cada um destes computadores é detalhada na Figura 5.2.

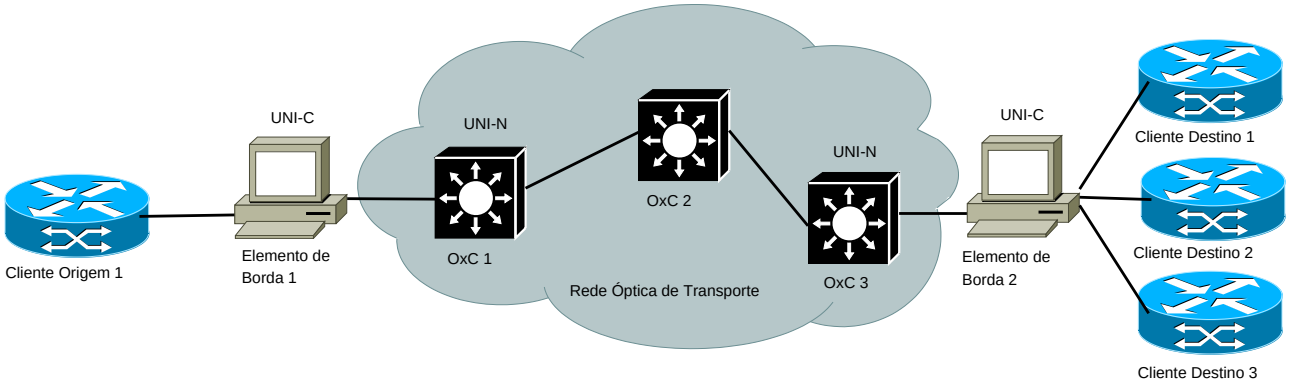

Fig. 5.1: Topologia utilizada nas simulações.

Conforme mencionado acima, todos os módulos do protótipo são servidores *Socket* e na Figura 5.2 é possível observar o número de porta *Socket* utilizado pelos módulos nas simulações. Esta característica do *Socket* permite a simulação de ambientes constituídos por grandes topologias em um número reduzido de computadores através da alteração do número das portas *Socket* utilizadas por instâncias diferentes dos mesmos módulos do protótipo em um mesmo computador. Por exemplo, na Figura 5.2 a máquina Ducati executa três instâncias do protocolo de sinalização MPLS RSVP-TE. Do ponto de vista das simulações, a máquina Ducati está simulando três nós das redes clientes ligadas à rede de transporte.

Para atender ao último item mencionado, ou seja, facilitar as simulações e a coleta e análise dos resultados do protótipo, foi utilizado *Swing* na construção de uma interface gráfica centralizada

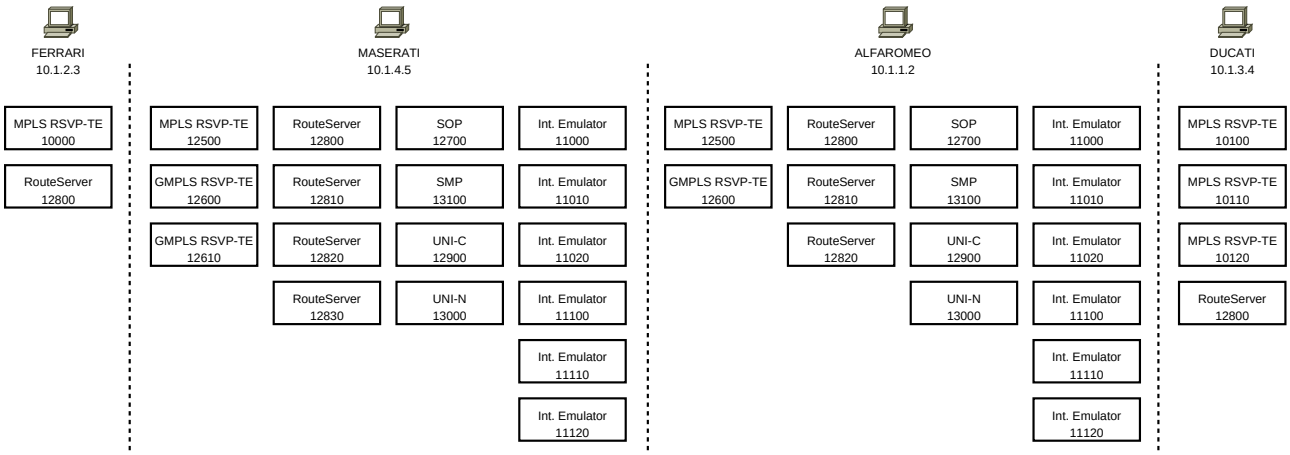

Fig. 5.2: Distribuição dos módulos do protótipo.

de onde é possível disparar diversas ações no protótipo e, também, verificar os eventos que estão ocorrendo nas simulações a partir de um *log* centralizado. A utilização do *log* gerado na verificação do funcionamento da arquitetura é detalhado na Seção 5.2.1. A Figura 5.3 apresenta a interface gráfica desenvolvida.

O botão *Start* presente na interface inicia todos os módulos da arquitetura nos diversos computadores necessários à simulação e o botão *Stop* interrompe o funcionamento destes módulos liberando os recursos que estavam sendo consumidos em todos os computadores. Esta é uma funcionalidade muito interessante do protótipo pois são utilizados *scripts* para instanciar ou finalizar os módulos nos computadores automaticamente. A Figura 5.4 ilustra o *script* utilizado para instanciar os módulos utilizados pela topologia descrita anteriormente.

Observando a Figura 5.4 é possível verificar que são utilizados diversos comandos *ssh* para os 4 computadores utilizados nas simulações, cada um deles instanciando um módulo utilizado pelo protótipo. Estes comandos *ssh* executam as classes JAVA passando os parâmetros necessários a cada módulo automatizando a tarefa de instanciar o protótipo para executar as simulações. Dentre os parâmetros utilizados pelos módulos temos: o endereço IP e o número de porta *Socket* associados aos módulos e, também, um arquivo XML. Estes arquivos XML possuem as características individuais de configuração de cada módulo. Por exemplo, a Figura 5.5 apresenta o XML do SOP que contém:

- o nome do computador onde ele está sendo executado (*Host Name*);
- o endereço IP e a porta *Socket* do console central onde as mensagens de *log* devem ser enviadas;
- o endereço IP e a porta *Socket* do servidor de rotas associado a ele;
- o endereço IP e a porta *Socket* da UNI-C à qual ele está associado;

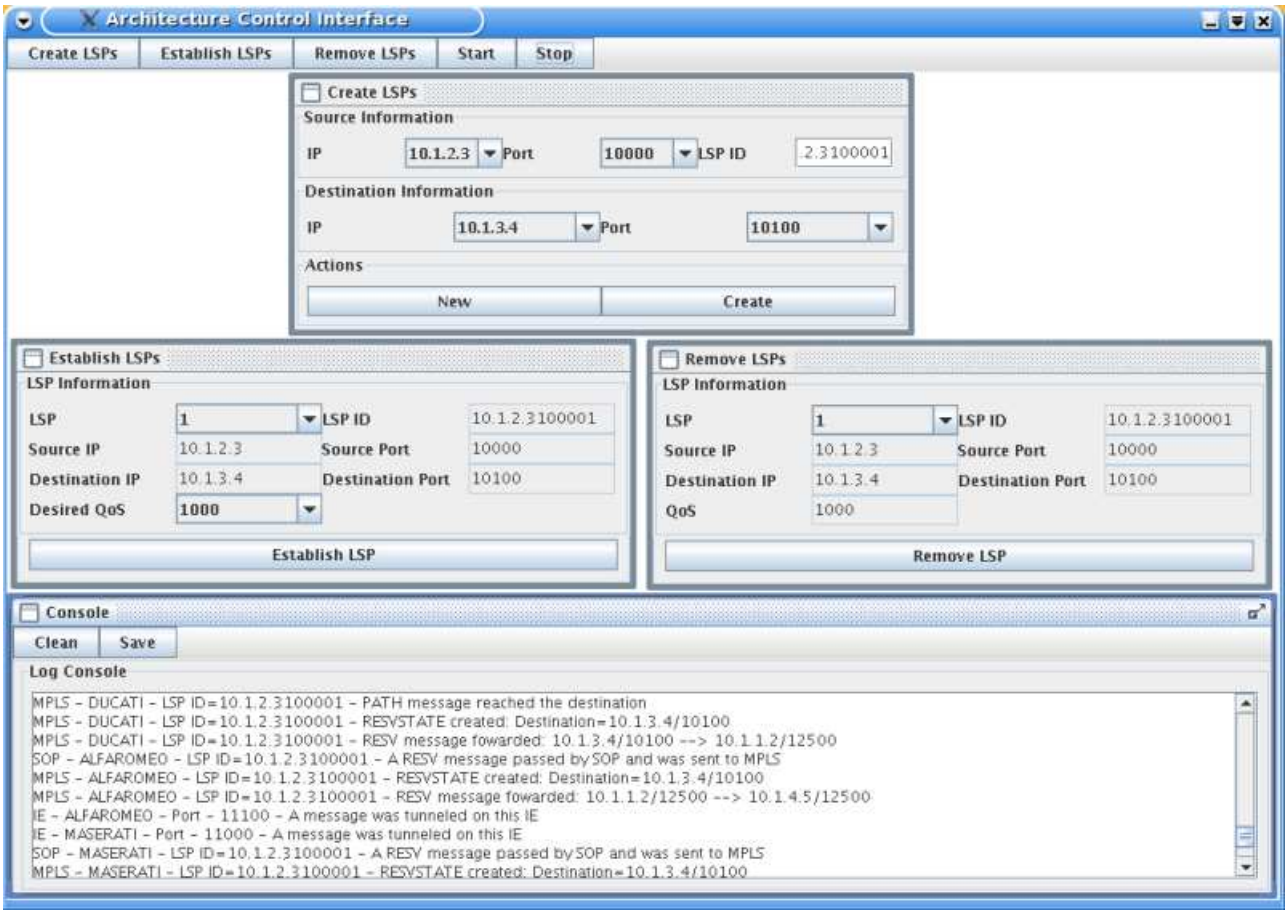

Fig. 5.3: Interface Gráfica do Protótipo.

- o endereço IP e a porta *Socket* do módulo MPLS RSVP-TE do qual o SOP intercepta as mensagens de sinalização para automatizar a interação com a UNI.

Os outros módulos possuem parâmetros diferentes. O SMP, por exemplo, possui o endereço IP e a porta *Socket* da UNI-N, o MPLS RSVP-TE possui os parâmetros de QoS disponível nos nós que constituem as redes clientes, o servidor de rotas possui informações do roteamento, tais como, o endereço IP e a porta *Socket* do próximo nó ao qual os módulos devem encaminhar suas mensagens de sinalização. A utilização de arquivos XML para efetuar este tipo de configuração facilita o desenvolvimento de diferentes topologias pois os módulos são facilmente configurados.

Através desta interface centralizada é possível ainda disparar o estabelecimento de novos LSPs no protótipo (*Establish LSPs*). Ao selecionar o caminho óptico que se deseja estabelecer, o console gera uma mensagem de PATH MPLS e a envia ao MPLS RSVP-TE de origem da rede cliente. É possível ainda remover os LSPs que foram estabelecidos a partir desta interface (*Remove LSPs*). Basta selecionar qual o caminho que se deseja remover e o console gera uma mensagem de PATHTEAR

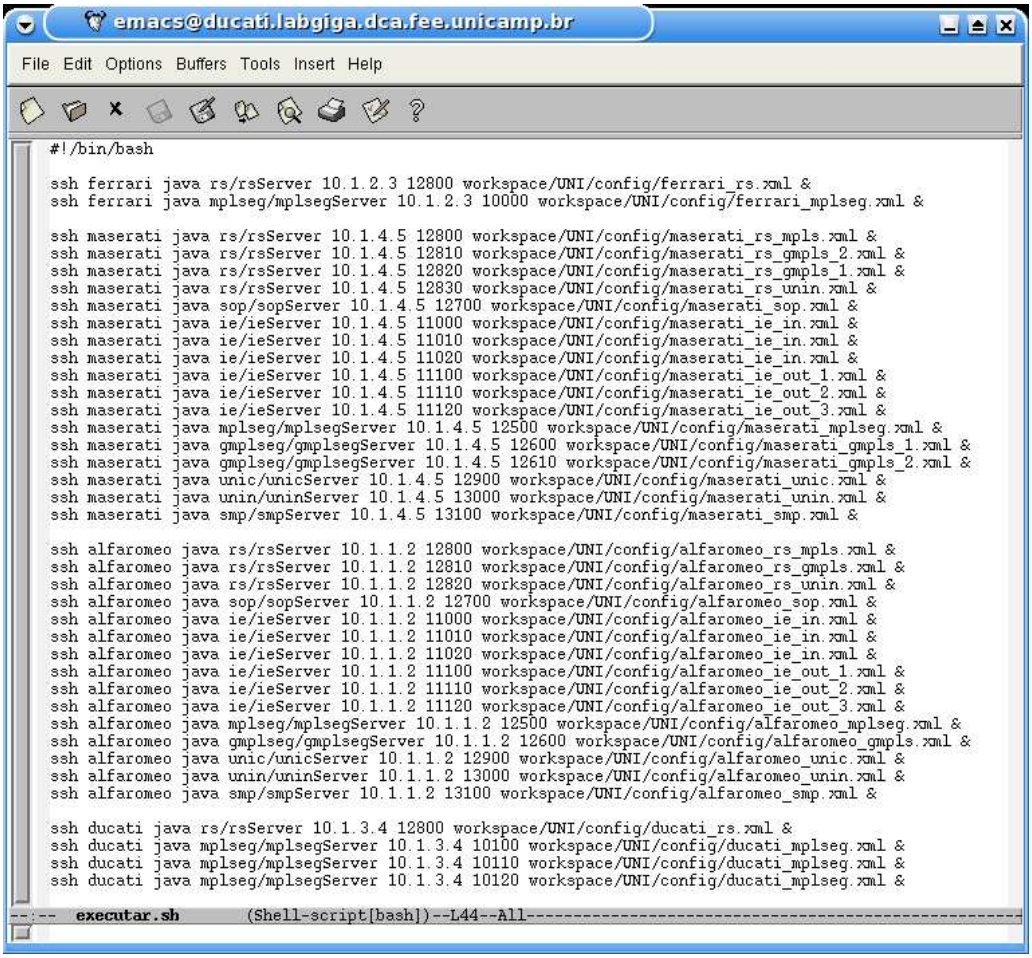

Fig. 5.4: Script utilizado para inicializar todos os módulos do protótipo.

MPLS e a envia ao nó de origem da rede cliente. Todo o processo de estabelecimento ou remoção é acompanhado a partir do *log*, pois todos os módulos envolvidos na simulação enviam mensagens para este console centralizado.

A arquitetura especifica a utilização de um protocolo de roteamento para atribuir rotas no interior da rede óptica de transporte. No desenvolvimento do protótipo implementou-se um Servidor de Rotas que é invocado pela UNI-N. Este módulo segue o padrão descrito acima, ou seja, as informações de roteamento são carregadas através de um arquivo XML que é passado como parâmetro. Embora pareça simples, a estrutura desenvolvida para o arquivo XML do servidor de rotas possibilita um certo grau de engenharia de tráfego, atribuindo rotas distintas aos LSPs através da associação do identificador do LSP às informação de roteamento. Por exemplo, um mesmo destino (DUCATI - MPLS RSVP-TE 10100) pode ser atingido através de caminhos ópticos diferentes na rede de transporte pois podem existir entradas diferentes na tabela de roteamento associadas ao identificador (único) de cada LSP.

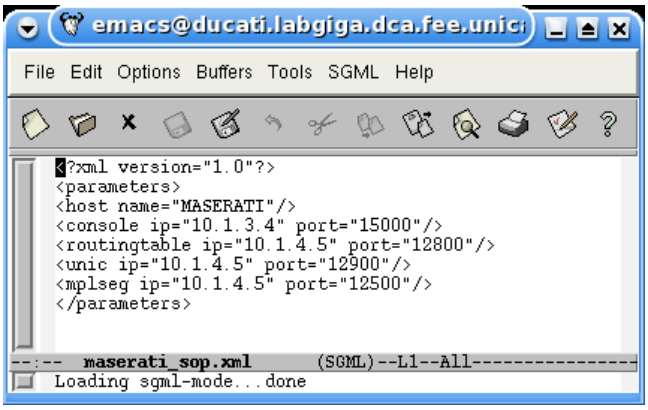

Fig. 5.5: Exemplo de XML utilizado para instanciar os módulos do protótipo.

No cenário real do Projeto GIGA, conforme definido na especificação da arquitetura, a sinalização da rede cliente é tunelada no caminho óptico estabelecido através da rede óptica de transporte. Para contornar a limitação da não disponibilidade de uma rede óptica, um Emulador de Interfaces Ópticas foi implementado para simular o caminho óptico (enlace TE) entre os elementos de borda da rede cliente. Os emuladores de interface são divididos em dois grupos, entrada e saída (*in* e *out*). Na Figura 5.2 é possível verificar a existência de seis emuladores de interface em Maserati e seis emuladores em Alfaromeo. A associação destes emuladores produz seis caminhos ópticos unidirecionais através da rede de transporte, sendo três deles em cada sentido (3 *upstream* e 3 *downstream*).

Um caminho de luz real efetua comutação no nível da camada física (camada 1), ou seja, não é feita nenhuma verificação nos pacotes de dados transmitidos por ele. Conseqüentemente, o resultado da utilização do Emulador de Interfaces é o mesmo. Por exemplo, o emulador de interface 11000 (*in*) da máquina Maserati está associado ao emulador de interfaces 11100 (*out*) da máquina Alfaromeo. Esta associação é feita através do XML passado como parâmetro no momento de instanciação destes módulos. No cenário real, ao finalizar o estabelecimento do caminho óptico através da rede de transporte, os nós de borda da rede cliente tornam-se vizinhos pois o protocolo de roteamento das redes clientes passa a trocar informações de roteamento e o resultado é o tunelamento da mensagem de sinalização no caminho óptico estabelecido (enlace TE). No protótipo, o SOP adiciona uma entrada na tabela de roteamento do servidor de rotas da rede cliente simulando a função do protocolo de roteamento da rede real. Com isto, o MPLS RSVP-TE vai enviar suas mensagens para o emulador de interfaces (túnel). O emulador de interfaces de entrada, por sua vez, ao receber uma mensagem de sinalização, apenas a encaminha ao emulador de interfaces de saída associado a ele sem fazer qualquer tipo de verificação no corpo da mensagem.

Com esta infra-estrutura desenvolvida é possível simular o estabelecimento e a remoção de conexões na rede do Projeto GIGA e verificar o funcionamento da arquitetura em cenários que

ocorram erros tais como:

- a qualidade de serviço solicitada pela rede cliente não está disponível na rede óptica de transporte;
- a rede cliente não está autorizada a solicitar serviços à rede óptica de transporte;
- o destino não é alcançável pela rede óptica de transporte, entre outros erros.

No caso onde a qualidade de serviço (largura de banda) solicitada pela rede cliente não pode ser atendida pela rede óptica de transporte, os módulos MPLS RSVP-TE são capazes de fazer uma contra proposta baseada em regras pré-definidas nos arquivos XML de configuração, a fim de tentar estabelecer novamente o LSP através da rede óptica de transporte e dar uma certa autonomia ao protótipo.

5.2 Resultados das Simulações

Esta seção traz os resultados obtidos nas simulações feitas com o protótipo descrito anteriormente divididos em duas partes. A primeira parte dos resultados trata da verificação do funcionamento da arquitetura através de uma comparação do *log* gerado nas simulações com os respectivos diagramas de seqüência. A segunda parte dos resultados apresenta os tempos de funcionamento da arquitetura que foram coletados com a intenção de oferecer uma visão do desempenho da arquitetura no caso do protótipo desenvolvido. Estes tempos são vistos, unicamente, como parâmetros de referência para uma eventual implementação da UNI proposta neste trabalho.

5.2.1 Verificação do Funcionamento da Arquitetura Proposta

A motivação principal para o desenvolvimento do protótipo foi a necessidade de poder expor a arquitetura proposta a um ambiente de testes no qual várias características pudessem ser testadas a fim de se encontrar possíveis falhas na sua especificação e, conseqüentemente, corrigi-las antes da efetiva implementação da arquitetura na rede do Projeto GIGA.

A análise de funcionamento da arquitetura é feita através dos *logs* gerados durante as simulações. Este *log* apresenta a série de eventos que ocorreram em todas as fases da simulação, ou seja, é possível fazer uma verificação desde a fase de instanciação dos módulos do protótipo até as ações desempenhadas por eles durante as fases de estabelecimento, remoção e tratamento de falhas dos LSPs. A Figura 5.6 exemplifica o *log* gerado durante a fase de instanciação dos módulos. As mensagens geradas possuem uma padronização quanto aos detalhes reportados por elas, dentre as informações estão:

- o módulo que executou a tarefa;
- em qual computador o módulo está sendo executado;
- a qual LSP pertence a mensagem (LSP ID);
- qual o procedimento executado pelo módulo;
- qual a decisão tomada pelo módulo.

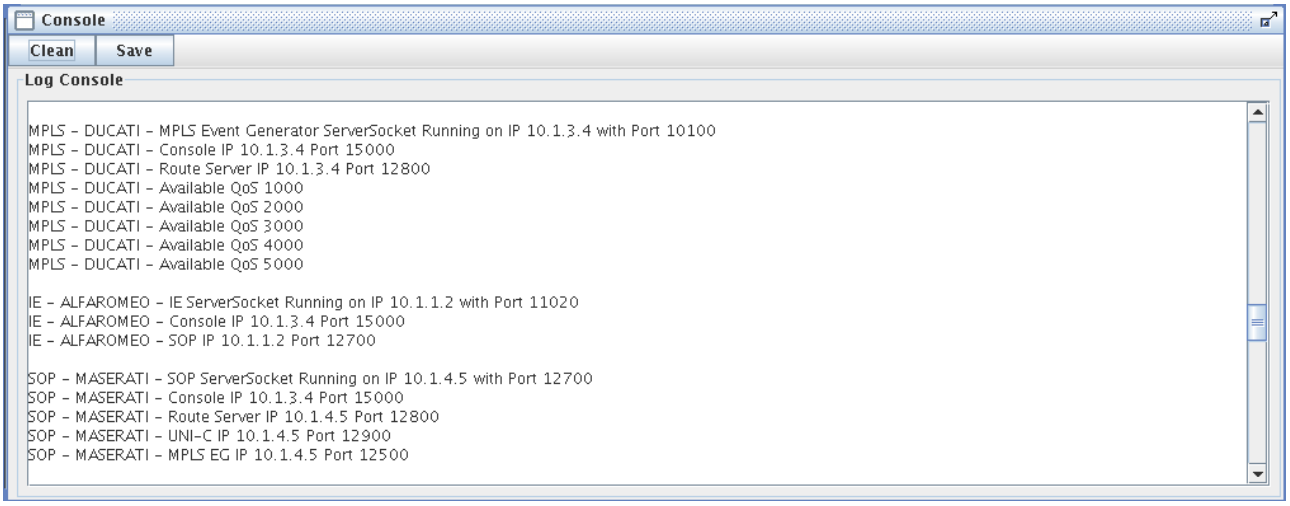

Fig. 5.6: Trecho do *log* gerado durante a instanciação dos módulos do protótipo.

A metodologia empregada na verificação do funcionamento da arquitetura é a comparação entre os *logs* gerados durante as fases das simulações e os respectivos diagramas de seqüência especificados para a arquitetura. Para exemplificar o processo de validação da arquitetura foram selecionadas duas importantes funcionalidades da arquitetura, ou seja, o estabelecimento e a remoção de conexões. Entretanto, ambas as funcionalidades possuem um elevado número de ações e utilizam muitos módulos no desempenho destas ações. Conseqüentemente, para que a documentação destas funcionalidades através de diagramas de seqüência fosse clara, foi preciso dividir cada uma delas em três diagramas de seqüência que são apresentados no Apêndice A.

As Figuras 5.7 e 5.8 ilustram os trechos dos *logs* gerados durante as fases de estabelecimento e remoção de conexões que correspondem aos diagramas de seqüência ilustrados nas Figuras 5.9 e 5.10. Tanto as figuras onde os *logs* são apresentados quanto as figuras que trazem os respectivos diagramas de seqüência foram numeradas para facilitar o acompanhamento.

No primeiro exemplo, através do acompanhamento dos trechos de *log* apresentados na Figura 5.7 e do diagrama de seqüência apresentado na Figura 5.9, é possível verificar os 27 passos que foram

executados pela arquitetura durante o estabelecimento de uma conexão. A topologia utilizada nos exemplos é a apresentada na Figura 5.1 e ela está organizada conforme a distribuição dos módulos apresentada na Figura 5.2. Os passos correspondem às seguintes ações:

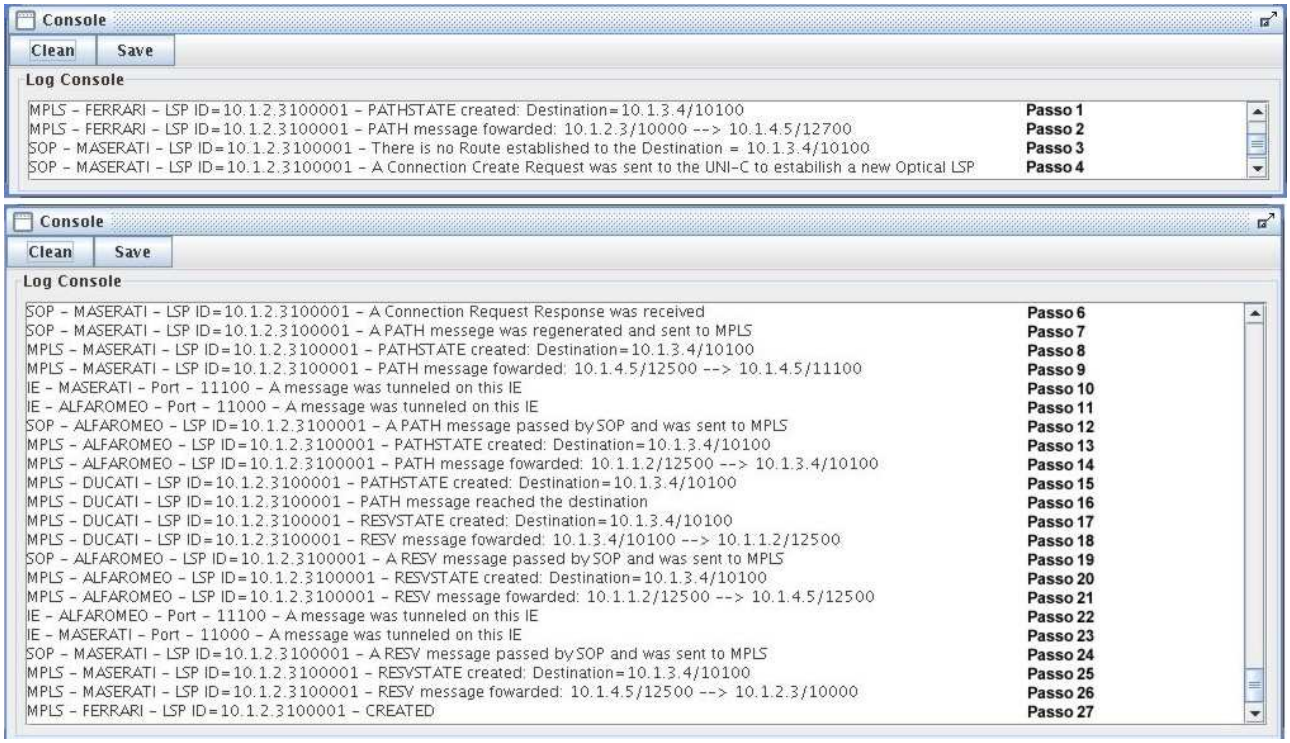

Fig. 5.7: Trechos do *log* gerado durante o estabelecimento de um LSP no protótipo.

- os passos 1 e 2 correspondem ao processamento e ao encaminhamento da mensagem de PATH MPLS do LSP com o identificador 10.1.2.3100001 pelo módulo MPLS do nó de origem da rede cliente localizado na máquina Ferrari;
- os passos 3 e 4 correspondem ao instante em que o módulo SOP (localizado na máquina Maserati) consulta o roteamento e descobre que não há nenhum caminho óptico estabelecido que atenda ao destino informado na mensagem de PATH MPLS (10.1.3.4/10100); armazena a mensagem em seu interior e invoca a UNI-C para que um novo caminho óptico seja estabelecido através da rede óptica de transporte;
- o passo 5 não está presente na Figura 5.7 pois corresponde a todas as ações que são desempenhadas no interior da rede óptica de transporte definidas nos diagramas de seqüência A.2 e A.3 do Apêndice A;
- os passos 6 e 7 correspondem ao instante em que a resposta (*Connection Request Response*) vinda da UNI-C atinge o módulo SOP e ele, por sua vez, recupera a mensagem de PATH

MPLS que ele havia armazenado em seu interior e a entrega ao módulo MPLS localizado no nó de borda da rede óptica de transporte (máquina Maserati);

- os passos 8 e 9 correspondem ao processamento e ao encaminhamento da mensagem de PATH MPLS do LSP com o identificador 10.1.2.3100001 pelo módulo MPLS do nó localizado na borda da rede óptica de transporte (máquina Maserati);

- os passos 10 e 11 correspondem aos instantes em que a mensagem de PATH MPLS é tunelamento no caminho óptico (módulos emuladores de interface localizados em Maserati e Alfaromeo) e recuperada no seu egresso;

- o passo 12 refere-se ao recebimento da mensagem de PATH MPLS (LSP 10.1.2.3100001) pelo módulo SOP localizado do outro lado da rede óptica de transporte (máquina Alfaromeo) e ao encaminhamento dela ao módulo MPLS;

- os passos 13 e 14 correspondem ao processamento e ao encaminhamento da mensagem de PATH MPLS do LSP com o identificador 10.1.2.3100001 pelo módulo MPLS do nó localizado na borda de egresso da rede óptica de transporte (máquina Alfaromeo);

- os passos 15, 16, 17 e 18 correspondem ao processamento da mensagem de PATH MPLS do LSP com o identificador 10.1.2.3100001 pelo módulo MPLS do nó de destino da rede cliente (máquina Ducati) e a respectiva criação e encaminhamento da mensagem de RESV MPLS;

- o passo 19 refere-se ao recebimento da mensagem de RESV MPLS (LSP 10.1.2.3100001) pelo módulo SOP localizado na borda da rede óptica de transporte (máquina Alfaromeo) e ao encaminhamento dela ao módulo MPLS;

- os passos 20 e 21 correspondem ao processamento e ao encaminhamento da mensagem de RESV MPLS do LSP com o identificador 10.1.2.3100001 pelo módulo MPLS do nó localizado na borda da rede óptica de transporte (máquina Alfaromeo);

- os passos 22 e 23 correspondem aos instantes em que a mensagem de RESV MPLS é tunelamento no caminho óptico inverso (módulos emuladores de interface localizados em Alfaromeo e Maserati) e recuperada no seu egresso;

- o passo 24 refere-se ao recebimento da mensagem de RESV MPLS (LSP 10.1.2.3100001) pelo módulo SOP localizado na borda da rede óptica de transporte (máquina Maserati) e ao encaminhamento dela ao módulo MPLS;

- os passos 25 e 26 correspondem ao processamento e ao encaminhamento da mensagem de RESV MPLS do LSP com o identificador 10.1.2.3100001 pelo módulo MPLS do nó localizado na borda da rede óptica de transporte (máquina Maserati);

- o passo 27 refere-se ao instante em que a mensagem de RESV MPLS atinge o nó de origem da rede cliente e o LSP com o identificador 10.1.2.3100001 é totalmente estabelecido.

No segundo exemplo, através do acompanhamento dos trechos de *log* apresentados na Figura 5.8 e do diagrama de seqüência apresentado na Figura 5.10 é possível verificar os 17 passos que foram executados pela arquitetura durante a remoção de uma conexão. Os passos correspondem às seguintes ações:

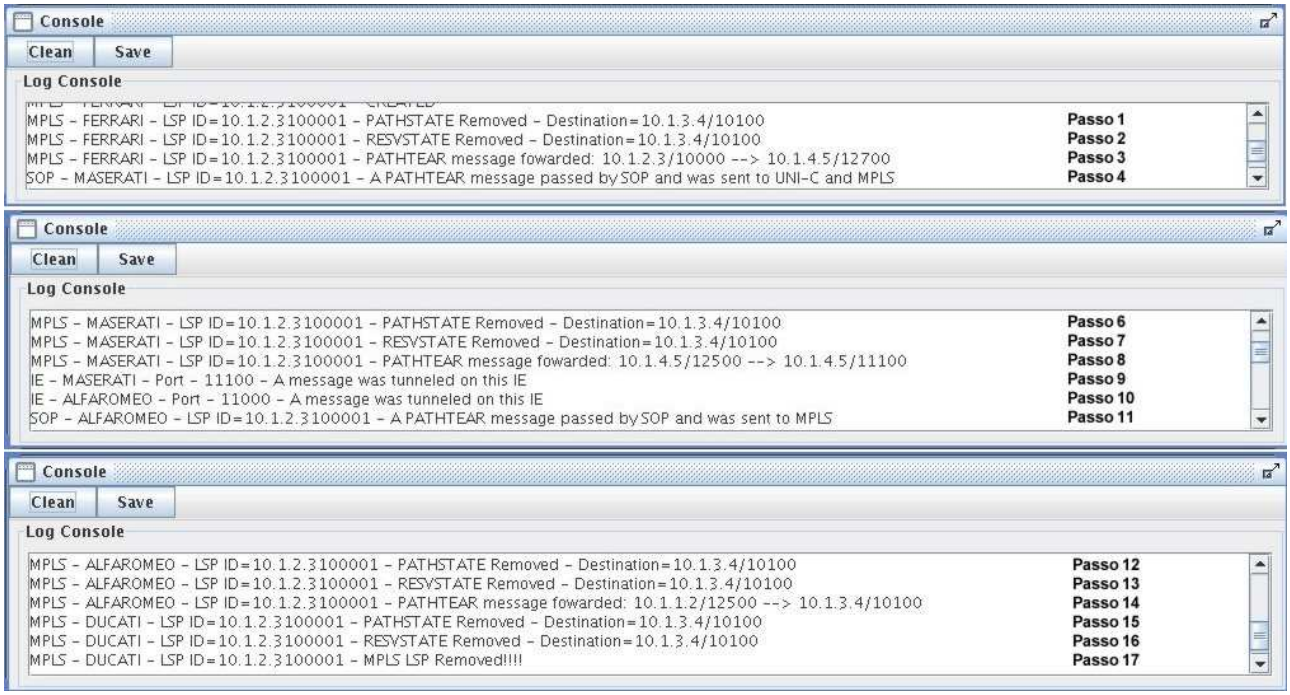

Fig. 5.8: Trechos do *log* gerado durante a remoção de um LSP no protótipo.

- os passos 1, 2 e 3 correspondem ao processamento e ao encaminhamento da mensagem de PATHTEAR MPLS do LSP com o identificador 10.1.2.3100001 pelo módulo MPLS do nó de origem da rede cliente localizado na máquina Ferrari. Note que a mensagem de PATHTEAR remove ambos os estados de PATH e RESV;

- o passo 4 refere-se ao instante em que o módulo SOP recebe a mensagem de PATHTEAR MPLS e a encaminha paralelamente ao módulo MPLS do nó localizado na borda da rede óptica de transporte e à UNI-C para que seja removida a conexão MPLS da rede cliente e, também, o caminho óptico que foi estabelecido na rede óptica de transporte;

- o passo 5 não está presente na Figura 5.8 pois corresponde a todas as ações que são desempenhadas no interior da rede óptica de transporte definidas nos diagramas de seqüência A.5 e A.6 do Apêndice A;

- os passos 6, 7 e 8 correspondem ao processamento (remover estados de PATH e RESV) e ao encaminhamento da mensagem de PATHTEAR MPLS do LSP com o identificador 10.1.2.3100001 pelo módulo MPLS do nó localizado na borda da rede óptica de transporte (máquina Maserati);

- os passos 9 e 10 correspondem aos instantes em que a mensagem de PATHTEAR MPLS é tunelamento no caminho óptico (módulos emuladores de interface localizados em Maserati e Alfaromeo) e recuperada no seu egresso;

- o passo 11 refere-se ao recebimento da mensagem de PATHTEAR MPLS (LSP 10.1.2.3100001) pelo módulo SOP localizado do outro lado da rede óptica de transporte (máquina Alfaromeo) e ao encaminhamento dela ao módulo MPLS;

- os passos 12, 13 e 14 correspondem ao processamento (remover estados de PATH e RESV) e ao encaminhamento da mensagem de PATHTEAR MPLS do LSP com o identificador 10.1.2.3100001 pelo módulo MPLS do nó localizado na borda de egresso da rede óptica de transporte (máquina Alfaromeo);

- os passos 15 e 16 correspondem ao processamento (remover estados de PATH e RESV) da mensagem de PATHTEAR MPLS do LSP com o identificador 10.1.2.3100001 pelo módulo MPLS do nó de destino da rede cliente (máquina Ducati);

- o passo 17 refere-se ao instante em que a conexão MPLS da rede cliente é totalmente removida.

É importante destacar que o protótipo desenvolvido é uma ferramenta transitória entre as fases de especificação da arquitetura da UNI e a sua efetiva implementação no Projeto GIGA. Sua utilização não foi apenas importante na verificação do funcionamento da arquitetura mas também na detecção e correção de pontos de inconsistência. A seção a seguir apresenta os tempos obtidos nas simulações.

5.2.2 Análise de Desempenho da Arquitetura Proposta

Esta seção apresenta os tempos de funcionamento da arquitetura que foram coletados nas simulações feitas com o protótipo desenvolvido. A topologia utilizada é a mesma que foi apresentada na Figura 5.1 e detalhada na Figura 5.2. Por se tratar de um ambiente simulado, implementado em

JAVA e utilizando XML na composição das mensagens de sinalização, os tempos apresentados nesta seção são vistos apenas como valores de referência para uma implementação real da UNI proposta.

O principal impacto da implementação em XML das mensagens dos protocolos é que todos os processos da simulação envolvem a análise das mensagens XML pelo *parser*. O tempo necessário para a execução do *parser* está diretamente relacionado ao tamanho das mensagens. As mensagens XML utilizadas no protótipo são em geral grandes e este é o preço a ser pago pela flexibilidade introduzida pelo XML. A Tabela 5.1 apresenta o tamanho médio de algumas das mensagens XML utilizadas nas simulações.

Tab. 5.1: Tamanho médio de algumas mensagens XML utilizadas no protótipo.

Mensagem	**Tamanho (em bytes)**
Mensagem de PATH MPLS	315
Mensagem de PATH GMPLS	750
Mensagem de RESV MPLS	300
Mensagem de RESV GMPLS	430
Mensagem de PATHTEAR MPLS	280
Mensagem de PATHTEAR GMPLS	285

Durante as simulações foram coletados os tempos de funcionamento do protótipo nas fases de estabelecimento e remoção de conexões e estes tempos são apresentados nas Tabelas 5.2 e 5.3.

Tab. 5.2: Tempos médios obtidos no estabelecimento de LSPs no protótipo.

Ação	**Tempo (em ms)**
Estabelecimento fim-a-fim	550
SOP - SMP (PATH)	60
SMP - UNI-C (PATH)	30
UNI-C - SMP (RESV)	35
SMP - SOP (RESV)	50

A Tabela 5.2 apresenta o tempo necessário para o estabelecimento fim-a-fim. Os tempos da UNI que estão contidos no tempo total são referentes à troca das mensagens de PATH e RESV (processo de sinalização em duas vias). São necessários em média 550 ms para estabelecer uma conexão no protótipo. Destes 550 ms totais, temos que:

- 60 ms são consumidos no ingresso da rede de transporte durante o tratamento da mensagem de PATH entre os módulos SOP, UNI-C, UNI-N e SMP;
- 30 ms no egresso da rede de transporte para tratar a mensagem de PATH entre os módulos SMP, UNI-N e UNI-C;
- 35 ms no egresso da rede devido ao tratamento da mensagem de RESV entre os módulos UNI-C, UNI-N e SMP;

- 50 ms no ingresso da rede para tratar a mensagem de RESV entre os módulos SMP, UNI-N, UNI-C e SOP;

- o restante do tempo (375 ms) é referente ao processo de sinalização que ocorre na rede cliente e no plano de controle da rede de tranporte.

Tab. 5.3: Tempos médios obtidos na remoção de LSPs no protótipo.

Ação	**Tempo (em ms)**
Remoção fim-a-fim	355
SOP - SMP (PATH)	60
SMP - UNI-C (PATH)	15
UNI-C - SMP (RESV)	20
SMP - UNI-C (RESV)	15
UNI-C - SMP (PATHTEAR)	60
SMP - UNI-C (PATHTEAR)	35

A Tabela 5.3 apresenta o tempo necessário para a remoção fim-a-fim de uma conexão utilizando o protótipo juntamente com os tempos do processo de remoção em três vias da UNI. Na média são gastos 355 ms para remover uma conexão, dos quais:

- 60 ms são gastos no processamento da mensagem de PATH no ingresso da rede de transporte pelos módulos SOP, UNI-C, UNI-N e SMP;

- 15 ms no processamento da mensagem de PATH no egresso da rede pelos módulos SMP, UNI-N e UNI-C;

- 20 ms no processamento da mensagem de RESV no egresso da rede pelos módulos UNI-C, UNI-N e SMP;

- 15 ms para processar a mensagem de RESV no ingresso da rede de tranporte entre os módulos SMP, UNI-N e UNI-C;

- 60 ms na efetivação da remoção através da mensagem de PATHTEAR entre os módulos UNI-C, UNI-N e SMP;

- 35 ms no egresso da rede para terminar de remover a conexão óptica entre os módulos SMP, UNI-N e UNI-C;

- 150 ms devido ao processamento da sinalização das redes cliente e de transporte.

A partir das duas tabelas de tempos médios é possível extrair os tempos da arquitetura proposta para a UNI que foi implementada no protótipo. Somando-se os tempos de execução da UNI temos 175 ms para estabelecer um LSP (60 ms + 30 ms + 35 ms + 50 ms) e 205 ms para remover um LSP (60 ms + 15 ms + 20 ms + 15 ms + 60 ms + 35 ms). Estes são os tempos que devem ser considerados como resultados da avaliação de despenho da arquitetura proposta.

Para melhor exemplificar a quantidade de tempo que seria necessário para estabelecer um LSP em uma rede óptica real utilizando a arquitetura proposta da UNI, como ela está implementada no protótipo, a referência [Margaria et al., 2005] traz alguns testes de desempenho feitos em uma rede óptica GMPLS constituída por nós ópticos reais e que implementa um protótipo de um plano de controle GMPLS. Foram realizados alguns testes com topologias diferentes, mas o tempo relevante para esta comparação é o tempo obtido no estabelecimento de um LSP constituído de três nós ópticos, ou seja, 561 ms. Como conclusão, utilizando a arquitetura proposta da UNI para solicitar o estabelecimento de um LSP através da rede óptica de transporte apresentada em [Margaria et al., 2005] seriam necessários 736 ms, dos quais 175 ms referem-se ao tempo da UNI para estabelecer uma conexão e 561 referem-se a todo o processo de sinalização que ocorre na rede óptica de transporte da referência em questão.

Por último, é importante ressaltar mais uma vez que os tempos apresentados servem apenas como uma referência do provisionamento de conexões ópticas fim-a-fim em um ambiente que implementa o Modelo Overlay com uma UNI baseada na arquitetura proposta neste trabalho.

5.3 Resumo

Este capítulo apresentou detalhes da implementação do protótipo utilizado na verificação de consistência da arquitetura proposta para a UNI neste trabalho e, também, os resultados obtidos nas simulações que comprovam o correto funcionamento da arquitetura e servem como referência de tempo para uma possível implementação real. O capítulo a seguir traz as conclusões deste trabalho e alguns trabalhos futuros.

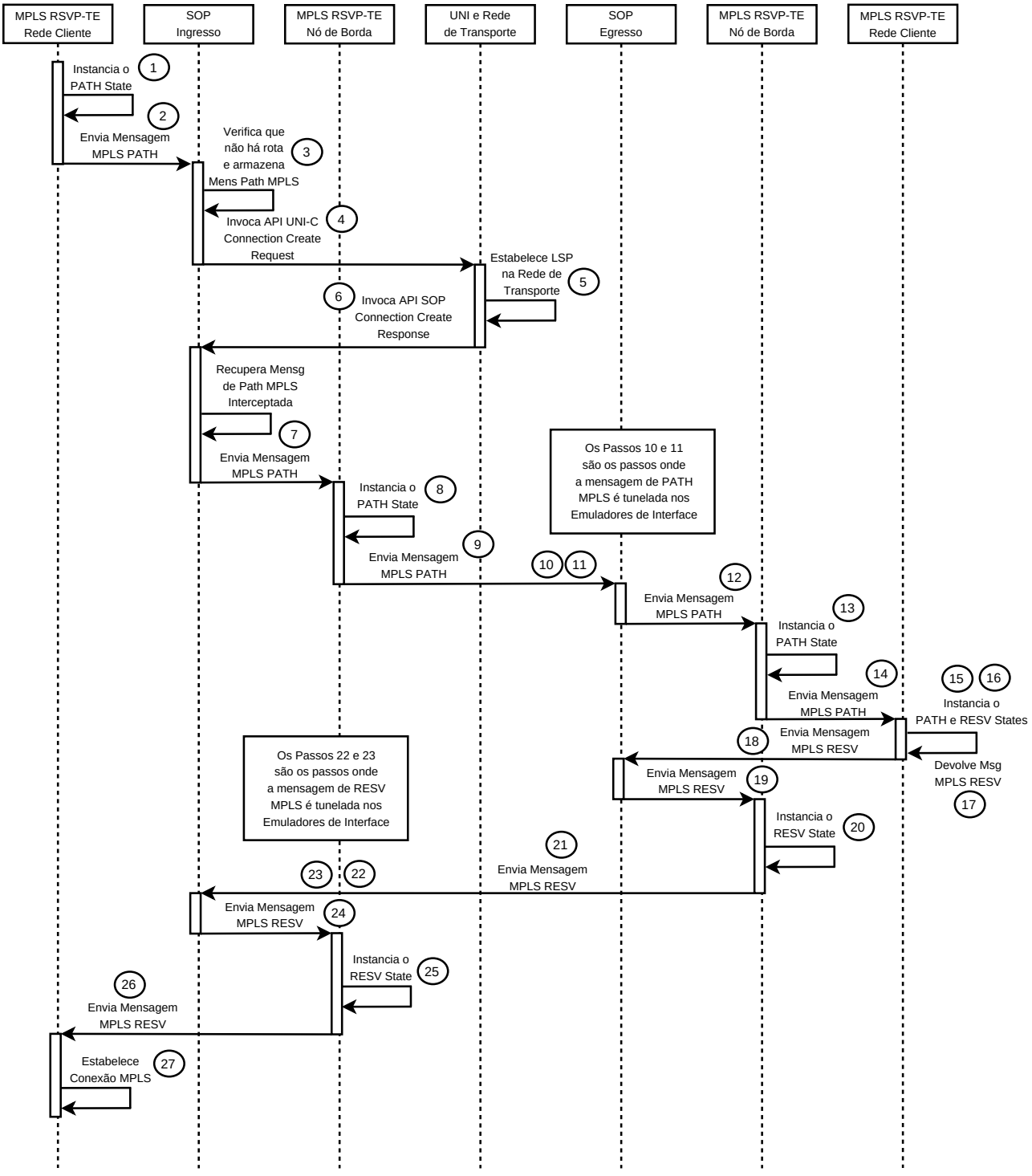

Fig. 5.9: Diagrama de Seqüência do Estabelecimento de LSPs na Rede Cliente.

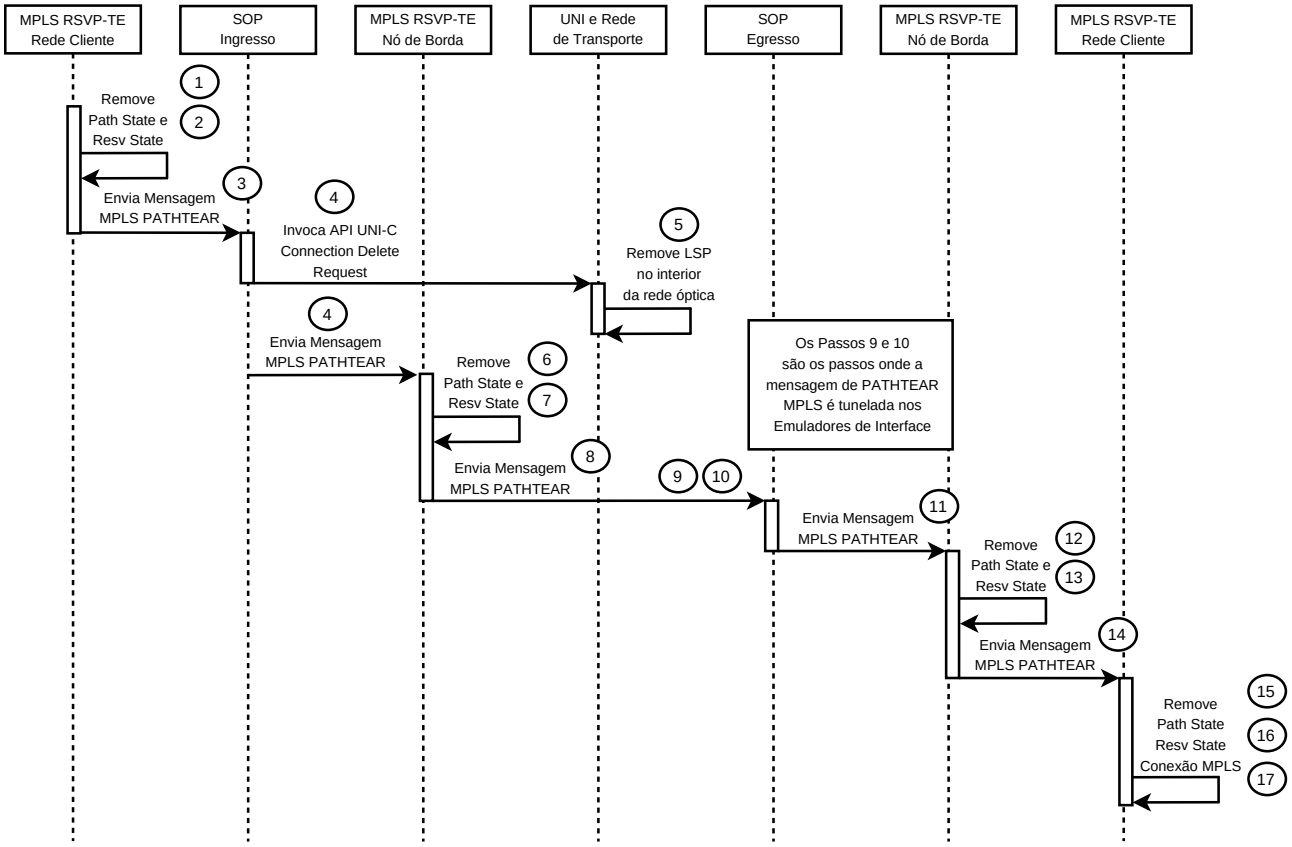

Fig. 5.10: Diagrama de Seqüência da Remoção de LSPs na Rede Cliente.

Capítulo 6

Conclusões e Trabalho Futuros

Este capítulo apresenta as conclusões deste trabalho e apresenta alguns trabalhos futuros que podem ser desenvolvidos a partir da estrutura desenvolvida neste trabalho.

6.1 Conclusões e Trabalhos Futuros

Este trabalho apresentou uma arquitetura de implementação da UNI que possibilita a utilização dos recursos oferecidos pela rede óptica de transporte pelas redes clientes, sem que estas precisem estar cientes da existência da UNI. A arquitetura proposta foi avaliada a partir dos pontos de vista teórico e experimental e em ambas as avaliações os resultados foram consistentes.

A partir do ponto de vista teórico, a arquitetura da UNI proposta é capaz de trabalhar na rede do Projeto GIGA bem como em todos os outros cenários onde a UNI faz-se necessária para prover a interação entre a rede cliente e a rede de transporte.

A partir do ponto de vista experimental, o protótipo desenvolvido é uma importante ferramenta entre a fase de especificação e a fase de efetiva implementação da UNI na rede do Projeto GIGA e os resultados apresentados neste trabalho mostram o correto funcionamento da arquitetura proposta para a UNI.

A próxima fase do Projeto GIGA é a efetiva implementação do plano de controle de sua rede. Novamente é preciso destacar os módulos SOP e SMP. No caso de não haver uma real implementação da UNI, este protótipo pode vir a ser utilizado na rede do Projeto GIGA. Para isto, é apenas necessário especializar estes dois módulos para que eles trabalhem com os protocolos de sinalização existentes na rede cliente e rede de transporte do Projeto GIGA. Obviamente, algumas outras mudanças no protótipo são necessárias para atender alguns requisitos do Projeto GIGA tais como a interação com o algoritmo RWA e o uso dos parâmetros de tráfego especificados pelo Projeto GIGA.

Como mudança na arquitetura, um trabalho futuro poderia investigar o funcionamento dela com

o Modelo Aumentado (Seção 2.3.3). Uma sugestão seria adaptá-la para trabalhar em conformidade com a especificação da UNI do IETF (Seção 3.2), onde apenas uma sessão é estabelecida no interior da rede óptica de transporte.

Existem algumas outras funções, não implementadas neste trabalho, que são desejáveis à arquitetura. São elas: 1) a capacidade de efetuar a recuperação de falhas em múltiplas-camadas e 2) a capacidade de fazer *grooming* [Maesschalck et al., 2003] nas bordas da rede óptica de transporte para proporcionar uma melhor utilização dos recursos disponíveis.

Referências Bibliográficas

[Agrawal, 2002] Agrawal, G. P. (2002). *Fiber-Optic Communication Systems, 3rd Edition*. John Wiley.

[Almquist, 1992] Almquist, P. (1992). Type of Service in the Internet Protocol Suite. *RFC 1349*.

[ASON, 2001] ASON (2001). ITU-T: Architecture for the Automatically Switched Optical Network (ASON), G.8080/Y.1304.

[Awduche et al., 2001a] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and Swallow, G. (2001a). RSVP-TE: Extensions to RSVP for LSP Tunnels. *RFC 3209*.

[Awduche et al., 2001b] Awduche, D., Hannan, A., and Xiao, X. (2001b). Applicability Statement for Extensions to RSVP for LSP-Tunnels. *RFC 3210*.

[Baker et al., 1997] Baker, F., Krawczyk, J., and Sastry, A. (1997). RSVP Management Information Base using SMIv2. *RFC 2206*.

[Berger, 2003a] Berger, L. (2003a). Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description. *RFC 3471*.

[Berger, 2003b] Berger, L. (2003b). Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource reSerVation Protocol-Traffic Engineering (RSPV-TE) Extensions. *RFC 3473*.

[Berger and O'Malley, 1997] Berger, L. and O'Malley, T. (1997). RSVP Extensions for IPSEC Data Flows. *RFC 2207*.

[Blake et al., 1998] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W. (1998). An Architecture for Differentiated Services. *RFC 2475*.

[Braden, 1989] Braden, R. (1989). Requirements for Internet Hosts – Communication Layers. *RFC 1122*.

[Braden et al., 1994] Braden, R., Clark, D., and Shenker, S. (1994). Integrated Services in the Internet Architecture: an Overview. *RFC 1633*.

[Braden and Zhang, 1997] Braden, R. and Zhang, L. (1997). Resource reSerVation Protocol (RSVP) – Version 1 Message Processing Rules. *RFC 2209*.

[Braden et al., 1997] Braden, R., Zhang, L., Berson, S., Herzog, S., and Jamin, S. (1997). Resource reSerVation Protocol (RSVP) - Version 1 Functional Specification. *RFC 2205*.

[Chlamtac et al., 1996] Chlamtac, I., Farago, A., and Zhang, T. (1996). Lightpath (Wavelength) Routing in Large WDM Networks. *IEEE/ACM Transactions on Communications, Volume 14, Number 5*, pages 909 – 913.

[Chlamtac et al., 1992] Chlamtac, I., Ganz, A., and Karmi, G. (1992). Lightpath Communications: an Approach to high-bandwidth Optical WANs. *IEEE/ACM Transactions on Communications, Volume 40, Number 7*, pages 1171 – 1182.

[Dubuc et al., 2006] Dubuc, M., Nadeau, T., Lang, J., and McGinnis, E. (2006). Link Management Protocol (LMP) Management Information Base (MIB). *RFC 4327*.

[Durham and Yavatkar, 1999] Durham, D. and Yavatkar, R. (1999). *Inside the Internet's Resource reSerVation Protocol - Foundations for Quality of Service*. Wiley Computer Publishing.

[Fedyk et al., 2006] Fedyk, D., Aboul-Magd, O., Brungard, D., Lang, J., and Papadimitriou, D. (2006). A Transport Network View of the Link Management Protocol (LMP). *RFC 4394*.

[Fredette and Lang, 2005] Fredette, A. and Lang, J. (2005). Link Management Protocol (LMP) for Dense Wavelength Division Multiplexing (DWDM) Optical Line Systems. *RFC 4209*.

[Heinanen et al., 1999] Heinanen, J., Baker, F., Weiss, W., and Wroclawski, J. (1999). Assured Forwarding PHB Group. *RFC 2597*.

[Jacobson et al., 1999] Jacobson, V., Nichols, K., and Poduri, K. (1999). An Expedited Forwarding PHB. *RFC 2598*.

[Katz et al., 2003] Katz, D., Kompella, K., and Yeung, D. (2003). Traffic Engineering (TE) Extensions to OSPF Version 2. *RFC 3630*.

[Keshav, 1997] Keshav, S. (1997). *An Engineering Approach to Computer Networking*. Addison-Wesley.

[Kompella and Rekhter, 2003] Kompella, K. and Rekhter, Y. (2003). Signalling Unnumbered Links in Resource reSerVation Protocol - Traffic Engineering (RSVP-TE). *RFC 3477*.

[Kompella and Rekhter, 2005] Kompella, K. and Rekhter, Y. (2005). OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS). *RFC 4203*.

[Lang, 2005] Lang, J. (2005). Link Management Protocol (LMP). *RFC 4204*.

[Lang and Papadimitriou, 2005] Lang, J. and Papadimitriou, D. (2005). Synchronous Optical Network (SONET)/Synchronous Digital Hierarchy (SDH) Encoding for Link Management Protocol (LMP) Test Messages. *RFC 4207*.

[Laubach, 1994] Laubach, M. (1994). Classical IP and ARP over ATM. *RFC 1577*.

[Lin and Pendarakis, 2003] Lin, Z. and Pendarakis, D. (2003). Documentation of IANA assignments for Generalized MultiProtocol Label Switching (GMPLS) Resource reServation Protocol - Traffic Engineering (RSVP-TE) Usage and Extensions for Automatically Switched Optical Network (ASON). *RFC 3474*.

[Luciani, 1998] Luciani, J. (1998). Classical IP and ARP over ATM to NHRP transition. *RFC 2336*.

[Luciani et al., 1998] Luciani, J., Katz, D., Piscitello, D., Cole, B., and Doraswamy, N. (1998). NBMA Next Hop Resolution Protocol (NHRP). *RFC 2332*.

[Maesschalck et al., 2003] Maesschalck, S. D., Pickavet, M., Colle, D., and Demeester, P. (2003). Multi-layer Traffic Grooming in networks with an IP/MPLS Layer on top of a meshed Optical Layer. *GLOBECOM*, pages 2750 – 2754.

[Magalhães and Cardozo, 2003a] Magalhães, M. F. and Cardozo, E. (2003a). Introdução à Comutação IP por Rótulos Através de MPLS. *Apostila da disciplina IA-003*.

[Magalhães and Cardozo, 2003b] Magalhães, M. F. and Cardozo, E. (2003b). Qualidade de Serviço. *Apostila da disciplina IA-003*.

[Mankin et al., 1997] Mankin, A., Baker, F., Braden, B., Bradner, S., O'Dell, M., Romanow, A., Weinrib, A., and Zhang, L. (1997). Resource reSerVation Protocol (RSVP) Version 1 Applicability Statement Some Guidelines on Deployment. *RFC 2208*.

[Mannie, 2004] Mannie, E. (2004). Generalized Multi-Protocol Label Switching (GMPLS) Architecture. *RFC 3945*.

[Margaria et al., 2005] Margaria, C., Juillot, G., and Autenrieth, A. (2005). Performance Evaluation of a GMPLS Prototype. *Broadband Communications and Distributed Systems - IV Workshop in MPLS/GMPLS networks*, pages 211 – 222.

[Moy, 1991] Moy, J. (1991). OSPF Version 2. *RFC 1247*.

[Newman et al., 1998] Newman, P., Minshall, G., and Lyon, T. L. (1998). IP Switching - ATM Under IP. *IEEE/ACM Transactions on Networking, Volume 6, Number 2*, pages 117 – 129.

[Nichols et al., 1998] Nichols, K., Blake, S., Baker, F., and Black, D. (1998). Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. *RFC 2474*.

[Papadimitriou and Verchere, 2005] Papadimitriou, D. and Verchere, D. (2005). GMPLS User-Network Interface in Support of End-to-End Rerouting. *IEEE Communication Magazine*, pages 35 – 43.

[Rajagopalan, 2003] Rajagopalan, B. (2003). Documentation of IANA Assignments for Label Distribution Protocol (LDP), Resource reSerVation Protocol (RSVP), and Resource reSerVation Protocol-Traffic Engineering (RSVP-TE) Extensions for Optical UNI Signaling. *RFC 3476*.

[Ramaswami and Sivarajan, 1995] Ramaswami, R. and Sivarajan, K. (1995). Routing and Wavelength Assignment in all-optical Networks. *IEEE/ACM Transactions on Networking, Volume 3, Number 5*, pages 489 – 500.

[Rekhter et al., 1997] Rekhter, Y., Davie, B., Katz, D., Rosen, E., and Swallow, G. (1997). Cisco Systems' Tag Switching Architecture Overview. *RFC 2105*.

[Rosen et al., 2001] Rosen, E., Viswanathan, A., and Callon, R. (2001). Multiprotocol Label Switching Architecture. *RFC 3031*.

[Siqueira, 2002] Siqueira, M. (2002). Uma Arquitetura de Políticas Para Gerência de Redes MPLS. Dissertação de Mestrado, Unicamp - FEEC - DCA. Orientador: Prof. Dr. Maurício Ferreira Magalhães. [*Online*]. Disponível em: `http://libdigi.unicamp.br`.

[Swallow et al., 2005] Swallow, G., Drake, J., Ishimatsu, H., and Rekhter, Y. (2005). Generalized Multi-Protocol Label Switching (GMPLS) User-Network Interface (UNI): Resource reSerVation Protocol-Traffic Engineering (RSPV-TE) Support for the Overlay Model. *RFC 4208*.

[Talpade and Ammar, 1997] Talpade, R. and Ammar, M. (1997). Multicast Server Architectures for MARS-based ATM multicasting. *RFC 2149*.

[Tanenbaum, 1997] Tanenbaum, A. S. (1997). *Redes de Computadores, 3ª Edição*. Editora Campus.

[UNI Common Part, 2004] UNI Common Part (2004). OIF User Network Interface (UNI) 1.0 Signaling Specification, Release 2: Common Part.

[UNI RSVP Extensions, 2004] UNI RSVP Extensions (2004). RSVP Extensions for User Network Interface (UNI) 1.0 Signaling, Release 2.

[Wroclawski, 1997] Wroclawski, J. (1997). The Use of RSVP with IETF Integrated Services. *RFC 2210*.

[Zuliani, 2006] Zuliani, L. (2006). Arquitetura e Implementação de um serviço de Informações Topológicas e de Engenharia de Tráfego para sistemas RWA. Dissertação de Mestrado, Unicamp - FEEC - DCA. Orientador: Prof. Dr. Maurício Ferreira Magalhães. [*Online*]. Disponível em: `http://libdigi.unicamp.br`.

Apêndice A

Diagramas de Seqüência

Este apêndice traz os diagramas de seqüência que detalham o estabelecimento e a remoção de conexões utilizando a arquitetura proposta na rede do Projeto GIGA.

O estabelecimento de conexões foi detalhado em três diagramas de seqüência. O primeiro diagrama apresenta os procedimentos que ocorrem na rede cliente sem apresentar os detalhes do estabelecimento na rede de transporte. Os outros dois diagramas descrevem os procedimentos que ocorrem durante o estabelecimento de uma conexão no interior da rede de transporte. Eles foram desenhados separadamente para poder facilitar a visualização dos procedimentos.

A remoção de conexões também foi detalhada em três diagramas de seqüência. Similarmente aos diagramas de seqüência de estabelecimento de conexões, o primeiro descreve os procedimentos que ocorrem na rede cliente sem apresentar os detalhes da rede de transporte durante a remoção de conexões e os outros dois diagramas descrevem os procedimentos que ocorrem no interior da rede de transporte durante a remoção de conexões. Os dois últimos diagramas também foram desenhados separadamente para facilitar a visualização dos procedimentos envolvidos.

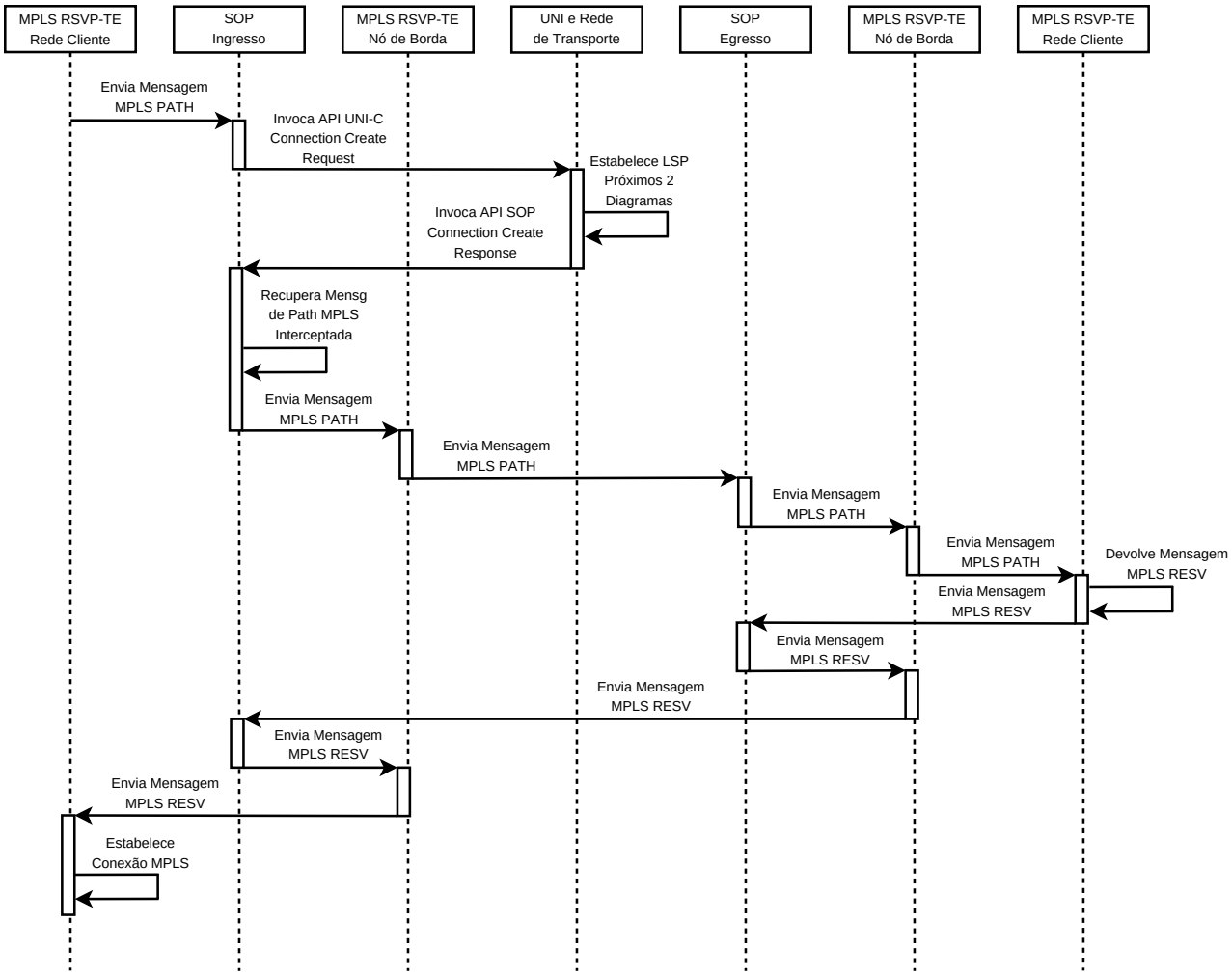

Fig. A.1: Diagrama de Seqüência do Estabelecimento de LSPs - Ações que ocorrem na Rede Cliente.

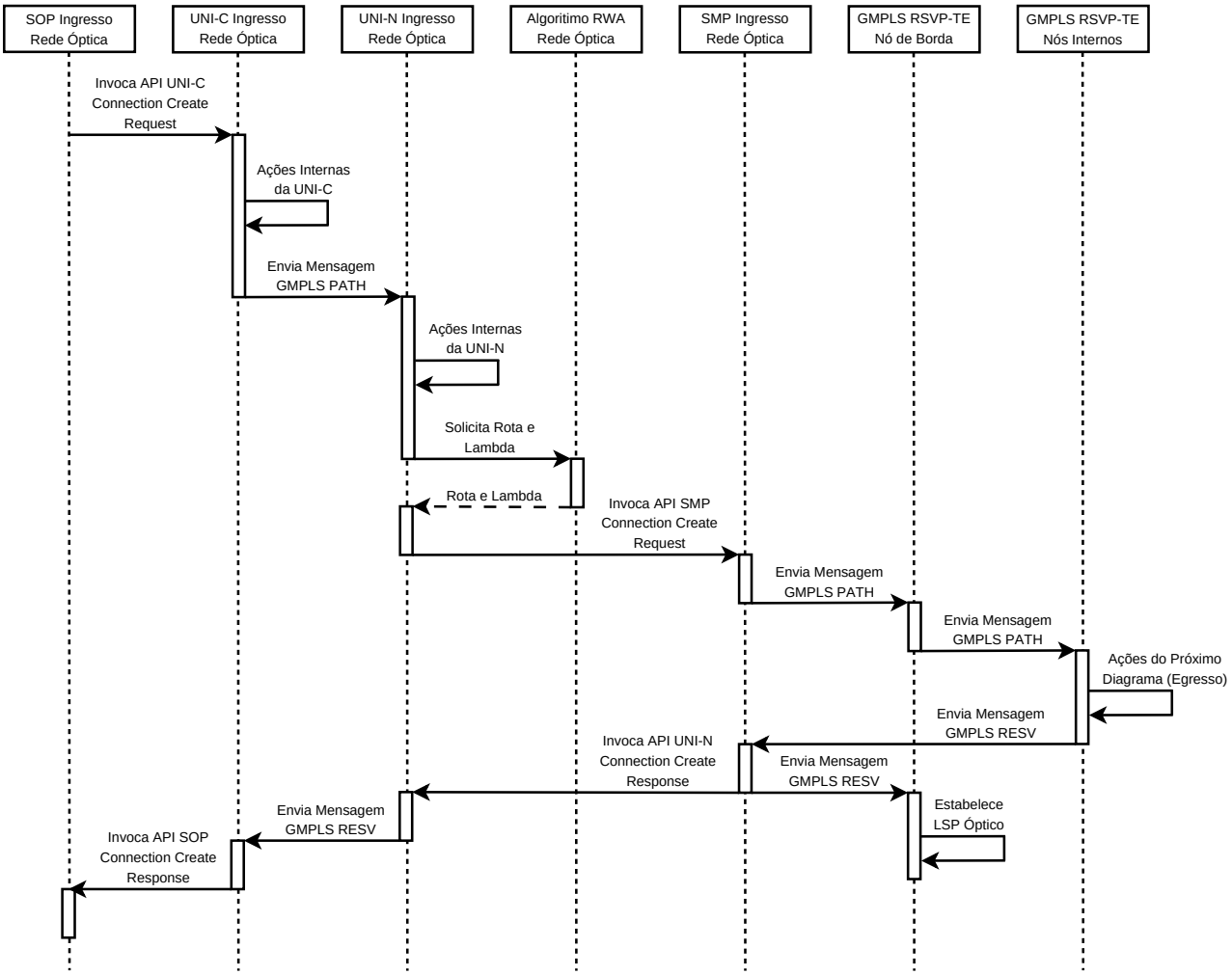

Fig. A.2: Diagrama de Seqüência do Estabelecimento de LSPs - Ações que ocorrem no Ingresso da Rede de Transporte.

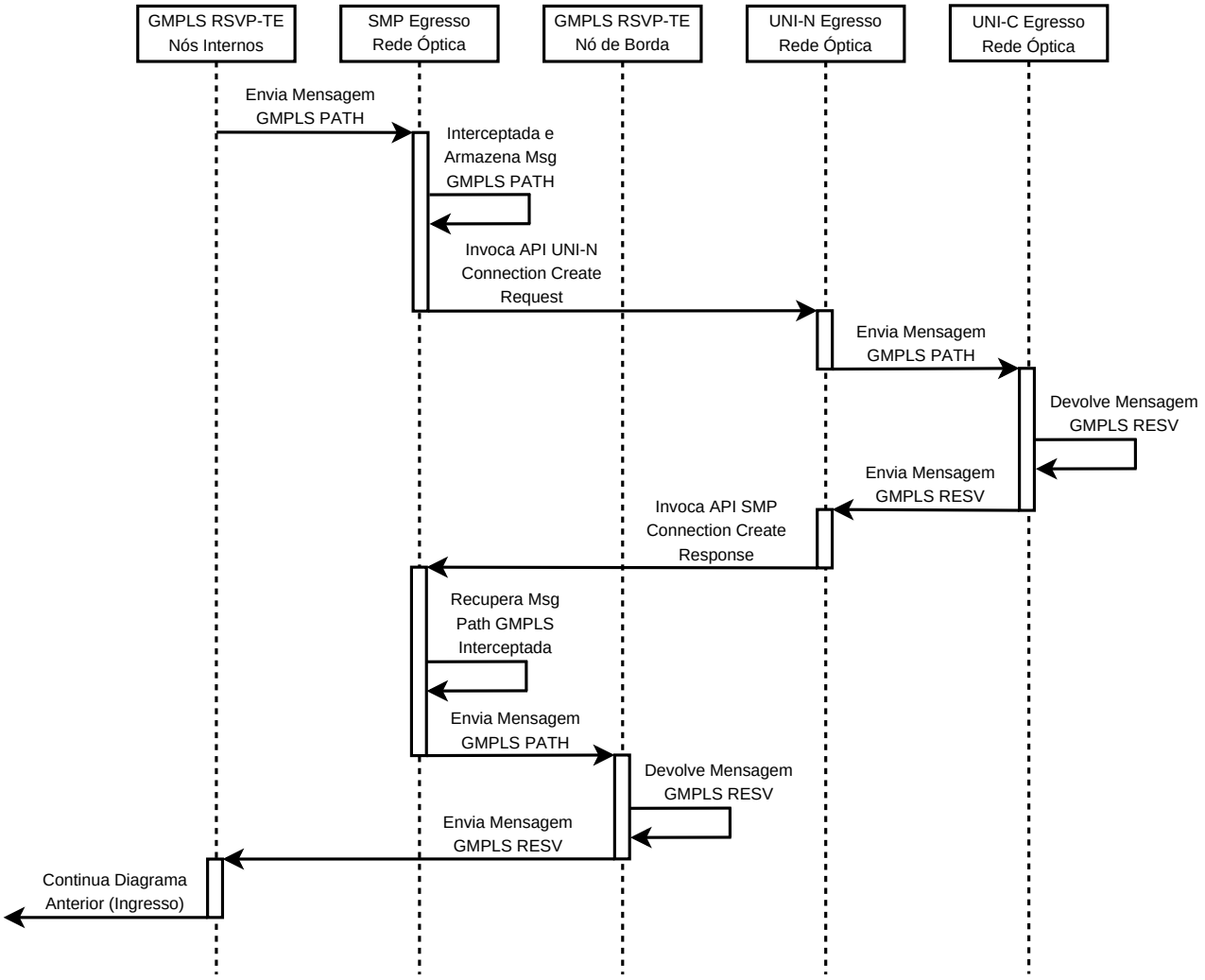

Fig. A.3: Diagrama de Seqüência do Estabelecimento de LSPs - Ações que ocorrem no Egresso da Rede de Transporte.

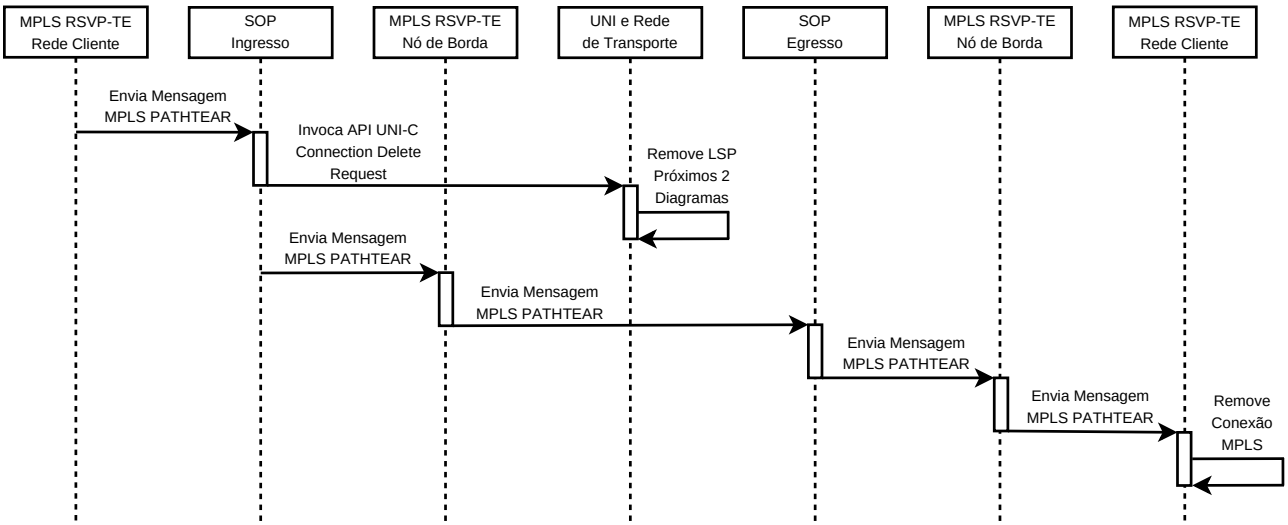

Fig. A.4: Diagrama de Seqüência da Remoção de LSPs - Ações que ocorrem na Rede Cliente.

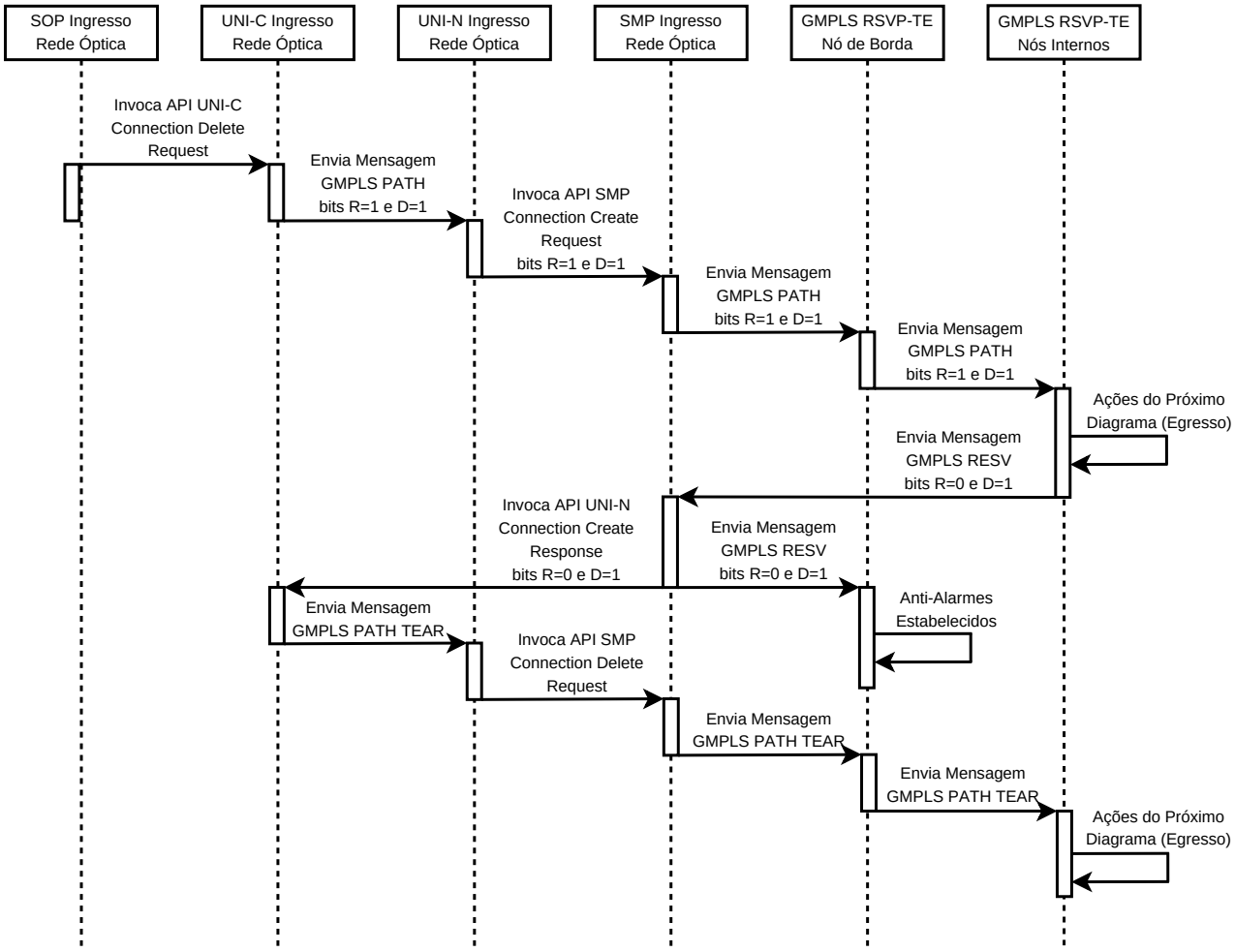

Fig. A.5: Diagrama de Seqüência da Remoção de LSPs - Ações que ocorrem no Ingresso da Rede de Transporte.

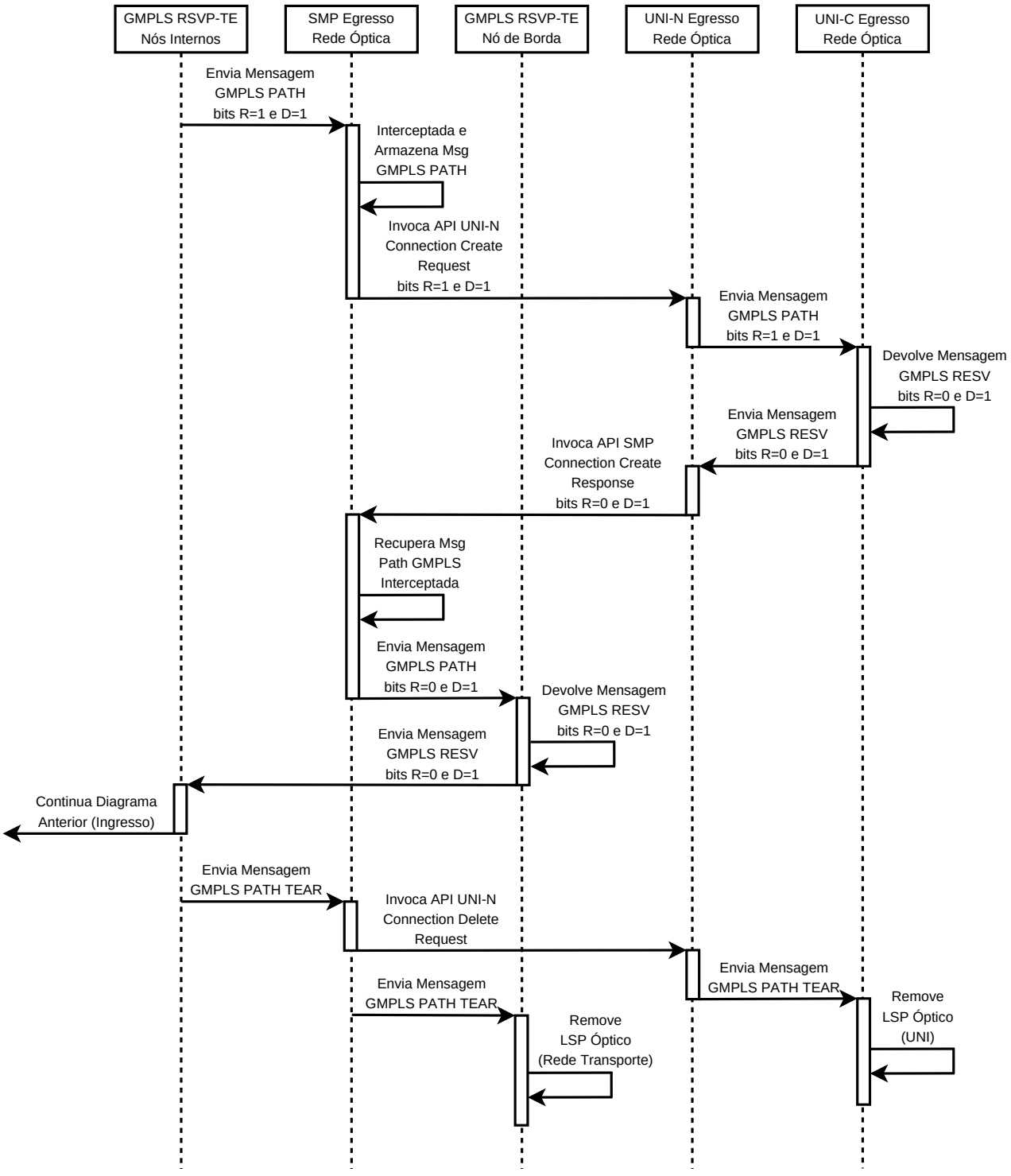

Fig. A.6: Diagrama de Seqüência da Remoção de LSPs - Ações que ocorrem no Egresso da Rede de Transporte.

Apêndice B

Diagramas de Classes

Este apêndice traz os diagramas de classes que detalham a especificação dos módulos que constituem o protótipo implementado.

Primeiramente, são apresentados os diagramas de classes dos módulos que constituem a Arquitetura proposta (SOP, UNI-C, UNI-N e SMP). Logo após, são apresentados os diagramas de classes dos módulos que foram desenvolvidos para que fosse possível criar um ambiente de testes para a Arquitetura proposta (MPLS RSVP-TE, GMPLS RSVP-TE, Servidor de Rotas e Emulador de Interfaces Ópticas).

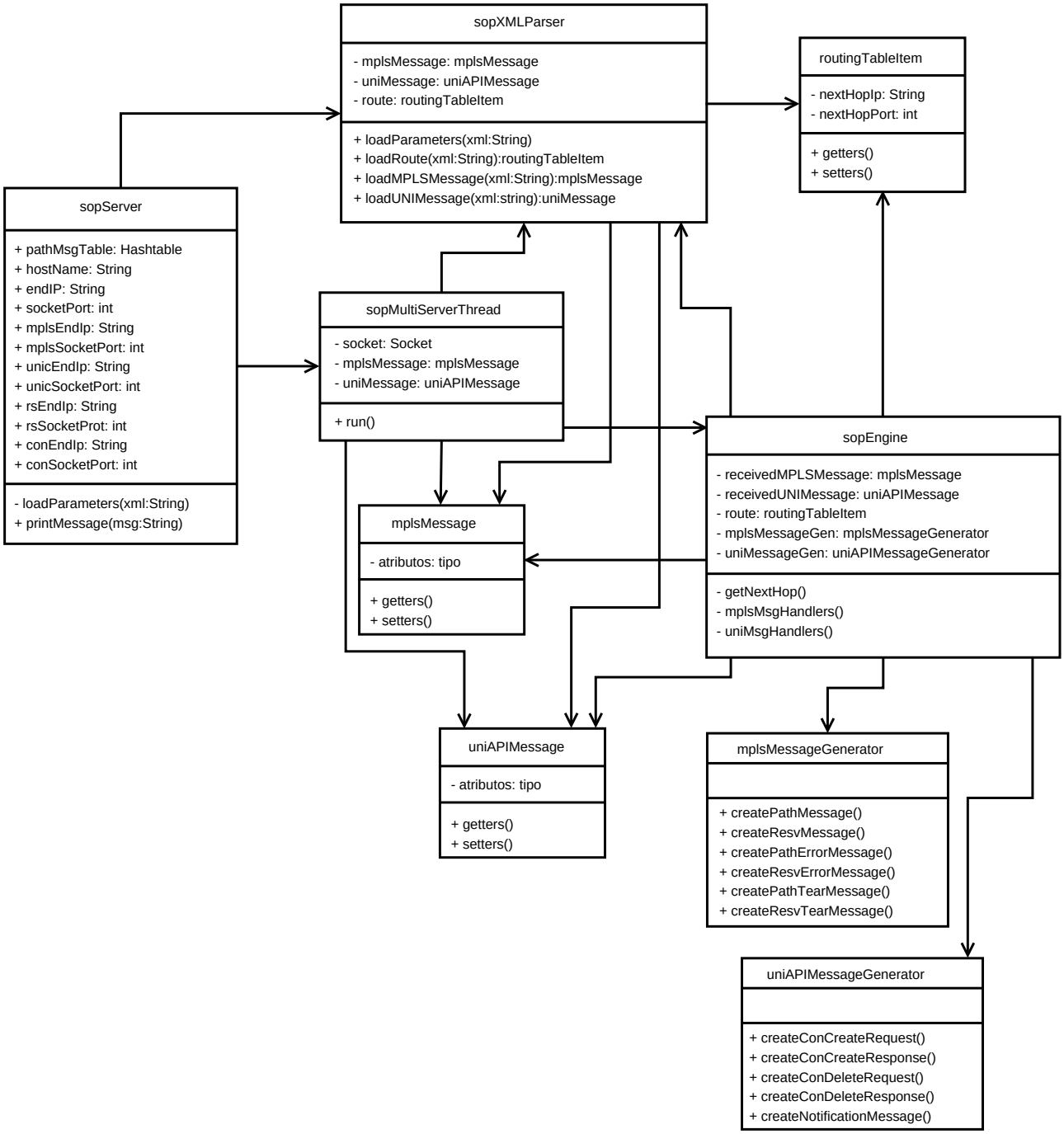

Fig. B.1: Diagrama de Classes do SOP.

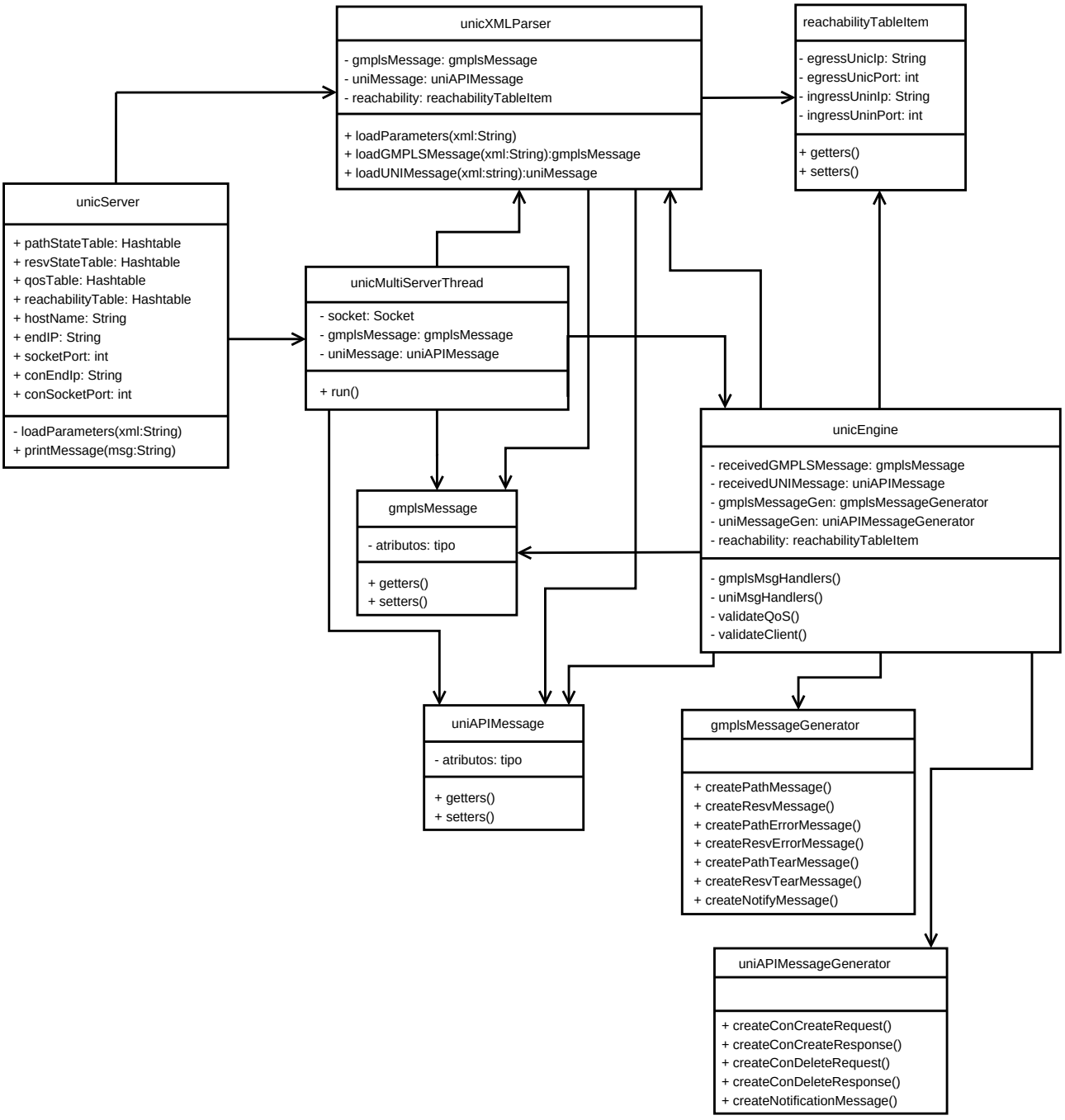

Fig. B.2: Diagrama de Classes da UNI-C.

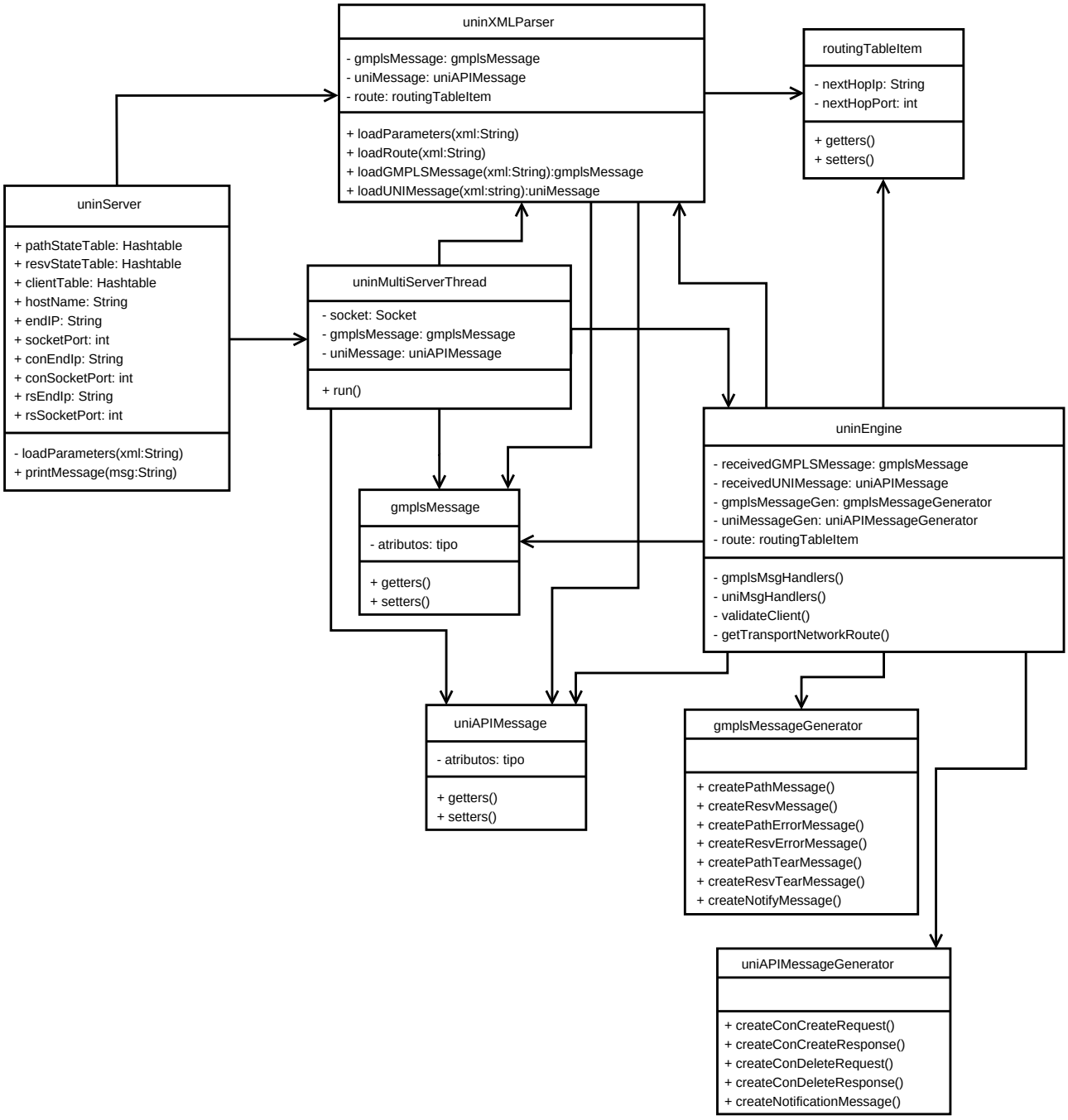

Fig. B.3: Diagrama de Classes da UNI-N.

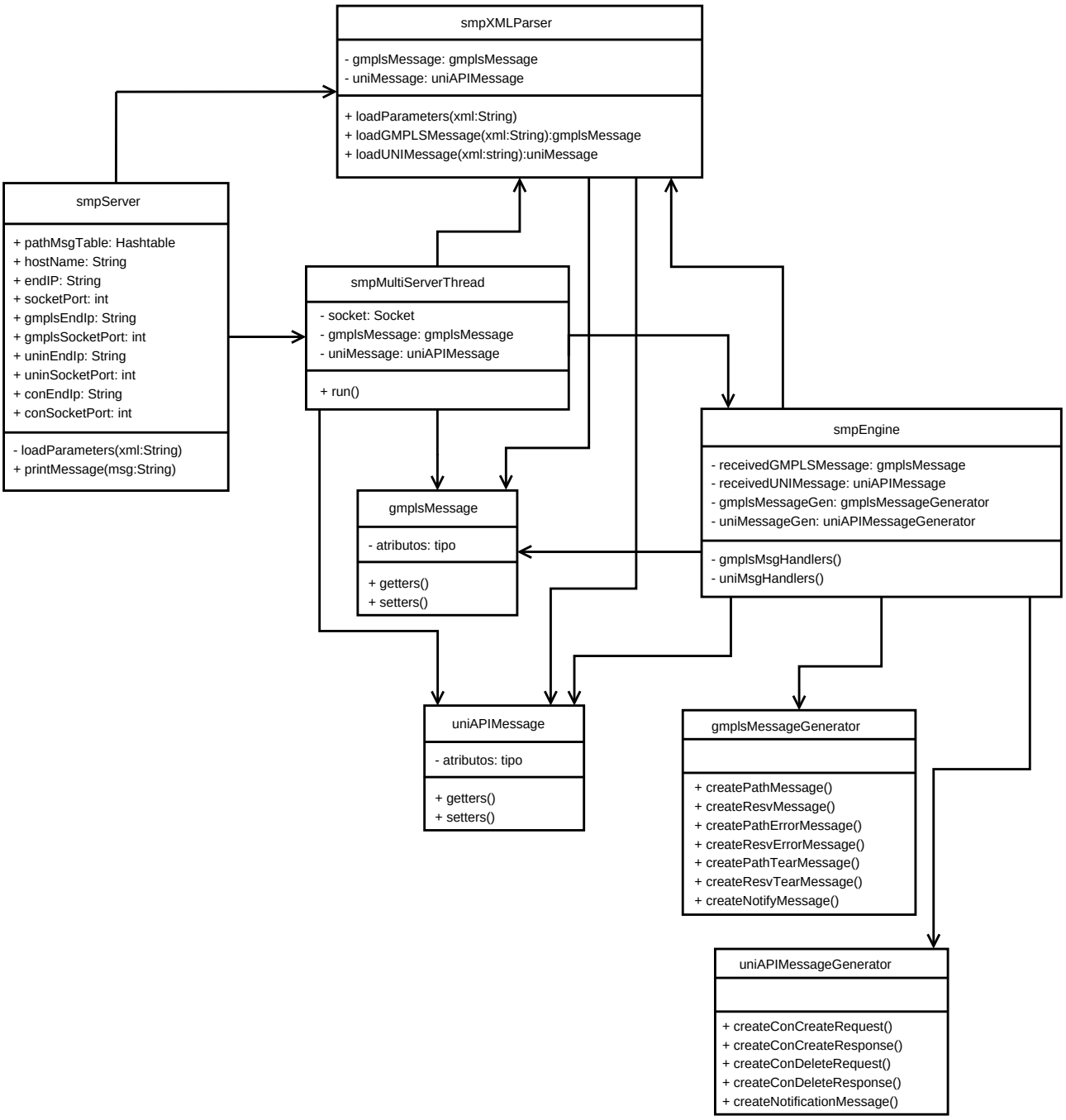

Fig. B.4: Diagrama de Classes do SMP.

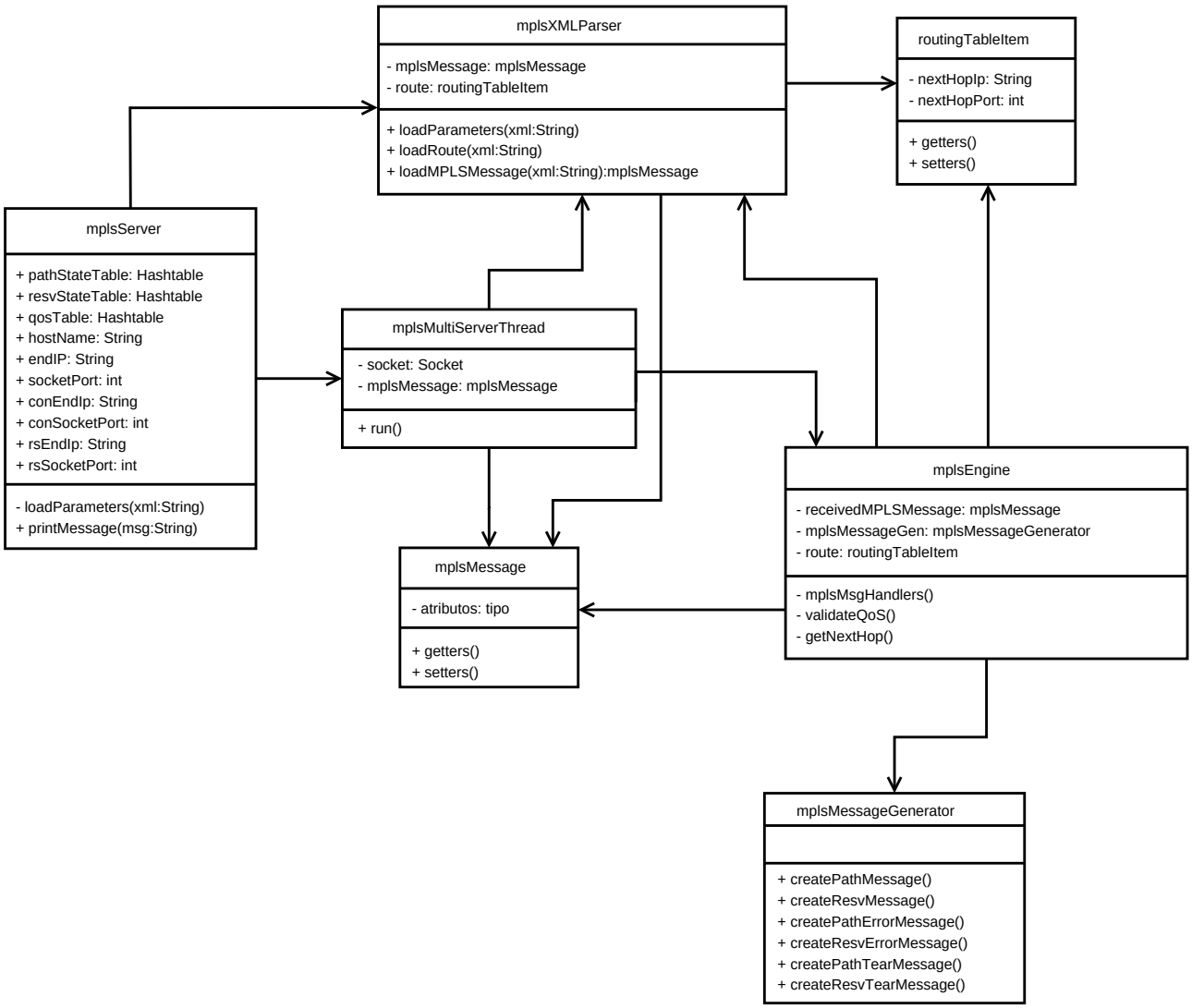

Fig. B.5: Diagrama de Classes do MPLS RSVP-TE.

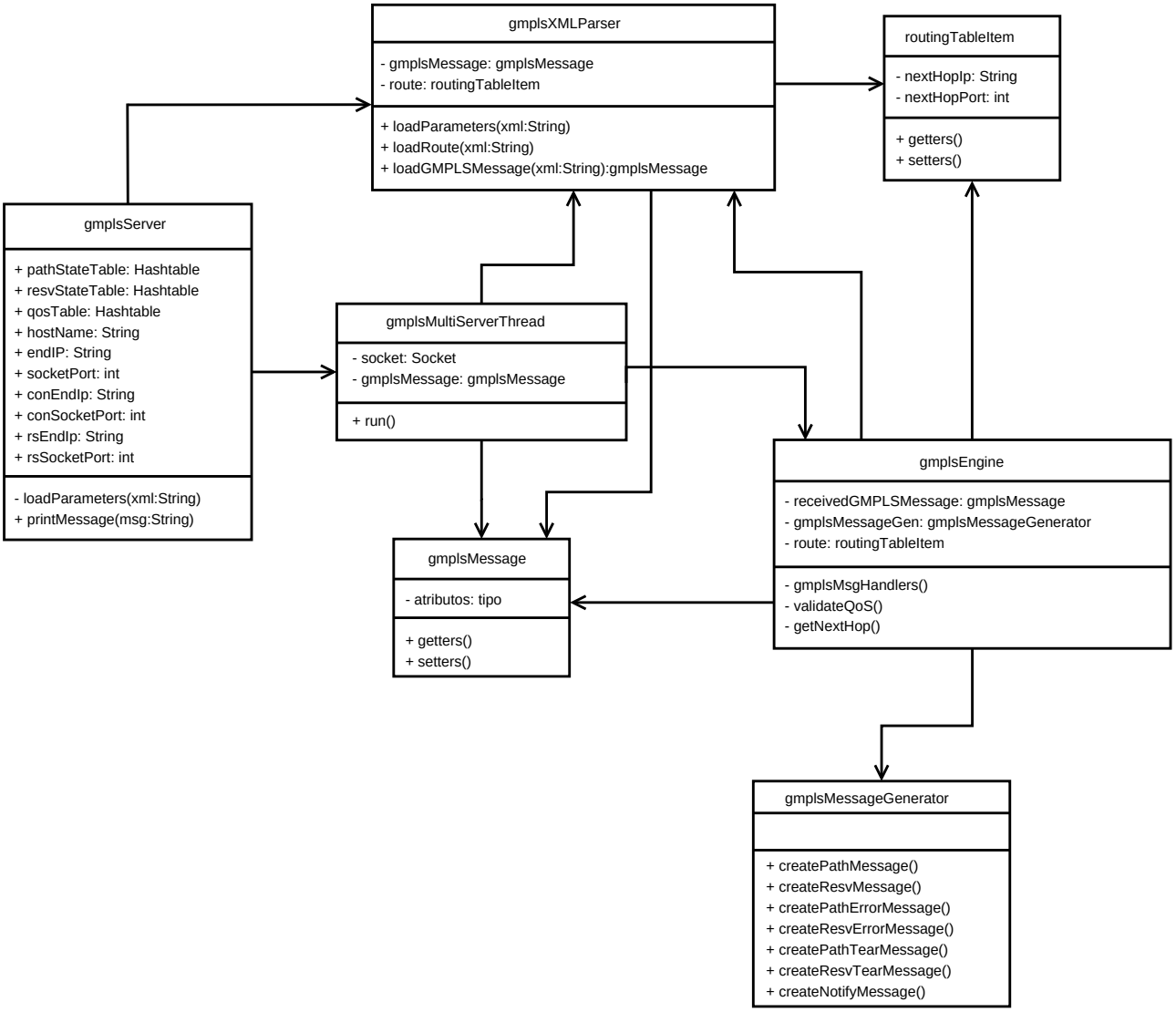

Fig. B.6: Diagrama de Classes do GMPLS RSVP-TE.

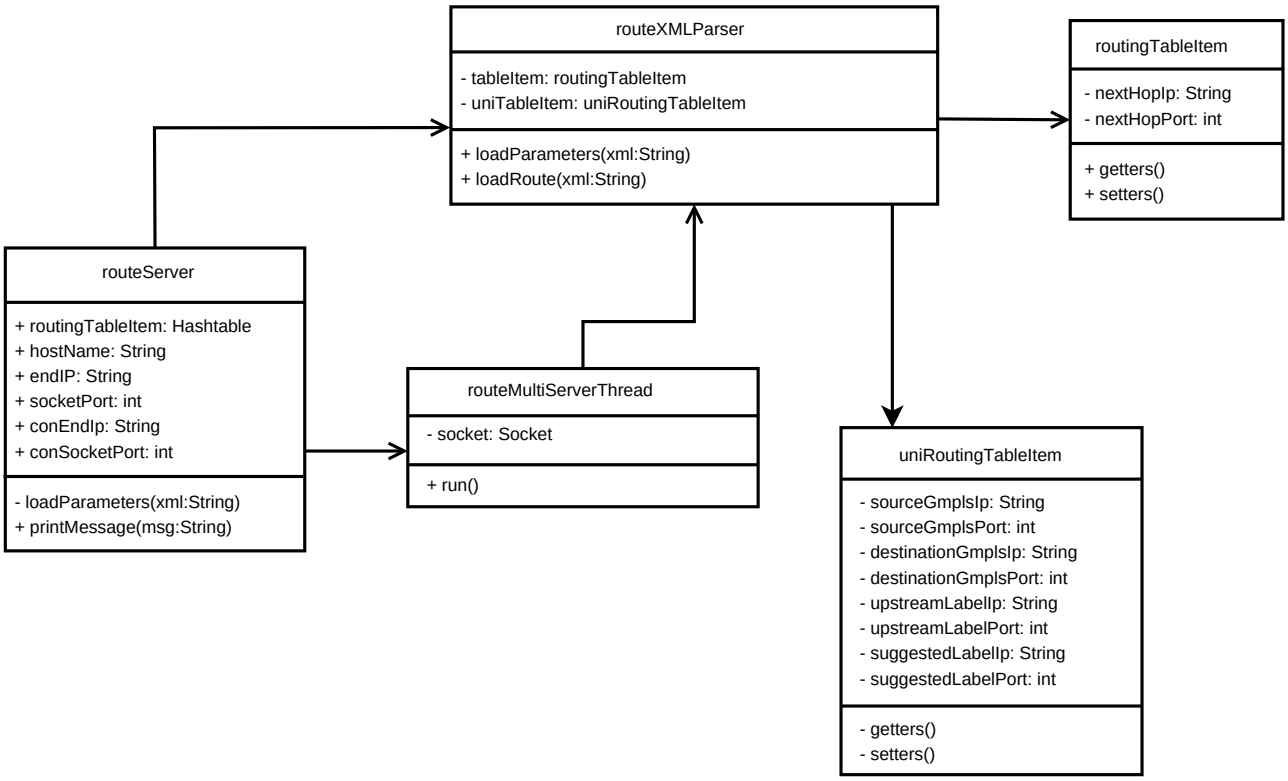

Fig. B.7: Diagrama de Classes do Servidor de Rotas.

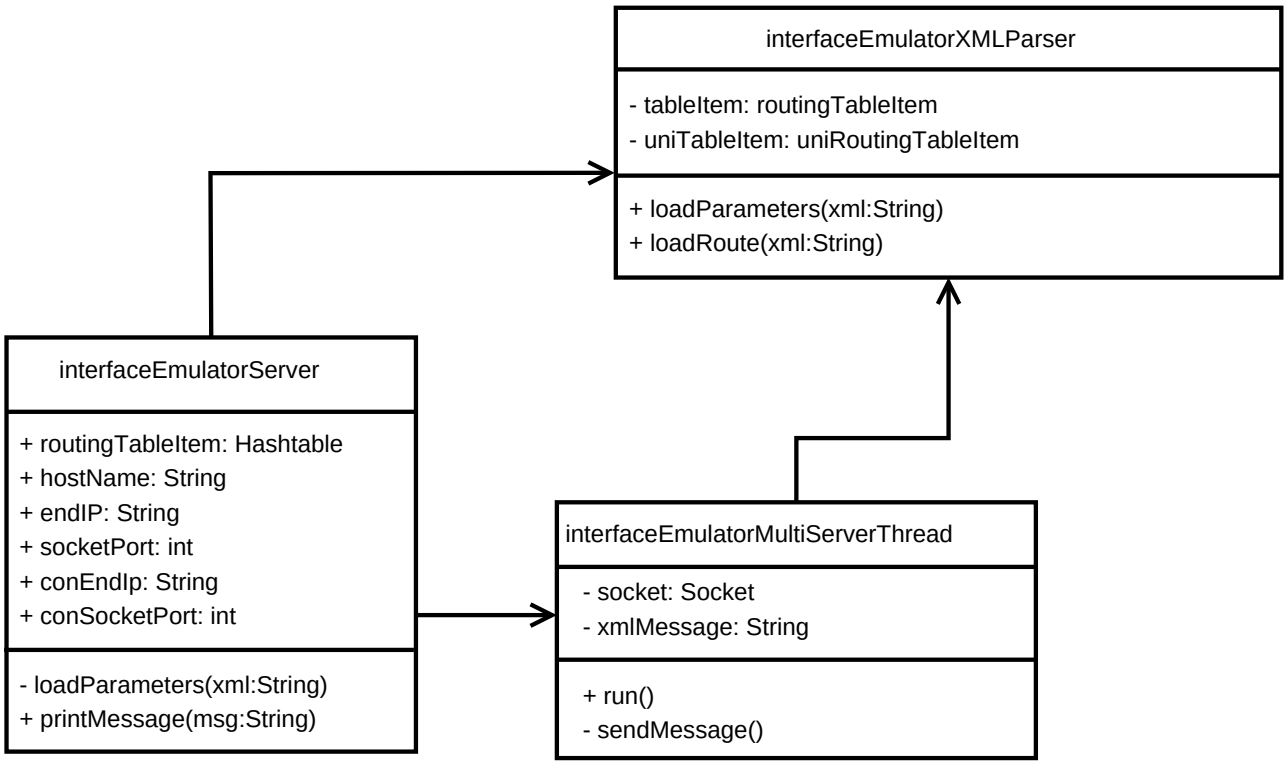

Fig. B.8: Diagrama de Classes do Emulador de Interfaces Ópticas.