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

Fábio Luciano Verdi

Uma Arquitetura para Provisionamento e Gerência de Serviços em Redes Ópticas

Campinas, SP

2006

Fábio Luciano Verdi

Uma Arquitetura para Provisionamento e Gerência de Serviços em Redes Ópticas

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: **Engenharia de Computação**.

Orientador: Prof. Dr. Maurício Ferreira Magalhães
Co-orientador: Edmundo Roberto Mauro Madeira

Campinas, SP

2006

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

Verdi, Fábio Luciano
V584a Uma Arquitetura para Provisionamento e Gerência de Serviços em Redes Ópticas / Fábio Luciano Verdi. – Campinas, SP: [s.n.], 2006.

Orientadores: Maurício Ferreira Magalhães; Edmundo Roberto Mauro Madeira.
Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Arquitetura de redes computador. 2. Interconexão de redes (Telecomunicações). 3. Internet (Redes de computação). 4. Redes de computação - Protocolos. 5. Comunicações ópticas. 6. Programação (Computadores). I. Magalhães, Maurício Ferreira. II. Madeira, Edmundo Roberto Mauro. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título

Título em Inglês:	An Architecture for Services Provisioning and Management in Optical Networks
Palavras-chave em Inglês:	Provisioning and Management of Services, Optical Networks, Policies, Interdomain Services, Web Services
Área de concentração:	Engenharia de Computação
Titulação:	Doutor em Engenharia Elétrica
Banca Examinadora:	Eleri Cardozo , Hélio Waldman, Leonardo de Souza Mendes, Amilcar Careli César e Lisandro Zambenedetti Granville
Data da defesa:	27/11/2006

Fábio Luciano Verdi

Uma Arquitetura para Provisionamento e Gerência de Serviços em Redes Ópticas

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: **Engenharia de Computação**.

Banca Examinadora:

Prof. Dr. Maurício Ferreira Magalhães - DCA/FEEC/Unicamp
Prof. Dr. Amilcar Careli César - USP-São Carlos
Prof. Dr. Eleri Cardozo - DCA/FEEC/Unicamp
Prof. Dr. Hélio Waldman - DECOM/FEEC/Unicamp
Prof. Dr. Leonardo de Souza Mendes - DECOM/FEEC/Unicamp
Prof. Dr. Lisandro Zambenedetti Granville - Instituto de Informática/UFRGS

Campinas, SP

2006

Resumo

Nos últimos anos têm-se discutido uma maneira para envio de fluxos de pacotes entre domínios que leve em consideração aspectos relacionados com a qualidade de serviço de rede (banda, atraso, etc.). Entretanto, as soluções até agora apresentadas dependem de extensões que precisam ser feitas principalmente nos protocolos de roteamento entre domínios para suportar a distribuição de informações de engenharia de tráfego. Porém, tais extensões dependem de longos processos de padronização que, muitas vezes, apesar de profundas discussões, não conseguem efetivamente definir um padrão. Isto tem ocorrido nas redes IPs onde extensões ao BGP (*Border Gateway Protocol*) para distribuição de informações de engenharia de tráfego têm sido propostas, porém, nunca colocadas em prática. Recentemente, a mesma discussão surgiu nos cenários envolvendo redes ópticas para estabelecimento de conexões entre domínios. Esta tese apresenta uma alternativa em relação aos processos de padronização que propõem extensões aos protocolos para que os mesmos suportem o provisionamento de serviços entre domínios. Especificamente, a arquitetura para provisionamento e gerência de serviços entre domínios ópticos age de forma independente do plano de controle, ou seja, a proposta apresentada neste trabalho não depende de extensões aos protocolos usados atualmente no roteamento e sinalização entre domínios. A arquitetura apresentada nesta tese cria uma camada de serviços que facilita a interação entre diferentes domínios administrativos. Todas as interações para troca de informações de roteamento e estabelecimento de conexões são feitas na camada de serviços. Internamente a cada domínio, os caminhos de luz são estabelecidos usando tecnologias locais tais como a arquitetura GMPLS (*Generalized Multiprotocol Label Switching*) ou ASON (*Automatic Switched Optical Network*). Porém, o provisionamento de serviços entre domínios é realizado não pelo plano de controle, mas sim pelo plano de serviços. Embora o foco da tese seja o provisionamento de serviços entre domínios, a arquitetura suporta também o provisionamento de serviços intra domínios. Os serviços de conexões e VPNs (*Virtual Private Networks*) dentro de um domínio e entre domínios administrativos diferentes serão oferecidos. A arquitetura é baseada no modelo TMN (*Telecommunications Management Network*), principalmente no que se refere às camadas do modelo e à nomenclatura. Usamos a tecnologia *Web Services* para implementação da arquitetura e uma análise de tal tecnologia foi realizada a fim de avaliarmos o seu uso para estabelecimento de serviços entre domínios. Os resultados apresentados indicam que o provisionamento de serviços realizado através da camada de serviços facilita e flexibiliza as interações entre domínios oferecendo aos provedores uma novo mecanismo para suporte a estas interações.

Palavras-chave: Provisionamento e Gerência de Serviços, Redes Ópticas, Políticas, Serviços entre Domínios, *Web Services*.

Abstract

In the last few years it has been discussed a way for sending interdomain packet flows taking into account some aspects related to the network quality of service (bandwidth, delay, etc.). However, the

solutions presented so far depend on the extensions that need to be done mainly in the interdomain routing protocols to support the distribution of traffic engineering information. Nevertheless, such extensions depend on long-process of standardization that, in many cases, does not reach a consensus and a standard is not effectively defined. An example of this is that in IP networks the proposed extensions for the BGP (Border Gateway Protocol) to support the distribution of traffic engineering information were not put into practice until now. Recently, the same discussion came through in the optical network scenario. This thesis presents an alternative to standardization processes. We propose an architecture for provisioning and management of interdomain services in optical networks. Such architecture acts independent of the control plane and does not depend on the protocol extensions that are needed to support interdomain routing and signaling. The architecture has a service layer by which all the interactions between different optical domains are done. Within each domain, the lightpaths are established using local solutions such as GMPLS (Generalized Multiprotocol Label Switching) or ASON (Automatic Switched Optical Network). However, the provisioning of interdomain services is not done in the control plane. Instead, it is done in the services layer. Although the focus of this thesis is on the provisioning of interdomain services, the architecture also supports the establishment of intradomain services. We are particularly interested in the provisioning of connections and VPNs (Virtual Private Networks) within a domain as well as between domains. The architecture is based on the TMN model (Telecommunications Management Network). The TMN model is used to divide the architecture into layers and to define the nomenclature of each layer. The Web Services technology was used to implement the architecture. The implementation was done to validate the architecture and to analyse the usage of Web Services to establish interdomain optical network services. The results presented in this work highlight that by doing the provisioning of services through the services layer facilitate the implementation of the interactions among different domains and offer a new mechanism to support such interactions.

Keywords: Provisioning and Management of Services, Optical Networks, Policies, Interdomain Services, Web Services.

Agradecimentos

Primeiramente agradeço a Deus pela motivação e força dadas durante toda a minha trajetória acadêmica.

Agradeço à minha família que mesmo estando longe sempre me apoiou com palavras de conforto e perseverança.

Ao meu orientador Prof. Maurício Magalhães pela sua capacidade técnica e excelente orientação. Não foi apenas um orientador, mas sim um amigo sempre disposto a conversar, aceitar idéias e propor soluções.

Ao meu co-orientador Prof. Edmundo Madeira, que vem fazendo parte da minha vida acadêmica desde o mestrado. Também um grande amigo sempre presente em todos os momentos do doutorado.

Ao grupo de amigos do LCA e do IC principalmente os vinculados ao projeto *Web Services* e projeto GIGA: Rafael Duarte, Fabrízzio de Lacerda, Rafael Pasquini, Luiz Gustavo Zuliani, Cláudio Carvalho, Neumar Malheiros, Daniel Barboza e Walter Wong. Foi uma honra ter trabalhado com todos, tanto pelas suas intelectualidades peculiares mas também pela dedicação aos projetos. Esta tese é o resultado da união de todos. Agradeço pela convivência, apoio e companheirismo.

Aos demais colegas da pós-graduação, pelos momentos de discontração proporcionados durante esta jornada.

À Ericsson e à CAPES pelo apoio financeiro.

A Deus e à minha família

Sumário

Lista de Figuras

Lista de Tabelas

Glossário

AC	Admission Control
ADM	Add-Drop Multiplexor
AS	Advertising Service
ASON	Automatically Switched Optical Network
BGP	Border Gateway Protocol
BPMN	Business Process Modeling Notation
CC/NMI	Connection Controller/Network Management Interface
CIM	Common Information Model
CORBA	Common Object Request Broker Architecture
CPI	Customer Port Identifier
DCOM	Distributed Component Object Model
DWDM	Dense Wavelength Division Multiplexing
E-NNI	External Network-to-Network Interface
E2ECS	End-to-End Connection Service
E2ENS	End-to-End Negotiation Service
FA	Forwarding Adjacencies
FCAPS	Fault, Configuration, Accounting, Performance and Security
FM	Fault Manager
GMPLS	Generalized MultiProtocol Label Switching
I-NNI	Internal Network-to-Network Interface
IDL	Interface Definition Language

IETF	Internet Engineering Task Force
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
LSP	Label Switched Path
LSR	Label Switched Router
MIB	Management Information Base
MM	Membership Manager,
NEA	Network Element Agent
NNI	Network-to-Network Interface
OIF	Optical Internet Working Forum
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OVPNS	Optical Virtual Private Network Service
PCE	Path Computation Element
PCIM	Policy Common Information Model
PIB	Policy Information Base
PIT	Port Information Table
PM	Policy Manager
PPI	Provider Port Identifier
PXC	Photonic cross-Connect
QoS	Quality of Service
RM	Resource Manager
RMI	Remote Method Invocation
RPC	Remote Procedure Call
RSVP	Resource Reservation Protocol
SNMP	Simple Network Management Protocol
SOA	Service-Oriented Architecture

SOAP	Simple Object Accesss Protocol
SPC	Soft Permanent Connection
TE	Traffic Engineering
TMN	Telecommunications Management Network
TS	Trading Service
UDDI	Universal Description, Discovery, and Integration
UML	Unified Modeling Language
UNI	User-to-Network Interface
VPN	Virtual Private Network
WSBPEL	Web Services Business Process Execution Language
WSDL	Web Service Description Language
XML	Extensible Markup Language

Trabalhos Publicados Pelo Autor

Trabalhos Publicados Pelo Autor Diretamente Relacionados com a Tese

1. F. L. Verdi, E. Madeira, M. Magalhães e Annikki Welin. "The Virtual Topology Service: A Mechanism for QoS-enabled Interdomain Routing". *The 6th IEEE International Workshop on IP Operations & Management (IPOM 06), LNCS-Springer-Verlag*. Vol. 4268, pg. 205-217, Dublin, Ireland, October 2006.

2. F. L. Verdi, R. Pasquini, E. Madeira e M. Magalhães. "Web Services for the New Internet: Discussion and Evaluation of the Provisioning of Interdomain Services". *IEEE International Telecommunications Symposium (ITS'06)*, Fortaleza, Brasil, September 2006.

3. N. Malheiros, F. L. Verdi, E. Madeira e M. Magalhães. "Uma Arquitetura Baseada em Políticas para Gerência de VPNs de Camada 1". *Simpósio Brasileiro de Redes de Computadores (SBRC'06)*, Curitiba, Brasil, Maio 2006.

4. N. Malheiros, F. L. Verdi, E. Madeira e M. Magalhães. "A Management Architecture for Layer 1 VPN Services". *IEEE International Conference on Broadband Communications, Networks and Systems (Broadnets'06)*, San Jose, USA, October 2006.

5. C. Carvalho, F. L. Verdi, E. Madeira e M. Magalhães. "Gerência de Falhas baseada em Políticas para Redes Ópticas". *Simpósio Brasileiro de Redes de Computadores (SBRC'06)*, Curitiba, Brasil, Maio 2006.

6. F. L. Verdi, E. Madeira e M. Magalhães. "On the Performance of Interdomain Provisioning of Connections in Optical Networks using Web Services". *IEEE International Symposium on Computers and Communications (ISCC'06)*. Sardinia, Italy. June 2006.

7. F. L. Verdi, E. Madeira e M. Magalhães. "Web Services and SOA as Facilitators for ISPs". *International Conference on Telecommunications (ICT'06)*. Madeira Island, Portugal. May 2006.

8. F. L. Verdi, F. de Lacerda, R. Duarte, E. Madeira, E. Cardozo e M. Magalhães. "Provisioning and Management of Inter-Domain Connections in Optical Networks: A Service Oriented Architecture-based Approach.". *IEEE/IFIP Network Operations and Management Symposium (NOMS 2006)*. April 2006.

9. F. L. Verdi, C. Carvalho, E. Madeira e M. Magalhães. "Policy-based Grooming in Optical Networks". *4th IEEE Latin American Network Operations and Management Symposium (LANOMS 2005)*. pg. 125-136. August 2005.

10. C. Carvalho, F. L. Verdi, E. Madeira e M. Magalhães. "Policy-based Fault Management for Integrating IP over Optical Networks". *The 5th IEEE International Workshop on IP Operations & Management (IPOM'05), LNCS-Springer-Verlag*. Vol. 3751, pg. 88-97. October 2005.

11. F. L. Verdi, F. de Lacerda, R. Duarte, E. Madeira, E. Cardozo e M. Magalhães.. "Web Services-based Provisioning of Connections in GMPLS Optical Networks". *The Brazilian Symposium on Computer Networks (SBRC 2005)*. Maio 2005.

12. M. Siqueira, R. Pasquini, F. L. Verdi, R. Duarte, F. C. de Lacerda, E. Madeira and M. Magalhães. "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), Fortaleza, Ceará, Brasil, pg. 742-747, Maio 2005.

13. F. L. Verdi, E. Madeira e M. Magalhães. "Policy-based Admission Control in GMPLS Optical Networks". *"First IEEE Broadnets'04 (formerly OptiComm)*. pg. 337-339. October 2004.

Trabalhos Aceitos para Publicação

1. F. L. Verdi, E. Madeira, M. Magalhães, E. Cardozo and A. Welin. "A Service Oriented Architecture-based Approach for Interdomain Optical Network Services". *Aceito para publicação no Journal of Network and Systems Management (JNSM), Springer*, 2007.

2. F. L. Verdi, C. Carvalho, E. Madeira and M. Magalhães. "Policy-based Grooming in Optical Networks". *Aceito para publicação no Journal of Network and Systems Management (JNSM), Springer*, 2007.

Trabalhos Publicados Pelo Autor Parcialmente Relacionados com a Tese

1. R. Pasquini, F. L. Verdi, L. G. Zuliani e M. Magalhães. "An Optical UNI Architecture for the GIGA Project Testbed Network". *IEEE International Telecommunications Symposium (ITS'06)*, Fortaleza, Brasil, September 2006.

2. L. G. Zuliani, F. L. Verdi, R. Pasquini, G. Pavani, Márcio Savasini e M. Magalhães. "An Implementation of an OSPF-TE to Support GMPLS-controlled All-Optical WDM Networks". *IEEE International Telecommunications Symposium (ITS'06)*, Fortaleza, Brasil, September 2006.

Outros Trabalhos Publicados Pelo Autor

1. M. Siqueira, F. L. Verdi, R. Pasquini e M. Magalhães. "An Architecture for Autonomic Management of Ambient Networks". *International Conference on Autonomic Management and Services (IntellComm'06)*, LNCS-Springer. Paris, France, September 2006.

Capítulo 1

Introdução

Nos últimos anos, a tecnologia de redes ópticas tem ganhado atenção da comunidade científica devido a sua grande capacidade para transmissão de dados. Seu crescimento se deu, principalmente, na década de 90 e vem apresentando uma forte adesão de empresas de Telecom a fim de aumentarem suas capacidades de transporte de tráfego. A capacidade de transmissão de uma rede óptica depende basicamente da capacidade de transmissão de cada comprimento de onda (conhecido como *wavelength* ou *lambda*) e de quantos comprimentos de onda são colocados em cada fibra óptica. Atualmente a capacidade de transmissão de um comprimento de onda pode ser de 2.5Gb/s, 10Gb/s e mais recentemente 40Gb/s. A quantidade de comprimentos de onda em cada fibra pode chegar a 160Gb/s. Com isso pode-se obter valores da ordem de Terabytes de transmissão [de Maesschalck et al., 2003].

Com a evolução da tecnologia de redes ópticas, também evoluíram os sistemas e mecanismos para estabelecimento de circuitos ópticos (*lightpaths*). Os protocolos definidos para a arquitetura MPLS estão sendo estendidos de forma a suportar não somente a comutação de pacotes mas também a comutação de *slots* de tempo (*Time Division Multiplexing - TDM*), comutação por comprimento de onda (*Wavelength Division Multiplexing - WDM*) e comutação por fibra. O plano de controle do MPLS (*Multiprotocol Label Switching*) [Rosen et al., 2001] originou o *Generalized* MPLS (GMPLS) [Mannie, 2004] e um conjunto de protocolos vem sendo especificado a fim de atender as diversas tecnologias de comutação.

Enquanto o GMPLS é definido pelo IETF (*Internet Engineering Task Force*), o ITU-T (*International Telecommunication Union - Telecommunication Standardization Sector*) também possui um modelo arquitetural para estabelecimento de circuitos ópticos denominado *Automatically Switched Optical Network* (ASON) [ASON, 2001]. O modelo ASON define um conjunto de funcionalidades para o plano de controle a fim de suportar o fornecimento automático de conexões ópticas fim-a-fim. Entretanto, o modelo ASON apenas define funções abstratas necessárias para

esta automatização. Não são definidos protocolos, mas sim, mapeamentos das funcionalidades abstratas para protocolos existentes. Por outro lado, o GMPLS define e especifica protocolos para cada função que faz parte do provisionamento das conexões ópticas. Exemplos de protocolos que foram estendidos do MPLS para suportar o GMPLS incluem o *Resource ReserVation Protocol - Traffic Engineering* (RSVP-TE) para sinalização [Berger, 2003] e o *Open Shortest Path First - Traffic Engineering* (OSPF-TE) para roteamento [Kompella and Rekhter, 2005b]. Além disso, o GMPLS definiu um protocolo completamente novo para a gerência do enlace físico em redes ópticas conhecido como *Link Management Protocol* (LMP) [Lang, 2005].

A automação proposta pelas arquiteturas GMPLS e ASON é, sem dúvida, algo essencial para as empresas provedoras de serviços (*Internet Service Providers - ISPs*). Tal automação permite que o estabelecimento de conexões seja feito de forma rápida evitando, desta maneira, a intervenção humana e permitindo a oferta de novos serviços que atenda a demanda dos clientes. Porém, a grande dificuldade atual relacionada ao estabelecimento de conexões ocorre quando tais conexões precisam atravessar vários domínios administrativos diferentes. Enquanto as arquiteturas GMPLS e ASON apresentam soluções bem definidas e amplamente discutidas para serem usadas localmente por um domínio, as interações inter-domínios ainda se encontram em um estado bastante inicial de padronização. Definir soluções para provisionamento de serviços intra-domínios é tecnicamente mais fácil. O estabelecimento de conexões locais em um domínio pode ser feito utilizando soluções locais, proprietárias ou padronizadas. Localmente, cada administrador pode estabelecer conexões de forma manual ou da forma mais automatizada possível utilizando, por exemplo, GMPLS ou ASON. Porém, o estabelecimento de conexões que atravessam vários domínios não somente envolve aspectos técnicos mas, também, aspectos de negócios que caracterizam os vários tipos de relacionamentos existentes entre os domínios.

Esta tese é uma alternativa em relação à forma como o estabelecimento de serviços entre domínios vem sendo proposta. Enquanto os órgãos de padronização apostam em extensões nos protocolos atuais para suportar o estabelecimento de conexões ópticas entre domínios, entendemos que tais extensões não serão colocadas em prática a curto ou talvez a médio prazo. Tais extensões visam principalmente suportar o estabelecimento de serviços (tipicamente conexões) levando-se em consideração aspectos de QoS (*Quality Of Service*). O principal problema neste tipo de cenário está relacionado com as extensões que precisam ser feitas ao protocolo de roteamento entre domínios, especificamente o BGP (*Border Gateway Protocol*) [Rekhter and Li, 2006], cuja funcionalidade principal é a divulgação de alcançabilidade entre prefixos de rede.

O OIF (*Optical Internet Working Forum*) [OIF, 2006] tem atuado na definição de uma interface para a troca de mensagens entre domínios para o estabelecimento de conexões. A *External Network-to-Network Interface* (E-NNI) [E-NNI, 2004] permite que informações de controle sejam

trocadas entre diferentes domínios de forma que a sinalização de uma conexão iniciada em um domínio possa atravessar uma cadeia de domínios até chegar no destino. Embora esta especificação esteja sendo feita, vários aspectos relacionados ao mapeamento das funcionalidades abstratas em protocolos específicos ainda estão em um estado inicial. Atualmente, a especificação E-NNI mapeia tais funcionalidades nos protocolos RSVP-TE e CR-LDP (*Constraint-Based Label Distribution Protocol*). Porém, nada tem sido feito neste aspecto pelo IETF.

A proposta apresentada nesta tese é justamente uma alternativa às extensões necessárias aos protocolos usados no plano de controle. As relações entre domínios são mais facilmente executadas no plano de serviços onde cada domínio pode oferecer serviços específicos para entidades externas (clientes e outros domínios). Desta forma, cada domínio passa a ser visto como um conjunto de serviços que podem ser invocados para realizarem tarefas relacionadas ao estabelecimento de conexões entre domínios sem depender das extensões aos protocolos.

A proposta deste trabalho tem forte relação com a abordagem apresentada pelo projeto CANARIE [CANARIE Project, 2006]. O projeto CANARIE apresenta uma proposta para estabelecimento de conexões entre domínios onde as funcionalidades tradicionais do plano de controle passam a ser realizadas na forma de serviços. O modelo tradicional baseado em protocolos tais como o RSVP é substituído por um modelo onde as interações entre domínios para sinalização e roteamento ocorrem no plano de serviços. A definição destes serviços é então facilitada uma vez que cada domínio é responsável por oferecer serviços para outros domínios sem depender de extensões aos protocolos. A abordagem do projeto CANARIE também considera que os usuários possuem controle total sobre a criação e remoção de caminhos de luz. Entretanto, a proposta apresentada neste trabalho não adota esta abordagem pela qual o cliente possui tal controle. Nesta tese, os provedores de serviços são responsáveis pelo controle da criação e remoção de conexões. O usuário cliente apenas solicita o serviço. A gerência e o controle do provisionamento dos serviços é responsabilidade do provedor.

Embora o foco da tese seja o provisionamento de serviços entre domínios, dividimos a arquitetura em duas partes. Primeiramente focamos nos aspectos relacionados ao provisionamento e gerência de serviços dentro de um domínio (intra-domínio). Na segunda parte, focamos no provisionamento e gerência de serviços entre domínios. Esta divisão facilitou o desenvolvimento dos módulos de forma que inicialmente apenas funcionalidades para o provisionamento de serviços dentro de um domínio fossem consideradas. Após, funcionalidades relacionadas ao provisionamento de serviços entre domínios foram incorporadas aos módulos. Definimos um modelo baseado no modelo de referência TMN (*Telecommunications Management Network*) [TMN, 1985] que incorpora as funcionalidades FCAPS (*Fault, Configuration, Accounting, Performance, Security*) definidas pela ISO. O modelo proposto serve como base para a definição das camadas para o fornecimento dos serviços tanto intra como inter-domínios. O modelo de referência TMN define quatro camadas de gerência: a camada

de gerência do elemento de rede, a camada de gerência de rede, a camada de gerência de serviços e a camada de gerência de negócios. O modelo proposto foi então instanciado primeiramente para o provisionamento de serviços dentro de um domínio e, após, o modelo proposto foi instanciado para o provisionamento de serviços entre domínios.

Na arquitetura para provisionamento de serviços internos ao domínio, focamos no provisionamento de dois serviços: o serviço de conexões ópticas e o serviço de VPNs[1]. Estes dois serviços são considerados os serviços básicos e principais a serem oferecidos pelos provedores aos seus clientes.

A arquitetura elaborada para o provisionamento de serviços dentro de um domínio é dividida em três trabalhos que deram origem a três dissertações de mestrado. O primeiro trabalho desenvolveu a arquitetura e seus módulos [Duarte, 2006, Verdi et al., 2005b, Verdi et al., 2006a]. A arquitetura contemplava o estabelecimento de conexões dentro de um domínio e a definição das interfaces para interação entre plano de gerência e plano de controle. O segundo trabalho teve como motivação a grande quantidade de banda existente em cada *lightpath*. Devido a isso, a agregação de tráfego cliente dentro dos caminhos de luz[2] é vista como um mecanismo que permite um maior aproveitamento de tal banda. Esta agregação é conhecida como *traffic grooming* [Zhu and Mukherjee, 2002] e é normalmente realizada nos dispositivos de borda dos domínios ópticos. Neste segundo trabalho, foram definidas políticas para realizar agregação de tráfego IP/MPLS nos caminhos de luz de forma a maximizar o uso dos recursos ópticos nos domínios [Verdi et al., 2004, Verdi et al., 2005a]. Além disso, políticas para minimizar o impacto de falhas no domínio óptico [Carvalho, 2006, Carvalho et al., 2005, Carvalho et al., 2006, Verdi et al., 2007a] foram elaboradas. Finalmente, o terceiro trabalho de mestrado considerou o provisionamento do serviço de VPNs dentro de um domínio. O provisionamento de VPNs em um único domínio tem recebido grande atenção da comunidade científica nos grupos de trabalho do IETF [Ould-Brahim, H. and Rekhter, Y., 2005], ITU-T [ITU-T, 2003], e em vários artigos encontrados na literatura atual [Takeda et al., 2005, Takeda et al., 2004, French and Pendarakis, 2004]. Neste terceiro trabalho, adicionamos funcionalidades aos módulos da arquitetura para suportar o provisionamento do serviço de VPN em domínios ópticos [Malheiros, 2006, Malheiros et al., 2006a, Malheiros et al., 2006b].

O provisionamento de serviços entre domínios [Verdi et al., 2007b] também consiste em estabelecer conexões e VPNs ópticas entre diferentes domínios administrativos. Atualmente, existe uma grande necessidade por parte dos provedores em oferecer serviços que atravessam vários e diferentes domínios administrativos. O serviço de conexões entre domínios é o primeiro passo

[1]Nesta tese, o termo VPN se refere às VPNs ópticas. As VPNs ópticas são um tipo específico de VPNs de camada 1 (Layer 1 VPN-L1VPN). Assim, o termo L1VPN, quando usado, também se refere unicamente às VPNs ópticas.

[2]Os termos caminho de luz e *lightpath* possuem o mesmo significado e serão usados sem distinção nesta tese.

para suportar outros serviços mais complexos. Ao mesmo tempo, enquanto o provisionamento de VPNs dentro de um domínio tem recebido grande atenção, o próprio grupo de trabalho do IETF menciona que os aspectos relacionados ao provisionamento de VPNs entre domínios na camada 1 serão discutidos futuramente [L1Charter, 2006].

A arquitetura desenvolvida neste trabalho facilita a forma como os domínios interagem uma vez que cada domínio expõe suas capacidades na forma de interfaces. Uma conseqüência imediata desta solução é o rápido desenvolvimento de serviços sem precisar esperar pela definição de padrões que especifiquem como as interações entre domínios deveriam ocorrer (como acontece com a E-NNI). A arquitetura apresentada nesta tese vai ao encontro de uma tendência atual que se caracteriza por abstrair os detalhes de como os serviços são oferecidos.

Nesta tese identificamos alguns dos serviços que são necessários para o fornecimento de conexões e VPNs entre domínios ópticos. Além disso, empregamos um conceito conhecido como topologia virtual (*Virtual Topology - VT*) que abstrai os detalhes físicos internos de cada domínio óptico. Assim, aspectos relacionados à topologia física de cada domínio óptico são preservados. As VTs são formadas por caminhos ópticos que atravessam o domínio e possuem características que representam a oferta de determinados serviços. Cada domínio óptico define sua VT conforme as capacidades físicas locais seguindo políticas de relacionamento com seus pares.

O mecanismo de VTs serve como base para a migração de um cenário intra-domínios para cenários entre domínios. É através do mecanismo de topologias virtuais que as relações entre domínios são definidas uma vez que tais topologias servem como base para a criação de outros serviços. Uma VT pode representar o interesse momentâneo do domínio óptico em atrair conexões. A idéia principal da VT é criar na camada de serviços um modelo de oferta e procura por conexões que atendam os requisitos pelo menor preço. Como cada domínio anuncia uma VT, domínios que necessitam estabelecer uma conexão fim-a-fim entre domínios poderão analisar qual o melhor caminho para alcançar o destino. O conceito de VT foi implementado através do serviço de divulgação (*Advertising Service - AS*). Tal serviço, além de divulgar as topologias virtuais, também distribui as informações de correlação de portas das VPNs.

A implementação da arquitetura foi realizada usando *Web Services*, uma tecnologia atual e que vem sendo utilizada como uma forma de instanciar a arquitetura *Service Oriented Architecture* (SOA). O modelo SOA caracteriza-se principalmente pela interação entre serviços através do fraco acoplamento e age como um facilitador para a definição dos serviços e interações de negócios entre os domínios. Nesta tese, todos os serviços que fazem parte da camada de serviços foram implementados como *Web Services*. Além disso, os processos de negócios e as atividades necessárias para o estabelecimento de serviços entre domínios foram modelados usando a notação BPMN (*Business Process Modeling Notation*) [BPMN, 2006]. O principal objetivo da BPMN é permitir a criação

de diagramas que sejam compreensíveis para os analistas de negócios e para os desenvolvedores técnicos. A notação BPMN é vista como uma alternativa para a modelagem em alto nível como sendo um passo anterior às especificações que se aproximam muito dos detalhes de implementação, tal como a linguagem WSBPEL (*Web Services Business Process Execution Language*) [OASIS, 2006]. A notação BPMN define um modelo abstrato e uma gramática para especificação de processos genéricos podendo ser usada em aplicações de colaborações multi-organizacionais e composição de serviços *Web*. Outra vantagem da BPMN é permitir que os processos em alto nível sejam mapeados para linguagens de orquestração e coreografia a critério do desenvolvedor. Atualmente existem ferramentas que, a partir de um diagrama BPMN, geram código em BPEL [eclarus, 2006].

Além dos módulos que compõem a camada de serviços, definimos também módulos responsáveis pela gerência local em cada domínio. Basicamente foram definidos módulos para controle de admissão, aplicação de políticas, controle de falhas locais e gerência de recursos.

O protótipo desenvolvido para validar a arquitetura aqui descrita tem como objetivo analisar o uso dos *Web Services* como uma tecnologia viável para o tipo de cenário considerado neste trabalho. A utilização de *Web Services* para provisionamento de conexões em redes ópticas também vem sendo considerada pelo projeto CANARIE no Canadá [Boutaba et al., 2004], e tem se mostrado factível e apropriada para o modelo adotado por aquele projeto que, no entanto, difere do modelo adotado nesta tese. Naquele projeto, não há o conceito de topologia virtual. Além disso, tal projeto considera que o controle dos caminhos de luz passa a ser feito pelo usuário. Nesta tese, como já mencionado, consideramos um modelo mais tradicional onde o controle e a gerência dos recursos ficam sob responsabilidade de cada domínio.

Os dois modelos (TMN e FCAPS) quando usados de forma conjunta oferecem um ferramental bastante completo para a gerência de redes. Não é nossa intenção usarmos o modelo TMN em sua totalidade uma vez que o modelo apresenta uma arquitetura funcional, de informação e física bastante completa. As interfaces do modelo TMN são usadas apenas como referência em relação aos pontos de acesso entre as diferentes camadas. Além disso, a rigidez do modelo TMN está fazendo com que o padrão XML seja cada vez mais utilizado para o desenvolvimento de soluções de gerência interoperáveis e troca de informações. Porém, o modelo TMN, com sua arquitetura em forma de pirâmide e suas interfaces, serve como base para a definição da nossa arquitetura para provisionamento e gerência de serviços em redes ópticas.

Apresentamos a arquitetura primeiramente ilustrando os aspectos relacionados ao provisionamento e gerência dos serviços dentro de um domínio. Após isto, apresentamos o desenvolvimento da arquitetura para o provisionamento de serviços entre domínios. Como a fase responsável pelo provisionamento de serviços em um domínio deu origem a três dissertações de mestrado e, portanto, todos os detalhes são encontrados nos artigos que serão referenciados e em cada

dissertação, nesta tese fornecemos apenas as informações sobre cada trabalho que são necessárias para a compreensão da arquitetura. Porém, o foco desta tese será dado ao fornecimento de serviços entre domínios.

1.1 Motivações para esta Tese

As principais motivações para esta tese são:

- Falta de um plano de controle entre domínios que seja padronizado. Atualmente tal padronização depende de extensões de alguns protocolos;
- O projeto CANARIE e o uso de *Web Services* como plataforma de serviços;
- O suporte à QoS entre domínios ainda não possui soluções bem definidas;
- O crescimento do uso de tecnologias de redes ópticas como solução para os limites atuais de banda.

1.2 Principais Objetivos desta Tese

Dentre os objetivos deste projeto, os principais são:

- Desenvolver uma arquitetura para provisionamento e gerência de serviços em redes ópticas. A arquitetura deve suportar o provisionamento de serviços intra e inter-domínios;
- Facilitar a interação entre domínios de forma que a sinalização de conexões ópticas seja feita no plano de serviços através de protocolos de negociação e reserva de recursos. Entendemos que isto facilita a interação entre domínios e evita a espera pelo processo de padronização de interfaces de sinalização entre domínios, atualmente sendo especificadas por organizações internacionais tais como o IETF e OIF;
- Fazer com que cada domínio óptico seja visto como um serviço. Cada domínio deve oferecer um conjunto de serviços em forma de interfaces flexibilizando as interações cliente/provedor e provedor/provedor [3];

[3] Na linguagem da Internet, um provedor pode ser responsável por vários domínios. Para oferecer um serviço, o provedor pode precisar atravessar vários domínios. Um domínio representa alguma divisão lógica (as vezes física) seguindo algum critério. Por exemplo, um domínio pode ser dividido por tecnologias ou marcas de roteadores. Um Sistema Autônomo é visto como um domínio administrativo independente. Nesta tese, o provedor e o domínio são a mesma entidade que oferece e suporta os serviços oferecidos para os clientes ou para outros provedores/domínios.

- Permitir que cada domínio seja independente, podendo aplicar suas políticas locais levando-se em consideração aspectos administrativos e o uso dos recursos;
- Adotar a tecnologia *Web Services* como sendo uma instância do modelo SOA para implementar a arquitetura. Testar a arquitetura a fim de validar e analisar as vantagens, desvantagens e o impacto em termos de tempo e consumo de banda para o provisionamento dos serviços usando *Web Services*.

1.3 Organização da Tese

No próximo capítulo apresentamos os conceitos básicos necessários para a compreensão deste trabalho e alguns trabalhos relacionados com a proposta desta tese. No Capítulo 3 apresentamos o modelo proposto para este trabalho. Mostramos como o modelo de referência TMN/FCAPS foi utilizado para definirmos o modelo para gerência e provisionamento de serviços em redes ópticas. Após, no Capítulo 4, detalhamos o mecanismo de topologias virtuais. Discutimos suas vantagens e como o mecanismo pode ser utilizado para abstração dos detalhes internos ao domínio óptico e facilitar o fornecimento de serviços entre domínios. No Capítulo 5, detalhamos a instanciação do modelo proposto especificamente para a arquitetura para o provisionamento de serviços em um domínio. A seguir, no Capítulo 6, apresentamos a instanciação do modelo para o provisionamento de serviços entre domínios. Apresentamos os módulos internos da arquitetura, os módulos que compõem a camada de serviços e a modelagem dos processos de negócios. No Capítulo 7 focamos nos aspectos relacionados com a implementação, testes, validação e obtenção de resultados. Mostramos como o protótipo implementado se comporta e qual o impacto do uso dos *Web Services* nos cenários considerados. Finalmente, no Capítulo 8 concluímos este trabalho discutindo as contribuições desta tese, os objetivos alcançados e os trabalhos futuros.

Capítulo 2

Conceitos Básicos e Trabalhos Relacionados

Neste capítulo, apresentamos primeiramente os conceitos básicos necessários para a compreensão deste trabalho. Tais conceitos incluem uma apresentação sobre a tecnologia *Web Services*, o modelo TMN, a notação BPMN e redes ópticas (ASON e GMPLS). Na segunda seção deste capítulo, discutiremos alguns trabalhos que estão direta ou parcialmente relacionados com a proposta desta tese.

2.1 Conceitos Básicos

2.1.1 *Web Services*

A tecnologia *Web Services* é uma forma específica de tecnologia XML (*Extensible Markup Language*) orientada à Internet. Desenvolvida e padronizada pelo W3C (*World Wide Web Consortium*), os *Web Services* têm sido utilizados para o desenvolvimento de sistemas de gerenciamento de redes. Os trabalhos e pesquisas realizados em [Thurm, 2002], [Pras et al., 2004] e [Pavlou et al., 2004] são alguns exemplos que comprovam sua aceitação e utilização no meio científico.

Em 2000, o W3C aceitou uma submissão para o SOAP (*Simple Object Access Protocol*). Esta especificação definia um formato de mensagem baseado em XML que poderia ser usado para transmitir informações entre aplicações distribuídas através do protocolo HTTP. Por ser uma tecnologia sem características proprietárias, o SOAP tornou-se uma alternativa atrativa aos protocolos tradicionais, tais como CORBA (*Common Object Request Broker Architecture*) e DCOM (*Distributed Component Object Model*). Durante o ano seguinte, o W3C publicou a especificação do WSDL (*Web Service Description Language*). O WSDL é um padrão que fornece uma linguagem baseada em XML para a descrição da interface dos *Web Services*. E, finalmente, com a introdução da especificação do

UDDI (*Universal Description, Discovery, and Integration*) que fornece um mecanismo padrão para a descoberta dinâmica de descrições de serviços, a primeira geração da plataforma *Web Services* foi estabelecida [Erl, 2004a]. A Figura 2.1 mostra em alto nível o relacionamento entre estes padrões.

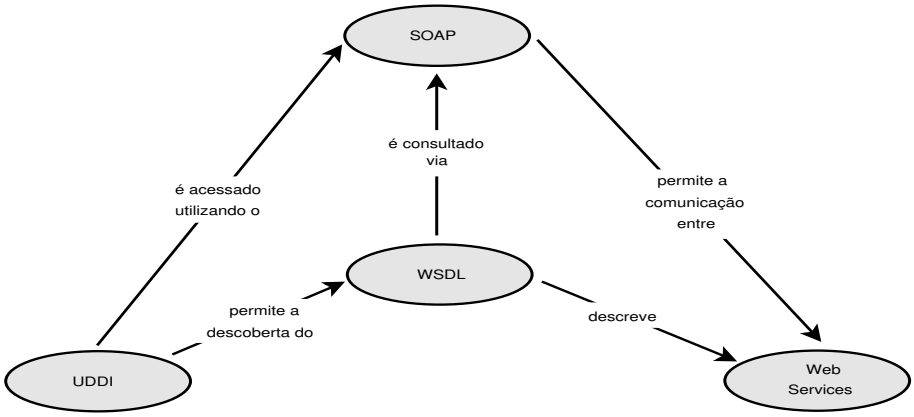


Fig. 2.1: Relacionamento entre as especificações de primeira geração [Erl, 2004a].

Um conceito anterior aos *Web Services* que merece uma breve discussão é a arquitetura orientada a serviços (SOA - *Service-Oriented Architecture*) que propõe um mecanismo de interação totalmente interoperável entre unidades funcionais modulares (serviços). Atualmente, a tecnologia *Web Services* é uma forma (talvez a mais conhecida e usada) de instanciar a arquitetura orientada a serviços. Porém, com um certo esforço, tecnologias mais antigas como DCOM e CORBA também podem implementar a SOA.

A arquitetura orientada a serviços (SOA) define um estilo cujo objetivo é viabilizar o fraco acoplamento entre agentes de *software*. Os agentes de *software* são componentes da arquitetura SOA que realizam operações de publicação, procura e execução de serviços. Um serviço é a implementação modularizada de uma função específica que pode ser invocada através da rede. Todo serviço possui uma interface que define as suas funcionalidades visíveis para o mundo externo e os meios para acessar estas funcionalidades.

Toda arquitetura SOA deve possuir três componentes básicos indicados a seguir:

- Provedor de serviços: responsável pela criação e publicação das interfaces dos serviços além de prover a implementação real dos serviços para responder qualquer requisição de uso;
- Registro: este componente registra e categoriza serviços públicos publicados por vários provedores de serviços. O Registro também oferece serviços de busca. Estes serviços funcionam como páginas amarelas permitindo a busca por serviços classificados por categorização;
- Requerente de serviços: este componente representa o usuário dos serviços. O requerente descobre a localização dos serviços procurando em repositórios mantidos pelos registros. Após

encontrar os serviços, o requerente comunica com os provedores através da invocação destes serviços.

Modelo *Web Services*

O modelo *Web Services* é caracterizado pelos papéis e operações envolvidos no funcionamento de um *Web Service*. Os papéis são diferentes tipos de entidades que compõem o modelo e as operações são as funções realizadas por estas entidades [Hendricks et al., 2002]. A Figura 2.2 mostra o modelo *Web services* formado por três tipos de papéis e as operações que eles executam.

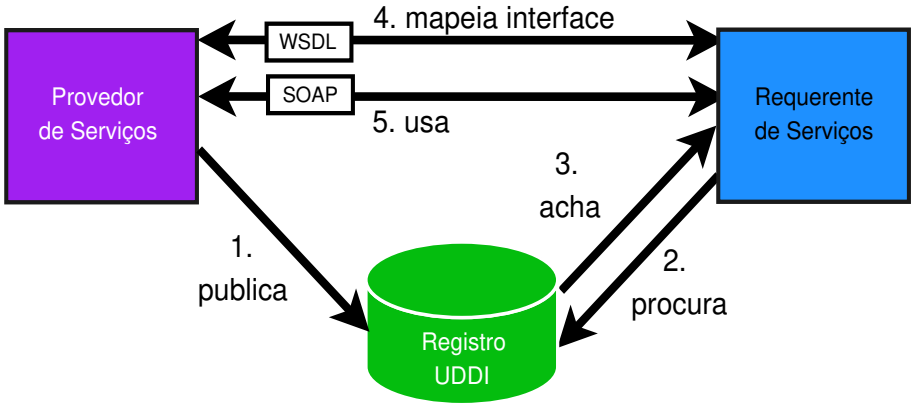


Fig. 2.2: Modelo Web Services.

Os papéis do modelo *Web Services* realizam as mesmas operações dos componentes que formam a SOA. O modelo *Web Service* visto acima é formado por três operações fundamentais: "publica", "procura" e "mapeia/executa". Para alcançar a comunicação entre aplicações (serviços) sem as restrições sobre qual linguagem o serviço está escrito e sobre qual plataforma o serviço está rodando, é necessário a utilização de padrões para cada uma destas três operações e um formato padrão para o provedor de serviços descrever os seus *Web Services*. Os padrões utilizados são os seguintes [Hendricks et al., 2002]:

- **WSDL (*Web Service Description Language*)**: O WSDL [Christensen et al., 2001] é um documento baseado em XML utilizado na descrição dos *Web Services*. Basicamente, o WSDL define os métodos que estão presentes no *Web Service*, os parâmetros de entrada e saída para cada um dos métodos, os tipos de dados, o protocolo de transporte utilizado e o *endpoint* (localização) do *Web Service*. O WSDL é um vocabulário XML que pode ser utilizado para descrever *Web Services* em um formato neutro de plataforma e linguagem de programação. Um documento WSDL representa a interface externa para um *Web service*. O WSDL é para o *Web Service* o que o IDL (*Interface Definition Language*) é para o mundo CORBA;

- **UDDI (*Universal Description, Discovery, and Integration*)**: O UDDI [Hately et al., 2005] define um protocolo para publicação e procura de serviços. A publicação é utilizada por provedores de serviços para publicar em um diretório de serviços os detalhes sobre sua organização e os *Web Services* que eles oferecem. A procura de serviços é utilizada por requerentes de serviços para descobrir serviços localizados em provedores de serviços. O UDDI é uma especificação para criar um serviço de registro que cataloga organizações e seus *Web Services*. Uma implementação da especificação UDDI é chamada de Registro UDDI e corresponde a uma base de dados que fornece um conjunto de estruturas de dados padrões definidas pela especificação UDDI. As estruturas de dados modelam informações sobre organizações e os requisitos técnicos para acessar os *Web Services* mantidos por estas organizações. Através de um registro UDDI é possível fazer buscas sobre tipos específicos de empresas ou *Web Services*. Em relação ao registro UDDI, pode ser feita uma analogia comparando-o a um sistema eletrônico de "Páginas Amarelas" pelo qual podem ser feitas procuras por organizações e tipos específicos de *Web Services*;

- **SOAP (*Simple Object Access Protocol*)**: O SOAP [W3C, 2005] é um protocolo baseado em XML utilizado na troca de informações entre *Web Services* sem levar em consideração detalhes sobre o sistema operacional e linguagem de programação. O SOAP comunica com sistemas distribuídos utilizando a linguagem XML baseada em texto ao invés do formato binário utilizado por outros protocolos de computação distribuída, tais como CORBA, RMI (*Remote Method Invocation*) e DCOM. Isto torna o SOAP altamente interoperável através de plataformas de *hardware*, sistemas operacionais e linguagens de programação. O SOAP pode ser transportado sobre o HTTP e, conseqüentemente, beneficiar-se da infra-estrutura criada para o HTTP como servidores *Web*, servidores *proxy* e *firewalls*.

  A mensagem SOAP é um tipo de documento XML formado por um envelope constituído de duas partes: cabeçalho (*header*) e o corpo da mensagem (*body*). A Figura 2.3 mostra a representação gráfica da mensagem SOAP.

  O envelope SOAP é o elemento raiz da mensagem SOAP e contém um cabeçalho SOAP opcional e um corpo SOAP obrigatório. O envelope SOAP é denotado em uma mensagem SOAP pelo elemento <*Envelope*>. O cabeçalho SOAP é uma maneira genérica e flexível de adicionar características a uma mensagem SOAP como autenticação, transação, assinatura digital, etc. Um cabeçalho é especificado em uma mensagem SOAP pelo elemento <*Header*>. O corpo SOAP é a área da mensagem SOAP que contém as informações a serem trocadas entre aplicações através da mensagem. As informações da mensagem devem estar no elemento <*Body*> da mensagem SOAP.

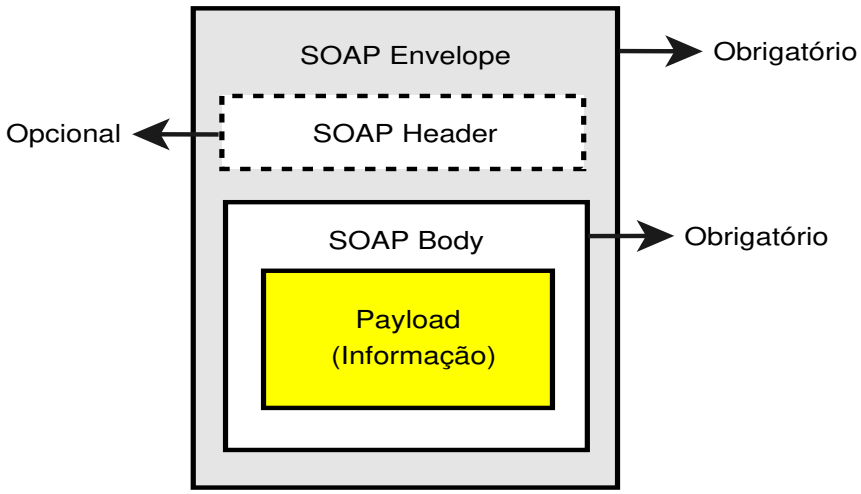


Fig. 2.3: Conceitualização da mensagem SOAP.

Os estilos de comunicação SOAP referem-se ao formato da mensagem SOAP trocada entre aplicações *Web Services*. O estilo de comunicação a ser utilizado entre um cliente (aplicação ou *Web Service*) e um *Web Service* vai depender da forma como o *Web Sevice* está implementado. Os *Web Services* oferecem dois tipos de modelos para que clientes o invoquem: estilo RPC (*Remote Procedure Call*) e estilo *Document*.

O estilo RPC permite modelar chamadas de métodos com parâmetros e receber valores de retorno. Neste estilo, uma mensagem SOAP leva em seu corpo (elemento *Body*) o nome do método a ser executado e os parâmetros de entrada da chamada. Na mensagem RPC de resposta, um valor de retorno (ou uma falha) do método invocado é retornado. No estilo de comunicação *Document*, o corpo da mensagem SOAP contém um fragmento de documento XML que será enviado ao *Web service* ao invés de um conjunto de valores de parâmetros.

O tipo de estilo de comunicação SOAP implementado nos *Web Services* influi diretamente no seu desempenho e confiabilidade. Ao contrário da implementação de *Web Services* estilo RPC que necessita somente da elaboração da interface de suas operações, o estilo *Document* exige maior esforço de implementação pois requer a criação de um XML *Schema* com a descrição dos elementos que correspondam às operações do *Web Service* e os tipos de dados utilizados nos parâmetros destas operações. O estilo *Document* se caracteriza por possuir um contrato menos rígido e permite que modificações sejam feitas em seu XML *Schema* sem comprometer a chamada de aplicações clientes. O desempenho do estilo RPC é inferior ao desempenho do estilo *Document*. Isto ocorre devido ao fato de que o tamanho das mensagens SOAP trocadas entre aplicações *Web Services* estilo RPC são maiores em relação ao tamanho das mensagens SOAP trocadas entre aplicações *Web Services* estilo *Document*, aumentando significativamente o tempo de processamento. Uma análise comparativa entre os dois estilos é apresentada no

Capítulo 7 a fim de avaliar tais tempos de processamento em cenários para provisionamento de serviços entre domínios de redes ópticas.

2.1.2 O Modelo TMN

O modelo TMN (*Telecommunications Management Network*) foi definido pelo ITU-T na recomendação M.3000 [TMN, 1985]. O modelo TMN divide a gerência em quatro camadas, como podemos ver na Figura 2.4.

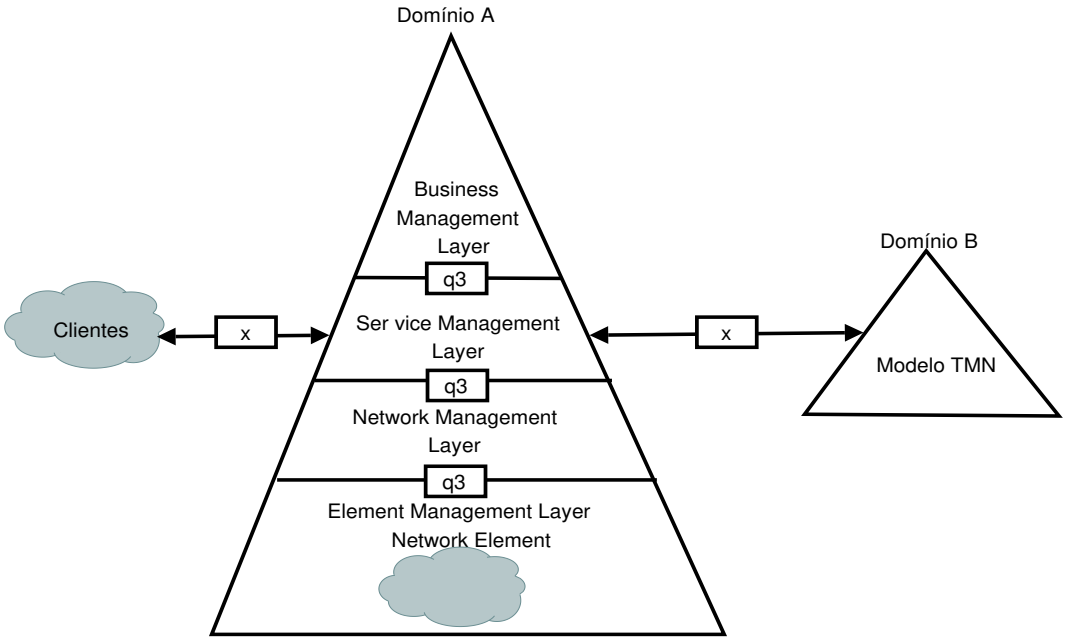


Fig. 2.4: O Modelo arquitetural do TMN.

A camada de gerência de negócios (*Business Management Layer*) realiza funções relacionadas aos aspectos de negócios e analisa tendências, políticas administrativas, lucratividade, contratos, etc. A camada de gerência de serviços (*Services Management Layer*) realiza as funções para a gerência e provisionamento de serviços na rede. A camada de serviços usa informações da camada de gerência de redes para gerenciar os serviços contratados. A camada de serviços é também responsável pela interação com clientes e outros domínios administrativos. A camada de gerência de redes (*Network Management Layer*) possui a visibilidade da rede através da obtenção de informações dos elementos de rede e dos serviços estabelecidos. A camada de gerência de redes gerencia cada elemento de rede individualmente ou todos os elementos como um grupo. Além disso, a camada de gerência de redes deve atender as demandas da camada de serviços e implementá-las na camada de rede. Por fim, a camada dos elementos de rede (*Element Management Layer*) possui informações gerenciáveis em cada elemento de rede. A camada de gerência dos elementos de rede possui funções para

tratamento de alarmes, manipulação de informações, *backup*, *logs*, etc. A camada de elementos de rede representa efetivamente os elementos físicos da rede gerenciada.

A Figura 2.4 também mostra as interfaces definidas pelo modelo TMN. Tais interfaces são usadas para comunicação entre os componentes localizados na mesma camada ou em camadas diferentes do modelo TMN. A interface *q* é responsável pelo interfaceamento entre os elementos funcionais dentro de um mesmo domínio TMN. Ela está localizada entre a camada de negócio e a camada de serviços, entre a camada de serviços e a camada de gerência de redes, e entre a camada de gerência de redes e a camada dos elementos de rede. A interface *x* realiza o interfaceamento entre dois modelos TMN localizados em domínios diferentes. Ela está localizada entre os clientes e a camada de serviços e entre a camada de serviços de outros domínios administrativos.

Do ponto de vista funcional, o modelo TMN incorporou um conjunto de funcionalidades de gerência que se relacionam com as 4 camadas do modelo. Estas funcionalidades são conhecidas como FCAPS (*Fault, Configuration, Accounting, Performance and Security*) [Hamada et al., 2001] e são brevemente explicadas abaixo.

1. Falha
   O gerenciamento de falhas deve ser capaz de detectar, isolar e alertar as falhas além de corrigi-las;

2. Configuração
   O gerenciamento de configuração refere-se ao monitoramento e controle de operações definidas por configuração. Estas operações são executadas com a freqüência estabelecida em sua configuração. As operações podem ser, por exemplo, o estabelecimento e remoção de conexões e VPNs;

3. Contabilização
   A gerência de contabilidade determina a distribuição adequada dos recursos entre usuários de um provedor de serviços. Isto ajuda a minimizar o custo da operação através do uso mais efetivo dos sistemas disponíveis. Este nível também é responsável para assegurar a tarifação apropriada dos usuários;

4. Desempenho
   O gerenciamento de desempenho monitora o desempenho total das redes. Os problemas potenciais e os gargalos são identificados e o *throughput* é maximizado. As melhorias que renderão os maiores benefícios ao desempenho total são identificadas;

5. Segurança
   O gerenciamento de segurança é responsável pela proteção da rede de usuários não autorizados

e de ações mal intencionadas. Além disso, é responsável pela autenticação e autorização de usuários. Desta forma, mantém-se a confidencialidade da informação de cada usuário.

2.1.3 A Notação BPMN

A notação BPMN define um diagrama de processo de negócios baseada na composição de modelos gráficos que quando colocados em seqüência caracterizam um processo de negócios. Um processo de negócios pode então ser definido como uma rede de objetos gráficos formada por atividades e controles de fluxos que definem a ordem de acontecimento destas atividades. Nesta seção iremos apresentar apenas os elementos da notação BPMN que foram usados neste trabalho. Mais informações sobre a notação podem ser obtidas em [BPMN, 2006]. A notação BPMN é dividida em quatro categorias de elementos:

- Objetos de Fluxo;
- Objetos de Conexão;
- *Swimlanes*; e
- Artefatos.

No final da explicação de todos os elementos, apresentamos um exemplo simples que envolve a maioria dos elementos mencionados.

Objetos de Fluxo

Os objetos de fluxo são divididos em eventos, atividades e *gateways*. Os eventos são representados por círculos e podem caracterizar eventos iniciais, eventos intermediários ou eventos finais. Uma atividade é representada por um retângulo com cantos arredondados. Uma atividade pode representar uma tarefa ou um sub-processo. Um sub-processo possui um pequeno sinal de "mais" no centro inferior do retângulo. Um sub-processo significa que a atividade possui um nível de detalhe que não é mostrado no diagrama. Outro diagrama pode expandir especificamente o sub-processo mostrando os detalhes internos da atividade. Uma atividade também pode ser repetida várias vezes. Esta repetição é representada através de um sinal de *looping* no centro inferior do retângulo. A atividade também pode ser instanciada várias vezes (múltiplas instâncias). O símbolo de múltiplas instâncias é representado por duas pequenas barras paralelas no sentido vertical no centro inferior do retângulo. Finalmente, os *gateways* são representados na forma de um quadrado inclinado. São usados para controlar divergências e convergências na seqüência de fluxos. A Figura 2.5 mostra os três tipos de objetos de fluxos e suas variações.

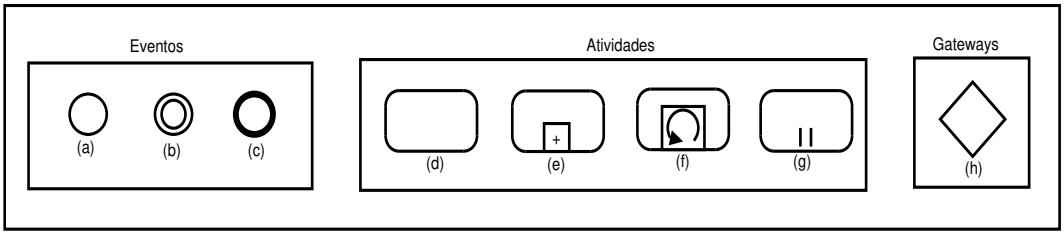


Fig. 2.5: Tipos de objetos de fluxo. Eventos: (a) evento inicial, (b) evento intermediário, (c) evento final. Atividades: (d) tarefa, (e) sub-processo, (f) *looping*, (g) múltiplas instâncias. *Gateways*: (h).

Objetos de Conexão

Os objetos de conexão são utilizados em um diagrama para criar a estrutura básica de um processo de negócios. A notação BPMN define três tipos de objetos de conexão: fluxo de seqüência, fluxo de mensagens e associação. Um fluxo de seqüência é usado para mostrar a seqüência em que as atividades serão realizadas dentro de um mesmo processo. Um fluxo de mensagens é usado para representar os fluxos de mensagens entre dois processos participantes. Uma associação é usada para associar dados, textos e outros artefatos com objetos de fluxo. A Figura 2.6 mostra a representação gráfica dos três tipos de objetos de conexão.

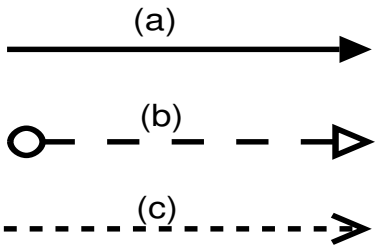


Fig. 2.6: Tipos de objetos de conexão. (a) fluxo de seqüência, (b) fluxo de mensagens, (c) associação.

Swimlanes

As *swimlanes* são mecanismos para organizar atividades em categorias visuais para demonstrar separadamente as capacidades funcionais de cada categoria. Existem dois tipos de *swimlanes*: *Pool* e *Lane*. Uma *Pool* representa um participante em um processo e acomoda graficamente suas atividades separando-as das atividades de outros participantes. Uma *Lane* é uma sub-partição dentro de uma *Pool*. As *Lanes* são usadas para organizar e categorizar as atividades dentro de um participante. A Figura 2.7 mostra graficamente a representação de *Pools* e *Lanes*.

A figura mostra uma *Pool* vazia (*Pool1*) e uma *Pool* com duas *Lanes* (*Pool2*). *Pools* são usadas quando o diagrama BPMN envolve dois ou mais participantes e são fisicamente separadas no diagrama. A comunicação entre duas *Pools* não pode ser feita usando o objeto de conexão fluxo

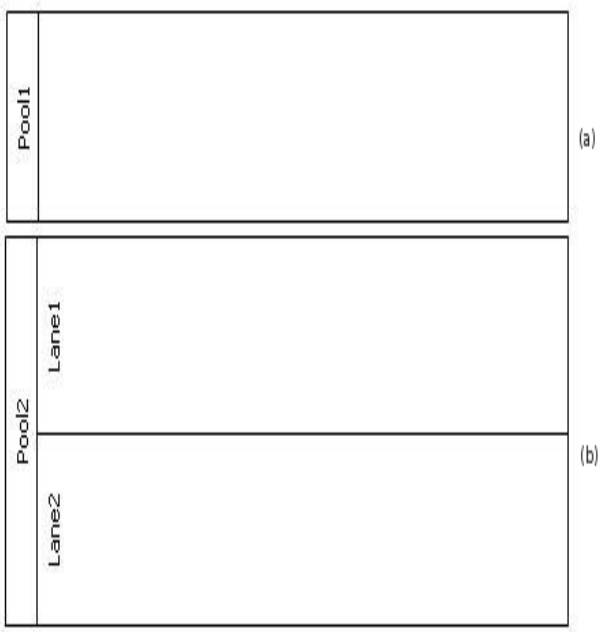


Fig. 2.7: Tipos de *Swimlanes*: (a) *Pool*, (b) Uma *Pool* com duas *Lanes*.

de seqüência. A comunicação entre *Pools* deve ser realizada através do objeto de conexão fluxo de mensagens. As *Lanes* são freqüentemente utilizadas para separar as atividades associadas com uma companhia específica, um domínio ou um papel. A comunicação entre *Lanes* dentro da mesma *Pool* é feita através do objeto de conexão fluxo de seqüência.

Artefatos

Os artefatos permitem adicionar contextos a uma situação particular do diagrama. A especificação BPMN define três tipos de artefatos: objetos de dados, grupos e anotações. Os objetos de dados são usados para mostrar como o dado deve ser enviado para uma atividade e como o dado é produzido por uma atividade. O artefato grupo é usado para documentação e agrupamento de atividades que formam um bloco comum com base em algum critério. O artefato anotação permite adicionar anotações textuais com informações para o leitor do diagrama. A Figura 2.8 ilustra a forma gráfica de cada artefato.

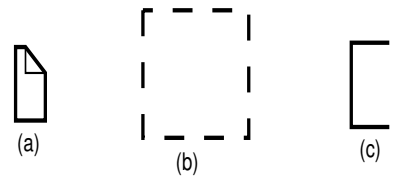


Fig. 2.8: Tipos de artefatos: (a) Objetos de dados, (b) Grupos, (c) Anotação.

Exemplo

A Figura 2.9 mostra um exemplo simples de um processo para compra de CD em uma loja on-line. O processo é formado por dois participantes (duas *pools*): cliente/usuário e a loja on-line. Do lado do cliente/usuário, não há nenhuma subdivisão de atividades. Do lado da loja on-line as atividades estão divididas em duas sub-partições (duas *lanes*): interface e estoque.

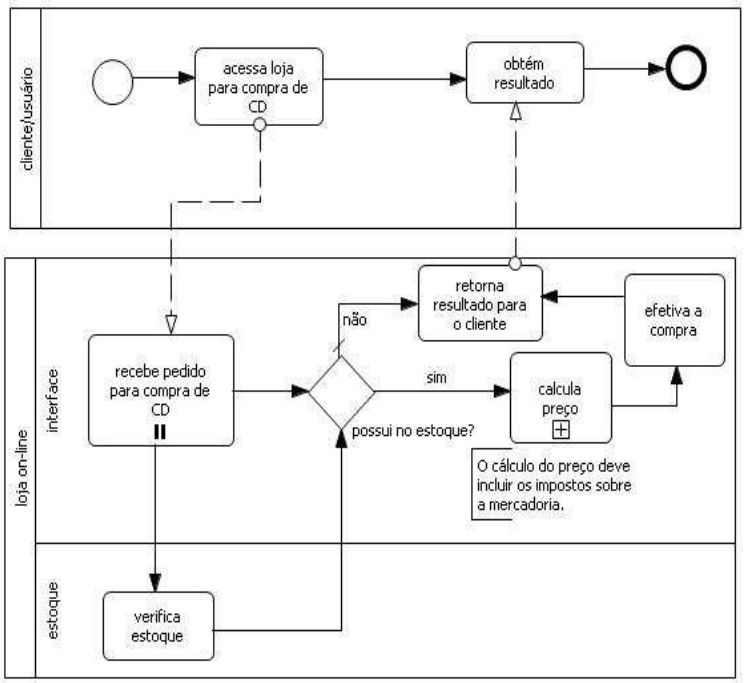


Fig. 2.9: Exemplo de uso da notação BPMN.

O evento inicial dispara a atividade "acessa loja para compra de CD" no participante "cliente/usuário". A atividade "recebe pedido para compra" no participante "loja on-line" possui o símbolo de múltiplas instâncias uma vez que ela poderá atender vários pedidos ao mesmo tempo. A comunicação entre os dois participantes é feita usando o objeto de conexão fluxo de mensagens. A comunicação entre *lanes* dentro de um mesmo participante ocorre usando o objeto de conexão fluxo de seqüência. O *gateway* "possui no estoque?" representa um ponto de decisão no fluxo das mensagens. Note a presença de uma pequena barra atravessando o fluxo de seqüência "não" do *gateway*. Este símbolo representa o fluxo padrão, ou seja, quando qualquer comportamento anormal ocorrer, este deverá ser o fluxo seguido. O sub-processo "calcula preço" possui outras atividades internas a ele que poderiam ser mostradas de forma expandida em outro diagrama. Note a presença de uma anotação textual abaixo da atividade "calcula preço". O evento final representa o fim do processo de compra do CD na loja.

2.1.4 Redes Ópticas

A tecnologia de redes ópticas surgiu como uma solução para os problemas de banda e gargalo encontrados nas redes atuais. Organizações e comunidades internacionais tais como o IETF, ITU-T e OIF estão criando especificações a fim de definir padrões para possibilitar o desenvolvimento de novas soluções relacionadas às redes ópticas. Todas estas organizações concordam que as futuras redes de transmissão terão de dezenas a centenas de Gbps de banda disponível para atender a todos os tipos de aplicações que requerem taxas de transmissão mais elevadas. As redes ópticas consistem basicamente de elementos tais como roteadores, *switches*, multiplexadores *Add-Drop* (ADM - *Add-Drop Multiplexor*), comutadores fotônicos (PXC - *Photonic cross-Connects*) e comutadores ópticos (OXC - *Optical cross-Connects*) [Mannie, 2004]. Além disso, as redes ópticas fazem uso de técnicas de multiplexação tal como DWDM (*Dense Wavelength Division Multiplexing*). Através destas tecnologias, arquiteturas de redes ópticas inteligentes de próxima geração estão sendo definidas. Os maiores benefícios destas redes ópticas inteligentes são o provisionamento de conexões de forma automática e dinâmica, a descoberta automática de topologias e recursos, a engenharia de tráfego para a otimização dos recursos de rede e a capacidade automática de proteção e restauração de falhas.

Nesta seção, iremos apresentar duas destas arquiteturas: ASON e GMPLS. A primeira delas serve como modelo de referência para o provisionamento automático de conexões ópticas. A arquitetura ASON define um modelo abstrato e as funcionalidades necessárias para prover conexões ópticas de forma automática. A arquitetura GMPLS, diferentemente do modelo ASON, propõe um conjunto de protocolos para o provisionamento automático de conexões ópticas. Tal arquitetura estende os protocolos definidos para o MPLS a fim de atender aos novos tipos de tecnologia de comutação de rótulos.

ASON

Definida pelo ITU-T como uma arquitetura de referência para o plano de controle, o ASON [ASON, 2001] introduz inteligência para a rede de transporte óptica. O ASON tem como proposta facilitar a configuração de conexões permanentes e comutadas (ver definições abaixo). A recomendação ASON, além de definir uma arquitetura para o plano de controle óptico, identifica a relação básica entre os planos de controle, gerência e transporte (ver Figura 2.10). Esta relação é usada nesta tese como um modelo para a definição das camadas que compõem a arquitetura de gerência e provisionamento de serviços dentro de um domínio e que será apresentada no Capítulo 5.

As funcionalidades de cada um dos planos são apresentadas a seguir:

- O plano de transporte inclui todos os equipamentos de rede, fibras e elementos responsáveis

pelo envio dos dados através dos canais ópticos;

- O plano de controle provê a inteligência da rede através das capacidades de roteamento e sinalização para o estabelecimento de conexões. Ou seja, um controlador de roteamento faz a descoberta de topologia e um protocolo de sinalização distribui a requisição de conexão através da rede e aloca recursos. Todas as informações trocadas pelos protocolos do plano de controle navegam através de um canal de controle. A rede utilizada pelo plano de controle é a rede de comunicação de dados conhecida como DCN (*Data Communication Network*);

- O plano de gerência é responsável por receber e realizar as requisições para o estabelecimento, remoção e manutenção das conexões ópticas. Além disso, o plano de gerência implementa (em conjunto com o plano de controle) as cinco áreas funcionais de gerenciamento FCAPS.

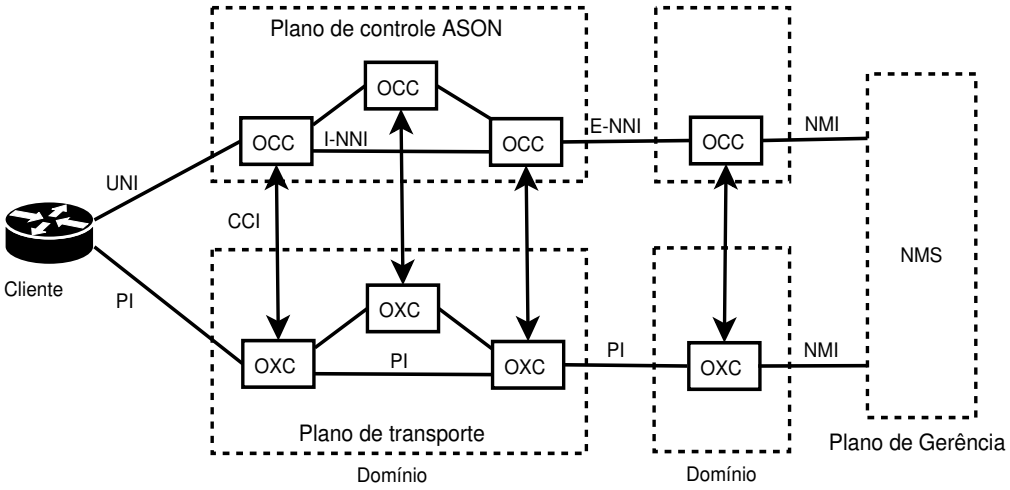


Fig. 2.10: Arquitetura ASON.

O plano de controle ASON não é uma coleção de protocolos, mas sim, uma arquitetura que define diferentes componentes funcionais para realizar funções específicas, dentre elas, sinalização e roteamento. As interações entre estes componentes e o fluxo de informações requerido para a comunicação entre componentes trafegam via interfaces. Estas interfaces (UNI, I-NNI e E-NNI) são conhecidas no ASON como "pontos de referência". A interface UNI (*User-to-Network Interface*) permite a troca de informações de sinalização entre um cliente (por exemplo, uma rede IP/MPLS) e a rede óptica. As interfaces I-NNI (*Internal Network-to-Network Interface*) e E-NNI (*External Network-to-Network Interface*) permitem o fluxo de mensagens para sinalização e roteamento dentro

e entre domínios, respectivamente. A interface NMI é responsável pela troca de informações entre o sistema de gerência (NMS - *Network Management System*) e os planos de controle e transporte.

No contexto do ITU-T, o ASON define dois conceitos importantes:

- *Call*: É uma associação entre elementos de rede que provê uma instância de um serviço;
- *Connection*: É uma concatenação de conexões em enlaces e sub-redes que permite o transporte de informações do usuário entre os pontos de ingresso e egresso de uma sub-rede.

Uma *call* não provê uma conectividade real para transmissão de tráfego do usuário, mas apenas constrói um relacionamento pelo qual futuras conexões poderão ser estabelecidas. Desta forma, uma *call* pode conter zero, uma ou múltiplas conexões. O ASON permite o estabelecimento de três tipos de conexões ópticas:

- *Permanent Connection* - PC: É uma conexão estabelecida pela configuração de todos os elementos de rede ao longo do caminho com os parâmetros requeridos para estabelecer uma conexão fim-a-fim. Tal provisionamento é feito pelo sistema de gerência ou intervenção manual;
- *Soft-permanent Connection* - SPC: É uma conexão pela qual um sistema de gerência configura o nó de origem enquanto os protocolos de roteamento e sinalização são utilizados para estabelecer a conexão fim-a-fim ao longo do caminho dentro do domínio;
- *Switched Connection* - SC: É uma conexão iniciada por uma rede cliente (exemplo, redes IP/MPLS) e estabelecida através dos protocolos de roteamento e sinalização. Neste caso há uma interação entre o lado cliente UNI (UNI-C) e o lado de rede UNI (UNI-N) a fim de trocarem mensagens de sinalização.

Embora ocorra a mesma sinalização para o estabelecimento das conexões SPC e SC, uma conexão SPC é solicitada somente através de um sistema de gerenciamento, enquanto uma conexão SC é solicitada por uma rede cliente. A Figura 2.11 mostra o modelo utilizado neste trabalho e a sinalização das conexões SPC e SC. Nesta tese, a arquitetura proposta leva em conta apenas o provisionamento de SPCs e não considera o conceito de *Call*.

GMPLS

O GMPLS [Mannie, 2004] apresenta uma arquitetura que estende o MPLS para prover um plano de controle (sinalização e roteamento) não somente para dispositivos que realizam a comutação de pacotes, mas também, dispositivos com capacidade de comutação em *slots* de tempo, comprimentos

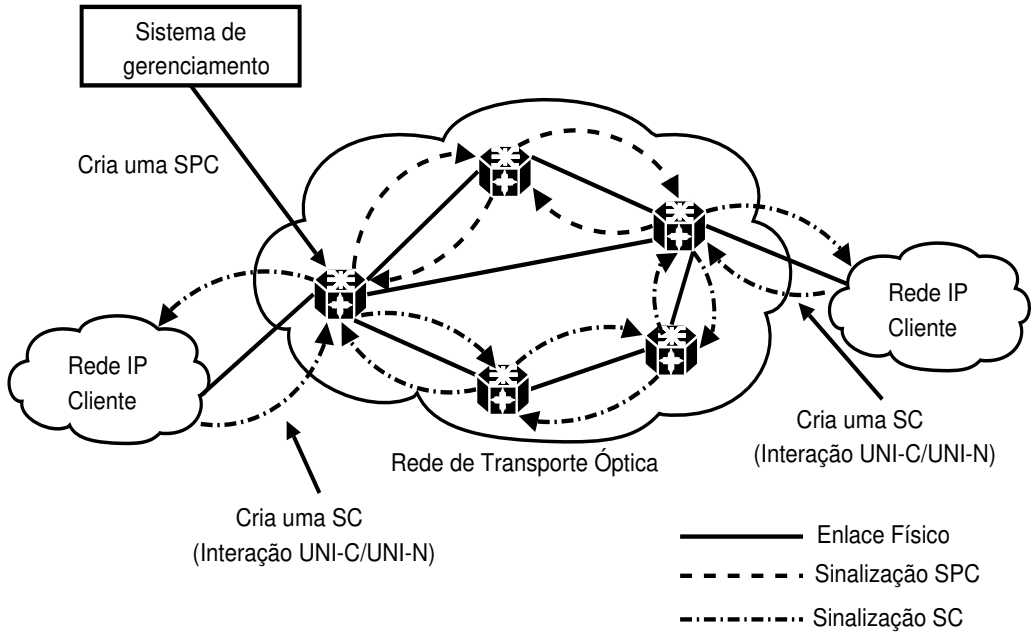


Fig. 2.11: Sinalização SPC e SC.

de onda e fibras. Estes dispositivos são roteadores (LSRs - *Label Switching Routers*) com um conjunto de interfaces que executam outras operações de comutação além da comutação de pacotes. Estas interfaces podem ser classificadas como [Mannie, 2004]:

- Interfaces PSC (*Packet Switch Capable*): são interfaces que recebem pacotes de entrada e encaminham os dados baseado no conteúdo (conhecido como rótulo) do cabeçalho do pacote. Exemplo, roteadores MPLS;

- Interfaces L2SC (*Layer-2 Switch Capable*): são interfaces que recebem quadros/células e comutam os dados baseados no conteúdo do cabeçalho do quadro/célula. Exemplos, interfaces em pontes (*bridges*) *Ethernet* que comutam dados baseado no conteúdo do cabeçalho MAC e interfaces em comutadores ATM (ATM-LSRs) que encaminham dados baseado no VPI/VCI ATM;

- Interfaces TDM (*Time-Division Multiplex Capable*): são interfaces que comutam dados baseados nos *slots* de tempo em um ciclo de repetição. Exemplos, interfaces em comutadores SONET/SDH (XC - SONET/SDH *Cross-Connect*), multiplexadores (TM - *Terminal Multiplexer*) e multiplexadores *Add-Drop* (ADM - *Add-Drop Multiplexer*);

- Interfaces LSC (*Lambda Switch Capable*): são interfaces que comutam o comprimento de onda em que o sinal é recebido. Exemplos, comutadores fotônicos (PXCs - *Photonic Cross-Connects*) e comutadores ópticos (OXCs - *Optical Cross-Connects*);

- Interfaces FSC (*Fiber-Switch Capable*): são interfaces que comutam dados baseadas na posição dos dados no espaço físico (fibra, porta). Exemplo, PXC e OXC podem operar no nível de fibras simples ou múltiplas .

Deve ser observado que há uma correlação entre a operação de comutação de rótulos, definido em interfaces PSC e que têm como exemplo roteadores MPLS, com as demais operações de comutação. Enquanto o rótulo utilizado no MPLS para comutação é explícito, em outras operações de comutação o rótulo foi substituído por outros esquemas similares de encaminhamento, baseados em *slots* de tempo, comprimentos de onda e fibra.

Desde que o termo *Generalized* MPLS (GMPLS) foi adotado para denotar a generalização do plano de controle MPLS a fim de prover múltiplos tipos de redes comutadas, o termo "LSP" (*Label Switched Path*) é utilizado no GMPLS para denotar diferentes tipos de circuitos, tais como, conexões SONET/SDH, um caminho óptico, um LSP MPLS e assim por diante.

No GMPLS, um conjunto de protocolos foi definido para o plano de controle a fim de atender três funções principais: gerenciamento de enlace, roteamento e sinalização [Bernstein et al., 2003]. O gerenciamento de enlaces é uma função implementada entre cada par de nós vizinhos. Esta é uma nova função que não existia no MPLS e que foi incorporada ao GMPLS através da criação do protocolo LMP (*Link Management Protocol*). O roteamento GMPLS foi estendido a partir do MPLS-TE para suportar a descoberta de recursos e topologias. O roteamento GMPLS é representado pelos protocolos OSPF-TE e IS-IS-TE e permite a alocação de vários atributos aos enlaces (por exemplo, proteção 1+1, 1:1, sem proteção) e a propagação de conectividades e informações de atributos (recursos) de um nó para todos os outros nós da rede. A sinalização GMPLS utiliza os protocolos de sinalização do MPLS-TE (RSVP-TE e CR-LDP) com extensões para manipulação de múltiplas tecnologias de comutação. Algumas extensões significativas incluem rótulo generalizado, bidirecionalidade e a separação dos planos de controle e dados.

Nesta tese, os protocolos GMPLS foram usados para a instanciação da arquitetura para provisionamento de serviços dentro de um domínio.

2.2 Trabalhos Relacionados

A motivação para esta tese tem origem principalmente em dois grupos de trabalhos relacionados. O primeiro vem sendo desenvolvido na rede CANARIE no Canadá [CANARIE Project, 2006] e tem como objetivo dar aos usuários da rede óptica o controle sobre os recursos ópticos. O segundo grupo de trabalhos relacionados refere-se ao conjunto de documentos (*drafts* e RFCs) do IETF relacionados com o fornecimento de conexões entre domínios baseado no GMPLS. Os documentos IETF mais recentes especulam sobre mecanismos para estabelecimento de conexões entre domínios, porém, tais

mecanismos dependem de extensões que precisam ser feitas nos protocolos atuais, tais como o BGP. Entretanto, a extensão de protocolos é algo que exige cautela uma vez que a substituição de protocolos atuais por novos protocolos acaba sendo o maior desafio destas soluções [Xiao et al., 2004].

2.2.1 Rede CANARIE

O grupo da rede CANARIE considera que usuários possam alugar e gerenciar seus próprios recursos (caminhos ópticos) criando um conceito conhecido como *User Controlled LightPath* (UCLP). Os recursos possuem características próprias (banda, proteção, etc.) e são representados como "*Lightpath Objects*"(LPOs) e armazenados em um registro de LPOs. Por meio de um sistema de gerência, usuários criam túneis fim-a-fim que atravessam vários domínios. A criação de túneis fim-a-fim é realizada através da reserva de LPOs em todos os domínios por onde passa o túnel. Os LPOs também podem ser concatenados, particionados, divulgados/alugados, e estabelecidos fim-a-fim. Dois grupos têm atuado no desenvolvimento do UCLP. O primeiro [Boutaba et al., 2004] possui uma abordagem bastante promissora e definiu um conjunto de operações que permitem o estabelecimento dos túneis entre domínios. As operações são as seguintes:

- Criar LPO: Esta funcionalidade permite criar caminhos de luz entre dois dispositivos fisicamente adjacentes. Esta é a operação mais básica do sistema pois é através dela que as outras operações podem ser realizadas;
- Concatenar LPOs: Esta operação permite “crossconectar” vários LPOs a fim de formar um LPO maior conectando dispositivos fisicamente não adjacentes. Esta concatenação dá origem a outro LPO que armazena a lista dos identificadores dos LPOs concatenados;
- Particionamento de LPOs: O particionamento de LPOs realiza a divisão de um LPO “pai” dando origem a múltiplos LPOs “filhos” com banda menor. Os LPOs “filhos” possuem os mesmos nós de origem e destino do LPO “pai”;
- Divulgação de LPOs: A divulgação de LPOs permite que os proprietários de LPOS divulguem recursos não utilizados para outros usuários. Os recursos podem ser divulgados e possuir uma período de validade. Outros usuários podem obter estes LPOs, particioná-los e divulgá-los novamente;
- Estabelecimento de LPO fim-a-fim: Esta operação permite o estabelecimento de um caminho de luz entre um par de dispositivos ópticos. Ela é executada em duas partes. Primeiramente encontra-se a rota entre os dois dispositivos. Após, o usuário deve reservar os LPOs da rota escolhida entre os dispositivos. Esta reserva consiste em realizar a operação de concatenação explicada acima.

A arquitetura proposta possui uma camada de acesso para usuários pela qual clientes acessam o sistema usando HTTP. Há também uma camada de provisionamento de serviços onde toda a lógica para realização das tarefas citadas acima é implementada. Esta camada foi desenvolvida usando a tecnologia *Web Services*. Entretanto, uma das deficiências do trabalho é não considerar as restrições impostas pelos domínios. O estabelecimento de caminhos ópticos fim-a-fim não leva em conta as políticas locais de cada domínio.

Por outro lado, o trabalho apresentado em [Truong et al., 2004], também vinculado ao projeto CANARIE, discute uma solução a fim de garantir que regras de gerenciamento de cada domínio sejam aplicadas. Um mecanismo de reserva de recursos em duas fases permite que os caminhos de luz sejam reservados respeitando as regras locais em cada domínio administrativo. Tal solução também implementa um mecanismo de reservas baseado em prioridades garantindo que usuários com mais alta prioridade tenham preferência durante a reserva dos recursos. A implementação desta proposta foi testada em um cenário real e os tempos obtidos serão usados para serem comparados com os tempos obtidos nesta tese.

Enquanto os dois projetos mencionados acima permitem que os usuários gerenciem seus próprios caminhos de luz, no modelo proposto nesta tese a gerência de todos os recursos ópticos é responsabilidade do domínio. Acreditamos que o modelo UCLP, embora promissor, precisa passar por um processo de amadurecimento e convencimento dos provedores. Recentemente, o desenvolvimento do UCLP tem considerado o uso de máquinas de orquestração para o controle dos fluxos de trabalho (*Workflows*) e instrumentação de sensores, *grids* e redes [Arnaud, 2004].

2.2.2 GMPLS entre Domínios

Esta seção é, em sua maior parte, baseada no conjunto de *drafts* e RFCs IETF que propõem mecanismos para sinalização de LSPs entre domínios. Também apresentamos alguns trabalhos recentes que propõem idéias similares às apresentadas nesta tese. Inicialmente, discutimos quatro maneiras para estabelecimento de LSPs entre domínios:

- *Nesting* de LSPs

Forwarding Adjacencies (FAs) são introduzidas e explicadas com detalhes por Kompella e Rekhter [Kompella and Rekhter, 2005a]. Uma FA é definida como um LSP que conecta dois elementos de rede e é divulgada como um enlace TE (*Traffic Engineering*). Este enlace poderá ser usado pelos algoritmos de roteamento para o cálculo de rotas. Note que uma FA não precisa ter adjacência física entre os dois elementos de rede. Em particular, uma FA pode ser usada para oferecer conectividade entre pares de nós em um domínio.

A técnica usada para carregar um LSP em outro é chamada de *Nesting* de LSPs. Uma FA deve prover um túnel TE para transportar (i.e. aninhar) múltiplos LSPs-TE através de uma parte comum em seus caminhos. Alternativamente, um LSP-TE pode transportar um único LSP em um mapeamento um-para-um. O *trigger* para o estabelecimento de um LSP FA pode ser a recepção de uma requisição de sinalização do LSP que será transportado, ou pode ser uma ação da gerência (“pré-engenharia") no domínio a ser atravessado pelo LSP que seria usado como FA para o tráfego que atravessa o domínio. Note que um LSP hierárquico deve ser construído para atravessar múltiplos domínios. Porém, a forma que este LSP deve ser divulgado como uma FA que atravessa um domínio não foi especificada.

- LSPs Contíguos

Um único LSP contíguo é estabelecido do ingresso ao egresso em uma única troca de sinalização. Não são necessários LSPs adicionais para suportar este LSP. A sinalização ocorre somente entre vizinhos adjacentes (sem tunelamento), sendo que neste caso somente é necessária sinalização *hop-by-hop*.

- *Stitching* de LSPs

No modelo LSP *Stitching*, LSPs diferentes (referidos como segmentos de LSP-TE) são estabelecidos e “colados" (*stitched*) no plano de dados de forma que um único LSP é formado. A distinção é que os LSPs componentes dos segmentos são sinalizados como LSPs-TE distintos no plano de controle. Cada segmento pode ser usado para suportar o trecho de cada domínio em um LSP entre domínios. Desta forma, o ingresso e egresso de cada trecho podem ser os nós de borda de cada domínio. O *trigger* para a sinalização de estabelecimento de um segmento do LSP-TE pode ser o estabelecimento do LSP no segmento anterior, o recebimento de uma requisição de estabelecimento de um LSP que queira realizar o *stitch*, ou mesmo uma ação de gerenciamento.

- Modelos Híbridos

Farrel et al. [Farrel et al., 2005] sugerem a possibilidade da mistura dos métodos de sinalização descritos para estabelecimento de um LSP-TE fim-a-fim entre domínios, não indicando, entretanto como isto pode ser feito.

Nesta tese estamos particularmente interessados nos modelos *nesting* e *stitching*. O modelo *nesting* é usado através das políticas para agregação de LSPs em outros LSPs. O modelo *stitching* é usado pois o estabelecimento de conexões entre domínios realiza “crossconexões” nos elementos de borda de um domínio fazendo com que os caminhos de luz (segmentos) sejam “colados” para formar o *lightpath* fim-a-fim. Estas “crossconexões” são realizadas via ações de gerenciamento.

As técnicas para a computação de caminhos discutidas por Farrel et al. [Farrel et al., 2005] estão fortemente relacionadas com os métodos de sinalização descritos acima, de forma que certas técnicas podem requerer o uso de técnicas de sinalização específicas. Conforme os documentos do IETF, podemos definir três maneiras para a computação dos caminhos. A primeira e mais simples é via gerência. A segunda é feita pelo roteador *Head end*, e a terceira é realizada via um componente conhecido como *Path Computation Element* (PCE).

A computação de caminhos via gerência pode ser desempenhada por ferramentas *offline* ou através de planejamento de rede. O caminho resultante deve ser passado ao LSR (*Label Switched Router*) de ingresso como parte da requisição do LSP-TE, e deve ser codificado pelo LSR de ingresso no ERO (*Explicit Route Object*) na mensagem PATH a ser enviada (no caso da sinalização ser baseada no RSVP). Se a ferramenta *offline* ou o planejador da rede tiver informações suficientes de todos os domínios, a rota pode conter todos os nós do caminho atravessando todos os domínios.

O LSR de ingresso (*Head end*) pode assumir a responsabilidade de computação de caminhos quando o operador provê parte ou não provê a rota explícita. Neste caso, pelo menos o endereço de destino deve ser fornecido. Se o LSR de ingresso tiver visibilidade suficiente da topologia e informações de TE para todos os domínios por onde passa a rota do LSP, então este deve computar o caminho completo. A qualidade do caminho é melhor se o ingresso tiver visibilidade completa de todos os domínios relevantes. Por outro lado, se o ingresso não tiver visibilidade completa dos domínios relevantes para o caminho em questão, porém tiver visibilidade sobre as fronteiras dos domínios além da disponibilidade dos recursos de TE, este pode prover os seguintes componentes:

- Lista explícita de nós dentro do domínio local;
- Rota frouxa (*loose route*) dos roteadores de ingresso de cada domínio a ser percorrido (ou mesmo um identificador de cada domínio).

Esta estratégia requer um mecanismo adicional para a computação dos caminhos internos de cada domínio. Finalmente, se o ingresso tiver visibilidade somente do domínio local, este é capaz de criar a rota somente até o ponto de saída do domínio. O ponto de saída é representado como um nó (*loose hop*) identificando o egresso.

As técnicas apresentadas acima confiam nos elementos de borda de cada domínio e possuem problemas de escalabilidade. Além disso, os LSRs de borda dos domínios escolhidos podem não ser parte do caminho ótimo se consideradas as restrições de TE. Uma técnica alternativa confia a responsabilidade da computação de caminhos ao PCE. Pode haver somente um PCE centralizado, ou múltiplos (cada um tendo visibilidade local e colaborando de forma distribuída para computar um caminho fim-a-fim). O PCE deve coletar informações topológicas e de TE da mesma forma que os LSRs em um domínio, ou através de outros meios. Quando um LSR de ingresso necessita de uma

rota, a computação do caminho deve ser solicitada para o PCE de seu domínio. Somente um PCE com visibilidade sobre todos os domínios pode ser usado. Porém, considerando que cada domínio está sob uma responsabilidade administrativa, o caso mais comum deve ser um ou mais PCEs por domínio.

Um dos principais problemas para cálculo de caminhos entre domínios está relacionado à divulgação de informações sobre os recursos TE para outros domínios. Enquanto informações de TE são distribuídas dentro dos domínios através de extensões a IGPs bem definidas, a distribuição de tais informações entre domínios é ainda motivo de muitas discussões. Além disso, os protocolos de comunicação entre PCEs estão em fase inicial de especificação [Fang et al., 2005].

A maneira como os PCEs interagem para obtenção de informações de outros domínios é parcialmente explicada em [Ricciato et al., 2005]. Entretanto, uma descrição completa sobre o mecanismo PCE foi encontrada em um *draft* expirado [Vasseur et al., 2004] referenciado, durante a escrita desta tese, em [How the PCE Works, 2006]. O mecanismo proposto pelos dois documentos consiste em obter topologias virtuais através da interação entre os PCEs de domínios diferentes. O PCE *i* interage com o PCE *i+1* até alcançar o PCE do último domínio da rota. A resposta de retorno entre os PCEs agrupa informações sobre enlaces virtuais de menor custo (levando-se em consideração aspectos de QoS) em cada domínio em direção ao destino. Estas informações são processadas no PCE (1) que então encontra uma rota de menor custo até o destino.

Entretanto, como mencionado em [Ricciato et al., 2005] e em [Fang et al., 2005], o PCE apenas encontra a melhor rota no nível de nós de uma rota de domínios já definida. Ou seja, o PCE não possui capacidade para escolher a melhor rota entre as várias possíveis entre diferentes sistemas autônomos. O PCE usa o BGP para escolher tal rota e interage com os PCEs desta rota para obter as informações topológicas de TE. Porém, o BGP divulga apenas informações de alcançabilidade e a rota escolhida em direção a cada prefixo de rede não leva em conta nenhum atributo de QoS. Nestas circunstâncias, retornamos ao problema da extensões aos protocolos atualmente usados na Internet. Se para calcularmos uma rota entre domínios com QoS dependemos de tais extensões, a forma como o PCE realiza suas tarefa não resolve o problema de QoS entre domínios.

Esta dependência por extensões nos protocolos e a demora pelas padronizações fizeram com que soluções na camada de gerência e serviços fossem propostas. A idéia básica destas soluções é criar uma camada que ofereça mecanismos para sinalização e estabelecimento de conexões entre domínios com QoS sem alterar os protocolos tradicionais da Internet. Algumas propostas especificamente para redes IP podem ser encontradas em [Mahajan et al., 2005, Agarwal et al., 2003, Lakshminarayanan et al., 2004], entre outras.

Para complicar estas extensões aos protocolos, as informações a serem distribuídas em redes GMPLS incluindo redes TDM, redes IP e redes ópticas, são mais diversificadas uma vez que o tipo

de comutação das interfaces depende da tecnologia do domínio. Além disso, um elemento de rede pode possuir diferentes interfaces. Informações de capacidade de comutação de cada interface, tipo de proteção, bandas máximas e mínimas reserváveis e tipo de codificação (Ethernet, SONET/SDH, Lambda) precisam ser consideradas durante a computação do caminho óptico. A extensão dos protocolos para suportar este tipo de informação ainda encontra-se em fase inicial de discussão. Recentemente, um *draft* IETF propôs extensões ao BGP para que o mesmo carregue informações de engenharia de tráfego entre domínios administrativos [Ould-Brahim et al., 2006b]. O documento define um atributo denominado de "atributo para engenharia de tráfego" que contém, entre outros dados, informações relacionadas com banda máxima reservável para cada nível de prioridade de tráfego e a capacidade de comutação da interface. Um trabalho de implementação foi realizado em [Valancius, 2006] onde extensões para disseminar informações GMPLS entre domínios foram feitas no protocolo BGP. O trabalho estendeu o protocolo BGP da *suite* Quagga [Quagga, 2006] e avaliou as extensões em um cenário real simples. Porém, um modelo formal foi definido para analisar o comportamento do protocolo em cenários maiores. As conclusões preliminares confirmam o que se esperava: estender o BGP para distribuição de informações GMPLS é possível, porém colocar o protocolo com tais extensões num cenário real na escala da Internet pode não ser viável.

Outro trabalho recente que propõe mecanismos similares aos propostos nesta tese pode ser encontrado em [Yang et al., 2006]. Tal trabalho, denominado DRAGON (*Dynamic Resource Allocation via GMPLS Optical Networks*), de certa forma resume todas as outras propostas para provisionamento de serviços entre domínios uma vez que a maioria dos problemas enfrentados atualmente são considerados pelo trabalho. Os autores destacam as dificuldades relacionadas ao provisionamento de serviços entre domínios e a demora nos organismos de padronização para definirem especificações que incorporem soluções para este tipo de problema. Os autores propõem a distribuição de informações resumidas sobre a topologia interna de cada domínio (similar ao conceito de topologia virtual). Esta distribuição é realizada através de um módulo que usa uma versão modificada do OSPF-TE. O cálculo da rota entre domínios é feito usando as informações distribuídas entre os domínios e pode envolver a interação entre domínios realizada por alguns módulos desenvolvidos pelo grupo. A rota retornada pode ser uma rota "frouxa" ou estrita e será sinalizada pelo RSVP. Naquele trabalho, apesar da proposta não exigir uma extensão no BGP, uma versão modificada do OSPF-TE é necessária. Além disso, não há um mecanismo de negociação para realizar a reserva dos recursos. A implementação da arquitetura proposta está numa fase inicial e nenhum teste de desempenho foi realizado até o momento. Porém, acreditamos que tal projeto seja um dos mais completos e promissores atualmente.

O provisionamento de VPNs entre domínios consiste basicamente em estabelecer conexões entre as portas localizadas nos diferentes domínios. Portanto, as mesmas dificuldades para estabelecimento

de conexões entre domínios estão presentes no cenário das VPNs entre domínios. Além disso, deve haver um mecanismo para distribuição de informações de membros das VPNs a fim de que as portas localizadas em outros domínios sejam conhecidas por todos os domínios que pertencem às VPNs. Recentemente, um *draft* IETF [Li and Gao, 2006] apresentou algumas discussões com idéias relacionadas a como distribuir os mapeamentos de portas entre as VPNs de camada 1 (L1VPNs). As idéias apresentadas naquele *draft* possuem algumas semelhanças com as propostas apresentadas nesta tese. Os autores propõem o uso de um diretório centralizado ou distribuído que armazene e faça a correlação de portas das VPNs. Eles também definem os modelos *Push* e *Pull* de forma similar a como foi definido neste trabalho.

Finalizações do capítulo

No próximo capítulo, apresentamos o modelo proposto para a gerência e provisionamento de serviços em redes ópticas. O modelo, baseado no modelo de referência TMN/FCAPS, incorpora as funcionalidades necessárias para o provisionamento de serviços tanto dentro de um domínio como entre diferentes domínios administrativos.

Capítulo 3

Definição do Modelo para Provisionamento e Gerência de Serviços em Redes Ópticas

Neste capítulo, o modelo proposto para provisionamento de serviços em redes ópticas é descrito. O modelo é baseado no modelo de referência TMN/FCAPS que serviu como base para a definição de um modelo específico para provisionamento de serviços em redes ópticas. A partir deste modelo, a arquitetura foi definida para suportar o provisionamento de serviços em um domínio e o provisionamento de serviços entre domínios. A Figura 3.1 ilustra a forma como a arquitetura foi desenvolvida.

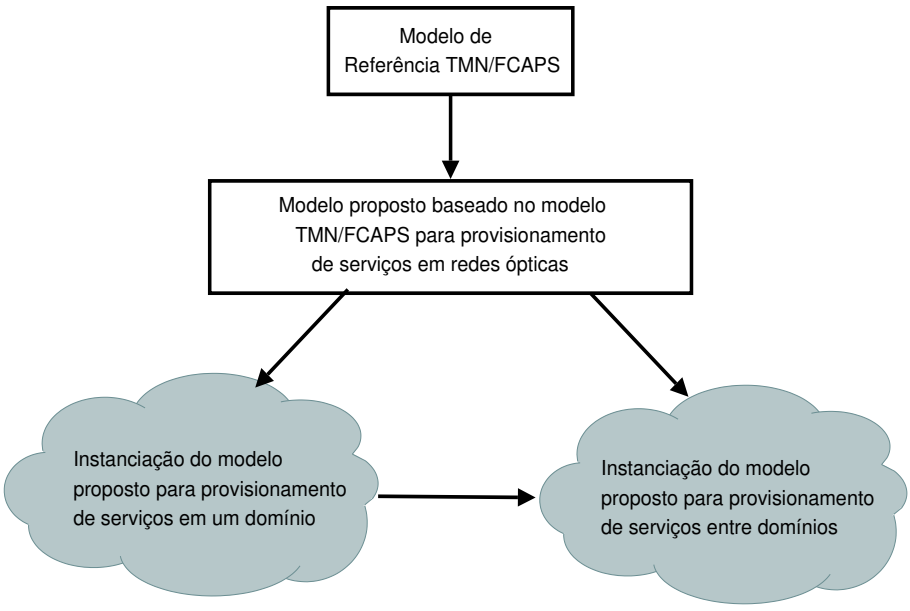


Fig. 3.1: Modelo proposto e sua instanciação.

O modelo proposto nesta tese faz uso do modelo de referência TMN/FCAPS para dividir

em camadas as funcionalidades da arquitetura. Em cada camada do modelo foram definidas funcionalidades para suportar o provisionamento e a gerência de serviços em redes ópticas. Como mencionado no Capítulo 2, o modelo TMN é divido em 4 partes: camada de gerenciamento do elemento de rede, camada de gerenciamento de rede, camada de gerenciamento de serviços e camada de gerenciamento de negócios. Como estamos interessados em gerenciar aspectos relacionados ao modelo ISO FCAPS e consideramos a independência de cada domínio, temos como resultado o modelo ilustrado na Figura 3.2.

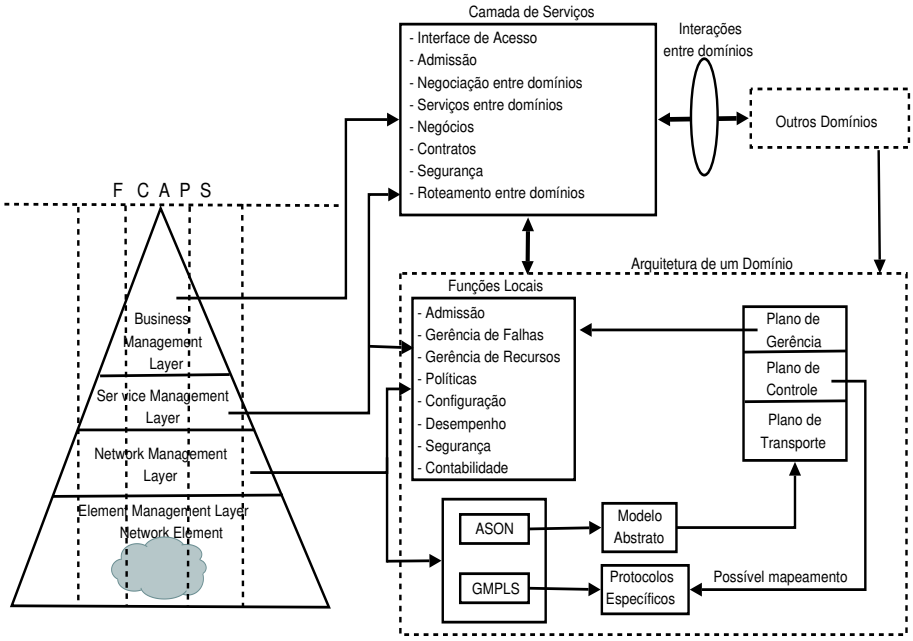


Fig. 3.2: Modelo proposto baseado no modelo de referência (TMN/FCAPS).

No lado esquerdo da Figura 3.2, pode-se perceber que as áreas de gerenciamento FCAPS atuam em todas as camadas da pirâmide do modelo TMN podendo gerenciar e interferir na forma como cada camada se comporta. Na camada mais inferior da pirâmide (*Element Management Layer*) estão localizados os dispositivos ópticos e os elementos de rede que fazem parte da rede óptica. Esta camada representa a camada de transporte no modelo ASON. A gerência dos elementos de rede é responsabilidade de cada domínio.

A camada de gerência de redes (*Network Management Layer*) se caracteriza por oferecer funcionalidades para o estabelecimento, remoção e gerência de conexões ópticas. O estabelecimento e remoção das conexões ocorre através da utilização de arquiteturas como ASON e GMPLS, ambas apresentadas no Capítulo 2. Enquanto a arquitetura ASON define apenas um modelo abstrato para provisionamento automático de conexões ópticas, a arquitetura GMPLS define os protocolos para a realização de tal provisionamento. Porém, a gerência e o controle do provisionamento das conexões não são definidas em nenhuma arquitetura. Desta forma, o modelo de referência proposto para este

trabalho define algumas funcionalidades necessárias e genéricas que são realizadas em cada domínio administrativo. Embora a arquitetura ASON defina três níveis de funcionalidades, sendo eles o plano de transporte, o plano de controle e o plano de gerência, apenas as funções dos dois primeiros estão expostas de forma clara. As funcionalidades do plano de gerência são muito dependentes de cada domínio em relação ao que gerenciar e como gerenciar.

No modelo proposto, o plano de gerência é instanciado através do módulo Funções Locais. Nesta tese, as Funções Locais incorporam as cinco diferentes áreas de gerenciamento do modelo ISO FCAPS. Além disso, consideramos outras tarefas importantes como controle de admissão, gerência de recursos e o uso de políticas para a realização de funcionalidades como agregação de tráfego (*traffic grooming*). Estes módulos são acessados através dos serviços (localizados na camada de serviços) para solicitação de estabelecimento de conexões e VPNs, tanto dentro do domínio local como entre domínios. Dentro do bloco de Funções Locais destacamos as seguintes funcionalidades: Admissão, Gerência de Falhas, Gerência de Recursos, Políticas, Configuração, Desempenho, Segurança e Contabilidade. Em termos gerais, a admissão deve ser responsável pela verificação de permissões para estabelecimento de serviços. Regras simples de autorização podem ser definidas a fim de analisar contratos pré-definidos entre clientes e o domínio provedor dos serviços. A Gerência de Falhas deve oferecer funcionalidades para que o plano de gerência possa de alguma forma tentar minimizar o impacto de uma falha na rede óptica. Além disso, a Gerência de Falhas deve manter um histórico de falhas na rede de transporte. A Gerência de Recursos deve ser capaz de controlar todos os recursos da rede óptica: recursos físicos, caminhos de luz estabelecidos, contratos de clientes, VPNs estabelecidas, etc. As políticas armazenam regras que podem ser usadas durante a admissão e após uma falha na rede de transporte. Nesta tese, as políticas foram desenvolvidas para realizarem a agregação de tráfego da rede cliente em caminhos de luz da rede óptica. A configuração realiza o estabelecimento das conexões e VPNs. A área de gerência relacionada com o desempenho oferece funcionalidades para monitorar os serviços estabelecidos. Nesta tese estamos particularmente interessados em monitorar o tempo para estabelecimento dos serviços. A gerência de segurança neste nível deve prover mecanismos para verificação da identidade do solicitante. Finalmente, a gerência de contabilidade deve prover funcionalidades para gerenciar o uso dos recursos em relação aos aspectos relacionados com a tarifação dos serviços.

A camada de gerência de serviços (*Service Management Layer*) possui os serviços oferecidos em cada domínio. Tais serviços refletem as funcionalidades específicas de um domínio e a maneira como ele expõe cada serviço através de interfaces bem definidas. Para a Camada de Serviços definimos as seguintes funcionalidades: Interface de Acesso, Admissão, Negociação entre Domínios, Serviços entre Domínios, Negócios, Contratos, Segurança e Roteamento Entre Domínios. A interface de acesso define a forma como clientes e outros domínios devem invocar os serviços oferecidos pelo

domínio local. A admissão permite que algum controle possa ser feito na camada de serviços antes de encaminhar a requisição para as Funções Locais. A negociação e o roteamento entre domínios oferecem funcionalidades para que os domínios possam interagir para oferecer serviços aos seus clientes. Aspectos de negócios e contratos também podem ser considerados na camada de serviços. Finalmente, a segurança neste nível deve ser implementada através de mecanismos de cifragem usando tecnologias para garantir a integridade e confiabilidade dos dados. Neste trabalho, a camada de gerenciamento de serviços, como descrita pelo modelo TMN, foi adaptada para refletir as necessidades da arquitetura proposta. Dividimos a camada de gerenciamento de serviços em duas partes como pode-se observar na Figura 3.3

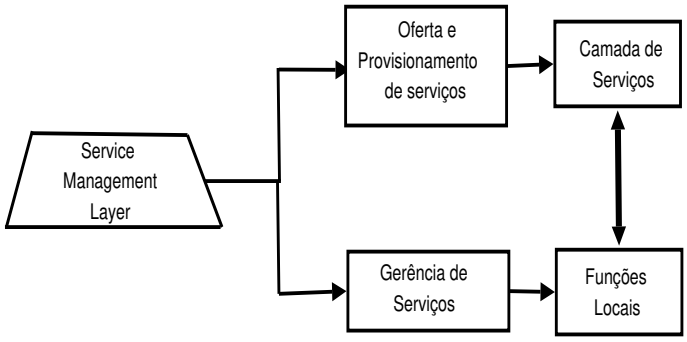


Fig. 3.3: Funcionalidades da Camada de Serviços.

Esta divisão foi necessária pois entendemos que o provisionamento de serviços e a gerência dos mesmos diferem na forma como suas funcionalidades são organizadas. O provisionamento de serviços se caracteriza por um conjunto de interfaces que neste trabalho tem como foco a oferta de serviços entre domínios. Estes serviços incluem tarefas como negociação, divulgação de topologias virtuais, estabelecimento de contratos, etc. Há também serviços oferecidos para realização de tarefas locais (em um domínio) tais como o provisionamento de conexões ópticas (SPCs) e VPNs.

Com esta divisão, a gerência dos serviços fica então responsável pelas funcionalidades associadas ao modelo FCAPS. Aspectos relacionados ao gerenciamento dos serviços, tais como consultas sobre conexões estabelecidas, serviços provisionados entre domínios, contratos atualmente estabelecidos, pontos de falhas no domínio e agregação de tráfego são alguns exemplos de gerência que estamos considerando e que são realizados pelas Funções Locais.

A Figura 3.3 mostra que a camada de serviços utiliza as Funções Locais para a realização das tarefas necessárias para o provisionamento dos serviços. Todas as requisições que chegam na camada de serviços são encaminhadas para os módulos locais do domínio a fim de serem processadas. Funções relacionadas com admissão e reserva de recursos são executadas pelos módulos locais. Na verdade, toda a lógica de controle para o provisionamento de serviços está localizada nos módulos que representam as funções locais. A camada de serviços age como uma camada de abstração que

oferece acesso às entidades externas e troca os parâmetros necessários para o estabelecimento dos serviços. A forma, tanto lógica como tecnológica, de como cada serviço é implementado em cada domínio é uma decisão local não sendo transparente para as entidades invocadoras dos serviços.

Finalmente, a camada de gerência de negócios (*Business Management Layer*) representa os aspectos relacionados à forma como um determinado domínio define suas estratégias para atrair clientes. Isto irá se refletir nos serviços oferecidos pela camada de serviços. Nesta tese, as estratégias de negócios consistem basicamente em divulgar topologias virtuais que reflitam o estado atual da rede óptica assim como os interesses atuais do domínio. Na prática, isto significa um melhor uso dos recursos ópticos e um maior número possível de conexões estabelecidas no domínio.

O modelo proposto na Figura 3.2 e descrito acima foi instanciado na forma de uma arquitetura que mapeia as funcionalidades citadas em módulos responsáveis pela realização de determinadas atividades. Estas atividades foram modeladas usando a notação BPMN para descrever os processos de negócios necessários para o estabelecimento dos serviços propostos neste trabalho.

Finalizações do capítulo

Antes de apresentarmos a instanciação do modelo, discutiremos o mecanismo de topologias virtuais. A proposta apresentada nesta tese abstrai os detalhes internos de cada domínio usando o conceito de topologia virtual e distribuindo as informações necessárias na camada de serviços. A quantidade de informações a ser trocada entre os domínios depende apenas da interação entre os serviços, não exigindo nenhum tipo de extensão aos protocolos existentes, principalmente extensões ao BGP. No próximo capítulo, apresentamos o mecanismo de topologias virtuais que é peça fundamental para a migração da arquitetura intra-domínios para uma arquitetura que suporte o provisionamento de serviços entre domínios. Iremos apresentar os dois modelos para obtenção de topologias virtuais, modelo *Push* e o modelo *Pull*.

Capítulo 4

O Mecanismo de Topologias Virtuais

O conceito de topologia virtual (*Virtual Topology - VT*) permite abstrair os detalhes físicos da rede óptica internos a um determinado domínio. Uma VT é formada por enlaces virtuais que representam as conexões ópticas dentro de um domínio e possuem características específicas de proteção, largura de banda, preço, etc. Tais conexões ópticas são conhecidas como *Forwarding Adjacencies - FAs*) [Farrel et al., 2005]. O estabelecimento das conexões ópticas em cada domínio pode ser feito usando GMPLS, ou qualquer outro conjunto de protocolos, ou de forma manual pelo administrador do domínio. A maneira como tais conexões são estabelecidas é uma decisão local. Este mecanismo é uma das vantagens iniciais do conceito de VT: manter a independência de cada domínio e a não divulgação dos detalhes internos ao domínio.

Nesta tese definimos duas formas para a obtenção da topologia virtual. A primeira considera grupos de domínios organizados na forma de condomínios. Dentro de cada condomínio a divulgação das topologias virtuais ocorre seguindo um modelo de negócios onde cada domínio divulga a VT para todos os outros domínios dentro do mesmo condomínio. Chamamos este modelo de modelo *Push*. O outro modelo considera a hierarquia atual da Internet organizada na forma de *Autonomous Systems* (ASes). Neste caso, as VTs não são divulgadas pelos domínios, mas são obtidas sob demanda seguindo rotas BGP específicas escolhidas pelo domínio que deseja estabelecer uma conexão entre domínios. Chamamos de modelo *Pull* ou *on-demand*. Isto será explicado na Seção 4.1 abaixo.

A divulgação de VTs segue políticas locais de cada domínio e reflete o estado atual da rede. As regiões do domínio óptico cujos recursos estão ociosos podem ser mapeadas em VTs com um custo financeiro baixo de forma a atrair clientes para o domínio. O administrador do domínio também pode fazer "promoções" de VTs levando em conta possíveis demandas futuras por conexões. Um determinado domínio pode definir várias VTs, uma para cada situação e divulgá-las para domínios diferentes seguindo as regras de relacionamento entre os domínios. Uma VT pode ser divulgada para o estabelecimento de VPNs enquanto outra pode ser divulgada para o estabelecimento de conexões

simples com proteção 1+1. A variedade de opções disponíveis e a forma de uso do mecanismo de VT se apresenta como uma ferramenta bastante útil para os administradores dos domínios. Eles podem até mesmo oferecer uma VT cujas conexões ópticas não estejam estabelecidas para um determinado enlace virtual. Desta forma, os recursos ópticos podem ser usados para atender a outras conexões com prioridade baixa podendo ser "preemptadas" futuramente.

A seguir são listadas outras vantagens obtidas com o uso do mecanismo de VT.

- **Configuração e divulgação de VTs utilizando políticas:** Cada domínio óptico pode configurar e divulgar suas VTs utilizando políticas e regras locais. Estas políticas consideram os aspectos de negócios e os relacionamentos comerciais entre os domínios. Os domínios podem definir uma VT padrão que é usada em situações normais e outras VTs que podem ser usadas quando situações específicas são detectadas na rede, por exemplo, falhas na rede, alta demanda por determinados tipos de conexões, etc.;

- **Oferta promocional de VTs:** Os domínios podem oferecer VTs promocionais com custos inferiores aos praticados normalmente. Tais ofertas podem ser feitas em determinados dias da semana ou do mês e normalmente têm um prazo de validade. As promoções também podem ser feitas caso haja ociosidade dos recursos na rede;

- **Oferta de VTs específicas para aplicações específicas:** Os domínios podem oferecer VTs que atendam a demanda específica de alguns clientes. Clientes que necessitam estabelecer VPNs com uma determinada banda ou clientes que precisam de conexões com um certo grau de proteção podem requisitar ao seu domínio este tipo de oferta de serviço específico;

- **Reserva de Recursos:** Uma VT representa um conjunto de conexões ópticas que estão estabelecidas (ou não) sobre cada enlace virtual que forma a VT. A reserva de um recurso em um enlace virtual garante o acesso futuro ao mesmo por quem fez a reserva. Este tipo de mecanismo é bastante utilizado em serviços de VPN. A reserva é feita previamente de forma a garantir que quando a VPN precisa efetivamente ser estabelecida, os recursos estarão disponíveis exclusivamente para o proprietário da VPN. Nesta tese, desenvolvemos e implementamos o serviço de VPN com suporte a reserva de recursos;

- **Re-divulgação de novas VTs:** Na medida em que os recursos são consumidos em cada enlace virtual, o administrador local do domínio poderá disparar a re-divulgação de novas VTs de forma a evitar o bloqueio de conexões. Esta re-divulgação pode ser feita com base em políticas e em limiares específicos para cada VT dependendo das relações entre domínios e dos objetivos locais de cada administrador;

- **O roteamento entre domínios é feito sobre as VTs divulgadas:** Enquanto o BGP divulga apenas informação sobre alcançabilidade de prefixos de rede limitando o roteamento com QoS entre domínios, o mecanismo de VT permite obter um grau de informação maior em relação às possíveis rotas em direção a um determinado destino. Dependendo da quantidade de informações divulgadas (isto será explicado abaixo) em cada VT, o roteamento pode procurar por rotas que atendam a critérios específicos de QoS desejados para as conexões. Além disso, o modelo *Push* permite que todos os domínios dentro do mesmo condomínio tenham uma visão global sobre o roteamento dentro do condomínio;

- **As VTs são vistas como *commodities*[1]:** As VTs não são somente vistas como um conjunto de nós e enlaces. Elas são vistas como bens que podem ser negociados usando interesses comerciais a fim de chamar a atenção dos clientes. Localmente, cada domínio pode usar algoritmos de otimização eficientes baseando-se em matrizes de tráfego de forma a maximizar o uso dos recursos ópticos. Quanto melhor a otimização, melhor será a oferta de VTs;

- **Clientes *Multi-Homed* podem decidir melhor sobre qual provedor usar:** Atualmente, domínios clientes decidem com base nas informações divulgadas pelo BGP e nos relacionamentos que eles mantém com seus provedores qual a "melhor" rota em direção a um determinado prefixo de rede. Com o mecanismo de VT os domínios clientes terão uma visão que vai além de seus provedores podendo escolher uma melhor rota com requisitos de QoS que atendam as suas necessidades específicas.

A Figura 4.1 ilustra o mecanismo de distribuição de VTs. Observe que a mesma rede física é mapeada em duas VTs diferentes (VT1 e VT2). Na rede física também não há conexões entre os nós A e C. Entretanto, nas duas VTs, 1 e 2, os nós A e C passam a ser vizinhos através de um enlace virtual. Esta vizinhança virtual ocorre pois o enlace virtual entre os nós A e C representa as conexões ópticas estabelecidas entre os dois nós na rede. As rotas físicas usadas para o estabelecimento das conexões ópticas entre os nós é abstraída na VT.

Na Figura 4.1 duas VTs diferentes estão sendo divulgadas para domínios diferentes. A quantidade de informações a ser divulgada em cada VT depende dos relacionamentos e contratos previamente estabelecidos entre os domínios. Dependendo da relação, a quantidade de informações pode ser maior ou menor. Neste trabalho, definimos três níveis de divulgação de informações como pode ser observado na Figura 4.2.

O nível mais restrito divulga apenas um custo abstrato em cada enlace virtual (Figura 4.2 (a)). Não há nenhuma outra informação sobre o significado deste custo abstrato. Entretanto, ele deve refletir a QoS do enlace virtual em relação à proteção, banda, BER e preço. A escolha da rota irá levar em

[1] Mercadoria, produtos de consumo.

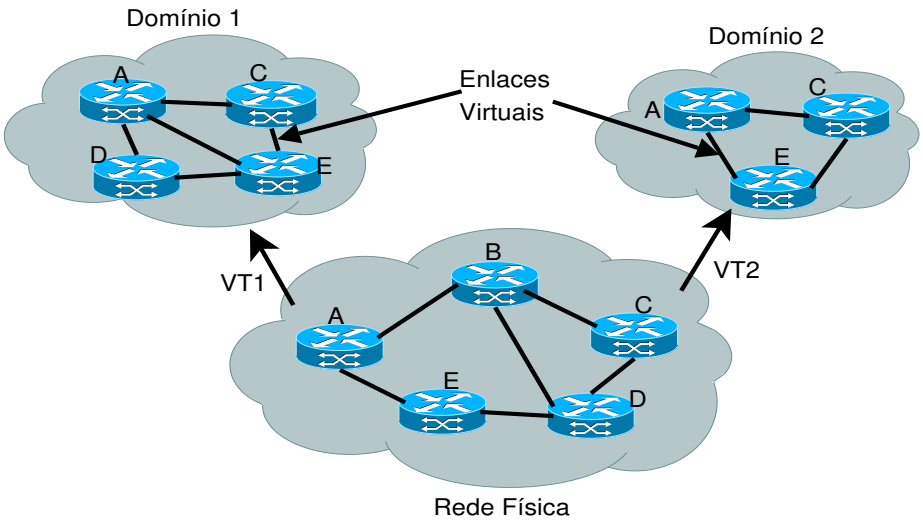


Fig. 4.1: Divulgando VTs.

conta este custo, porém detalhes sobre o que ele significa são obtidos durante a negociação. O nível intermediário associa o custo especificamente a um atributo. Por exemplo, na Figura 4.2 (b), o enlace virtual A-C possui a tupla 4,8 associada a ele. O primeiro valor (4) poderia significar o custo abstrato para obter o que é oferecido pelo segundo valor da tupla (8). O segundo valor poderia representar uma proteção 1:1 ou uma banda de 2.5 Gbps. Neste caso, é necessário uma maneira para descrever o significado de cada atributo que forma a tupla a fim de que os domínios entendam o que está sendo divulgado. Uma solução para isto seria a utilização de *Web Semantic* e ontologias [OWLS, 2006] que poderiam ser usadas para descrever os parâmetros divulgados. Finalmente, o nível menos restrito divulga informações relacionadas a todos os atributos que podem ser relevantes para a escolha de uma rota. No exemplo da Figura 4.2 (c), os enlaces virtuais possuem 4 valores associados. Estes valores representam, em termos gerais, a QoS de cada enlace virtual. Considere o enlace virtual A-C. O valor 8 poderia significar proteção 1:1, o valor 10 o BER, 2.5 seria a banda em Gbps e o valor 15 poderia representar o custo monetário para usar um recurso naquele enlace virtual.

O nível menos restrito de divulgação de informações permite que os domínios calculem suas rotas usando atributos específicos de QoS. Caso algum domínio deseje, por exemplo, proteção 1+1, o algoritmo de roteamento deveria levar em conta somente as possibilidades de rotas que ofereçam tal proteção e então escolher a mais barata. Porém, nem todos os domínios têm interesse em divulgar todos os atributos. O nível mais restrito garante uma maior privacidade para os domínios. Entretanto, o cálculo de rotas ocorre unicamente levando-se em conta um custo abstrato. Normalmente, o caminho mais curto será então considerado sem saber mais detalhes sobre a possibilidade de suportar ou não um determinado atributo de QoS.

Além disso, alguns domínios podem adotar um nível de divulgação de informações, enquanto

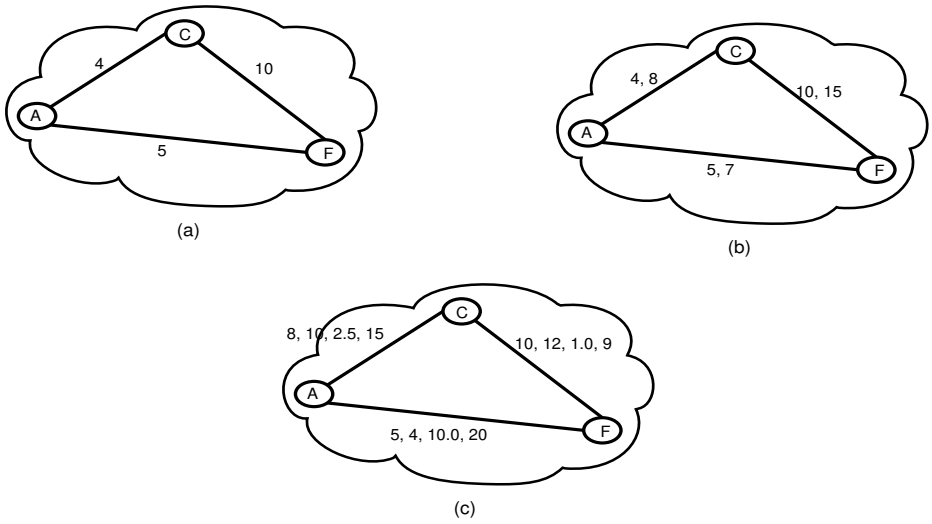


Fig. 4.2: Níveis de Divulgação de Informação.

outros podem adotar outros níveis. Um mesmo domínio pode ainda usar níveis de informações diferentes entre as VTs. A escolha do nível de informações é uma decisão local em cada domínio e depende dos relacionamentos comerciais entre os domínios.

Uma VT é formada por nós físicos, portas virtuais e enlaces virtuais. Uma porta virtual representa uma ou mais portas físicas. Um enlace virtual representa uma ou mais conexões físicas (caminhos de luz). A Figura 4.3 ilustra estes conceitos.

A Figura 4.3 (a) mostra uma rede física formada por enlaces físicos e portas físicas. O nó A possui três portas físicas, cada uma com sua característica específica. Uma porta pode representar uma fibra, um comprimento de onda, um conjunto (*bundle*) de comprimentos de onda ou ainda um conjunto de fibras. Nesta tese nós assumimos que uma porta é uma fibra. Dentro de cada fibra há um conjunto de comprimentos de onda disponíveis e que podem ser usados para o estabelecimento de caminhos de luz. Quando os caminhos de luz são estabelecidos, as diferentes portas físicas são agrupadas para formar uma porta virtual. Como exemplo, considere que três caminhos de luz são criados entre os nós A e C na Figura 4.3 (a). O primeiro caminho de luz usa a rota A, B, C. O segundo usa a rota A, D, C e o terceiro caminho de luz usa a rota A, E, F, G, C. Estas três portas físicas são agrupadas e formam a porta virtual ilustrada na Figura 4.3 (b), e os três caminhos de luz criados são representados pelo enlace virtual 4 na Figura 4.3 (c). Se um quarto caminho de luz fosse criado, por exemplo usando a rota A, D, F, G, C, a porta virtual não seria alterada uma vez que apenas mais um comprimento de onda da porta física 2 estaria sendo usado.

Em termos de recursos, o enlace virtual 4 possui três recursos disponíveis, i.e., três caminhos de luz que podem ser usados. Especificamente para o enlace virtual 4, três conexões podem ser estabelecidas. Como mencionado anteriormente, cada domínio é responsável pela gerência dos

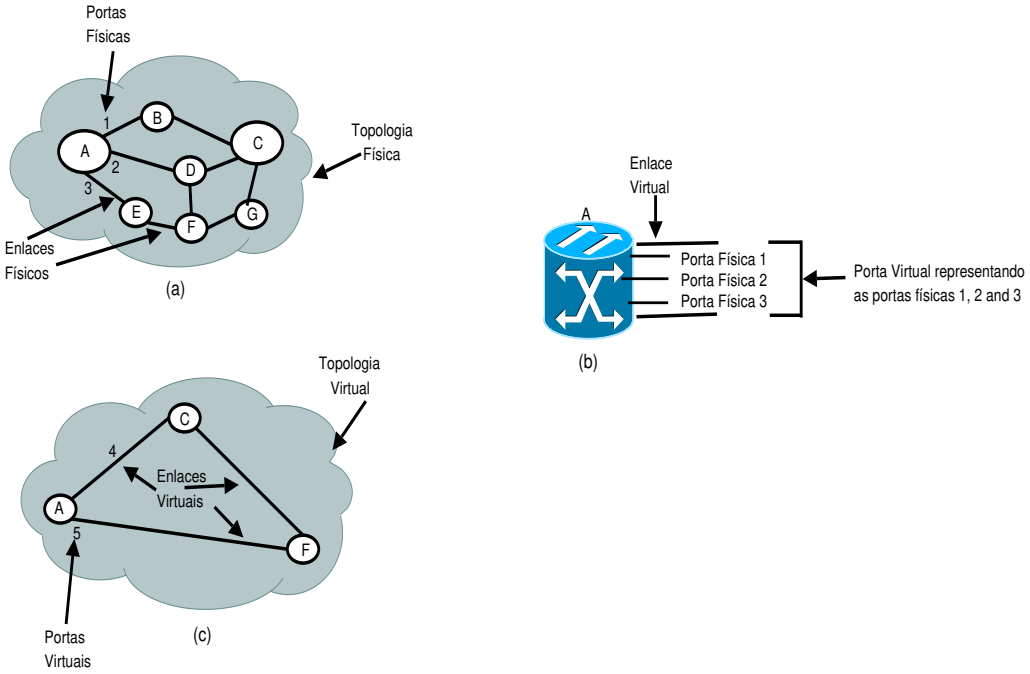


Fig. 4.3: Portas e Enlaces Virtuais.

recursos locais. O administrador do domínio poderia definir políticas para a criação de outros caminhos de luz entre A e C caso haja demanda. Da mesma forma, uma nova VT poderia ser divulgada sem o enlace virtual 4 caso não fosse possível a criação de novos caminhos de luz entre A e C.

4.1 Como as VTs são Obtidas

Neste trabalho consideramos dois tipos de mecanismos para obtenção das VTs. O primeiro modelo denominado *Push* considera que cada domínio divulga (*advertise*) suas VTs para os outros domínios. Este modelo caracteriza melhor o conceito de *commodities* uma vez que cada domínio oferece VTs como produtos a serem comprados num mercado de negócios. O segundo modelo é denominado modelo *Pull* e funciona de forma contrária ao modelo *Push*. O modelo *Pull* também pode ser visto como um modelo por demanda (*on-demand*) pois as VTs são obtidas pelos domínios na medida em que eles precisam delas para o estabelecimento de serviços entre domínios. Um dado domínio que deseje estabelecer uma conexão entre domínios usa as rotas BGP como ponto inicial para obtenção de VTs unicamente naquelas rotas.

4.1.1 O Modelo *Push*

Neste modelo as VTs são divulgadas para todos os domínios. Cada domínio óptico terá as VTs de todos os outros domínios obtendo assim uma visão global sobre a topologia fim-a-fim de qualquer nó fonte para qualquer nó destino. A Figura 4.4 ilustra este cenário.

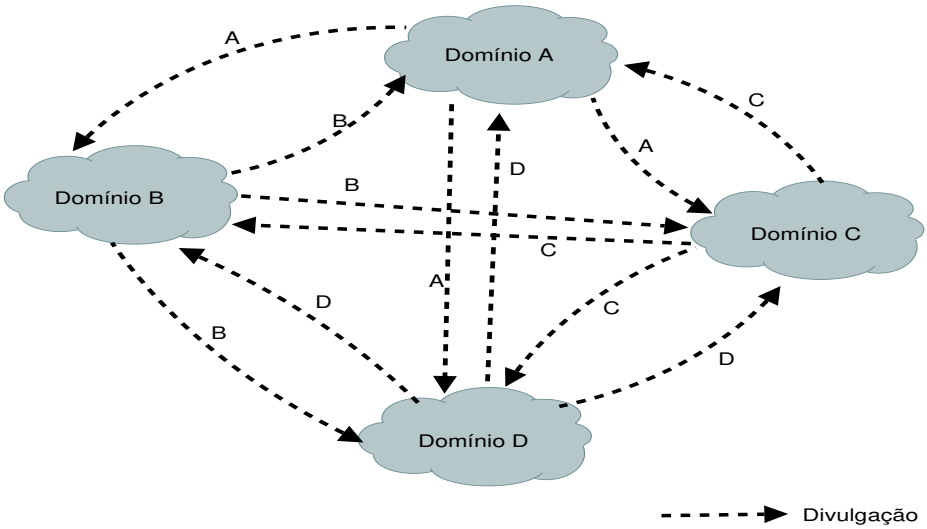


Fig. 4.4: Obtendo VTs (Modelo *Push*).

Note que todos os domínios da Figura 4.4 divulgam suas topologias para os outros domínios. No modelo *Push* as relações BGP são enfraquecidas. A divulgação de uma VT significa que o domínio aceita receber conexões sobre uma topologia diferente daquela divulgada pelo BGP. O modelo *Push* considera que as relações entre domínios são mais confiáveis semelhantemente ao modelo de pares (*peering*) atualmente usado pelo BGP. O modelo de pares considera um acordo entre domínios pelo qual tráfegos dos próprios domínios e de seus clientes são transferidos sem custo para os domínios envolvidos.

O modelo *Push* pressupõe a existência de condomínios de domínios. Este modelo vem sendo analisado principalmente pelo grupo da rede CANARIE no Canadá [CANARIE Project, 2006]. Tal grupo considera a formação de condomínios de escolas, universidades, municípios, bairros, etc. Alguns exemplos de organização relacionados com condomínios podem ser encontrados em [CivicNet, 2001, Stokab, 2006]. Dentro desses condomínios, a divulgação das VTs é feita entre todos os domínios. Uma maneira possível para a definição dos condomínios seria organizar as relações cliente-provedor para serem relações de pares (*peering*). A construção destes condomínios poderia levar em conta aspectos geográficos e aspectos de negócios. Além disso, poderia haver um segundo nível de distribuição de topologias virtuais que seria criado entre os condomínios. Neste caso, cada domínio dentro de um condomínio seria visto como um nó e o condomínio seria visto como um domínio. A quantidade de informações a ser divulgada entre condomínios seria a mesma quantidade

de informações divulgada entre domínios porém, não mais no nível de nós físicos, mas sim, no nível de nós virtuais que representam os domínios em cada condomínio. Este modelo poderia ser hierarquicamente desenvolvido de forma recursiva para níveis mais altos, caso seja necessário. Nesta tese, estamos considerando apenas um nível de hierarquia, ou seja, a divulgação de topologias virtuais ocorre entre domínios dentro de um condomínio.

4.1.2 O Modelo *Pull*

Diferentemente do modelo *Push*, o modelo *Pull* não divulga as VTs para outros domínios. O mecanismo por demanda espera que os domínios interessados em obter as VTs perguntem para um número limitado de opções de rotas. O modelo *Pull* usa as tabelas BGP para obter possíveis rotas para um determinado destino. Quando um domínio deseja estabelecer uma conexão entre domínios, primeiramente ele obtém as rotas locais divulgadas pelo BGP que alcancem um destino desejado. Com estas rotas, o domínio local obtém as VTs apenas dos domínios que compõem as rotas BGP e então calcula um caminho até o destino. A Figura 4.5 ilustra o cenário *Pull*.

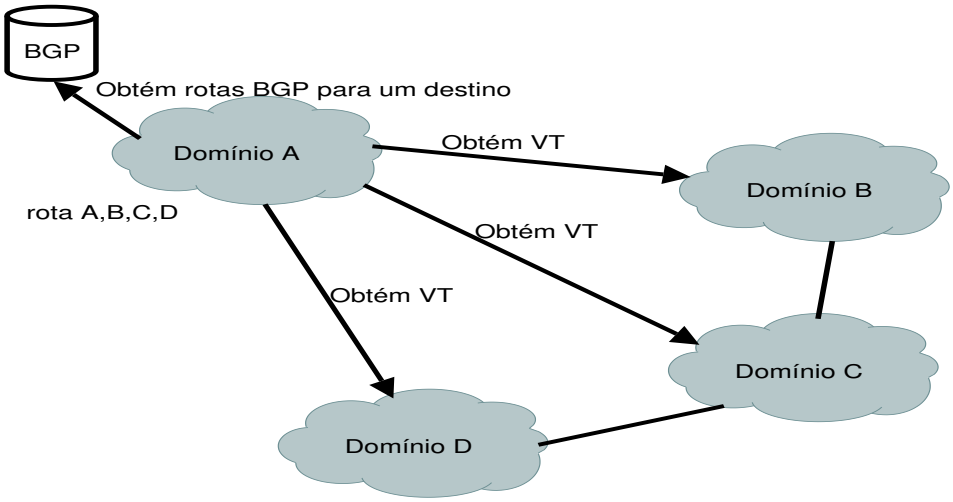


Fig. 4.5: Obtendo VTs (Modelo *Pull*).

Suponha que o domínio A deseje estabelecer uma conexão com o domínio D. Para isto, o domínio A obtém das tabelas BGP locais as rotas que alcancem D. A quantidade de rotas a ser obtida depende da quantidade de provedores aos quais o domínio A está conectado (*Multi-homing*). Neste exemplo, o domínio A possui apenas o domínio B como provedor e, portanto, apenas uma rota para D é obtida. Após obter esta rota, o domínio A deve perguntar para cada domínio da rota a VT para o próximo domínio em direção ao destino. No exemplo da Figura 4.5, o domínio A obtém do domínio B a VT que conecta o domínio B com o domínio C para chegar em D. Esta operação é feita para cada domínio na rota obtida.

Se, através das rotas BGP locais nenhum caminho que atenda os requisitos de QoS é encontrado, o domínio local pode "subir" um nível na hierarquia de ASes e obter as rotas BGP não divulgadas pelos seus provedores. Lembre-se que o BGP apenas divulga a "melhor" rota em direção a cada prefixo de rede. A melhor rota é uma decisão local de cada AS e na maioria das vezes não possui nenhum significado para o AS que recebe o anúncio da rota BGP. Ao subir um nível na hierarquia de ASes, o domínio obtém outras rotas BGP que alcancem o destino e que podem ser usadas para obtenção de VTs e posterior cálculo de caminhos. A Figura 4.6 ilustra um exemplo simples deste cenário.

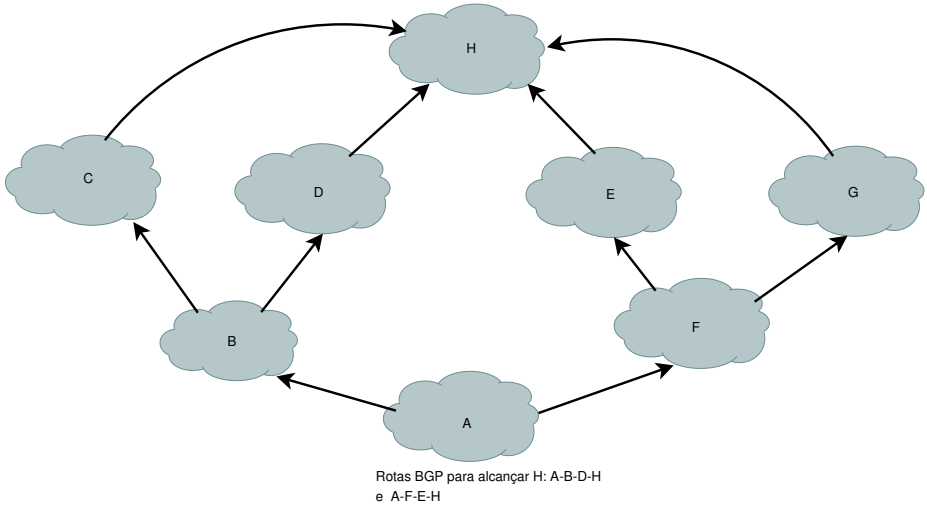


Fig. 4.6: Subindo na hierarquia de ASes para obter mais VTs (Modelo *Pull*).

O domínio A possui dois provedores, B e F. Suponha que o domínio A precise estabelecer uma conexão com o domínio H. Suas tabelas BGP indicam duas rotas possíveis: A-B-D-H e A-F-E-H. Considerando que depois de obter as VTs destas duas rotas, o domínio A não consegue achar um caminho que atenda seus requisitos de QoS, o domínio A pode subir um nível e perguntar aos seus dois provedores por outras rotas BGP que alcancem H e que não foram divulgadas para o domínio A. O domínio A obtém duas novas alternativas: A-F-G-H e A-B-C-H. Neste instante, o domínio A pode obter as VTs dos domínios que compõem estas duas novas rotas. Obviamente, o domínio pode ainda não obter um caminho que satisfaça seus requisitos de QoS.

O modelo *Pull* é uma alternativa que se integra muito bem ao roteamento da Internet atual e que utiliza o BGP como protocolo de roteamento entre domínios. Ao mesmo tempo, o modelo *Pull* se beneficia da forma como a Internet está organizada atualmente. Hierarquicamente, a Internet está dividida em cinco camadas, cada uma com uma certa quantidade de ASes [Subramanian et al., 2002]. O nível mais alto (nível 0), chamado de *core* possui 20 ASes. A camada de trânsito (nível 1) possui 129 ASes. A camada conhecida como *outer core* (nível 2) possui 897 ASes. A camada formada por pequenos provedores de acesso (nível 3) possui 971 ASes. Finalmente, a camada mais inferior (nível

4) da hierarquia é formada pelos ASes clientes (*customers*) também conhecidos como *stubs*. Esta camada possui a maioria dos ASes, 8898. Estes números correspondem ao ano de 2002. Atualmente, a Internet possui aproximadamente 17000 ASes e a quantidade de ASes em cada camada deve ser maior para refletir os números atuais.

Através desta divisão em camadas, outros estudos observaram que na medida em que subirmos os níveis na hierarquia, a quantidade de rotas alternativas aumenta devido ao fato de que o grau de conexões entre os ASes aumenta [Subramanian et al., 2002]. Como um exemplo, há 2409 arestas conectando o nível 3 com o nível 4 e 3752 arestas conectando o nível 2 com o nível 3. Este mesmo estudo revelou que existe um grande número de conexões entre as camadas intermediárias na hierarquia.

O exemplo ilustrado na Figura 4.6 tem como objetivo apresentar o mecanismo. Porém, a quantidade de rotas alternativas obtidas ao subirmos um nível na hierarquia é, conforme comentado acima, satisfatoriamente grande para atender os domínios do nível imediatamente inferior. Obviamente, os provedores podem usar políticas locais de forma a divulgar seletivamente as rotas BGP. Além disso, após obterem as rotas BGP de seus provedores, os domínios clientes podem também, com base em políticas, selecionar apenas algumas rotas para obtenção das VTs. Isto normalmente pode ser feito caso a quantidade de rotas para um determinado destino seja demasiadamente grande [Verdi et al., 2006c].

4.1.3 Comparando os dois Modelos

Algumas vantagens do modelo *Push* são:

- Quando um domínio deseja estabelecer uma conexão, as VTs dos outros domínios já estão disponíveis uma vez que elas foram divulgadas previamente. Não é necessário obtê-las no momento em que a requisição para o estabelecimento é feita;
- O modelo *Push* é mais voltado para negócios permitindo a divulgação de VTs promocionais de forma rápida em uma tentativa de adquirir clientes;
- O modelo *Push* é mais dinâmico pois permite divulgar outras VTs caso algum evento local ao domínio ocorra, como por exemplo uma falha.

Algumas desvantagens do modelo *Push* são:

- É mais futurista. Ele exige que condomínios de ASes sejam estabelecidos de forma a criar um nível de confiança que permita a divulgação das VTs entre todos os membros do condomínio. Isto deveria ser feito de forma incremental num cenário real;

- Gera mais tráfego na rede, muitas vezes de forma desnecessária. A divulgação das VTs ocorre sem saber se elas serão usadas pelos outros domínios;

- Pode causar problema de escalabilidade num cenário real. Não podemos considerar que todos os ASes estarão agrupados num único condomínio e que as VTs estejam sendo divulgadas entre todos. Para resolver isto, os condomínios poderiam ser definidos de forma incremental e então um modelo hierárquico entre os condomínios poderia ser definido. Algo semelhante ao apresentado em [Li and Mohapatra, 2004]. Nesta tese consideramos apenas um nível hierárquico, ou seja, apenas um condomínio.

Algumas vantagens do modelo *Pull* são:

- A principal vantagem é a sua simplicidade que permite a aplicação sem restrições a um cenário real. A integração com o BGP não exige nenhuma mudança na forma como o roteamento é feito atualmente. O modelo *Pull* leva em consideração a forma como a Internet é organizada hoje e por isso poderia ser usado dentro de um curto ou médio prazo;

- Não há problemas de escalabilidade. As VTs são obtidas por demanda considerando apenas algumas rotas BGP. Não há divulgação de VTs entre todos os pares de domínios;

- Não requer a definição de condomínios. Basta apenas a definição da camada de serviços para obtenção de VTs e negociação de contratos.

Algumas desvantagens do modelo *Pull* são:

- É necessário mais tempo para o estabelecimento de uma conexão uma vez que as VTs não estão disponíveis no momento da requisição. As rotas BGP precisam ser obtidas para então requisitar as VTs para o cálculo do caminho. Uma análise de tempo foi feita para esta tese a fim de verificar as diferenças entre os modelos;

- É menos orientado a negócios. Não há um mecanismo para divulgar as VTs levando-se em conta promoções e o estado atual da rede;

- É menos dinâmico. Um domínio não pode divulgar novas VTs caso algum evento interno ao domínio ocorra. Em casos específicos de falhas, o domínio pode enviar uma mensagem de notificação para os domínios que estejam usando as VTs a fim de que outra VT seja obtida de forma a contornar a falha.

Finalizações do capítulo

Nos próximos dois capítulos, apresentamos a instanciação do modelo discutido no Capítulo 3. No Capítulo 5 a seguir, detalhamos a instanciação do modelo em uma arquitetura para o provisionamento e gerência de serviços dentro de um domínio. A evolução da arquitetura como uma instância do modelo para o provisionamento de serviços entre domínios é apresentada no Capítulo 6.

Capítulo 5

Instanciação do Modelo para Provisionamento e Gerência de Serviços dentro de um Domínio

Este capítulo descreve a instanciação do modelo apresentado no capítulo anterior para o provisionamento e gerência de serviços em um domínio. A instanciação do modelo deu origem a uma arquitetura que, através da implementação de um protótipo, permitiu avaliar as funcionalidades definidas para o modelo. Este capítulo é resultado de trabalhos individuais que deram origem a três dissertações de mestrado e um conjunto de artigos. Dividimos este capítulo em duas seções. A primeira (Seção 5.1) apresenta a arquitetura e seus módulos para o provisionamento de serviços dentro de um domínio. Esta seção deu origem à dissertação de mestrado referenciada em [Duarte, 2006] assim como os artigos referenciados em [Verdi et al., 2005b, Verdi et al., 2006a]. A segunda seção (Seção 5.2) apresenta os outros dois trabalhos. Neste sentido, especial atenção foi dedicada para o uso de políticas para agregação de tráfego de redes clientes na rede de transporte. O trabalho desenvolvido com políticas de agregação teve como resultado uma dissertação de mestrado [Carvalho, 2006] e um conjunto de artigos citados em [Verdi et al., 2004, Verdi et al., 2005a, Carvalho et al., 2005, Carvalho et al., 2006]. Finalmente, a arquitetura incorporou funcionalidades para o provisionamento de VPNs em um domínio. O serviço de VPN originou a dissertação de mestrado referenciada em [Malheiros, 2006] e os artigos citados em [Malheiros et al., 2006a, Malheiros et al., 2006b]. Esta segunda seção apresenta um resumo e alguns resultados obtidos nestes dois últimos trabalhos.

5.1 Definição da Arquitetura

A definição da arquitetura para provisionamento e gerência de serviços dentro de um domínio tem como foco principal a definição dos módulos que fazem parte da camada de gerência de redes do modelo TMN. Para esta fase, a camada de serviços possui apenas um serviço denominado *End-to-End Connection Service* (E2ECS), que oferece acesso às funcionalidades para o provisionamento e gerência de SPCs e VPNs. Os outros módulos da arquitetura instanciam as funções locais do modelo apresentado no capítulo anterior e pertencem ao plano de gerência do modelo ASON [Verdi et al., 2005b]. A Figura 5.1 mostra a arquitetura desenvolvida para esta fase [Verdi et al., 2005b, Duarte, 2006].

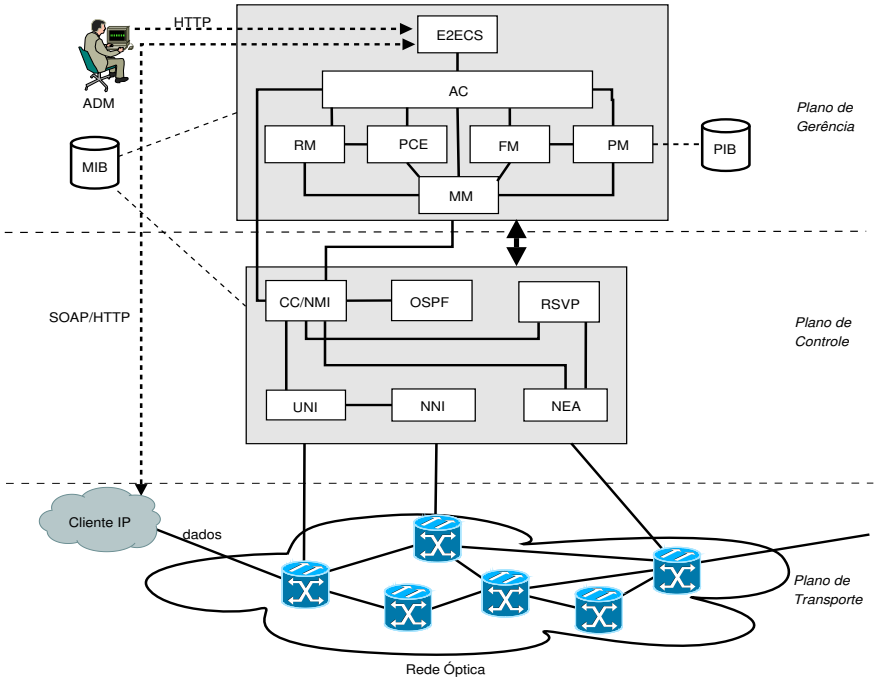


Fig. 5.1: Arquitetura para Provisionamento e Gerência de Serviços dentro de um Domínio.

Note que os três planos do modelo (abstrato) ASON apresentados no capítulo anterior são instanciados pela arquitetura definida nesta tese. O plano de gerência é composto pelos módulos que realizam o controle de admissão, a gerência de falhas, a gerência de recursos, a aplicação de políticas, a configuração, a gerência de desempenho e gerência de contabilidade. Cada funcionalidade citada é implementada por um ou mais módulos do plano de gerência. Muitas vezes, algumas funcionalidades também precisam do plano de controle para serem realizadas, como por exemplo, o estabelecimento e remoção de conexões e a gerência de falhas.

O plano de controle, instanciado através dos protocolos da arquitetura GMPLS, é responsável pela sinalização de SPCs, roteamento e interfaceamento com a rede cliente para provisionamento

das *Switched Connections* (SCs). O plano de transporte é instanciado através de uma rede óptica de comunicação. Abaixo, definimos as funcionalidades específicas de cada módulo em seus respectivos planos.

5.1.1 Módulos do Plano de Controle

- RSVP - *Resource Reservation Protocol*: protocolo de sinalização escolhido para este trabalho. Utilizado para estabelecer e remover LSPs, já vem sendo usado na arquitetura IntServ [Braden et al., 1994] e MPLS [Rosen et al., 2001] para a reserva de recursos. Foi estendido para suportar a arquitetura GMPLS [Berger, 2003]. O RSVP também é responsável por notificar o módulo CC/NMI (*Connection Controller/Network Management Interface*) sobre as ocorrências de falhas na rede óptica. As falhas de nós ou enlaces na rede de transporte são notificadas ao RSVP através de um protocolo encarregado pelo gerenciamento dos enlaces (por exemplo, LMP). Estas notificações são enviadas ao RSVP na forma de eventos. O RSVP pode então notificar o plano de gerência usando a mensagem de *notify* especificada para a arquitetura GMPLS;

- OSPF - *Open Shortest Path First*: protocolo de roteamento baseado em estado de enlaces utilizado em redes IP e que foi estendido [Kompella and Rekhter, 2005c] para suportar as capacidades do GMPLS;

- UNI e NNI - *User-to-Network Interface e Network-to-Network Interface*: a UNI é uma interface composta por dois módulos. O primeiro módulo localizado no lado cliente (UNI-C) envia as mensagens de sinalização para o lado servidor (UNI-N) localizado na rede de transporte. A UNI é tipicamente utilizada em cenários *overlay* onde o provedor age como um servidor oferecendo serviços para as redes clientes (redes IP/MPLS por exemplo). A interface NNI é utilizada para o trânsito das mensagens de sinalização dentro de um mesmo domínio e entre domínios administrativos diferentes. O RSVP tem uma forte relação com as interfaces UNI e NNI uma vez que as mensagens RSVP são transferidas através destas interfaces;

- NEA - *Network Element Agent*: este módulo faz parte dos elementos na borda da rede e tem por finalidade realizar a crossconexão em cada comutador óptico (OXC) localizado na fronteira do domínio óptico. Este módulo foi desenvolvido para representar a "crossconexão"nos OXCs agindo como uma tabela de encaminhamento para uma mensagem PATH sair de um caminho de luz e entrar em outro;

- CC/NMI - *Connection Controller/Network Management Interface*: o CC/NMI possui duas interfaces. A primeira (CC) é utilizada pelo plano de controle para enviar eventos (por exemplo,

notificações de falhas) ao plano de gerência. A segunda (NMI) é utilizada pelo plano de gerência para acessar os módulos do plano de controle. A interface NMI define os métodos que o plano de gerência pode invocar no plano de controle para o estabelecimento e remoção de SPCs.

5.1.2 Módulos do Plano de Gerência

- AC - *Admission Control*: é o ponto de entrada para os módulos do plano de gerência. É encarregado de receber as requisições para o estabelecimento e remoção de conexões fim-a-fim e encaminhá-las aos outros módulos do sistema para que sejam processadas. Além disso, é responsável principalmente por verificar os contratos pré-definidos (SLAs - *Service Level Agreements*) sobre as requisições recebidas. O AC, sempre que necessário, acessa os outros módulos para aplicação de políticas (*Policy Manager*, PM), consulta de recursos (*Resource Manager*, RM), verificação de membros de VPNs (*Membership Manager*, MM) e controle de falhas (*Fault Manager*, FM). O AC pode também receber notificações do plano de controle através do módulo CC/NMI que resultarão na invocação dos módulos necessários para gerenciar o evento ocorrido. Os SLAs definidos para o AC neste trabalho têm por finalidade autorizar a criação e remoção de caminhos ópticos e VPNs ópticas através da verificação das permissões de usuários e da consistência dos dados das requisições;

- PM - *Policy Manager*: o módulo PM, também conhecido como PDP (*Policy Decision Point*), é o responsável por aplicar políticas que foram definidas para o domínio que está sendo gerenciado. Um repositório de políticas conhecido como PIB (*Policy Information Base*) é utilizado para armazenar políticas principalmente relacionadas ao *grooming*, autenticação e criação de VPNs. O módulo PM foi implementado em [Verdi et al., 2004, Verdi et al., 2005a, Carvalho et al., 2005, Carvalho et al., 2006, Carvalho, 2006] para aplicação das políticas de *grooming* e também em [Malheiros et al., 2006a, Malheiros et al., 2006b, Malheiros, 2006] para a aplicação das políticas de criação de VPNs. A próxima seção (Seção 5.2) apresenta estes dois trabalhos;

- FM - *Fault Manager*: o módulo FM tem a função de receber as notificações de falhas enviadas pelo plano de controle e manter a gerência atualizada sobre as alterações ocorridas no estado da rede, as quais foram desencadeadas pela ocorrência de uma falha. O módulo FM foi desenvolvido no trabalho [Carvalho, 2006];

- RM - *Resource Manager*: é responsável por gerenciar todos os recursos da rede de transporte como por exemplo, pontos de *grooming*, caminhos de luz estabelecidos (SPCs e SCs), VPNs

ativas, topologias virtuais, etc;

- PCE - *Path Computation Element*: é responsável por encontrar uma rota para o estabelecimento de conexões intra e inter-domínios através da aplicação de um algoritmo de roteamento. Nesta primeira fase, o PCE é um módulo interno responsável pelo cálculo de rotas dentro do domínio. Na fase seguinte, onde o foco passa a ser o oferecimento de serviços entre domínios, o PCE torna-se um módulo *Web Service* que pode ser acessado por outras entidades externas ao domínio. Pode inclusive representar o roteamento de uma determinada região formada por vários domínios;

- MM - *Membership Manager*: é o módulo que gerencia as informações sobre quais membros pertencem a cada VPN. Essas informações podem ser fornecidas ao sistema de forma estática (configuração). No caso de uma arquitetura distribuída, as informações sobre membros de VPNs podem ser compartilhadas de forma automática, por meio de um mecanismo de *VPN Membership Auto-Discovery*, por exemplo, utilizando-se o protocolo BGP, como descrito em [Ould-Brahim et al., 2006a, Takeda et al., 2005];

- E2ECS - *End-to-End Connection Service*: é um módulo *Web Service* que expõe uma interface de acesso ao sistema de gerência. É através dele que as funcionalidades do sistema são invocadas pelo administrador e pelos clientes do domínio. Administradores invocam a interface para realizarem tarefas de gerência, enquanto clientes invocam a interface para solicitarem serviços de estabelecimento de conexões e VPNs. Os clientes também podem invocar este módulo para obterem informações sobre os seus serviços, como por exemplo, conexões estabelecidas, VPNs configuradas, etc.

Além dos módulos do plano de gerência, instanciamos o plano de controle com os protocolos que fazem parte da arquitetura GMPLS. O estabelecimento das SPCs é realizado através do simulador GLASS [GLASS, 2006] que implementa os protocolos da arquitetura GMPLS. A interação do plano de gerência com o plano de controle ocorre através de módulos que estão localizados no plano de controle e que permitem o estabelecimento e a remoção de SPCs.

Os módulos RSVP, OSPF, UNI e NNI, pertencem ao simulador GLASS e são utilizados pela arquitetura para instanciação do plano de controle. Os módulos NEA e CC/NMI são módulos que foram desenvolvidos para o plano de controle a fim de realizar tarefas que dão suporte ao plano de gerência. Os módulos RSVP e OSPF citados acima correspondem aos módulos RSVP-TE e OSPF-TE com extensões para prover as capacidades do GMPLS.

O módulo NEA foi especificamente utilizado após realizarmos uma adaptação no simulador GLASS a fim de criarmos um cenário de sinalização entre domínios. O cenário era formado por um

conjunto de máquinas Linux sendo que cada máquina possuía uma instância do simulador GLASS (Figura 5.2). Cada máquina representa um domínio administrativo independente.

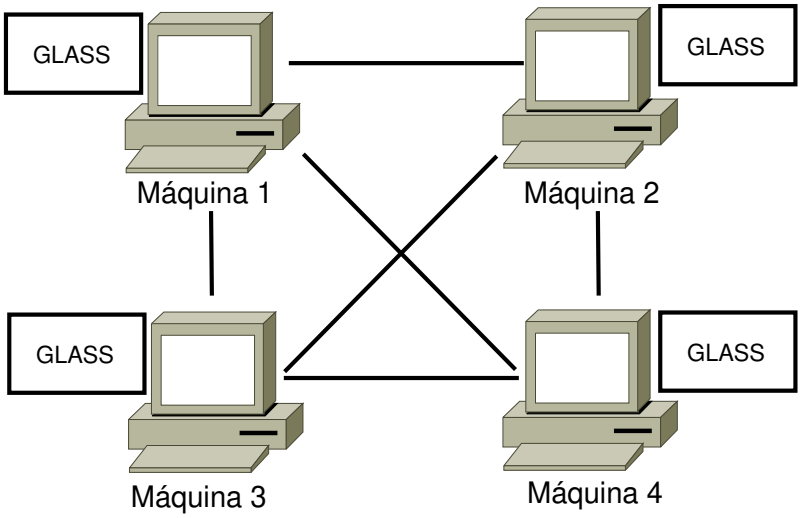


Fig. 5.2: Cenário com quatro máquinas executando o simulador GLASS.

No exemplo da Figura 5.2, uma mensagem de sinalização PATH é originada no primeiro domínio com destino a um nó pertencente a outro domínio. Para alcançar o destino, a mensagem PATH deve atravessar uma seqüência de domínios e voltar na forma de mensagem RESV. A intenção com este experimento foi averiguar as dificuldades e os desafios para a construção de uma interface para sinalização entre domínios, algo similar à E-NNI. Identificamos que não está claro quais informações precisam ser trocadas para o estabelecimento de uma conexão entre domínios, principalmente informações relacionadas ao roteamento. Aspectos envolvendo contratos acabam não sendo considerados neste nível de sinalização. Não se pode negociar atributos, muito menos agendar serviços para futura ativação. Não há possibilidade de reserva de recursos. Este trabalho investigativo confirmou a idéia de que algumas funcionalidades, como as citadas acima, poderiam ou deveriam ser, em um primeiro momento, implementadas num plano superior de gerência e de serviços.

Os módulos da arquitetura apresentados acima são formados por um conjunto de classes e interfaces. A forma como estes elementos se relacionam na arquitetura está demonstrada através de diagramas de classes segundo a especificação UML (*Unified Modeling Language*) e apresentada na Figura A.1 do Apêndice A. Na Figura 5.3, apresentamos o diagrama de seqüência para estabelecimento de uma SPC usando os módulos do plano de controle e do plano de gerência apresentados acima. A seguir, a descrição dos passos do diagrama de seqüência:

1. O gerente (ADM), ou um cliente, envia uma requisição SPC para o E2ECS utilizando uma interface *web*, ou uma aplicação cliente *Web Service* com todas as informações necessárias para iniciar o estabelecimento da SPC;

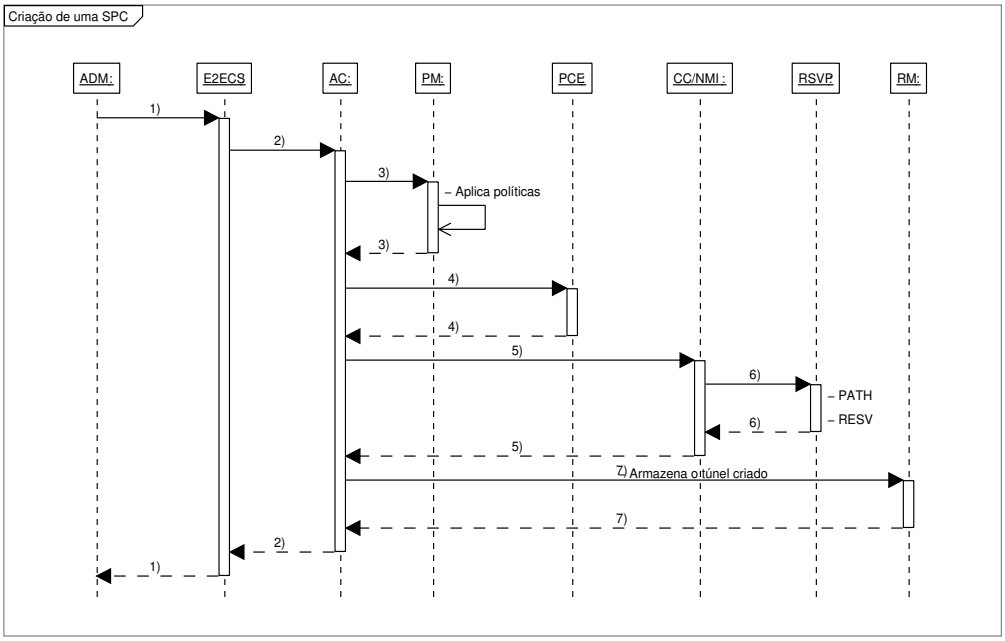


Fig. 5.3: Estabelecendo uma SPC.

2. O E2ECS encaminha a requisição para o AC. Nenhum tipo de validação ou controle é feito pelo E2ECS. Entretanto, uma verificação preliminar dos parâmetros pode ser feita neste ponto;

3. O AC, após receber a requisição, faz uma validação dos dados do usuário e da requisição. Basicamente neste trabalho, os parâmetros são analisados a fim de verificar se o requisitante tem autorização para estabelecer a SPC. Após esta verificação, o AC invoca o módulo PM a fim de aplicar políticas na requisição. Neste cenário, as políticas são simples e apenas autorizam ou não o estabelecimento da SPC entre os nós informados na requisição;

4. O AC invoca o PCE para obter uma rota entre os dois nós informados na requisição;

5. O AC faz uma invocação ao módulo CC/NMI no plano de controle para que a sinalização seja iniciada. O resultado desta sinalização, em caso de sucesso, é o retorno de um objeto do tipo *Tunnel* que identifica o caminho óptico criado via sinalização no plano de controle;

6. No CC/NMI, os parâmetros da requisição SPC são enviados para o RSVP a fim de iniciar a sinalização do caminho de luz. Primeiramente, se uma rota explícita não é passada ao RSVP, então este invoca um módulo de roteamento do próprio GLASS para encontrar uma rota baseada nos nós de ingresso e egresso do domínio óptico. Com a rota do caminho, é iniciada a sinalização com as mensagens de PATH e RESV juntamente com os dados da requisição;

7. O CC/NMI retorna um objeto do tipo *Tunnel* ao AC que é então armazenado no RM. Estes caminhos de luz criados serão posteriormente usados para agregação de fluxos IP/MPLS através da aplicação de políticas.

As informações de entrada necessárias para o estabelecimento de uma SPC são as seguintes:

- Nó e interface de ingresso: o nó cliente MPLS de ingresso e a sua interface;
- Nó e interface de egresso: o nó cliente MPLS de egresso e a sua interface;
- Nó de ingresso do domínio óptico;
- Nó de egresso do domínio óptico;
- Taxa: a largura de banda do caminho óptico;
- Nível de proteção: 1+1, 1:1, 1:N, sem proteção.

A interface Web desenvolvida para a criação da SPC pela qual as informações acima são inseridas está localizada na Figura B.1 do Apêndice B. O trecho WSDL que corresponde ao método do serviço invocado está na Figura C.1 do Apêndice C.

Nesta seção apresentamos a arquitetura e os módulos localizados no plano de controle e no plano de gerência. As funcionalidades dos módulos do plano de gerência foram descritas e detalhamos como uma SPC é estabelecida. O estabelecimento das SPCs permite que os tráfegos das redes clientes do domínio óptico sejam posteriormente agregados nestes caminhos de luz. A forma como este tráfego é agregado reflete no grau de utilização dos caminhos de luz.

No escopo local, ou seja, em um domínio, o uso de políticas para agregação de tráfego a fim de maximizar o uso dos recursos e diminuir o impacto de uma falha na rede óptica teve grande importância e destaca o papel do módulo PM na arquitetura. Como um extenso trabalho com políticas foi realizado, a próxima seção é dedicada a apresentar tais políticas, alguns resultados e os módulos utilizados para aplicação das políticas. A próxima seção é dividida em duas partes. A primeira subseção discute resumidamente as políticas definidas para a agregação de tráfego e alguns resultados obtidos com a aplicação de tais políticas. Na subseção seguinte apresentamos o provisionamento de VPNs e alguns resultados obtidos considerando a aplicação de políticas para gerenciamento de serviços L1VPN. O objetivo da próxima seção é apresentar de forma resumida alguns resultados obtidos com a arquitetura proposta para provisionamento e gerência de serviços em um domínio. Os detalhes de implementação, assim como outros resultados, encontram-se nas referências citadas.

5.2 Políticas de *Grooming* e Provisionamento do Serviço L1VPN

5.2.1 Políticas de Grooming

Como mencionado anteriormente, a arquitetura definida para a fase intra-domínio é formada por diferentes módulos do plano de gerência cujas funcionalidades foram exploradas de forma individual dando origem a diferentes trabalhos. O desenvolvimento de alguns destes módulos foi bastante influenciado pelo uso de políticas para agregação (*traffic grooming*) de fluxos IP/MPLS dentro dos caminhos de luz estabelecidos na rede óptica (SPCs). Grande ênfase foi dada para políticas simples de agregação [Verdi et al., 2004, Verdi et al., 2005a], assim como para políticas mais complexas que levam em consideração falhas na rede óptica [Carvalho et al., 2005, Carvalho et al., 2006, Carvalho, 2006].

As políticas simples consideram a existência de dois tipos de tráfego cliente: *High Priority-HP* e *Low Priority-LP*. Tráfegos HP possuem prioridade sobre tráfegos LP e sempre que for necessário, uma requisição HP pode remover um tráfego LP a fim de ser alocada em um caminho de luz. Outro aspecto importante é a possibilidade de que tráfegos HP podem solicitar mais banda após serem agregados em um caminho de luz [1]. Esta solicitação por mais banda pode causar a remoção (“preempção”) de tráfegos LP para acomodar o tráfego HP no caminho de luz. Remover um tráfego e tentar alocá-lo em outro caminho de luz é um processo demasiadamente custoso para a rede óptica. Assim, busca-se evitar este processo de “preempção” através do uso de políticas de agregação.

Resumidamente, as políticas simples definidas foram as seguintes: considere R uma requisição e L um caminho de luz. Se R é HP, a **política 1** acomoda R em L se L não está vazio e possui apenas tráfego HP. Se R é HP, a **política 2** acomoda R em L se L não está vazio e possui os dois tipos de tráfego. Se R é HP, a **política 3** acomoda R em L se L está vazio. Se R é LP, a **política 4** acomoda R em L se L está vazio. Se R é LP, a **política 5** acomoda R em L se L não está vazio e possui apenas tráfego LP. Se R é LP, a **política 6** acomoda R em L se L possui os dois tipos de tráfego. Variações na ordem da aplicação das políticas foram testadas a fim de encontrar a melhor ordem. A ordem apresentada é a ordem que apresentou os melhores resultados pela qual os gráficos foram gerados.

A Figura 5.4 representa o cenário pelo qual todos os LSPs HPs solicitam um aumento de banda de 50% em relação a banda atual. Neste cenário, a rede óptica está com 66% de tráfego HP e 33% de tráfego LP. O eixo X representa a capacidade da rede e o eixo Y a quantidade de tráfego LP removido.

Para a obtenção dos resultados, geramos 200 requisições cujas bandas variavam entre 50Mb/s e 400Mb/s. Cada caminho de luz possui 1Gb/s de banda. A capacidade da rede varia de 10Gb/s a 40Gb/s. Executamos 300 testes e a média foi então calculada.

[1] Este mecanismo dinâmico de solicitação e liberação de banda é conhecido como elasticidade e foi inicialmente abordado por [Iovanna et al., 2003].

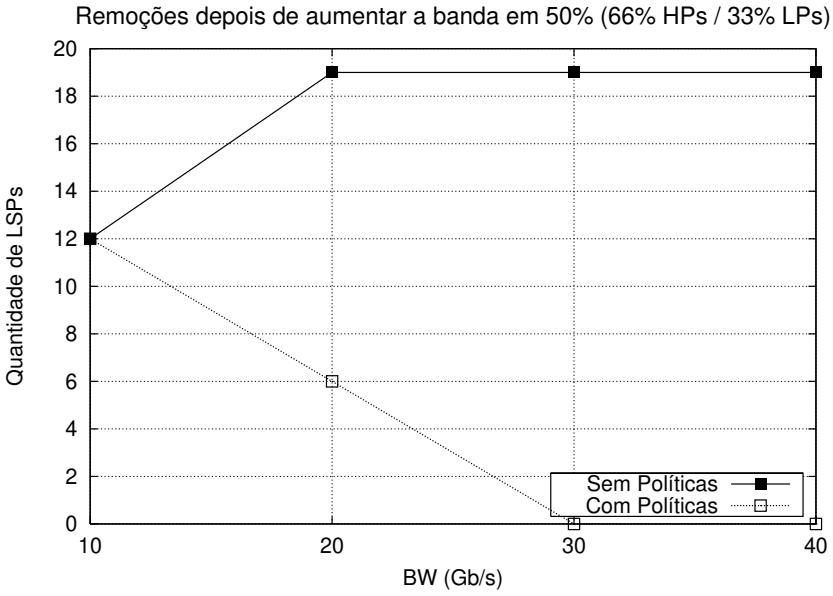


Fig. 5.4: Tráfego LP removido depois de aumentar a banda em 50%.

Observe que enquanto a quantidade de tráfego LP removido aumenta e mantém-se constante a partir de 20Gb/s sem o uso de políticas, a quantidade de tráfego LP removido com o uso de políticas diminui na medida em que a capacidade da rede aumenta. Isto ocorre porque durante a admissão do tráfego, a acomodação com o uso das políticas foi feita de forma a separar o tráfego HP do tráfego LP. Outra observação reflete o fato de que quanto mais LSPs são admitidos, mais LSPs precisam ser removidos durante o aumento da banda dos HPs. Se eles são agregados seguindo as políticas, tais remoções são minimizadas.

Os dados apresentados acima representam a aplicação das políticas simples. A arquitetura para esta fase envolveu somente os módulos AC e PM. Entretanto, sabe-se que um dos grandes problemas a ser considerado em redes ópticas diz respeito às falhas que ocorrem na rede. Como cada caminho de luz possui uma grande capacidade de banda, atualmente variando de 2.5Gb/s a 10Gb/s, uma falha num caminho de luz ou numa fibra inteira pode causar a interrupção de dezenas de Terabits de dados.

Muito embora as redes ópticas possuam mecanismos de proteção bastante conhecidos tais como 1+1, 1:1 e 1:N, constatou-se que o uso de políticas para a agregação do tráfego nos caminhos de luz, levando-se em conta a capacidade de proteção de cada caminho de luz, poderia diminuir o impacto de uma falha na rede. Como conseqüência, estendemos as políticas e elaboramos novas políticas que levassem em conta, não somente o tipo do tráfego (HP ou LP), mas também o tipo de proteção dos caminhos de luz. O conjunto de políticas resultante tornou-se obviamente mais complexo uma vez que as possibilidades de análise aumentaram e a heurística para a aplicação das políticas precisou ser mais elaborada.

Após um período de análises e com o intuito de melhor explorar a classificação dos mais variados sub-casos das políticas, convergiu-se para três grupos de políticas com diferentes complexidades. Tais grupos estão explicados em detalhes na Seção D.1 do Apêndice D. Resumidamente, o grupo G1 é o mais simples e considera o processo normal de agregação de tráfego como se não tivesse políticas levando em conta apenas o tipo de proteção desejado. O grupo G2 considera também a classe de serviço da requisição. O grupo G3 é o mais completo e considera a agregação de tráfego em caminhos de luz cuja proteção seja maior do que a solicitada na requisição além de permitir a quebra de grupos 1:N. Para a aplicação das políticas dos três grupos, a arquitetura envolveu o módulo de falhas (FM). A arquitetura é apresentada na Figura 5.5.

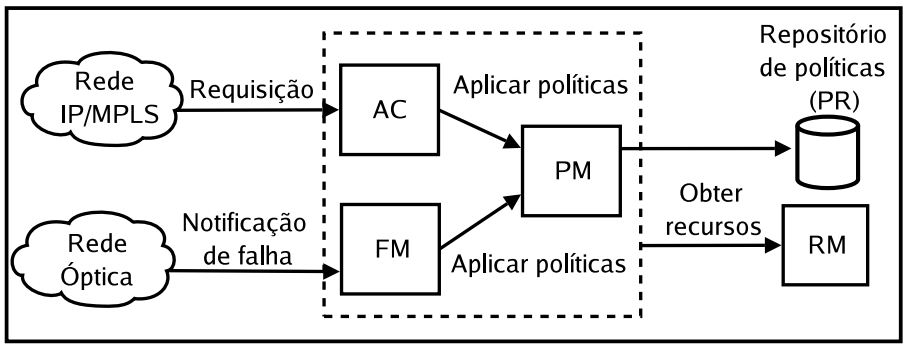


Fig. 5.5: Módulos envolvidos para a aplicação de políticas para falhas.

O diagrama de classes do PM encontra-se na Figura A.2 do Apêndice A. Abaixo apresentamos um gráfico (Figura 5.7) que demonstra as vantagens do uso de políticas para um cenário de testes. A topologia física da rede NSFNet foi utilizada nas simulações (Figura 5.6). Os *lightpaths* são criados do nó 0 para o nó 12, seguindo diferentes rotas. Cada enlace físico tem duas fibras unidirecionais (uma para cada direção) e cada fibra tem 10 lambdas (*wavelengths*) de 1 Gb/s. Com a configuração desta rede física criou-se 36 *lightpaths* (36 Gb/s) entre o nó de ingresso e o nó de egresso, distribuídos da seguinte forma: quatro desprotegidos, vinte e quatro 1:N, quatro 1:1 e quatro 1+1. Para a proteção 1:N foi definido que existe um *lightpath* de *backup* para três primários, configurando uma proteção 1:3. Isto resulta em seis grupos de 1:3 ($6 * (1+3) = 24$ *lightpaths*). No caso de 1:1 e 1+1, para cada *lightpath* primário existe um *lightpath* de *backup*. Resumindo, foram criados 36 *lightpaths*, são eles: 6 grupos 1:N, 2 grupos 1:1, 2 grupos 1+1 e 4 grupos desprotegidos.

Para validar as políticas, foram injetadas oito diferentes cargas de tráfego na rede, de 80% (0.8) a 200% (2.0) da largura de banda da rede (36 Gb/s). Com estas cargas foi possível avaliar o comportamento das políticas em cenários onde a quantidade de tráfego gerada é menor do que a capacidade da rede e, no outro extremo, quando a rede encontra-se em sobrecarga. A porcentagem de fluxo de tráfego gerada para cada requisição foi de: 35% para desprotegido, 15% para 1:N, 20% para 1+1 e 30% para 1:1. Estes fluxos de tráfego foram gerados levando-se em consideração a

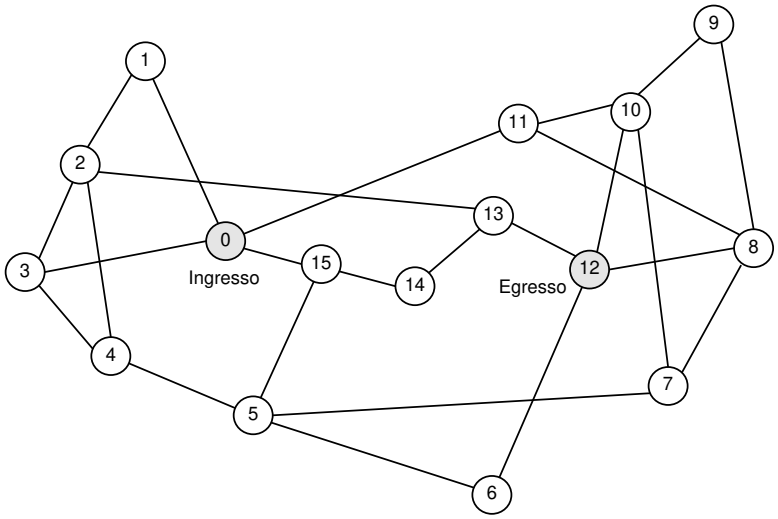


Fig. 5.6: Topologia da rede NSFNet.

porcentagem da carga da rede. Por exemplo, a quantidade de requisições geradas para a carga de 120% (1.2) considerando o tipo de proteção 1:1 é de: 36 Gb (capacidade da rede) $*$ 1.2 (carga a ser gerada) $*$ 0.3 (porcentagem de 1:1) = 13 Gb/s. A largura de banda mínima de uma requisição é 50 Mb/s e a máxima é 400 Mb/s. Estatisticamente, a largura de banda média para cada requisição é 225 Mb/s. No total, 50% do tráfego gerado é HP e 50% é LP. As simulações foram executadas 20 vezes para que o resultado seja obtido através da média entre as iterações. Uma falha de fibra simples é gerada aleatoriamente para cada iteração.

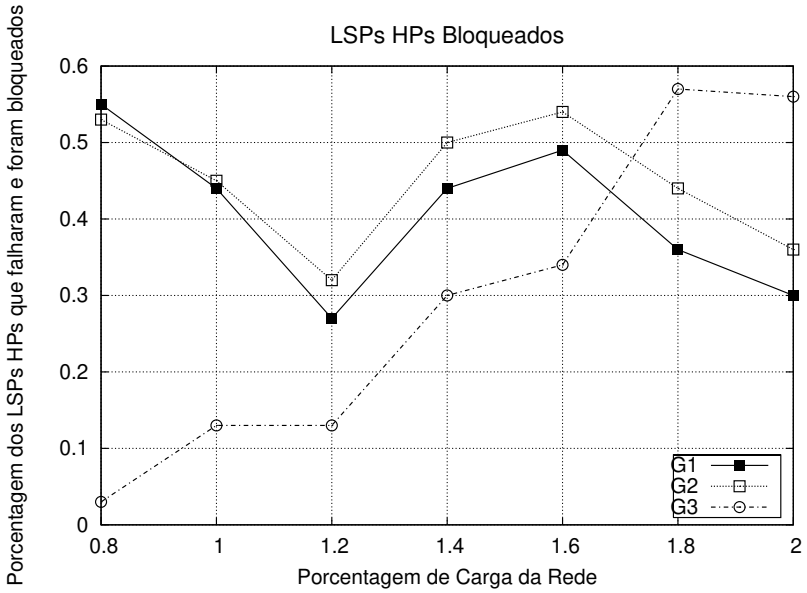


Fig. 5.7: Porcentagem de tráfego HP bloqueado após a falha.

A Figura 5.7 mostra a quantidade de fluxos HP afetados pela falha e que foram bloqueados após a tentativa de readmissão. O objetivo principal da Figura 5.7 é comparar o grupo G2 com o grupo G3. O grupo G1, por possuir políticas mais simples, admitiu menos tráfego e, portanto, bloqueou menos requisições no cenário com carga acima de 1.6. Embora o grupo G3 tenha apresentado a maior quantidade de tráfego HP afetado pela falha, o grupo G3 obteve o menor número de bloqueios até a carga de 1.6. O grupo G2 se mostrou mais eficiente do que o G3 somente em cenários onde a rede estava sobrecarregada, carga acima de 1.6, bloqueando uma menor quantidade de fluxos HP. De uma forma geral, o grupo G3 se mostrou interessante para reduzir a quantidade de tráfego bloqueado após a ocorrência de uma falha, considerando que a carga da rede esteja abaixo de 1.6. Para casos de sobrecarga, com uma carga acima de 1.6, o grupo G2 se mostrou mais eficiente, bloqueando uma menor quantidade de tráfego.

Esta subseção apresentou de forma resumida as políticas e o seu uso para a admissão de tráfego IP/MPLS na rede óptica. Primeiramente, apresentamos políticas simples que levam em consideração apenas a classe do tráfego (HP e LP). Após isto, as políticas foram estendidas e outras foram definidas para considerar possíveis falhas na rede. É importante destacar que as políticas aqui apresentadas refletem o estado atual dos nossos estudos levando-se em conta os mais variados cenários de testes efetuados. Obviamente, as políticas possuem inúmeros sub-casos e variações que poderiam ser testadas fazendo com que elas passem por alterações de forma a obter melhores resultados. A evolução das políticas com o objetivo de maximizar o uso dos recursos ópticos é algo constante e que pode evoluir na medida em que diferentes cenários são considerados.

A próxima subseção é dedicada ao provisionamento de VPNs ópticas dentro de um domínio. Tal subseção encerra o provisionamento e gerência de serviços dentro de um domínio óptico.

5.2.2 Provisionamento do Serviço L1VPN

Nesta subseção apresentamos resumidamente o serviço de VPNs ópticas e alguns resultados da aplicação de políticas para o provisionamento de tal serviço. Mais detalhes sobre o serviço, cenários e resultados de aplicações de políticas podem ser obtidos em [Malheiros et al., 2006b] e em [Malheiros et al., 2006a].

Antes de apresentarmos a arquitetura proposta para o provisionamento de VPNs, apresentamos um modelo de referência para o serviço de L1VPN (VPNs de camada 1) conforme discutido em [Takeda et al., 2004]. A Figura 5.8 ilustra este modelo.

Um *Customer Edge device* (CE) é um nó da rede do cliente que está conectado à rede do provedor. Um *Provider Edge device* (PE) é um nó da rede do provedor ao qual pelo menos um CE está conectado. O PE provê serviços L1VPN para o CE. Um *Provider device* (P) é um nó do núcleo da rede do provedor que não está conectado a nós das redes clientes, mas somente a nós da rede do

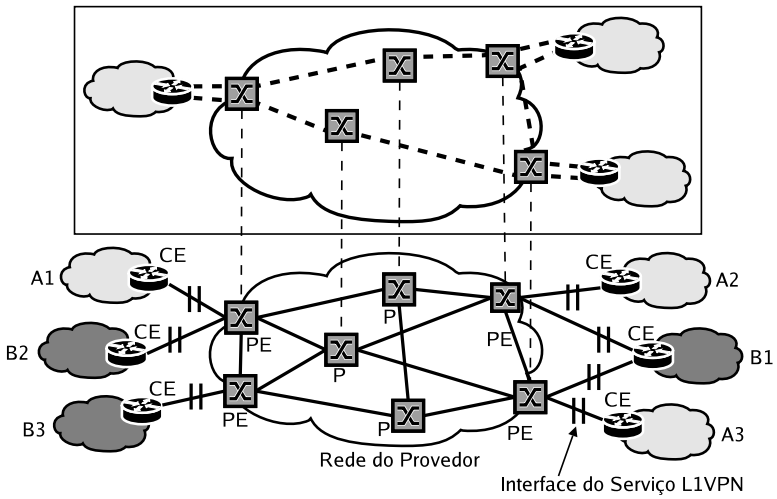


Fig. 5.8: Modelo de referência para o serviço L1VPN.

provedor. Neste exemplo, a rede de transporte do provedor é compartilhada por dois clientes, A e B. A parte superior da figura mostra uma possível topologia para "interconectar" as redes do cliente A. As conexões da VPN estão representadas pelas linhas tracejadas entre os CEs.

O provisionamento de L1VPN também segue a abordagem baseada em políticas. Para este trabalho, três classes de políticas para gerência de serviços L1VPN foram definidas [Malheiros et al., 2006b]: políticas de configuração, políticas de admissão e políticas de roteamento. As **Políticas de Configuração** são utilizadas para definir parâmetros de configuração que controlam a operação de serviços L1VPN. As **Políticas de Controle de Admissão** permitem definir regras adicionais para controlar o processo de admissão de conexões, que considera também informações sobre membros da VPN e disponibilidade de recursos. Finalmente, as **Políticas de Roteamento** são utilizadas para controlar o cálculo ou a seleção de rotas. Elas permitem definir métricas e restrições para o cálculo de rotas, assim como otimizar a seleção de uma rota para uma conexão quando existem várias rotas disponíveis. Na Figura D.1 do Apêndice D, apresentamos um exemplo de uma política descrita em XML.

A arquitetura proposta considera o modelo baseado em gerência para serviços L1VPN pelo qual uma requisição para estabelecimento de uma VPN é feita através de uma interface de gerência oferecida pelo provedor do serviço. Também consideramos que a rede do provedor implementa um plano de controle GMPLS. O cenário considerado está ilustrado na Figura 5.9. Para criar uma conexão VPN entre dois CEs, o cliente envia uma requisição de conexão a um sistema de gerência de serviços L1VPN. Este sistema se encarrega de requisitar ao PE de ingresso uma conexão entre os PEs correspondentes. O Sistema de Gerência de L1VPN possui os módulos da camada de gerência apresentados anteriormente. Especificamente para o estabelecimento de L1VPNs, dois módulos se

destacam: o *Policy Manager* (PM) para aplicação das políticas relacionadas com o serviço de L1VPN e o *Membership Manager* (MM) para verificação de correlação de portas.

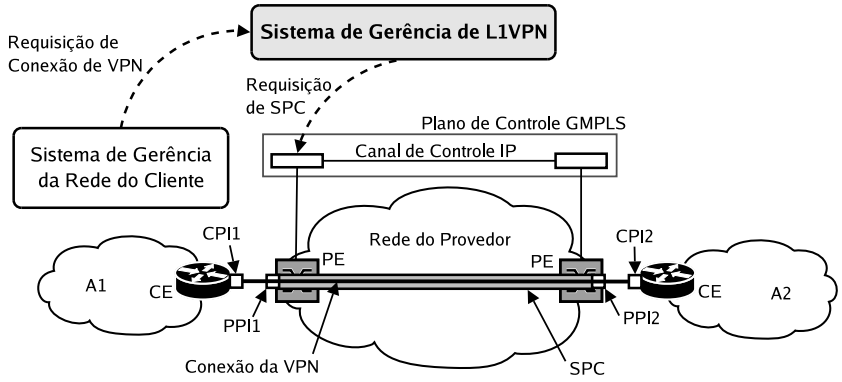


Fig. 5.9: Cenário de aplicação.

O estabelecimento da conexão no núcleo da rede do provedor é efetuado pelo mecanismo de sinalização do plano de controle, por exemplo, por meio do protocolo RSVP. Uma VPN consiste no estabelecimento de uma ou mais SPCs entre diferentes PEs. O estabelecimento de SPCs segue os passos apresentados no início deste capítulo cujo diagrama de seqüência foi ilustrado na Figura 5.3.

Um membro de uma VPN é designado por um par de portas (lógicas) que representam os extremos de uma conexão estática entre um CE e um PE. Assim, esse par é formado por dois identificadores: o identificador de porta do cliente (CPI – *Customer Port Identifier*) e o identificador de porta do provedor (PPI – *Provider Port Identifier*). Portanto, um par CPI-PPI identifica um membro da VPN. Normalmente, o formato de cada identificador consiste da união do endereço do nó com o endereço da porta. Ainda considerando a Figura 5.9, os CEs das redes A1 e A2 de um cliente A podem ser identificados como dois membros CPI1-PPI1 e CPI2-PPI2 da L1VPN A. Para estabelecer uma conexão VPN entre suas redes A1 e A2, o cliente requisita ao sistema de gerência uma conexão entre aqueles dois membros. O sistema de gerência então requisita ao plano de controle uma SPC entre os PEs correspondentes, identificados pelas portas PPI1 e PPI2. Dessa forma, o tráfego enviado pela porta CPI1 é transmitido até a porta CPI2 pela conexão estabelecida na rede do provedor.

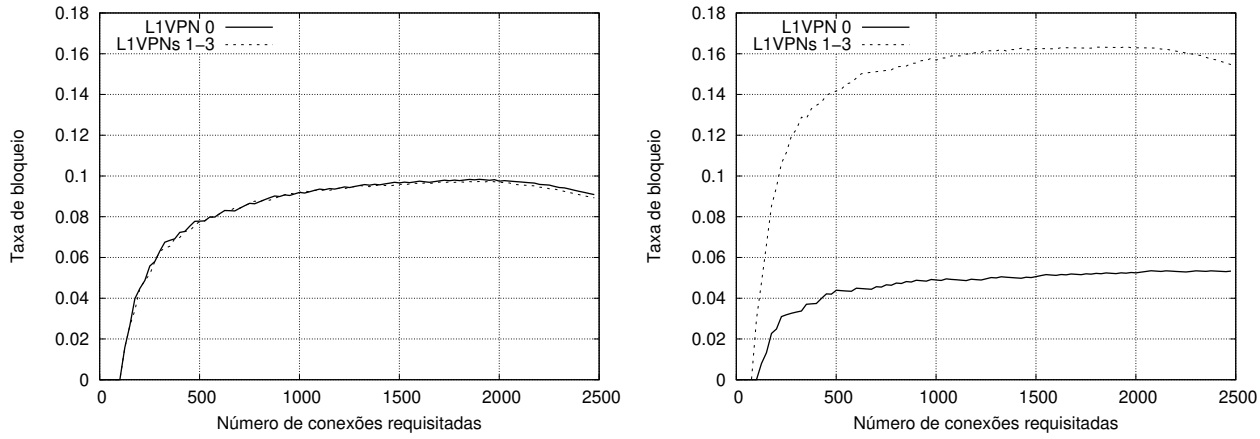
Para finalizar esta subseção, apresentamos um gráfico que representa a aplicação de políticas de configuração para o serviço de L1VPN [Malheiros et al., 2006b]. Cada cliente solicita um total de 2500 conexões. Os pares origem e destino para as conexões são selecionados do conjunto de CEs membros de uma VPN, segundo uma distribuição aleatória uniforme. A taxa de requisição de conexões segue uma distribuição de Poisson, e é dada por número de requisições por segundo. O tempo de duração de uma conexão (até que os recursos sejam liberados) e o intervalo entre requisições seguem distribuições exponenciais. Os resultados são a média de 100 repetições das simulações. Consideramos 4 serviços (L1VPN 0-3) e 32 comprimentos de onda em cada enlace da

rede. Avaliamos o efeito das seguintes políticas sobre a taxa de bloqueio de conexões:

1. Se a VPN é de alta prioridade, então: a alocação de recursos segue o modelo dedicado e um subconjunto dos recursos do provedor é dedicado para a VPN. No modelo dedicado, recursos da rede do provedor são reservados exclusivamente para uma L1VPN. Esses recursos não podem ser alocados para nenhuma outra L1VPN mesmo que eles não estejam sendo usados;

2. Se a VPN é de baixa prioridade, então: a alocação de recursos segue o modelo compartilhado e a VPN disputa com outras VPNs a alocação de recursos compartilhados. No modelo compartilhado, os recursos são compartilhados no tempo e podem ser alocados por qualquer L1VPN [Malheiros et al., 2006b] na medida em que eles são liberados.

Primeiramente, a simulação foi realizada sem considerar as políticas. A Figura 5.10(a) apresenta a taxa de bloqueio da L1VPN 0 e a média da taxa de bloqueio das outras L1VPNs. A Figura 5.10(b) mostra a alteração nesses valores quando as políticas são aplicadas. Neste cenário, a L1VPN 0 é considerada de alta prioridade, sendo reservados para ela 10 comprimentos de onda em cada enlace. Os resultados demonstram uma queda na taxa de bloqueio da L1VPN 0 e um aumento na média da taxa de bloqueio das outras L1VPNs quando as políticas são utilizadas.



(a) Sem políticas (b) Com políticas

Fig. 5.10: Taxa de bloqueio de conexões.

Este cenário representa um estudo de caso para demonstrar como as políticas podem ser aplicadas e seus diferentes efeitos. Foi demonstrado que políticas de configuração podem ser definidas para oferecer serviços com prioridades diferenciadas. Além disso, constatou-se que políticas que melhoram o desempenho de um serviço podem afetar o desempenho de outros serviços.

Finalizações do capítulo

A subseção acima finaliza a apresentação da arquitetura para o provisionamento e gerência de serviços dentro de um domínio. No próximo capítulo apresentamos como a arquitetura evoluiu para instanciar o modelo proposto para prover serviços entre domínios, foco principal desta tese.

Capítulo 6

Instanciação do Modelo para Provisionamento e Gerência de Serviços entre Domínios

No capítulo anterior apresentamos a instanciação do modelo dando origem à uma arquitetura para provisionamento e gerência de serviços dentro de um domínio. Naquele capítulo, focamos especificamente na camada de gerência de redes do modelo proposto. Neste capítulo nos concentraremos na camada de serviços e de negócios do modelo. Apresentamos a instanciação do modelo e a incorporação de funcionalidades à arquitetura a fim de suportar o provisionamento e gerência de serviços entre domínios. Através do modelo TMN, definimos a arquitetura na forma de camadas e uma divisão das atividades entre as camadas foi feita. Basicamente, definimos que a camada de gerência de redes deveria ser responsável pelas tarefas internas ao domínio e realizar toda a lógica para suportar o provisionamento de serviços, tanto dentro de um domínio como entre domínios. A camada de serviços deve oferecer apenas uma interface interoperável para as entidades externas invocarem as funcionalidades suportadas pelo domínio.

Nesta fase, identificamos dois tipos de serviços: serviços de relacionamento com o cliente e serviços de suporte à infra-estrutura. A interação entre estes dois blocos de serviços foi feita por uma camada que chamamos de camada de integração. Esta camada de integração nada mais é do que a camada de gerência de redes estendida para suportar a interação entre os serviços dos dois blocos. A camada de serviços permitiu, entre outras coisas, que as interações entre domínios para estabelecimento de serviços fossem realizadas sem precisar estender protocolos de roteamento e sinalização.

Este capítulo está organizado da seguinte forma. Primeiramente, apresentamos os pré-requisitos identificados para o desenvolvimento da arquitetura. A definição dos pré-requisitos serve como base

para a elaboração dos serviços que farão parte da arquitetura proposta. A seguir, na Seção 6.2, apresentamos os serviços definidos e suas atividades. Na Seção 6.3, a arquitetura é apresentada e uma descrição detalhada de cada serviço é realizada. A Seção 6.4 é dedicada à modelagem BPMN. Os diagramas de negócios em BPMN descrevem os processos e as atividades e como os módulos interagem para suportar os serviços propostos neste trabalho. Por fim, na Seção 6.5, faremos uma discussão final sobre o desenvolvimento da arquitetura.

6.1 Identificação dos Pré-Requisitos

Umas das principais contribuições desta tese é realizar o mapeamento dos processos de negócios que representam os serviços oferecidos para módulos que, de forma conjunta, possam oferecer funcionalidades para suportar conexões e VPNs entre domínios ópticos administrativamente diferentes. Inicialmente, definimos alguns pré-requisitos que devem ser atendidos ao estendermos a arquitetura para suportar o provisionamento de serviços entre domínios. Após, apresentamos a arquitetura e os serviços que foram definidos para implementar estes pré-requisitos, juntamente com a modelagem BPMN dos processos que utilizam os serviços definidos.

Diante da necessidade atual dos provedores, um dos principais serviços a ser oferecido é o serviço de conexões entre domínios [Howarth et al., 2005]. É cada vez mais importante a interação entre domínios de forma a suportar novas aplicações tais como multimídia, distribuição de vídeos e conteúdo, VoIP, etc. Além disso, há uma crescente demanda por VPNs cujas portas estão distribuídas em um cenário envolvendo vários domínios administrativos diferentes. O compartilhamento da infra-estrutura de um domínio entre vários clientes tem feito com que provedores explorem cada vez mais os serviços de VPN a fim de aumentarem suas receitas. Diante disso, neste trabalho focamos em dois serviços. O primeiro consiste em prover um serviço de conexões entre domínios. Tais conexões têm como objetivo estabelecer caminhos de luz que interconectam domínios administrativos diferentes. O segundo tipo de serviço consiste em prover um serviço de VPN entre domínios. Enquanto a maioria das propostas atuais considera o provisionamento de VPNs em um mesmo domínio, sabemos que o serviço de VPNs entre domínios é algo importante e necessário, porém encontra-se ainda em fase de discussão nos órgãos internacionais de padronização [Li and Gao, 2006].

Após apresentarmos os dois serviços entre domínios que estamos interessados em prover, é possível definirmos os pré-requisitos gerais que norteiam esta fase. Os requisitos listados abaixo são uma compilação do que se pode encontrar atualmente na literatura [Zhang et al., 2004, Takeda et al., 2004, Yang et al., 2006]. Tais requisitos refletem as principais necessidades em relação ao estabelecimento de conexões e VPNs em cenários envolvendo vários domínios administrativos.

São eles:

1. Oferecer um serviço de conexões fim-a-fim entre domínios;
2. Os provedores devem estabelecer os serviços de forma transparente para os clientes;
3. Permitir que os domínios possam selecionar as rotas fim-a-fim entre domínios levando-se em conta características de QoS;
4. Permitir que os domínios divulguem seus recursos ópticos sem revelar aspectos internos ao domínio;
5. Permitir que os domínios possam negociar entre eles a fim de que os recursos sejam reservados em cada domínio e que políticas locais sejam aplicadas;
6. Facilitar esta negociação entre domínios fazendo com que as interações entre eles sejam feitas de forma flexível;
7. Oferecer um serviço de VPN entre domínios;
8. Permitir que os domínios possam fazer ofertas para outros domínios a fim de atrair clientes;
9. Definir uma camada de serviços que não interfira nas decisões locais de cada domínio;
10. Criar e manter um modelo simples através desta camada de serviços que suporte os dois serviços entre domínios desejados.

Os requisitos 1 e 7 correspondem aos dois serviços entre domínios considerados nesta tese. Os requisitos 3, 4 e 8 estão relacionados à forma como os domínios podem, ao mesmo tempo, divulgar informações sobre recursos disponíveis sem divulgar os detalhes internos de cada domínio. O restante dos requisitos estão relacionados com a independência e transparência entre o cliente e o domínio e entre domínios para o provisionamento dos serviços.

6.2 Identificação e Definição dos Serviços

Para atender a tais requisitos, criamos seis serviços que completam a arquitetura e instanciam o modelo da Figura 3.2. Dividimos os serviços em dois blocos. O primeiro corresponde aos serviços que chamamos de serviços de relacionamento ou serviços de negócios. O segundo bloco corresponde aos serviços de suporte à infra-estrutura ou serviços utilitários. Os serviços de relacionamento são aqueles que permitem uma relação de negócios com os clientes. Eles oferecem interfaces para os

clientes a fim de que serviços sejam solicitados ao domínio. Os serviços de suporte à infra-estrutura são utilizados para oferecer o suporte básico para os serviços de negócios. Não oferecem interfaces aos clientes. São invocados por outros serviços (tipicamente serviços de negócios) para a realização de tarefas específicas.

Em cada domínio, os serviços de relacionamento interagem com os serviços de suporte à infra-estrutura através de uma camada de integração. Esta camada de integração é formada por módulos internos (locais ao domínio) e realizam tarefas locais tais como controle de admissão, aplicação de políticas, etc. A camada de integração é efetivamente a camada de gerência de redes que foi estendida para suportar as atividades de provisionamento de serviços entre domínios e integrar os serviços de relacionamento com os serviços utilitários. A camada de gerência de redes foi apresentada e discutida em detalhes nos Capítulos 3 e 5. Os módulos da camada de gerência de redes foram estendidos para realizar não somente as atividades para provisionamento de serviços dentro de um domínio mas também para suportar o provisionamento de serviços entre domínios. A Figura 6.1 ilustra o cenário mencionado.

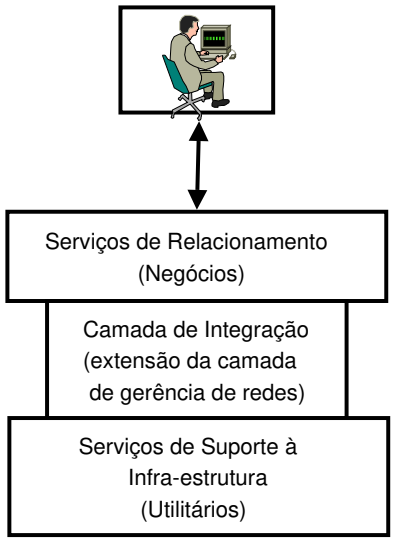


Fig. 6.1: Serviços de relacionamento, serviços de suporte à infra-estrutura e camada de integração.

Um refinamento maior dos pré-requisitos pode ser realizado através das definições apresentadas em [Erl, 2004b]. As definições permitem uma separação das atividades de negócios em alto nível, porém oferecendo um grau de detalhamento suficiente para a compreensão de cada processo. A referência mencionada apresenta os seguintes conceitos:

- Serviço Primitivo de Negócios: representa um serviço auto-contido e auto-suficiente. Não precisa de outros serviços para realizar suas tarefas e normalmente é usado para participar

como parte de um conjunto de serviços compostos. Em relação à SOA, possui granularidade de um serviço;

- Atividade Primitiva de Negócios: corresponde à peça mais básica da arquitetura SOA. O agrupamento de múltiplas atividades primitivas forma logicamente um serviço primitivo de negócios. Em termos técnicos, uma atividade primitiva de negócios é vista como uma operação/método.

Além dos conceitos acima, para esta tese um conceito novo foi definido e denominamos de Serviço Estendido de Negócios. Um Serviço Estendido de Negócios consiste da união de dois ou mais Serviços Primitivos de Negócios. Em relação à arquitetura SOA, este conceito é conhecido como composição de serviços.

Aplicando as definições citadas acima neste trabalho, temos:

- Seis Serviços Primitivos de Negócios e suas Atividades Primitivas de Negócios (apresentamos apenas as principais atividades primitivas de negócios representadas por asteriscos dentro de cada Serviço Primitivo):
  - Divulgar topologia virtual
    - obter topologia virtual local;
    - divulgar topologia virtual local para outros domínios.
  - Obter topologia virtual
    - obter topologia virtual de outros domínios;
    - armazenar topologias virtuais no domínio local.
  - Divulgar informações sobre correlação de portas de VPNs
    - obter tabelas de portas das VPNs no domínio local;
    - divulgar as tabelas para os outros domínios.
  - Negociar com outros domínios
    - reservar recursos em outros domínios;
    - verificar se todos os domínios realizaram a reserva;
    - confirmar reserva;
    - desfazer reserva.
  - Calcular rotas entre domínios
    - unir as topologias virtuais de todos os domínios;

        * aplicar algoritmos para cálculo de rotas.

- Monitorar e ativar VPN
  - obter informações sobre VPN;
  - ativar VPN;
  - desativar VPN.

- Dois Serviços Estendidos de Negócios e suas Atividades Primitivas de Negócios (apresentamos apenas as principais atividades primitivas de negócios representadas por asteriscos dentro de cada Serviço Estendido):
  - Estabelecer conexão entre domínios
    * receber invocação de clientes (1);
    * verificar contratos (2);
    * aplicar políticas (3);
    * reservar recursos;
    * iniciar negociação;
    * finalizar estabelecimento de conexões.
  - Reservar recursos para VPN
    * executar o três primeiros passos acima;
    * realizar correlação de portas;
    * Para cada par de portas, estabelecer uma conexão;
    * finalizar reserva de recursos.

A definição dos seis serviços elaborados para esta tese ocorreu da seguinte forma. Os três primeiros Serviços Primitivos de Negócios foram agrupados em um único serviço uma vez que suas atividades são similares. Temos então que o serviço de divulgação e obtenção de topologias virtuais, assim como o serviço de divulgação de informações sobre correlação de portas da VPNs, fazem parte de um único serviço que denominamos *Advertising Service*(AS). O serviço primitivo de negociação foi mapeado para o serviço que chamamos de *End-to-End Negotiation Service*(E2ENS). O serviço primitivo de cálculo de rotas é representado pelo serviço *Path Computation Element*(PCE). O serviço primitivo que monitora e ativa VPNs é representado pelo *Optical Virtual Private Network Service*(O-VPNS). Um serviço foi definido para cada serviço estendido de negócio. O serviço estendido para estabelecer conexões entre domínios é representado pelo *End-to-End Connection*

Service(E2ECS). O serviço estendido que realiza a reserva de recursos para VPNs é mapeado para o *Trading Service*(TS). Todos os serviços serão detalhados abaixo.

Após este mapeamento, podemos definir mais facilmente quais são os serviços de suporte à infra-estrutura (utilitários) e quais são os serviços para relacionamento (negócios). Os serviços AS, E2ENS e PCE são considerados serviços de suporte à infra-estrutura. Eles são usados pelos serviços de negócios e não oferecem nenhuma interface para interação com clientes. O serviços E2ECS, TS e O-VPNS são invocados pelos clientes para solicitação de serviços ao domínio e invocam os serviços utilitários para realização das tarefas. A Figura 6.2 mostra os dois blocos e seus serviços.

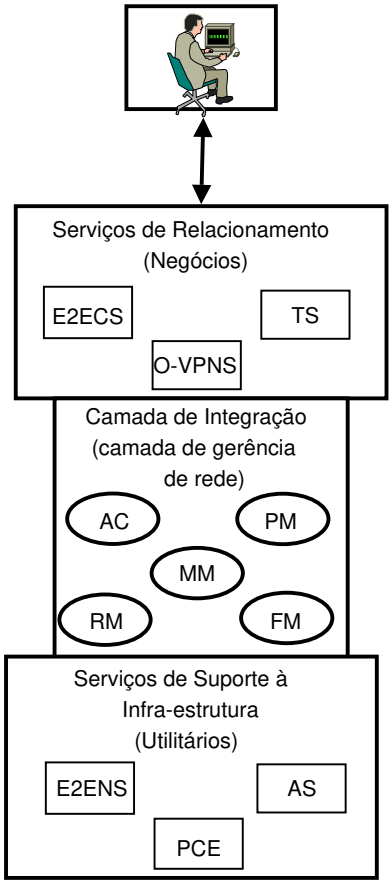


Fig. 6.2: Serviços utilitários e serviços de relacionamento.

A camada de integração entre os dois blocos de serviços realiza tarefas locais em nome dos serviços. Cada serviço invoca os módulos da camada de integração dependendo da atividade que deverá ser realizada. Tanto a camada de serviços de relacionamento como a camada de serviços utilitários agem apenas como uma interface para as entidades externas (clientes e outros domínios). As requisições são sempre encaminhadas para a camada de integração que possui a lógica para o tratamento de cada requisição. A camada de integração é formada pelos módulos definidos

anteriormente para a camada de gerência de redes (AC, PM, RM, FM e MM).

6.3 Apresentação da Arquitetura

A arquitetura com a camada de serviços é apresentada na Figura 6.3. Os serviços definidos realizam as tarefas relacionadas ao provisionamento de conexões e VPNs entre domínios atendendo aos pré-requisitos citados acima. Outros serviços podem ser definidos através da utilização dos serviços atuais.

A Figura 6.3 mostra a arquitetura completa com os módulos anteriores (agora referenciados como módulos internos fazendo parte da camada de integração) e com os módulos novos que representam a camada de serviços. A arquitetura apresentada na Figura 6.3 é uma evolução da arquitetura para um domínio discutida no capítulo anterior. Para esta fase, não estamos interessados em como o plano de controle estabelece as conexões ópticas dentro do domínio. As topologias virtuais que representam as conexões estabelecidas abstraem os detalhes internos de cada domínio.

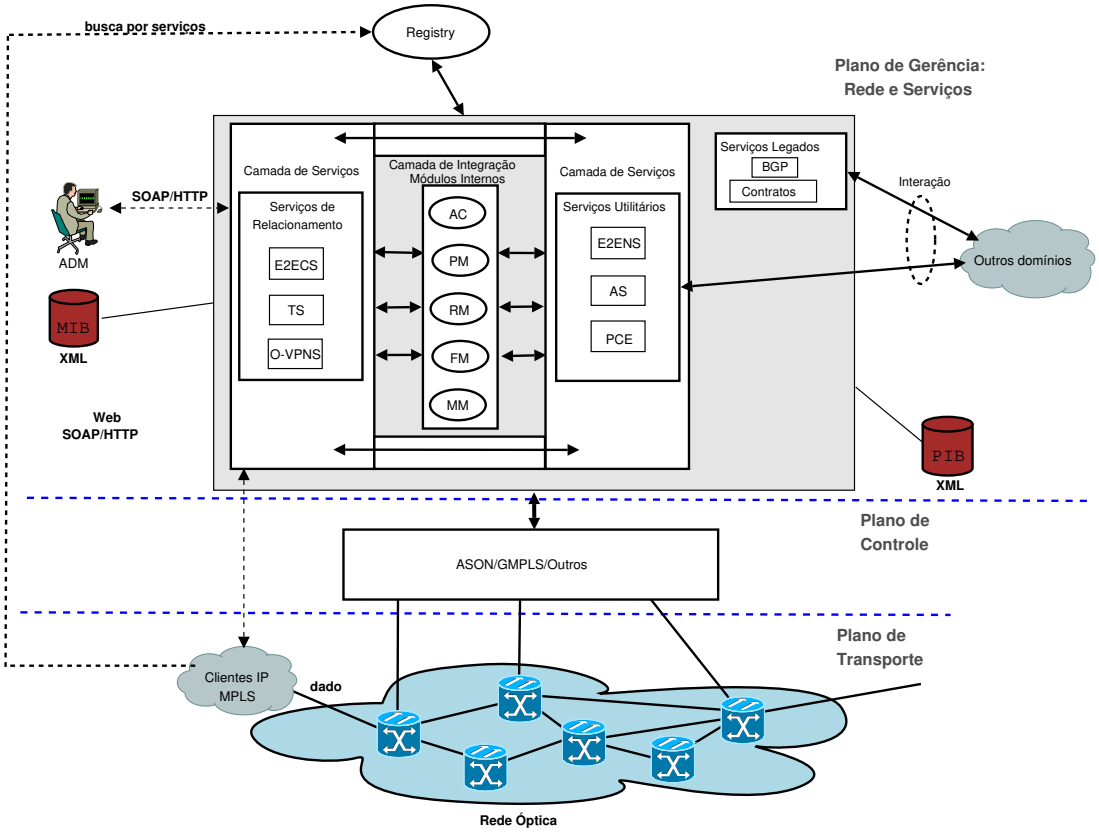


Fig. 6.3: Arquitetura para o provisionamento de serviços entre domínios.

Os módulos da camada de integração são responsáveis pelo controle, gerência, admissão e

configuração em cada domínio. A camada de serviços surge agora com novos módulos além do E2ECS definido anteriormente. Cada domínio óptico passa a ser visto como uma "caixa preta" representada através de um conjunto de serviços definidos através de suas interfaces. A Figura 6.4 mostra este conceito.

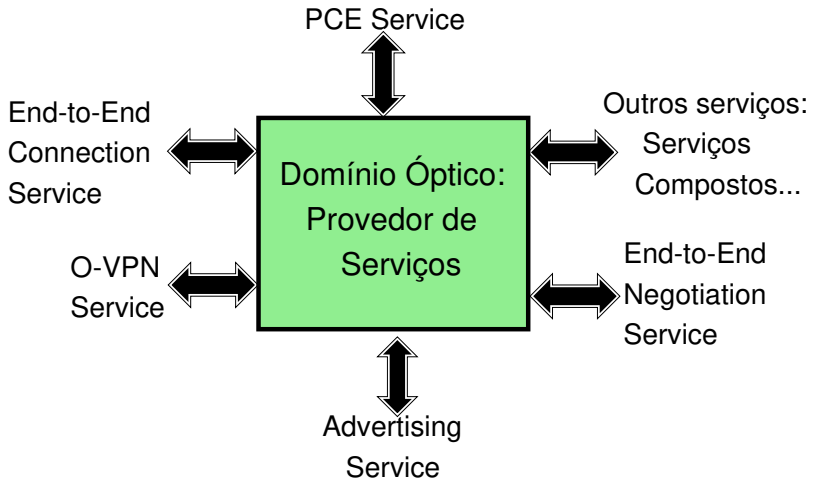


Fig. 6.4: Domínio óptico visto como um conjunto de serviços.

A seguir, detalhamos as funcionalidades de cada módulo da camada de serviços e como os pré-requisitos definidos acima são atendidos por tais módulos.

6.3.1 *Advertising Service - AS*

Este serviço é responsável por implementar o conceito de topologia virtual (VT) apresentado no Capítulo 4. O AS é responsável por duas tarefas. A primeira tarefa se refere à divulgação e obtenção das VTs. Tanto o modelo *Push* como o modelo *Pull* são suportados pelo AS. As VTs são definidas internamente em cada domínio óptico e refletem o estado atual da rede e podem representar as regras de negócios do domínio. No modelo *Push*, o AS invoca outros ASes em outros domínios para divulgação das VTs. Os domínios podem "assinar" o serviço de divulgação de VTs a fim de receber as divulgações em intervalos de tempo pré-definidos. No modelo *Pull*, os ASes invocam outros ASes para obter as VTs dos outros domínios. A segunda tarefa do AS se refere à divulgação dos membros de cada VPN entre domínios. O AS divulga as tabelas que correlacionam quais portas pertencem a quais VPNs ópticas. Estas tabelas são então armazenadas no *Membership Manager* que

fará a verificação de portas durante o pedido de estabelecimento de uma VPN entre domínios. Os pré-requisitos 3, 4 e 8 são atendidos pelo AS;

6.3.2 *End-to-End Negotiation Service - E2ENS*

Este serviço é responsável pela negociação de conexões entre domínios. Para este trabalho foi adotado o modelo em estrela de duas fases para o protocolo de negociação. No modelo em estrela, o domínio que deseja estabelecer uma conexão entre domínios negocia com todos os domínios que fazem parte da rota em direção ao destino. A Figura 6.5 ilustra o caso em que todos os domínios conseguem reservar recursos. A Figura 6.6 ilustra o cenário onde um dos domínios não consegue reservar os recursos. Na primeira fase do protocolo, o domínio requisitante informa os parâmetros requeridos para o estabelecimento da conexão. Tipicamente, os parâmetros são um identificador do domínio requisitante, a banda desejada, o nível de proteção da conexão óptica e o tipo do tráfego (HP ou LP). Outros parâmetros também podem ser negociados. Cada parâmetro é analisado internamente em cada domínio seguindo as regras locais. Caso algum dos parâmetros não possa ser atendido, a requisição é rejeitada pelo domínio invocado. Caso contrário, uma reserva é feita em cada enlace virtual que será utilizado para atravessar o domínio. O E2ENS em cada domínio invoca os módulos internos (especificamente o *Resource Manager*) para realizar a reserva. Lembrando que a reserva consiste em selecionar um caminho de luz disponível nos enlaces virtuais que atenda a todos os parâmetros enviados. Se todos os domínios possuem os recursos desejados, a segunda fase confirma (*commit*) a reserva feita na primeira fase e um contrato de serviço é estabelecido. O E2ENS em cada domínio confirma a reserva invocando novamente os módulos internos (Figura 6.5). Na segunda fase, cada domínio é responsável por realizar a "crossconexão" nos OXCs de borda que conectam os domínios. Se algum domínio não possui os recursos desejados, a segunda fase do protocolo de negociação desfaz (*rollback*) a reserva efetuada nos domínios cujos recursos foram reservados (Figura 6.6).

Note que no cenário de falha (Figura 6.6), a operação para desfazer a reserva é realizada apenas nos domínios cuja reserva foi efetuada.

É sabido que o processo de provisionamento de serviços em um único domínio é bem mais simples do que quando avançamos as fronteiras para oferecer serviços que envolvem vários domínios administrativos diferentes. O principal problema consiste justamente na negociação entre estes domínios. Nesta tese, o objetivo consistiu em deixar o serviço de negociação simples. Porém, um aspecto importante a ser adicionado ao serviço é a possibilidade de descrição dos atributos que compõem a interface do serviço, ou seja, adicionar semântica a cada atributo. Com esta descrição, os domínios interessados em negociar podem obter mais informações sobre os parâmetros a serem enviados durante o processo de negociação. Em especial, no caso dos *Web Services*, uma

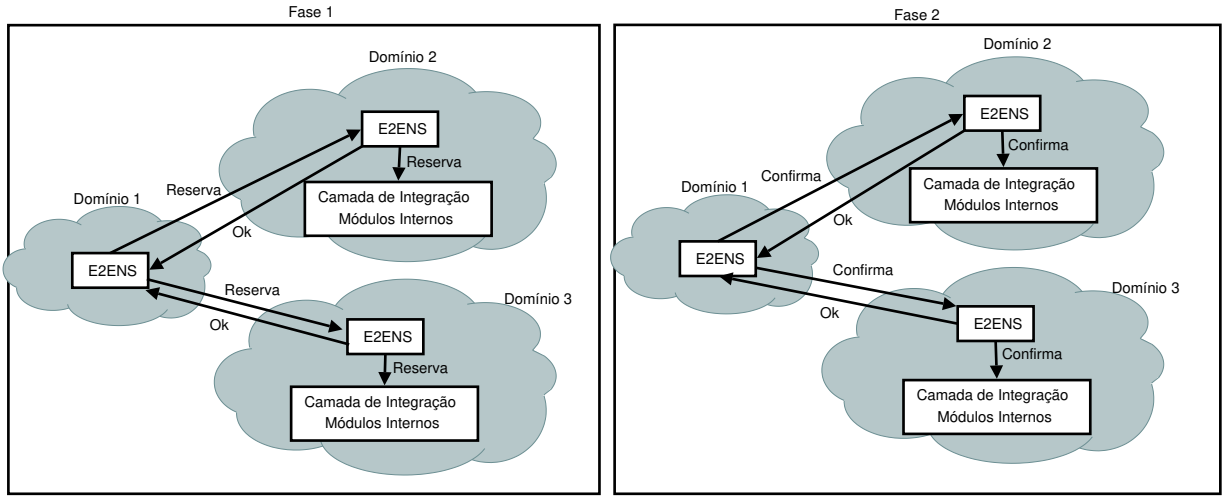


Fig. 6.5: Negociação entre domínios (caso de sucesso).

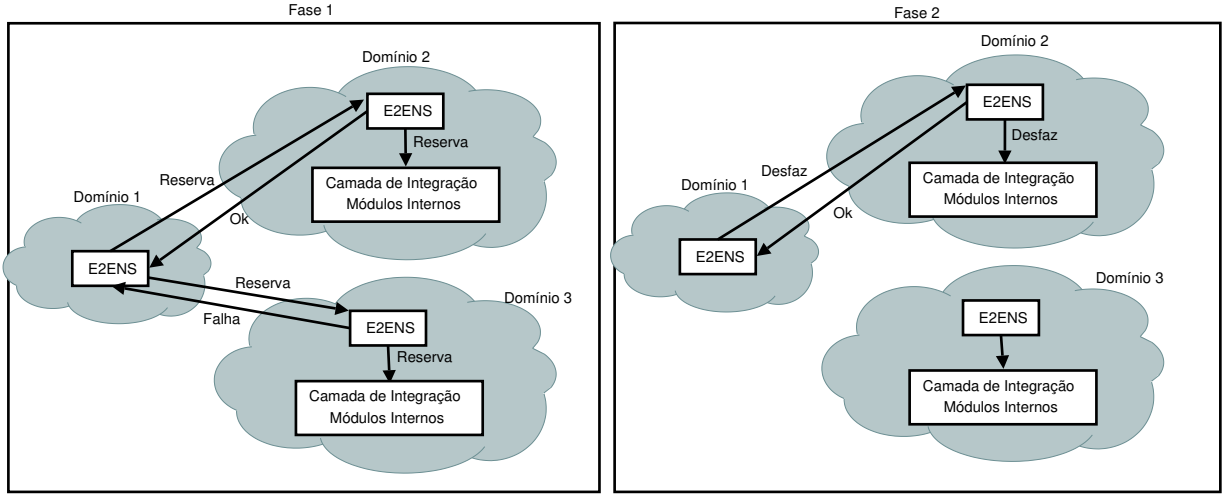


Fig. 6.6: Negociação entre domínios (caso de falha).

meta-linguagem conhecida como *OWL-S: Semantic Markup for Web Services* [OWLS, 2006] pode ser usada para descrever interfaces, métodos e atributos.

O serviço de negociação não interfere na forma como cada domínio seleciona os recursos locais. O E2ENS apenas troca os parâmetros a serem negociados para o estabelecimento de uma conexão fim-a-fim. Com isso, atendemos os pré-requisitos 5 e 6;

6.3.3 *End-to-End Connection Service - E2ECS*

Este serviço já estava presente na arquitetura definida para um domínio. Ele oferece acesso a todas as funcionalidades do sistema de gerência conforme explicado anteriormente. Porém, as

funcionalidades relacionadas ao serviço de VPN entre domínios passam a ser realizadas pelo serviço de *Trading* (ver abaixo) e pelo serviço de VPN (ver abaixo). O E2ECS atende os requisitos 1 e 2;

6.3.4 *Path Computation Element (PCE) Service*

O PCE é responsável pelo cálculo de rotas entre domínios. Após montar uma topologia virtual única com todas as VTs recebidas, o PCE aplica um algoritmo de roteamento sobre esta topologia. O PCE, conforme definido pelo IETF [Farrel et al., 2006, Ricciato et al., 2005], não é somente responsável pelo cálculo de rotas. Ele também realiza a obtenção das topologias, sejam elas virtuais ou não, de outros domínios. Nesta tese, estas duas funcionalidades foram divididas pois entendemos que o PCE é uma entidade que pode evoluir de forma separada do serviço de divulgação e obtenção de topologias. A obtenção e divulgação de VTs é responsabilidade do AS. O PCE pode ser um serviço oferecido por um provedor podendo representar uma área formada por vários domínios. Ele pode agir como um serviço de roteamento oferecido por alguma entidade (um terceiro) certificada e creditada pelo grupo de domínios. Assim, domínios que não possuem um serviço de roteamento podem contratar tal serviço deste terceiro (*third-party provider*).

6.3.5 *Trading Service - TS*

O TS permite que os recursos de uma VPN sejam reservados antes de sua ativação. Consideramos que os recursos ópticos podem ser reservados e depois ativados para uso. A reserva garante que os recursos não serão usados por outras VPNs e passam a ser exclusivos de quem os reservou. Os recursos só poderão ser usados por outras VPNs quando ocorrer a liberação dos recursos de quem os possui. A ativação dos recursos representa o instante real em que os recursos serão utilizados para trânsito de dados da VPN. É neste momento que o processo de cobrança (*billing*) é disparado pelo provedor. O TS reserva os recursos em cada domínio para cada conexão que pertence à VPN sendo estabelecida. Como uma VPN é formada por um conjunto de conexões (SPCs), o TS negocia os recursos individualmente para cada conexão. O estabelecimento das conexões é feito com base nas portas CPIs informadas pelo cliente. O *Membership Manager* é responsável pela correlação das portas a fim de analisar se os pares informados pelo cliente pertencem à VPN em fase de estabelecimento. Cada domínio pode também tarifar a reserva dos recursos. Além disso, em cada domínio, os recursos reservados podem ser usados para transferir tráfego de baixa prioridade enquanto a VPN não é ativada. Tal tráfego será removido no momento da ativação da VPN. O TS atende os requisitos 2 e 7;

6.3.6 *Optical-VPN Service - O-VPNS*

Este serviço é responsável pela ativação da VPN. Ao ativar a VPN, o sistema de cobrança é então disparado para iniciar a tarifação do serviço. Cada domínio possui seu próprio sistema de *billing*, porém deveria haver uma padronização na forma como um serviço oferecido por domínios administrativos diferentes é tarifado. Algumas alternativas podem ser encontradas em [Fang et al., 2005]. Basicamente, existem duas possibilidades de tarifação na Internet atual. A primeira considera que os domínios de trânsito cobrem de seus usuários pelo serviço de entrega de dados tanto local quanto para outro domínio. A segunda possibilidade considera que os domínios (tipicamente de camada 1) especifiquem parcerias de forma que a entrega de dados seja feita sem cobrança especificamente para suas redes e seus clientes. Porém, novos tipos de modelos de tarifação são necessários para dar suporte aos novos tipos de serviço [Fang et al., 2005]. Tais modelos precisam considerar o tipo de serviço utilizado, nível de QoS, banda utilizada e freqüência de uso. Em redes ópticas outros fatores precisam ser considerados tal como o tipo de proteção.

O O-VPNS também oferece uma interface para que as VPNs sejam gerenciadas. Informações relacionadas a cada VPN estabelecida podem ser obtidas. As informações que podem ser obtidas pelos clientes sobre as VPNs depende dos contratos existentes entre clientes e provedores. No instante da ativação, tráfegos de baixa prioridade que estiverem usando a VPN deverão ser removidos. O O-VPNS atende os requisitos 2 e 7;

Pode-se observar na Figura 6.3 a presença de um módulo denominado *Registry*. Ele age como um sistema de registro e busca de serviços. Com ele, provedores registram seus serviços que serão oferecidos aos clientes. Por sua vez, os clientes realizam buscas no registro para obterem serviços que atendam suas necessidades. Além disso, a figura também apresenta uma PIB (*Policy Information Base*) e uma MIB (*Management Information Base*). Tais bases foram definidas especificamente para este trabalho e não possuem nenhuma relação com o modelo PCIM e CIM, respectivamente.

Finalmente, o módulo "serviços legados" permite que contratos previamente estabelecidos entre os domínios sejam considerados durante o provisionamento de serviços. Tipicamente, condições e regras de roteamento entre os domínios são controladas por este módulo.

Como mencionado na Seção 2.1.3, um processo de negócios é uma seqüência de atividades coordenadas e conectadas através de controles de fluxos. Nesta tese, dois processos de negócios foram identificados e correspondem aos dois serviços oferecidos entre domínios: estabelecimento de conexões e VPNs entre domínios. A seguir apresentamos como os dois processos de negócios são realizados. Primeiramente, apresentamos o processo para estabelecimento de conexões entre domínios. A seguir, ilustramos o processo para estabelecimento de VPNs entre domínios. A modelagem em BPMN de cada processo será apresentada. Antes de detalharmos cada processo, ilustramos a modelagem do processo de negócios em alto nível.

6.4 Modelagem dos Processos de Negócios

Nesta seção, os processos de negócios para estabelecimento de conexões e O-VPNs entre domínios utilizam o modelo *Push* para distribuição das topologias virtuais. Assim, os passos descritos e a modelagem realizada para provisionamento dos dois serviços são feitos considerando o modelo *Push*. A descrição do modelo *Pull* e sua comparação com o modelo *Push* será realizada no Capítulo 7.

6.4.1 Modelagem do Processo de Negócios em Alto Nível

A notação BPNM, como explicado na Seção 2.1.3, permite uma especificação em alto nível dos processos de negócios. Neste trabalho, usamos a notação para especificar os dois processos definidos no contexto de estabelecimento de serviços entre domínios em redes ópticas. Uma das vantagens da notação BPMN é a possibilidade de expansão de atividades permitindo que, em um primeiro momento, as atividades sejam definidas de forma resumida e, posteriormente, sejam expandidas para visualização interna dos detalhes. A Figura 6.7 apresenta o diagrama em alto nível do processo de negócios para estabelecimento de serviços entre domínios.

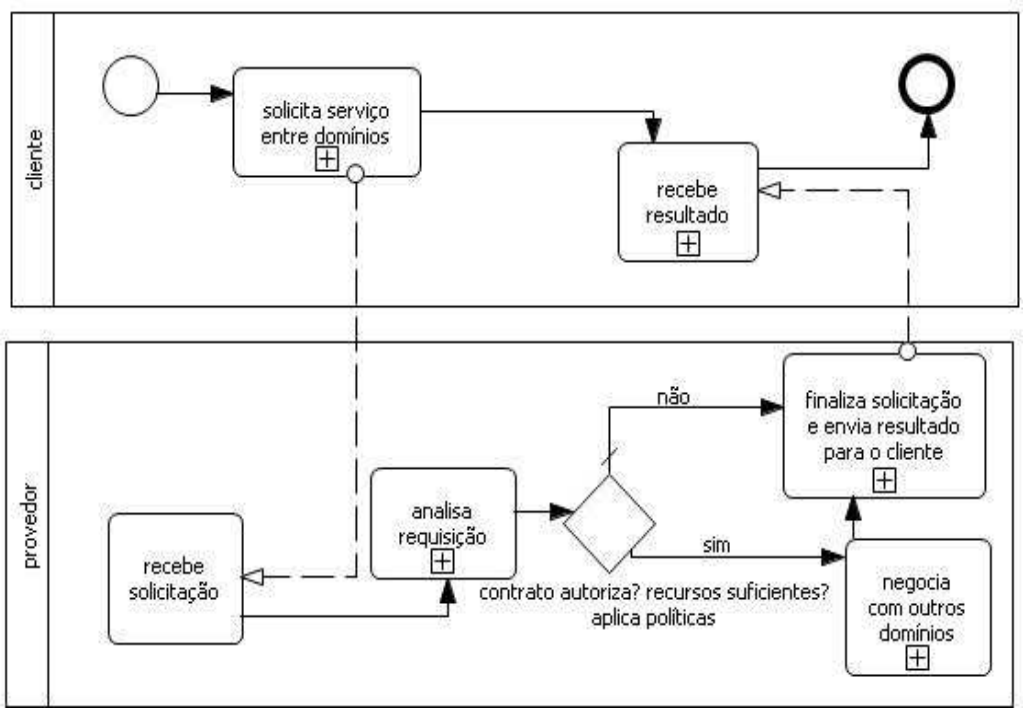


Fig. 6.7: Processo de negócios em alto nível.

A seqüência de passos se inicia pelo interesse do cliente em solicitar algum serviço do domínio provedor. A solicitação é montada e enviada para o provedor que, por sua vez, analisa a requisição.

Após analisar os contratos, verificar os recursos e aplicar políticas, o domínio decide por realizar a negociação com outros domínios para estabelecer o serviço ou encerrar a requisição. Se optar pela negociação e caso ela tenha sucesso, o serviço será estabelecido. O resultado é enviado ao cliente que então poderá usar o serviço solicitado. Note que apenas os passos em alto nível são apresentados. Com isso pode-se ter uma idéia geral de como as atividades serão organizadas para suportar cada serviço. Note também que neste diagrama em alto nível, não há menção sobre qual processo de negócios está sendo realizado (estabelecimento de conexão ou VPN). O diagrama pode ser usado tanto para um processo como para outro. A seguir, iremos apresentar os detalhes de cada processo e suas respectivas modelagens. Iremos expandir as duas principais atividades do diagrama: "analisa requisição" e "negocia com outros domínios".

6.4.2 Processo de Negócios para Estabelecimento de Conexões entre Domínios

Em resumo, o estabelecimento de uma conexão entre domínios consiste em receber a requisição, analisar os parâmetros no domínio local, encontrar uma rota, negociar com outros domínios e, em caso de sucesso, estabelecer um contrato. A negociação ocorre apenas se o domínio requisitante consegue atender os parâmetros solicitados pelo cliente. O domínio requisitante é aquele pelo qual o cliente fez a invocação do serviço. Tal domínio tem a flexibilidade de escolher outras rotas, caso a primeira não atenda aos requisitos de QoS informados pelo cliente. A quantidade de tentativas fica a critério do cliente e/ou do provedor e pode ser definida através de um parâmetro enviado na requisição feita pelo cliente. Esta quantidade de tentativas também depende do nível de divulgação de informações utilizado nas topologias virtuais. Se o nível mais restrito (apenas o custo abstrato do enlace) for usado, a probabilidade de encontrar uma rota que não atenda os requisitos de QoS aumenta. Se o nível menos restrito (todos os atributos de QoS estão presentes no enlace virtual) for usado, o cálculo da rota leva em conta apenas os enlaces que atendem ao(s) requisito(s) de QoS desejado(s).

A Figura 6.8 ilustra os passos necessários para o estabelecimento de uma conexão entre domínios.

Primeiramente, o cliente obtém o serviço desejado do registro (passo 1). Dependendo da forma como os serviços estão implementados, um localizador para a interface do serviço é retornado[1]. Após a busca do registro, o cliente pode invocar o E2ECS (passo 2) informando todos os parâmetros necessários para o estabelecimento da conexão. O E2ECS repassa a requisição para os módulos internos da camada de integração (passo 3). Esta interface entre a camada de serviços e a camada de integração deve ser bem definida de forma que as interações entre as duas camadas ocorram de maneira transparente. A camada de serviços não deve interferir na forma como as tarefas internas são realizadas. Com isso atendemos o pré-requisito número 9. Especificamente, a requisição é repassada

[1]No caso dos *Web Services* um localizador para a interface WSDL é retornado.

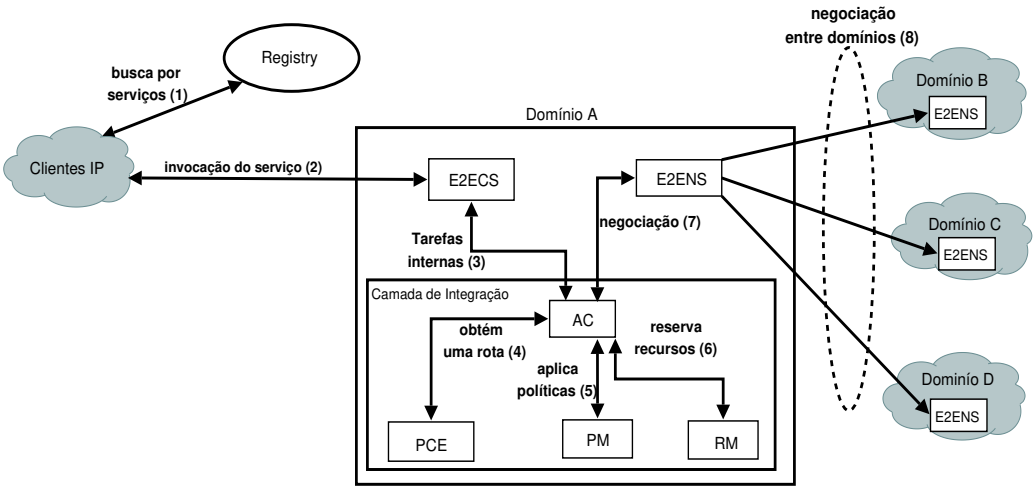


Fig. 6.8: Passos para estabelecer uma conexão entre domínios.

para o controle de admissão (AC) que então valida os parâmetros informados. O AC obtém uma rota entre domínios passando os nós de origem e destino para o PCE (passo 4). O PCE por sua vez monta uma topologia virtual única e encontra uma rota entre o par origem/destino. Após obter a rota, o AC invoca o gerente de políticas (PM) para que regras locais sejam aplicadas para permitir ou não o estabelecimento da conexão no domínio (passo 5). Neste cenário, as políticas são simples e apenas verificam se o cliente requisitante tem ou não permissão para estabelecer a conexão com os parâmetros solicitados. Se a conexão é permitida, o AC invoca o gerente de recursos (RM) a fim de reservar, para cada enlace virtual, um caminho de luz (passo 6). A Figura 6.9 mostra como é feita esta reserva.

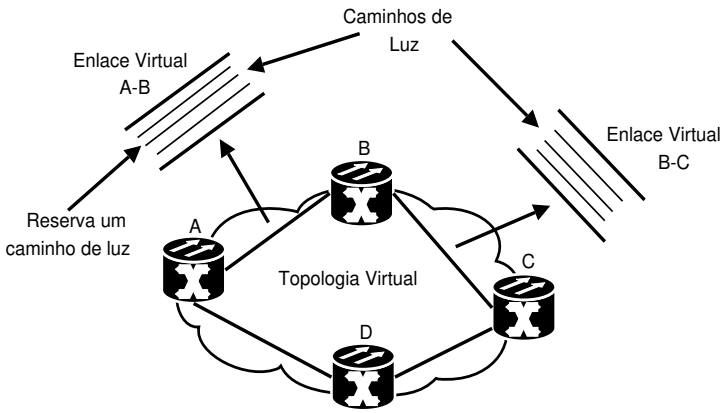


Fig. 6.9: Reserva de recursos nos enlaces virtuais.

Considerando que a rota escolhida para atravessar o domínio passe pelos enlaces virtuais A-B e B-C, dois recursos precisam ser reservados, um para cada enlace. Em cenários mais sofisticados, nos quais o cliente informa o tempo de início e fim da conexão, o RM age como um escalonador

e a seleção dos recursos deve levar em conta os horários de uso de cada recurso. Nesta tese, o provisionamento de serviços entre domínios considera apenas a disponibilidade do recurso.

Finalmente, os passos 7 e 8 representam a negociação entre domínios. Eles somente são executados caso os passos anteriores tenham tido sucesso no domínio local. O passo 8 representa tanto a fase de reserva como a fase de confirmação do protocolo de negociação. Nos outros domínios, os passos 3, 5 e 6 são executados para a realização das tarefas locais durante a primeira fase da negociação. Na segunda fase do protocolo de negociação, a reserva é confirmada ou desfeita caso algum domínio não aceite a conexão. Em caso de confirmação, a "crossconexão" nos OXCs de borda é então realizada. Em caso de rejeição, a reserva dos recursos locais em cada enlace virtual deve ser desfeita. Atualmente, o protocolo de negociação definido para este trabalho não aceita contra-propostas.

Em alguns casos e dependendo dos relacionamentos definidos entre os domínios, a rota encontrada pelo PCE não pode ser usada. Nestes casos, rotas BGP devem ser consideradas a fim de atender os contratos pré-definidos. Isto pode ocorrer no modelo *Pull* que obtém as VTs apenas de algumas rotas. Tais rotas podem ser aquelas divulgadas pelo BGP. No modelo *Push*, onde há uma distribuição de VTs entre os domínios, a rota PCE será utilizada para o estabelecimento das conexões. Estas relações podem ser controladas por algum módulo localizado no plano de gerência. O módulo "serviços legados" na arquitetura representa estas relações.

A Figura 6.10 apresenta o diagrama BPMN que modela o processo para estabelecimento de conexões entre domínios apresentado acima. O diagrama mostra, principalmente, a expansão da atividade "analisa requisição" do processo em alto nível apresentado na Figura 6.7. Porém, a atividade "solicita serviço entre domínios" também foi expandida. Do lado esquerdo do diagrama aparecem os módulos responsáveis pelas atividades. Note no participante domínio provedor, a separação entre os serviços de relacionamento e os serviços utilitários. De um lado (parte superior do diagrama), o E2ECS oferece uma interface para a invocação do serviço e portanto, representa a interação com os clientes. Do outro lado (parte inferior do diagrama), o E2ENS (serviço utilitário) realiza a atividade de negociação para suportar o serviço solicitado. O E2ECS não invoca diretamente o E2ENS. Todas as atividades locais necessárias para prover o serviço são realizadas pela (e através da) camada de integração.

A maioria das atividades representa a descrição dos passos da Figura 6.8. Porém, algumas atividades merecem breves comentários. A atividade "realiza reserva de recursos" é um sub-processo que consiste de outras atividades menores tais como reservar recursos internos ao domínio e reservar um recurso entre domínios (caminho de luz que conecta o domínio local ao próximo domínio na rota). A atividade "desfaz reserva no domínio local" também possui sub-processos para desfazer as reservas dos recursos locais. A atividade "finaliza solicitação" possui sub-processos para coordenação

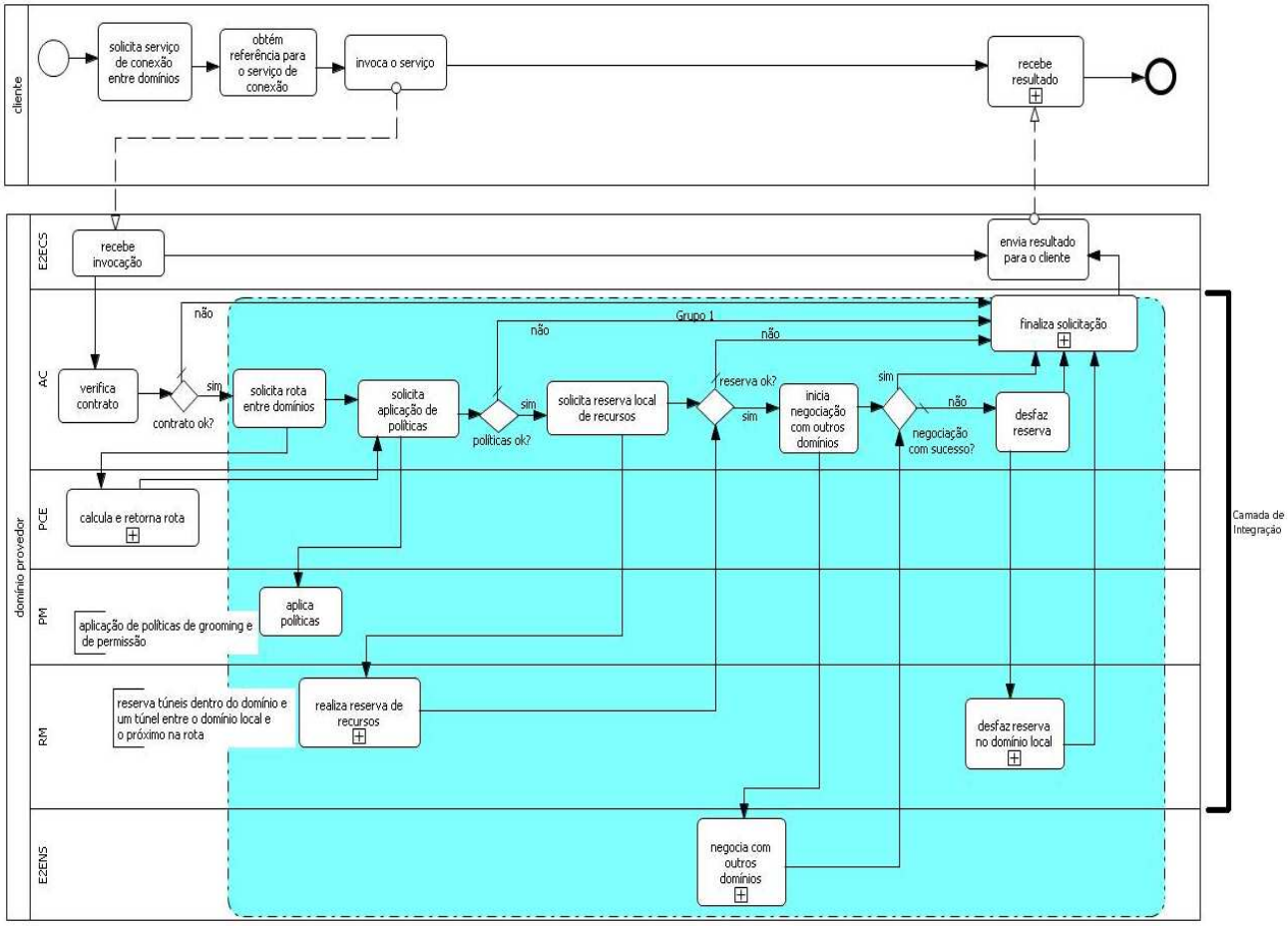


Fig. 6.10: Diagrama BPMN para o processo de estabelecimento de conexões entre domínios.

dos resultados, montagem da mensagem de retorno dependendo dos resultados (sucesso, falha, não permissão, etc.) e envio da mensagem para o E2ECS. O Grupo 1, demarcado pela caixa pontilhada ao redor da maioria das atividades do domínio provedor, será usado como um agrupamento de atividades a serem realizadas durante o processo de estabelecimento de VPNs entre domínios. A atividade "negocia com outros domínios", como já mencionado e devido a sua complexidade, é expandida no diagrama BPMN da Figura 6.11.

A atividade "negocia com outros domínios" inicia a negociação entre domínios ao receber a requisição do módulo AC. O E2ENS do domínio invocador interage com o E2ENS do(s) domínio(s) invocado(s). O participante denominado "domínio invocado" representa os E2ENS dos outros domínios. As atividades "executa primeira fase do protocolo de negociação" e "executa segunda fase do protocolo de negociação" possuem o símbolo de múltiplas instâncias da notação BPMN pois a quantidade de atividades a serem instanciadas depende da quantidade de domínios que serão invocados para realizar a negociação. Em termos técnicos, as múltiplas instâncias são implementadas através de *Threads*.

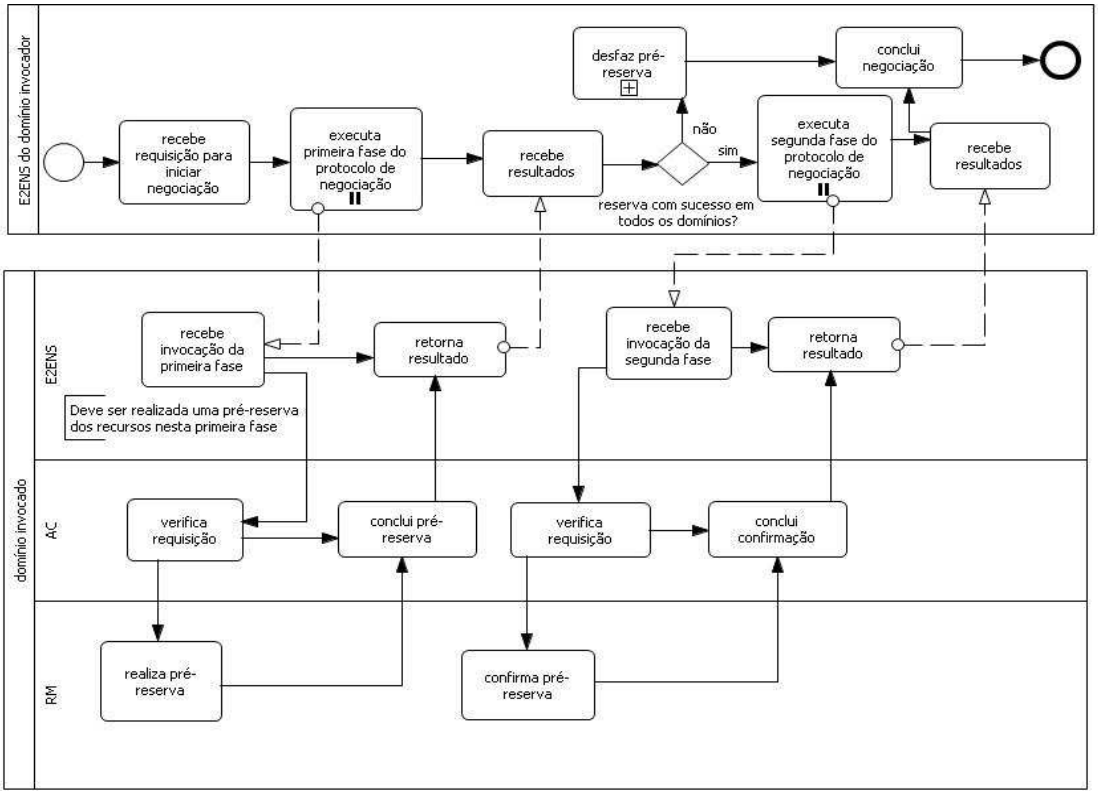


Fig. 6.11: Diagrama BPMN para a atividade “negocia com outros domínios”.

Entre a primeira e a segunda fase do protocolo de negociação, o E2ENS do domínio invocador verifica se todos os domínios invocados conseguiram realizar a reserva. Em caso positivo, a segunda fase do protocolo é iniciada. Caso contrário, deve-se desfazer a reserva nos domínios onde ela foi realizada. A atividade “desfaz pré-reserva” é expandida na Figura 6.12.

A atividade “invoca domínio para desfazer pré-reserva” também possui o símbolo de múltiplas instâncias uma vez que a quantidade de domínios cuja reserva deve ser desfeita depende da quantidade de domínios que conseguiram realizar a reserva.

A seguir, apresentamos o processo de negócios para o serviço de estabelecimento de VPNs entre domínios ópticos.

6.4.3 Processo de Negócios para Provisionamento de VPNs entre Domínios

O provisionamento de VPNs entre domínios basicamente consiste em estabelecer conexões ópticas que conectem os pares de CPIs definidas pelos clientes das VPNs. A principal diferença entre o estabelecimento de uma VPN e uma conexão se refere ao fato de que a VPN necessita de um controle de membros a fim de correlacionar quais pares de portas pertencem a quais VPNs, evitando-se assim que clientes de uma VPN usem portas de outras VPNs. Em um domínio local,

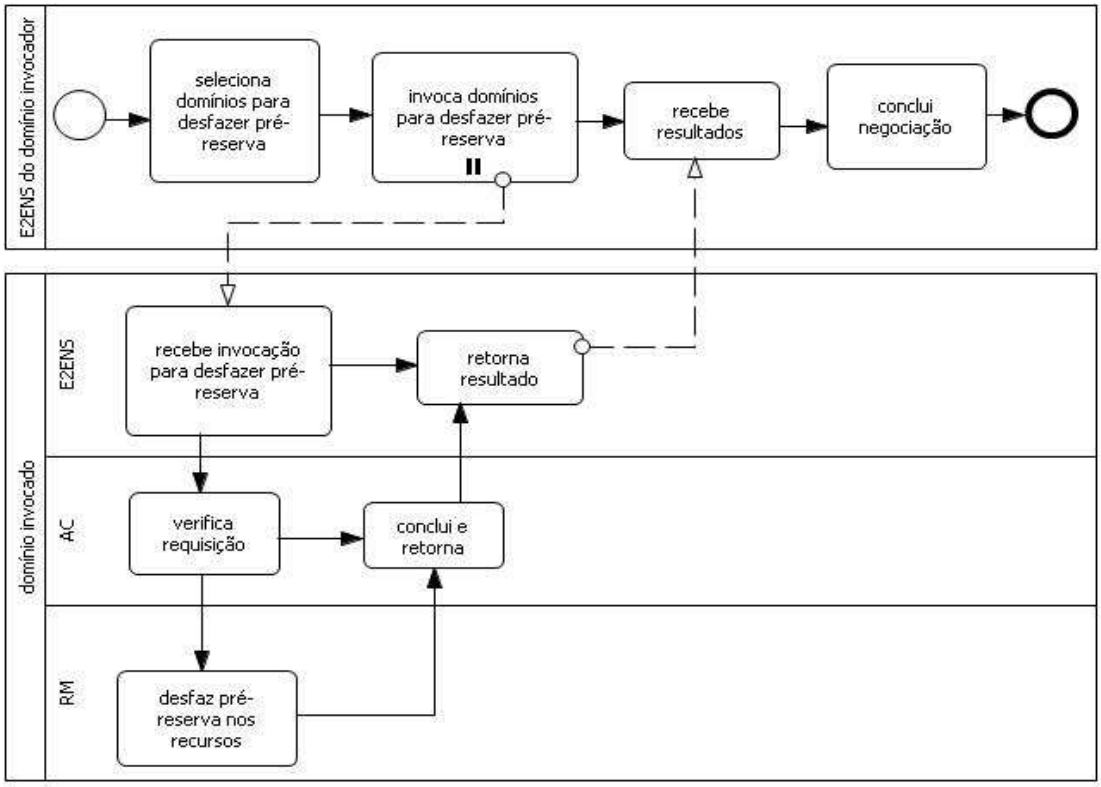


Fig. 6.12: Diagrama BPMN para a atividade “desfaz pré-reserva”.

conhecer as portas e seus mapeamentos em cada VPN é uma tarefa fácil pois apenas um domínio administrativo está envolvido. Porém, estabelecer uma VPN entre domínios exige que cada domínio tenha conhecimento das portas que pertencem a cada VPN em outros domínios. Uma solução para isto seria usar o BGP. Entretanto, o BGP apenas divulga alcançabilidade e precisaria ser estendido para suportar esta funcionalidade. Além disso, estamos interessados em prover VPNs que possam levar em conta aspectos de QoS. O BGP não suporta este tipo de informação.

Ao caracterizarmos que uma VPN nada mais é do que um conjunto de conexões ópticas (SPCs), e devido ao fato de que já tínhamos criado todo o suporte para prover tais conexões, concluímos que a principal funcionalidade ainda não atendida pela arquitetura era a distribuição de informação de pares de portas das VPNs. Esta funcionalidade foi então incorporada ao serviço de divulgação conforme explicado anteriormente, ficando o AS responsável pela divulgação de topologias virtuais e divulgação de informação de correlação de portas das VPNs.

Em cada domínio óptico, os pares de portas são definidos localmente pelo administrador do domínio. Tais pares são unicamente representados pelos seus identificadores de porta CPI e PPI conforme explicado na Seção 5.2.2. Esta configuração é feita de forma estática e o conjunto destes pares em cada domínio forma uma *Port Information Table*(PIT) [Takeda et al., 2005] para cada VPN. Cada PIT deve ser então enviada para os outros domínios que possuam pelo menos uma porta

pertencente à VPN correspondente à PIT distribuída. Em cada domínio, as PITs são recebidas e agrupadas em uma única PIT para cada VPN. A Figura 6.13 ilustra um cenário onde a VPN A está distribuída em dois domínios administrativos diferentes (domínio X e domínio Y).

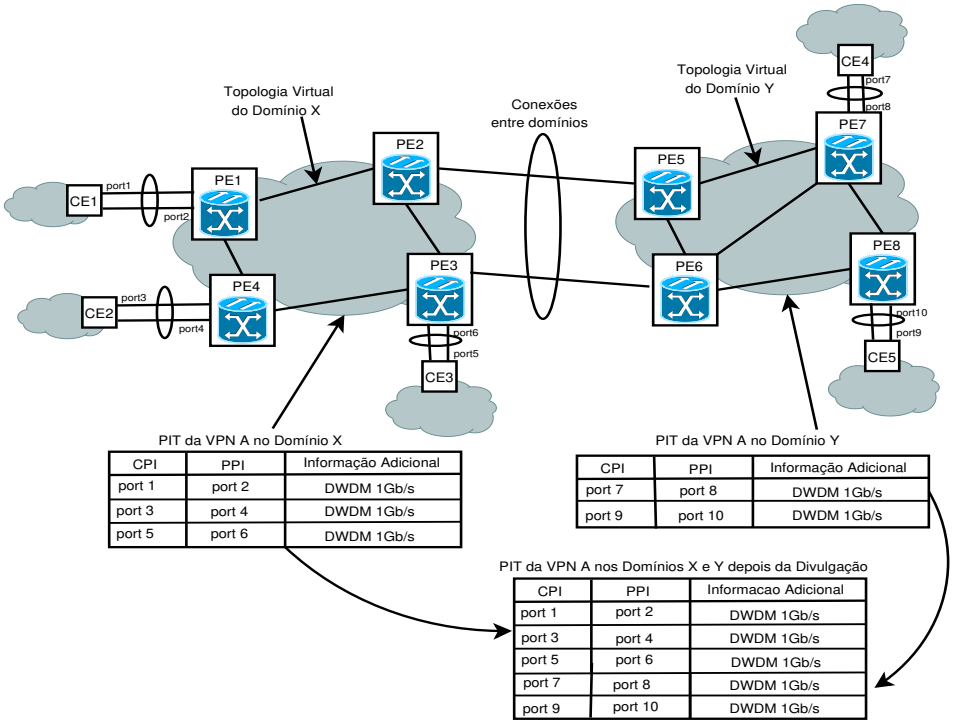


Fig. 6.13: Exemplo de distribuição de PITs.

No domínio X, a PIT possui os pares de portas para a VPN A naquele domínio. No domínio Y, a VPN A também possui sua PIT. Após a distribuição das tabelas entre os domínios X e Y, uma única PIT em cada domínio é então formada. Com isso, clientes dos dois domínios podem solicitar o estabelecimento de VPNs cujas portas estão localizadas em outros domínios. O gerente de membros (MM) é o responsável pela correlação de portas e membros a fim de analisar se a conexão solicitada entre os pares de portas pode ou não ser estabelecida. Os clientes podem solicitar do provedor as informações de quais portas eles podem estabelecer VPNs, caso eles ainda não tenham esta informação. Esta funcionalidade deve ser oferecida pelos domínios que suportam o serviço de VPN.

O estabelecimento de uma VPN consiste no envio, por parte dos clientes, dos pares de portas CPIs cujas conexões devem ser estabelecidas. Após a verificação de portas feita pelo MM, o estabelecimento de cada conexão segue exatamente o mesmo processo explicado na seção anterior. Isto fica claro ao observarmos a Figura 6.14 que mostra os passos para estabelecimento de uma VPN.

Note que apenas o passo 3.1 foi adicionado para o estabelecimento de VPNs. Após a verificação das portas, o controle de admissão (AC) coordena o estabelecimento do conjunto de conexões que

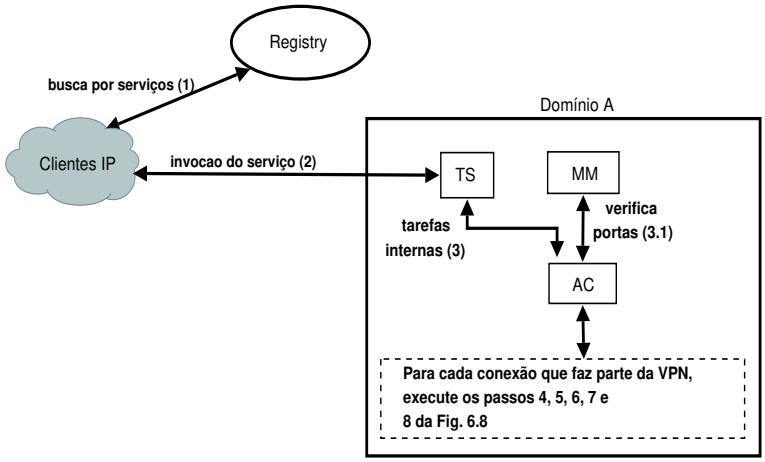


Fig. 6.14: Passos para estabelecer uma VPN entre domínios.

formam a VPN. O estabelecimento das conexões é feito entre os pares de portas PPIs. A identificação destes pares é realizada através da tupla CPI/PPI na tabela PIT. Observe que a Figura 6.14 representa a reserva de recursos feita pelo TS. A ativação e a gerência da VPN é feita pelo O-VPNS.

A modelagem BPMN para o processo de estabelecimento de VPNs entre domínios é apresentada na Figura 6.15. A atividade "realiza atividades do Grupo 1 para cada par de portas" representa a execução das atividades agregadas no Grupo 1 apresentado no diagrama BPMN da Figura 6.10. Como mencionado anteriormente, o provisionamento de uma VPN consiste no estabelecimento de conexões entre os pares de portas informados pelo cliente. O conjunto de atividades representado pelo Grupo 1 realiza as tarefas necessárias para o estabelecimento de cada conexão entre cada par de portas individualmente.

6.5 Discussão Final

Para finalizar este capítulo, é importante destacar a maneira como o desenvolvimento da arquitetura evoluiu. Primeiramente, iniciamos a definição de uma arquitetura responsável pelo provisionamento de serviços dentro de um domínio. Focamos no fornecimento de conexões (SPCs) e VPNs internas a um domínio. Especial atenção foi dedicada ao uso de políticas, principalmente relacionadas à agregação (*grooming*) de tráfego IP/MPLS nos caminhos de luz da rede óptica. As políticas também incorporaram aspectos relacionados ao impacto das falhas em redes ópticas. Tais políticas tentam minimizar os efeitos negativos de tais falhas. A arquitetura para provisionamento de VPNs foi implementada e testada e também incorporou políticas de prioridade para estabelecimento de VPNs.

A arquitetura evoluiu de forma a oferecer serviços que fossem além de um domínio local. A camada de serviços foi definida de forma a suportar dois serviços de negócios, o serviço de conexão

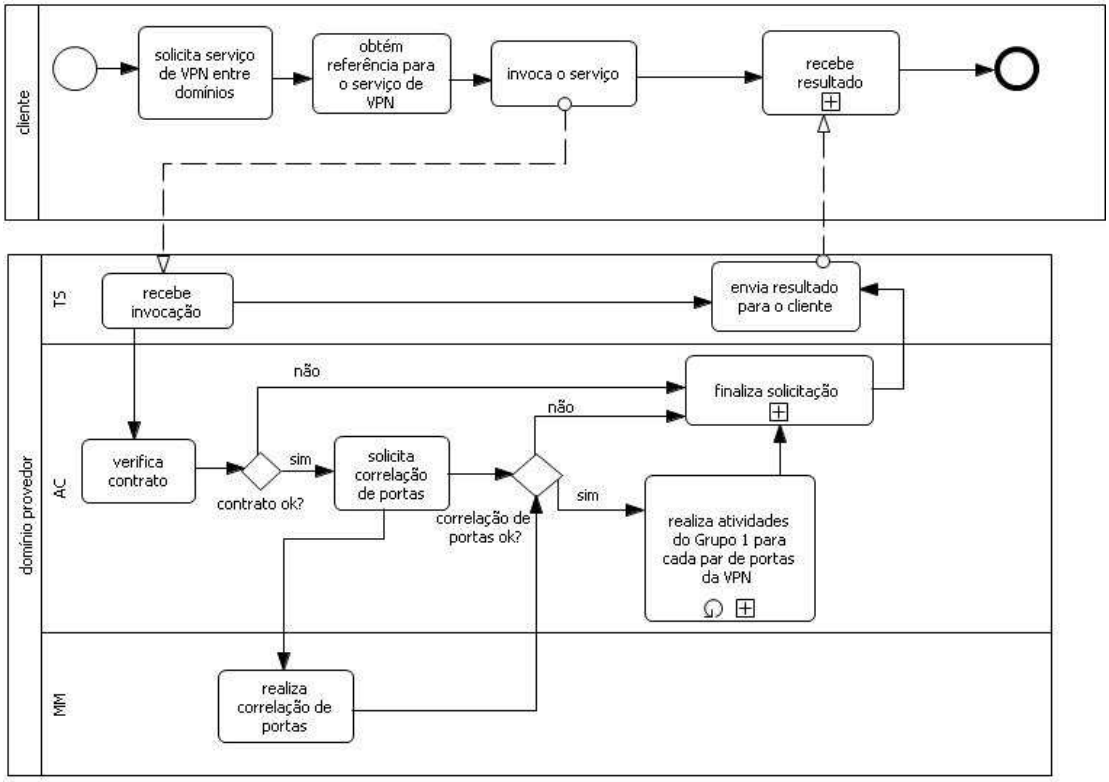


Fig. 6.15: Diagrama BPMN para o processo de estabelecimento de VPNs entre domínios.

e o serviço de VPN entre domínios. Manter a arquitetura simples foi um dos principais objetivos também nesta fase. Pode-se perceber que o serviço de VPN entre domínios não exigiu grande esforço para o desenvolvimento uma vez que foi baseado nas funcionalidades já criadas para o serviço de conexão. A separação entre os módulos da camada de serviços e os módulos internos da camada de integração em cada domínio permite que extensões sejam feitas separadamente. Com isso, conseguimos atender ao pré-requisito número 10. A adição de outros serviços à arquitetura se torna uma tarefa flexível uma vez que a camada de serviços apenas expõe a interface de acesso aos serviços. Os módulos internos da camada de integração são os responsáveis efetivamente pelo controle do provisionamento dos serviços em cada domínio. Além disso, a interação entre os serviços de relacionamento e os serviços utilitários é feita através da camada de integração, garantindo uma separação lógica entre os dois blocos facilitando a extensão dos serviços atuais e a criação de novos serviços.

A modelagem usando BPMN dos processos e das atividades permitiu mostrar como os módulos interagem e quais são as atividades de cada módulo. A definição de processo de negócios facilitou a identificação das atividades necessárias para os serviços de estabelecimento de conexões e VPNs entre domínios. Um refinamento das principais atividades foi feito na forma de expansão através da notação BPMN, facilitando a implementação da arquitetura e seus módulos.

No próximo capítulo iremos apresentar os detalhes relacionados ao projeto e implementação da arquitetura proposta nesta tese. Iremos ilustrar os cenários de testes assim como os resultados obtidos. O foco do capítulo será exclusivamente direcionado para o provisionamento dos serviços entre domínios uma vez que os detalhes dos serviços locais foram explorados nas dissertações de mestrado e nos artigos mencionados no Capítulo 5.

Capítulo 7

Implementação, Validação e Resultados Obtidos

Neste capítulo apresentamos os aspectos relacionados com a implementação da arquitetura proposta no capítulo anterior. O objetivo principal é validar a arquitetura e seus módulos através da implementação de um protótipo. Obviamente, não iremos esgotar todos as possibilidades de cenários mas sim, analisar como os módulos da camada de integração e os módulos da camada de serviços interagem para prover os serviços definidos. A implementação levou em consideração a modelagem em BPMN realizada no capítulo anterior. As atividades definidas para cada módulo serão implementadas a fim de suportar os serviços de estabelecimento de conexões e VPNs entre domínios. Iremos mostrar as ferramentas e tecnologias usadas na implementação e, principalmente, avaliar o uso dos *Web Services* para o provisionamento de serviços entre domínios.

A avaliação do protótipo consiste em medir os tempos para estabelecimento de conexões e VPNs entre domínios assim como o consumo de banda para prover tais serviços. Apresentaremos apenas os detalhes e as análises especificamente para a funcionalidade de estabelecimento de serviços entre domínios, embora o protótipo suporte também a remoção e a consulta dos serviços estabelecidos. Além disso, iremos também apresentar uma análise sobre o tamanho da mensagem SOAP e tempo para distribuição das topologias virtuais.

Nesta tese estamos particularmente interessados em três áreas de gerência: falha, configuração, e desempenho. Aspectos relacionados com a segurança são, sem dúvida, de grande importância principalmente em um cenário que envolve diferentes domínios administrativos. Porém, entendemos que as ferramentas atualmente disponíveis para controle de segurança estão suficientemente maduras e poderiam ser usadas na nossa arquitetura de forma natural. Um simples mecanismo de segurança seria a utilização de HTTPS para a troca de mensagens SOAP fazendo uso de algum mecanismo de autenticação no lado servidor. Além disso, mecanismos de criptografia poderiam ser usados para

aumentar o grau de segurança para a troca de mensagens. Recentemente, o consórcio OASIS definiu um padrão de segurança para *Web Services*. Tal padrão é conhecido como *Web Services Security* [WS-Security, 2004] atualmente em sua versão 1.1, e propõe o uso de mecanismos para prover integridade e confidencialidade nas mensagens SOAP. Em relação aos aspectos relacionados com a contabilidade, dedicamos uma seção neste capítulo para discutirmos brevemente possíveis soluções para tarifação entre domínios.

Os testes foram feitos em nosso laboratório usando primeiramente 5 máquinas. Após, testes foram feitos em um cluster usando 8 máquinas. Cada máquina representa um domínio óptico. As topologias virtuais em cada domínio são armazenadas em arquivos XML que são criados de forma estática. Cada topologia virtual representa os caminhos de luz criados em cada domínio óptico. A arquitetura foi inteiramente implementada em Java 1.4 e algumas ferramentas não comerciais foram utilizadas para facilitar a implementação. Para dar suporte ao desenvolvimento e execução dos *Web Services* utilizou-se a plataforma AXIS 1.2 [Apache AXIS, 2006] e o servidor Apache Tomcat 5.0.18 [Apache Tomcat, 2006] para abrigar os *Web Services*. Para auxiliar na manipulação das informações de gerência em XML foram utilizadas as APIs SAX, DOM e XPath. Para a interação do usuário com o sistema de gerenciamento foi criada uma interface baseada na tecnologia JSP e executada no servidor Apache Tomcat. Todos os testes foram feitos em nossa intranet usando máquinas P4 3GHz (*HT-enabled*) com 1GB de memória RAM e placa de rede 10/100 Mb/s. Todas com o sistema operacional Linux Slackware versão 10.

Organizamos este capítulo da seguinte forma: primeiramente apresentamos os detalhes para o provisionamento de conexões entre domínios. Discutiremos pontos importantes e decisões de projeto que servem como base para a implementação da arquitetura. Informações sobre a avaliação em termos de tempo e consumo de banda serão apresentadas. A seguir, focamos no provisionamento do serviço de VPN entre domínios. Mostraremos como as PITs são representadas em cada domínio e uma análise de tempo de estabelecimento de uma VPN entre domínios será apresentada. Após, comentamos brevemente algumas idéias iniciais sobre o processo de tarifação entre diferentes domínios administrativos. Em seguida comparamos os modelos *Pull* e *Push*. Mostraremos como o modelo *Pull* foi integrado com o BGP em um cenário real criado em nosso laboratório. Finalmente, apresentaremos algumas ferramentas de auxílio que foram desenvolvidas pelo nosso grupo a fim de facilitar os testes do protótipo. Desenvolvemos um serviço de registros próprio que permite o cadastro e busca de *Web Services*. Também desenvolvemos uma aplicação para criação e distribuição de topologias virtuais. Esta ferramenta permite monitorar o consumo dos recursos em cada domínio óptico.

7.1 Detalhando o Serviço de Conexões entre Domínios

7.1.1 Implementação

Nesta tese, o modelo TMN serve somente como base para a nomenclatura das camadas de gerência e para a identificação das interfaces. A implementação da camada de serviços ocorre através de *Web Services*. A implementação da camada de gerência de redes (camada de integração) ocorre usando RMI. Porém, a tecnologia usada internamente em cada domínio é uma decisão local e outras tecnologias poderiam ser utilizadas tais como CORBA, J2EE, etc., e não devem influenciar na forma como os serviços são oferecidos na camada de serviços. A interface *x* que conecta dois modelos TMN em domínios administrativos diferentes foi implementada usando WSDL. O trabalho citado em [Chen and Li, 2005] também representa a interface *x* usando *Web Services*. A interface *q3* que conecta a camada de serviços com a camada de gerência de redes foi implementada usando RMI. A Figura 7.1 identifica estas interfaces.

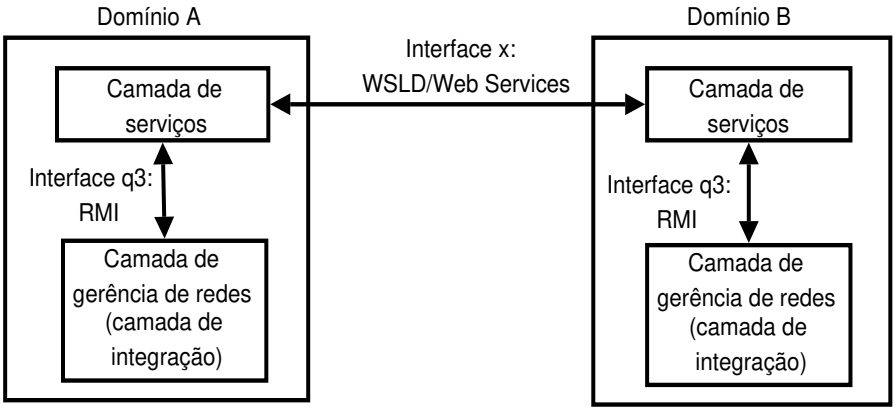


Fig. 7.1: Identificação das interfaces.

As interações que representam a tecnologia de comunicação utilizada entre cada módulo são apresentadas na Figura 7.2. A figura mostra as interações entre os módulos que estão na mesma camada (camada de serviços e camada de gerência de redes) e também as interações entre os módulos de camadas diferentes. Tal figura será novamente utilizada ao apresentarmos a avaliação dos tempos e o consumo de banda para o estabelecimento de conexões entre domínios.

Os blocos pontilhados representam os serviços da camada de serviços (serviços de relacionamento e serviços utilitários) e os blocos contíguos representam os módulos da camada de integração da arquitetura. Os módulos da arquitetura mostrados na Figura 7.2 estão disponibilizados de forma fisicamente centralizada em uma máquina em cada domínio. Não há uma distribuição dos módulos em várias máquinas em cada domínio. A seguir, apresentamos resumidamente as funcionalidades oferecidas pelas interfaces dos serviços e pelas interfaces dos módulos internos. Os módulos usados

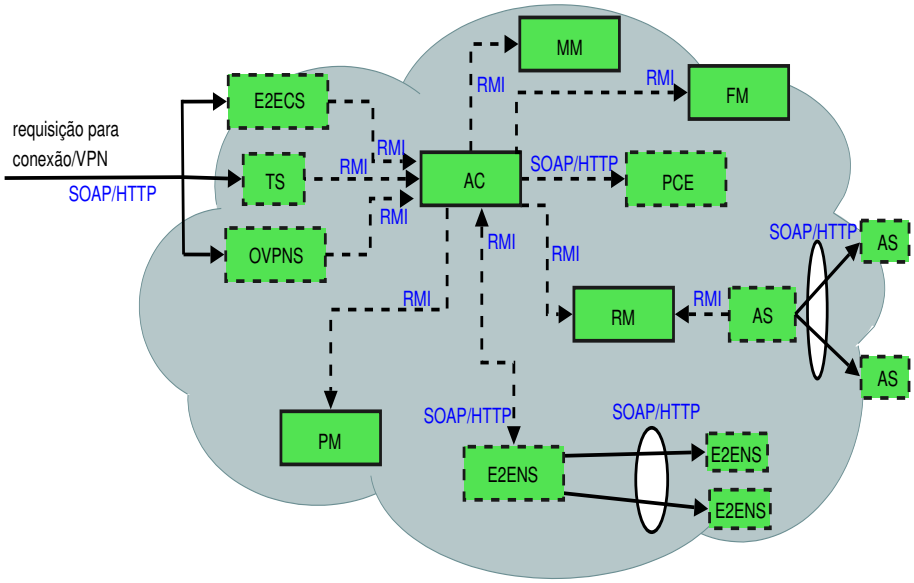


Fig. 7.2: Tecnologias de comunicação para interação entre os módulos da arquitetura.

apenas para o provisionamento de VPNs entre domínios serão apresentados na Seção 7.2.

A interface do serviço E2ECS oferece principalmente as funcionalidades para estabelecimento/remoção de SPCs, estabelecimento/remoção de conexões entre domínios e estabelecimento/remoção de VPNs dentro de um domínio. Algumas funcionalidades relacionadas à gerência são também fornecidas tais como listagem de conexões (intra e inter-domínios), obtenção de topologias virtuais, listagem das rotas das conexões, quantidade de banda usada e disponível nos caminhos de luz, etc. Um trecho do WSDL do serviço E2ECS é mostrado na Figura C.2 do Apêndice C.

As funcionalidades oferecidas pelo serviço de negociação (E2ENS) estão relacionadas ao protocolo de duas fases para a reserva dos recursos de cada conexão entre domínios. O E2ENS implementa o protocolo de duas fases e realiza o *rollback* quando necessário. O WSDL do E2ENS é apresentado na Figura C.3 do Apêndice C.

O AS possui em sua interface as funcionalidades para divulgação e obtenção de topologias virtuais, assim como funcionalidades para distribuição de informações sobre a correlação de portas das VPNs entre domínios. Uma topologia virtual é formada por nós físicos e enlaces virtuais e está descrita em um arquivo XML. Apesar de termos definidos três níveis de divulgação de topologias virtuais (ver Capítulo 4), para este protótipo implementamos o nível mais restrito, ou seja, aquele que divulga em cada enlace virtual apenas o custo abstrato do enlace. A Figura 7.3 mostra um exemplo de uma topologia virtual descrita em XML.

A figura representa a topologia virtual de um domínio denominado “mosqueiro” que possui três nós (“mosqueiro/0”, “mosqueiro/1” e “mosqueiro/2”) e três enlaces virtuais: “mosqueiro/1”

```
<?xml version="1.0"?>
<graph>
<node id="mosqueiro/0" weight ="0"/>
<node id="mosqueiro/1" weight ="0"/>
<node id="mosqueiro/2" weight ="0"/>
<edge source="mosqueiro/1" target="mosqueiro/2" weight="10"/>
<edge source="mosqueiro/0" target="mosqueiro/1" weight="6"/>
<edge source="mosqueiro/0" target="mosqueiro/2" weight="6"/>
</graph>
```

Fig. 7.3: Descrição em XML de uma topologia virtual.

conectado com "mosqueiro/2" e custo 10, "mosqueiro/0" conectado com "mosqueiro/1" e custo 6 e finalmente "mosqueiro/0" conectado com "mosqueiro/2" e custo 6. Um trecho do WSDL do AS é apresentado na Figura C.4 do Apêndice C.

O PCE possui apenas uma funcionalidade em sua interface e está relacionada à obtenção de uma rota entre dois nós, estejam eles localizados dentro do mesmo domínio ou em domínios diferentes.

Em relação aos módulos internos, temos que o controle de admissão (AC) age como uma interface entre a camada de serviços e as funções locais. O AC é um controlador que recebe as requisições provenientes da camada de serviços, as processa e usa os outros módulos internos conforme sua necessidade. Todas as funcionalidades oferecidas pelos serviços de relacionamento E2ECS, TS e OVPNS devem estar de alguma forma representadas na interface do AC. Algumas delas são: criação/remoção de SPCs, criação/remoção de conexões entre domínios, reserva/liberação de recursos de VPNs, ativação/desativação de VPNs, além de algumas funcionalidades de gerência como listagens, consultas, etc.

As funcionalidades dos outros módulos internos são mostradas no diagrama de classes completo, incluindo os serviços da camada de serviços e todos os módulos que compõem a camada de gerência de redes. O diagrama de classes é apresentado na Figura A.3 do Apêndice A.

O diagrama de seqüência para estabelecimento de uma conexão fim-a-fim entre domínios considerando o caso de sucesso é apresentado na Figura 7.4. O diagrama de seqüência mostra os passos necessários e as interações que ocorrem para estabelecer uma conexão entre domínios.

O cliente ou o gerente do domínio invoca o E2ECS para solicitar o estabelecimento de uma conexão entre domínios (passo 1). O E2ECS encaminha a requisição para o AC (passo 2) que valida os parâmetros e solicita uma rota ao PCE (passo 3). Após a aplicação de políticas (passo 4)[1], o AC invoca o RM para realizar a reserva dos túneis intra e inter-domínios para a conexão solicitada (passo 5). Após a reserva local, o AC dispara o processo de negociação (passo 6). O E2ENS é responsável por negociar com os outros domínios. Os passos 7 e 8 representam a fase de reserva do protocolo

[1] As políticas de *grooming* podem ser aplicadas aqui.

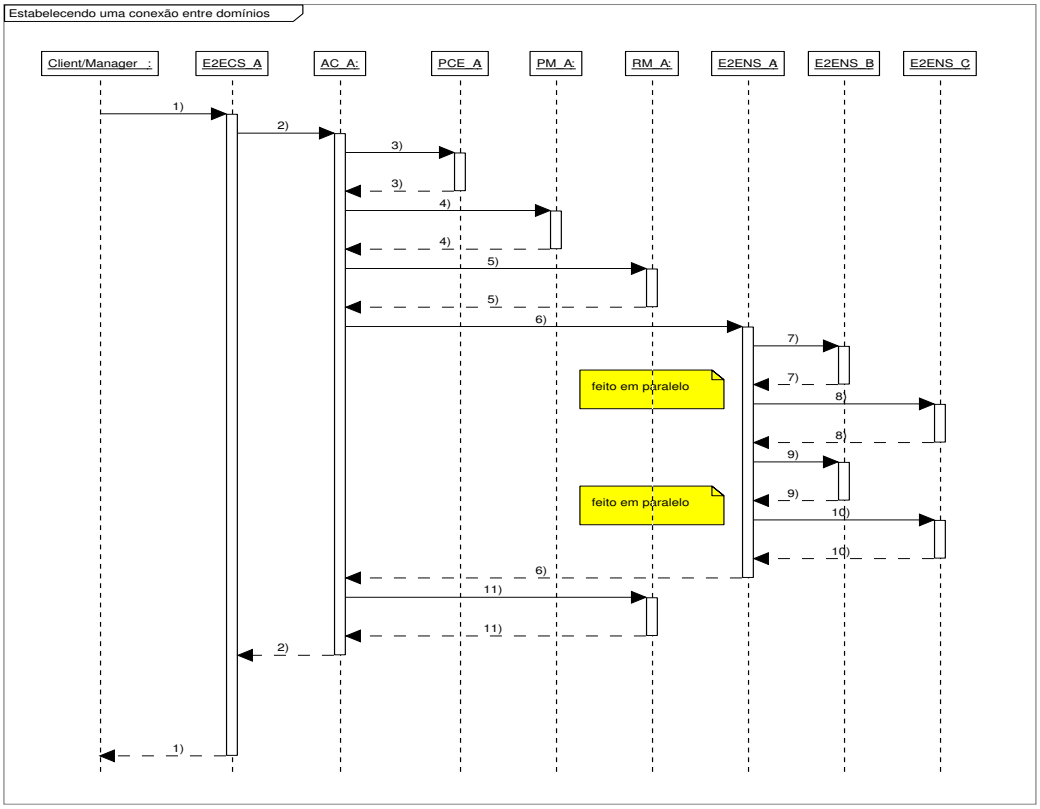


Fig. 7.4: Diagrama de seqüência simplificado para estabelecimento de conexões entre domínios.

de negociação. Estes passos são executados em paralelo. Os passos 9 e 10 representam a fase de confirmação da reserva e também são executados em paralelo através de *threads*. O passo 11 cria um objeto que representa a conexão entre domínios estabelecida e armazena tal objeto no RM. Em cada domínio, quando a reserva é confirmada (segunda fase do protocolo de negociação), um contrato é gerado com as informações da conexão sendo estabelecida. Não definimos nenhum mecanismo para a monitoração do contrato estabelecido. Tal tarefa é deixada como trabalho futuro.

A negociação entre domínios é detalhada no diagrama de atividades apresentado na Figura 7.5. O estado inicial representa o momento em que o E2ENS recebe a invocação do AC para iniciar a negociação. Note a semelhança dos fluxos e das atividades da modelagem BPMN apresentada na Figura 6.11 com o diagrama de atividades apresentado abaixo. A principal diferença entre os dois está relacionada ao fato de que enquanto a modelagem em BPMN do mecanismo de negociação descreve as atividades em mais alto nível, o diagrama de atividades em UML permite incorporar detalhes técnicos que se aproximam da implementação.

Para cada etapa da negociação, um tipo de *thread* foi definido. O diagrama de classes da Figura 7.6 mostra os três tipos de *threads*. Uma classe para a fase de reserva (*E2EThread*); outra classe para a fase de confirmação da reserva para os casos de sucesso (*CommitReserveThread*), e a terceira

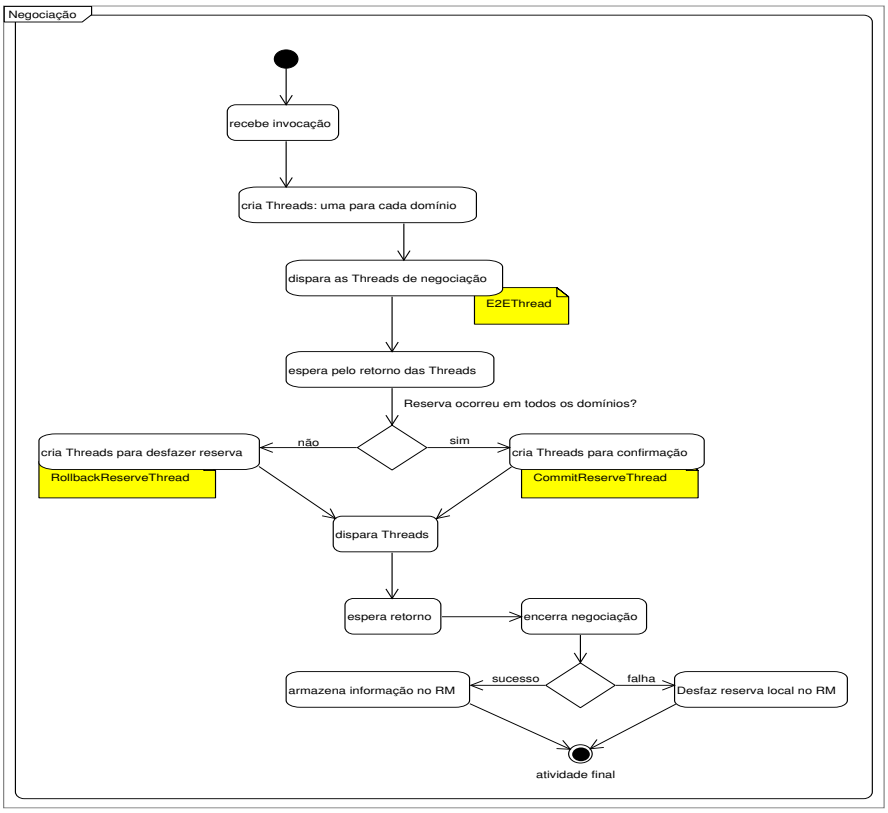


Fig. 7.5: Diagrama de atividades para a negociação.

classe para a fase de cancelamento de reserva em caso de falha de reserva em outros domínios (*RollbackReserveThread*). Primeiramente, o E2ENS cria *threads* para realizar a fase de reserva da negociação. Uma *thread* para cada domínio a ser negociado será criada. Note no diagrama de classes que a *thread* para a fase de reserva entre domínios possui os atributos que serão negociados com os outros domínios. O E2ENS espera pelo retorno de todas as *threads* e então analisa se foi possível realizar a reserva em todos os domínios. Em caso positivo, o E2ENS cria e dispara as *threads* para a confirmação da reserva. Se algum domínio não conseguiu realizar a reserva, o E2ENS cria e dispara *threads* para desfazer as reservas nos domínios onde elas foram possíveis. O encerramento da negociação pode ocorrer de duas maneiras. Em caso de sucesso (a reserva em todos os domínios foi possível), um objeto do tipo *InterDomainConnection* (veja a Figura 7.7) é criado e armazenado no RM. O objeto representa uma conexão entre domínios. Ele armazena os túneis selecionados no domínio local, a rota da conexão entre domínios, o túnel no sentido *upstream* que conecta o domínio local com o próximo domínio e o identificador único da conexão. Em caso de falha (a negociação não conseguiu reservar todos os recursos), a reserva feita previamente nos recursos locais deve ser desfeita.

A Figura 7.7 mostra algumas das principais classes do modelo de informação utilizado. A classe

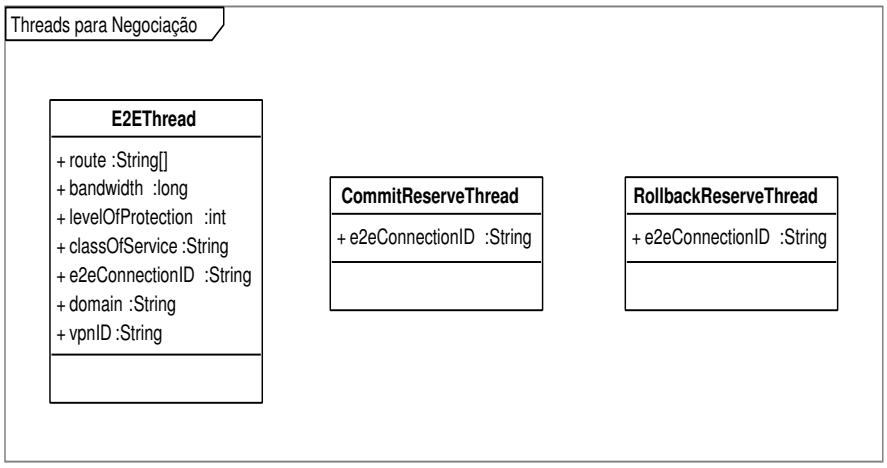


Fig. 7.6: Threads definidas para a negociação.

Tunnel possui alguns atributos usados na fase de negociação entre domínios. O atributo *reserved* é marcado como *true* na primeira fase da negociação garantindo a reserva no caminho de luz. O atributo *used* é marcado *true* na segunda fase da negociação para confirmar a reserva. Este mecanismo garante que um determinado recurso, após reservado, não estará disponível para atender outras conexões.

A classe *Tunnel* também possui uma lista de LSPs que foram agregados através das políticas de *grooming* apresentadas na Seção 5.2. O atributo *eRoute* armazena a rota física do caminho de luz. Os atributos *ingressNode* e *egressNode* armazenam, respectivamente, o nó de ingresso e egresso da rede óptica. A classe *LSP* possui um atributo que representa a banda solicitada (*req*) e outro atributo que representa a banda máxima que poderá ser solicitada pelo cliente que possui este LSP (*max*). Este último atributo deve estar em conformidade como o atributo (*maxBW*) da classe *SLA* que define as características de tráfego que um determinado cliente pode solicitar. Este contrato é previamente estabelecido entre os domínios usando mecanismos *off-line* (por exemplo, telefone, e-mail, etc.). Os atributos *ingressNode* e *egressNode* da classe *LSP* armazenam, respectivamente, o nó de ingresso e egresso da rede cliente. Os atributos *levelOfProtection* e *trafficType* da classe *LSP* também devem estar em conformidade com os atributos da classe *SLA*. Estas conformidades são verificadas durante o processo de admissão e são realizadas pelo AC e pelo PM.

7.1.2 Testes e Avaliação

Os resultados apresentados nesta seção são baseados nos artigos publicados em [Verdi et al., 2006d], [Verdi et al., 2006b] e [Verdi et al., 2006e]. Os testes foram feitos a fim de obtermos uma estimativa de consumo de banda e tempo para estabelecimento de conexões entre

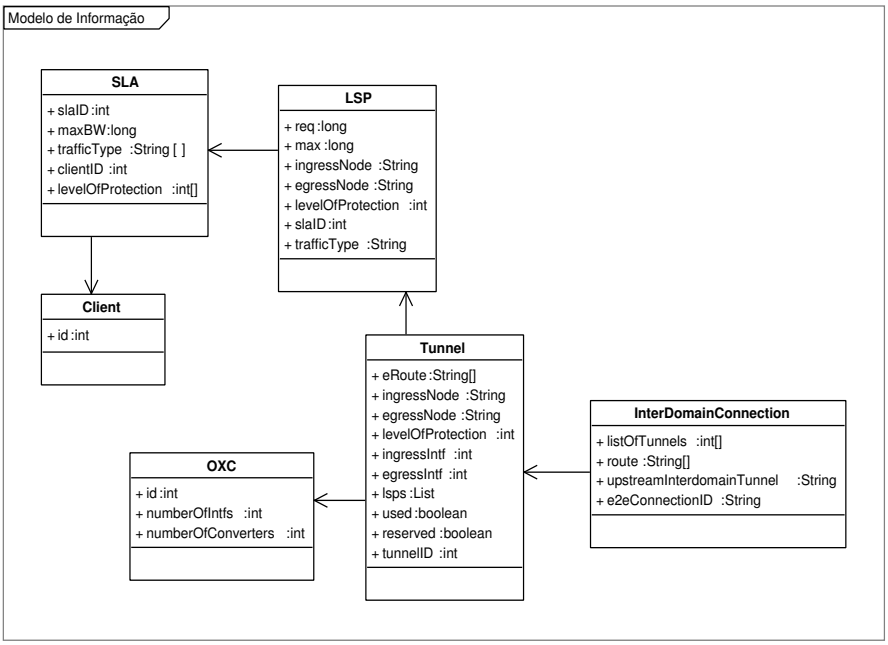


Fig. 7.7: Algumas classes do modelo de informação e suas relações.

domínios. O objetivo principal é avaliar o uso de *Web Services* para este tipo de cenário. O maior impacto no tamanho de uma mensagem SOAP é causado pela quantidade de informações que são transferidas de um lado para outro. As mensagens SOAP são descritas em XML e, portanto, sensíveis ao texto. Isto significa que para cada novo caractere inserido na mensagem, um *byte* a mais precisa ser enviado [Verdi et al., 2006b]. Uma maneira de diminuir este impacto seria reduzir o tamanho dos nomes dos elementos XML usando códigos ou algum outro mecanismo. Porém, ao usarmos este tipo de solução vamos contra o principio básico do XML que é representar os elementos de uma forma padrão e textual que facilite a leitura dos mesmos. Outra solução que tem sido cogitada é o uso de XML binário, um formato compacto que reduz o tamanho da mensagem [Geer, 2005]. Porém, este formato aumenta o tempo de processamento local, tanto no lado cliente como no lado servidor. No protótipo implementado, usamos o XML em sua forma simples sem nenhum tipo de otimização.

Iremos comparar também dois estilos de comunicação usados pela tecnologia *Web Services*. O primeiro estilo representa uma invocação remota RPC. O segundo estilo é conhecido como estilo *document*. O estilo RPC é caracterizado por invocar métodos em serviços remotos enquanto que o modelo *document* é caracterizado por usar o estilo de fraco-acoplamento, onde os parâmetros são validados através de um esquema XML no lado servidor. A principal diferença entre os dois está relacionada ao tamanho da mensagem SOAP. Mensagens RPC são maiores do que as mensagens do estilo *document*.

O tempo final depende do tempo de comunicação entre cada par de módulos e do tempo de

processamento em cada módulo. O tempo de comunicação considera o tempo para *marshalling* e *unmarshalling*, "parseamento" da mensagem SOAP, invocação HTTP, propagação dos dados na rede e entrega da requisição/resposta para o serviço. O tempo de comunicação é diretamente influenciado pelo tamanho das mensagens SOAP. A Tabela I apresenta os tempos de comunicação entre os serviços e os tamanhos das mensagens SOAP considerando o estilo RPC de invocação[2]. Como estamos interessados em analisar a tecnologia *Web Services*, os tempos mostrados não consideram o processamento interno de cada módulo mas apenas o tempo de comunicação. Os valores mostrados representam a média de 100 requisições.

Tab. I: Tempo médio para cada interação SOAP e tamanho da mensagem (estilo RPC).

interação	tempo (em ms)	tamanho da mensagem SOAP (em bytes)
cliente para E2ECS	10	1156 (req) + 650 (resp) Total = 1806
AC para PCE	9,5	720 (req) + 593 (resp) Total = 1313
AC para E2ENS	9,4	1486 (req) + 581 (resp) Total = 2067
E2ENS para E2ENS (RP)	13,95	1156 (req) + 564 (resp) Total = 1720
E2ENS para E2ENS (CP)	10	520 (req) + 557 (resp) Total = 1077
Total: tempo e tamanho	52,85	7983

A invocação do cliente para o E2ECS representa um pedido para estabelecimento de uma conexão entre domínios. Os parâmetros para esta requisição são o identificador do cliente, o nível de proteção desejado, a banda requerida, os nós de origem e destino, e o tipo de tráfego (HP ou LP).

A invocação do AC para o PCE é realizada a fim de obter uma rota entre os nós informados pelo cliente. Os parâmetros de entrada são apenas os nós de origem e destino e como resposta obtém-se uma rota de caminho mais curto conectando os dois nós. Os parâmetros enviados do AC para o E2ENS e do E2ENS para o E2ENS (*reservation phase-RP*) são basicamente os mesmos (banda, tipo de tráfego, nível de proteção, identificador da conexão e a rota). O tamanho desta última invocação é 1720 bytes enquanto que a invocação do AC para o E2ENS é de 2067 bytes. Porém, como a invocação do AC para o E2ENS é local pois é realizada dentro mesmo *Web Services container*, o

[2]A Figura 7.2 pode ser usada para acompanhar as interações apresentadas na tabela.

tempo é menor. Finalmente, a invocação do E2ENS para o E2ENS (*commit phase-CP*) possui dois parâmetros: o identificador da conexão e o identificador do túnel que conecta cada par de domínios.

O tempo total para estabelecimento de uma conexão entre domínios considerando apenas as comunicações SOAP é de 52,85ms e ocupa uma banda de aproximadamente 8KB. Calculamos o tempo para as invocações RMI e temos um tempo final de 55,85ms. Nos próximos gráficos apresentaremos o tempo total considerando o momento em que o cliente realiza o pedido por uma conexão entre domínios até o recebimento da confirmação de tal pedido.

A expressão que representa a quantidade de mensagens SOAP necessárias para estabelecer uma conexão entre domínios é a seguinte: *2 + 2*N*, onde 2 corresponde às mensagens do AC para o PCE e do AC para o E2ENS. O termo *2*N* corresponde ao protocolo de negociação de duas fases negociando com *N* domínios. Não estamos considerando a invocação do cliente para o E2ECS pois nem sempre o cliente estará realizando uma invocação SOAP. Acreditamos que esta invocação é apenas uma requisição HTTP sem a mensagem SOAP. Se também considerarmos as respostas SOAP, temos o seguinte: *4 + 4*N*. Estes números representam o cenário de sucesso, ou seja, em cada domínio a reserva foi feita e confirmada. Em casos onde, por algum motivo, a reserva não foi possível em algum domínio (por exemplo, falta de recursos), a operação de *rollback* precisa ser realizada para liberar os recursos reservados nos domínios onde havia recursos disponíveis. Neste caso, a quantidade de mensagens SOAP é: (*4 + 4*N*) - (*2*qtd. de domínios que não conseguiram realizar a reserva*).

Como o objetivo consistiu em analisar o desempenho da implementação dos *Web Services*, sem levar em consideração o perfil das requisições, o fluxo de chamadas para os gráficos apresentados a seguir consiste em enviar uma requisição imediatamente após o retorno da requisição anterior.

A Figura 7.8 mostra o tempo médio necessário para o estabelecimento de conexões entre domínios. A topologia usada para os testes possui 5 domínios e é mostrada na Figura 7.11[3].

Para coletar os números da Figura 7.8, executamos 40000 requisições. Primeiramente, consideramos um cenário com 3 domínios, depois com 4 e finalmente usamos todos os 5 domínios da topologia. No cenário com 5 domínios, o tamanho médio da rota é de 3 domínios. Ou seja, a reserva é feita no domínio local e em mais dois domínios. Este número está de acordo com cenários reais da Internet. Alguns estudos [Pujol et al., 2005] constataram que o tamanho médio da rota no nível de domínios na Internet atual é de 3 a 4 domínios, ou seja, em média há de 3 a 4 domínios entre um nó fonte e um nó destino.

O objetivo é analisar o impacto em relação à quantidade de domínios envolvidos para estabelecer o serviço. No cenário da Figura 7.8, estamos considerando que apenas um domínio está requisitando o serviço de conexão entre domínios. Os números do gráfico foram plotados considerando a média de 500 pontos, ou seja, para cada 500 requisições calculamos a média e plotamos um ponto no gráfico

[3]Esta topologia também será usada para os testes do serviço de VPN.

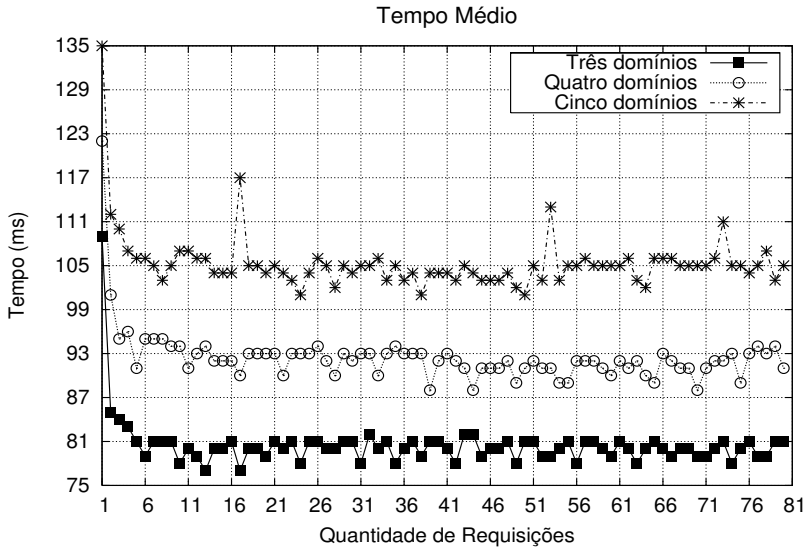


Fig. 7.8: Tempo médio para estabelecimento de conexões fim-a-fim entre domínios (um domínio requisitante).

(eixo X do gráfico). O tempo em milissegundos é mostrado no eixo Y.

Observamos que no cenário onde 3 domínios foram usados, o tempo médio para estabelecimento de uma conexão entre domínios é de aproximadamente 81ms. Com 4 domínios o tempo sobe para 93ms e com 5 domínios este tempo é de 105ms. Este aumento do tempo se deve ao crescimento no tamanho da rota uma vez que a quantidade de domínios envolvidos aumenta. Na medida em que o tamanho da rota aumenta, mais recursos precisam ser reservados e conseqüentemente o tempo final aumenta. Há um aumento de 25ms em relação ao cenário com 3 domínios e o cenário com 5 domínios. Outro ponto importante a ser analisado se refere ao fato de que o protocolo de negociação em duas fases usando o modelo em estrela não causa aumento no tempo final quando a quantidade de domínios cresce. Isto ocorre pois a negociação com cada domínio é feita disparando-se *threads* em paralelo, uma para cada domínio. Constatamos apenas alguns picos no cenário com 5 domínios. Estes picos ocorrem devido ao fato de que, apesar do protocolo de negociação não afetar o tempo final, a gerência das *threads* e a quantidade de mensagens trocadas na rede causam um certo *overhead* em determinados momentos. Este comportamento aparece na forma de picos no gráfico. Além disso, os maiores tempos no início da execução dos testes se devem ao *overhead* inicial para carregamento das classes na memória e à abertura das conexões TCP (*three-way hand shake*). Por fim, é importante lembrar que deste tempo total, aproximadamente 55ms correspondem ao tempo de comunicação das mensagens SOAP (ver Tabela I) e o restante corresponde ao processamento interno em cada módulo.

A Figura 7.9 analisa o impacto em um domínio em cenários onde vários domínios realizam requisições simultaneamente. O objetivo é avaliar o comportamento do protótipo em situações de extrema demanda pelos serviços. Para este cenário, todos os testes foram feitos envolvendo

a topologia com os 5 domínios. A quantidade de domínios realizando requisições varia de 1 a 5 sendo que cada domínio invoca 10000 requisições. Da mesma forma que o gráfico anterior, o eixo X representa a quantidade de requisições e o eixo Y o tempo em milissegundos.

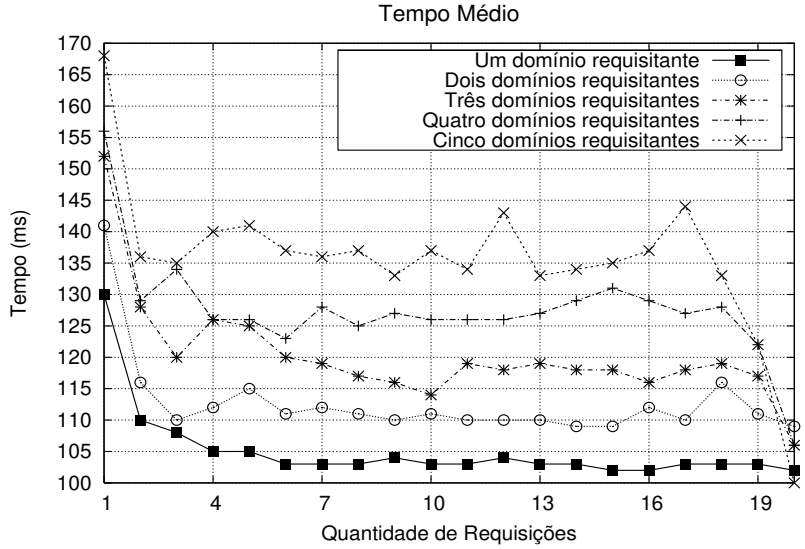


Fig. 7.9: Tempo médio para estabelecimento de conexões fim-a-fim entre domínios (vários domínios requisitantes).

Os números da figura mostram que, quando apenas um domínio está realizando requisições (cenário da Figura 7.8) o tempo permanece em 105ms. Na medida em que a quantidade de domínios que solicitam o serviço aumenta, o tempo cresce levemente. Com 5 domínios realizando requisições, temos um tempo médio de 135ms. Ao observarmos o tempo em relação ao primeiro cenário e o último, temos um aumento de tempo de aproximadamente 30ms. A redução no tempo do lado direito do gráfico se deve ao fato de que, na medida em que alguns domínios requisitantes terminam suas invocações, os outros domínios conseguem processar mais rápido e, portanto, diminuem seus tempos médios de processamento para cada invocação.

Para finalizar as análises com o estabelecimento de conexões entre domínios, realizamos alguns testes num cluster de 16 máquinas. Usamos 8 destas máquinas para representar os domínios ópticos. A topologia usada para estes testes encontra-se na Figura E.1 do Apêndice E. Além de aumentarmos a quantidade de domínios envolvidos nos testes e criarmos topologias virtuais mais complexas, o objetivo foi também comparar o estilo RPC com o estilo *document*. A Tabela II mostra a diferença no tamanho das mensagens SOAP entre os dois estilos.

Observe como exemplo que a mensagem de requisição (Req.) do E2ENS para o E2ENS (*Reservation Phase* (RP) possui tamanho de 491 *bytes* quando usamos o estilo *document*. Porém, a mesma mensagem no estilo RPC possui tamanho de 1502 *bytes*, ou seja, três vezes maior.

A Figura 7.10 mostra os tempos coletados para os dois estilos de comunicação. Neste cenário,

Tab. II: Tamanhos das mensagens SOAP (em bytes): estilo RPC X *document*.

Interação	*document*		RPC	
	Req.	Resp.	Req.	Resp.
cliente para E2ECS	428	433	1152	598
AC para PCE	351	400	597	588
AC para E2ENS	542	411	1640	573
E2ENS para E2ENS (RP)	491	419	1502	553
E2ENS para E2ENS (CP)	353	387	518	549

todos os 8 domínios foram envolvidos e a quantidade de domínios fazendo requisições varia de 1 a 4 (eixo X do gráfico). O eixo Y representa o tempo em milissegundos.

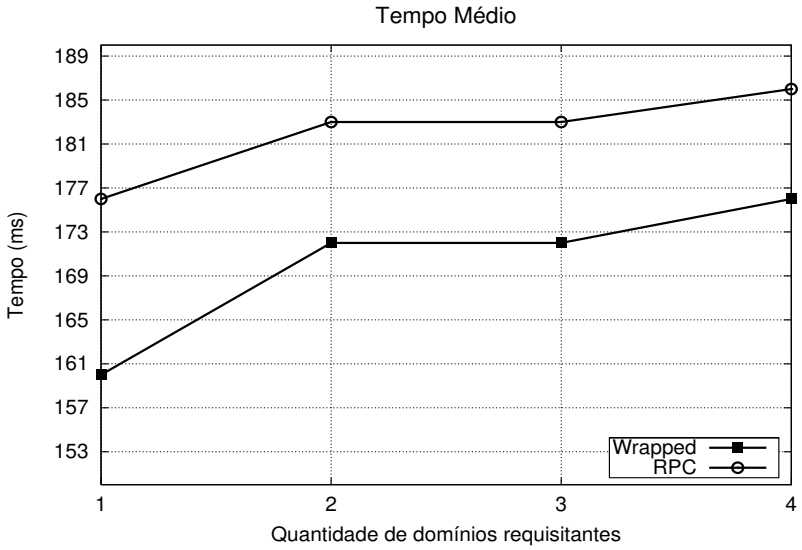


Fig. 7.10: Comparação entre os estilos RPC e *document*.

Como não poderia ser diferente, o modelo *document* apresenta um tempo menor do que o tempo do estilo RPC. Uma vez que o tamanho das mensagens do estilo *document* é menor, o tempo de comunicação entre os módulos é também reduzido. No cenário com 4 domínios realizando requisições, o tempo médio para estabelecimento de uma conexão no estilo RPC é de aproximadamente 186ms, enquanto que com o estilo *document* o tempo médio é reduzido para aproximadamente 176ms.

A distribuição de topologias virtuais também foi analisada. Embora o foco principal tenha sido realizar a análise para o tempo de estabelecimento das conexões entre domínios, um entendimento do desempenho em relação à distribuição de topologias virtuais se faz necessário. Não é nossa intenção

analisar e comparar o mecanismo de distribuição de topologias virtuais com o BGP ou qualquer outro protocolo de roteamento entre domínios. O BGP possui como princípio básico o roteamento global na Internet e para tal, o tamanho de sua mensagem de *UPDATE* possui 19 *bytes* de cabeçalho e o tamanho restante da mensagem pode variar dependendo do tamanho da rota sendo anunciada. O tamanho máximo de uma mensagem BGP é de 4096 *bytes* [Rekhter and Li, 2006]. Ao usarmos XML para divulgação de topologias, obviamente o tamanho de tais mensagens aumenta consideravelmente. Nossa intenção é apenas apresentar algumas informações sobre o comportamento do mecanismo de distribuição de topologias virtuais no plano de serviços. Outros comentários serão feitos abaixo na seção de Discussão Final.

O tamanho da mensagem SOAP para distribuição das topologias virtuais depende do tamanho da topologia virtual em termos de quantidade de nós e enlaces presentes na topologia. Além disso, o tamanho também depende do formato dos identificadores dos nós na rede. Como explicado anteriormente, o XML é sensível ao texto e, com isso, o formato dos elementos XML reflete no tamanho da mensagem final. Os endereços IPs possuem diferentes tamanhos dependendo da classe a que eles pertencem, máscaras, etc. Por exemplo, o endereço IP 200.176.3.142 possui textualmente 13 *bytes* de tamanho. O endereço IP 10.1.1.2 possui 8 *bytes* de tamanho. Neste trabalho, usamos identificadores textuais (como pode ser visto na Figura 7.3) que possuem um tamanho médio (11 *bytes*) em relação aos possíveis tamanhos dos endereços IP.

Constatamos que para cada enlace virtual adicionado à topologia virtual há um aumento de 98 *bytes*, e para cada nó adicionado há um aumento de 63 *bytes*. Considerando *i* a quantidade de enlaces, *j* a quantidade de nós e *c* o tamanho do cabeçalho da mensagem SOAP mais outras informações fixas da topologia, temos que: *(i * 98) + (j * 63) + c* é igual ao tamanho da mensagem SOAP para distribuir uma topologia virtual com *i* enlaces e *j* nós. Considerando o modelo *Push* e *n* a quantidade de domínios, cada topologia é enviada para os *n-1* domínios. Temos então que: $\sum_{k=1}^{n}$ *((i * 98) + (j * 63) + c) * n-1* é igual a quantidade total de *bytes* para divulgação de topologias virtuais entre domínios em uma rodada. Como exemplo, a topologia da rede NSFNet possui 14 nós e 25 enlaces, resultando em um tamanho de 3332 *bytes*. Fizemos o teste para envio da topologia NSFNet e obtemos um total de 3674 *bytes*. Temos então que deste total, aproximadamente 300 *bytes* (considerando o modelo *document*) pertencem ao cabeçalho da mensagem SOAP que pode variar dependendo do estilo de comunicação usado (RPC ou *document*). O tempo médio de envio para esta topologia de um domínio para outro considerando 100 execuções foi de 18 milissegundos.

7.1.3 Discussão Final

Os tempos coletados servem como uma estimativa do impacto do uso dos *Web Services* para estabelecimento de conexões ópticas entre domínios. Obviamente, tais tempos podem variar em

função da quantidade de informações a ser negociada entre os domínios. Porém, as informações utilizadas no protótipo implementado podem ser consideradas suficientes para um cenário de redes ópticas real. Além disso, seria interessante analisar outras arquiteturas e protótipos a fim de comparar os tempos nos cenários considerados. Um dos grupos do projeto CANARIE desenvolveu uma proposta cujos mecanismos para estabelecimento de conexões entre domínios são similares aos apresentados nesta tese. A proposta possui um mecanismo para reserva de recursos em duas fases e usa *Web Services*. Os testes do protótipo da arquitetura [Truong et al., 2004] foram feitos considerando o estabelecimento de conexões que atravessam três domínios administrativos em um cenário real. A "crossconexão" nos dispositivos é feita utilizando comandos TL1. O tempo médio obtido para reserva de recursos em cada domínio foi de 5.83 segundos. Segundo os autores, a maior parte deste tempo pertence à consulta que precisa ser feita em um diretório de políticas e em uma base de recursos em cada domínio. Outro fator é o tempo de processamento da mensagem SOAP. A reserva em cada domínio é realizada de forma seqüencial. Desta forma, o tempo necessário para estabelecer um conexão entre *N* domínios é de *N * 5.83* segundos. No nosso modelo, a reserva em duas fases é realizada em paralelo em todos os domínios que compõem a rota fim-a-fim. Os tempos coletados pelo trabalho vinculado ao projeto CANARIE são, segundo os autores, aceitáveis uma vez que as conexões ópticas não são estabelecidas para serem usadas imediatamente.

Os números obtidos pelo trabalho mencionado acima envolvem um cenário real onde há um conjunto de fatores reais tais como a "crossconexão" nos dispositivos e o tempo de envio das mensagens SOAP na rede. Eles servem como um parâmetro de comparação com os números coletados neste trabalho. É importante comentar que nos testes realizados nesta tese, não estamos considerando o tempo de "crossconexão" nos OXCs de borda. Em [Margaria et al., 2005] comenta-se que estes tempos são de aproximadamente 60 ms por dispositivo, dependendo do cenário. Esta "crossconexão" é feita durante a segunda fase do protocolo de negociação e depende do tempo que cada OXC necessita para "crossconectar" dois comprimentos de onda. Além disso, os testes foram feitos em nosso laboratório com alta disponibilidade de banda. Considerando um cenário externo, tais tempos tendem a aumentar devido ao intrínseco gargalo e limitação de banda atualmente presentes nas conexões Internet. Portanto, os números obtidos nesta tese refletem, em sua grande parte, o tempo que é necessário para o processamento da mensagem SOAP.

Outros testes encontrados na literatura mostraram que o tempo necessário para estabelecimento de um caminho de luz usando GMPLS dentro de um domínio óptico é de aproximadamente 600 ms em uma rede com 4 nós ópticos [Margaria et al., 2005]. Os testes foram feitos utilizando o protocolo RSVP. Neste tempo estão incluídos os tempos de "crossconexão" nos OXCs de ingresso, de trânsito e de egresso, além dos tempos para envio das mensagens PATH e RESV.

Em relação aos estilos utilizados, constatamos que o estilo *document* se mostrou mais apropriado

não somente pelo tempo e tamanho de mensagens menores em relação ao estilo RPC, mas também por oferecer um modelo mais orientado a serviços através de um fraco acoplamento entre os mesmos. O estilo *document* vem substituindo o estilo RPC, modelo que foi inicialmente muito utilizado no desenvolvimento de *Web Services*. Além disso, com o crescimento da arquitetura SOA, o modelo *document* se torna uma alternativa que representa melhor os requisitos de tal arquitetura.

Por fim, uma análise da distribuição das topologias virtuais foi realizada. Acreditamos que os números obtidos são factíveis considerando os dois modelos para obtenção de topologias virtuais definidos nesta tese: modelo *Push* e modelo *Pull*. No modelo *Push*, condomínios de domínios são definidos e a divulgação das topologias virtuais ocorre apenas dentro destes condomínios. Entendemos que estes condomínios não tendem a ser grandes e, portanto, a divulgação de topologias virtuais, considerando os números obtidos, não deve enfrentar problemas relacionados com a escalabilidade. A divulgação de topologias entre condomínios ocorre abstraindo-se os detalhes internos de cada domínio fazendo com que cada domínio passe a ser visto como um nó no nível dos condomínios. Com este modelo, a quantidade de informações a ser divulgada não aumenta, mesmo que a divulgação de topologias virtuais seja feita entre condomínios. Porém, a divulgação de topologias virtuais envolvendo mais de um nível hierárquico necessita de mais estudos. No modelo *Pull*, as topologias virtuais não são divulgadas entre os domínios. Elas são obtidas apenas sob-demanda no momento em que um determinado domínio deseja estabelecer a conexão entre domínios. A quantidade de topologias virtuais a ser obtida depende da quantidade de rotas BGP e de políticas locais do domínio requisitante.

Entretanto, o principal ponto a ser destacado está relacionado ao fato de que as conexões ópticas, devido a sua grande capacidade de transmissão, tendem a ser solicitadas por clientes que oferecem acesso a outros clientes (usuários finais) fazendo com que os pedidos de conexões não sejam tão freqüentes. Exemplos de clientes neste caso são os provedores de acesso que utilizam a conexão óptica para agregação de vários fluxos de tráfego IP [Brunner et al., 2004]. Estas conexões ópticas tendem a permanecer estabelecidas por horas, dias e até meses. Desta forma, tem-se aceitado bem a estimativa de tempo para provisionamento de conexões entre domínios na faixa de segundos e até de dezenas de segundos [Truong et al., 2004, Yang et al., 2006]. Com isso, a freqüência com que a distribuição das topologias virtuais deve ocorrer também é reduzida pois o consumo de recursos segue a freqüência dos pedidos de conexões.

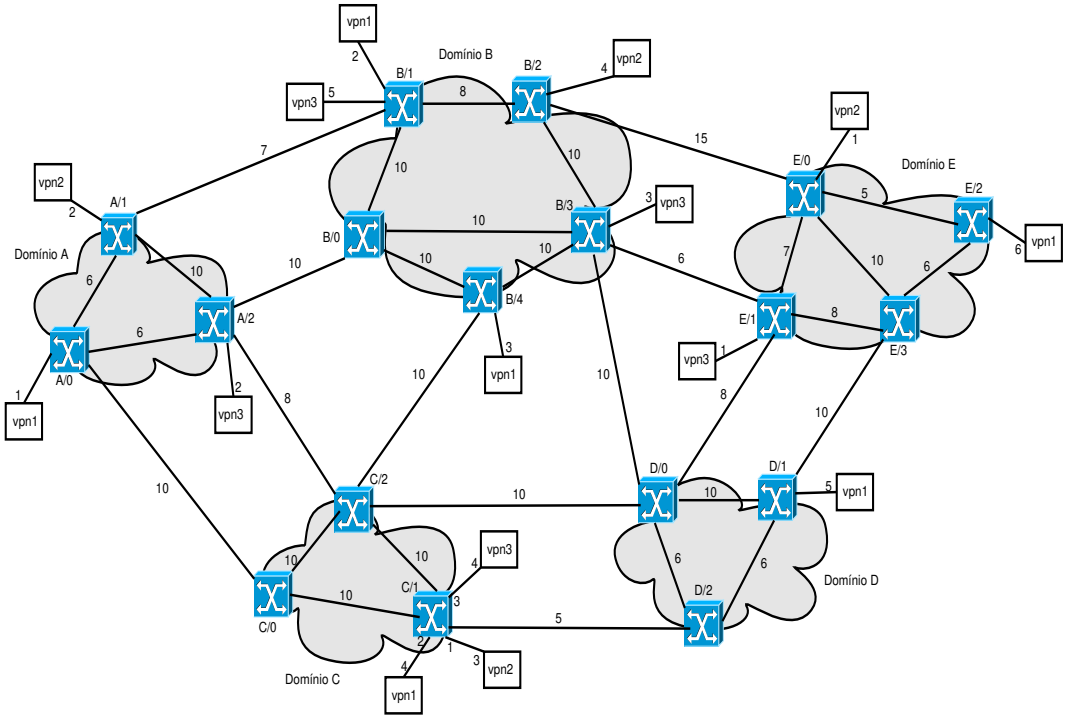


Fig. 7.11: Topologias virtuais usadas nos testes para cinco domínios.

7.2 Detalhando o Serviço de VPNs entre Domínios

7.2.1 Implementação

A implementação do serviço de VPN entre domínios foi facilitada uma vez que a maioria das funcionalidades necessárias para prover o serviço de VPN foram implementadas para o provisionamento de conexões entre domínios. Basicamente, as diferenças consistem em analisar a correlação de portas no momento em que os clientes solicitam o estabelecimento das VPNs. A correlação de portas é feita pelo *Membership Manager* (MM), responsável por analisar se os pares de portas informados pelos clientes possuem autorização para o estabelecimento de conexões entre eles.

A Figura 7.12 mostra a descrição em XML da PIT do serviço da VPN 2 no domínio C (ver Figura 7.11). Como pode ser observado na Figura 7.11, a VPN 2 possui portas distribuídas pelos domínios A, B, C e E. O elemento *domains* indica para quais domínios o serviço de divulgação (AS) deve divulgar a PIT (menos para o domínio local, neste caso o domínio C). O elemento *client* identifica unicamente o proprietário da VPN. O elemento *cpi* informa a porta do lado cliente e o elemento *ppi* informa a porta do lado da rede óptica. Este último está representado através da união do endereço do nó com uma porta. No exemplo da Figura 7.11, a CPI 3 da VPN 2 no domínio C está conectada ao nós C/1 na porta 1. Os endereços das portas nos nós ópticos são numerados de forma crescente. O formato destes endereços depende da tecnologia usada tanto na rede cliente como na rede óptica.

```
<?xml version=1.0>
<vpn>
      <domains>A,B,E</domains>
      <port>
            <client>vpn2</client>
            <cpi>3</cpi>
            <ppi>C/1.1</ppi>
       </port>
</vpn>
```

Fig. 7.12: PIT da VPN 2 no domínio C descrita em XML.

Os domínios A, B e E também possuem suas PITs locais que mapeiam as portas da VPN 2. Quando o AS as divulga, cada domínio terá uma PIT única com todos os pares de portas da VPN 2.

O diagrama de seqüência para estabelecimento de uma VPN entre domínios considerando o caso de sucesso é apresentado na Figura 7.13. Ao receber uma requisição para estabelecimento de uma VPN entre domínios, o *Trading Service* (TS) encaminha tal requisição para o AC. O AC por sua vez verifica se o cliente possui autorização para criar VPNs. Em caso positivo, o MM realiza a correlação de portas a fim de verificar se as portas informadas pelo cliente pertencem a faixa de portas reservadas para a VPN em questão. Se as portas estiverem corretas, o AC dispara o estabelecimento das conexões individuais que conectam os pares de portas PPIs. Neste momento, o estabelecimento consiste em realizar os passos para estabelecimento de conexões entre domínios. O conjunto destas conexões forma a VPN. Se os pares de nós origem e destino não pertencem ao domínio local (domínio onde a requisição foi realizada), o estabelecimento das conexões ocorre normalmente. A diferença é que não haverá reserva de recursos no domínio óptico local.

O pseudo-código para estabelecimento de uma VPN é apresentado na Figura 7.14. O pseudo-código entre a linha 5 e linha 10 mostra o tratamento a ser realizado quando não é possível estabelecer a conexão entre as duas portas. Neste caso, todas as conexões que pertencem à VPN que já foram estabelecidas precisam ser removidas. Uma melhoria neste algoritmo seria implementar o protocolo de duas fases no nível das VPNs. Assim, primeiramente as reservas seriam feitas para cada conexão. Se todas as reservas ocorreram, a confirmação é realizada na segunda fase. Com isso, evitaríamos os passos necessários para remover as conexões previamente estabelecidas.

7.2.2 Testes e Avaliação

Os testes foram realizados utilizando a topologia apresentada na Figura 7.11. Definimos três serviços de VPNs cujas portas estão distribuídas entre os cinco domínios da Figura 7.11. O serviço da VPN 1 possui 6 portas, o serviço da VPN 2 possui 4 portas e o serviço da VPN 3 possui 5 portas.

As Figuras 7.15 e 7.16 mostram os tempos médios coletados para o estabelecimento de VPNs

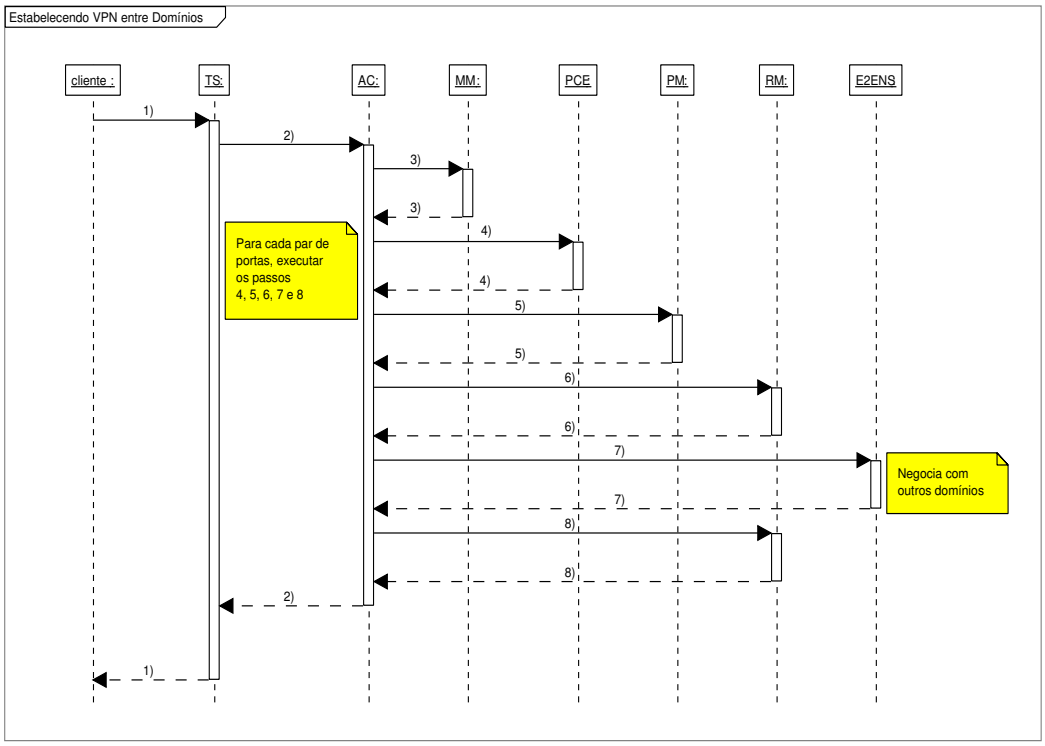


Fig. 7.13: Diagrama de seqüência para estabelecimento de VPNs entre domínios.

entre domínios. A Figura 7.15 apresenta o tempo médio quando apenas um cliente (um domínio requisitante) realiza requisições para estabelecimento de VPNs. A Figura 7.16 mostra os valores que refletem o cenário com vários domínios fazendo requisições ao serviço.

Para coletar os dados das figuras, cada cliente requisitou o estabelecimento de 1000 VPNs. Os pontos nos gráficos aparecem como a média de 50 requisições. Na Figura 7.15 apenas o cliente do serviço da VPN 1 realiza requisições solicitando o estabelecimento de VPNs entre as portas espalhadas pelos domínios ópticos. O cliente solicita o estabelecimento de VPNs com 4, 5 e 6 portas. O eixo X representa a quantidade de requisições e o eixo Y o tempo consumido em milissegundos.

O tempo médio para estabelecer uma VPN entre domínios com 4 portas é de 330ms, enquanto que para estabelecer uma VPN entre domínios com 6 portas o tempo médio é de 800ms. Quanto maior o número de portas a serem conectadas, maior o tempo para estabelecimento da VPN. Estamos considerando que cada porta da VPN é conectada com as outras *n-1* portas, caracterizando uma rede totalmente conectada (*full meshed*). Como exemplo, para estabelecer uma VPN com 4 portas, são necessárias 6 conexões. Uma VPN com 6 portas são necessárias 15 conexões. Temos que a quantidade de conexões é obtida por: *i*(i-1)/2*, onde *i* é a quantidade de portas informadas pelo cliente. O número de mensagens SOAP para estabelecer uma VPN entre domínios é: *(i*(i-1)/2) * (4+4*N)*. Obviamente estamos considerando o pior caso (*full meshed*), porém as VPNs nem sempre

```
1  para cada par de portas (p1,p2)
2          estabelece_conexao(p1,p2)
3          se o estabelecimento da conexão foi possível
4              guarda o identificador da conexão em um vetor
5          senão
6              percorre o vetor de identificadores das conexões
7              já estabelecidas e remove todas as conexões
8              (invoca função remove_conexao(id))
9              retorna falso
10         fim senão
11 fim para
12 guarda no RM as informações sobre a VPN estabelecida
13 retorna verdadeiro
```

Fig. 7.14: Pseudo-código para estabelecimento de uma VPN entre domínios.

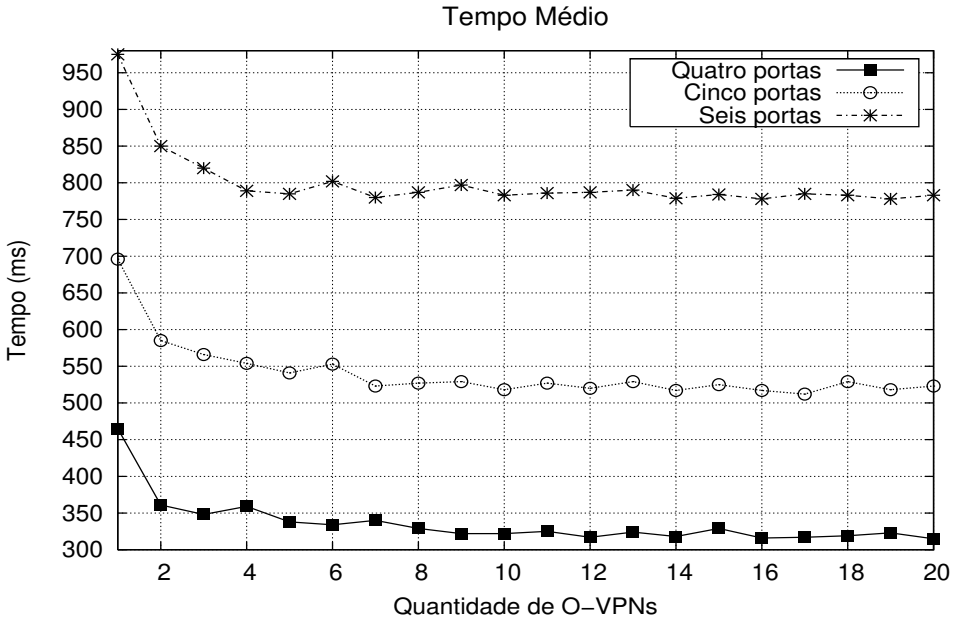


Fig. 7.15: Tempos para estabelecimento de VPNs entre domínios (um domínio requisitante).

serão conectadas desta forma.

A Figura 7.16 mostra o cenário pelo qual os três clientes dos serviços de VPN (VPN 1, VPN 2 e VPN 3) solicitam o estabelecimento de VPNs simultaneamente. Todos estão solicitando VPNs com 4 portas. O cliente do serviço da VPN 1 realiza requisições a partir do domínio B. O cliente da VPN 2 realiza requisições a partir do domínio E e o cliente do serviço da VPN 3 realiza requisições a partir do domínio C. Observe que os tempos entre os diferentes cenários praticamente não se altera na medida em que mais domínios fazem requisições simultaneamente. As diferenças de consumo de tempo entre o cenário com um domínio requisitante e três domínios requisitantes é praticamente desprezível.

Os tempos mostrados nos gráficos se referem à reserva e negociação dos recursos para as VPNs

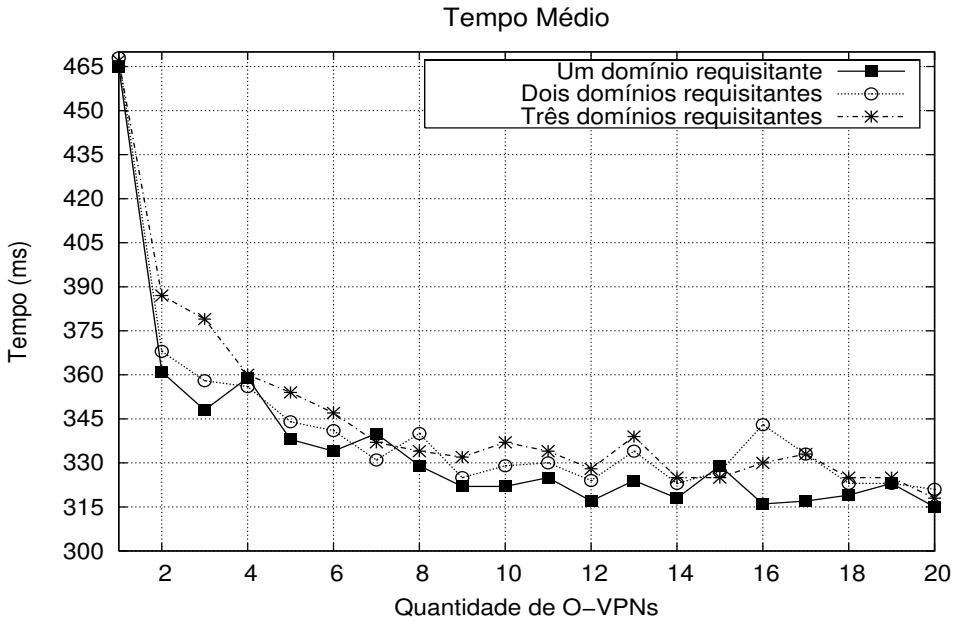


Fig. 7.16: Tempos para estabelecimento de VPNs entre domínios (vários domínios requisitantes).

entre domínios. As principais tarefas para o provisionamento de uma VPN entre domínios ocorrem nesta fase. Porém, a ativação da VPN ocorre no momento em que ela realmente for usada (poderá ser logo após a reserva). A ativação é feita através da invocação do serviço OVPNS e o processo de cobrança (*billing*) em cada domínio deve ser iniciado. O processo de ativação é simples sendo suficiente o envio de uma mensagem para os domínios na qual a VPN está estabelecida. O processo de tarifação é realizado em cada domínio seguindo contratos pré-estabelecidos.

A análise de desempenho para a distribuição de informações de correlação de portas das VPNs foi feita de maneira similar à análise para distribuição de topologias virtuais. Localmente em um domínio, para cada porta adicionada à VPN, há um aumento de 120 *bytes*. Como exemplo, uma VPN com 2 portas locais gera uma mensagem SOAP com tamanho de 670 *bytes*. Com 3 portas, o tamanho aumenta para 790 *bytes*. Considerando *np* o número de portas locais em um domínio e *c* o tamanho do cabeçalho da mensagem SOAP mais outras informações fixas da PIT, temos que: (*np * 120) + c* é igual ao tamanho da mensagem SOAP para enviar informações de correlação de portas de uma VPN em um domínio. Para cada VPN em cada domínio, tais informações precisam ser enviadas para os outros domínios que possuam portas na mesma VPN. Considerando *ndVPN* a quantidade de domínios que a VPN possui portas, temos então que: $\sum_{k=1}^{ndVPN}$ (*((np * 120) + c) * ndVPN-1*) é igual a quantidade de *bytes* para divulgação de informações de correlação de portas para uma VPN que possui portas em *ndVPN* domínios.

7.2.3 Discussão Final

De maneira similar ao estabelecimento de conexões entre domínios, a análise em relação à freqüência de pedidos por VPNs e distribuição de informações sobre correlação de portas também é válida para o estabelecimento de VPNs. Até onde sabemos, não há nenhuma arquitetura que realize o provisionamento de VPNs entre domínios de forma automática. Como já mencionado no Capítulo 2, um *draft* do IETF propõe mecanismos similares aos apresentados aqui para distribuição de informações de membros. Acreditamos que as idéias apresentadas nesta tese e no *draft* mencionado possam servir como um ponto de partida para futuras discussões em relação ao provisionamento de VPNs entre domínios.

7.3 Tarifação entre Diferentes Domínios Administrativos

O processo de tarifação (*billing*) entre domínios administrativos diferentes é, sem dúvida, um dos obstáculos ainda por ser resolvido neste processo de provisionamento automático de serviços entre domínios. Nesta tese, não adotamos nenhum mecanismo de cobrança entre domínios. Porém, nesta pequena seção iremos fazer alguns comentários gerais sobre possíveis soluções que acreditamos serem praticáveis em cenários reais.

O modelo de tarifação de serviços em redes ópticas deve considerar aspectos específicos encontrados apenas em tais redes. O trabalho citado em [French and Pendarakis, 2004] menciona alguns aspectos que o modelo deve levar em conta para tarifação em redes ópticas. Deve-se considerar que certos componentes encontrados nas redes ópticas não estão presentes nas redes IP tradicionais. Alguns exemplos incluem as portas *add/drop* e suas capacidades de transmissão (2.5Gbp/s ou 10Gbp/s) e o tipo de proteção nos caminhos de luz (1:1, 1:N, 1+1 e sem proteção). Outros fatores já conhecidos tais como o tempo de uso do serviço e o grau de controle a ser oferecido aos clientes e a outros domínios também são considerados. A soma de todos estes atributos e suas diferentes combinações gera um custo para cada conexão estabelecida. No caso das VPNs entre domínios, a soma dos custos de todas as conexões que formam a VPN gera o custo final.

O modelo tradicional de tarifação na Internet atual consiste em: (1) pagar por serviços adquiridos em outros domínios e/ou (2) fazer acordos de parcerias onde através da troca de serviços mútuos os domínios aceitam prover serviços em troca de outros sem tarifação (modelo *peering*). Se o primeiro modelo for usado na proposta desta tese, consideramos que durante a primeira fase do protocolo de negociação, o domínio invocado deveria retornar em sua resposta o valor a ser cobrado pelos recursos sendo oferecidos. O domínio requisitante poderia avaliar este valor aceitando ou recusando a oferta. Caso aceite, a confirmação é realizada na segunda fase do protocolo de negociação. Caso não aceite, a segunda fase do protocolo poderia ainda realizar uma contra-oferta ao domínio invocado.

Se o domínio invocado aceita a proposta, a reserva é confirmada e a negociação é concluída. Caso contrário, o domínio invocado desfaz a pré-reserva, recusa a contra-oferta e a negociação é concluída. Observe que este modelo mantém o protocolo de duas fases e não aumenta a quantidade de mensagens SOAP necessárias para negociar e reservar os recursos.

Se o segundo mecanismo for usado, um pré-contrato entre os domínios deveria existir a fim de regular o que pode ser solicitado e por quem. Durante a negociação, seria apenas necessário conferir os parâmetros da requisição a fim de analisar se os valores estão dentro da faixa acordada no contrato e disparar o processo de tarifação em cada domínio.

Outros modelos poderiam consistir na criação de uma entidade autenticada pela qual os domínios administrativos poderiam divulgar seus serviços com as características específicas de cada serviço. Nesta divulgação, os valores a serem cobrados pelos serviços também poderiam ser divulgados. Este mecanismo é semelhante ao modelo menos restrito apresentado na Seção 4 onde todos os atributos de QoS, incluindo o preço, são divulgados na topologia virtual. Porém, cabe lembrar que muitos domínios não estão dispostos a divulgar tal nível de informações para outros domínios.

Por fim, alguns estudos têm considerado, com alguma adaptação, o uso de protocolos utilizados em outros cenários para a negociação dinâmica de serviços de próxima geração. Um trabalho recente, citado em [Sarangan and Chen, 2006], realizou uma coletânea e apresenta algumas possíveis soluções de protocolos que poderiam ser usados para negociação de contratos e preços. Porém, o trabalho conclui afirmando que nenhum dos protocolos foi realmente testado em cenários reais e que ainda dependemos de organizações tais como IETF para a definição padronizada dos protocolos. Entretanto, acreditamos que a tecnologia *Web Services* pode ser usada também para este tipo de cenário, facilitando a forma como os mecanismos de tarifação são desenvolvidos. Estudos mais detalhados sobre mecanismos de tarifação são deixados como trabalhos futuros para esta tese.

7.4 Comparação entre os Modelos *Push* e *Pull*

Esta seção tem como objetivo analisar o modelo *Pull* e sua integração com um cenário real usando BGP. Tal seção é baseada no artigo referenciado em [Verdi et al., 2006c]. Durante toda a tese, comentamos que o modelo *Pull* pode ser integrado mais facilmente a um cenário real da Internet. Enquanto que o modelo *Push* considera condomínios de domínios e a divisão lógica de tais condomínios precisa ser feita seguindo algum critério, o modelo *Pull* depende apenas da definição de alguns serviços e protocolos para que eles interajam.

Ao desenvolvermos a arquitetura, percebemos que a mesma poderia ser usada não somente em redes ópticas mas também para oferecer uma camada de serviços nas redes IP tradicionais. Tal camada de serviços teria como objetivo prover um conjunto de informações sobre QoS em direção a

determinados destinos na rede. Obviamente o objetivo não é eliminar o roteamento BGP, mas sim, prover um conjunto de funcionalidades na camada de serviços de forma a oferecer mais informações para seleção de rotas considerando aspectos de QoS. O conceito de topologia virtual poderia ser usado também neste cenário. Ao invés dele representar os caminhos de luz estabelecidos em um domínio óptico, as topologias virtuais representariam a QoS que um determinado domínio poderia oferecer para outros domínios ou clientes. Neste caso, o modelo *Pull* é mais apropriado uma vez que somente as topologias virtuais de alguns domínios serão obtidas para análise de QoS.

O modelo *Pull* foi então adaptado para atuar de forma complementar ao BGP fazendo com que as rotas distribuídas por tal protocolo possam ser utilizadas pelo serviço de divulgação (AS). Esta tentativa de prover informações de QoS para cálculos de rotas nas redes IP tradicionais não é uma idéia nova. Oferecer QoS aos fluxos que precisam atravessar vários domínios administrativos diferentes é ainda uma desafio. O uso das topologias virtuais e a integração com o BGP é uma tentativa em direção ao fornecimento de QoS aos fluxos IP entre domínios. Enquanto o BGP divulga apenas informações sobre alcançabilidade, as topologias virtuais oferecem aos domínios um nível de informações maior em relação à QoS em direção a determinados prefixos de rede. A Figura 7.17 ilustra este conceito.

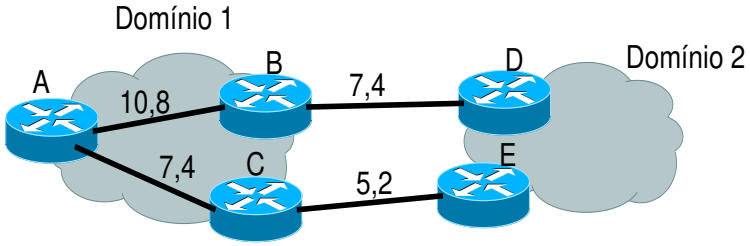


Fig. 7.17: Topologia virtual em redes IP.

A topologia virtual nas redes IP é usada da mesma forma de como ela é usada nas redes ópticas. O nível de informações a ser divulgado com as topologias virtuais pode variar do mais restrito ao menos restrito. Na Figura 7.17, as tuplas podem significar diferentes atributos de QoS. Por exemplo, o enlace virtual A-B possui valores 10 e 8. O valor 10 poderia ser a banda disponível no enlace e 8 a latência. A obtenção destes valores pode ser feita através de *probes* usando ferramentas tais como *ping* e *traceroute* como feito em [Li and Mohapatra, 2004]. Cada domínio é responsável por implementar a QoS informada nos enlaces virtuais através de tecnologias e mecanismos de engenharia de tráfego locais. Tipicamente, os domínios poderiam usar DiffServ ou MPLS para garantir a QoS divulgada. Técnicas de encapsulamento e tunelamento também podem ser utilizadas [Xu and Rexford, 2006].

Quando um determinado domínio precisa enviar um fluxo de pacotes em direção a um prefixo de rede, o domínio requisitante primeiramente obtém as rotas BGP locais a fim de acionar o serviço de divulgação (AS) para que as topologias virtuais dos domínios que pertencem às rotas BGP sejam

obtidas. Se, após obter tais topologias virtuais e negociar com os domínios, o domínio requisitante não consegue uma rota que satisfaça a QoS desejada, tal domínio pode subir um nível na hierarquia da Internet e obter as rotas BGP não divulgadas. Com as novas rotas BGP, o AS do domínio requisitante realiza os passos anteriores a fim de obter as topologias virtuais dos novos domínios.

Como mencionado no Capítulo 4, o modelo *Pull* faz uso da hierarquia da Internet para obtenção de novas rotas em direção a um determinado prefixo de rede. Ao subir um nível na hierarquia, o AS obtém rotas BGP não divulgadas pelos seus provedores. O cenário de testes para demonstrar a integração do modelo *Pull* com o BGP consistiu na criação de uma rede BGP real com a topologia apresentada na Figura 7.18. O objetivo principal foi analisar esta integração da camada de serviços com o protocolo BGP e comparar os tempos de estabelecimento de conexões[4] entre domínios usando os modelos *Push* e *Pull*.

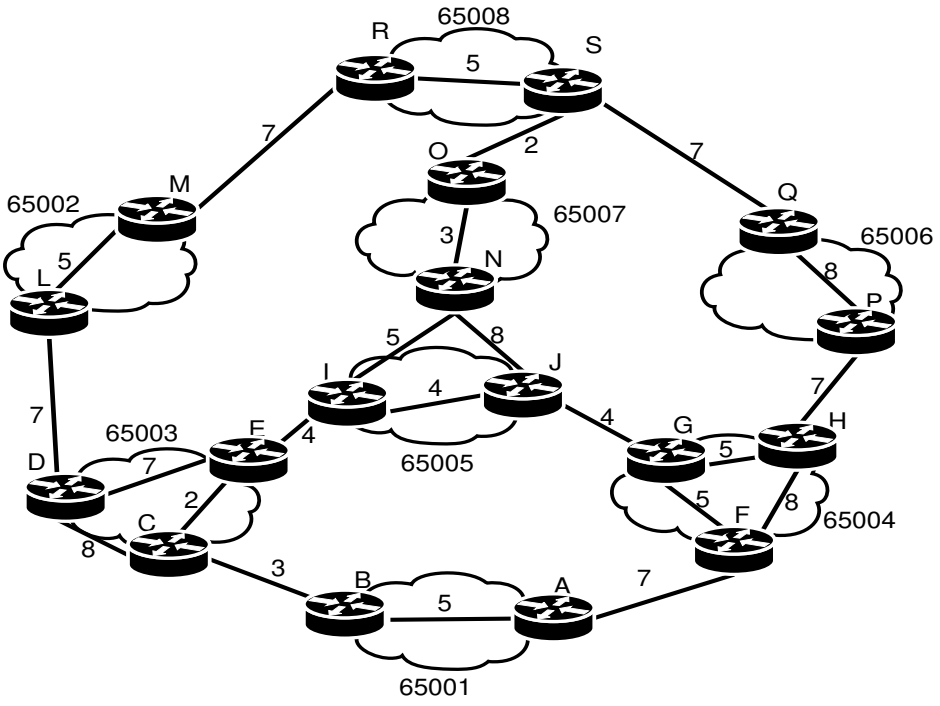


Fig. 7.18: Topologia usada para testar a integração da camada de serviços com o BGP.

O cenário é formado por 8 *Autonomous Systems* (ASes) sendo que o AS 65001 deseja enviar um fluxo com certas garantias de QoS para o AS 65008. Cada enlace virtual possui um custo abstrato que representa a QoS do enlace. Primeiramente, o AS 65001 obtém as rotas BGP locais em direção ao AS 65008. Duas rotas possíveis são identificadas: a rota através dos ASes 65003, 65002 e 65008 e a rota através dos ASes 65004, 65006 e 65008. Com isso, o serviço de divulgação (*Advertising Service*) do domínio requisitante (65001) na camada de serviços invoca o serviço de divulgação dos

[4]O termo conexão neste contexto representa simplesmente um fluxo IP, similar a um LSP.

outros domínios das duas rotas BGP a fim de obter as topologias virtuais em direção ao prefixo de rede informado. Cada domínio invocado retorna apenas as topologias virtuais em direção ao próximo domínio informado na rota BGP. Por exemplo, o domínio 65004 retorna para o domínio 65001 apenas a topologia virtual formada pelo enlace virtual F-H e H-P. Este mecanismo permite que cada domínio divulgue apenas o necessário mantendo um controle sobre quais rotas podem ser divulgadas.

Após a coleta destas topologias virtuais, o domínio 65001 escolhe a rota de menor custo e inicia o processo de negociação com os domínios a fim de verificar se os atributos de QoS desejados podem ser atendidos pelos domínios da rota escolhida. Caso a rota escolhida não satisfaça os requisitos de QoS, a outra rota pode ser usada. Note que se o nível menos restrito de divulgação de informações estivesse sendo usado, o cálculo da rota já poderia levar em conta os atributos de QoS informados, e a rota encontrada, caso haja uma, seria a melhor rota em relação ao(s) atributo(s) de QoS desejado(s).

Se nenhuma rota obtida localmente consegue atender os requisitos de QoS desejados, o domínio requisitante (65001) sobe um nível na hierarquia e obtém as rotas não divulgadas pelo BGP. No nosso cenário, o domínio 65001 através do serviço de divulgação, obtém dos domínios 65004 e 65003 as rotas BGP não divulgadas. A rota 65003, 65005, 65007 e 65008 e a rota 65004, 65005, 65007 e 65008 são então obtidas pelo domínio 65001. Com isso, o serviço de divulgação (*Advertising Service*) obtém as topologias virtuais destes domínios que compõem tais rotas e novamente o processo se repete como anteriormente a fim de encontrar uma rota que atenda aos requisitos de QoS. Caso não haja nenhuma rota que satisfaça a tais requisitos, o fluxo de pacotes IP pode ainda ser enviado através do mecanismo de melhor esforço.

A quantidade de rotas a ser obtida ao subirmos um nível na hierarquia é razoavelmente grande [Subramanian et al., 2002]. Devido a isso, os provedores podem limitar a quantidade de rotas a ser devolvida para o domínio requisitante. Além disso, o próprio domínio requisitante pode selecionar quais rotas são mais apropriadas para obtenção das topologias virtuais com base em políticas locais ou algum critério administrativo. Uma outra possível otimização seria obter, durante a primeira fase de busca das topologias virtuais (usando apenas as rotas BGP locais), as rotas BGP dos provedores não divulgadas pelo BGP. Assim, caso as topologias virtuais obtidas usando as rotas locais não atendam aos requisitos de QoS, as outras rotas BGP já estarão disponíveis fazendo com que o domínio local possa então obter as topologias virtuais correspondentes àquelas rotas.

Um aspecto importante sobre a integração da topologia virtual com o BGP deve-se ao fato de que o BGP não leva em conta nenhum aspecto de QoS para a seleção das rotas em direção aos prefixos de rede. Neste sentido, os pacotes IP em um determinado domínio são enviados em direção ao roteador de saída do domínio que foi selecionado pelo BGP no momento em que a rota para cada prefixo foi escolhida. A decisão pela "melhor" rota é feita pelo protocolo BGP levando-se em consideração alguns aspectos administrativos que são implementados através de atributos, tal como o *Local_Pref*

que permite mudar o roteador de saída de um domínio em direção a um determinado prefixo de rede. Neste caso, ao usarmos o mecanismo de topologias virtuais para a seleção de rotas com QoS, o roteador de saída de um determinado domínio selecionado pelo BGP pode não ser o mesmo quando usamos as topologias virtuais. Sendo assim, o atributo *Local_pref* deve ser alterado para refletir a preferência local levando-se em conta os aspectos de QoS. Todos os domínios na rota devem alterar seu atributo caso o roteador de saída escolhido pelo BGP seja diferente do roteador de saída escolhido usando as topologias virtuais.

Este tipo de solução que altera o atributo *Local_Pref* vem sendo usada por alguns domínios *stubs* a fim de selecionarem seus provedores [Quoitin et al., 2003]. Primeiramente, os domínios *stubs* medem a carga dos enlaces com seus provedores a fim de detectarem quais estão mais congestionados. Após esta identificação, os administradores (manualmente ou através de uma ferramenta de gerência automática) alteram o valor do atributo *Local_Pref* associando um peso maior ao enlace com mais banda disponível.

Nesta tese, não analisamos qual o impacto da mudança no atributo *Local_Pref*. O objetivo foi analisar a integração da camada de serviços com o BGP. Para isso, implementamos o cenário da Figura 7.18 da seguinte forma: cada roteador de borda é representado por uma máquina virtual Linux. Usamos a máquina virtual QEMU [QEMU, 2006]. Cada máquina virtual está executando um *daemon* BGP responsável pela troca de informações de alcançabilidade entre os domínios. As MIBs BGP são obtidas usando o protocolo SNMP. Usamos a *suite* UCD-SNMP [Net-SNMP, 2006] (atualmente conhecida como Net-SNMP). A comunicação entre o agente SNMP e o processo BGP ocorre através do protocolo SMUX [Rose, 1991]. A Figura 7.19 apresenta a arquitetura que integra o serviço de divulgação com o protocolo BGP.

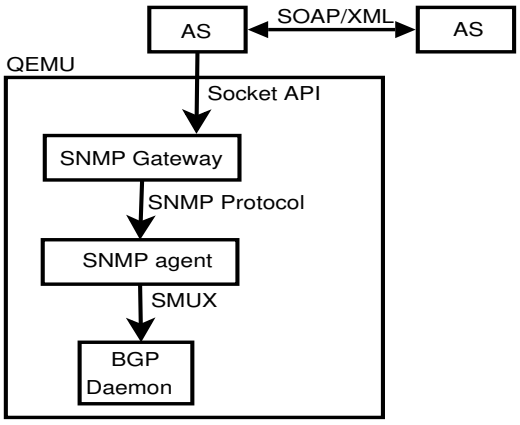


Fig. 7.19: Integração da camada de serviços com o protocolo BGP.

A camada de serviços, através do *Advertising Service* (AS), invoca um *gateway* para obtenção das rotas BGP locais. O SNMP *gateway* é responsável pela integração entre a camada de serviços e o protocolo BGP. Ele age como um cliente SNMP que invoca as operações no agente SNMP. O

comando SNMP invocado é o *snmpwalk*. O *gateway* recebe a invocação da camada de serviços e a converte para o comando SNMP. Ao receber a resposta do agente SNMP, o *gateway* a converte no formato compreendido pela camada de serviços.

Os testes consistiram em analisar os tempos para estabelecimento de conexões entre domínios usando os modelos *Push* e *Pull*. A principal diferença entre os dois modelos está relacionada ao fato de que no modelo *Push*, quando o domínio requisitante deseja enviar pacotes IP com QoS, as topologias virtuais dos domínios já estão disponíveis localmente. No modelo *Pull*, as topologias virtuais precisam ser obtidas no momento da requisição. Os testes foram feitos usando a topologia apresentada na Figura 7.18. Executamos 100 requisições e calculamos a média. O tempo médio para estabelecimento de uma conexão entre domínios neste cenário usando o modelo *Push* é de 205 ms. Este tempo representa exatamente os passos e as análises feitas na Seção 7.1. Quando o modelo *Pull* é usado, este tempo sobe para aproximadamente 1s, considerando apenas as rotas BGP locais do AS 65001. Este tempo inclui, além dos passos normais apresentados na Seção 7.1, a consulta ao agente SNMP para obtenção das rotas locais, a manipulação das mensagens entre o *gateway* e a camada de serviços, e a busca pelas topologias virtuais nas duas rotas BGP consideradas. Quando o domínio local (65001) sobe um nível na hierarquia da Figura 7.18 para obter outras rotas BGP, este tempo atinge 3s para atender cada requisição. Este tempo inclui, além dos passos anteriores, a invocação do serviço de divulgação dos domínios 65004 e 65003 para obtenção das rotas BGP não divulgadas por aqueles domínios.

O tempo para obtenção das rotas BGP via agente SNMP considerando o momento em que a requisição sai da camada de serviços e volta com as rotas coletadas é de aproximadamente 660 ms. Em cenários reais onde as tabelas BGP tendem a ser grandes, aconselha-se usar o comando *snmpbulk*.

7.4.1 Discussão Final

Os testes realizados mostraram que o modelo *Pull* possui tempos maiores do que os tempos coletados para o modelo *Push*. No modelo *Pull* as topologias virtuais precisam ser obtidas no momento da requisição, e devido a isso, tais tempos tendem a aumentar. Além disso, a invocação do agente SNMP também gera um consumo de tempo que não existe no modelo *Push*. Entretanto, o modelo *Pull* se mostra mais apropriado para cenários reais, tais como redes IP tradicionais.

Os contratos entre os domínios administrativos deveriam levar em consideração a agregação de fluxos IP a fim de evitar a oscilação e a instabilidade das tabelas de roteamento BGP. Isto poderia ser feito através do uso de matrizes de tráfego. Acreditamos que realizar todo o processo de obtenção de topologias virtuais e negociação entre domínios para cada fluxo IP não é viável. Além disso, fluxos dinâmicos que não foram considerados na matriz de tráfego poderiam ser agregados em conexões já estabelecidas desde que não afetem a QoS de tais conexões.

Por fim, mais estudos a fim de analisar a viabilidade da solução apresentada precisam ser feitos. Trabalhos futuros nesta direção incluem a análise de mudança dos atributos BGP e o estabelecimento de contratos que garantam a QoS solicitada. Recentemente um artigo [Xu and Rexford, 2006] apresentou idéias bastante próximas das apresentadas nesta seção. O trabalho citado também usa o modelo *Pull* para obtenção de mais rotas em direção aos prefixos de rede. Além disso, o trabalho mencionado realiza uma negociação bilateral a fim de obter outras rotas e negociar rotas alternativas àquelas disseminadas pelo BGP.

7.5 Ferramentas de Auxílio

O desenvolvimento de ferramentas auxiliares que dessem suporte à implementação da arquitetura geraram alguns trabalhos cujos resultados serão brevemente apresentados aqui. O primeiro deles é uma ferramenta para auxílio na criação e distribuição das topologias virtuais. Tal ferramenta permite a criação das topologias virtuais através da definição de nós físicos e enlaces virtuais com seus respectivos custos. A segunda ferramenta permite monitorar o consumo de recursos das topologias virtuais. O terceiro trabalho originou um Registro de *Web Services*. Tal registro permite que *Web Services* sejam adicionados, consultados e removidos. Na verdade, o registro implementa em menor número as funcionalidades dos UDDIs. Devido ao fato de que durante o desenvolvimento desta tese os UDDIs ainda estavam em definição, optamos por desenvolver o nosso próprio registro. Atualmente, a especificação do UDDI está sendo definida pelo consórcio OASIS [UDDI, 2005] e as versões 2 e 3 se tornaram padrões OASIS.

7.5.1 Ferramenta para Criação e Distribuição de Topologias Virtuais

A ferramenta possui uma interface gráfica que permite o desenvolvimento de topologias virtuais. Após a criação das topologias, a ferramenta gera os arquivos XML que correspondem a cada topologia virtual criada. Estes arquivos são então distribuídos para os respectivos domínios de forma automática. A ferramenta permite criar as topologias virtuais internas de um domínio assim como as conexões virtuais entre domínios. A Figura 7.20 mostra duas interfaces da ferramenta.

O lado esquerdo da figura mostra a interface para inserção e remoção de nós e arestas para criação da topologia virtual de um domínio. O botão *Save* permite salvar a topologia no formato XML e o botão *Advertise* permite distribuir a topologia virtual para o seu respectivo domínio. O lado direito da figura mostra a criação de conexões virtuais entre domínios. Na figura aparece a criação das conexões virtuais entre os domínios *alfaromeo* e *aprilia*.

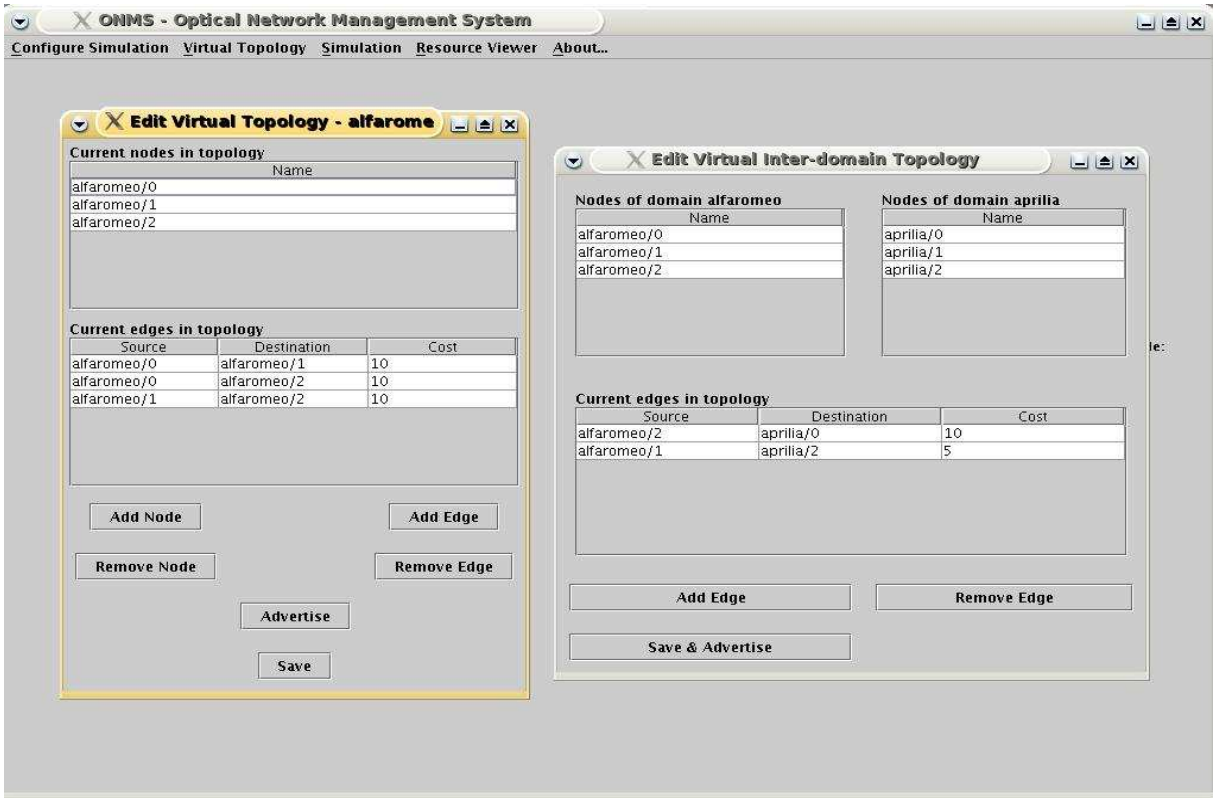


Fig. 7.20: Interface para criação de topologias virtuais.

7.5.2 Ferramenta para Monitoramento do Consumo de Recursos nas Topologias Virtuais

Esta ferramenta foi desenvolvida para que o gerente possa monitorar o consumo dos recursos ópticos nos domínios assim como os recursos ópticos nos enlaces virtuais entre domínios. A interface gráfica permite identificar os locais nas topologias virtuais onde há uma maior demanda pelos recursos ópticos. Com isso, o administrador do domínio pode realizar ações de gerência tais como estabelecer mais caminhos de luz em determinados enlaces virtuais ou divulgar novas topologias virtuais. A Figura 7.21 mostra a ferramenta de monitoramento e o estado dos enlaces virtuais entre os domínios.

Os rótulos em cada enlace virtual representam a quantidade de recursos consumida e a quantidade total de recursos naquele enlace. Por exemplo, o enlace virtual que conecta o domínio *ducati* com o domínio *aprilia* possui o rótulo *0/8* (do lado do domínio *ducati*) significando que dos oito recursos disponíveis nenhum foi consumido. Porém, o rótulo *3/8* (do lado do domínio *aprilia*) no mesmo enlace, significa que dos oito recursos disponíveis, três estão usados. Como os enlaces são bidirecionais, aparecem dois rótulos em cada enlace virtual, um para cada direção. Ao clicar duplamente em cada domínio da interface, outra interface correspondente àquele domínio se abre e mostra o estado dos enlaces do referido domínio.

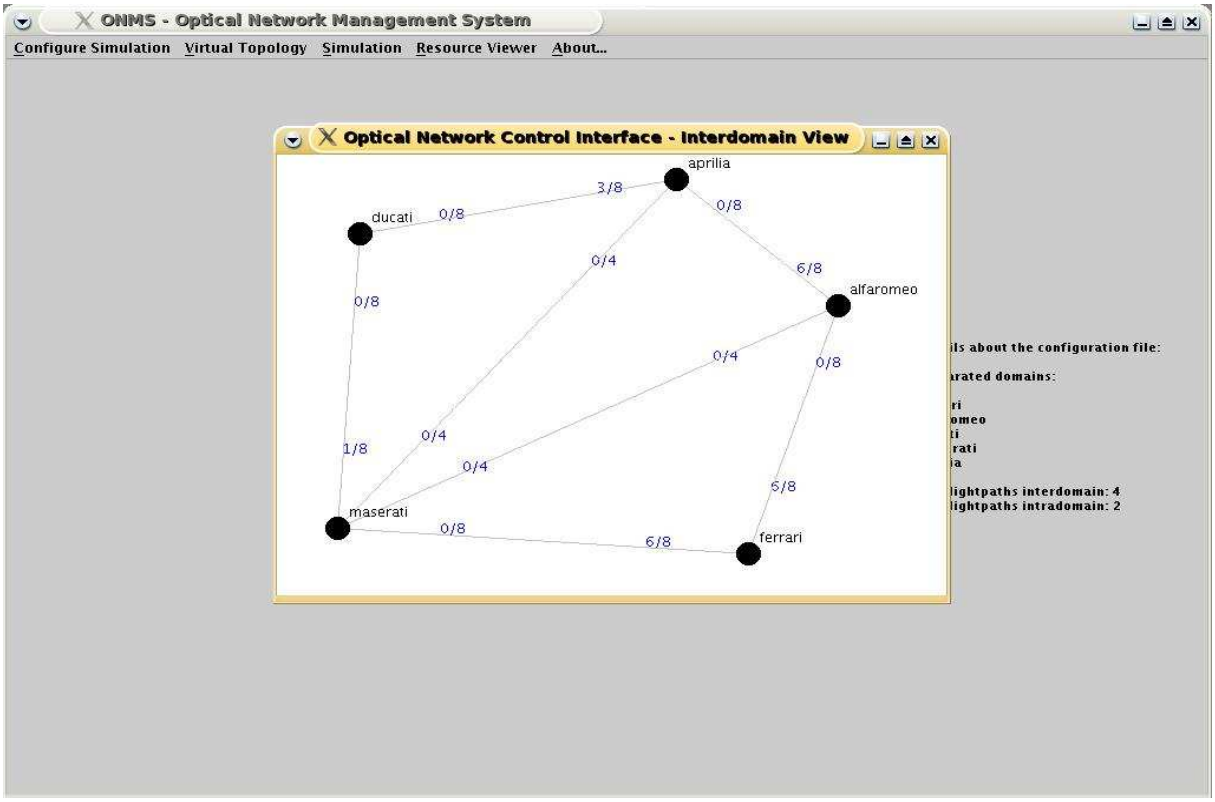


Fig. 7.21: Interface para monitoramento dos enlaces virtuais entre domínios.

A Figura 7.22 mostra a interface de monitoramento e o estado dos enlaces virtuais dentro de alguns domínios. Observe que o enlace virtual conectando o nó *alfaromeo/0* com o nó *alfaromeo/1* não possui mais recursos ópticos na direção do primeiro para o segundo. O rótulo *2/2* aparece em vermelho e significa que todos os recursos ópticos naquele enlace já estão usados.

7.5.3 Registro de *Web Services*

O serviço de registro de *Web Services* permite registrar e remover serviços assim como pesquisar por serviços através de palavras-chave. O serviço de registro oferece uma interface Web que permite a navegação e o acesso às funcionalidades do registro.

A inserção (registro) de serviços consiste em inserir informações relacionadas ao serviço sendo registrado. Informações típicas necessárias incluem a URL onde o WSDL do serviço está localizado, uma descrição do serviço, o estilo de comunicação do serviço (RPC ou *document*) e quem está oferecendo o serviço. A Figura 7.23 mostra a interface Web utilizada para registrar um serviço.

Uma das principais funcionalidades do serviço de registros é a busca por serviços oferecidos pelos provedores. Neste sentido, o serviço de registros desenvolvido permite listar todos os serviços registrados ou realizar uma busca por palavras-chave. A Figura 7.24 mostra a interface Web após a

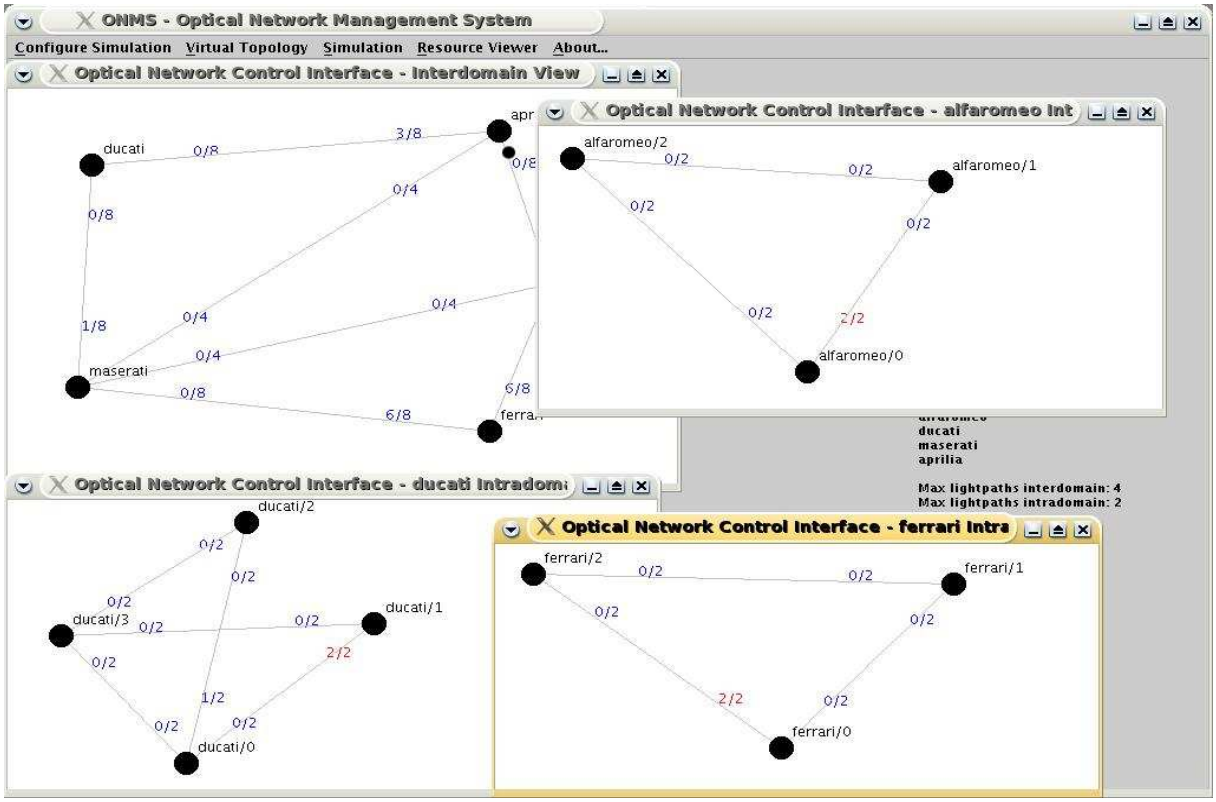


Fig. 7.22: Interface para monitoramento dos enlaces virtuais dentro dos domínios.

busca pela palavra-chave “e2e”.

Finalizações do capítulo

Neste capítulo apresentamos os detalhes sobre a implementação da arquitetura. Focamos na análise de desempenho da tecnologia *Web Services* (tempo e consumo de banda) para o estabelecimento de conexões e VPNs entre domínios. Além disso, mostramos como a arquitetura pode ser integrada ao BGP a fim de ser usada em um cenário com redes IP tradicionais e comparamos os modelos *Push* e *Pull*. No próximo capítulo, apresentamos as conclusões e as contribuições deste trabalho.

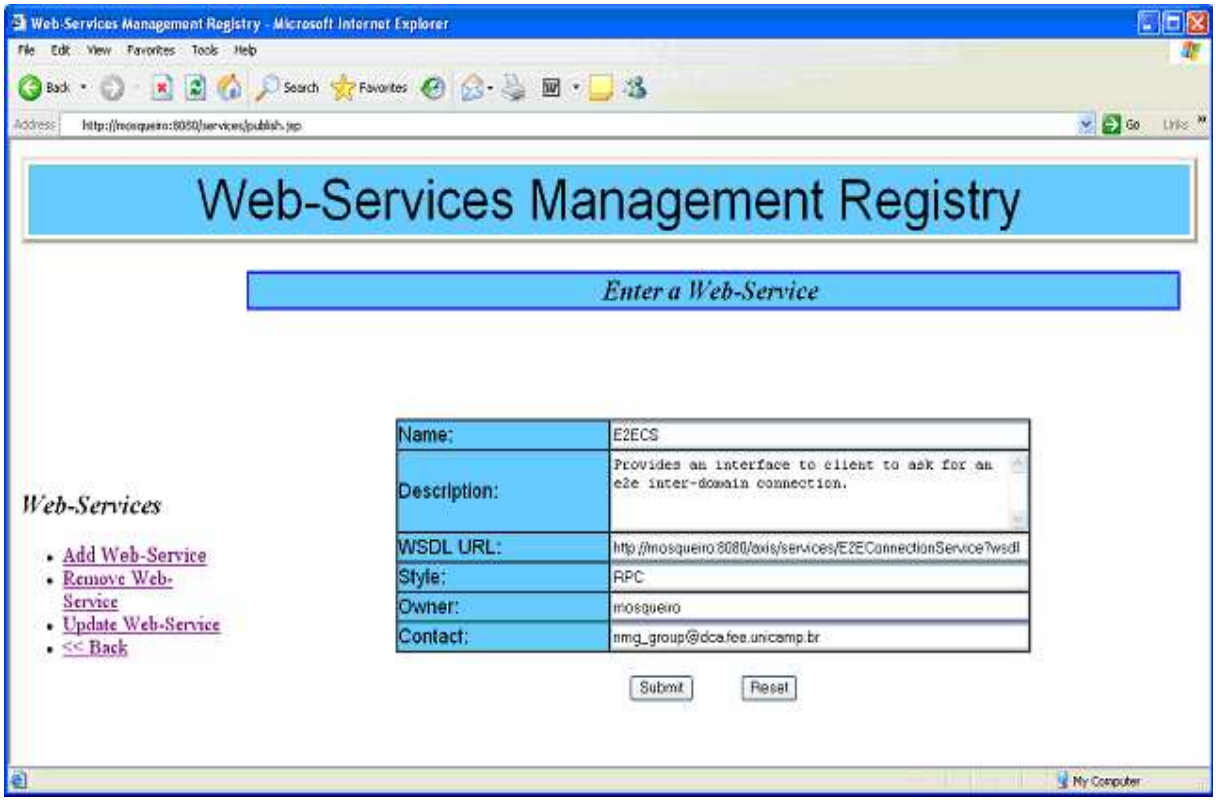


Fig. 7.23: Interface Web para registro de serviços.

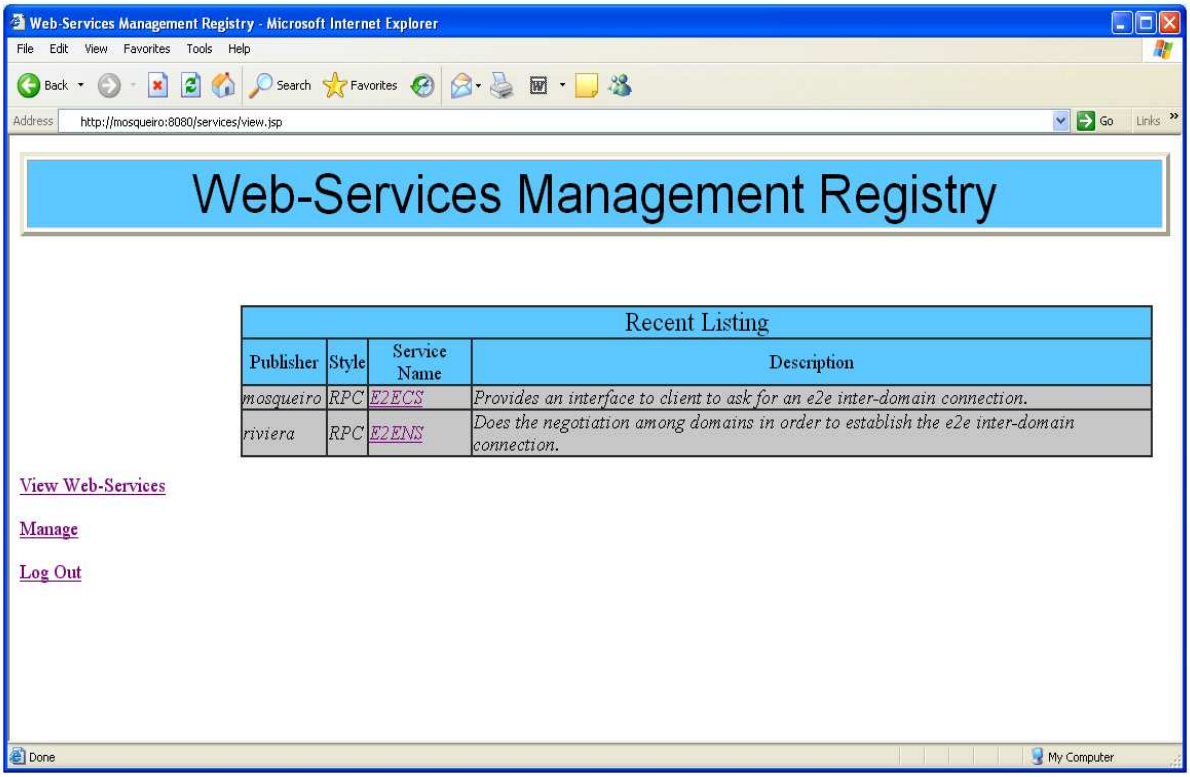


Fig. 7.24: Interface Web para listagem de serviços.

Capítulo 8

Conclusão

Nesta tese desenvolvemos uma arquitetura para provisionamento e gerência de serviços em redes ópticas. Tal arquitetura suporta tanto o provisionamento e gerência de serviços dentro de um domínio como também o provisionamento e gerência de serviços entre domínios administrativos diferentes. Localmente, ou seja, dentro de um mesmo domínio, estamos particularmente interessados em oferecer serviços para provisionamento de conexões (SPCs) e VPNs. Além disso, um conjunto de políticas para realização de agregação (*grooming*) de tráfego foi definido. Estas políticas melhoram o uso da banda disponível em cada caminho de luz. Políticas mais complexas foram desenvolvidas para considerar aspectos relacionados à capacidade de proteção de cada caminho de luz levando-se em conta falhas na rede óptica. Tais políticas reduzem o impacto de uma falha na rede.

Em relação ao provisionamento e gerência de serviços entre domínios, a arquitetura suporta o estabelecimento de conexões e VPNs. Ambos os serviços fazem uso de topologias virtuais para cálculo de rota. As topologias virtuais representam os recursos ópticos que estão disponíveis em cada domínio e podem conter informações de QoS tais como custo, tipo de proteção e banda dos recursos em cada enlace virtual.

A arquitetura foi validada através do desenvolvimento de um protótipo que foi testado em nosso laboratório. Os detalhes relacionados à arquitetura para o provisionamento dos serviços internos ao domínio e as políticas de *grooming* fazem parte de três dissertações de mestrado. Nesta tese, mostramos apenas alguns resultados obtidos com tais trabalhos. Os mesmos foram desenvolvidos de forma conjunta a fim de obtermos um resultado final mais completo. Especificamente para esta tese, os testes foram feitos a fim de analisar o uso da tecnologia *Web Services* para o provisionamento de serviços entre domínios.

Em relação às principais contribuições desta tese, destacamos:

- Desenvolvemos um modelo baseado no modelo de referência TMN/FCAPS para provisionamento de serviços em redes ópticas. Instanciamos o modelo em uma arquitetura

que suporta o provisionamento e gerência de serviços intra e inter-domínios;

- O plano de serviços definido para a arquitetura permite que a sinalização de conexões entre domínios seja feita sem esperar pelos longos processos de padronização de interfaces. As interações necessárias para o estabelecimento de conexões e VPNs entre domínios ocorrem neste plano de serviços. Os domínios interagem para distribuir topologias virtuais e reservar recursos bastando para isso definirem as interfaces dos serviços;

- Usamos o conceito de topologias virtuais para abstração dos recursos locais em cada domínio e suporte ao provisionamento de serviços entre domínios ópticos. Recentemente, um *draft* IETF [Shiomoto et al., 2006] comenta brevemente sobre o uso de topologias virtuais através das *Forwarding Adjacencies*. É possível que tal mecanismo receba mais atenção e evolua em algum grupo de trabalho do IETF;

- A tecnologia *Web Services* foi testada em termos de consumo de tempo e tamanho das mensagens SOAP para o provisionamento dos serviços entre domínios. Concluímos que tal tecnologia é apropriada para este tipo de cenário uma vez que os tempos coletados assim como o tamanho das mensagens SOAP são pequenos considerando todo o processo para estabelecimento de uma conexão entre domínios e os atributos sendo trocados. Além disso, as ferramentas livres não comerciais utilizadas se mostraram bastante maduras para o desenvolvimento de aplicações *Web Services*.

Como contribuições secundárias, destacamos:

- A notação BPMN para modelagem dos processos de negócios e suas atividades facilitou a identificação das relações entre os módulos que compõem a camada de serviços e os módulos que compõem a camada de gerência de redes (camada de integração). As formas gráficas da notação permitiram modelar facilmente as atividades responsáveis pelo provisionamento dos dois serviços entre domínios definidos nesta tese;

- Dois serviços entre domínios foram desenvolvidos e testados: o serviço de conexões e o serviço de VPNs;

- Os serviços definidos para este tese podem servir como base para o desenvolvimento de outros serviços compostos mais complexos;

- Definimos dois modelos para obtenção das topologias virtuais: o modelo *Push* e o modelo *Pull*. O modelo *Pull* vem sendo usado também em outros trabalhos recentemente publicados [Xu and Rexford, 2006].

- O uso das topologias virtuais em redes IP tradicionais através do modelo *Pull* se mostrou apropriado para prover QoS entre domínios administrativos diferentes;
- O modelo TMN se mostrou apropriado para a definição da nomenclatura e a divisão da arquitetura em camadas de gerência;
- A arquitetura é flexível e permite a criação de outros serviços de forma independente dos existentes. Além disso, os módulos internos também podem ser estendidos sem afetar a camada de serviços.

Em relação aos trabalhos futuros vislumbramos algumas tarefas que completam esta tese podendo dar origem a outros trabalhos. Alguns deles são:

- Usar máquinas de orquestração e coreografia para composição dos serviços. O módulo AC poderia ser um orquestrador responsável pelo controle do estabelecimento dos serviços de conexão e VPN. Os diagramas BPMN que modelam as atividades poderiam ser usados como ponto de partida para a definição dos processos em uma linguagem de orquestração e coreografia. Algumas ferramentas que mapeiam diagramas BPMN em linguagem BPEL poderiam ser usadas nesta tarefa. O uso de máquinas de orquestração é algo que já foi iniciado em nosso grupo. Um trabalho de mestrado propôs serviços de "crossconexão" para OXCs [de Souza and Cardozo, 2005, de Souza, 2006]. Tais serviços permitem que a sinalização de um caminho de luz seja feita através da orquestração de serviços ao invés de ser feita através de protocolos GMPLS, como por exemplo o RSVP;
- Criar um modelo de tarifação entre domínios que considere os atributos específicos para redes ópticas. Esta tarefa também poderia incluir a definição de um mecanismo para monitoramento dos contratos estabelecidos entre os domínios;
- Criar um mecanismo de gerência de falhas entre domínios;
- Substituir o serviço de registros (*Registry*) desenvolvido para este trabalho por uma implementação UDDI. Os UDDIs permitem realizar uma busca mais completa sobre os serviços registrados. Assim, mais critérios de seleção de serviços poderiam ser usados pelos clientes e pelos domínios;
- Usar um mecanismo de segurança para criptografia das mensagens e autenticação das requisições; e
- Testar a integração do modelo *Pull* com o BGP através da alteração do atributo *Local_Pref* e também usando técnicas de encapsulamento e tunelamento.

A união da tecnologia *Web Services* com redes ópticas teve origem em dois projetos. Um deles, no contexto do projeto GIGA, tinha como objetivo criar um plano de controle baseado no GMPLS para provisionamento de conexões ópticas em um domínio. O outro projeto tinha como objetivo usar *Web Services* para a gerência de novas tecnologias de redes. Esta tese agrega as propostas que foram desenvolvidas nos dois projetos e apresenta um modelo e uma arquitetura que consideram o provisionamento e a gerência de serviços intra e entre domínios ópticos.

A arquitetura proposta nesta tese pode ser usada tanto para o provisionamento de conexões simples como para o estabelecimento de serviços mais sofisticados como o serviço de VPN entre domínios. Até onde sabemos, esta é a primeira arquitetura que considera o provisionamento de serviços intra e inter-domínios ópticos. Também sabemos que tal arquitetura pode evoluir através dos trabalhos futuros mencionados acima podendo ser incorporados à arquitetura atual ou desenvolvidos separadamente.

Referências Bibliográficas

[Agarwal et al., 2003] Agarwal, S., Chuah, C., and Katz, R. (2003). OPCA: Robust Interdomain Policy Routing and Traffic Control. *OPENARCH, San Francisco, USA*.

[Apache AXIS, 2006] Apache AXIS (2006). http://ws.apache.org/axis/.

[Apache Tomcat, 2006] Apache Tomcat (2006). http://tomcat.apache.org/.

[Arnaud, 2004] Arnaud, B. (2004). CA*net 4 Research Program Update - UCLP Roadmap. Web Services Workflow for Connecting Research Instruments and Sensors to Networks. *Draft*.

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

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

[Bernstein et al., 2003] Bernstein, G., Rajagopalan, B., and Saha, D. (2003). *Optical Network Control. Architecture, Protocols, and Standards*, chapter Modern Optical Network Control Plane, pages 155–158.

[Boutaba et al., 2004] Boutaba, R., Golab, W., Iraqui, Y., and Arnaud, B. S. (2004). Lightapths on Demand: A Web-Services-Based Management System. *IEEE Communications Magazine*, 42(7):2–9.

[BPMN, 2006] BPMN (2006). BPMN Specification. http://bmi.omg.org/.

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

[Brunner et al., 2004] Brunner, M., Nunzi, G., Dietz, T., and Kazuhiko, I. (2004). Customer-Oriented GMPLS Service Management and Resilience Differentation. *eTransactions on Network and Service Management*, pages 2–12.

[CANARIE Project, 2006] CANARIE Project (2006). http://www.canarie.ca/.

[Carvalho, 2006] Carvalho, C. (2006). Gerência de Falhas baseada em Políticas para Redes Ópticas. Dissertação de Mestrado, Unicamp - Instituto de Computação. Orientador: Prof. Dr. Edmundo Madeira.

[Carvalho et al., 2005] Carvalho, C., Verdi, F. L., Madeira, E., and Magalhães, M. (2005). Policy-based Fault Management for Integrating IP over Optical Networks. *The 5th IEEE International Workshop on IP Operations & Management (IPOM'05), LNCS-Springer-Verlag. Barcelona, Spain*, 3751:88–97.

[Carvalho et al., 2006] Carvalho, C., Verdi, F. L., Madeira, E., and Magalhães, M. (2006). Gerência de Falhas baseada em Políticas para Redes Ópticas. *Simpósio Brasileiro de Redes de Computadores (SBRC 06), Curitiba, Brasil*.

[Chen and Li, 2005] Chen, L. and Li, M. (2005). Using Web Services in TMN environment. *Proceedings of the IEEE EEE05 International Workshop on Business Services Networks, Hong Kong, China*.

[Christensen et al., 2001] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S. (2001). Web Services Description Language (WSDL) 1.1. W3C Note, Microsoft, IBM Research. `http://www.w3.org/TR/wsdl`.

[CivicNet, 2001] CivicNet (2001). Chicago CivicNet: Request For Information Chicago CivicNet.

[de Maesschalck et al., 2003] de Maesschalck, S., 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. *IEEE GLOBECOMM. San Francisco, USA*, 5:2750–2754.

[de Souza, 2006] de Souza, V. (2006). Uma Arquitetura Orientada a Serviços para Desenvolvimento, Gerenciamento e Instalação de Serviços de Rede. Dissertação de Mestrado, Unicamp - FEEC-DCA. Orientador: Prof. Dr. Eleri Cardozo.

[de Souza and Cardozo, 2005] de Souza, V. and Cardozo, E. (2005). A Service Oriented Architecture for Deploying and Managing Network Services. *Proceedings of the 3rd International Conference on Service Oriented Computing (ICSOC'05), LNCS-Springer-Verlag. Amsterdam, The Netherlands*, pages 465–477.

[Duarte, 2006] Duarte, R. (2006). Provisionamento baseado em Web Services de conexões fim-a-fim em Redes Ópticas GMPLS. Dissertação de Mestrado, Unicamp - FEEC- DCA. Orientador: Prof. Dr. Maurício Ferreira Magalhães.

[E-NNI, 2004] E-NNI (2004). OIF Intra-Carrier E-NNI 01.0 Signaling Specification.

[eclarus, 2006] eclarus (2006). eClarus Software. http://www.eclarus.com/.

[Erl, 2004a] Erl, T. (2004a). *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*, chapter Introduction to Web services technologies.

[Erl, 2004b] Erl, T. (2004b). *Service-Oriented Architecture. A Field Guide to Integrating XML and Web Services*. Prentice Hall.

[Fang et al., 2005] Fang, L., Bita, N., Roux, J., and Miles, J. (2005). Interprovider IP-MPLS Services: Requirements, Implementations, and Challenges. *IEEE Communications Magazine*, 43(6):119–128.

[Farrel et al., 2006] Farrel, A., Vasseur, J.-F., and Ash, J. (2006). Path Computation Element (PCE) Architecture. *IETF draft, work in progress*.

[Farrel et al., 2005] Farrel, A., Vasseur, J.-F., and Ayyangar, A. (2005). A Framework for Inter-Domain MPLS Traffic Engineering. *IETF draft, work in progress*.

[French and Pendarakis, 2004] French, S. and Pendarakis, D. (2004). Optical Virtual Private Networks: Applications, Functionality and Implementation. *Photonic Network Communications*, 7(3):227–238.

[Geer, 2005] Geer, D. (2005). Will Binary XML Speed Network Traffic. *IEEE Computer*, 38(4):16–18.

[GLASS, 2006] GLASS (2006). GLASS Simulator: http://dns.antd.nist.gov/glass/.

[Hamada et al., 2001] Hamada, T., Bystrom, L., and Berndt, H. (2001). Networking Technology Convergence in the Photonic Age - TINA Vision on IP Control and Management. *IEICE Transaction Communications*, E84-B(12).

[Hately et al., 2005] Hately, A., Riegen, C., and Rogers, T. (2005). UDDI Version 3.0.2. UDDI Spec Technical, IBM, SAP AG, Computer Associates. `http://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.htm`.

[Hendricks et al., 2002] Hendricks, M., Galbraith, B., Irani, R., Milbery, J., Modi, T., Tost, A., Toussaint, A., Basha, S. J., and Cable, S. (2002). *Professional Java Web Services*, chapter Architecture for Web Services.

[How the PCE Works, 2006] How the PCE Works (2006). http://www.bind.com/rfc/internet-drafts/draft-vasse

[Howarth et al., 2005] Howarth, M. P., Flegkas, P., Pavlou, G., Wang, N., and Trimintzios, P. (2005). Provisioning for Interdomain Quality of Service: the MESCAL Approach. *IEEE Communications Magazine*, 43(6):129–137.

[Iovanna et al., 2003] Iovanna, P., Setembre, M., and Sabella, R. (2003). A Traffic Engineering System for Multilayer Networks Based on the GMPLS Paradigm. *IEEE Network*, 17(2):28–37.

[ITU-T, 2003] ITU-T (2003). ITU-T Recommendation Y.1312: Layer 1 Virtual Private Network Generic Requirements and Architecture Elements.

[Kompella and Rekhter, 2005a] Kompella, K. and Rekhter, Y. (2005a). Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE). *IETF RFC 4206*.

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

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

[L1Charter, 2006] L1Charter (2006). IETF L1 Charter: http://www.ietf.org/html.charters/l1vpn-charter.html.

[Lakshminarayanan et al., 2004] Lakshminarayanan, K., Stoica, I., and Shenker, S. (2004). Routing as a Service. *UCB Technical Report No. UCB/CSD-04-1327*.

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

[Li and Gao, 2006] Li, D. and Gao, J. (2006). Directory Server based Information Distribution for L1VPN. *IETF draft, work in progress*.

[Li and Mohapatra, 2004] Li, Z. and Mohapatra, P. (2004). QRON: QoS-Aware Routing in Overlay Networks. *IEEE Journal on Selected Areas in Communications*, 22(1):29–40.

[Mahajan et al., 2005] Mahajan, R., Wetherall, D., and Anderson, T. (2005). Negotiation-based Routing Between Neighboring ISPs. *Networked Systems Design and Implementation (NSDI). Boston, USA*.

[Malheiros, 2006] Malheiros, N. (2006). Uma Arquitetura Baseada em Políticas para Gerência de VPNs de Camada 1. Dissertação de Mestrado, Unicamp - Instituto de Computação. Orientador: Prof. Dr. Edmundo Madeira.

[Malheiros et al., 2006a] Malheiros, N., Verdi, F. L., Madeira, E., and Magalhães, M. (2006a). A Management Architecture for Layer 1 VPN Services. *IEEE International Conference on Broadband Communications, Networks and Systems (Broadnets'06), San Jose, USA*.

[Malheiros et al., 2006b] Malheiros, N., Verdi, F. L., Madeira, E., and Magalhães, M. (2006b). Uma Arquitetura Baseada em Políticas para Gerência de VPNs de Camada 1. *Simpósio Brasileiro de Redes de Computadores (SBRC 06), Curitiba, Brasil*.

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

[Margaria et al., 2005] Margaria, C., Juillot, G., and Autenrieth, A. (2005). Performance Evaluation of a GMPLS Prototype. *IV Workshop in G/MPLS Networks. Girona, Spain*, pages 211–222.

[Net-SNMP, 2006] Net-SNMP (2006). http://net-snmp.sourceforge.net/.

[OASIS, 2006] OASIS (2006). OASIS Consortium: http://www.oasis-open.org/.

[OIF, 2006] OIF (2006). OIF Forum: http://www.oiforum.com.

[Ould-Brahim et al., 2006a] Ould-Brahim, H., Fedyk, D., and Rekhter, Y. (2006a). BGP-based auto-discovery for L1VPNs. *IETF draft, work in progress*.

[Ould-Brahim et al., 2006b] Ould-Brahim, H., Fedyk, D., and Rekhter, Y. (2006b). Traffic Engineering Attribute. *IETF draft, work in progress*.

[Ould-Brahim, H. and Rekhter, Y., 2005] Ould-Brahim, H. and Rekhter, Y. (2005). GVPN Services: Generalized VPN Services using BGP and GMPLS Toolkit. IETF draft, work in progress.

[OWLS, 2006] OWLS (2006). http://www.w3.org/Submission/OWL-S/.

[Pavlou et al., 2004] Pavlou, G., Flegkas, P., Gouveris, S., and Liotta, A. (2004). On Management Technologies and the Potential of Web Services. *IEEE Communications Magazine*, 42(7):58–66.

[Pras et al., 2004] Pras, A., Drevers, T., Meent, R., and Quartel, D. (2004). Comparing the Performance of SNMP and Web Services-Based Management. *IEEE eTransactions on Network and Service Management*, 1(2):72–82.

[Pujol et al., 2005] Pujol, J., Schmid, S., Eggert, L., Brunner, M., and Quittek, J. (2005). Scalability Analysis of the TurfNet Naming and Routing Architecture. *First International ACM Workshop on Dynamic Interconnection of Networks, Cologne, Germany*, pages 28–32.

[QEMU, 2006] QEMU (2006). http://fabrice.bellard.free.fr/qemu/.

[Quagga, 2006] Quagga (2006). Quagga. http://www.quagga.net/.

[Quoitin et al., 2003] Quoitin, B., Pelsser, C., Swinnen, L., Bonaventure, O., and Uhlig, S. (2003). Interdomain Traffic Engineering with BGP. *IEEE Communications Magazine*, 41(5):122–128.

[Rekhter and Li, 2006] Rekhter, Y. and Li, T. (2006). A Border Gateway Protocol 4 (BGP-4). *IETF RFC 4271*.

[Ricciato et al., 2005] Ricciato, F., Monaco, U., and Ali, D. (2005). Distributed Schemes for Diverse Path Computation in Multidomain MPLS Networks. *IEEE Communications Magazine*, 43(6):138–146.

[Rose, 1991] Rose, K. (1991). SNMP MUX Protocol and MIB. *IETF RFC 1227*.

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

[Sarangan and Chen, 2006] Sarangan, V. and Chen, J.-C. (2006). Comparative Study of Protocols for Dynamic Service Negotiation in the Nex-Generation Internet. *IEEE Communications Magazine*, 44(3):151–156.

[Shiomoto et al., 2006] Shiomoto, K., Papadimitriou, D., Roux, J.-L., Vigoureux, M., and Brungard, D. (2006). Requirements for GMPLS-based multi-region and multi-layer networks (MRN/MLN). *IETF draft, work in progress*.

[Stokab, 2006] Stokab (2006). Stockholm Municipal Network Stokab: http://www.stokab.se.

[Subramanian et al., 2002] Subramanian, L., Agarwal, S., Rexford, J., and Katz, R. (2002). Characterizing the Internet Hierarchy from Multiple Vantage Points. *IEEE Infocom. New York, USA*.

[Takeda et al., 2005] Takeda, T., Brungard, D., Papadimitriou, D., and Ould-Brahim, H. (2005). Layer 1 Virtual Private Networks: Driving Forces and Realization by GMPLS. *IEEE Communications Magazine*, 43(7):60–67.

[Takeda et al., 2004] Takeda, T., Inoue, I., Aubin, R., and Carugi, M. (2004). Layer 1 Virtual Private Networks: Service Concepts, Architecture Requirements, and Related Advances in Standardization. *IEEE Communications Magazine*, 42(6):132–138.

[Thurm, 2002] Thurm, B. (2002). Web Services for Network Management - A Universal Architecture and Its Application to MPLS Networks. *27th Annual IEEE Conference on Local Computer Networks (LCN'02). Tampa, USA*.

[TMN, 1985] TMN (1985). ITU-T: Telecommunications Management Network (TMN), M.3000.

[Truong et al., 2004] Truong, D. L., Cherkaoui, O., Elbiaze, H., Rico, N., and Aboulhamid, M. (2004). A Policy-based approach for user controlled Lightpath Provisioning. *IFIP/IEEE NOMS. Seoul, Korea*, pages 859–872.

[UDDI, 2005] UDDI (2005). OASIS UDDI Specification: http://www.oasis-open.org/.

[Valancius, 2006] Valancius, V. (2006). GMPLS Extensions to BGP. *Master Thesis. http://web.it.kth.se/ vval/thesis/index.html*.

[Vasseur et al., 2004] Vasseur, J., Ayyangar, A., and Zhang, R. (2004). Inter-domain Traffic Engineering LSP path computation methods. *IETF draft, work in progress*.

[Verdi et al., 2005a] Verdi, F. L., Carvalho, C., Madeira, E., and Magalhães, M. (2005a). Policy-based Grooming in Optical Networks. *4th IEEE Latin American Network Operations and Management Symposium (LANOMS 2005). Porto Alegre, Brasil*, pages 125–136.

[Verdi et al., 2007a] Verdi, F. L., Carvalho, C., Madeira, E., and Magalhães, M. (2007a). Policy-based Grooming in Optical Networks. *Journal of Network and Systems Management (JNSM), Springer. Aceito para publicação. Versão estendida do artigo publicado no LANOMS 2005*.

[Verdi et al., 2006a] Verdi, F. L., de Lacerda, F., Duarte, R., Madeira, E., Cardozo, E., and Magalhães, M. (2006a). Provisioning and Management of Inter-Domain Connections in Optical Networks: A Service Oriented Architecture-based Approach. *IEEE/IFIP Network Operations and Management Symposium (NOMS'06). Vancouver, Canada*.

[Verdi et al., 2005b] Verdi, F. L., Duarte, R., de Lacerda, F., Madeira, E., Cardozo, E., and Magalhães., M. (2005b). Web Services-based Provisioning of Connections in GMPLS Optical Networks. *The Brazilian Symposium on Computer Networks (SBRC 2005). Fortaleza, Brasil*.

[Verdi et al., 2004] Verdi, F. L., Madeira, E., and Magalhães, M. (2004). Policy-based Admission Control in GMPLS Optical Networks. *First IEEE Broadnets'04 (formerly OptiComm), San Jose, USA*, pages 337–339.

[Verdi et al., 2006b] Verdi, F. L., Madeira, E., and Magalhães, M. (2006b). On the Performance of Interdomain Provisioning of Connections in Optical Networks using Web Services. *IEEE International Symposium on Computers and Communications (ISCC'06). Sardinia, Italy.*, pages 955–960.

[Verdi et al., 2006c] Verdi, F. L., Madeira, E., and Magalhães, M. (2006c). The Virtual Topology Service: A Mechanism for QoS-enabled Interdomain Routing. *The 6th IEEE International Workshop on IP Operations & Management (IPOM'06), LNCS-Springer-Verlag. Dublin, Ireland*, 4268:205–217.

[Verdi et al., 2006d] Verdi, F. L., Madeira, E., and Magalhães, M. (2006d). Web Services and SOA as Facilitators for ISPs. *International Conference on Telecommunications (ICT'06). Madeira Island, Portugal*.

[Verdi et al., 2006e] Verdi, F. L., Madeira, E., and Magalhães, M. (2006e). Web Services for the New Internet: Discussion and Evaluation of the Provisioning of Interdomain Services. *IEEE International Telecommunications Symposium (ITS'06). Fortaleza, Brasil*.

[Verdi et al., 2007b] Verdi, F. L., Madeira, E., Magalhães, M., Cardozo, E., and Welin, A. (2007b). A Service Oriented Architecture-based Approach for Interdomain Optical Network Services. *Journal of Network and Systems Management (JNSM), Springer. Aceito para publicação*.

[W3C, 2005] W3C (2005). SOAP Specifications. Technical report, W3C. `http://www.w3.org/TR/soap/`.

[WS-Security, 2004] WS-Security (2004). OASIS Consortium: http://www.oasis-open.org/.

[Xiao et al., 2004] Xiao, L., Wang, J., Lui, K.-S., and Nahrstedt, K. (2004). Advertising Interdomain Qos Routing. *IEEE Journal on Selected Areas in Communications*, 22(10):1949–1964.

[Xu and Rexford, 2006] Xu, W. and Rexford, J. (2006). MIRO: Multi-path Interdomain Routing. *IEEE SIGCOMM'06. Pisa, Italy*, pages 171–182.

[Yang et al., 2006] Yang, X., Lehman, T., Tracy, C., and adn S. Gong, J. S. (2006). Policy-based Resource Management and Service Provisioning in GMPLS Networks. *Workshop on Adaptive Policy-based Management in Network Management and Control. In Conjunction with IEEE Infocom. Barcelona, Spain*.

[Zhang et al., 2004] Zhang, Z., Zhang, Y.-Q., Chu, X., and Li, B. (2004). An Overview of Virtual Private Network (VPN): IP VPN and Optical VPN. *Photonic Network Communications*, 7(3):213–225.

[Zhu and Mukherjee, 2002] Zhu, K. and Mukherjee, B. (2002). Traffic Grooming in an Optical WDM Mesh Network. *IEEE Journal on Selected Areas in Communications*, 20(1):122–133.

Apêndice A

Diagramas de Classes

A.1 Diagrama de Classes da Arquitetura (Parcial)

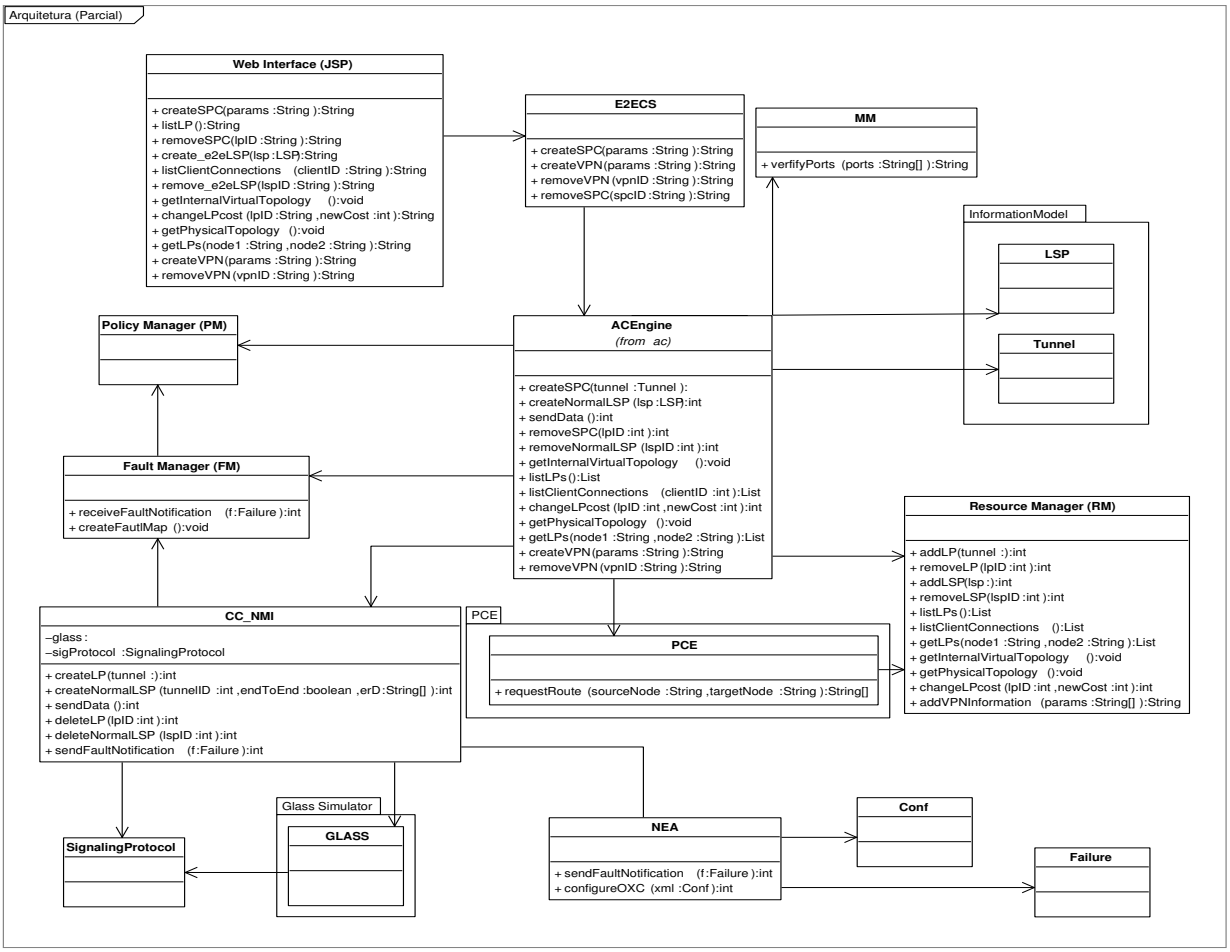


Fig. A.1: Diagrama de Classes da Arquitetura.

A.2 Diagrama de Classes do *Policy Manager*

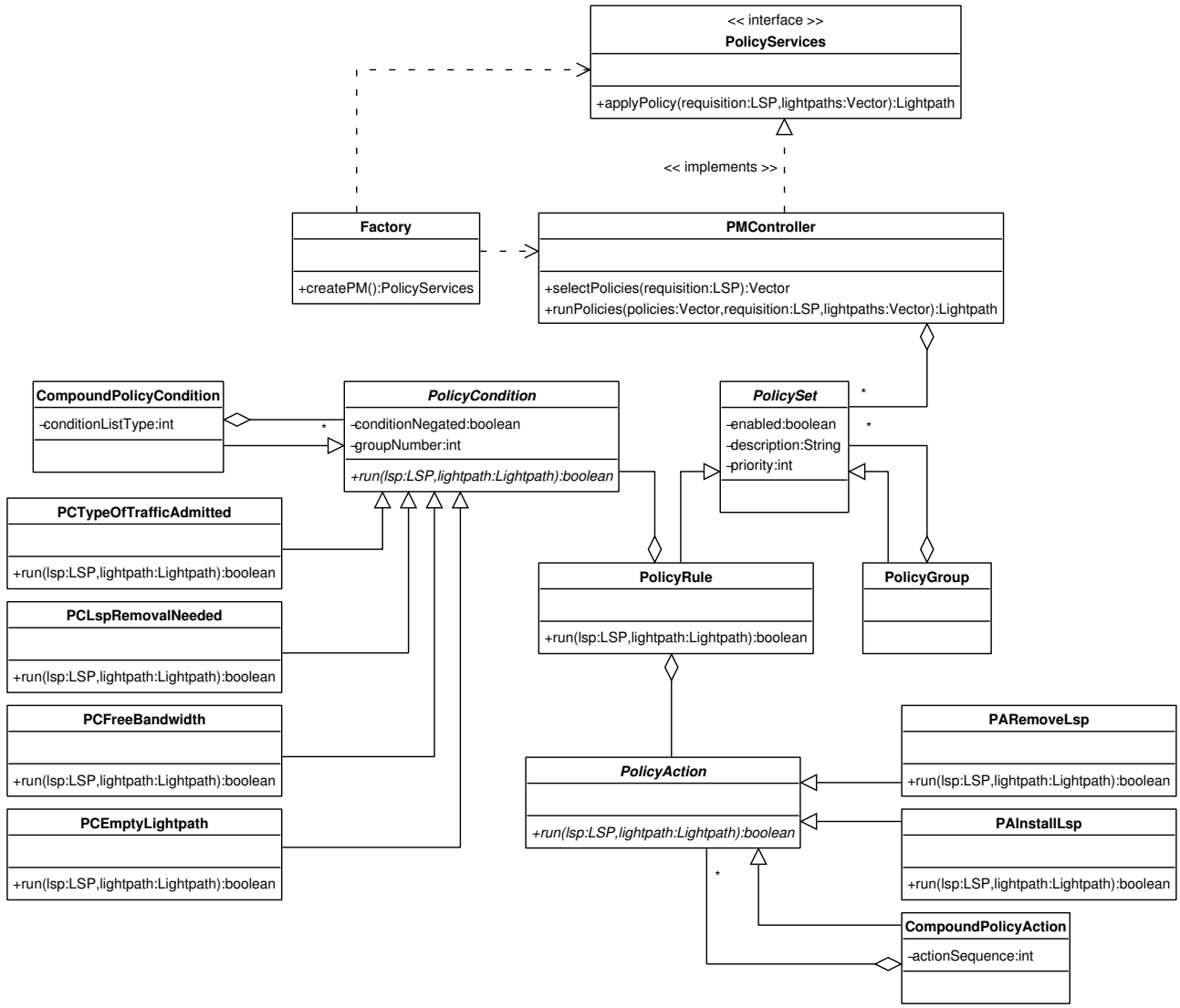


Fig. A.2: Diagrama de classes do PM.

A.3 Diagrama de Classes da Arquitetura (Completo)

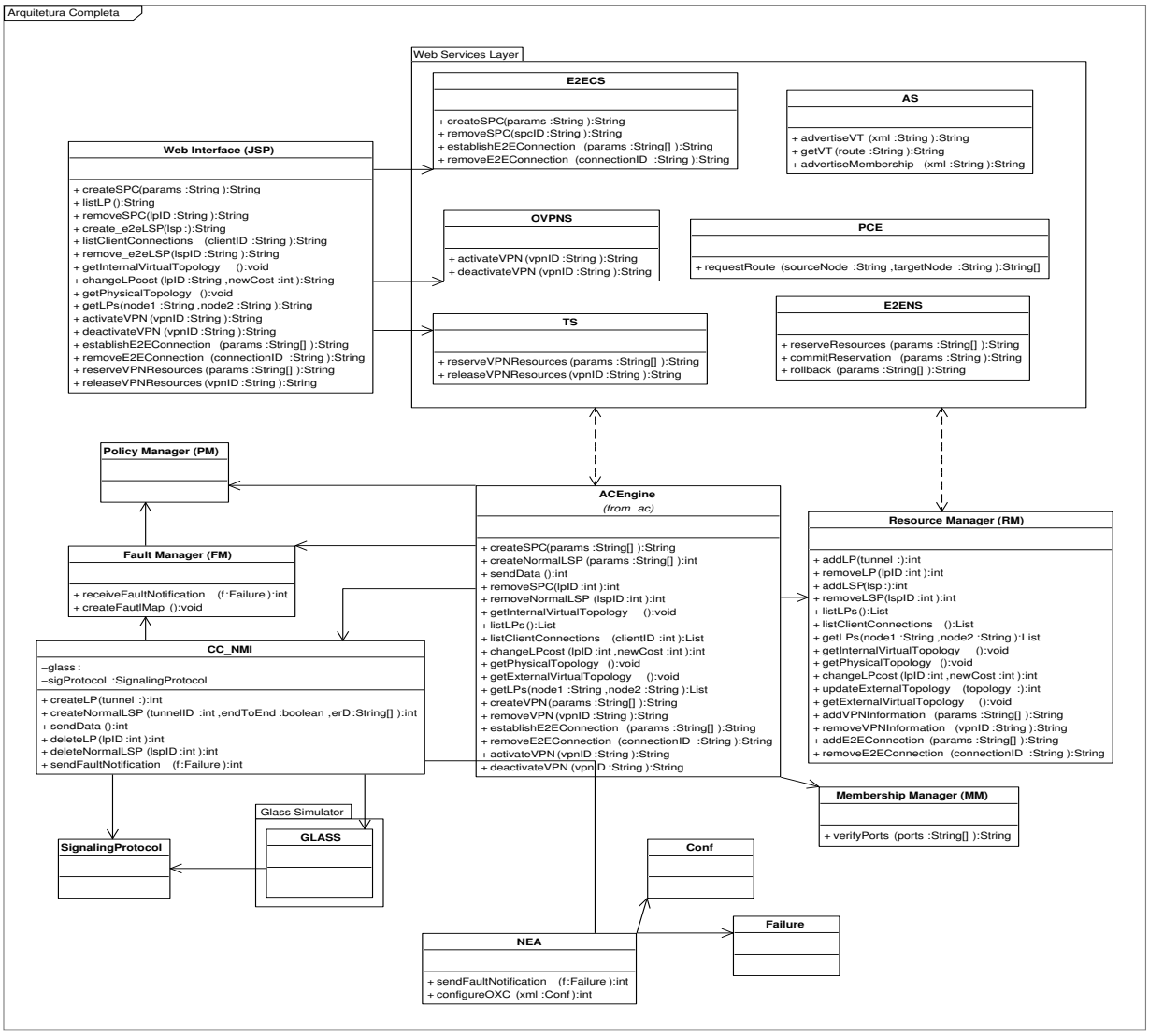


Fig. A.3: Diagrama de Classes da Arquitetura.

Apêndice B

Interfaces

B.1 Interface Web para Criação de uma SPC

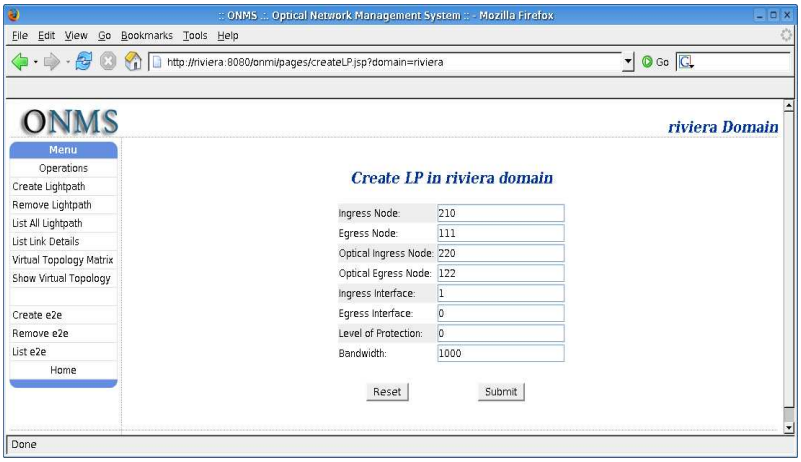


Fig. B.1: Página web para criação de uma conexão SPC.

Apêndice C

WSDL dos Serviços

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="urn:http://www.dca.fee.unicamp.br">
    <wsdl:message name="createLightpathRequest">
        <wsdl:part name="in0" type="xsd:string"/>          <!-- Nó IP/MPLS de ingresso -->
        <wsdl:part name="in1" type="xsd:string"/>          <!-- Nó IP/MPLS de egresso -->
        <wsdl:part name="in2" type="xsd:string"/>          <!-- Interface de ingresso -->
        <wsdl:part name="in3" type="xsd:string"/>          <!-- Interface de egresso -->
        <wsdl:part name="in4" type="xsd:string"/>          <!-- Largura de banda do caminho óptico -->
        <wsdl:part name="in5" type="xsd:string"/>          <!-- Nível de proteção -->
        <wsdl:part name="in6" type="xsd:string"/>          <!-- Nó de ingresso do domínio óptico -->
        <wsdl:part name="in7" type="xsd:string"/>          <!-- Nó de egresso do domínio óptico -->
    </wsdl:message>
    <wsdl:message name="createLightpathResponse">
        <wsdl:part name="createLightpathReturn" type="xsd:string"/> <!-- Confirmação de retorno -->
    </wsdl:message>
    ..................      ..................      ..................
    <wsdl:service name="SPCService">
        <wsdl:port binding="impl:SPCSoapBinding" name="SPC">
            <wsdlsoap:address location="http://localhost:8080/axis/services/SPC"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

Fig. C.1: Trecho WSDL do Web Service para Estabelecimento de uma SPC.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="urn:E2EConnectionService" xmlns:apachesoap="http://xml.apache.org/xml-soap" x
<!--WSDL created by Apache Axis version: 1.2
Built on May 03, 2005 (02:20:24 EDT)-->
 <wsdl:types>
  <schema elementFormDefault="qualified" targetNamespace="urn:E2EConnectionService" xmlns="http://www.w3.org/200
   <element name="establishE2EConnection">
    <complexType>
     <sequence>
      <element name="in0" type="xsd:string"/>
      <element name="in1" type="xsd:string"/>
      <element name="in2" type="xsd:int"/>
      <element name="in3" type="xsd:int"/>
      <element name="in4" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="establishE2EConnectionResponse">
    <complexType>
     <sequence>
      <element name="establishE2EConnectionReturn" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="removeE2EConnection">
    <complexType>
     <sequence>
      <element name="in0" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="removeE2EConnectionResponse">
    <complexType>
     <sequence>
      <element name="removeE2EConnectionReturn" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
  </schema>
 </wsdl:types>
...
...
```

Fig. C.2: Trecho WSDL do E2ECS.

```
...
...
 <schema elementFormDefault="qualified" targetNamespace="urn:E2ENegotiationService" xmlns="http://www.w3.org/2
   <element name="startE2ENegotiation">
    <complexType>
     <sequence>
      <element maxOccurs="unbounded" name="in0" type="xsd:string"/>
      <element name="in1" type="xsd:int"/>
      <element name="in2" type="xsd:int"/>
      <element name="in3" type="xsd:string"/>
      <element name="in4" type="xsd:string"/>
      <element name="in5" type="xsd:string"/>
      <element name="in6" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="startE2ENegotiationResponse">
    <complexType>
     <sequence>
      <element name="startE2ENegotiationReturn" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="e2ENegotiation">
    <complexType>
     <sequence>
      <element maxOccurs="unbounded" name="in0" type="xsd:string"/>
      <element name="in1" type="xsd:int"/>
      <element name="in2" type="xsd:int"/>
      <element name="in3" type="xsd:string"/>
      <element name="in4" type="xsd:string"/>
      <element name="in5" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="e2ENegotiationResponse">
    <complexType>
     <sequence>
      <element name="e2ENegotiationReturn" type="xsd:string"/>
     </sequence>
    </complexType>
     </element>
   <element name="commitReserve">
    <complexType>
     <sequence>
      <element name="in0" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="commitReserveResponse">
    <complexType>
     <sequence>
      <element name="commitReserveReturn" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="rollbackReserve">
    <complexType>
     <sequence>
      <element name="in0" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="rollbackReserveResponse">
    <complexType>
     <sequence>
      <element name="rollbackReserveReturn" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="startRemoveE2EConnection">
...
...
```

Fig. C.3: Trecho WSDL do E2ENS.

```
...
...
 <wsdl:types>
  <schema elementFormDefault="qualified" targetNamespace="urn:VTS" xmlns="http://www.w3.org/2001/XMLSchema">
   <element name="initialAdvertiseVT">
    <complexType>
     <sequence>
      <element name="in0" type="impl:Topologies"/>
      <element name="in1" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <complexType name="Topologies">
    <sequence>
     <element maxOccurs="unbounded" name="idc" nillable="true" type="xsd:string"/>
     <element maxOccurs="unbounded" name="name" nillable="true" type="xsd:string"/>
     <element maxOccurs="unbounded" name="vit" nillable="true" type="xsd:string"/>
    </sequence>
   </complexType>
   <element name="initialAdvertiseVTResponse">
    <complexType>
     <sequence>
      <element name="initialAdvertiseVTReturn" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="advertiseVT">
...
...
   <element name="advertiseMembership">
    <complexType>
     <sequence>
      <element maxOccurs="unbounded" name="in0" type="xsd:string"/>
      <element name="in1" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="advertiseMembershipResponse">
    <complexType>
     <sequence>
      <element name="advertiseMembershipReturn" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="receiveMembership">
    <complexType>
     <sequence>
      <element name="in0" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
...
...
```

Fig. C.4: Trecho WSDL do AS.

Apêndice D

Políticas

D.1 Grupos de Políticas

- ▷ *Grupo de Políticas 1* (G1) : Este é o grupo de menor complexidade. Quando uma requisição é recebida pelo PM, a admissão é feita considerando apenas a proteção exigida. Por exemplo, se uma requisição 1+1 chegar ao PM, sua admissão poderá ser feita somente em algum caminho de luz com proteção 1+1 que tenha disponível a largura de banda exigida pela requisição. Esta analogia também é válida para a admissão de fluxos que exigem outros tipos de proteções;

- ▷ *Grupo de Políticas 2* (G2): O grupo G2 tem uma complexidade intermediária. Suas políticas estendem as políticas definidas no G1, considerando também a classe de serviço da requisição recebida como critério para a admissão. Este critério é usado somente na admissão de tráfego desprotegido. Ao receber uma requisição desprotegida, sua admissão é feita conforme a seguinte ordem de prioridade. Primeiro, busca-se $(P1)$ admiti-la em algum caminho de luz desprotegido que tenha somente fluxos com a mesma classe de serviço da requisição. Em seguida, busca-se $(P2)$ admiti-la em algum caminho de luz desprotegido que esteja vazio. Estas duas políticas procuram manter juntos os fluxos de tráfego de mesma classe de serviço; no entanto, quando os recursos da rede se tornam escassos, outras políticas são executadas. Para este caso, são definidas três políticas. Caso a requisição desprotegida ainda não tenha sido admitida, então, primeiramente, busca-se $(P3)$ admiti-la em algum caminho de luz desprotegido independente da classe de serviço dos fluxos já admitidos no caminho de luz. Depois, busca-se admiti-la $(P4)$ em algum caminho de luz desprotegido, preemptando, ou apenas removendo, fluxos de tráfego de mais baixa prioridade do caminho de luz. Finalmente, $(P5)$ busca-se admiti-la em algum caminho de luz protegido (exceto 1+1), preenchendo primeiro os caminhos de luz de *backup* e, posteriormente, os primários. Para esta política existem duas restrições: $(P5a)$ a requisição recebida deve ser desprotegida e $(P5b)$ ser de baixa prioridade (LP). Descritas estas cinco políticas para tráfego desprotegido, observa-se que a classe de serviço foi um critério bastante utilizado durante o processo de admissão de tráfego desprotegido. Porém, nas políticas definidas para tráfego protegido, a classe de serviço não é considerada como um critério para a admissão. Ao receber uma requisição protegida, o PM busca $(P6)$ admiti-la em algum caminho de luz primário que tenha a mesma proteção, independente da classe de serviço da requisição recebida. Depois, busca-se admiti-la $(P7)$ em

algum caminho de luz de mesma proteção, porém, preemptando, ou apenas removendo, fluxos de tráfego desprotegidos que sejam de baixa prioridade admitidos no caminho de luz;

- *Grupo de Políticas 3* (G3): Este grupo de políticas é o mais complexo, diferindo em dois pontos das políticas definidas no G2. O primeiro, é que se não houver recurso disponível com o mesmo nível de proteção requerido, então busca-se admiti-la em algum caminho de luz que tenha um nível de proteção maior do que o requerido. Este método é utilizado especificamente para 1:N e, como conseqüência, uma requisição 1:N pode ser instalada em um caminho de luz primário 1:1. Os caminhos de luz 1+1 são exclusivos para tráfego 1+1. A segunda diferença é que este grupo de políticas permite a quebra de um determinado grupo de caminhos de luz 1:N para atender requisições 1:1. Isto é, ao chegar uma requisição 1:1, o PM poderá admiti-la em um grupo 1:N que não esteja sendo utilizado. Para isso, é necessário fazer a quebra deste grupo, resultando em um grupo 1:1 e N-1 grupos de caminhos de luz desprotegidos;

  A quebra de grupos 1:N é feita somente na gerência, no entanto, é necessário ter algumas precauções durante o processo de recuperação de tráfego. Após a ocorrência de uma falha, não deve ser permitido ao plano de controle, realizar a recuperação de fluxos de tráfego admitidos em grupos de caminhos de luz que foram quebrados. Caso contrário, algum dos N-1 caminhos de luz desprotegidos poderia vir a receber a proteção 1:N que não mais o pertence.

D.2 Descrição de uma Política de Gerência de Serviço L1VPN usando XML

A Figura D.1 apresenta um exemplo de uma política em XML, considerando as propostas de classes do modelo de informações de políticas PCIM. Essa política define duas regras, uma de configuração e outra de admissão de controle. A primeira regra define três parâmetros de configuração para a "VPN A", como está estabelecido na condição da regra (linhas 7 e 8). Ela define o modelo de alocação de recursos (como dedicado), a classe de serviço e um mecanismo de recuperação a ser utilizado (linhas 11 a 13, respectivamente). A regra de admissão define um exemplo de restrição de conectividade. Neste caso, ela foi usada para impedir que dois membros específicos da VPN A possam estabelecer uma conexão entre si. A condição define quais são esses membros (linhas 21 e 23) e a ação determina que uma requisição entre esses membros deve ser rejeitada (linha 26).

```
<?xml version='1.0'?>
<!DOCTYPE Policy SYSTEM ''l1vpnPolicy.dtd''>
<Policy id=''001''>
  <PolicySet>
    <PolicyRule type=''configuration''>
      <PolicyCondition>
        <VPNServiceIDVariable/>
        <PolicyValue>vpnA</PolicyValue>
      </PolicyCondition>
      <PolicyAction>
        <ResourceAllocationAction allocationModel=''dedicated''/>
        <CoSAction model=''basic'' class=''gold''/>
        <RecoveryAction recoveryScheme=''protection:1+1''/>
      </PolicyAction>
    </PolicyRule>
    <PolicyRule type=''admissionControl''>
      <PolicyCondition>
        <VPNServiceIDVariable/>
        <PolicyValue>vpnA</PolicyValue>
        <IngressMemberIDVariable/>
        <PolicyValue>CPI1-PPI1</PolicyValue>
        <EgressMemberIDVariable/>
        <PolicyValue>CPI2-PPI2</PolicyValue>
      </PolicyCondition>
      <PolicyAction>
        <RejectAction/>
      </PolicyAction>
    </PolicyRule>
  </PolicySet>
</Policy>
```

Fig. D.1: Exemplo de política em XML.

Apêndice E

Topologia Usada nos Testes

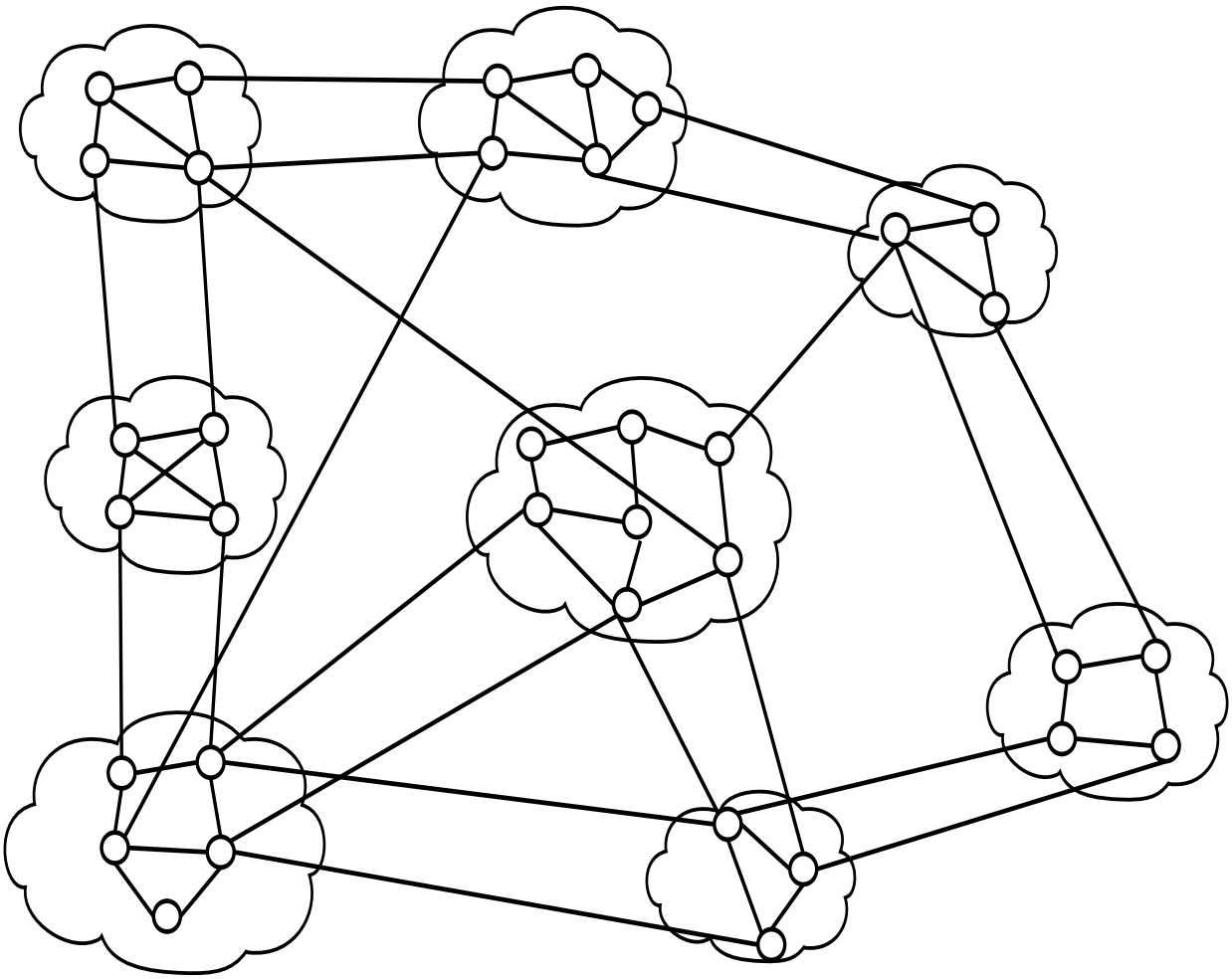


Fig. E.1: Topologia de oito domínios usada nos testes.