



Universidade Estadual de Campinas  
Instituto de Computação



Rafael Lopes Gomes

Mecanismos para Gerenciamento de SLA em Redes  
Virtuais Definidas por Software baseados em Classes de  
QoS

CAMPINAS  
2015



Universidade Estadual de Campinas  
Instituto de Computação



**Rafael Lopes Gomes**

**Mecanismos para Gerenciamento de SLA em Redes Virtuais  
Definidas por Software baseados em Classes de QoS**

Tese apresentada ao Instituto de Computação da  
Universidade Estadual de Campinas como parte  
dos requisitos para a obtenção do título de Dou-  
tor em Ciência da Computação.

**Orientador: Prof. Dr. Edmundo Roberto Mauro Madeira**  
**Coorientador: Prof. Dr. Luiz Fernando Bittencourt**

Este exemplar corresponde à versão final da  
Tese defendida por Rafael Lopes Gomes e ori-  
entada pelo Prof. Dr. Edmundo Roberto  
Mauro Madeira.

CAMPINAS  
2015

**Agência(s) de fomento e nº(s) de processo(s):** FAPESP, 2012/04945-7

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca do Instituto de Matemática, Estatística e Computação Científica  
Maria Fabiana Bezerra Muller - CRB 8/6162

G585m Gomes, Rafael Lopes, 1987-  
Mecanismos para gerenciamento de SLA em redes virtuais definidas por software baseados em classes de QoS / Rafael Lopes Gomes. – Campinas, SP : [s.n.], 2015.

Orientador: Edmundo Roberto Mauro Madeira.

Coorientador: Luiz Fernando Bittencourt.

Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Redes de computadores. 2. Virtualização de redes. 3. Qualidade de serviço (Redes de computadores). I. Madeira, Edmundo Roberto Mauro, 1958-. II. Bittencourt, Luiz Fernando, 1981-. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

#### Informações para Biblioteca Digital

**Título em outro idioma:** Mechanisms for management of SLA in virtual software defined networks based on QoS classes

**Palavras-chave em inglês:**

Computer networks

Network virtualization

Quality of service (Computer networks)

**Área de concentração:** Ciência da Computação

**Titulação:** Doutor em Ciência da Computação

**Banca examinadora:**

Edmundo Roberto Mauro Madeira [Orientador]

Lisandro Zambenedetti Granville

Fabio Luciano Verdi

Islene Calciolari Garcia

Christian Rodolfo Esteve Rothenberg

**Data de defesa:** 23-10-2015

**Programa de Pós-Graduação:** Ciência da Computação



Universidade Estadual de Campinas  
Instituto de Computação



Rafael Lopes Gomes

## Mecanismos para Gerenciamento de SLA em Redes Virtuais Definidas por Software baseados em Classes de QoS

### Banca Examinadora:

- Profa. Dra. Islene Calciolari Garcia  
Instituto de Computação - UNICAMP
- Prof. Dr. Christian Esteve Rothenberg  
Faculdade de Engenharia Elétrica e Computação - UNICAMP
- Prof. Dr. Lisandro Zambenedetti Granville  
Instituto de Informática - UFRGS
- Prof. Dr. Fábio Luciano Verdi  
Centro de Ciências em Gestão e Tecnologia - UFSCAR
- Prof. Dr. Nelson Luis Saldanha da Fonseca  
Instituto de Computação - UNICAMP (*Suplente*)
- Profa. Dra. Juliana Freitag Borin  
Instituto de Computação - UNICAMP (*Suplente*)
- Prof. Dr. Elias Procópio Duarte Jr.  
Departamento de Informática - UFPR (*Suplente*)

A ata da defesa, onde constam as assinaturas dos membros da banca, está arquivada pela Universidade Estadual de Campinas.

# Agradecimentos

Quero agradecer a todas as pessoas que se fizeram presentes, que se preocuparam, que foram solidárias, que torceram por mim. Sei que agradecer é sempre difícil, podendo cometer certos erros esquecendo de pessoas que me ajudaram. Adicionalmente, gostaria de agradecer à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP - Processo 2012/04945-7) por ter financiado meu projeto de doutorado, bem como a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela oportunidade do doutorado sanduíche.

Primeiramente, gostaria de agradecer ao meu orientador, o professor Edmundo Madeira, pela sua dedicação e compreensão em muitos momentos de dúvidas e erros. Da mesma forma, agradeço ao meu co-orientador Luiz Fernando Bitencourt pela dedicação e ajuda prestada ao decorrer desses anos. Além deles, gostaria de agradecer aos professores Mario Gerla e Eduardo Cerqueira, pela oportunidade de realizar o doutorado sanduíche, bem como pela orientação e cooperação durante o período e que se faz durar até hoje.

Agradeço também a todos os meus amigos que fizeram parte dessa caminhada. Aos meus amigos do Laboratório de Redes (LRC) pela paciência e cooperação nestes anos que passei como aluno. Agradeço à equipe de natação da UNICAMP (USSR) e a todos seus membros, obrigado pela oportunidade de poder voltar a nadar, com certeza os colegas da equipe ajudaram muito para poder levar até o fim o meu doutorado. Dos membros da equipe, não poderia deixar de dar um destaque as magrinhas Maysa e Bruna que me ajudaram quando mais precisei, muito obrigado pelos favores e o constante apoio.

Minha família merece poucas palavras, mas aquelas que me são mais caras. Meu pai Daniel, minha Mãe Rosângela, minha irmã Daniela, e aos demais (meus primos Zé e Carol, minha vó Carmen, minha tia Minda, e os demais), obrigado por vocês existirem. Obrigado por depositarem em mim a confiança para todas as horas. Sei que vocês se orgulham por eu ter passado por mais uma etapa na minha vida. Mas este orgulho que sentem por mim, converto numa obrigação de a cada dia ser mais digno de os representar.

Por fim gostaria de agradecer a minha namorada, Bruna Porto, por toda a paciência e compreensão nos momentos de raiva, ansiedade e dúvida. Muito obrigado por sempre me apoiar, você sempre esteve comigo nos momentos difíceis.

# Resumo

Ao longo dos anos a Internet vem se tornando o principal meio de comunicação, onde os usuários esperam ter acesso à Internet em todos os lugares e o tempo todo com qualidade. Como consequência, nos últimos anos aumentou a demanda por recursos para acesso à Internet. Entretanto, visto que a Internet atual não provê garantias de Qualidade de Serviço (*Quality of Service* – QoS), as empresas realizam um Acordo de Nível de Serviços (*Service Level Agreement* – SLA) com os provedores de Internet. A partir disso, os provedores de Internet visam melhorar o uso dos recursos de rede e a prestação de serviço aos usuários através da implantação de redes virtuais (*Virtual Networks* - VNs) sobre redes definidas por software (*Software Defined Network* - SDN). Contudo, muitos aspectos ligados à gerência deste tipo de ambiente são desafios em aberto, e devido a isso, esta tese apresenta mecanismos para o gerenciamento de redes virtuais definidas por software. A ideia principal destes mecanismos é negociar, implantar e adaptar as redes virtuais de acordo com o estado corrente da infraestrutura de rede SDN e das especificações no SLA.

# Abstract

Over the years the Internet has become the primary means of communication, where the users expect to access the Internet anytime, anywhere, and with a certain quality level. As a consequence, in the past few years the traffic demand to Internet access has increased. However, since the current Internet does not guarantee Quality of Service (QoS), the companies apply a Service Level Agreements (SLA) with Internet service providers. In this way, the Internet service providers aim to improve utilization of network resources and service delivery to the users through the deployment of Virtual Networks (VNs) over Software Defined Networks (SDNs). Nevertheless, some aspects related to the management of this type of environment still present open issues. In this context, this thesis presents mechanisms to manage virtual software defined networks. The objectives of the developed mechanisms are to negotiate, to deploy and to adapt the virtual software defined networks according to the current state of the network infrastructure and the SLA definition.

# Lista de Figuras

2.1	Redes Definidas por Software. . . . .	17
2.2	Funcionamento do Mininet. . . . .	18
2.3	Cenário representando o contexto desta tese. . . . .	19
3.1	Visão Geral da Arquitetura Proposta . . . . .	23
3.2	Passos realizados pela arquitetura . . . . .	28
3.3	Negociação do SLA . . . . .	28
3.4	Implantação da rede virtual . . . . .	29
3.5	Adaptação da rede virtual . . . . .	30
4.1	<i>Accuracy</i> dos Classificadores . . . . .	33
4.2	<i>Precision</i> dos Classificadores . . . . .	33
4.3	Tempo para classificação . . . . .	34
5.1	Diagrama que representa o modelo de especificação de SLA . . . . .	36
5.2	Diagrama de Sequência . . . . .	37
5.3	Comportamento da métrica $S_{VN}$ . . . . .	42
5.4	Função de adesão <i>Avaliação</i> . . . . .	45
5.5	Resultados . . . . .	48
6.1	Divisão das regiões . . . . .	60
6.2	Topologia da rede Internet2. . . . .	62
6.3	Alocações realizadas com sucesso. . . . .	63
6.4	Largura de banda disponível. . . . .	64
6.5	Eficiência energética da rede. . . . .	64
6.6	Estado de comunicação pós-falha. . . . .	65
7.1	Relação entre <i>bitrate</i> e QoE. . . . .	69
7.2	Aspectos de Custo Financeiro . . . . .	72
7.3	Desperdício de recursos . . . . .	73
7.4	Valores de SSIM . . . . .	74
7.5	Resultado da Pontuação MOS . . . . .	74
7.6	Frames do vídeo Puskas-2013. . . . .	75
7.7	Configuração do Testbed . . . . .	76
7.8	Diferença entre alocação e demanda . . . . .	76
7.9	Perda Média . . . . .	77

# Lista de Tabelas

2.1	Trabalhos Relacionados . . . . .	21
4.1	Matriz de Confusão do classificador Naive Bayes . . . . .	32
4.2	Matriz de Confusão do classificador <i>Decision Tree</i> . . . . .	32
5.1	OTU para Dados Binários [19] . . . . .	40
5.2	Conjunto de Regras . . . . .	46
5.3	Grau de Importância . . . . .	47
5.4	Similaridade entre os Protocolos de Roteamento . . . . .	47
6.1	Notação Utilizada . . . . .	52
6.2	Conjunto de Regras do Modelo de Risco . . . . .	58
6.3	Informações sobre o Modelo de Risco . . . . .	59
7.1	Informações sobre os Videos . . . . .	69

# Sumário

<b>1</b>	<b>Introdução</b>	<b>12</b>
<b>2</b>	<b>Fundamentação Teórica</b>	<b>15</b>
2.1	Acordos de Nível de Serviços (SLA)	15
2.2	SDN e OpenFlow	16
2.3	Virtualização de Redes e Network Hypervisors	18
2.4	Contexto	19
2.5	Trabalhos Relacionados	20
<b>3</b>	<b>Arquitetura Proposta</b>	<b>22</b>
3.1	Descrição da arquitetura e seus módulos	23
3.1.1	Central Manager	24
3.1.2	SLA Analyzer	24
3.1.3	Virtual Network Deployer	25
3.1.4	Allocation Processer	25
3.1.5	Infrastructure Manager	25
3.1.6	Resource Adjuster	26
3.1.7	Traffic Classifier	26
3.1.8	Passive Tracker	26
3.1.9	Resource Allocator	27
3.2	Interação entre os Módulos da Arquitetura	27
3.2.1	Negociação do SLA	27
3.2.2	Implantação da rede virtual	28
3.2.3	Adaptação da rede virtual	29
<b>4</b>	<b>Classificação de Tráfego</b>	<b>31</b>
<b>5</b>	<b>Negociação de SLA</b>	<b>35</b>
5.1	Negociação de SLA para Redes Virtuais	35
5.1.1	Modelo de Especificação de SLA	35
5.1.2	Protocolo de Negociação Desenvolvido	37
5.2	Modelo de Similaridade para Negociação de Redes Virtuais	38
5.2.1	Modelagem de Similaridade Proposta	39
5.2.2	Medição de Similaridade	40
5.3	Mecanismo de Apoio à Negociação de SLA	43
5.3.1	Fuzzificação	44
5.3.2	Sistema de Inferência	45
5.3.3	Defuzzificação	45

5.4	Estudo de Caso . . . . .	46
5.4.1	Questionário . . . . .	46
5.4.2	Resultados . . . . .	47
<b>6</b>	<b>Alocação de Redes Virtuais Definidas por Software</b>	<b>49</b>
6.1	Contexto . . . . .	49
6.2	Confiabilidade da Rede . . . . .	50
6.3	Algoritmo Base para Alocação . . . . .	51
6.4	Algoritmos de Definição de Caminho . . . . .	55
6.4.1	Weighted Path . . . . .	56
6.4.2	Maximum Available Bandwidth . . . . .	57
6.4.3	Feasible-Bw . . . . .	57
6.4.4	Bw-Risk-Ratio . . . . .	57
6.4.5	Energy-Aware . . . . .	59
6.4.6	BEE-Focus . . . . .	60
6.4.7	DA-BEE Algorithm . . . . .	61
6.5	Experimentos . . . . .	62
<b>7</b>	<b>Adaptação de Redes Virtuais Definidas por Software</b>	<b>66</b>
7.1	Contexto . . . . .	66
7.2	RAAND: Resource Adjustment According to Network Demand . . . . .	67
7.3	BEAVER: Bitrate and Experience Aware adaptation of Virtual Edge Re- sources . . . . .	68
7.4	Experimentos . . . . .	71
7.4.1	Ambiente de Emulação . . . . .	71
7.4.2	Testbed . . . . .	75
<b>8</b>	<b>Conclusão</b>	<b>78</b>
	<b>Referências Bibliográficas</b>	<b>80</b>

# Capítulo 1

## Introdução

Na última década, a sociedade vem evoluindo seu tradicional paradigma de comunicação baseado em chamadas de voz e mensagens de texto, agregando mais aplicações de interação multimídia como chamadas de vídeo tempo real e/ou compartilhando informações em redes sociais através da Internet. Este fato criou um novo paradigma onde espera-se ter acesso à Internet em todo os lugares e o tempo todo, com um nível de Qualidade de Serviço (em inglês *Quality of Service* - QoS) desejado [64]. Por exemplo, nos últimos anos aumentou a demanda por recursos para acesso à Internet ao redor do mundo, onde constata-se que o conteúdo multimídia acessado pelos usuários representa 55% do tráfego da Internet e estima-se uma expansão para 92% até 2020 [4]. Adicionalmente, tem-se que a demanda de tráfego muda ao decorrer do dia devido à mobilidade dos usuários [71], onde surge um cenário de demanda elástica para os recursos de rede.

Este novo paradigma gera um conjunto de requisitos a serem atendidos pelos provedores de Internet (em inglês *Internet Service Providers* - ISPs), tornando um desafio garantir QoS aos usuários no que se refere ao acesso à Internet. Os ISPs, em sua maioria, fornecem serviços através de Acordos de Nível de Serviços (em inglês *Service Level Agreement* - SLA), que são usados como uma base contratual para definir certas propriedades funcionais e não-funcionais (por exemplo, tempo de resposta, taxa de perda e outros) a serem atendidas pelo ISP a fim de garantir QoS ao cliente.

Em geral, os usuários ficam frustrados quando o acesso à Internet sofre de problemas como lentidão, interrupção de serviço, desconexões constantes, etc [1], visto que muitas aplicações necessitam de requisitos de rede diferenciados, como por exemplo: largura de banda (em inglês *Bandwidth* - Bw), confiabilidade da rede, etc. Portanto, é preciso incorporar aspectos de resiliência aos ISPs, ou seja, deve-se ser capaz de prover um nível de qualidade mínimo especificado em casos de sobrecarga de tráfego e falhas na infraestrutura de rede [75], e desta forma atender as especificações definidas no SLA vigente com o cliente.

Adicionalmente, cada SLA envolve um custo financeiro ao cliente relacionado à prestação de serviço do ISPs. Portanto, além de atender os requisitos do SLA, os ISPs visam maximizar os lucros. Dois aspectos são relacionados ao lucro de um ISP [17, 68]: número de SLAs vigentes (clientes) e o consumo de energia da infraestrutura de rede. Sendo assim, o ISP deve fazer um melhor uso dos recursos da infraestrutura de rede para poder aumentar o número de SLAs, e conseqüentemente o seu lucro [6]. Por outro lado, o consumo de energia tornou-se um aspecto com influencia direta no custo de operação dos

ISPs [11]. O consumo de energia está relacionado ao uso eficiente da infraestrutura de rede, ou seja, a utilização dos equipamentos de rede somente em caso de necessidade.

A fim de aprimorar o acesso à Internet e o uso dos recursos de rede, os ISPs tendem a aplicar os princípios de virtualização de redes (em inglês *Network Virtualization* - NV) e redes definidas por software (em inglês *Software Defined Network* - SDN) para usar os recursos da infraestrutura de rede com maior flexibilidade e tornar o comportamento das redes personalizável [42, 67, 20, 10]. A junção das abordagens SDN e NV ocorre através de um *Network Hypervisor*, o qual permite a divisão da infraestrutura de redes em camadas, onde cada camada é uma rede virtual (em inglês *Virtual Network* - VN) customizável, possuindo um conjunto particular de recursos e protocolos. Uma rede virtual sobre uma rede definida por software pode ser chamada de rede virtual definida por software (em inglês *Virtual Software Defined Network* - VSDN). Portanto, durante esta tese, o termo “rede virtual” é equivalente ao termo “rede virtual definida por software”.

A aplicação de SDNs habilita as seguintes vantagens para os aspectos de gerenciamento [43]: (i) é mais fácil modificar o comportamento da rede; e (ii) a visão centralizada permite um melhor conhecimento sobre a situação da infraestrutura de rede. Além disso, a implantação de redes virtuais habilita o isolamento e customização do comportamento das redes, visto que tal comportamento depende da configuração do controlador. Portanto, é possível definir o comportamento desejado de uma rede virtual específica sem alterar as demais redes virtuais.

A comunidade científica tem feito muitos esforços e desenvolvido propostas para melhorar a aplicabilidade e implantação dos conceitos de SDN e NV, onde os mesmos possuem uma forte tendência a serem aplicados nas redes 5G [18, 40, 15, 84, 42, 67, 20, 10]. Entretanto, muitos aspectos ligados a estes ambientes são desafios em aberto, visto que diversos tipos de acontecimento podem ocorrer (por exemplo, falhas, sobrecarga de tráfego, manutenção de equipamentos, alterações nas tarifas de cobrança, etc), afetando a QoS vivenciada pelos usuários, os SLAs definidos e os recursos disponíveis do ISP[20]. Desta forma, a negociação, implantação e gerenciamento de redes virtuais deve considerar ações reativas de acordo com o estado da rede, bem como planejamento estratégico.

Dentro deste contexto, esta tese de doutorado apresenta mecanismos para gerenciamento de redes virtuais definidas por software, englobando a implantação da rede virtual definida por software como um todo, ou seja, os mecanismos habilitam desde a negociação do SLA para a rede virtual, passam pela implantação da rede virtual na infraestrutura SDN e, por final, gerenciam os aspectos pós-implantação.

Sendo assim, os mecanismos propostos permitem a realização das seguintes tarefas: (i) negociação dos parâmetros da rede virtual no SLA, especificando os requisitos desejados pelo cliente; (ii) identificação da classe de tráfego oriunda do cliente, visando atender melhor suas particularidades; (iii) alocação da rede virtual, considerando os parâmetros definidos no SLA, bem como as informações dos recursos na infraestrutura de rede; e (iv) adaptação da rede virtual caso identifique-se a necessidade, visando atender os parâmetros especificados no SLA e/ou melhorar a utilização dos recursos de rede.

Os mecanismos propostos têm como objetivos: (i) proporcionar ao cliente a capacidade de definir os parâmetros, bem como identificar os mais adequados, da rede virtual no SLA; (ii) garantir que o ISP atenda os parâmetros especificados no SLA e a qualidade do

serviço experimentada pelos usuários; e (iii) melhorar a utilização dos recursos de rede e a eficiência energética do ISP.

A fim de avaliar os mecanismos propostos foram realizados experimentos utilizando: (a) *web service* para negociação de SLA; (b) simulador de alocação de redes virtuais; (c) emulador de redes definidas por software junto com um *network hypervisor*; e (d) experimentação real em um *testbed* na Universidade da Califórnia Los Angeles (UCLA). Os resultados obtidos nos experimentos realizados sugerem a eficiência dos mecanismos propostos, bem como cada contribuição individual deste projeto de acordo com seus objetivos específicos.

De maneira geral, as contribuições deste trabalho foram:

- O projeto de uma arquitetura para interconexão dos mecanismos para gerenciamento de redes virtuais definidas por software;
- Um classificador de tráfego baseado em classes de QoS;
- Um protocolo de negociação de SLA para redes virtuais;
- Um mecanismo de apoio à negociação de redes virtuais;
- Algoritmos de alocação de redes virtuais;
- Dois mecanismos de ajuste de redes virtuais; e
- A avaliação dos mecanismos propostos.

O restante do trabalho está organizado da seguinte maneira: a Seção 2 mostra os aspectos relacionados aos conceitos de SDN e NV, os principais projetos e tecnologias para prover suporte à mesma, bem como trabalhos relacionados a esta tese; a Seção 3 descreve a arquitetura projetada para interconexão dos mecanismos propostos, detalhando os módulos projetados bem como a interação entre eles; a Seção 4 trata da proposta de classificação de tráfego desenvolvida, mostrando desde os aspectos gerais de até os detalhes do seu desenvolvimento; a Seção 5 descreve o protocolo de negociação de SLA desenvolvido para a negociação de redes virtuais e o mecanismo de apoio à negociação; a Seção 6 apresenta os algoritmos de alocação desenvolvidos; a Seção 7 descreve os dois mecanismos de ajuste de redes virtuais propostos; e finalmente, na Seção 8 conclui-se o trabalho e mostra-se alguns trabalhos futuros.

# Capítulo 2

## Fundamentação Teórica

Esta seção apresenta os conceitos base para o contexto deste trabalho, ou seja, descreve as tecnologias SDN e NV, as quais são aplicadas no mesmo, bem como os aspectos relacionados aos Acordos de Nível de Serviço (SLA). Além disso, são mostrados alguns trabalhos relacionados ao tema desta tese.

### 2.1 Acordos de Nível de Serviços (SLA)

O setor de informática cada vez mais influencia a capacidade das empresas de serem competitivas no mercado, onde ocorrem mudanças contínuas em suas condições. Ao longo dos anos os serviços e funções de muitas organizações se tornaram dependentes da infraestrutura provida pela área de informática.

Assim como as empresas, os serviços prestados evoluíram. Muitos desses novos serviços são provisionados através do uso da Internet, necessitando assim de um maior uso da infra-estrutura de rede das organizações ou empresas em geral. Exemplos desses serviços são: *Voice Over IP* (VoIP), vídeo sobre demanda, transferência de dados, entre outros.

SLA é um contrato, entre um provedor de serviço (*Service Provider* - SP) e um cliente, que especifica, normalmente em termos mensuráveis, quais serviços o SP irá prover e as ações que o mesmo cumprirá se o serviço prestado não for compatível com os objetivos estabelecidos no contrato firmado. Para a definição de um SLA são necessários alguns aspectos relacionados ao gerenciamento, especificação e negociação do mesmo.

O gerenciamento de SLA irá garantir ao cliente a confiabilidade do serviço prestado, através de estatísticas desse serviço. Por exemplo, quando certo contrato delimita aspectos de um serviço de rede, geralmente são medidas estatísticas referentes à vazão, atraso e outras métricas de redes comumente usadas.

SLA é um acordo negociável entre partes, por exemplo um provedor e seu cliente. Quando o cliente requisita um serviço ao provedor, um SLA é negociado e um contrato é feito. Negociação é um processo de decisão no qual duas ou mais partes realizam decisões e interagem umas com as outras para um ganho mútuo [39].

O processo de negociação pode ser feito automaticamente ou diretamente pelas partes interessadas. A negociação efetuada entre pessoas possui algumas questões desinteressantes como: alto tempo para tomada de decisão, problemas culturais, ego e outros aspectos.

Então, uma abordagem mais sofisticada para o modelo de negócio é necessária [39].

No caso de uma negociação automatizada, as pessoas são substituídas por negociadores automatizados que tentam alcançar o objetivo definido em sua configuração. A negociação automatizada é importante quando o cliente de um certo serviço é um sistema que negocia o SLA em tempo real [62].

A negociação automatizada de SLA é um processo complexo e que consome um certo tempo, até mesmo quando dois usuários têm que encontrar uma solução baseada em vários critérios [9]. Quando pelo menos dois recursos são levados em consideração ao mesmo tempo, algumas etapas precisam ser realizadas antes de se alcançar um acordo entre o provedor dos recursos e o cliente. Quando se trata de SLAs relacionados a recursos de redes, a negociação automatizada tem três principais aspectos: o protocolo de negociação, os objetos negociados e o modelo de tomada de decisão.

Desta forma, na negociação automatizada, os negociadores possuem um método de tomada de decisão, ou seja, um algoritmo para decidir a ação a ser tomada quando é recebida uma oferta na fase da negociação. O modelo de decisão em um processo de negociação implica em [62]: definir os limites dos atributos negociados (restrições); identificar os objetivos buscados; e definir a prioridade (se houver) de cada objetivo avaliando o *trade-off* entre eles.

## 2.2 SDN e OpenFlow

O SDN é uma proposta tecnológica que separa os planos de controle e de dados em comutadores de pacotes (em inglês *switches*) para reduzir a carga dos componentes de rede e habilitar um melhor uso dos recursos de rede [79]. A proposta é fundamentada em comutadores *Ethernet* comerciais e define um protocolo padrão para controlar o estado destes comutadores. O conceito de fluxo habilita a definição do plano de encaminhamento na rede, conforme os objetivos definidos pelas novas propostas de arquiteturas e protocolos de rede. Adicionalmente, SDN também define um novo elemento de rede, o controlador, o qual contém um software de controle executando nele. A protocolo mais popular para a comunicação entre os comutadores e o controlador é o Openflow [55].

O OpenFlow é uma proposta tecnológica que, baseada na separação dos planos de controle e de dados em comutadores de pacotes, permite que pesquisadores executem seus experimentos em redes utilizadas no dia-a-dia, sem interferir no tráfego de produção. O conceito de fluxo é o bloco fundamental que habilita aos pesquisadores definir o plano de encaminhamento na rede, conforme os objetivos definidos pelas novas propostas de arquiteturas e protocolos de rede.

O OpenFlow explora as tabelas de fluxos que já existem nos equipamentos atuais, e normalmente são utilizadas para implementar serviços como *Network Address Translation* (NAT), *firewall* e *Virtual Local Area Networks* (VLANs). Um comutador OpenFlow possui uma tabela de fluxos e um evento associado a cada entrada na tabela. Basicamente, a arquitetura do OpenFlow é composta por três partes:

- Tabela de Fluxos: Cada entrada na tabela de fluxos contém uma ação associada, e consiste em campos do cabeçalho (utilizado para definir um fluxo), ações (define

como os pacotes devem ser processados) e contadores (utilizados para estatísticas e remoção de fluxos inativos).

- Canal Seguro: Para que a rede não sofra ataques de elementos mal intencionados, o canal seguro garante confiabilidade na troca de informações entre o comutador e o controlador.
- Protocolo OpenFlow: Disponibiliza um protocolo aberto para estabelecer a comunicação entre o comutador e o controlador, fornecendo uma interface externa que atue sobre os fluxos de um comutador, o protocolo OpenFlow (OFP - *OpenFlow Protocol*) evita a necessidade de um comutador programável.

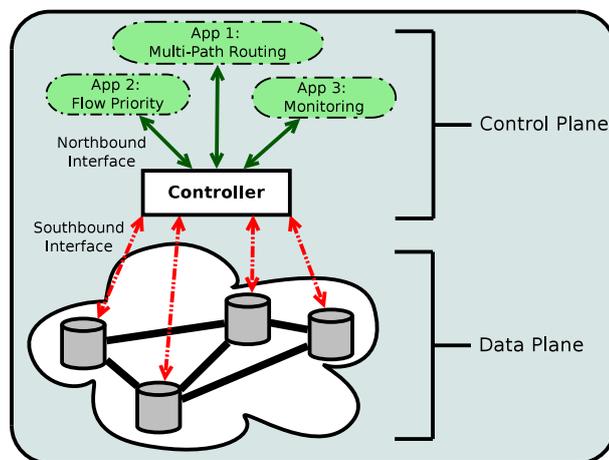


Figura 2.1: Redes Definidas por Software.

Uma visão da organização da arquitetura SDN é apresentada na Figura 2.1. *Southbound* é a interface de comunicação entre os equipamentos da infraestrutura de rede com o controlador, o maior exemplo deste tipo de interface é o protocolo OpenFlow [55]. Por outro lado, *Northbound* é a interface de programação entre o controlador e as aplicações de rede, um exemplo deste tipo de interface é a REST API [85].

A criação de redes SDN pode ocorrer em diversos ambientes [77]: simulação, emulação ou experimentação real. O simulador SDN mais utilizado é o EstiNet [78], o qual usa uma metodologia de simulação de reentrada no *kernel* para habilitar que aplicações executem em redes SDN simuladas. Por outro lado, o Mininet [47] é um emulador que cria redes Openflow escaláveis (até centenas de nós, dependendo da configuração) em um único computador usando distintos processos. Assim, o Mininet permite ao usuário rapidamente criar, interagir, personalizar e compartilhar um protótipo de SDN, e fornece um ambiente com comportamento similar ao real. Uma visão geral do funcionamento do emulador Mininet é mostrada na Figura 2.2.

O uso de testbeds para realizar experimentos em ambientes reais pode ser feito através de *switches* que permitem a habilitação do protocolo Openflow, por exemplo Pica8<sup>1</sup> ou NEC<sup>2</sup>. Esta abordagem visa combinar o comportamento tradicional dos *switches* e as

<sup>1</sup>[www.pica8.com](http://www.pica8.com)

<sup>2</sup>[www.necam.com/SDN/](http://www.necam.com/SDN/)

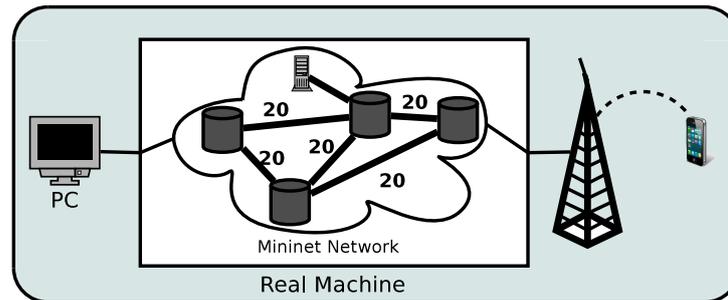


Figura 2.2: Funcionamento do Mininet.

soluções SDN, portanto, pode-se habilitar que certas portas do *switch* se comportem como SDN, ou seja, interajam com o controlador. Apesar de ser a abordagem que mais representa ambientes SDN, a implantação de redes utilizando esses equipamentos possui um alto custo. Sendo assim, trabalhos existentes [59, 44] comprovam a viabilidade de utilizar equipamentos de redes de baixo custo, como NetFPGA<sup>3</sup> e Raspberry-Pi<sup>4</sup>, para desenvolver redes SDN.

### 2.3 Virtualização de Redes e Network Hypervisors

Similar à virtualização de computadores, a virtualização de redes promete melhorar a alocação de recursos, permitindo o compartilhamento dos mesmos equipamentos de forma controlada e isolada. Portanto, analogamente, a rede em si deve ter uma camada de abstração de hardware, similar ao que acontece na virtualização de computadores, chamada de *Network Hypervisor*. Esta camada deve ser facilmente particionada, para que múltiplas redes completamente diferentes possam ser executadas simultaneamente, sem interferir umas com as outras. Ou seja, acima desta camada de abstração de hardware, têm-se novos protocolos e formatos de endereçamento rodando independentemente e sua própria “fatia” de rede. No contexto de redes SDN, existem três abordagens mais populares para a implantação de redes virtuais: Flowvisor [72], VeRTIGO [22] e OpenVirtex [5].

O FlowVisor é um controlador especializado que atua como um *proxy* transparente entre os comutadores de uma rede OpenFlow e seus múltiplos controladores [72]. Todas as mensagens do protocolo OpenFlow são interceptadas através do FlowVisor, assim, os controladores não necessitam de modificações. Cada fatia está vinculada a um controlador, onde o FlowVisor define uma fatia como um conjunto de fluxos, possibilitando o mapeamento entre as fatias criadas e os fluxos passantes na rede. As características como a virtualização transparente, o isolamento entre as fatias e a política de definição de *flowspace*s tornam o FlowVisor uma ferramenta interessante no que diz respeito à virtualização e implementação de redes programáveis orientadas a software. Portanto, a integração FlowVisor e OpenFlow permite que em uma rede OpenFlow possam ser criadas várias fatias de recursos executando simultaneamente e isoladamente.

<sup>3</sup>netfpga.org

<sup>4</sup>www.raspberrypi.org

VeRTIGO é uma extensão do Flowvisor que permite a definição de topologias virtuais [22], ou seja, ele permite a definição de enlaces virtuais a partir do mapeamento de caminhos na infraestrutura de rede. Contudo, assim como o Flowvisor, o VeRTIGO não possui a capacidade de prover o isolamento de endereçamento de rede.

Recentemente, o OpenVirtex (OVX) foi proposto a fim de suprir os aspectos ainda não tratados pelo Flowvisor e VeRTIGO [5]. OVX é um Network Hypervisor que cria múltiplas redes virtuais com um espaço de endereçamento completo, especificando a topologia de rede desejada em tempo real.

## 2.4 Contexto

A integração das abordagens SDN e VN visam trazer maior flexibilidade e isolamento para as redes. Um network hypervisor é utilizado para habilitar essa integração, onde cada rede virtual é customizável a partir da configuração do controlador ligado à rede virtual, onde um controlador é responsável por uma rede virtual.

Portanto, no contexto de redes virtuais definidas por software (VSDN) surge a capacidade de customizar os parâmetros das redes e os serviços providos pelo ISP para o cliente. Trabalhos existentes na literatura [26, 66, 81] usam esta flexibilidade para trazer ao cliente o serviço desejado, adaptando a qualidade do serviço de acordo com os requisitos dados pelo cliente. Consequentemente, o cliente paga um preço consistente com a qualidade do serviço prestado.

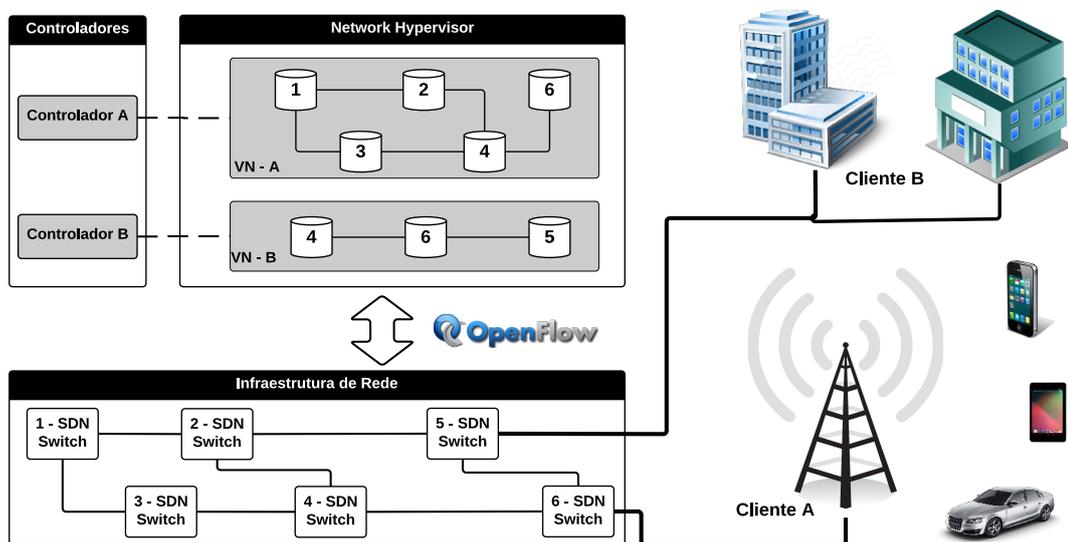


Figura 2.3: Cenário representando o contexto desta tese.

A Figura 2.3 ilustra um cenário que representa o contexto abordado por esta tese. O cliente e o ISP definem um SLA (especificação da rede virtual), onde o tráfego de cada cliente é atrelado a uma rede virtual que melhor molda seus requisitos sob o ponto de vista de recursos alocados e de comportamento da rede (por exemplo, controle de admissão de fluxo, prioridade de fluxo, descarte seletivo, dentre outros).

Um exemplo de implantação é mostrado na Figura 2.3: pode-se implantar a VN-A

vinculada ao controlador A; por outro lado, o IPS pode implantar de forma independente a *VN-B*, a qual seria ligada ao controlador B. Desta maneira, os ISPs podem direcionar o tráfego de cada cliente a uma rede virtual específica, e/ou diferenciar o tráfego passante a fim de customizar uma rede virtual a cada classe de tráfego definida (por exemplo, de dados, multimídia, etc).

## 2.5 Trabalhos Relacionados

Esta seção apresenta alguns trabalhos relacionados a estratégias de gerenciamento de ambiente com SDN, VNs, e redes de acesso sem fio. A Tabela 2.1 resume estes trabalhos, destacando as particularidades de cada proposta analisada.

Zhang et al. [83] mostram um esquema adaptativo de alocação de largura de banda para redes virtuais que periodicamente realiza ajustes na alocação corrente de acordo com a perspectiva do provedor, independente do SLA vigente com o cliente.

Carvalho et al. [13] propõem o *SLAPv*, um sistema de controle de SLA para ambientes com redes virtuais, o qual pune as redes virtuais que não respeitem o contrato firmado entre cliente e provedor. O sistema de controle é baseado em lógica fuzzy a fim de reforçar a alocação de recursos de acordo com o uso dos roteadores da rede e a carga do sistema.

Georgopoulos et al. [24] introduzem o *OpenFlow-assisted QoE Fairness Framework* (QFF) para o monitoramento de transmissões de vídeo em tempo real. Os autores definem um plano de controle para o cliente que ajusta as características dos fluxos de vídeo de acordo com as informações fornecidas pelo controlador da rede SDN. A proposta visa assegurar uma Qualidade de Experiência (*Quality of Experience* - QoE) equivalente e justa aos fluxos de vídeo.

Skoldstrom and Yedavalli [73] apresentam um sistema unificado para a virtualização sobre redes baseadas no protocolo Openflow. O sistema proposto pelos autores possui um mecanismo para assegurar a alocação de recursos dentre as diferentes partes da infraestrutura de rede.

Bueno et al. [12] propõem a adição de uma camada de controle para infraestruturas de redes heterogêneas baseadas em um paradigma *Network as a Service* sob SDN. Este trabalho realiza um ajuste nos recursos de rede orientado a fluxos a partir de mudanças nos requisitos especificados pelas aplicações.

Zhang et al. [82] utilizam as características das redes SDN (separar os planos de controle e encaminhamento) para concentrar funções de eficiência energética no controlador. Os autores examinam o *trade-off* entre otimização do consumo de energia e a qualidade de serviço oferecida ao tráfego de alta prioridade.

Hongyun et al. [50] introduzem uma arquitetura para gerenciamento de múltiplos serviços sobre ambientes totalmente confiáveis (*carrier-grade*) baseados em redes definidas por software. A arquitetura proposta pelos autores aplica redes virtuais para isolar o tráfego de dados e lidar com os requisitos dos serviços em tempo real.

Basta et al. [8] mostram quatro arquiteturas baseadas em Openflow para aplicar *Network Function Virtualization* (NFV) sobre redes *Long Term Evolution* (LTE). A proposta implanta um elemento extra ao protocolo Openflow, chamado NE+, para adicionar

novas funções de rede às redes LTE.

Lin et al. [52] apresentam o *Software Defined Infrastructure* (SDI) para realizar um controle integrado e gerenciar os recursos de rede e computação em centros de dados (em inglês *Datacenters*) baseados em computação em nuvem. Esta abordagem aplica dois controladores, um controlador SDN e outro controlador para a computação em nuvem.

Tabela 2.1: Trabalhos Relacionados

Referência	Contexto	Foco
Zhang et al. [83]	VN	Alocação periódica de recursos
Carvalho et al. [13]	VN	Monitoramento de SLA
Georgopoulos et al. [24]	Openflow	QoE equivalente para os vídeos
Skoldstrom and Yedavalli [73]	Openflow e VN	Alocação esparsa de recursos
Bueno et al. [12]	SDN e VN	Requisitos das aplicações
Zhang et al. [82]	SDN	Eficiência energética no controlador
Hongyun et al. [50]	SDN e <i>Carrier-Grade</i>	Gerenciamento de múltiplos serviços
Basta et al. [8]	SDN e LTE	Gerenciamento de NFV
Lin et al. [52]	SDN e Datacenter	Alocação de recursos na nuvem

Nenhum dos trabalhos citados encontrados na literatura foca no desenvolvimento de uma proposta para gerenciar e negociar redes virtuais definidas por software. Estes aspectos são tratados na arquitetura desenvolvida nesta tese, englobando as tarefas de negociação de SLA, identificação da classe de tráfego, alocação da rede virtual e adaptação da rede virtual pós-implantação. Desta forma, a abordagem desta tese agrega novos aspectos às pesquisas de redes de computadores no que se refere a redes virtuais e redes definidas por software.

## Capítulo 3

# Arquitetura Proposta

Ao longo dos anos, a Internet vem fazendo parte do cotidiano da sociedade, onde os usuários querem ter acesso à Internet em todo os lugares e o tempo todo com um nível mínimo de qualidade. Este fato faz com que a demanda de recursos para se ter acesso à Internet mude durante o dia, bem como com que os provedores de Internet tenham que melhorar a utilização dos recursos de rede, atendendo os requisitos mais específicos das aplicações.

Como consequência, há a tendência que os provedores de Internet apliquem redes virtuais definidas por software (ou seja, redes virtuais sobre redes definidas por software) para melhorar o uso dos recursos de rede e gerenciar com maior flexibilidade os serviços prestados através de SLAs.

Dentro deste contexto, foi desenvolvida uma arquitetura para interligação de mecanismos para gerenciamento de redes virtuais definidas por software. A arquitetura proposta engloba as seguintes tarefas:

- **Negociação do SLA:** a partir da especificação requisitada pelo cliente, a arquitetura permite a avaliação de se é possível implantar a rede virtual para o cliente, bem como os parâmetros que melhor atendem os requisitos desejados;
- **Implantação da rede virtual:** considerando os parâmetros definidos no SLA já negociados, bem como as informações dos recursos na infraestrutura de rede, a arquitetura habilita a alocação da rede virtual que melhor se ajusta a estas informações focando em aspectos como largura de banda disponível, eficiência energética, resiliência, dentre outros;
- **Adaptação da rede virtual:** visando atender os parâmetros especificados no SLA e/ou melhorar a utilização dos recursos de rede, a arquitetura permite o monitoramento das redes virtuais ajustando as características das mesmas caso identifique a necessidade.

As Seções 3.1 e 3.2 descrevem os módulos existentes na arquitetura proposta e como as tarefas citadas acima são realizadas pela mesma, respectivamente.

### 3.1 Descrição da arquitetura e seus módulos

A arquitetura proposta é ilustrada na Figura 3.1. Os componentes em verde representam as tecnologias existentes para a implantação de redes virtuais sobre uma infraestrutura de rede definida por software. Por outro lado, os componentes em cinza representam os módulos desenvolvidos nesta tese para fazerem parte da arquitetura proposta.

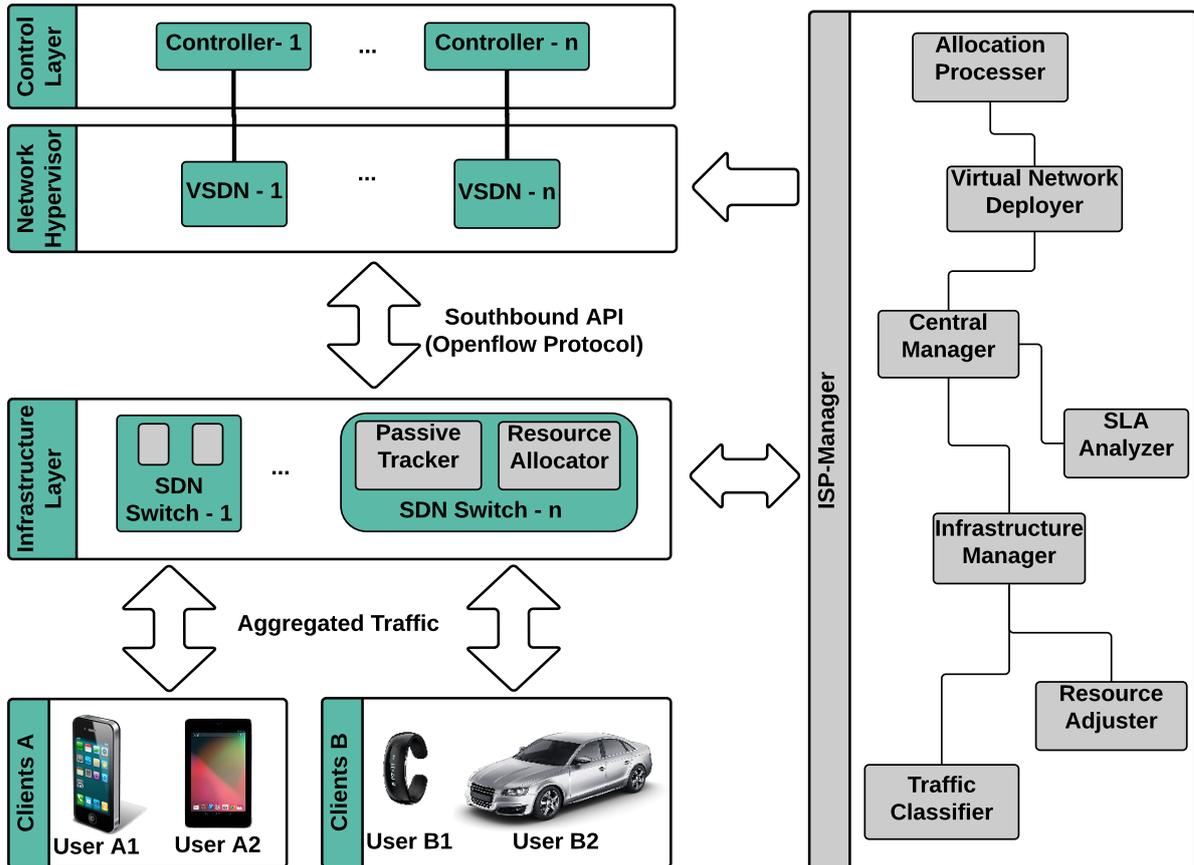


Figura 3.1: Visão Geral da Arquitetura Proposta

O ambiente *ISP-Manager* representa os módulos projetados para a arquitetura que atuam dentro do provedor, a fim de habilitar as tarefas citadas anteriormente. Este ambiente interage com a infraestrutura de rede (ambiente *Infrastructure Layer*) e o *Network Hypervisor* existente.

O ambiente *Network Hypervisor* permite a divisão da infraestrutura de redes em camadas, onde cada camada é uma rede virtual ligada a um controlador existente no ambiente *Control Layer*. Portanto, o ambiente *Control Layer* engloba todos os controladores instanciados.

O ambiente *Infrastructure Layer* representa os componentes físicos da rede definida por software, ou seja, os comutadores (*SDN switches*) que fazem parte da infraestrutura de rede. Em cada *SDN switch* há dois módulos desenvolvidos *Passive Tracker* e *Resource Allocator*, os quais visam permitir o monitoramento e controle dos recursos do *SDN switch* em questão, respectivamente.

O ambiente *Usuários* engloba os diversos dispositivos que permitem os usuários fazerem uso das redes virtuais implantadas, como por exemplo telefones celulares, *tablets*,

*smart tvs*, notebooks, veículos automotivos, *smart watches*, dentre outros.

Os módulos definidos são detalhados nas seções a seguir e uma visão geral do funcionamento da arquitetura, bem como a interação entre os módulos desenvolvidos, é apresentada na Seção 3.2.

### 3.1.1 Central Manager

A arquitetura proposta necessita de um módulo central para controlar a interação entre os demais módulos da arquitetura, onde o módulo *Central Manager* gerencia os aspectos gerais e a comunicação entre os módulos. Em geral, o módulo *Central Manager* controla o comportamento da arquitetura como um todo. O módulo *Central Manager* gerencia desde o processo de provisionamento de serviços pelo provedor de Internet até a implantação da rede virtual, bem como o processo de adaptação da mesma quando identificada a necessidade.

Primeiramente, este módulo define e negocia o SLA entre o cliente e provedor. Após o final da negociação, o módulo *Central Manager* inicia o processo de implantação da rede virtual. Adicionalmente, monitora-se as redes virtuais e a infraestrutura de rede a fim de gerencia-las e adapta-las quando identifica-se a necessidade. Portanto, para gerenciar essas tarefas o módulo *Central Manager* interage diretamente com os módulos *SLA Analyzer*, *Virtual Network Deployer* e *Infrastructure Manager*.

### 3.1.2 SLA Analyzer

O módulo *SLA Analyzer* especifica o SLA e gerencia a negociação entre o cliente e o provedor de Internet. É preciso ter em mente que a especificação do SLA define a rede virtual, ou seja, os(as) protocolos/aplicações de rede a serem instanciados(as) e os recursos iniciais da rede virtual alocada. Então, os passos realizados pelo módulo *SLA Analyzer* determinam se é viável implantar a rede virtual requisitada a partir dos(as) protocolos/aplicações e os recursos de rede disponíveis.

Sendo assim, este módulo consulta as informações referentes aos recursos disponíveis no módulo *Infrastructure Manager*. Da mesma forma, o módulo *SLA Analyzer* requisita ao módulo *Virtual Network Deployer* qual conjunto de protocolos/aplicações de rede está disponível para ser instanciado na rede virtual.

Adicionalmente, o módulo *SLA Analyzer* inicia o processo de implantação de uma nova rede virtual quando o processo de negociação é finalizado e o SLA é definido. Portanto, este módulo requisita a alocação dos recursos disponíveis ao módulo *Infrastructure Manager*. Da mesma forma, o módulo *SLA Analyzer* informa ao módulo *Virtual Network Deployer* qual conjunto de protocolos foi definido no SLA para ser instanciado na rede virtual.

Dentro deste contexto, foram desenvolvidos um protocolo de negociação de SLA e um mecanismo de apoio à negociação para lidar com as tarefas do módulo *SLA Analyzer*. A descrição completa e avaliação dos protocolos desenvolvidos é apresentada no Capítulo 5.

### 3.1.3 Virtual Network Deployer

O principal objetivo do módulo *Virtual Network Deployer* é interagir com as entidades relacionadas à virtualização sobre rede SDN, ou seja, com o *Network Hypervisor* e os controladores de cada rede virtual. Sendo assim, este módulo realiza as seguintes tarefas: (i) configura a topologia da rede virtual gerada no *Network Hypervisor* e instancia o controlador da rede virtual e o conjunto de protocolos definidos no SLA; e (ii) requisita a alocação dos recursos para a rede virtual.

Como uma ação prévia à execução das tarefas mencionadas, o módulo *Virtual Network Deployer* requisita ao módulo *Allocation Processer* a topologia mais adequada a ser alocada de acordo com a especificação de rede virtual definida no SLA e com o estado da infraestrutura do provedor. Uma vez que a topologia da rede virtual está definida, o módulo *Virtual Network Deployer* é responsável pela implantação da rede virtual no *Network Hypervisor* e posteriormente do controlador para a rede virtual em questão. Adicionalmente, o módulo *Virtual Network Deployer* comunica ao módulo *Infrastructure Manager* os recursos a serem alocados para a rede virtual em cada componente de rede.

### 3.1.4 Allocation Processer

A geração da topologia da rede virtual executada pelo módulo *Allocation Processer* visa determinar os componentes de rede (nós e enlaces) a serem alocados para melhor atender as especificações definidas no SLA. Adicionalmente, o módulo *Allocation Processer* pretende melhorar a prestação de serviço do provedor de Internet, considerando aspectos como resiliência, confiabilidade, demanda de tráfego elástica, largura de banda disponível, eficiência energética, e outros.

Desta forma, o módulo *Allocation Processer* recebe as informações sobre os parâmetros da rede virtual e sobre o estado atual da infraestrutura de rede do provedor, oriundas dos módulos *Infrastructure Manager* e *SLA Analyzer*, respectivamente. Após a definição da rede virtual a ser alocada, os componentes e recursos a serem alocados em cada um são informados ao módulo *Virtual Network Deployer* que realiza a implantação da rede virtual.

A fim de definir os componentes a fazerem parte da rede virtual requisitada, é necessário o provedor ter um algoritmo de alocação de rede virtual. O algoritmo de alocação visa encontrar a topologia (componentes de rede) mais adequada que atenda a especificação de rede virtual e faça um melhor uso da infraestrutura do provedor.

### 3.1.5 Infrastructure Manager

O módulo *Infrastructure Manager* é responsável por gerenciar os aspectos relacionados à infraestrutura de rede. O módulo interage diretamente com os módulos *Passive Tracker* e *Resource Allocator* localizados em cada *switch* da rede para controlar as informações sobre recursos (disponibilidade e uso) e a alocação dos recursos (incremento ou decremento na alocação), respectivamente. Portanto, cada *switch* da rede possui esses dois módulos citados.

Sendo assim, o módulo *Infrastructure Manager* atua como uma interface para os assuntos relacionados ao gerenciamento de recursos para o provedor, guardando as informações referentes aos recursos e componentes disponíveis na infraestrutura de rede. Estas informações são diferentes das obtidas sobre a perspectiva do controlador, o qual acessa somente os componentes de rede alocados para uma rede virtual específica.

Diante disso, o módulo provê as informações e participa do ajuste de recursos para uma rede virtual, visto que ambas as tarefas são correlatas. A fim de realizar os ajustes, é necessário verificar os recursos disponíveis; por outro lado, quando os recursos são ajustados, é preciso atualizar as informações sobre os recursos da infraestrutura de rede e do SLA firmado para cada rede virtual.

### 3.1.6 Resource Adjuster

O objetivo do módulo *Resource Adjuster* é definir um mecanismo que identifique quando o SLA e a alocação dos recursos atual para a rede virtual não representam mais o estado da rede, bem como quando os recursos alocados estão ociosos. Sendo assim, este módulo identifica quando e como adaptar a rede virtual de acordo com a demanda atual e as características do tráfego corrente. Em redes SDN baseadas no protocolo OpenFlow, as informações do tráfego corrente podem ser obtidas através das estatísticas dos fluxos que fazem parte de cada *switch* fornecidas pelo próprio protocolo OpenFlow.

A fim de identificar se existe a necessidade de requisitar um ajuste nas redes virtuais vigentes, é necessário ter um mecanismo de ajuste. O mecanismo de ajuste avalia as seguintes informações: a situação corrente da infraestrutura de rede do provedor e as especificação vigente do SLA e da rede virtual implantada. A partir disso, foram desenvolvidos dois mecanismos de ajuste apresentados no Capítulo 7, os quais utilizam as informações oriundas dos módulos *Infrastructure Manager* e *SLA Analyzer*.

### 3.1.7 Traffic Classifier

A fim de realizar um gerenciamento das redes virtuais de forma mais adequada, a arquitetura proposta identifica a classe de QoS a que um certo fluxo pertence. Esta identificação é feita pelo módulo *Traffic Classifier*. O uso do módulo *Traffic Classifier* habilita a implantação e adaptação da rede virtual da forma mais adequada ao tráfego passante, possibilitando um melhor uso dos recursos de rede e a qualidade do serviço prestado aos usuários.

A classificação de tráfego realizada pela arquitetura proposta analisa as informações dos fluxos oriundas do módulo *Infrastructure Manager* em tempo real. O processo de classificação aplicado, bem como as etapas de desenvolvimento e a avaliação do mesmo, é apresentado no Capítulo 4.

### 3.1.8 Passive Tracker

O módulo *Passive Tracker* está presente em cada nó da infraestrutura de rede e fornece as informações referentes aos recursos disponíveis nos *switches*. Além disso, este módulo mantém as informações sobre os fluxos ativos e o tráfego passante de cada rede virtual,

possibilitando medir a demanda de tráfego do cliente que faz uso da rede virtual, bem como o gerenciamento dos fluxos de uma forma mais eficaz.

Nesta tese, considera-se a demanda ou volume de tráfego a soma total dos *bytes* oriundos do cliente durante o tempo de monitoramento. Este tempo representa a quantidade de segundos para coletar as informações para checar o estado atual da rede. Por padrão o tempo de análise ocorre a cada um segundo, a fim de se adequar as métricas de avaliação de rede utilizadas nos equipamentos de rede, bem como manter as informações atualizadas. Adicionalmente, o gerenciamento de fluxos consiste em identificar os fluxos, verificando quais fluxos estão ativos. Sendo assim, depois de um tempo de inatividade os mesmos são removidos da base de gerenciamento.

### 3.1.9 Resource Allocator

Assim como o módulo *Passive Tracker*, o módulo *Resource Allocator* está presente em cada nó da rede. Este módulo controla a alocação dos recursos para a rede virtual, realizando a alocação e liberação dos recursos. Os provedores tipicamente oferecem largura de banda como recurso de rede, contudo, a arquitetura proposta provê suporte a utilização de outros tipos. Em redes SDN baseadas no protocolo OpenFlow, a alocação dos recursos é feita com a formação de filas virtuais para cada rede virtual, desta forma é possível controlar a alocação dos recursos para cada VSDN.

A tarefa de definir o incremento ou decremento dos recursos alocados para a rede virtual não é o objetivo deste módulo, este apenas realiza o ajuste dos recursos de acordo com a indicação do módulo *Infrastructure Manager*. O módulo *Resource Allocator* age diretamente no *switch*, e realiza a mudança na alocação dos recursos nas interfaces de rede em tempo real.

## 3.2 Interação entre os Módulos da Arquitetura

A partir da visão geral apresentada da arquitetura proposta, bem como a descrição do foco de cada módulo projetado, esta seção apresenta o comportamento e a interação entre os módulos da arquitetura proposta para realizar o gerenciamento e negociação das redes virtuais. A fim de ilustrar a interação entre os módulos, a Figura 3.2 mostra um diagrama de sequência que apresenta os passos realizados em cada tarefa da arquitetura.

Na Figura 3.2, cada tarefa habilitada pela arquitetura é ilustrada por uma cor diferente: (a) a negociação do SLA para redes virtuais pela cor azul; (b) a implantação da rede virtual negociada pela cor verde; e (c) a adaptação da rede virtual em caso de necessidade pela cor vermelha. A partir disso, a seguir serão descritas cada uma das tarefas permitidas pela arquitetura proposta para gerenciar e negociar as redes virtuais.

### 3.2.1 Negociação do SLA

O módulo *SLA Analyzer* interage com o cliente na tarefa de negociação do SLA. Ao receber a especificação do SLA desejado pelo cliente, o módulo *SLA Analyzer* consulta os recursos disponíveis na infraestrutura de rede no módulo *Infrastructure Manager*, o

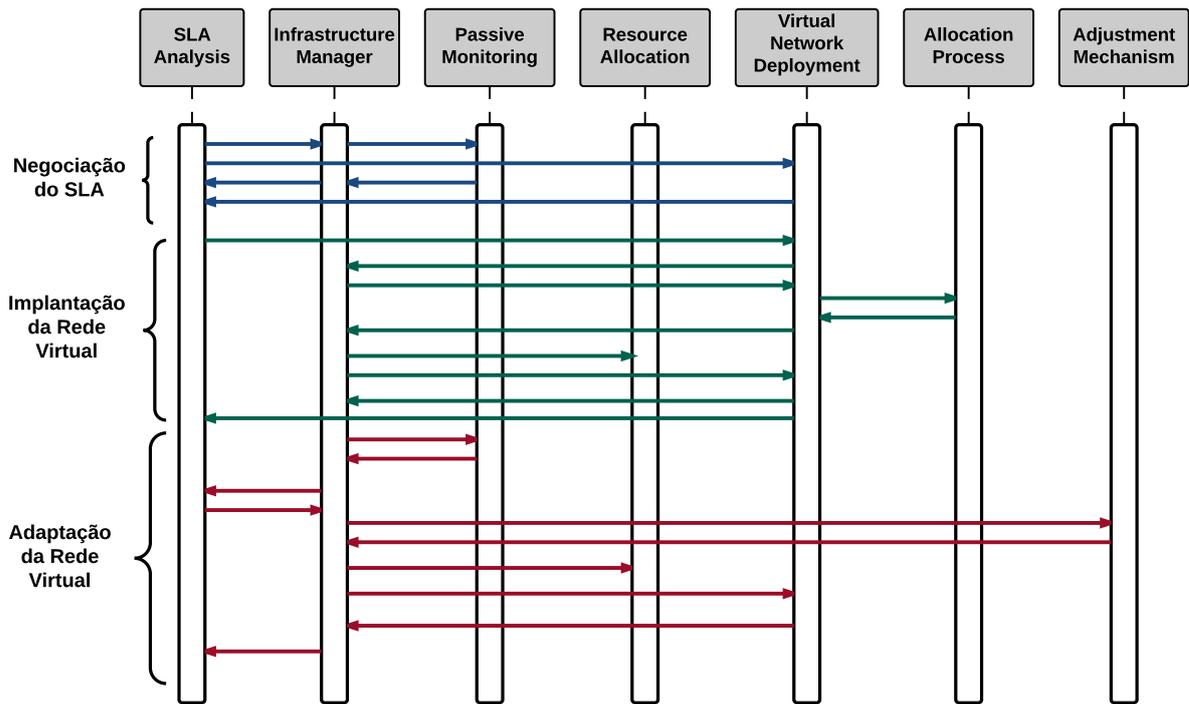


Figura 3.2: Passos realizados pela arquitetura

qual obtêm essas informações periodicamente através do módulo *Passive Tracker*. Adicionalmente, o módulo *SLA Analyzer* consulta no módulo *Virtual Network Deployer* quais protocolos/aplicações o provedor pode instanciar na rede virtual. Em posse das informações necessárias, o módulo *SLA Analyzer* analisa a especificação do SLA diante das informações coletadas e envia uma proposta de SLA para o cliente, o qual, em caso de aceitação, confirma a definição do SLA. Este processo é apresentado na Figura 3.3.

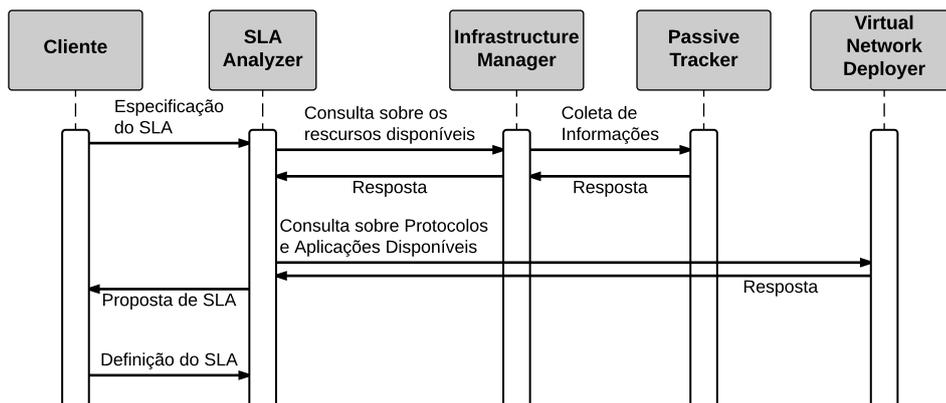


Figura 3.3: Negociação do SLA

### 3.2.2 Implantação da rede virtual

A Figura 3.4 ilustra os passos especificados pela arquitetura para implantar a rede virtual. Este processo ocorre após a definição do SLA entre o provedor e o cliente. Desta forma, os

passos mostrados na Figura 3.4 ocorrem quando a negociação do SLA, descrita na seção anterior, é finalizada.

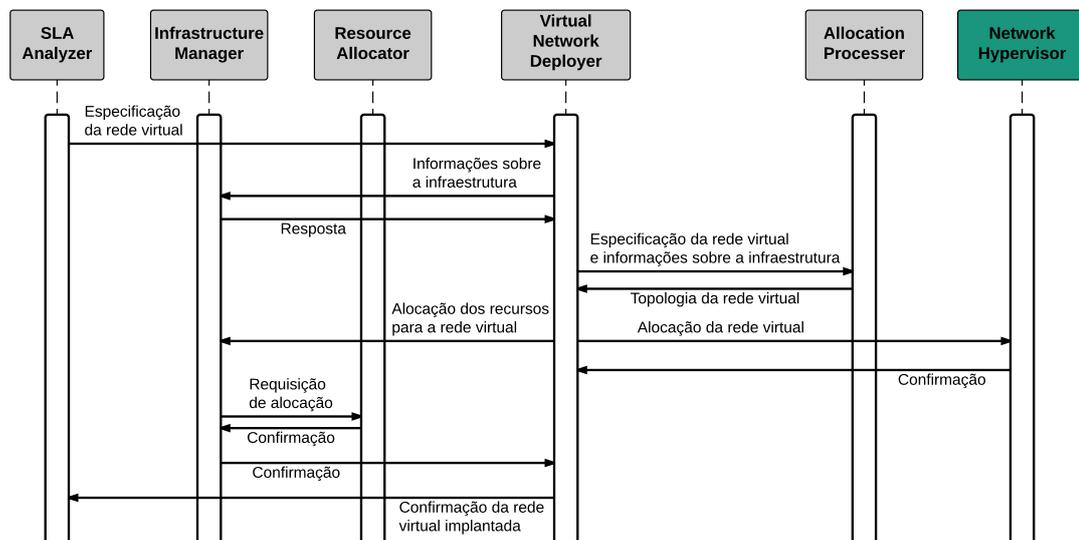


Figura 3.4: Implantação da rede virtual

Inicialmente, as informações referentes aos SLA definido são enviadas ao módulo *Virtual Network Deployer* pelo módulo *SLA Analyzer*. Adicionalmente, o módulo *Virtual Network Deployer* coleta as informações referentes ao estado da infraestrutura de rede (por exemplo, consumo de energia, largura de banda disponível, componentes de rede ativos, dentre outras) através do módulo *Infrastructure Manager*. A partir das informações sobre o SLA e a infraestrutura de rede, o módulo *Virtual Network Deployer* requisita ao módulo *Allocation Processer* que gere a topologia de rede mais adequada diante das informações fornecidas. Uma vez que a topologia da rede virtual foi definida, o módulo *Virtual Network Deployer* requisita que a topologia gerada seja alocada pelo *Network Hypervisor*, bem como que os recursos de rede para a rede virtual sejam alocados pelo módulo *Infrastructure Manager* através do módulo *Resource Allocator*. Finalizadas as etapas de alocação, a rede virtual é considerada implantada.

### 3.2.3 Adaptação da rede virtual

Quando o processo de implantação da rede virtual é finalizado, esta é considerada ativa e operacional. A partir desse momento a arquitetura permite o monitoramento periódico do estado das redes virtuais diante das condições da infraestrutura de rede e do SLA vigente, visando identificar se há a necessidade de adaptar as características da rede virtual.

Os passos realizados para a adaptação da rede virtual são ilustrados na Figura 3.5. O módulo *Infrastructure Manager* coleta periodicamente as informações sobre o volume de tráfego e as características dos fluxos ativos no módulo *Passive Tracker*. Adicionalmente, o módulo *Infrastructure Manager* obtêm a especificação do SLA.

Utilizando as informações coletadas, o módulo *Infrastructure Manager* requisita ao módulo *Resource Adjuster* se há a necessidade de adaptar as características da rede virtual. A fim de identificar o estado da rede de uma forma mais coerente, o módulo *Resource*

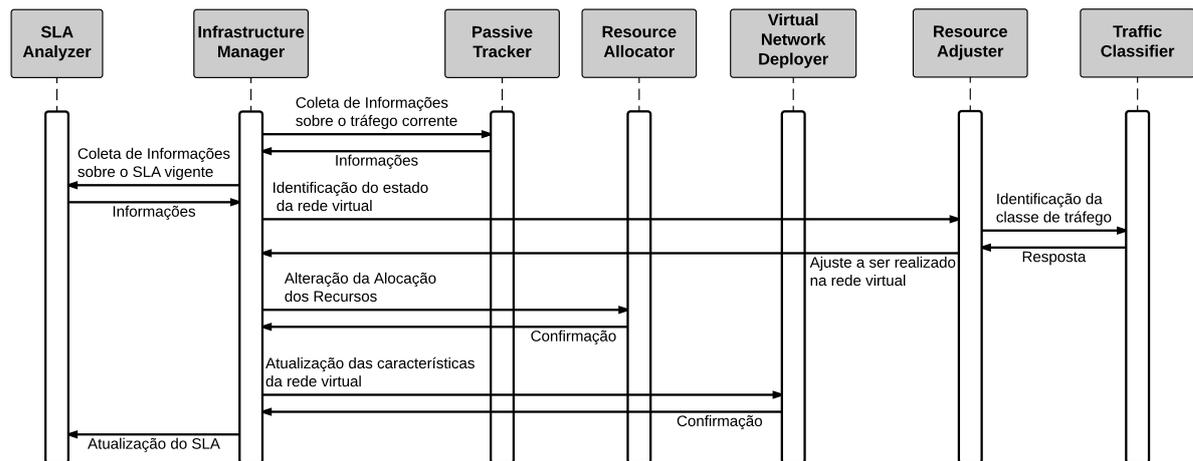


Figura 3.5: Adaptação da rede virtual

*Adjuster* identifica a classe de tráfego dos fluxos, através do módulo *Traffic Classifier*, e avalia se algum ajuste deve ser realizado para melhorar a prestação de serviço pelo provedor, a qualidade experienciada pelos usuários e/ou a utilização dos recursos de rede do provedor.

A partir do parecer oriundo do módulo *Resource Adjuster* considerando a necessidade de ajustar a rede virtual, o módulo *Infrastructure Manager* requisita a alteração dos recursos alocados para o módulo *Resource Allocator*, bem como notifica as características correntes da rede virtual no módulo *Virtual Network Deployer*. Finalizando o processo de ajuste, o módulo *Infrastructure Manager* atualiza o SLA no módulo *SLA Analyzer*.

A partir da arquitetura projetada apresentada neste capítulo é possível interligar os mecanismos propostos para o gerenciamento de redes virtuais definidas por software, proporcionando um funcionamento integrado e completo das tarefas necessárias. Portanto, nos próximos capítulos serão introduzidos os mecanismos propostos e seus objetivos específicos dentro das tarefas de gerenciamento como um todo.

# Capítulo 4

## Classificação de Tráfego

Este capítulo descreve o classificador de tráfego aplicado neste trabalho para fazer parte do módulo *Traffic Classification* (descrito na Seção 3.1.7). A classificação de tráfego deve ser realizada em tempo real, sendo assim necessita-se de uma classificação rápida e com bom índice de acerto.

Uma abordagem de 5 classes de QoS foi aplicada a fim de diferenciar os diversos tipos de tráfego, os quais têm diferentes requisitos de QoS. Desta forma, tem-se as seguintes classes de tráfego:

- *Audio*: engloba os pacotes de aplicações como *Voice over IP* (VoIP), rádios *online*, e outros;
- *Control*: inclui pacotes de controle e gerenciamento da rede;
- *Data*: representa as aplicações mais populares baseadas em tráfego em rajada;
- *Video On Demand*: engloba os pacotes referentes à recepção de vídeos sob demanda;
- *Video Real Time*: inclui os pacotes referentes às transmissões de vídeo ao vivo.

A referência [36] avalia as técnicas de Aprendizagem de Máquina (*Machine Learning - ML*) Naive Bayes, *Decision Tree*, *Linear Discriminant Analysis*, *Support Vector Machine* (SVM) e Redes Neurais, para uma classificação baseada em 4 classes de QoS: *Audio*, *Control*, *Data* e *Video*. As classes aplicadas são semelhantes à abordagem aplicada nesta tese, onde na referência [36] as classes *Video On Demand* e *Video Real Time* são consideradas equivalentes e em fundidas em uma única classe *Video*. De acordo com os resultados mostrados na referência [36], as técnicas Naive Bayes [60] e *Decision Tree* [69] obtiveram um desempenho superior para a classificação em tempo real, ou seja, uma boa precisão na classificação e um baixo tempo para realizar tal tarefa. Portanto, no trabalho avaliou-se essas duas técnicas sobre as abordagens de 4 e 5 classes.

Os classificadores foram treinados no software R<sup>1</sup>. Para realizar a classificação as seguintes informações dos pacotes são usadas: tamanho do cabeçalho IP, tamanho do *payload*, o tamanho total, IP *flag* usada, o campo *Time-to-live* (TTL), qual protocolo de camada de transporte é usado, porta de origem, porta de destino, campo *TCP offset*, e o

---

<sup>1</sup><http://www.r-project.org/>

campo *TCP flag*. As informações dos campos TCP são usadas somente se o protocolo em questão é utilizado pela aplicação. O classificador utiliza apenas as informações encontradas no momento em que o pacote passa pelo mesmo, visto que a classificação é realizada em tempo real.

Os pacotes da classe *Audio* foram coletados a partir de ligações VoIP (Skype e GTalk) e rádios online. A classe *Control* foi formada através dos pacotes SNMP, ICMP e IGMP. A classe *Data* engloba os pacotes referentes às aplicações HTTP, HTTPS, SMTP, POP3, FTP, DNS, SSH, Telnet e P2P (Gnutella e Torrent). Os pacotes referentes às classes *Video On Demand* e *Video Real Time* foram coletados através das aplicações: Skype, Google Hangout, Youtube e Livestream.

As Tabelas 4.1 e 4.2 apresentam a matriz de confusão de cada classificador avaliado. A matriz de confusão representa a quantidade de dados de classe  $j$  (coluna) o qual foi classificado como outra classe  $i$  (linha), a partir de um item específico  $(i, j)$ .

Por convenção, a classe padrão é representada pelas colunas, ou seja, a diagonal principal da matriz é a quantidade de pacotes que foram corretamente classificados, e os demais são erros de classificação. Por exemplo, a primeira linha da Tabela 4.1 representa que 1046 pacotes da classe *audio* foram classificados corretamente, enquanto que 49 pacotes da classe *audio* foram confundidos com pacotes da classe *data*.

Tabela 4.1: Matriz de Confusão do classificador Naive Bayes

Naive Bayes	audio	control	data	on demand	real time
audio	<b>1046</b>	0	49	1	2
control	2	<b>180</b>	18	0	4
data	4	12	<b>2992</b>	25	17
on demand	9	6	106	<b>982</b>	9
real time	0	0	2	0	<b>1108</b>

Tabela 4.2: Matriz de Confusão do classificador *Decision Tree*

Decision Tree	audio	control	data	on demand	real time
audio	<b>1024</b>	12	4	0	0
control	35	<b>184</b>	49	0	0
data	2	2	<b>3063</b>	0	0
on demand	0	0	51	<b>1008</b>	7
real time	0	0	0	0	<b>1133</b>

Baseado nas matrizes, pode-se determinar o desempenho dos classificadores, que é medido através de duas métricas: Acurácia (em inglês *Accuracy*) e Precisão (em inglês *Precision*), apresentadas nas Equações 4.1 e 4.2, respectivamente. Desta forma, nas equações  $T$  representa a matriz de confusão analisada e  $n$  o número de classes definidas. *Accuracy* representa a razão entre a soma das classificações corretas e o tamanho total da amostragem. Enquanto que *Precision* define quantas classificações certas ocorreram para cada classe  $i$  definida [37].

$$Accuracy = \frac{\sum_{i=1}^n T_{i,i}}{\sum_{i=1}^n \sum_{j=1}^n T_{i,j}} \quad (4.1)$$

$$Precision_i = \frac{T_{i,i}}{\sum_{j=1}^n T_{i,j}} \quad (4.2)$$

O desempenho de cada classificador pode ser visualizado nas Figuras 4.1 e 4.2. Na Figura 4.2 os valores da classe *Video* do classificador de 4 Classes foram duplicados para possibilitar a comparação com as classes *Video on Demand* e *Video Real Time* do classificador de 5 classes.

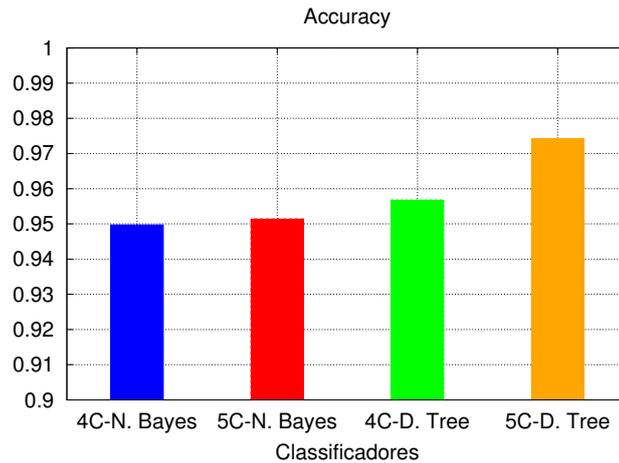


Figura 4.1: *Accuracy* dos Classificadores

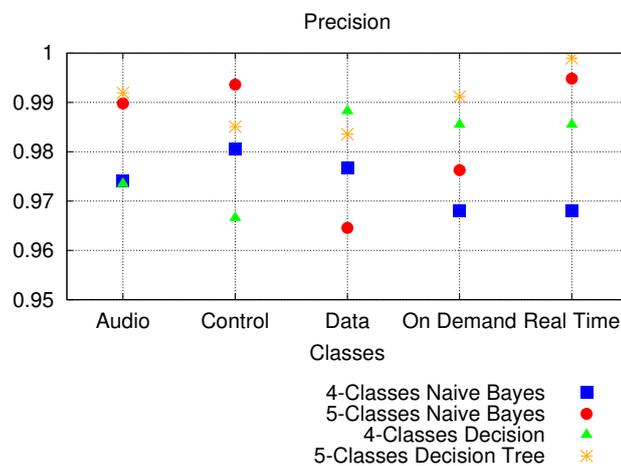


Figura 4.2: *Precision* dos Classificadores

Baseado nas Figuras 4.1 e 4.2, nota-se que todos os classificadores obtêm um bom desempenho. O aumento no desempenho do classificador de 5 classes pode ser justificado pelo fato da divisão da classe *Video* ter diminuído a interseção existente entre as duas classes e as demais [23].

Adicionalmente, avaliou-se o tempo para classificação de um pacote, onde foi aplicado um intervalo de confiança de 95%, o qual é mostrado na Figura 4.3. O desvio padrão do tempo de classificação foi muito pequeno quando comparado com o tempo de classificação em si, fato que dificulta a visualização do intervalo de confiança existente.

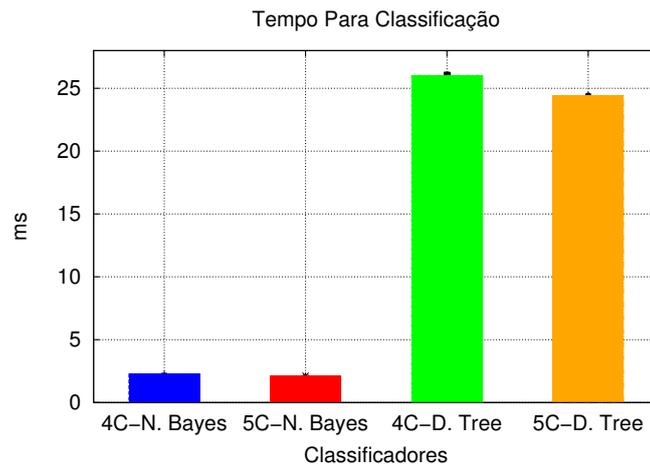


Figura 4.3: Tempo para classificação

Os classificadores Naive Bayes, de 4 e 5 classes, possuem tempo para classificação similar. Contudo, o tempo para classificação do *Decision Tree* de 5 classes é menor que o de 4 classes. Isto ocorre pois a altura da árvore gerada para o classificador de 5 classes é menor que a topologia do classificador de 4 classes, devido à divisão da classe *Video* em duas [69].

Portanto, o classificador Naive Bayes com 5 classes foi identificado como o mais adequado para fazer parte do projeto, visto que o mesmo possui uma boa taxa de acerto na classificação dos pacotes e gera um pequeno impacto no atraso fim-a-fim das aplicações.

Após a classificação individual dos pacotes, definiu-se uma estratégia para a classificação dos fluxos, pois a classificação de todos os pacotes adicionaria um atraso extra desnecessário para as aplicações. Sendo assim, uma abordagem para classificação parcial foi aplicada, onde somente alguns pacotes são classificados a fim de definir a classe de todos os pacotes do fluxo. Definiu-se para que a classificação do fluxo seja feita em até 5 pacotes, de acordo com as seguintes situações:

1. Dois pacotes do fluxo tem a mesma classificação em sequência;
2. Após cinco pacotes classificados, o maior número de classificações é assumido como a classe.

Na primeira situação, para um fluxo ser classificado erroneamente, o classificador precisa errar duas vezes seguidas. Usando o classificador Naive Bayes, que tem uma taxa de erro em torno de 5%, a probabilidade de uma classificação do fluxo errada é de 0.0025. Da mesma forma, na segunda situação, o classificador precisa errar três vezes para o fluxo ser classificado de forma errada, ou seja, tem-se uma probabilidade de 0.000125. Portanto, utilizando estas duas abordagens a probabilidade de se ter uma classificação do fluxo errada é muito baixa.

Após a determinação da classe do fluxo, esta é armazenada no campo *Type Of Service* (ToS) do cabeçalho de cada pacote. Isto permite a identificação da classe de cada pacote no processo de encaminhamento sem ser preciso a modificação da estrutura do pacote IP tradicional.

# Capítulo 5

## Negociação de SLA

Este capítulo detalha o comportamento do módulo *SLA Analysis* (apresentado na Seção 3.1.2), o qual permite a especificação do SLA e o gerenciamento da negociação entre o cliente e o provedor de Internet. SLA é um contrato, entre um provedor e um cliente, que especifica quais serviços o provedor irá fornecer e as ações que o mesmo cumprirá se o serviço prestado não for compatível com os objetivos estabelecidos no contrato firmado.

Um SLA só é válido se for gerenciado eficientemente. Devido a isso, o gerenciamento e negociação de SLA são considerados aspectos chave para se prover serviços de nova geração. Exemplos desses serviços são: voz sobre IP (*Voice Over IP* - VoIP), vídeo sob demanda, transferência de dados, entre outros. Para se ter um gerenciamento e negociação de SLA eficientes, necessita-se que os requisitos de ambos os lados do contrato sejam estabelecidos.

### 5.1 Negociação de SLA para Redes Virtuais

Devido à capacidade de flexibilidade e customização das redes virtuais, surgiu a necessidade de ter um processo de negociação de SLA, capaz de representar esse novo contexto. Desta forma, tem-se a necessidade de negociar não somente os recursos, mas também os(as) protocolos/aplicações de rede [57].

Dentro deste contexto, nesta tese são propostas: (i) um modelo de especificação de SLA baseada em classes, permitindo a negociação não somente dos recursos tradicionais de QoS, mas também dos(as) protocolos/aplicações de rede a serem utilizados, os quais representam o comportamento da rede como um todo; (ii) um protocolo de negociação de SLA para redes virtuais, considerando os recursos de rede e os(as) protocolos/aplicações para a rede virtual negociada.

#### 5.1.1 Modelo de Especificação de SLA

Além dos elementos tradicionais dos contratos SLA, definiu-se os seguintes componentes adicionais para o contexto de redes virtuais: a descrição das características da rede virtual, incluindo os(as) protocolos/aplicações de rede e os recursos a serem alocados. Uma visão geral do modelo de especificação pode ser visualizada na Figura 5.1.

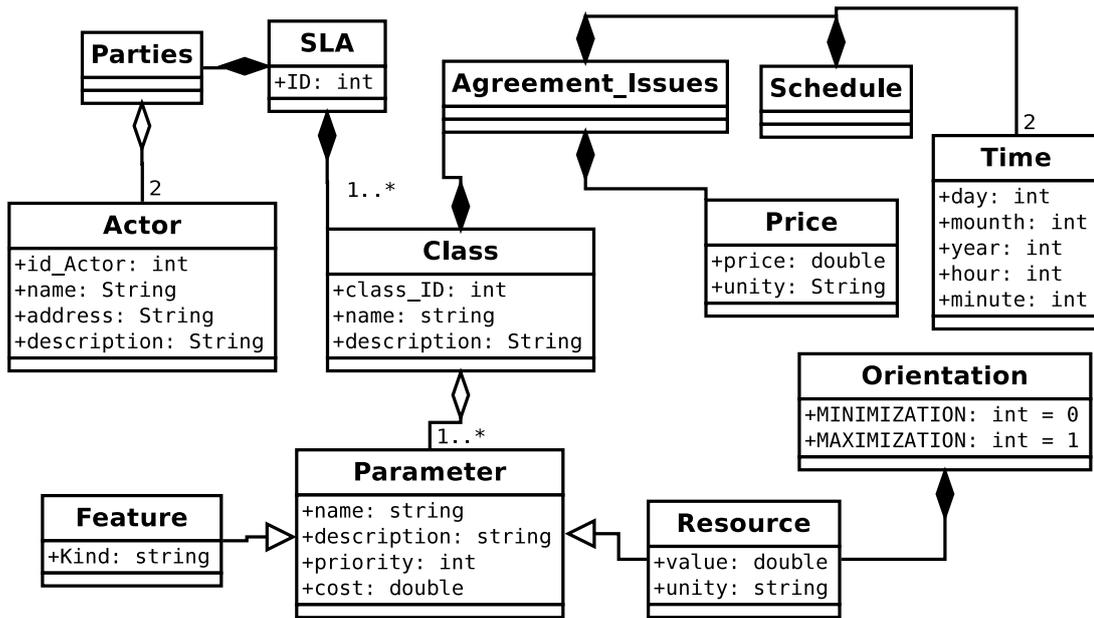


Figura 5.1: Diagrama que representa o modelo de especificação de SLA

O componente *SLA* é o componente raiz, contendo os componentes *Class* e *Parties*, além do identificador (*ID*) do contrato. Podem ser declarados diversos componentes *Class*, permitindo a negociação de quantas classes forem necessárias. Adicionalmente, o componente *Parties* define as partes envolvidas no contrato. Uma instância do componente *Parties* se relaciona com duas instâncias do componente *Actor*, que define as características de cada uma das partes envolvidas, além de possibilitar a execução do mecanismo de monitoramento e segurança.

Da mesma forma, o componente *Class* representa uma classe definida no SLA, onde pelo menos uma classe deve ser definida. O componente *Class* é formado por um ou mais componentes *Parameter*, o qual será explicado posteriormente. Cada componente *Class* possui uma instancia do componente *Agreement\_Issues*, o qual trata dos aspectos ligados ao contrato em relação à classe, como o tempo de duração do contrato (componente *Schedule*) e o preço relacionado ao mesmo (componente *Price*).

Ligado ao componente *Class*, o componente *Parameter* representa uma parâmetro a ser negociado no SLA, podendo ele ser do tipo *Feature* ou *Resource*. O tipo *Feature* representa os protocolos/aplicações de rede a serem negociados. Por outro lado, o tipo *Resource* caracteriza os elementos mensuráveis, sendo largura de banda, o principal exemplo. No componente *Parameter* o elemento *cost* representa o preço associado ao parâmetro em questão, assim como o elemento *priority* indica o nível de prioridade em um processo de negociação dos parâmetros de QoS do SLA.

Por outro lado, no componente *Resource* é definida uma orientação para a métrica (atributo *orientation*), o objetivo é indicar se a métrica deve ter os valores minimizados (*Minimization*) ou maximizados (*Maximization*). Por exemplo, as definições para uma métrica de atraso visam uma minimização visto que quanto menor o atraso do tráfego mais benéfico é para a aplicação, no caso de uma métrica de vazão ocorre o inverso, a mesma tende a ser maximizada.

O componente *Feature* possui apenas o elemento *kind*, responsável por determinar a qual estilo de elemento o parâmetro está associado, ou seja, visa diferenciar a negociação de um protocolo de roteamento da negociação de uma política de filas de pacotes a ser usada, proporcionando assim uma melhor avaliação da proposta do provedor em relação aos requisitos do usuário.

A partir do modelo definido, os clientes podem definir contratos SLA com os provedores, onde há a possibilidade de se determinar diversas redes virtuais, cada qual com suas particularidades (parâmetros de QoS, protocolos/aplicações de rede, duração do contrato, custo, etc).

### 5.1.2 Protocolo de Negociação Desenvolvido

Dando continuidade ao processo de negociação, foi desenvolvido um protocolo de negociação completa, ou seja, que possui a habilidade de negociar os recursos de rede e os(as) protocolos/aplicações para a rede virtual negociada. Além disso, o protocolo suporta a negociação de diversas classes, que podem caracterizar tipos diferentes de aplicações, possibilitando assim a negociação de diversas redes virtuais, cada qual com suas particularidades. O diagrama de sequência mostrado na Figura 5.2 representa o comportamento do protocolo, apresentando a troca de mensagens entre o cliente e um provedor.

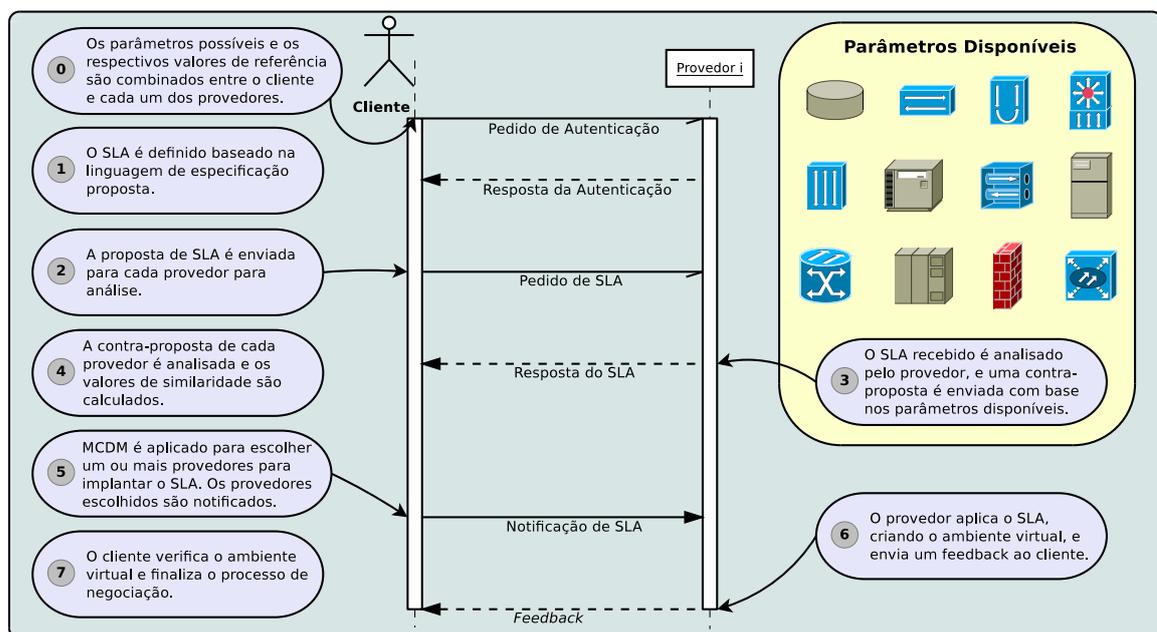


Figura 5.2: Diagrama de Sequência

Primeiramente, o cliente requisita um processo de autenticação para acessar os serviços disponíveis (1). Após isso, se a autenticação foi realizada com sucesso, o cliente envia uma proposta de SLA ao provedor (2), baseada no modelo de especificação mostrada anteriormente. Recebendo a proposta de SLA, para cada classe definida, o provedor analisa os aspectos relacionados ao acordo (*Agreement Issues*), aos parâmetros de rede (*Resources*) e aos(as) protocolos/aplicações (*Features*) (3). Em relação ao *Agreement Issues*, é verificado se o tempo de implantação da rede pode ser satisfeito.

Relacionado aos recursos da rede, é analisado se as métricas de rede, por exemplo largura de banda e/ou atraso, podem ser garantidos. Se a métrica pode ser totalmente garantida é enviado o valor original, mas se o provedor só puder garantir parcialmente a métrica, é enviado o montante que pode ser suportado. E se o provedor não tiver suporte a métrica em questão, esta não é incluída na resposta do SLA.

Em relação às características, caso o provedor possa implantar a rede virtual com os(as) protocolos/aplicações desejados pelo cliente, a mesma configuração de rede é respondida pelo provedor. Entretanto, se o provedor não suporta o protocolo requisitado, o mesmo envia outro protocolo de estilo compatível (*kind* do modelo). A ideia de determinar um *kind* representa o foco de atuação do protocolo, por exemplo roteamento, endereçamento, gerenciamento, etc. Para lidar com estes casos, foi projetado um mecanismo de apoio à negociação de SLA para redes virtuais, o qual será detalhado na Seção 5.3.

Por exemplo, se um cliente pede o protocolo de roteamento *Enhanced Interior Gateway Routing Protocol* (EIGRP), e o provedor não suporta, o provedor pode enviar o protocolo *Routing Information Protocol* (RIP) como resposta. Entretanto, na literatura ainda não há uma modelagem de similaridade que foque nos aspectos relacionados ao contexto de redes virtuais, a fim de avaliar quão adequado é um(a) protocolo/aplicação para substituir o(a) outro(a). Sendo assim, a seção a seguir apresenta uma proposta de modelo de similaridade com tal objetivo.

## 5.2 Modelo de Similaridade para Negociação de Redes Virtuais

ISPs possuem diferentes aspectos de infraestrutura e de protocolos/aplicações disponíveis para realizar as diversas tarefas de rede. Contudo, os clientes desejam algumas propriedades para os(as) protocolos/aplicações, os(as) quais definem o comportamento da rede virtual. Na literatura ainda não existe um trabalho que tenha por objetivo definir as propriedades que precisam ser analisadas a fim de negociar/comparar protocolos/aplicações de rede, ou seja, não existe um modelo em que se possa caracterizar os(as) protocolos/aplicações de redes a partir do seu estilo (focam na mesma tarefa), fato que permite a comparação entre protocolos/aplicações do mesmo estilo mas com comportamentos diferentes (conjunto de propriedades distintas).

Dentro deste contexto, propõe-se um modelo de similaridade para a negociação de redes virtuais. A proposta foca em dois aspectos: (i) descrever um modelo para caracterizar os principais estilos de protocolos/aplicações de rede que possam ser customizados(as) em uma rede virtual; e, (ii) propor uma abordagem para avaliar a similaridade entre protocolos/aplicações do mesmo estilo. O modelo proposto permite a livre concorrência entre ISPs e habilita o cliente a identificar qual protocolo/aplicação melhor atende os requisitos pedidos.

### 5.2.1 Modelagem de Similaridade Proposta

A abordagem mais popular para avaliar similaridade e caracterizar objetos é definir um vetor de características (*feature vector*) que representa as propriedades do objeto de acordo com seu estilo, onde cada posição do vetor de características representa uma propriedade do objeto [41].

A ideia de estilo representa o foco de um(a) protocolo/aplicação de rede, ou seja, qual tarefa este(a) protocolo/aplicação visa realizar. Esta ideia permite identificar os(as) protocolos/aplicações que possuem o mesmo estilo no processo de negociação. Por exemplo, não se pode comparar os protocolos *Open Shortest Path First* (OSPF) e IPv4 no processo de negociação. Então, aplicando a ideia de estilos identifica-se o IPv4 como um protocolo de endereçamento e o OSPF como protocolo de roteamento, com isso, cada protocolo pode ser comparado de forma mais justa e coerente.

Na modelagem proposta, aplicou-se a abordagem de vetor de características binário, ou seja, as posições do vetor representam a presença ou ausência de uma certa propriedade, geralmente retratadas pelos valores 1 e 0, respectivamente. Portanto, propõe-se a definição de um vetor de características para cada estilo de protocolo/aplicação de rede que pode ser customizado em uma rede virtual. Para o caso de características mensuráveis, definiu-se um valor para ser considerado como limiar para atender a característica ou não. Por exemplo, uma tarefa precisa ser realizada em até 10 milissegundos para ser considerada rápida.

A seguir será mostrado um exemplo de vetor de características, as propriedades de um protocolo de roteamento são enumeradas, as quais serão utilizadas posteriormente em experimentos realizados. Informações mais detalhadas sobre o modelo de similaridade podem ser encontradas no artigo [31].

#### Exemplo de *Feature Vector*: Protocolos de Roteamento

Um protocolo de roteamento especifica como os nós da rede comunicam-se uns com os outros, propagando as informações que permitem selecionar as rotas entre nós da rede [46]. Nesta modelagem, focou-se nos(as) protocolos/aplicações que trocam informações de roteamento dentro de um único domínio. Para este tipo, os protocolos mais populares são [7]: OSPF, RIP, *Interior Gateway Routing Protocol* (IGRP), *Intermediate System to Intermediate System* (IS-IS), e EIGRP.

A seguir, são mostradas as principais propriedades que definem o comportamento de um protocolo de roteamento, e então caracterizando-o:

1. Tempo de Convergência: é a quantidade de tempo entre uma mudança na rede e o restabelecimento da consistência das tabelas de roteamento. Então, se o protocolo é considerado rápido tem o valor 1, e o valor 0 se é considerado lento.
2. Consumo de Recursos: refere-se à quantidade de memória e processamento do roteador usada pelo protocolo. Sendo assim, quando um protocolo faz pouco uso dos recursos possui o valor 0, e caso contrário tem o valor 1.
3. Consumo de Rede: representa quanto dos recursos de rede o protocolo consome, onde leva-se em consideração principalmente o tamanho das mensagens e a frequência com

que elas são trocadas entre os nós da rede. Portanto, se o protocolo consome poucos recursos, este assume o valor 0, e o valor 1 caso contrário.

4. Múltiplos Caminhos: Nesta propriedade, o protocolo obtém o valor 1 se oferece suporte a abordagem de múltiplos caminhos, e o valor 0 caso contrário.
5. Escalabilidade: refere-se à característica do protocolo ser bem escalável de acordo com o tamanho da rede. Portanto, se o protocolo é escalável possui o valor 1, e 0 caso contrário.
6. Evitar *Loops*: se o protocolo previne um *loop* de roteamento, obtém o valor 1, e 0 caso contrário.

Usando as propriedades citadas, pode-se definir um vetor de características para os(as) protocolos/aplicações de roteamento. Por exemplo, pode-se definir os seguintes vetores de características para representar os protocolos RIP, IGRP, EIGRP, IS-IS e OSPF [46, 7]. Onde cada posição do vetor de características representa uma propriedade enumerada anteriormente. Portanto, a primeira posição refere-se ao tempo de convergência, a segunda ao consumo de recursos, e assim por diante.

$$RIP_{fetVec} = [0, 1, 0, 0, 0, 0]$$

$$IGRP_{fetVec} = [0, 0, 1, 1, 0, 0]$$

$$EIGRP_{fetVec} = [1, 1, 1, 1, 1, 1]$$

$$IS - IS_{fetVec} = [1, 1, 1, 0, 1, 1]$$

$$OSPF_{fetVec} = [1, 0, 1, 1, 1, 1]$$

### 5.2.2 Medição de Similaridade

Similaridade é definida como o grau de quão parecidos são dois objetos [19]. Geralmente, os valores de similaridade estão em um intervalo entre 0 e 1 [21]. Cada métrica de similaridade de dados binários possui particularidades pois focam em aspectos diferentes durante a avaliação, de acordo com o tipo e abordagem.

A Unidade Operacional de Taxonomia (*Operational Taxonomic Unit* - OTU) é a maneira mais comum de apresentar a similaridade entre dois objetos com características binárias [19]. Comparando os valores de cada propriedade para cada par de vetores binários,  $i$  e  $j$  tendo  $n$  propriedades, de acordo com a Tabela 5.1.

Tabela 5.1: OTU para Dados Binários [19]

OTU	1 ou Presença	0 ou Ausência
1 ou Presença	a	b
0 ou Ausência	c	d

Na Tabela 5.1,  $a$  é o número de propriedades que ambos possuem valor (*positive matches*);  $b$  é o número de propriedades em que as variáveis  $i$  e  $j$  são 1 e 0 (*positive*

*mismatches*), respectivamente;  $c$  é o número de propriedades em que as variáveis  $i$  e  $j$  são 0 e 1 (*negative mismatches*), respectivamente;  $e$ ,  $d$  é um número de propriedades que ambas as variáveis possuem valor 0 (*negative matches*).

A soma da diagonal principal da Tabela 5.1 ( $a + d$ ) representa o número total de casamentos/acertos entre as variáveis  $i$  e  $j$ , enquanto que a soma da diagonal inversa ( $b + c$ ) é o número de propriedades que não casaram, onde a soma total da tabela sempre é igual a  $n$ . No contexto de negociação de SLA para redes virtuais, o conjunto de propriedades do(a) protocolo/aplicação requisitado(a) é representado pelas variáveis  $i$  e as variáveis  $j$  representam o(a) protocolo/aplicação oferecido(a).

### Métricas de Similaridade Existentes

Algumas métricas de similaridade contam somente os *positive matches*, outras incluem os *negative matches*, enquanto que algumas aplicam pesos a cada combinação correta ou errada [19, 21]. Duas das métricas de similaridade mais populares são: *Jaccard* ( $S_J$ ) e *Sokal & Michener* ( $S_{SM}$ ) [19], as quais são ilustradas nas Equações 5.1 e 5.2, respectivamente.

$$S_J = \frac{a}{a + b + c} \quad (5.1)$$

$$S_{SM} = \frac{a + d}{a + b + c + d} \quad (5.2)$$

A métrica *Jaccard* exclui os *negative matches* ( $d$ ), sendo definida como a razão entre os *positive matches* ( $a$ ) e o número total de propriedades menos os *negative matches* ( $(a + b + c + d) - d$ ). A métrica *Sokal & Michener* é definida como a razão das combinações totais (incluindo os *negative matches*) pelo número total de propriedades. Ambas as métricas tentam avaliar quais propriedades estão presentes no vetor de características analisado. Cada métrica de similaridade possui aspectos particulares, os quais avaliam de diferentes formas a relação entre dois vetores de características binário. Portanto, aplicando-se diferentes métricas obtêm-se diferentes resultados para os mesmos dois objetos.

### Métrica de Similaridade Proposta

No contexto desta tese, os objetos são protocolos/aplicações para redes virtuais, caracterizados por um vetor de características de acordo com as definições apresentadas na Seção 5.2. Em geral, o cliente requisita um conjunto de propriedades a serem implantadas na VN, e o provedor responde a requisição com um(a) protocolo/aplicação que nem sempre possui todas as características desejadas pelo cliente.

A fim de avaliar quão apropriados são os(as) protocolos/aplicações oferecidos(as), é necessário considerar as propriedades requisitadas que são atendidas pelo(a) protocolo/aplicação (*positive matches* -  $a$ ) como um ponto positivo. Por outro lado, deve-se avaliar como ponto negativo as propriedades que não são atendidas (*negative matches* -  $b$ ). Além disso, se o(a) protocolo/aplicação oferecido(a) possui algumas propriedades que não foram requisitadas pelo cliente, estas adicionam funcionalidades à rede, representando um ponto positivo.

A partir disso, foi desenvolvida a métrica de similaridade  $S_{VN}$ , que oferece uma maior importância para a presença das propriedades desejadas, considerando que a ausência de uma propriedade possa ser parcialmente compensada pela adição de uma propriedade extra. A métrica  $S_{VN}$  é mostrada na Equação 5.3. Nos casos que a similaridade gera valores negativos, esta é considerada zero, fazendo a métrica estar no intervalo  $[0, 1]$ .

$$S_{VN} = \frac{3a - 2b + c}{3a + 2b + c} \quad (5.3)$$

O comportamento da métrica proposta é mostrado na Figura 5.3. Para melhorar a visualização, a Figura 5.3 limita o número de propriedades a 9, omitindo os desconsiderados *negative matches*. Deve-se notar que a soma total,  $a + b + c + d$ , é igual ao número de propriedades no vetor de características ( $n$ ), portanto no gráfico não estão os casos que fazem parte dos *negative matches* ( $d$ ).

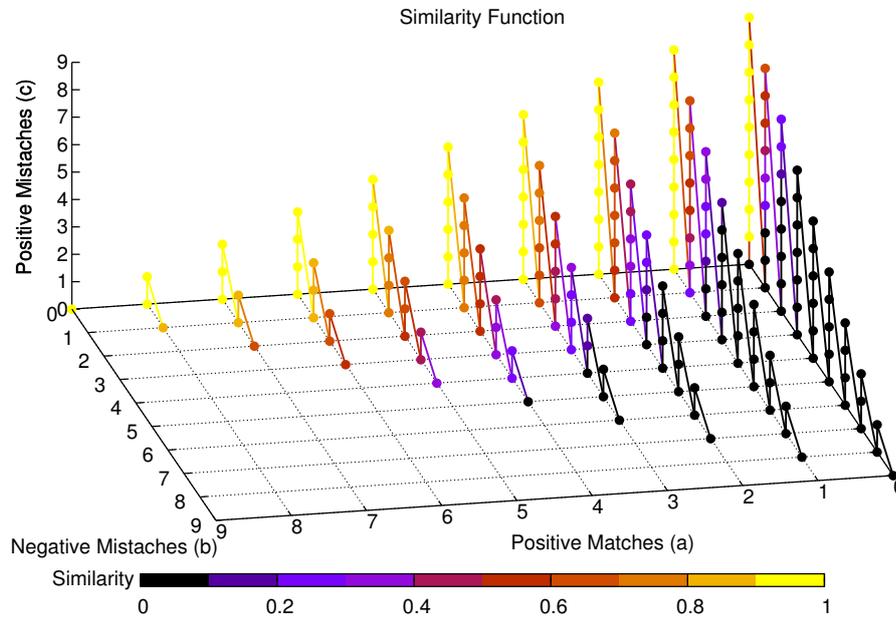


Figura 5.3: Comportamento da métrica  $S_{VN}$ .

De acordo com Lesot [49], a expressão  $\frac{a}{a+c}$  pode ser interpretada como a medida de quão o objeto requisitado está incluso no objeto oferecido. Por outro lado, a expressão  $\frac{a}{a+b}$  mede o inverso. Desta forma, a métrica proposta agrega as ideias de ambas as expressões, visando aumentar as propriedades atendidas e extras, enquanto penaliza de acordo com as propriedades não-atendidas.

### Medição de Similaridade com Pesos

Até o momento, para representar o(a) protocolo/aplicação desejado(a), todas as propriedades do vetor de características são consideradas equivalentes (mesmo grau de importância). Contudo, este fato nem sempre é realidade, visto que o cliente pode considerar algumas propriedades mais importantes que outras. A fim de melhorar a capacidade de diferenciação, pesos podem ser aplicados na medição da similaridade [14].

Por exemplo, no caso do cliente requisitar um(a) protocolo/aplicação com o seguinte vetor de características  $x = (1, 1, 1, 0, 1, 1, 1)$ , e receber duas contra-propostas  $y = (0, 1, 1, 1, 1, 1, 1)$  e  $z = (1, 1, 1, 1, 0, 1, 1)$ , ambas as propostas teriam o mesmo valor de similaridade, independente da métrica aplicada. Isto ocorre pois os valores de  $a$ ,  $b$ ,  $c$  e  $d$ , seriam os mesmos, no caso 5, 1, 1 e 0, respectivamente. Portanto, aplicando a métrica  $S_{VN}$  tem-se uma similaridade de 0.98 em ambos os casos ( $y$  e  $z$ ). Da mesma forma, aplicando as métricas *Jaccard* e *Sokal & Michener* tem-se 0.71 em ambos os casos, visto que o valor de  $d$  é zero.

Portanto, agregou-se a medição de similaridade uma abordagem de pesos, permitindo ao cliente priorizar as propriedades desejadas. Então agora para calcular a similaridade, além do vetor de características, necessita-se do vetor de prioridades  $w = (w_1, \dots, w_i, \dots, w_n)$ , que representa a importância ( $w_i$ ) de cada propriedade ( $i$ ).

A partir disso, pode-se quantificar os valores de  $a$ ,  $b$ ,  $c$ , e  $d$ , de forma diferente. Ao invés de cada propriedade adicionar em 1 o cálculo, esta adicionará o peso associado ( $w_i$ ) à mesma. Por exemplo, se a propriedade é um *positive match* com peso  $w_i$ , esta irá adicionar a  $a$  o valor de  $w_i$  ao invés de 1.

Aplicando a abordagem de pesos no exemplo anterior, caso defina-se o vetor de prioridades  $w = (3, 5, 4, 5, 4, 5, 5)$ , então o par  $(x, y)$  possui  $a = 23$  ( $w_2 + w_3 + w_5 + w_6 + w_7 = 5 + 4 + 4 + 5 + 5$ ),  $b = 3$  ( $w_1$ ),  $c = 5$  ( $w_4$ ) e  $d = 0$ . Da mesma forma, o par  $(x, z)$  tem  $a = 22$  ( $w_1 + w_2 + w_3 + w_6 + w_7 = 3 + 5 + 4 + 5 + 5$ ),  $b = 4$  ( $w_4$ ),  $c = 4$  ( $w_5$ ) e  $d = 0$ .

Portanto, aplicando-se a métrica  $S_{VN}$  resulta em uma similaridade de 0.998 para o par  $(x, y)$  e 0.997 para o par  $(x, z)$ , então  $y$  é considerado mais adequado para  $x$  do que  $z$ . Usando as métricas *Jaccard* e *Sokal & Michener* os valores de similaridade são de 0.74 para o par  $(x, y)$  e 0.70 para o par  $(x, z)$ .

### 5.3 Mecanismo de Apoio à Negociação de SLA

No contexto de VNs, os ISPs tem a capacidade de customizarem os parâmetros da rede e os serviços. Portanto, alguns trabalhos [27, 66, 81] usam esta flexibilidade para prover ao cliente o serviço desejado. O cliente pretende pagar o preço relativo a qualidade do serviço prestado. Desta forma, se o cliente deseja uma rede com poucas funcionalidades (propriedades), o mesmo espera pagar um preço proporcional por isso.

A partir da métrica de similaridade para o contexto de negociação de VNs (Seção 5.2.2), consegue-se estender as funcionalidades do protocolo de negociação de VNs proposto (Seção 5.1), para atender as expectativas do cliente sobre a perspectiva das propriedades da rede negociada.

Entretanto, até este momento existe uma lacuna referente ao custo financeiro envolvido no processo de negociação, bem como a relação deste custo com as propriedades existentes nos(as) protocolos/aplicações. Para suprir essa lacuna foi desenvolvido um mecanismo, chamado  $Fuzzy_{VN}$ , baseado em conjuntos *Fuzzy* [80] para identificar qual dos(as) protocolos/aplicações oferecidos(as) é o mais adequado ao conjunto de propriedades requisitadas pelo cliente utilizando dois critérios: (i) a similaridade entre o(a) protocolo/aplicação requisitado(a) e o(a) oferecido(a) no momento; e (ii) o preço vincu-

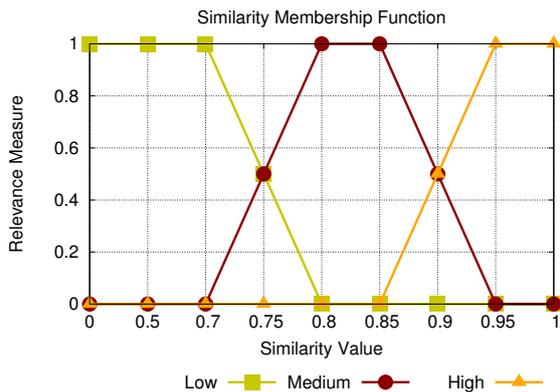
lado a este(a) protocolo/aplicação oferecido(a). O mecanismo proposto visa identificar o(a) protocolo/aplicação que melhor atenda as propriedades requisitadas.

No conceito tradicional de conjuntos, um elemento pertence totalmente ou não a um certo conjunto. Por outro lado, os conjuntos *Fuzzy* definem funções de adesão (*Membership Functions*) limitadas ao intervalo  $[0, 1]$ , para expressar o grau de adesão (em inglês *Relevance Measure*) de um elemento em relação ao conjunto em questão, representado por  $\mu(x)$  [3, 2, 80].

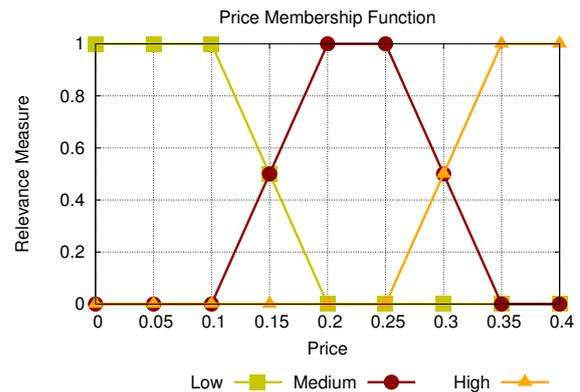
Além disso, os conjuntos *Fuzzy* são uma ferramenta para lidar com problemas de tomada de decisão, devido a duas razões: (i) lidam com o conceito de “grau de satisfação”; e, (ii) têm uma estrutura matemática para manipular informações vagas. Um sistema fuzzy possui as seguintes etapas: Fuzzificação (em inglês *Fuzzification*), Sistema de Inferência (em inglês *Inference System*) e Defuzzificação (em inglês *Defuzzification*).

### 5.3.1 Fuzzificação

O processo de fuzzificação recebe como entrada os valores de *Similaridade* e *Preço* (custo financeiro) os quais são transformados em variáveis linguísticas utilizadas no sistema de inferência. Portanto, duas funções de adesão são definidas, uma para cada critério. A função de adesão para os valores de Similaridade (em inglês *Similarity*) é mostrada na Figura 5.4(a), a qual tem três variáveis linguísticas: *High*, *Medium*, e *Low*. A variável *High* engloba os(as) protocolos/aplicações que tem valores de similaridade próximos do máximo possível. A variável *Medium* caracteriza os(as) protocolos/aplicações que de certa forma são similares ao desejado, ou seja, aqueles que dependendo do custo financeiro podem ser uma opção viável para o cliente. Por outro lado, a variável *Low* representa os(as) protocolos/aplicações que são considerados(as) somente quando não há outra opção.



(a) Função para os valores de *Similaridade*.



(b) Função para os valores de *Preço*.

A função de adesão usada para os valores de Preço (em inglês *Price*) é mostrada na Figura 5.4(b), a qual possui três variáveis linguísticas: *High*, *Medium*, e *Low*. Diferentemente dos valores da função de *Similaridade*, no caso da função de *Preço* quanto menor o preço, melhor para o(a) protocolo(aplicação). O valor de entrada é o preço normalizado dentre todos os(as) protocolos/aplicações oferecidos, onde para todos os  $m$  protocolos oferecidos  $P = (p_1, \dots, p_m)$ , o preço de entrada do  $i$  – *th* protocolo, chamado  $P_i^{input}$ , é o preço  $p_i$  dividido pela soma total de  $P$  (Equação 5.4).

$$P_i^{input} = \frac{p_i}{\sum_{i=1}^m p_i} \quad (5.4)$$

### 5.3.2 Sistema de Inferência

No sistema de inferência as variáveis linguísticas oriundas do processo de fuzzificação são aplicadas no conjunto de regras determinado e produzem um conjunto de variáveis linguísticas relacionadas à função de adesão Avaliação (do inglês *Evaluation*), a qual é usada como saída e é mostrada na Figura 5.4. A distribuição dos valores das variáveis linguísticas visa que um(a) protocolo/aplicação considerado(a) completamente *Medium* tenha um custo maior de um(a) protocolo/aplicação considerado(a) totalmente *High*. A mesma ideia é aplicada aos(as) protocolos/aplicações considerados(as) totalmente *Low* e *Medium*.

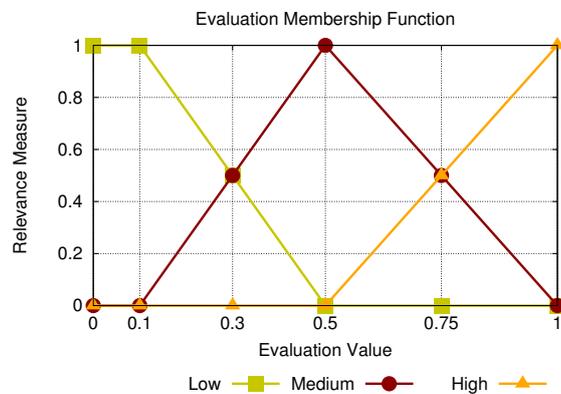


Figura 5.4: Função de adesão Avaliação

O sistema de inferência aplica o conjunto de regras apresentado na Tabela 5.2, os quais expressam as possíveis variáveis de saída de acordo com o processo de Fuzzificação. O operador “E” representa a interseção entre dois conjuntos fuzzy, que pode ser descrito a partir da função proposta em [80]:  $\mu A \cap B = \min [\mu A(x_i), \mu B(x_i)]$ . O conjunto de regras visa enaltecer os(as) protocolos/aplicações que possuem maior similaridade e um preço viável (*Low* ou *Medium*). Por outro lado, os(as) protocolos/aplicações que possuem valores de similaridade baixos são considerados(as) pouco adequados(as).

O processo de inferência visa transformar as variáveis linguísticas de entrada (saída da fuzzificação) em outras variáveis linguísticas de acordo com a função de saída (função Avaliação) e o conjunto de regras. Essas variáveis de saída são aplicadas no processo de Defuzzificação.

### 5.3.3 Defuzzificação

O processo de defuzzificação utiliza as variáveis linguísticas vindas do sistema de inferência e converte em um valor real de acordo com o método Weight Average Maximum (Equação 5.5), pois é um método eficaz e de baixo processamento, encaixando-se no escopo do trabalho que visa realizar as negociações de SLAs em tempo real. Este método produz

Tabela 5.2: Conjunto de Regras

Similaridade	Operação	Preço	Avaliação
High	E	High	Medium
High	E	Medium	High
High	E	Low	High
Medium	E	High	Low
Medium	E	Medium	Medium
Medium	E	Low	Medium
Low	E	High	Low
Low	E	Medium	Low
Low	E	Low	Medium

um valor numérico considerando o peso médio dos maiores valores (pico das variáveis linguísticas mostradas na Figura 5.4).

$$Fuzzy_{VN} = \frac{(0.1 * \mu_H(x)) + (0.5 * \mu_M(x)) + (1 * \mu_L(x))}{(\mu_H(x) + \mu_M(x) + \mu_L(x))} \quad (5.5)$$

Onde  $\mu_H(x)$  é o grau de adesão da variável *High*,  $\mu_M(x)$  é o grau de adesão da variável *Medium*, and  $\mu_L(x)$  é o grau de adesão da variável *Low*. Os valores 0.1, 0.5, e 1 são os valores máximos das variáveis *High*, *Medium*, e *Low*, respectivamente, como mostrado na Figura 5.4.

## 5.4 Estudo de Caso

Esta seção apresenta um estudo de caso da negociação de um protocolo de roteamento. Para isso precisa-se definir as propriedades que compõem o mesmo, as quais foram descritas anteriormente na Seção 5.2.1. Além disso, para ter uma base de comparação para o estudo de caso, foi conduzido um questionário, o qual será descrito na Seção 5.4.1. A Seção 5.4.2 apresenta os resultados obtidos.

### 5.4.1 Questionário

Normalmente, para avaliar a eficácia das métricas similaridade e dos modelos de tomada de decisão, utiliza-se uma base dados já conhecida. Contudo, no contexto deste trabalho, não existe tal base de dados. Sendo assim, para averiguar a qualidade do método proposto, foi conduzido um questionário sobre os protocolos de roteamento, onde 36 pesquisadores/profissionais responderam ao mesmo.

O questionário abordou dois aspectos: (i) o grau de importância (peso) de cada propriedade que caracteriza os protocolos de roteamento, a fim de ser aplicado de acordo com a Seção 5.2.2; e, (ii) quais protocolo são mais similares uns aos outros considerando as propriedades definidas, onde os protocolos RIP, IGRP, EIGRP, IS-IS, e OSPF foram analisados. Com relação ao grau de importância, os valores são apresentados em um intervalo

[0, 5] e os dados obtidos são resumidos na Tabela 5.3. Da mesma forma, as informações sobre quais protocolos são considerados mais próximos uns aos outros são apresentadas na Tabela 5.4.

Tabela 5.3: Grau de Importância

Propriedade	Primeiro	%	Segundo	%
Tempo de Convergência	5	63	4	29
Consumo de Recursos	3	43	4	29
Consumo de Rede	5	46	4	34
Múltiplos Caminhos	3	49	4	26
Escalabilidade	5	66	4	34
Evitar <i>Loops</i>	5	43	3	31

A partir dos dados mostrados na Tabela 5.3, pode-se notar que as propriedades Tempo de Convergência, Consumo de Rede, Escalabilidade, e Evitar *Loops* são considerados os mais importantes do conjunto. Portanto, os(as) protocolos/aplicações que suportam essas propriedades devem ter vantagem na medição de similaridade.

Tabela 5.4: Similaridade entre os Protocolos de Roteamento

Protocolo	Primeiro	%	Segundo	%
RIP	IGRP	91	OSPF	6
IGRP	RIP	51	EIGRP	43
OSPF	IS-IS	89	RIP	6
IS-IS	OSPF	89	EIGRP	9
EIGRP	IGRP	46	OSPF	31

Com relação às informações apresentadas na Tabela 5.4, os protocolos RIP, OSPF e IS-IS possuem o protocolo mais similar bem definido, visto que a primeira opção de cada possui valor em torno de 90%. Por outro lado, os protocolos IGRP e EIGRP obtiveram opções bem esparsadas, onde a primeira e segunda opção de cada tiveram valores próximos. Sendo assim, de acordo com o questionário, o protocolo mais similar para os casos de IGRP e EIGRP ainda não está claro.

## 5.4.2 Resultados

Avaliou-se a eficácia tanto da métrica de similaridade  $S_{VN}$  quanto do mecanismo  $Fuzzy_{VN}$  diante das métricas existentes *Jaccard* and *Sokal & Michener* [19, 14], na negociação dos três protocolos com similaridades bem definidas (de acordo com a Seção 5.4.1): RIP, IS-IS e OSPF.

O preço dos protocolos RIP, IGRP, EIGRP, IS-IS e OSPF foram configurados como 10, 20, 70, 50 e 45, respectivamente. O preço atribuído segue a ideia que quanto mais propriedades um protocolo tem, maior é o seu custo financeiro [25]. Adicionalmente, aplicou-se a abordagem de pesos descrita na Seção 5.2.2, onde o peso de cada propriedade seguiu as informações obtidas através do *survey* descrito na Seção 5.4.1. Sendo assim, o vetor de prioridades é  $w = (5, 3, 5, 3, 5, 5)$ .

Nos testes, o protocolo oferecido com maior pontuação é o escolhido como mais adequado. Quando mais de uma opção possui a maior pontuação, um critério alternativo pode ser adotado de acordo com a configuração do cliente, por exemplo o protocolo com menor preço, dentre os com maior pontuação.

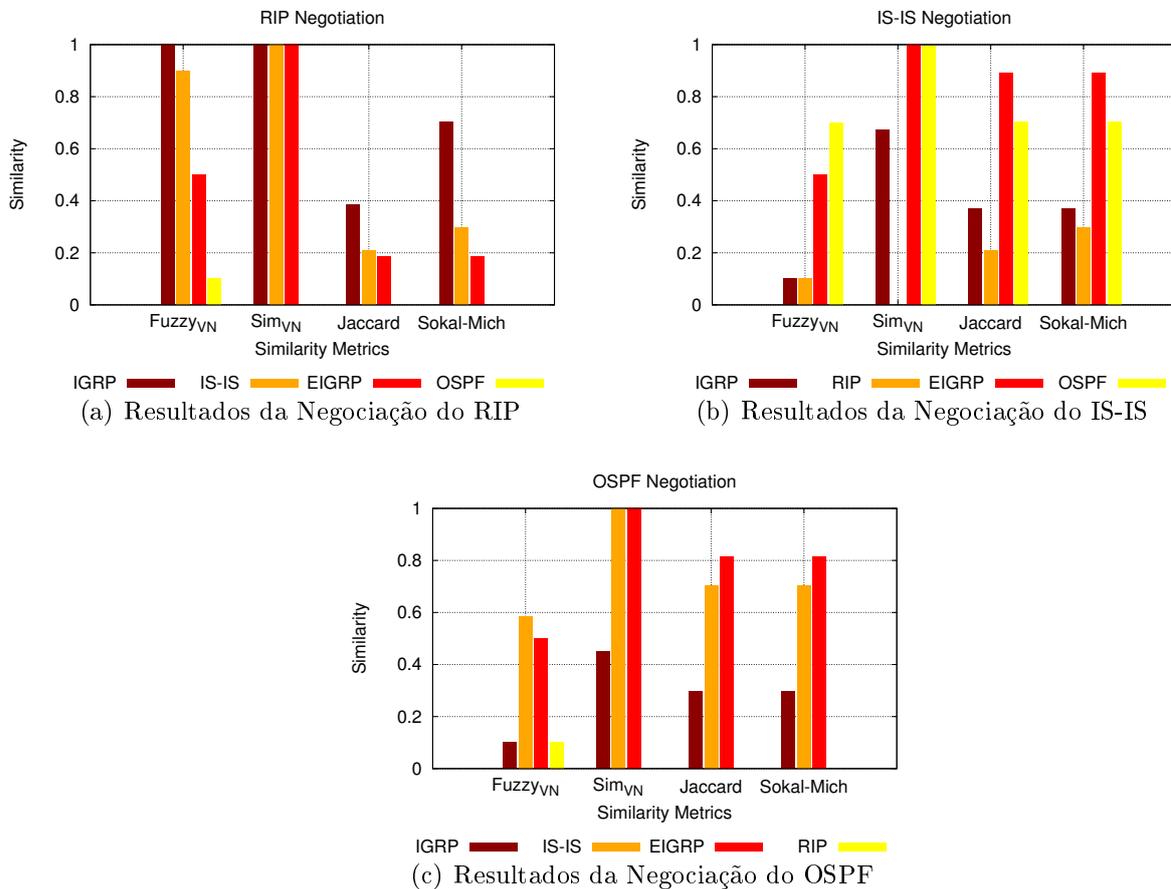


Figura 5.5: Resultados

Com relação a negociação do protocolo RIP, todas as métricas escolheram o mesmo protocolo, o qual foi o mesmo considerado mais adequado no questionário, como pode ser visto na Figura 5.5(a). Nos casos dos protocolos IS-IS e OSPF (Figuras 5.5(b) e 5.5(c), respectivamente) a vantagem do  $Fuzzy_{VN}$  para o contexto de negociação de VNs é clara, onde este é o único que escolhe os mesmos protocolos que foram considerados como melhor opção pelo questionário.

No geral, os resultados indicaram indícios da efetividade do mecanismo  $Fuzzy_{VN}$  proposto em identificar o protocolo mais adequado que melhor se encaixa ao conjunto de propriedades requisitados pelo cliente. Além disso, o modelo fuzzy proposto identifica o protocolo com melhor relação custo benefício para o cliente.

## Capítulo 6

# Alocação de Redes Virtuais Definidas por Software

Este capítulo descreve a geração da topologia da rede virtual executada pelo módulo *Allocation Process*. Esta geração é uma requisição do módulo *Virtual Network Deployment*, o qual necessita de uma topologia para a rede virtual que atenda a especificação de rede virtual definida no SLA de acordo com o módulo *SLA Analysis*. Adicionalmente, o módulo *Allocation Process* obtém a infraestrutura de rede do provedor (oriunda do módulo *Infrastructure Management*). A partir destas informações, um algoritmo de alocação define os componentes que farão parte da rede virtual requisitada.

Desta forma, este capítulo está organizado da seguinte forma: a Seção 6.1 contextualiza o cenário; a Seção 6.2 resume o método de cálculo de confiabilidade, a Seção 6.3 descreve o algoritmo base para alocação, e a Seção 6.4 apresenta os algoritmos de definição de caminho desenvolvidos no decorrer do trabalho, bem como os existentes na literatura.

### 6.1 Contexto

A partir do surgimento da virtualização de redes, surge a possibilidade dos ISPs customizarem o comportamento e topologia das redes de acordo com a necessidade especificada pelos clientes. Trabalhos como [27, 66, 81] usam essa flexibilidade para capacitarem os ISPs a proverem ao cliente o serviço desejado, adaptando a qualidade da rede de acordo com a demanda exigida pelos mesmos.

Os ISPs visam maximizar os lucros, e duas métricas são relacionadas à este aspecto [17, 68]: (i) número de clientes (SLA vigentes) e (ii) a energia consumida pela infraestrutura do ISP. Visto que quanto maior o número de clientes maior o lucro do provedor, pode-se maximizar o lucro dos provedores melhorando a utilização da largura de banda (em inglês *Bandwidth* – Bw) da infraestrutura de rede [6]. Por outro lado, o consumo de energia vem se tornando um ponto importante a ser considerado pelos ISPs. Recentemente, desencadeado pelo aumento no preço da energia, acordos para redução de poluição e expansão dos serviços oferecidos pelos ISPs, o princípio de eficiência energética se tornou um aspecto muito importante [11]. Em geral, eficiência energética refere-se à capacidade de prestar/realizar um serviço/tarefa utilizando energia de forma adequada. Portanto, no

contexto de redes virtuais, considera-se eficiência energética a quantidade de largura de banda alocada para os clientes pelo consumo de energia da infraestrutura de rede.

Além da maximização do lucro, os ISPs tem por objetivo cumprir os parâmetros do SLA negociado com o cliente, visto que estes parâmetros garantem a qualidade do serviço experienciada pelos usuários. Muitas aplicações necessitam de requisitos de rede diferenciados: largura de banda, confiabilidade da rede, etc. Dentro deste contexto, resiliência é um requisito chave para garantir QoS aos usuários. Resiliência é a capacidade da rede manter um nível especificado de qualidade mínimo quando falhas na operação usual da infraestrutura de rede ocorrem [75].

Resiliência engloba não somente ações reativas para gerenciar impactos pós-falha, mas também inclui planejamento estratégico para falhas. Arelado à resiliência, tem-se os conceitos de confiabilidade de rede e tolerância a tráfego como aspectos importantes a serem considerados durante a implantação das redes virtuais [61, 75, 48]. A confiabilidade é a probabilidade da rede estar operacional, ou seja, de conseguir interligar os roteadores de borda (entradas e saídas da rede) mesmo em caso de falha de algum componente (roteador, comutador ou enlace). De forma geral, a confiabilidade de rede aumenta quando caminhos alternativos são implantados. Da mesma forma, tolerância a tráfego é a capacidade da rede de lidar com os recursos alocados quando um aumento inesperado no volume de tráfego ocorre.

Consumo de energia e resiliência são aspectos importantes, entretanto possuem focos diferentes. O consumo de energia é minimizado quando poucos componentes de rede são utilizados, mas este fato pesa contra a confiabilidade da rede se não forem implantados caminhos alternativos para lidar com falhas. Portanto, é necessário considerar ambos os aspectos durante o gerenciamento e planejamento da implantação das redes virtuais.

A fim de implantar as redes virtuais, os ISPs precisam executar um algoritmo de alocação de redes virtuais, o qual decide quais componentes (enlaces e nós) da infraestrutura de rede devem fazer parte da rede virtual em questão. Para tornar o algoritmo desenvolvido eficiente e robusto, considera-se os aspectos de consumo de energia, bem como a junção deste com a largura de banda e confiabilidade, como critério de alocação de redes virtuais.

## 6.2 Confiabilidade da Rede

No contexto de alocação de redes virtuais, inicialmente a topologia da rede virtual é definida e posteriormente calcula-se a confiabilidade da rede. Sendo assim, um método para determinar a confiabilidade da rede pode ser aplicado para avaliar a rede virtual gerada considerando que cada componente (nó ou enlace) tem um valor de confiabilidade (probabilidade de estar operacional).

Neste trabalho foi aplicado o método proposto por Li et al [51] para determinar a probabilidade da rede continuar operacional em caso de falhas. Este método foi adaptado para o contexto de redes virtuais: o cálculo deve analisar a comunicação entre o nó do cliente e todos os nós destino definidos. Portanto, o método aplicado neste trabalho analisa a capacidade da rede virtual continuar provendo comunicação entre todos os nós

definidos quando ocorrem falhas na rede.

O método assume que, a qualquer momento, um componente de rede pode falhar aleatoriamente de acordo com uma probabilidade conhecida. Cada componente pode estar em um de dois estados: operacional ou falha. A falha de um enlace/link representa a remoção do mesmo da rede, enquanto que a falha de um nó representa a remoção do nó e todos os enlaces/links ligados ao mesmo.

Portanto, uma rede com  $n$  componentes pode estar em  $2^n$  estados. A probabilidade da infraestrutura de rede estar em um certo estado é dada pela Equação 6.1, onde tem-se a seguinte notação:  $S_k$  representa o possível estado  $k$ , onde  $k = 1, \dots, 2^n$ ;  $T_i(S_k)$  é 0 se o componente  $i$  está operacional no estado  $S_k$ , e 1 caso contrário;  $p_i$  é a probabilidade do componente  $i$  estar operacional; e  $q_i$  é a probabilidade de não estar operacional. Considerando-se que  $p_i > q_i \forall i$ , ou seja, a probabilidade de um componente estar operacional é maior que a probabilidade de falhar, o estado mais provável é  $S_1$ , o qual corresponde ao estado sem falhas.

$$P(S_k) = \prod_{i=1}^n p_i^{(q_i/p_i)^{T_i(S_k)}} \quad (6.1)$$

Adicionalmente, analisou-se os aspectos relacionados ao cálculo da confiabilidade da rede, onde identificou-se alguns *tradeoffs*: (i) entre o tamanho da rede e sua confiabilidade; bem como (ii) entre o tamanho da rede e o tempo para determinar a confiabilidade. Mais detalhes deste estudo e do método base [51] podem ser encontrados na referência [28].

Em relação ao primeiro *tradeoff*, quanto mais caminhos disjuntos interligando o nó do cliente e os destinos desejados, maior é a confiabilidade da rede. Contudo, a confiabilidade de um caminho diminui de acordo com o comprimento do mesmo, pois quanto mais componentes um caminho possui, maior é a probabilidade agregada de algum desses componentes falhar inutilizando o caminho definido.

Da mesma forma, o segundo *tradeoff*, entre o tamanho da rede e o tempo de cálculo, atesta que quanto maior a rede maior o número de componentes que são adicionados aos possíveis estados da rede. Isto faz com que o tempo de cálculo da confiabilidade cresça exponencialmente.

### 6.3 Algoritmo Base para Alocação

Esta seção apresenta o algoritmo usado como base para a alocação de redes virtuais. O Algoritmo 1 gera caminhos disjuntos, usando como critério de redundância o impacto da adição de novos enlaces nos três critérios considerados: (i) confiabilidade da rede; (ii) eficiência energética da rede; e (iii) largura de banda disponível na rede. A fim de facilitar a identificação das variáveis, a notação apresentada na Tabela 6.1 é usada ao decorrer deste capítulo para descrever o problema.

O peso/custo  $w_l$  é definido de acordo com o algoritmo de definição de caminho aplicado no problema (os quais serão descritos na Seção 6.4). Por exemplo,  $p = 0$  representa o caso sem redundância, ou seja, a topologia da rede virtual terá apenas um caminho para cada nó destino. Por outro lado,  $p = 1$  é o caso de redundância completa, ou seja, a

Tabela 6.1: Notação Utilizada

Símbolo	Descrição
$s$	nó fonte/raiz que representa o cliente
$D$	conjunto de destinos o qual deseja-se conectar
$L$	conjunto de enlaces/links da infraestrutura de rede
$N$	conjunto de nós da infraestrutura de rede
$l$	enlace/link entre dois nós
$w_l$	custo/peso do enlace $l$
$\wp$	quantidade perto de infinito
$\varepsilon$	quantidade próxima a zero
$e$	número de links a serem atualizados
$p$	percentual de redundância desejado ( $0 \leq p \leq 1$ )
$T_1$	árvore com os componentes de rede de $s$ para os nós em $D$
$G'$	grafo alternativo com $w_l$ atualizados
$T_2$	árvore os caminhos alternativos gerados a partir de $G'$
$G_f$	topologia final para a rede virtual (junção de $T_1$ e $T_2$ )
$Rr$	confiabilidade requisitada pelo cliente
$R$	confiabilidade da rede virtual $G_f$
$Bw_l$	largura de banda disponível no enlace $l$
$BwO_l$	largura de banda original do enlace $l$
$BwR$	largura de banda requisitada pelo cliente
$E_S$	energia gasta por um nó
$E_L(x)$	energia consumida por um enlace quando $x$ Mbps são alocados
$En_{Max}$	máximo de energia consumida pela infraestrutura de rede

topologia terá dois caminhos completamente disjuntos para cada nó destino. Da mesma forma,  $p = 0.5$  é o caso onde metade dos enlaces do caminho primário serão usados como base para a definição do caminho secundário.

Inicialmente, o Algoritmo 1 define um laço para iterar entre os possíveis fatores de redundância ( $p$ ). Dentro do laço, é encontrada uma árvore inicial  $T_1$  com o nó  $s$  como raiz executando-se o algoritmo de definição de caminho (função  $PathDefinition(Grafo, No)$ ). Esta função invoca um dos algoritmos que serão descritos na Seção 6.4.

Portanto,  $T_1$  contém os componentes de rede pertencentes aos caminhos de  $s$  para os nós contidos em  $D$ . A linha 4 atribui à  $e$  o número de enlaces a serem atualizados para gerar a redundância na topologia.  $e$  é calculado como o número de links em  $T_1$  ( $|T_1|$ ) de acordo com  $p$  ( $0 \leq p \leq 1$ ).

Após executar a função  $PathDefinition$ , o Algoritmo 1 atualiza o custo de cada enlace na rede, criando um novo grafo  $G'$ , o qual tem por objetivo evitar que os enlaces já alocados sejam evitados na busca por um caminho alternativo. Desta forma, o algoritmo substitui o peso  $w_l$  dos enlaces de acordo com a redundância, ou seja, ele substitui o peso  $w_l$  dos  $e$  primeiros enlaces de  $T_1$ . Enquanto o número de enlaces ( $e$ ) não é alcançado, o custo do enlace é substituído por  $\wp$  para evitar o seu uso, e após  $e$  enlaces serem processados, o custo dos enlaces é substituído por 0, a fim de encorajar a alocação do mesmo.

No próximo passo, o algoritmo encontra a árvore  $T_2$  no grafo  $G'$  (o qual possui o peso dos enlaces atualizados) a partir da execução da função  $PathDefinition$ . Posteriormente,

---

**Algoritmo 1** Geração de caminhos relativamente disjuntos para a rede virtual
 

---

```

1:  $p = 0$ ; ▷ Caso sem redundância
2: enquanto  $p \leq 1$  faça ▷ Redundância  $\neq$  Completa
3:    $T_1 = PathDefinition(G, s)$ ;
4:    $e = p * |T_1|$ ;
5:   para todo Enlace  $j \in T_1$  faça
6:     para todo Enlace  $i \in G$  faça
7:       se  $(i == j)$  então
8:         se  $(e > 0)$  então
9:            $w'_j = \varnothing$ ;
10:           $e = e - 1$ ;
11:        senão
12:           $w'_j = 0$ ;
13:        fim se
14:      fim se
15:    fim para
16:  fim para
17:   $T_2 = PathDefinition(G', s)$ ;
18:   $G_f = MergePaths(T_1, T_2)$ ;
19:  se  $(Bw_{Impact}(G_f) + En_{Impact}(G_f) < best)$  então
20:     $R = Reliability(G_f)$ ;
21:    se  $(R > Rr)$  então
22:       $best = Bw_{Impact}(G_f) + En_{Impact}(G_f)$ ;
23:    fim se
24:  fim se
25:   $Incrementar(p)$ ;
26: fim enquanto

```

---

o algoritmo faz a junção das árvores  $T_1$  e  $T_2$  (linha 11) para criar um grafo com os componentes de rede (enlaces e nós) que existem nas duas árvores, resultando na topologia final  $G_f$ .

Após a definição de  $G_f$ , o algoritmo verifica se esta é a melhor solução encontrada, ou seja, o algoritmo mede o impacto que a implantação de uma rede virtual com a topologia  $G_f$  terá na infraestrutura de rede, sobre as perspectivas de largura de banda disponível (função  $Bw_{Impact}(Grafo)$ ) e consumo de energia (função  $En_{Impact}(Grafo)$ ), descritas nas Equações 6.2 e 6.4, respectivamente.  $best$  é uma variável para identificar a topologia que possui menor impacto na largura de banda disponível e eficiência energética da rede. Se a “melhor” opção é encontrada, checa-se se  $G_f$  atende a confiabilidade requisitada pelo cliente no SLA ( $Rr$ ). A confiabilidade  $R$  de  $G_f$  é calculada de acordo com o método apresentado na Seção 6.2.

Finalizando o laço, a redundância desejada é incrementada para permitir que menos enlaces que foram utilizados no caminho primário sejam considerados na busca por um caminho alternativo. A quantidade do incremento é configurada pelo administrador da rede, por exemplo: 0.1, 0.25, ou 0.5 por iteração. Nos experimentos realizados, considerou-se um incremento de 0.25

$$Bw_{Impact}(G_f) = \sum_{l \in G_f} \frac{BwR}{Bw_l} \quad (6.2)$$

Com relação ao impacto da alocação de  $G_f$  na largura de banda, este cálculo é feito de acordo com a Equação 6.2. Por outro lado, o impacto da alocação de  $G_f$  no consumo de energia é medido de acordo com a Equação 6.4. Primeiramente, calcula-se o consumo de energia dos comutadores (do inglês *switch*) da rede, onde  $E_S(Switch)$  é a energia gasta pelo nó de acordo com a Equação (6.3) [53], e  $En_{Max}$  representa o máximo de energia consumida pela rede (ou seja, a energia consumida quando todos os componentes estão operacionais em máxima capacidade).

$$E_S(Switch\ o) = P_{ch} + (N_o * Pl) \quad (6.3)$$

Na Equação 6.3,  $P_{ch}$  é a energia consumida pelo chassi do *switch*;  $Pl$  é a energia consumida pelas portas de transmissão não ativadas, e  $N_o$  é o número de placas de transmissão conectadas ao *switch*  $o$ . De acordo com a referência [53],  $P_{ch}$  é 50 Watts e  $Pl$  é 40 watts.

$$En_{Impact}(G_f) = \sum_{o \in G_f} \frac{E_S(o)}{En_{Max}} + \sum_{l \in G_f} \frac{E_L(BwO_l + BwR - Bw_l) - E_L(BwO_l - Bw_l)}{En_{Max}} \quad (6.4)$$

Adicionalmente, a Equação 6.4 calcula o aumento na energia consumida nos enlaces devido ao aumento na alocação de largura de banda requisitada ( $BwR$ ) no mesmo. Lembrando que  $E_L(x)$  é a largura de banda consumida quando  $x$  Mbps são alocados no enlace de acordo com a Equação (6.5). Baseado na referência [53], assume-se que em geral para as velocidades de 10 Mbps, 100 Mbps e 1 Gbps, a energia consumida (em Watts) seja de 0.4, 0.5 e 1, respectivamente.

$$E_L(x) = \begin{cases} 0.4, & \text{se } 0 < x \leq 10\text{Mbps}; \\ 0.5, & \text{se } 10 < x \leq 100\text{Mbps}; \\ 1 & \text{se } 100 < x \leq 1\text{Gbps}; \\ 0, & \text{caso contrario}; \end{cases} \quad (6.5)$$

No geral, o Algoritmo 1 constrói uma árvore inicial e, a partir de uma atualização dos pesos dos enlaces, o algoritmo busca por caminhos alternativos através da adição de novos enlaces à árvore inicial. Esta adição é limitada pelo fator de redundância definido. A atualização do custo dos enlaces é usada para evitar o uso dos enlaces que já foram alocados, mas sem descartá-los como uma opção, forçando o algoritmo a procurar caminhos alternativos para alcançar os nós em  $D$ .

Executando o Algoritmo 1 gera-se uma topologia para redes virtuais com caminhos alternativos considerando aspectos de consumo de energia, largura de banda disponível e a confiabilidade requisitada pelo cliente. Entretanto, a topologia gerada ainda depende do algoritmo de definição de caminho aplicado. Portanto, foram propostos novos algoritmos considerando aspectos de consumo de energia, bem como a junção deste com a largura de banda.

## 6.4 Algoritmos de Definição de Caminho

A fim de buscar um conjunto de enlaces e nós que formem um caminho que conecte o nó fonte (cliente) aos nós destinos aplica-se um algoritmo de definição de caminho durante o processo de alocação de redes virtuais. Este algoritmo é representado pela função *PathDefinition* no Algoritmo 1.

Dentre os diversos aspectos que podem ser levados em consideração para a definição de um caminho entre dois ou mais nós, a Bw é bastante utilizada, pois é o recurso mais especificado nas negociações de SLA. Contudo, o consumo de energia tornou-se um aspecto a ser considerado pelos ISPs, devido ao aumento nos valores cobrados por consumo de energia e acordos de redução de poluição [11].

Sendo assim, os algoritmos *Max Bw Disponível* [56] e *Weighted Path* [28] são os algoritmos existentes mais correlacionados, pois focam na definição de caminhos que consideram a largura de banda na rede local, sendo apresentados nas Seções 6.4.1 e 6.4.2, respectivamente. Além disso, esta seção descreve os algoritmos propostos durante o desenvolvimento da tese, intitulados *Feasible-Bw*, *Bw-Risk-Ratio*, *Energy-Aware*, *BEE-Focus* e *DA-BEE*, a serem descritos nas Seções 6.4.3, 6.4.4, 6.4.5, 6.4.6 e 6.4.7, respectivamente.

Primeiramente, desenvolveu-se um algoritmo base para a definição de um caminho considerando uma métrica a ser aplicada. Os passos realizados são apresentados no Algoritmo 2. Sendo assim, o Algoritmo 2 utiliza o peso do enlace ( $w_l$ ) de acordo com o algoritmo em questão (*Bw-Risk-Ratio*, *Feasible-Bw*, *Energy-Aware*, *BEE-Focus* ou *DA-BEE*), onde cada algoritmo proposto possui uma abordagem particular e/ou alguma estratégia adicional.

A seguinte notação é usada no Algoritmo 2:  $S$  representa o conjunto de nós já analisados;  $S'$  é o conjunto de nós com análise ainda não realizada;  $W$  é uma lista de melhores caminhos para alcançar cada nó na rede a partir do nó  $s$ , onde  $W_i$  é o melhor caminho do

nó  $s$  ao nó  $i$ ; e as demais variáveis seguem a mesma notação aplicada ao decorrer deste relatório descritas na Tabela 6.1, como por exemplo  $s$  representando o nó fonte (cliente). A função *lower* retorna o nó que possui o menor custo na lista passada como parâmetro. Enquanto que a função *getLinkInfo* obtem as informações do enlace que conecta dois nós dados como parâmetro.

---

**Algoritmo 2** Algoritmo de definição de caminho
 

---

```

1: Conjunto  $S = \{s\}$ 
2: Conjunto  $S' = N - \{s\}$ ;
3: para todo No  $j \in S'$  faça
4:   Enlace  $l = getLinkInfo(s, j)$ ;
5:   se ( $w_l < \infty$ ) então
6:      $W_j = w_l$  ;
7:   senão
8:      $W_j = \infty$  ;
9:      $w_l = \infty$  ;
10:  fim se
11: fim para
12: enquanto  $S' \neq \emptyset$  faça
13:   No  $Min = lower(W)$  ;
14:    $S = S + Min$  ;
15:    $S' = S' - \{Min\}$ ;
16:   para todo No  $j \in S'$  faça
17:     Enlace  $l = getLinkInfo(Min, j)$ ;
18:     se ( $W(j) > W(Min) + w_l$ ) e ( $Bw_l > BwR$ ) então
19:        $W(j) = W(Min) + w_l$ ;
20:     fim se
21:   fim para
22: fim enquanto

```

---

Entre as linhas 1 e 10, as informações referentes aos enlaces entre o nó  $s$  e os demais da topologia são coletadas usando a função *getLinkInfo*( $No, No$ ). Cada enlace possui seu próprio custo/peso atribuído de acordo com a Equação (6.11). Portanto, caso um enlace direto exista, utiliza-se o peso do mesmo ( $w_l$ ), ou atribui-se  $\infty$  caso contrário (enlace inexistente). A partir da linha 11 até a linha 21, o algoritmo percorre os nós existentes verificando se o nó que possui o menor custo em relação ao nó  $s$  (de acordo com a função *lower*) pode ser usado como caminho menos custoso para alcançar outros nós (linhas 15 até 20).

### 6.4.1 Weighted Path

O algoritmo *Weighted Path*, visa alocar o caminho com menor número de saltos que satisfaça a largura de banda desejada. O algoritmo pune os enlaces que não possuem a largura de banda desejada e considera adequados os enlaces que possuem. Desta forma, este algoritmo não considera o estado da infraestrutura para definir os melhores componentes de rede a serem alocados. *Weighted Path* aplica o custo/peso do enlace apresentado na Equação 6.6.

$$w_l = \begin{cases} 1, & \text{se } BwR \leq Bw_l; \\ \infty, & \text{caso contrario;} \end{cases} \quad (6.6)$$

### 6.4.2 Maximum Available Bandwidth

O algoritmo *Max Bw Disponível* encontra o caminho com maior largura de banda disponível, independente do número de enlaces utilizados e a Bw requisitada pelo cliente. Portanto, este algoritmo não é baseado na ideia de acumulação de valores (custo dos enlaces), ele encontra o caminho com maior capacidade de fluxo (Bw Disponível). Portanto, o algoritmo *Max Bw Disponível* não aplica pesos aos enlaces, ele apenas usa o montante de Bw disponível como critério no processo de busca, como ilustrado na Equação 6.7.

$$w_l = Bw_l \quad (6.7)$$

### 6.4.3 Feasible-Bw

O algoritmo *Feasible-Bw* considera a largura de banda disponível nos enlaces para medir quão adequado os enlaces são em relação a uma requisição específica. A ideia deste algoritmo é maximizar o número de requisições de redes virtuais a partir de uma distribuição de alocação por toda a infraestrutura de rede, tentando assim balancear a largura de banda disponível nos enlaces.

Sendo assim, o custo/peso de cada enlace  $l$  é definido através da Equação 6.8, onde  $Bw_l$  é o valor atual disponível de capacidade de recurso do enlace e  $BwR$  é a largura de banda requisitada pelo cliente. Por exemplo, se o cliente requisita 10 Mbps e o enlace possui somente 8 Mbps disponíveis, o mesmo obtém um peso de 3.

$$w_l = 1 + 10 \left( 1 - \frac{Bw_l}{BwR} \right) \quad (6.8)$$

Esta estratégia visa priorizar o uso de enlaces que conseguem atender a demanda pedida pelo cliente, sendo que os enlaces sem capacidade disponível ( $Bw_l = 0$ ) são retirados do cálculo da topologia. Da mesma forma, os valores de  $w_l$  negativos são convertidos para o valor 1, representando a adequação do enlace à largura de banda requisitada.

### 6.4.4 Bw-Risk-Ratio

A fim de considerar um risco de falha diferente para componentes de rede que se situam em regiões geograficamente distintas, o algoritmo *Bw-Risk-Ratio* foi proposto. Este algoritmo tem por objetivo alocar o menor caminho com a maior quantidade de Bw disponível e menor risco relacionado a fatores externos (por exemplo, desastres naturais), assim economizando largura de banda e diminuindo o número de componentes alocados. No algoritmo *Bw-Risk-Ratio*, o peso dos enlaces segue a Equação 6.9, onde  $RiskFactor_i$  é o fator de risco definido para o evento  $i$ , de acordo com o modelo de risco a ser descrito a seguir.

$$w_l = \log \left( \frac{Bw_l}{BwR} \right) + \frac{\sum_{i=1}^n RiskFactor_i}{n}. \quad (6.9)$$

A infraestrutura de rede é sensível a certos riscos. A fim de determinar a confiabilidade da região a qual os componentes de rede (enlaces e roteadores/comutadores) se encontram é proposto um modelo de risco baseado em lógica Fuzzy.

O modelo de risco possui duas funções de entrada, *Occurrence* e *Impact*, e uma função de saída *Risk Factor*. *Occurrence* refere-se à frequência que um certo evento ocorre na região em questão. *Impact* refere-se aos danos que um certo evento causa à infraestrutura de rede. Seguindo os princípios de lógica Fuzzy, definiu-se quatro variáveis linguísticas: *None*, *Low*, *Medium*, e *High*. O conjunto de regras é apresentado na Tabela 6.2, e expressa as possíveis variáveis linguísticas de saída de acordo com as variáveis de entrada.

Tabela 6.2: Conjunto de Regras do Modelo de Risco

Occurrence	Operação	Impact	Risk Factor
High	e	High	High
High	e	Medium	High
High	e	Low	Medium
Medium	e	High	High
Medium	e	Medium	Medium
Medium	e	Low	Low
Low	e	High	High
Low	e	Medium	Low
Low	e	Low	Medium
None	ou	None	None

Neste modelo de risco, as variáveis linguísticas do *Risk Factor* possuem os seguintes valores: *None* é 0, *Low* é 1, *Medium* é 2, e *High* é 3. Isto significa, por exemplo, se um link está em uma região que possui um evento de risco com *Occurrence Medium* e *Impact Low*, este terá um *Risk Factor Low*, ou seja, um peso de 1, o qual contribuirá ao peso custo do enlace (Equação 6.9).

O modelo de risco proposto permite a identificação dos componentes de rede que possuem maior suscetibilidade a falhas do que outros, habilitando a alocação da rede virtual ser feita a partir de regiões de risco distintas. Isto faz a rede virtual mais resiliente, visto que caso um desastre ocorra em uma região, ao menos um caminho alternativo pode existir na rede virtual alocada. Portanto, o cliente teria uma qualidade mínima vinculada ao SLA.

A seguir, é mostrado um exemplo de aplicação do modelo de risco para a rede Internet2<sup>1</sup>. A configuração do modelo de risco é baseada nas informações estatísticas sobre ocorrências de falhas por desastres naturais e sobre o impacto destas falhas nos EUA apresentadas pelas referências [58, 63, 76], incluindo a categorização dos eventos. Portanto, modelou-se quatro eventos para o modelo de risco:

<sup>1</sup>internet2.edu/

- Geophysical: Low *Occurrence* e High *Impact* –  $RiskFactor_{Geophysical}$  is 3 (High *Risk Factor*);
- Meteorological: High *Occurrence* e High *Impact* –  $RiskFactor_{Meteorological}$  is 3 (High *Risk Factor*);
- Hydrological: High *Occurrence* e Medium *Impact* –  $RiskFactor_{Hydrological}$  is 3 (High *Risk Factor*);
- Climatological: Low *Occurrence* e Low *Impact* –  $RiskFactor_{Climatological}$  is 1 (Low *Risk Factor*).

Adicionalmente, a topologia da rede Internet2 foi segmentada em regiões, onde cada região é suscetível a um conjunto de eventos particular. A segmentação aplicada, seguindo as informações presentes nas referências [58, 63, 76], é resumida na Tabela 6.3, enquanto a Figura 6.1 ilustra a mesma.

Tabela 6.3: Informações sobre o Modelo de Risco

Região	Nós	Eventos	<i>Risk Score</i>
A	Seattle, Portland, Sunnyvale, Los Angeles e San Diego.	Meteorological e Geophysical.	1.5
B	Boise e Salt Lake City.	Meteorological e Climatological.	1
C	Albuquerque e El Paso.	Meteorological.	0.75
D	Kansas e Denver.	Meteorological e Climatological.	1
E	Houston, Baton Rouge e Jacksonville.	Meteorological e Hydrological.	1.5
F	Chicago, Indianapolis e Louisville.	Meteorological, Hydrological, e Geophysical.	2.25
G	Nashville, Atlanta, Charlotte e Raleigh.	Meteorological e Hydrological.	1.5
H	Cleveland, Washington, Boston, New York e Philadelphia.	Meteorological e Hydrological.	1.5

A partir deste exemplo de utilização do modelo de risco pode-se considerar a confiabilidade dos componentes de rede em suas respectivas regiões no algoritmo de definição de caminho apresentado no Algoritmo 1.

#### 6.4.5 Energy-Aware

A fim de considerar o consumo de energia da infraestrutura de rede, o algoritmo *Energy-Aware* aloca a rede virtual com caminhos que possuem menor impacto na eficiência energética, economizando energia e evitando o uso de caminhos que usem muitos componentes de rede ainda não ativos. Portanto, o algoritmo *Energy-Aware* não foca na redução do consumo de energia em um único componente de rede, e sim foca em gerenciar e planejar as alocações de redes virtuais a fim de melhorar a eficiência energética do ISP.

A eficiência energética do ISP é definida de acordo com a Equação (6.10), onde  $A$  é o conjunto de redes virtuais ativas,  $BwR_a$  é a largura de banda requisitada ( $BwR$ ) por uma

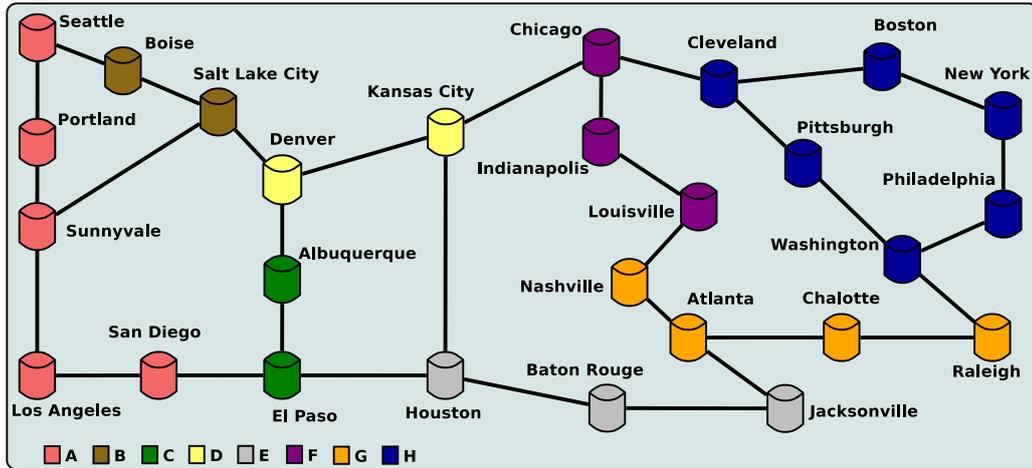


Figura 6.1: Divisão das regiões

requisição  $a$  e  $En_C$  é o consumo de energia atual do ISP. Portanto, eficiência energética mostra se o consumo de energia da rede é bem utilizado (maiores valores) ou não.

$$En_{Eff} = \sum_{a \in A} \frac{BwR_a}{En_C} \quad (6.10)$$

O algoritmo *Energy-Aware* considera a existência de dois aspectos na infraestrutura de rede: (i) os componentes de rede podem ser ligados e desligados de forma seletiva e independente e (ii) as capacidades de encaminhamento dos enlaces podem ser ajustadas dinamicamente de acordo com a largura de banda alocadas nos mesmos. Baseado nestas considerações, o algoritmo *Energy-Aware* usa a Equação (6.11) como peso/custo dos enlaces no processo de busca. O valor de  $E_L(x)$  é definido de acordo com a Equação (6.5) mostrada anteriormente.

$$w_l = \frac{E_L(Bw_l)}{E_L(Bw_l + BwR)} \quad (6.11)$$

A ideia é mensurar o impacto da alocação de largura de banda para a rede virtual requisitada pelo cliente na eficiência energética do ISP. Portanto, os enlaces que já estão em uso geram um menor impacto no consumo de energia.

#### 6.4.6 BEE-Focus

Tanto a largura de banda disponível quanto o consumo de energia da infraestrutura de rede possuem sua importância para os ISPs. Portanto, a incorporação de ambos os aspectos na alocação de redes virtuais tende a gerar mais benefícios aos ISPs, visto que quanto maior a largura de banda disponível, maior a capacidade do ISP de negociar novos SLAs. Da mesma forma, quanto maior a eficiência energética do ISP, menor os custos de energia gerados.

Sendo assim, foi desenvolvido o algoritmo *Bandwidth and Energy Efficiency Focus* (BEE-Focus) que visa definir caminhos com alta disponibilidade de largura de banda, bem como menor impacto na eficiência energética. O algoritmo *BEE-Focus* define o

custo/peso dos enlaces da topologia de acordo com a Equação 6.12, resultando na definição de caminhos que considerem tanto a largura de banda disponível quanto a energia consumida pelo enlace.

$$w_l = \log \left( \frac{Bw_l}{BwR} \right) + \frac{E_L(Bw_l)}{E_L(Bw_l + BwR)} \quad (6.12)$$

### 6.4.7 DA-BEE Algorithm

O estado da infraestrutura de rede do ISP muda ao decorrer do tempo, onde informações como a largura de banda disponível e eficiência energética podem mudar. Dependendo do estado atual da rede, faz-se necessário adaptar os critérios utilizados durante a alocação das redes virtuais.

Um exemplo dessa situação descrita ocorre quando a eficiência energética do ISP está baixa, sendo interessante dar uma maior prioridade a este aspecto. Da mesma forma, se a largura de banda disponível da rede está diminuindo, esta torna-se um fator crítico, a qual deve ter uma maior atenção no processo de alocação.

Dentro deste contexto, desenvolveu-se o algoritmo *Dynamic state of Bandwidth and Energy Efficiency* (DA-BEE), o qual adapta a importância dada aos critérios de eficiência energética e largura de banda de acordo com o estado atual da infraestrutura de rede do ISP. O objetivo é tornar a alocação das redes virtuais flexível e mais adequada ao estado atual do ISP.

No algoritmo *DA-BEE*, o peso dos enlaces de rede são definidos de acordo com a Equação (6.13), onde  $W_{En}$  representa o peso/importância (em um intervalo de 0 a 1) para o critério de eficiência energética e  $E_L(x)$  é definido de acordo com a Equação (6.5) apresentada na Seção 6.3.

$$w_l = W_{En} * \left( \frac{E_L(Bw_l)}{E_L(Bw_l + BwR)} \right) + (1 - W_{En}) * \left( \log \left( \frac{Bw_l}{BwR} \right) \right) \quad (6.13)$$

A atualização da importância  $W_{En}$  é feita de acordo com o Algoritmo 3.  $Bw_{AA}$  é o percentual da largura de banda média disponível na infraestrutura de rede do ISP, descrito na Equação 6.14, e  $En_{Eff}$  representa a eficiência energética do ISP, a qual é calculada a partir da Equação (6.10) descrita na Seção 6.4.5.

$$Bw_{AA} = \sum_{l \in G} \frac{Bw_l}{|L|} \quad (6.14)$$

Inicialmente, o valor de  $W_{En}$  é 0.5 a fim de prover uma importância equivalente a ambos os critérios. Conforme as redes virtuais são alocadas, o estado da rede vai variando, com isso o valor de  $W_{En}$  é decrementado (linha 2) ou incrementado (linha 5). Quando o estado da rede é considerado estável, o valor de  $W_{En}$  é ajustado para 0.5 novamente (linha 7) para balancear os critérios durante o processo de definição de caminho. Previamente a cada definição de rede virtual pelo Algoritmo 1, executa-se o Algoritmo 3 para atualizar  $W_{En}$  e assim realizar a busca de acordo com o estado atual do ISP.

**Algoritmo 3** Atualização do estado da rede

---

```

1: se  $Bw_{AA} < 0.5$  e  $W_{En} > 0.2$  então
2:    $W_{En-} = 0.1$ ;
3: senão
4:   se  $En_{Eff} < 0.5$  e  $W_{En} < 0.8$  então
5:      $W_{En+} = 0.1$ ;
6:   senão
7:      $W_{En} = 0.5$ ;
8:   fim se
9: fim se

```

---

## 6.5 Experimentos

Para avaliar os algoritmos propostos, foi desenvolvido um simulador de alocação de redes virtuais<sup>2</sup> a fim de analisar os aspectos relacionados a esta tarefa. Sendo assim o simulador engloba as seguintes tarefas:

1. Carregar as informações da infraestrutura de rede;
2. Gerenciar a infraestrutura e disponibilidade de recursos;
3. Definir as redes virtuais para atender um conjunto de requisições, de acordo com o algoritmo de alocação escolhido;
4. Verificar o estado de conectividade da rede quando falhas ocorrem.

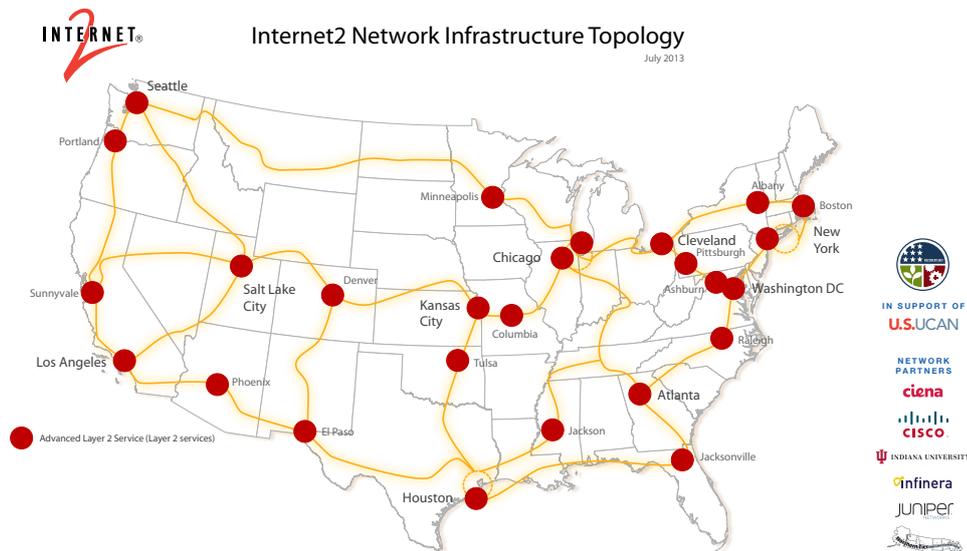


Figura 6.2: Topologia da rede Internet2.

O experimentos realizados utilizaram a topologia de rede real Internet2, ilustrada na Figura 6.2. Onde todos os algoritmos descritos na Seção 6.4 foram avaliados. A

<sup>2</sup><http://bitbucket.org/rafaellgom/vn-allocation/>

configuração do modelo de risco do algoritmo *Bw-Risk-Ratio* foi a mesma apresentada na Seção 6.4.4. Os experimentos visam avaliar a capacidade dos algoritmos de: (a) resolver um maior número de requisições de redes virtuais, (b) melhor utilizar os recursos de rede, (c) apresentar eficiência energética e (d) gerar as redes virtuais de tal modo que possam ser resilientes às falhas.

Foram geradas aleatoriamente 100 conjuntos de requisições e cada conjunto foi composto de 100 requisições, aplicando-se um intervalo de confiança de 95%. Os parâmetros de cada requisição foram: (i) nó do cliente (nó raiz/fonte); (ii) conjunto de nós destinos; (iii) a confiabilidade desejada para a rede; (iv) duração da requisição; e (v) a Bw requisitada (valores entre 10 e 100 Mbps).

Em geral, assume-se que nos modelos de tráfego de rede que o intervalo de chegada e duração dos fluxos, resultando na demanda de tráfego, seguem uma distribuição exponencial [16]. Sendo assim, os valores de largura de banda requisitada e a duração das requisições são gerados a partir de funções exponenciais. Com relação à infraestrutura de rede, cada link da topologia foi configurado com 1Gbps de Bw disponível. Os resultados dos experimentos realizados são mostrados a seguir com um intervalo de confiança de 95%.

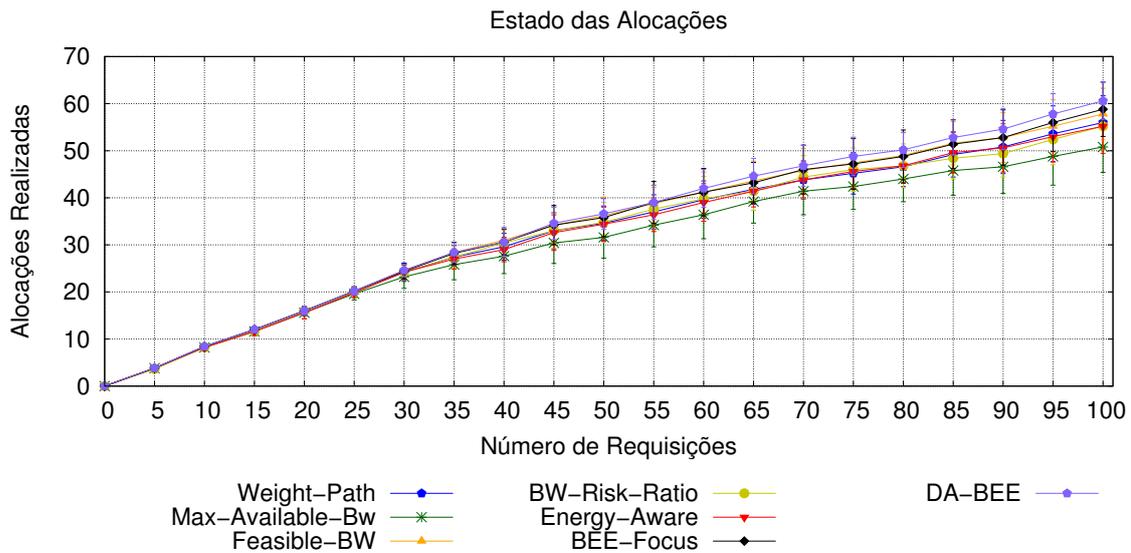


Figura 6.3: Alocações realizadas com sucesso.

Os resultados referentes ao estado das requisições (sucesso ou falha) são mostrados na Figura 6.3. Esta mostra o número acumulativo de requisições resolvidas. Baseado nisso, observa-se que todos os algoritmos tem um comportamento parecido até a 25<sup>a</sup> requisição. Neste ponto, os recursos da rede começam a se tornar escassos, devido às alocações anteriores.

A Figura 6.4 ilustra a largura de banda média restante em todos os enlaces da rede ao longo da análise das requisições. Dentre os algoritmos analisados, os algoritmos *Feasible-Bw*, *BEE-Focus* e *DA-BEE* são os que possuem menos consumo de largura de banda para atender um número maior de requisições.

A Figura 6.5 mostra os valores de eficiência energética da rede durante o processo de alocação das redes virtuais requisitadas (descrito na Equação 6.10). Baseado na Fi-

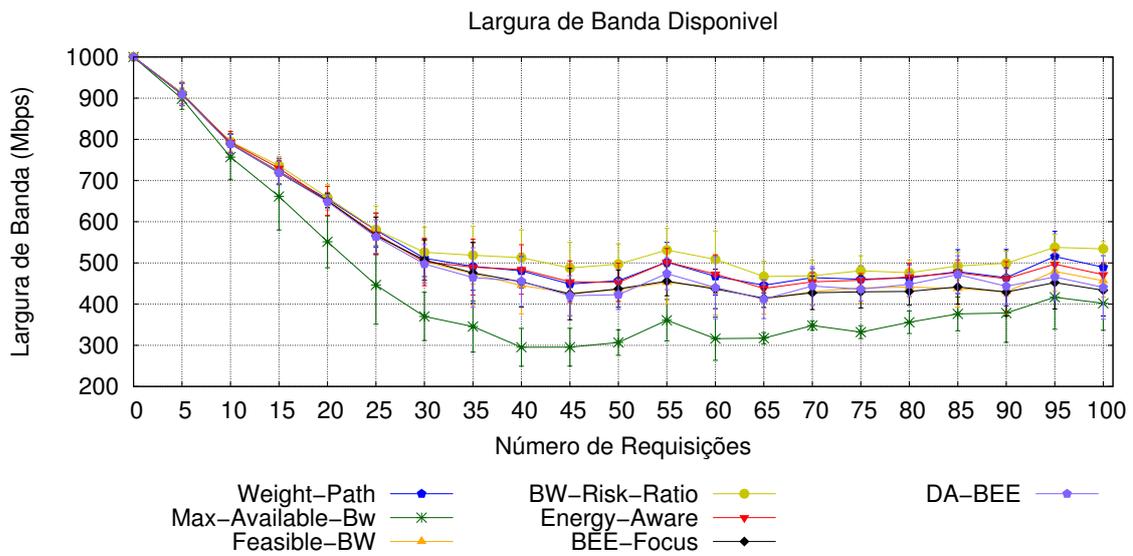


Figura 6.4: Largura de banda disponível.

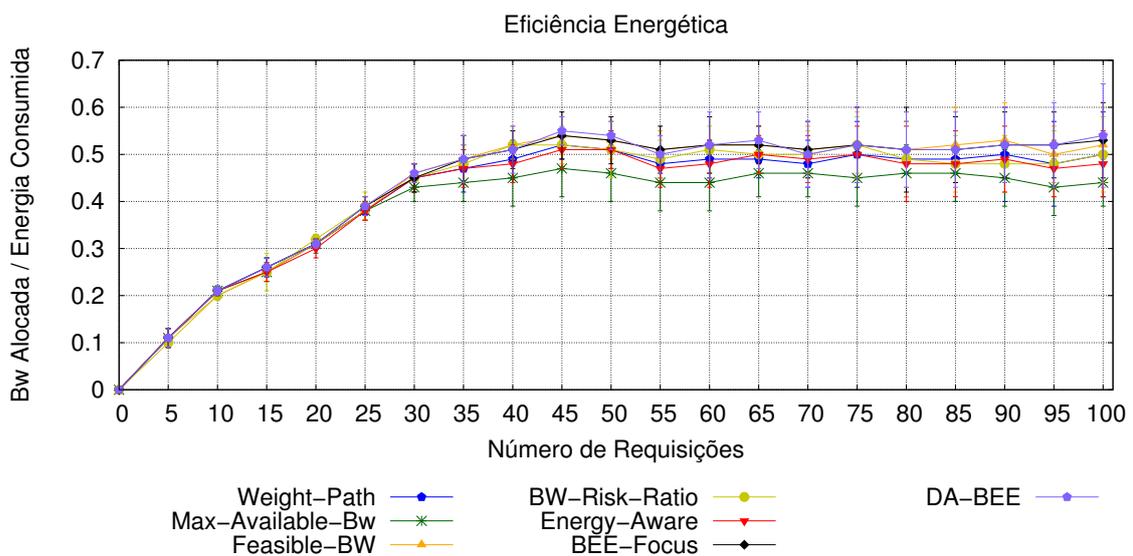


Figura 6.5: Eficiência energética da rede.

Figura 6.5, pode-se ver que os algoritmos *Feasible-Bw*, *BEE-Focus* e *DA-BEE* melhoram a eficiência energética da rede quando comparado aos demais algoritmos.

A fim de avaliar os algoritmos sobre uma perspectiva pós-falha, a Figura 6.6 apresenta o percentual médio de conectividade das redes ativas quando ocorrem falhas na rede. O percentual é definido de acordo com o número de destinos descritos na requisição. Por exemplo, se a rede virtual visa interligar o cliente com quatro nós e após a falha três continuam comunicáveis, então o percentual de comunicação é de 0.75.

Baseando-se nas informações apresentadas na Figura 6.6, percebe-se que a estratégia de confiabilidade desenvolvida (Algoritmo 1) consegue alcançar seu objetivo, visto que mesmo após falhas as redes virtuais ainda conseguem prover comunicação independente do algoritmo de definição de caminho utilizado.

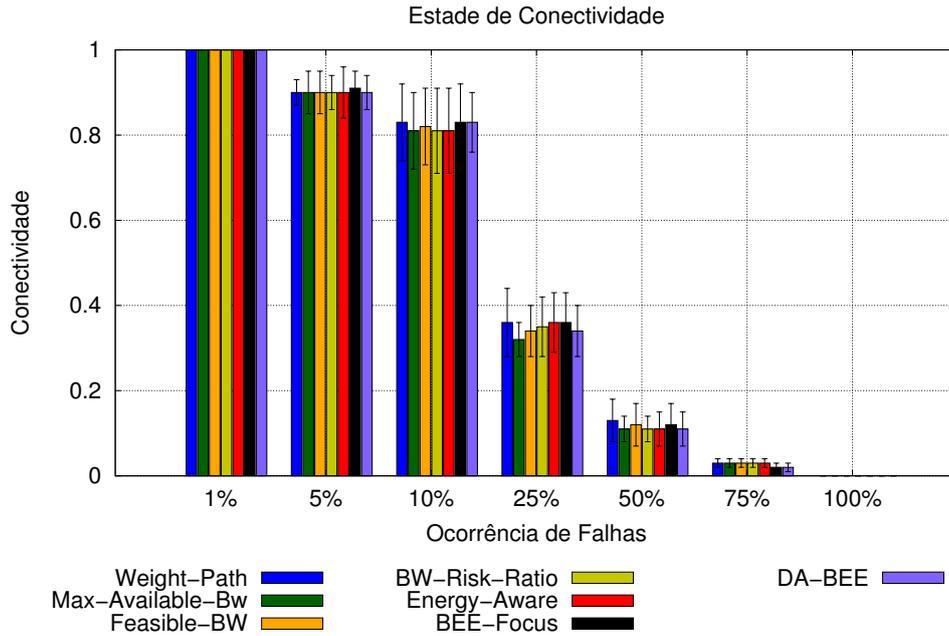


Figura 6.6: Estado de comunicação pós-falha.

Em geral, os experimentos realizados mostram que os algoritmos propostos alcançam os seus objetivos, incorporando benefícios ao ISP em relação aos aspectos de utilização dos recursos da rede e eficiência energética. O algoritmo *DA-BEE* apresenta, a partir de todos os aspectos analisados, o melhor desempenho em relação ao uso dos recursos e da infraestrutura de rede.

Por outro lado, apesar de ser mais simples, não considerando o estado atual da infraestrutura de rede e múltiplos critérios (largura de banda e consumo de energia), o algoritmo *Feasible-Bw* conseguiu atingir um desempenho próximo do algoritmo *DA-BEE*. Portanto, em cenários onde não se pode avaliar o estado atual da rede e obter informações referentes à eficiência energética do provedor, o algoritmo *Feasible-Bw* aparece como uma opção viável para alcançar um bom desempenho no processo de alocação de redes virtuais.

# Capítulo 7

## Adaptação de Redes Virtuais Definidas por Software

Este capítulo apresenta os mecanismos de ajuste desenvolvidos para serem aplicados no módulo *Adjustment Mechanism*. Como base para os mecanismos propostos, tem-se as informações providas pelos módulos *Passive Monitoring* e *SLA Analysis*. E, caso seja identificado que a alocação dos recursos atuais para a rede virtual não representam mais o estado da rede, ajusta-se os recursos alocados através do módulo *Resource Allocation*.

### 7.1 Contexto

Atualmente, muitas empresas utilizam a Internet como base para a prestação dos seus serviços. Adicionalmente, o acesso a Internet pelos usuários finais aumentou, onde estes esperam compartilhar/baixar conteúdo a todo momento. Este fato cria uma demanda de recursos variáveis, principalmente quando o cliente em questão está associado a dispositivos móveis, em redes onde a entrada e saída de dispositivos é dinâmica.

Sendo assim, há a necessidade de adaptar as redes virtuais implantadas inicialmente, tornando-as adequadas à demanda atual do cliente. Dentro deste contexto, tem-se dois objetivos principais: (i) evitar desperdício de recursos do ISP em casos de subutilização dos recursos pelo cliente, visto que para o cliente isso representa custo financeiro além do necessário, enquanto que para o ISP simboliza menos recursos disponíveis para futuras negociações com outros clientes; e, (ii) impedir uma queda na qualidade do serviço vivenciada pelo cliente, pois uma superutilização dos recursos da rede tende a gerar situações de grande perda de pacotes e alto atraso na comunicação.

O mecanismo de ajuste avalia as seguintes informações: a situação corrente da infraestrutura de rede do provedor, as especificação vigente do SLA, as características da rede virtual implantada e o volume de tráfego do cliente. A partir disso, define-se o ajuste a ser realizado nos recursos alocados. Sendo assim, foram desenvolvidos dois mecanismos de ajuste apresentados nas Seções 7.2 e 7.3.

## 7.2 RAAND: Resource Adjustment According to Network Demand

Esta seção descreve o mecanismo de ajuste chamado *Resource Adjustment According to Network Demand* (RAAND). O mecanismo RAAND visa definir se é necessário requisitar um ajuste baseado na demanda de tráfego mensurada.

Além da demanda de tráfego, o mecanismo RAAND trabalha com o conceito de limite de ação. Esses limites são os valores máximo e mínimo que o mecanismo RAAND pode requisitar. O valor mínimo representa os recursos definidos inicialmente no SLA. Por outro lado, o valor máximo é o limite superior que pode-se requisitar, seja por disponibilidade de recursos no ISP ou por custo financeiro máximo desejado pelo cliente, visto que o preço pago é proporcional aos recursos alocados.

A seguinte notação é utilizada para descrever o mecanismo:  $C$  é a demanda de tráfego atual;  $S_d$  é a quantidade de recursos alocados de acordo com o SLA;  $L_{MAX}$  e  $L_{MIN}$  são os limites máximo (recursos disponíveis na rede) e mínimo (valor inicial definido no SLA), respectivamente;  $T$  é o percentual dos recursos alocados  $S_d$  que podem ser tolerados acima ( $T_H = S_d * (1 + T)$ ) ou abaixo ( $T_L = S_d * (1 - T)$ );  $Time_T$  é tempo máximo que a rede virtual pode ficar fora do intervalo de  $T$  (maior ou menor), ou seja, é uma referência de tempo que indica se a demanda de tráfego está diferente dos recursos alocados por um longo tempo.

Portanto, o mecanismo RAAND considera a rede virtual em um dos seguintes estados/situações, de acordo com o valor de  $C$  e as demais configurações ( $T$ ,  $S_d$ ,  $L_{MAX}$  e  $L_{MIN}$ ):

- *Congestion*:  $S_d > T_H$ ;
- *Alert*:  $S_d * (1 + \frac{T}{2}) < C \leq T_H$ ;
- *Feasible*:  $S_d * (1 - \frac{T}{2}) \leq C \leq S_d * (1 + \frac{T}{2})$ ;
- *Idle*:  $T_L \leq C < S_d * (1 - \frac{T}{2})$ ;
- *Waste*:  $C < T_L$ .

O mecanismo RAAND aplica o Algoritmo 4 para identificar o estado atual da rede virtual e calcular um possível ajuste. A variável  $Adj$  representa o valor a ser requisitado ao ISP, sendo que quando  $Adj$  é zero, a requisição de ajuste não é feita (estado de *Feasible*).

Basicamente, o ajuste é relacionado ao estado atual da rede virtual, onde estados considerados críticos (*Congestion* e *Waste*) resultam em uma mudança maior nos recursos de rede alocados. Por outro lado, quando a demanda de tráfego não está distante da definição atual do SLA, um pequeno ajuste é requisitado. Por final, o algoritmo verifica se o ajuste está de acordo com os limites de ação definidos ( $L_{MAX}$  e  $L_{MIN}$ ).

**Algoritmo 4** Algoritmo de ajuste do RAAND

---

```

1:  $Adj = 0$ ; ▷ Ajuste a ser requisitado
2: se  $(S_d * (1 - \frac{T}{2}) \leq C \leq S_d * (1 + \frac{T}{2}))$  então
3:   Feasible: um ajuste não é necessário.
4: senão
5:   se  $C > S_d$  então
6:     se  $C < S_d * (1 + \frac{T}{2})$  então
7:       se Longo período ( $Time_T$ ) de estado de Alert então
8:          $Adj = T_H$ ;
9:       fim se
10:    senão
11:      Estado de Congestion:  $Adj = C * (1 + T)$ ;
12:    fim se
13:  senão
14:    se  $C > T_L$  então
15:      se Longo período ( $Time_T$ ) de estado de Idle então
16:         $Adj = T_L$ 
17:      fim se
18:    senão
19:      Estado de Waste:  $Adj = C * (1 + T)$ 
20:    fim se
21:  fim se
22:  Colocar  $Adj$  dentro do limite ( $L_{MIN} \leq Adj \leq L_{MAX}$ );
23: fim se

```

---

### 7.3 BEAVER: Bitrate and Experience Aware adaptation of Virtual Edge Resources

Atualmente, a Internet faz parte da vida das pessoas. Como consequência, o acesso a conteúdo multimídia via Internet cresceu nos últimos anos. A qualidade vivenciada pelos usuários em relação ao tráfego multimídia é chamada Qualidade de Experiência (*Quality of Experience* - QoE) [70]. Diferente do tráfego de dados, o tráfego multimídia é mais sensível a perda e atraso de acordo com a resolução do mesmo [54].

Sendo assim, foi proposto um mecanismo de ajuste voltado para tráfego multimídia. Nesta tese, foi proposto anteriormente um classificador de tráfego, o qual permite a aplicação de um mecanismo de ajuste em uma rede virtual voltada para tráfego multimídia.

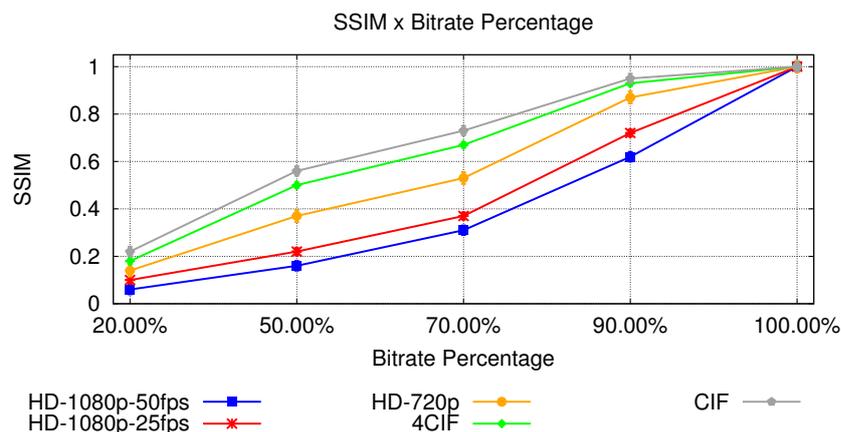
A fim de desenvolver este mecanismo, foi realizado um estudo para mensurar o impacto na qualidade do vídeo de acordo com os recursos de rede disponíveis. Esta qualidade pode ser mensurada a partir de métricas de QoE. Neste projeto foi utilizada a métrica *Structural SIMilarity* (SSIM), a qual é utilizada amplamente para análise de QoE [24, 45, 74, 65]. Um vídeo é considerado aceitável se possui SSIM maior que 0.7, e é considerado bom se possui SSIM maior que 0.8 [70].

Sendo assim, o estudo realizado verificou qual o percentual do *bitrate* do vídeo é necessário para manter boa qualidade ao usuário de acordo com a resolução do vídeo.

Os vídeos utilizados e suas respectivas características são resumidos na Tabela 7.1. Os vídeos CrowndRun, DanceKiss, Flagshoot, Crew, Harbour, Soccer, City, Foreman, Paris, News, e Akiyo estão disponíveis em Xiph.org<sup>1</sup> e são amplamente utilizados na literatura [24, 45, 74, 65]. Por outro lado, os vídeos Libertadores-2012<sup>2</sup>, Puskas-2013<sup>3</sup>, e 2014-Fifa<sup>4</sup> est] estão disponíveis no Youtube e foram escolhidos devido a aplicabilidade no contexto de transmissões de tempo real. O resultado do estudo é mostrado na Figura 7.1.

Tabela 7.1: Informações sobre os Vídeos

Video	Duração (segundos)	Bitrate(Kbps)	Fps	Resolução
CrowndRun	10	10900	50	HD 1080p
DanceKiss	10	10500	50	HD 1080p
ParkJoy	10	10300	50	HD 1080p
Flagshoot	20	7700	25	HD 1080p
Libertadores2012	10	3000	25	HD 720p
Puskas-2013	15	2500	30	HD 720p
2014-Fifa	50	2400	30	HD 720p
Crew	10	770	30	4CIF
Harbour	10	710	30	4CIF
Soccer	10	660	30	4CIF
City	10	550	30	4CIF
Foreman	10	150	30	CIF
Paris	35	120	30	CIF
News	10	90	30	CIF
Akiyo	10	70	30	CIF


 Figura 7.1: Relação entre *bitrate* e QoE.

Comparando-se os vídeos de alta definição (HD-1080p e HD-720p) com os vídeos de resolução inferior (4CIF e CIF), os vídeos HD são mais sensíveis a perda de pacotes. Como resultado a qualidade dos vídeos de alta resolução, de acordo com a métrica SSIM, tende

<sup>1</sup>media.xiph.org/video/derf/

<sup>2</sup>youtube.com/watch?v=tB04NBjyh5U

<sup>3</sup>youtube.com/watch?v=RDsxKXa0PbA

<sup>4</sup>www.youtube.com/watch?v=XpuYttdxFF8

a indicar um comportamento exponencial. Enquanto que o comportamento de queda na qualidade dos vídeos de resolução inferior aparenta ser logarítmico.

A partir do estudo, foi proposta a função *Bitrate-Aware Resources Suitability* (BARS) para expressar a adequabilidade dos recursos de rede alocados sobre a qualidade do vídeo. Para modelar essa relação foi aplicada uma regressão linear, [38], a qual relaciona duas variáveis a uma equação a partir de dados observados.

Portanto, a função BARS, apresentada na Equação (7.1), relaciona o valor de SSIM (qualidade do vídeo) com a adequabilidade dos recursos de rede alocados (percentual de largura de banda). A equação expressa o percentual necessário de um *bitrate*  $x$  para manter uma boa qualidade de vídeo. O *bitrate* de um vídeo pode ser estimado de acordo com o volume de tráfego gerado pelo fluxo de vídeo.

$$BARS(x) = (0.799 * x^{0.0742}) \quad (7.1)$$

Após a definição da função BARS, foi possível desenvolver um mecanismo de ajuste voltado ao tráfego multimídia, chamado *Bitrate and Experience Aware adaptation of Virtual Edge Resources* (BEAVER). Este mecanismo visa identificar os fluxos de vídeos e, a partir da aplicação da função BARS, definir se é necessário aplicar algum ajuste nos recursos de rede alocados. O comportamento do mecanismo *BEAVER* é apresentado no Algoritmo 5.

Assim como o mecanismo RAAND, o *BEAVER* trabalha com o conceito de limite de ação ( $L_{MAX}$  para máximo e  $L_{MIN}$  para mínimo). Da mesma forma, também aplica a mesma notação, onde  $C$  é a demanda de tráfego atual,  $S_d$  é a quantidade de recursos alocados;  $T$  é o percentual dos recursos alocados no momento ( $S_d$ ) que podem ser tolerados acima ( $T_H$ ) ou abaixo ( $T_L$ );  $Time_T$  é o tempo máximo que a rede virtual pode ficar fora do intervalo de  $T$  (maior ou menor), e  $time$  é a variável que controla o tempo em que a rede virtual se encontra fora do intervalo. Adicionalmente, a variável *Suitability* é a largura de banda mínima necessária para manter a qualidade dos fluxos de vídeo, enquanto que a variável *Waste* representa se há desperdício de recursos (valores positivos). *Videoflows* é o conjunto de fluxos de vídeos identificados na rede virtual em questão, onde *Bitrate* é a estimativa de *bitrate* de um fluxo individual.

No Algoritmo 5, inicialmente, calcula-se a diferença entre os recursos alocados ( $S_d$ ) e o volume de tráfego geral do momento ( $C$ ). Posteriormente, define-se a quantidade mínima de recursos a serem alocados para manter a qualidade dos vídeos boa (linha 5), assume-se que os fluxos presentes na rede virtual já foram identificados (de acordo com um classificador de tráfego presente no módulo *Traffic Classification*).

Entre as linhas 7 e 23, o algoritmo identifica se os recursos alocados estão maiores que o volume de tráfego ( $Waste \geq 0$ ), representando uma possível redução dos recursos alocados, ou se os recursos estão abaixo do volume de tráfego, gerando uma possível diminuição na qualidade vivenciada pelos usuários.

A parte final do Algoritmo 5 (entre as linhas 25 e 29) visa evitar uma requisição de ajuste quando o valor definido ( $Adj$ ) é próximo à quantidade de recursos alocada no momento ( $S_d$ ), visto que a requisição de um ajuste pode representar um acréscimo no custo financeiro da rede virtual para o cliente.

**Algoritmo 5** Mecanismo de Ajuste BEAVER

---

```

1:  $Adj = 0$ ; ▷ Ajuste a ser aplicado
2:  $time = 0$ ;
3:  $Waste = S_d - C$ ;
4: para todo Fluxo  $f \in Video_{flows}$  faça
5:    $Suitability+ = BARS(f.Bitrate) * f.Bitrate$ ;
6: fim para
7: se ( $Waste \geq 0$ ) então
8:   se ( ( $S_d > C + T_H$ ) ) and ( $time == 0$ ) ) então
9:      $Adj = T_H$ ;
10:  fim se
11:   $time = 0$ ;
12: senão
13:   $time ++$ ;
14:  se ( $Suitability > C - T_L$ ) então
15:    se ( $Time_T \leq time$ ) então
16:       $Adj = C + Cl(S_d)$ ;
17:    senão
18:       $Adj = Suitability + Cl(S_d)$ ;
19:    fim se
20:  senão
21:     $Adj = T_L$ ;
22:  fim se
23: fim se
24: Colocar  $Adj$  dentro do limite ( $L_{MIN} \leq Adj \leq L_{MAX}$ );
25: se ( $S_d * (1 - T/2) \leq Adj \leq S_d * (1 + T/2)$ ) então
26:   Mantêm-se o valor atual de  $S_d$ ;
27: senão
28:   Ajusta-se o valor de  $S_d$  para  $Adj$ ;
29: fim se

```

---

## 7.4 Experimentos

A fim de avaliar a arquitetura proposta, bem como os mecanismos de ajuste desenvolvidos (*RAAND* e *BEAVER*), foram realizados dois conjuntos de experimentos utilizando um ambiente de emulação e um *testbed* real, apresentados nas Seções 7.4.1 e 7.4.2, respectivamente.

### 7.4.1 Ambiente de Emulação

O ambiente de emulação utilizado nos experimentos foi o Mininet<sup>5</sup>, com o controlador Ryu<sup>6</sup>, enquanto o Flowvisor[72] foi utilizado como *network hypervisor*. Nos experimentos foi injetado um conjunto de fluxos de dados e transmissões de vídeo. O tráfego de dados consistiu de quatro fluxos TCP a partir da ferramenta Iperf<sup>7</sup> com duração de 50 segundos.

<sup>5</sup>mininet.org/

<sup>6</sup>github.com/osrg/ryu/

<sup>7</sup>sourceforge.net/projects/iperf/

Enquanto que o tráfego multimídia foi composto dos vídeos *DanceKiss*, *Puskas-2013*, *Libertadores-2012*, e *FlagShoot*, onde as informações dos vídeos foram apresentadas na Tabela 7.1. A transmissão dos vídeos foi iniciada com um intervalo de 10 segundos na ordem citada.

Nos experimentos, os mecanismos desenvolvidos *RAAND* e *BEAVER* são avaliados, e seus desempenhos são comparados com a alocação de recursos fixa de 5Mbps, 10Mbps e 15Mbps. Os mecanismos de ajuste foram configurados com os seguintes parâmetros:  $L_{Max}=10\text{Mbps}$ ,  $L_{Min}=5\text{Mbps}$ ,  $T=5\%$ , e  $Time_T = 3$  segundos. Os resultados dos experimentos são mostrados com um intervalo de confiança de 95%.

Avaliou-se os seguintes aspectos: (i) o custo financeiro estimado (em inglês *financial cost*); (ii) o desperdício de recursos de rede; e (iii) a qualidade experienciada pelos usuários. Considerou-se um cenário onde o custo financeiro de um ajuste nos recursos de rede depende da configuração do provedor e que cada ajuste representa um custo.

Portanto, a Equação 7.2 modela o custo financeiro de acordo com a seguinte notação:  $t$  é o tempo de duração do experimento (em segundos);  $P$  é o custo para realizar um ajuste na rede virtual, sendo que nos experimentos foi configurado o valor de 10 unidades monetárias;  $n$  é o número de ajustes realizados durante o experimento;  $Bw_i$  é o custo de alocação no instante  $i$ ; e  $P_i$  é o custo para cada unidade de recurso alocado, nos experimentos para cada unidade de Bw é cobrado  $P_i = 1$  unidades monetárias (ou seja, 20Mbps representam 20 unidades monetárias, e assim por diante).

$$Cost = \left( \sum_{i=1}^t Bw_i * P_i \right) + P * n \quad (7.2)$$

A Figura 7.2 apresenta as informações relacionadas ao custo financeiro gerado. Deve-se ressaltar que o custo financeiro depende do valor cobrado pelo ISP para realizar um ajuste. Portanto, considerou-se que cada ajuste requisitado representa um custo de 10 unidades monetárias.

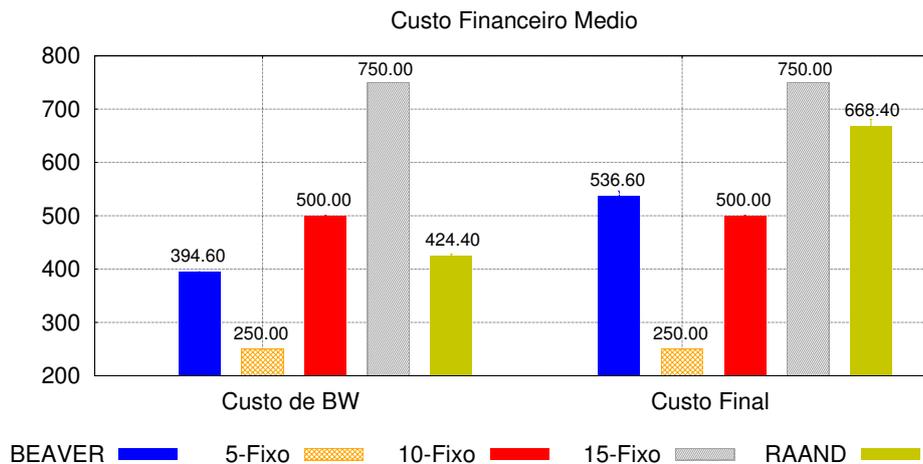


Figura 7.2: Aspectos de Custo Financeiro

As abordagens fixas possuem um custo financeiro estático, visto que elas não realizam ajustes. Contudo, essas abordagens comprometem a experiência do usuário devido ao fato que os fluxos TCP tendem a utilizar toda a largura de banda disponível, gerando

perdas e atraso nos fluxos de vídeo. Por outro lado, aplicando-se os mecanismos de ajuste os recursos alocados variam, e conseqüentemente o custo financeiro também.

O desperdício de recursos é o montante de recursos desnecessariamente alocados, sendo este modelado pela Equação 7.3, onde  $V_i$  é o volume de tráfego no instante  $i$ . Os resultados referentes ao desperdício de recursos são apresentados na Figura 7.3.

$$Waste = \sum_{i=1}^t f(i); \text{ onde } f(i) = \begin{cases} Bw_i - V_i, & \text{se } Bw_i \geq V_i \\ 0, & \text{caso contrário} \end{cases} \quad (7.3)$$

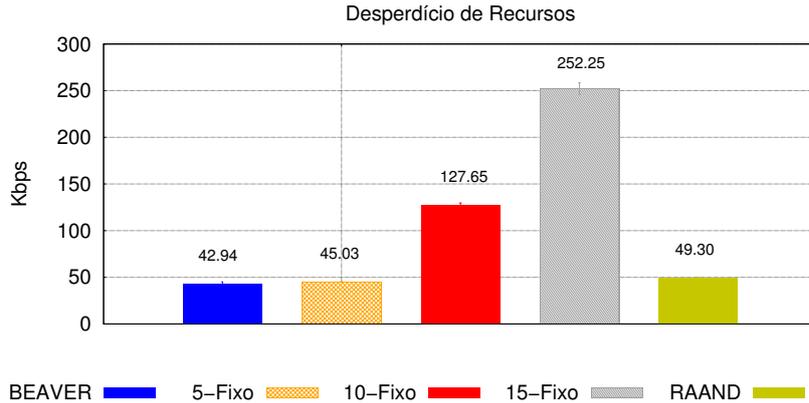


Figura 7.3: Desperdício de recursos

A partir das informações presentes na Figura 7.3, percebe-se que ambos os mecanismos de ajuste geram um pequeno desperdício, visto que analisam a demanda de tráfego corrente. Por outro lado, os casos *10-Fixo* e *15-Fixo* resultam em um desperdício em torno de três e cinco vezes maior, respectivamente. É válido mencionar que em um cenário com tráfego puramente multimídia, esta diferença entre os desperdícios de recursos resultante tendem a ser maior, pois os fluxos TCP tendem a utilizar a largura de banda disponível de acordo com o mecanismo de controle de congestionamento.

Comparando os mecanismos de ajuste, a principal diferença ocorre no número de ajustes efetuados. O mecanismo *RAAND* e *BEAVER* realizaram em média 24,40 (com desvio padrão de 1,06) e 14,20 (com desvio padrão de 0,99) requisições de ajuste, respectivamente. O mecanismo *RAAND*, por não considerar a adequabilidade dos recursos alocados em relação a demanda, realiza mais ajustes do que o mecanismo *BEAVER*.

Com relação a avaliação da percepção do usuário, foi utilizado o valor médio de SSIM e a pontuação MOS. Como descrito na Seção 7.3, SSIM correlaciona o vídeo original com o transmitido, obtendo valores entre 0 (pior caso) e 1 (melhor caso). MOS é uma pontuação dada pelo usuário que indica numericamente a satisfação do mesmo em relação à qualidade do vídeo, expressa em um valor entre 0 e 10. 0 é considerado o pior caso e 10 o melhor caso. Os vídeos transmitidos resultantes dos experimentos foram visualizados por 50 usuários.

As Figuras 7.4 e 7.5 mostram a qualidade dos vídeos transmitidos, ou seja, os valores médios de SSIM e MOS, respectivamente. Baseado nas informações mostradas na Figura 7.4, os mecanismos de ajuste *RAAND* e *BEAVER* mantêm a qualidade dos vídeos aceitável ( $SSIM \geq 0.7$ ). Enquanto que, com exceção do caso *15-Fixo*, as demais abordagens sofrem com a variação da demanda de tráfego, resultando em uma queda na qualidade dos

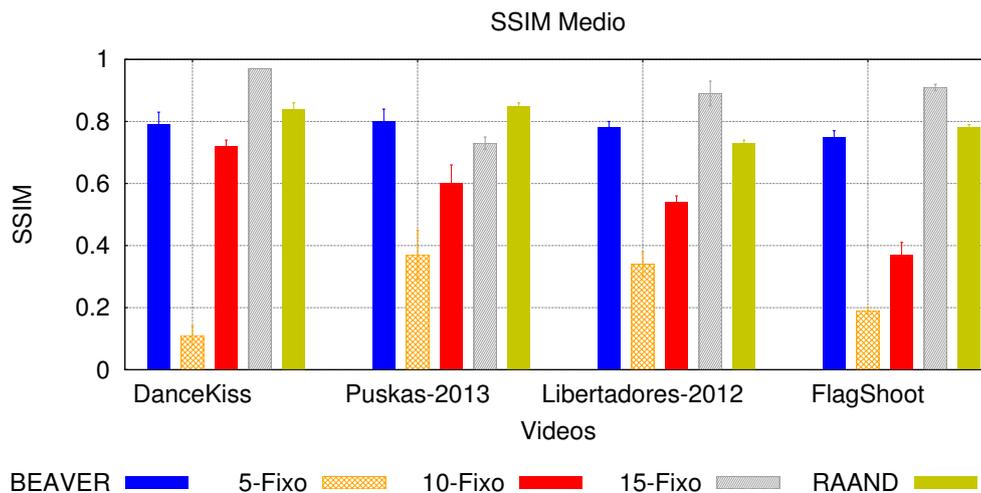


Figura 7.4: Valores de SSIM

vídeos transmitidos. Similarmente, os valores da pontuação MOS apresentados na Figura 7.5 refletem uma maior satisfação do usuário no caso *15-Fixo* e quando os mecanismos *RAAND* e *BEAVER* são aplicados.

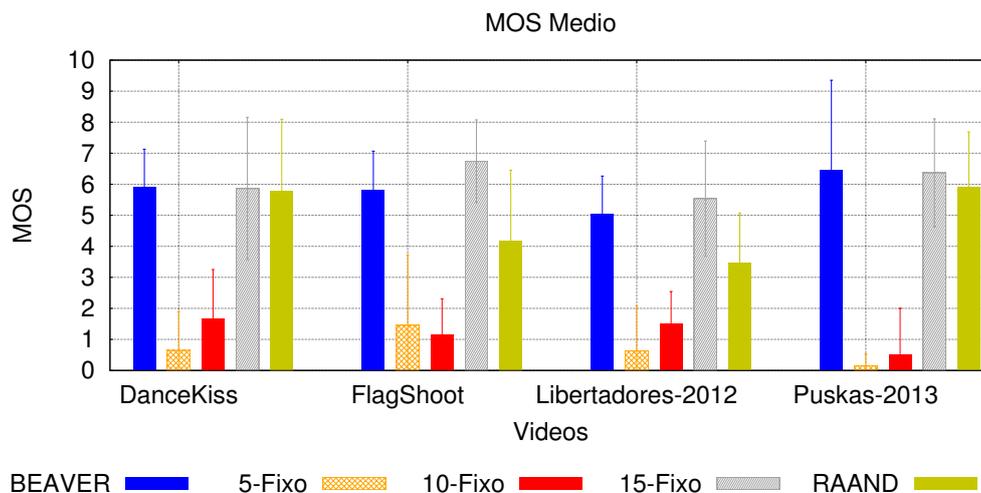


Figura 7.5: Resultado da Pontuação MOS

A fim de prover uma percepção visual da qualidade vivenciada pelos usuários, a Figura 7.6 apresenta alguns *frames* do vídeo Puskas-2013. São mostrados os *frames* relativos à metade de transmissão do vídeo, período de alta concorrência entre os fluxos, assim pode-se ter uma maior percepção da qualidade do vídeo experienciada pelos usuários. Percebe-se que os mecanismos de ajuste resultam em um *frame* mais próximo ao original quando comparado com as demais abordagens.

Baseado nos experimentos, conclui-se que a arquitetura proposta, bem como os mecanismos desenvolvidos conseguem obter um bom desempenho, evitando tanto um desperdício de recursos da rede, quanto uma degradação da percepção vivenciada pelo usuário em relação ao conteúdo do vídeo. No geral, para tráfegos multimídia o mecanismo *BEAVER* mostrou ser adequado para gerenciar os recursos de rede, obtendo um desempenho similar

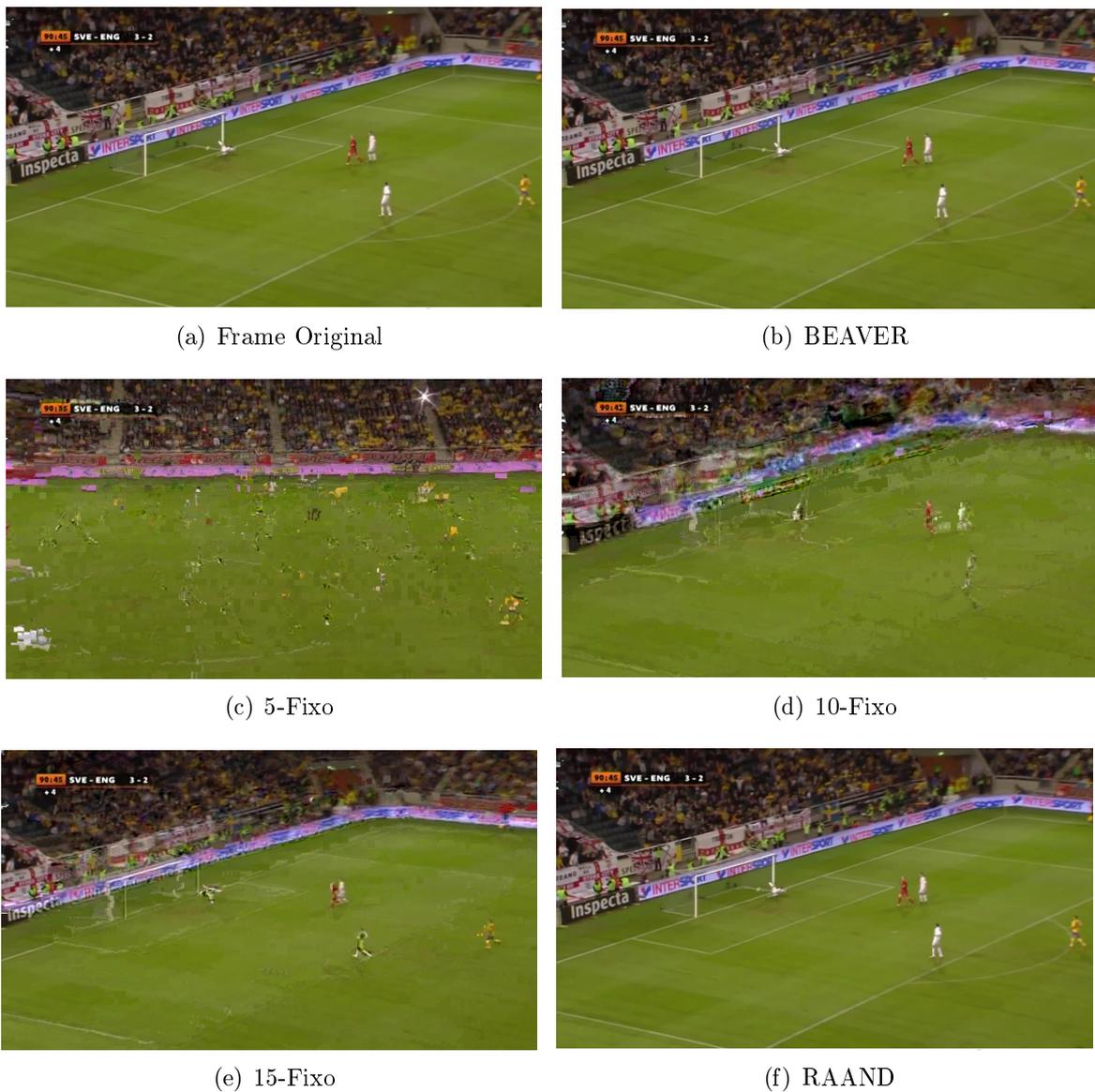


Figura 7.6: Frames do vídeo Puskas-2013.

ao mecanismo *RAAND*.

## 7.4.2 Testbed

A fim de avaliar a arquitetura desenvolvida, um protótipo da mesma foi desenvolvido para a realização de experimentos. Os experimentos realizados visaram avaliar a habilidade da arquitetura proposta em lidar com o processo de renegociação e ajuste da rede virtual. Os experimentos consistiram em injetar um conjunto de fluxos aleatórios do usuário para um servidor destino, onde a rede virtual é o meio de interconexão entre eles.

Os seguintes aspectos foram avaliados durante os experimentos: (i) Desperdício de recursos; e, (ii) Perda de pacotes. Desperdício de recursos é o montante de recursos alocados desnecessariamente. Perda de pacotes é o percentual de pacotes do fluxo que foram perdidos. A Figura 7.7 ilustra a configuração do ambiente para a realização dos experimentos.

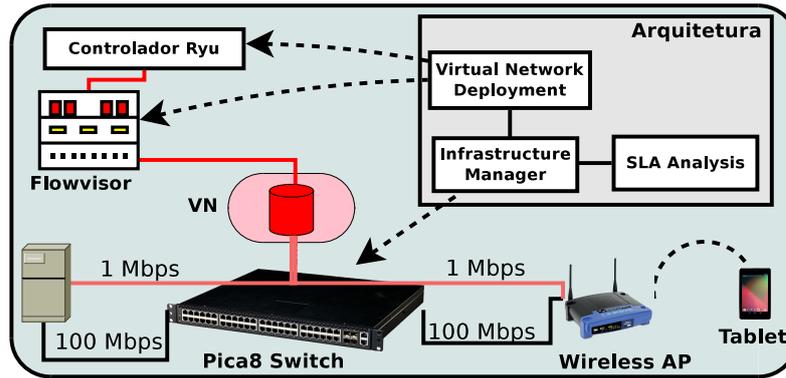


Figura 7.7: Configuração do Testbed

O ambiente foi composto de um switch Pica8<sup>8</sup> como infraestrutura, um ponto de acesso sem fio Linksys WRT54GL<sup>9</sup>, e um tablet Nexus<sup>10</sup> como usuário. Os experimentos foram realizados no *Network Research Laboratory (NRL)* na *University of California Los Angeles (UCLA)*. O mecanismo de ajuste RAAND foi aplicado, e devido a restrições de tempo não foi possível aplicar o mecanismo BEAVER nos experimentos.

Nos experimentos, foram gerados fluxos UDP a partir de uma distribuição exponencial, onde as médias utilizadas foram: 200Kbps de largura de banda, 500B de tamanho de pacote, 200ms de intervalo entre os fluxos, e 30 segundos de duração por fluxo. Cada experimento gerou fluxos durante 60 segundos. A ferramenta utilizada para geração de fluxos foi o Iperf.

O mecanismo de ajuste RAAND aloca a largura de banda com valores entre 1Mbps (Mínimo) e 10Mbps (Máximo) de acordo com a demanda de tráfego medida. Esta proposta foi comparada com a alocação de recursos estática: 1Mbps, 5Mbps e 10Mbps. Os experimentos foram realizados 50 vezes para cada caso e os resultados são apresentados com um intervalo de confiança de 95%.

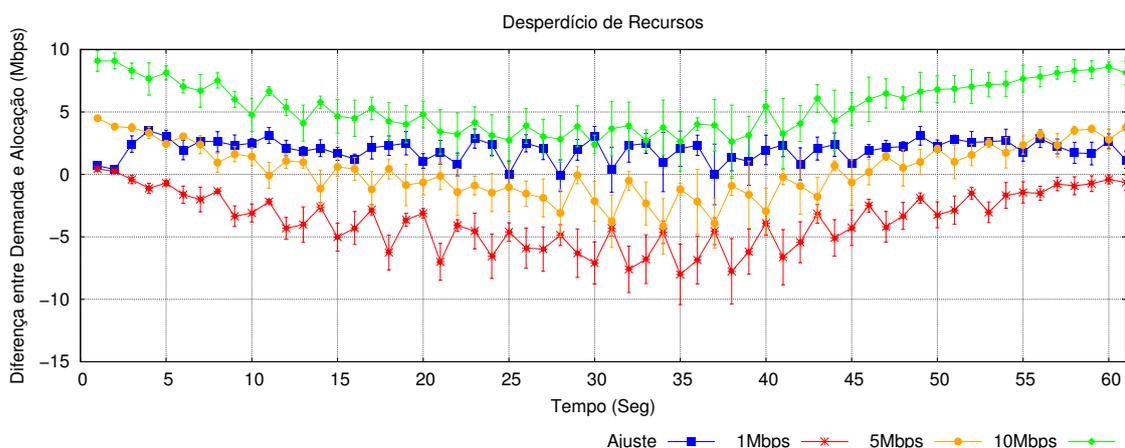


Figura 7.8: Diferença entre alocação e demanda

A Figura 7.8 apresenta o desperdício de recursos. Os valores positivos representam

<sup>8</sup><http://www.pica8.com/>

<sup>9</sup><http://www.linksys.com/en-eu/products/routers/WRT54GL>

<sup>10</sup><http://www.google.com.br/nexus/7/>

os casos de desperdício, enquanto que os valores negativos retratam uma situação de degradação da QoS, principalmente devido a perda de pacotes. A arquitetura desenvolvida consegue diminuir o desperdício e evitar uma queda na qualidade vivenciada pelo usuário.

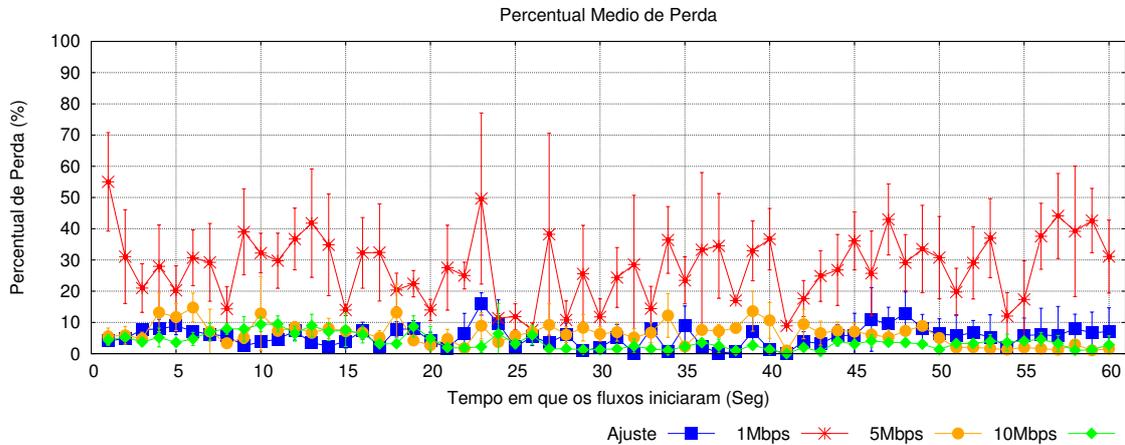


Figura 7.9: Perda Média

A Figura 7.9 mostra a perda média de todos os fluxos iniciados em cada segundo do experimento. Portanto, a Figura 7.9 ilustra a degradação da QoS causada pela diferença entre a demanda de tráfego e os recursos alocados. De acordo com as informações de perda, as perdas geradas pela arquitetura desenvolvida são 5 vezes menores do que no caso de 1Mbps estático, e equivalente aos casos de 5Mbps e 10Mbps.

Baseado nos experimentos, a arquitetura desenvolvida apresenta uma melhor performance, alcançando o menor desperdício de recursos e evitando uma degradação na QoS vivenciada pelos usuários. Sendo assim, aplicando a arquitetura desenvolvida, o cliente consegue manter o nível de QoS alto, enquanto o ISP consegue maximizar o número de clientes.

# Capítulo 8

## Conclusão

Esta tese apresentou mecanismos para permitir a negociação e o gerenciamento de redes virtuais definidas por software. Os mecanismos propostos englobam os passos para a implantação da rede virtual definida por software como um todo, habilitando a negociação do SLA para a rede virtual, a alocação da rede virtual na infraestrutura SDN e o gerenciamento dos aspectos pós-implantação.

Em geral, durante essa tese foram realizadas as seguintes contribuições: (i) O projeto de uma arquitetura para interligação dos mecanismos, habilitando a negociação e o gerenciamento de redes virtuais definidas por software; (ii) O desenvolvimento de um classificador de tráfego para identificar classes de QoS; (iii) A definição de um protocolo de negociação completa de SLA para redes virtuais, incluindo tanto parâmetros mensuráveis (recursos) quanto os não-mensuráveis (protocolos/aplicações) da rede; (iv) A criação de um mecanismo de apoio à negociação de redes virtuais, visando identificar qual a configuração mais se aproxima aos requisitos especificados; (v) O desenvolvimento de algoritmos para a alocação de redes virtuais; e, (vi) A definição de mecanismos de ajuste de redes virtuais.

O processo de negociação consiste na especificação do SLA e o gerenciamento da negociação entre o cliente e o provedor de Internet. Foi desenvolvido um protocolo de negociação completa, ou seja, que possui a habilidade de negociar os recursos de rede e os(as) protocolos/aplicações para a rede virtual negociada. Da mesma forma, foram propostos um modelo de similaridade para caracterizar os principais estilos de protocolos/aplicações de rede, bem como um mecanismo de suporte a negociação para identificar qual dos(as) protocolos/aplicações oferecidos(as) é o mais adequado ao conjunto de propriedades requisitadas pelo cliente.

A partir da definição do SLA, o provedor deve implantar a rede virtual para atender os requisitos do cliente, para isso os ISPs precisam executar um algoritmo de alocação de redes virtuais, o qual decide quais componentes (enlaces e nós) da infraestrutura de rede devem fazer parte da rede virtual em questão. O algoritmo desenvolvido engloba os aspectos importantes para a manutenção do SLA e para um melhor uso dos recursos de rede, como por exemplo: consumo de energia, largura de banda e confiabilidade.

Quando o processo de implantação da rede virtual é finalizado, os mecanismos permitem o monitoramento das redes virtuais diante das condições da infraestrutura de rede e do SLA vigente, identificando se é necessário adaptar as características da rede virtual.

Caso identifique-se a necessidade, visando atender os parâmetros especificados no SLA e/ou melhorar a utilização dos recursos de rede, uma alteração dos recursos alocados pode ser feita de acordo com um dos mecanismos de ajuste propostos.

Por final, os experimentos realizados mostraram as vantagens dos mecanismos propostos ao se definir os parâmetros mais adequados que cada rede virtual deve ter de acordo com os requisitos especificados pelo cliente. As vantagens proporcionadas pelos mecanismos englobam não somente os aspectos de implantação, mas também de adaptação após as redes virtuais estarem operacionais. Portanto, os mecanismos propostos atingem os objetivos esperados, melhorando a utilização dos recursos do provedor e garantindo uma melhor QoS aos usuários.

Como trabalhos futuros, pretende-se:

- Estudar maneiras de aprimorar a modelagem das características dos protocolos/aplicações de rede para o processo de negociação de SLA;
- Englobar aspectos de otimização no processo de alocação de redes virtuais;
- Estender a arquitetura para um gerenciamento englobando um ambiente sem fio como rede de acesso.

Com relação à produção científica oriunda dos mecanismos propostos na tese, foram publicados os seguintes trabalhos entre os anos de 2012 e 2015 [35, 36, 34, 33, 32, 31, 30, 29, 28, 27, 26].

# Referências Bibliográficas

- [1] I. Abdeljaouad and A. Karmouch. Utility function for predicting IPTV Quality of Experience based on delay in Overlay Networks. In *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*, pages 190–195, Jan 2013.
- [2] F. Abedin, Kuo-Ming Chao, and N. Godwin. A fuzzy group decision making process in a multi-agent negotiation environment. In *15th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 311–318, 2011.
- [3] H. Adeli and K.C. Sarma. *Cost Optimization of Structures: Fuzzy Logic, Genetic Algorithms, and Parallel Computing*. Wiley, 2006.
- [4] Elisangela Aguiar, André Riker, Eduardo Cerqueira, Antônio Abelém, Mu Mu, Torsten Braun, Marília Curado, and Sherali Zeadally. A real-time video quality estimator for emerging wireless multimedia systems. *Wireless Networks*, 20(7):1759–1776, 2014.
- [5] Ali Al-Shabibi, Marc De Leenheer, Matteo Gerola, Ayaka Koshibe, Guru Parulkar, Elio Salvadori, and Bill Snow. OpenVirteX: Make Your Virtual SDNs Programmable. In *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, HotSDN '14*, pages 25–30, New York, NY, USA, 2014. ACM.
- [6] Engin Arslan, Murat Yuksel, and Mehmet Hadi Gunes. Training network administrators in a game-like environment. *Journal of Network and Computer Applications*, 53(0):14 – 23, 2015.
- [7] S. Ballew. *Managing IP Networks with Cisco Routers*. O’Reilly Media, 1997.
- [8] A. Basta, W. Kellerer, M. Hoffmann, K. Hoffmann, and E.-D. Schmidt. A Virtual SDN-Enabled LTE EPC Architecture: A Case Study for S-/P-Gateways Functions. In *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for*, pages 1–7, Nov 2013.
- [9] D. Battre, F.M.T. Brazier, K.P. Clark, M. Oey, A. Papaspyrou, O. Wa andldrich, P. Wieder, and W. Ziegler. A proposal for ws-agreement negotiation. In *2010 11th IEEE/ACM International Conference on Grid Computing (GRID)*, pages 233 –241, oct. 2010.
- [10] Michael Till Beck and Marco Maier. Mobile Edge Computing: Challenges for Future Virtual Network Embedding Algorithms. In *The Eighth International Conference on*

- Advanced Engineering Computing and Applications in Sciences (ADVCOMP)*, pages 65–70, 2014.
- [11] R. Bolla, R. Bruschi, Franco Davoli, and F. Cucchietti. Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures. *IEEE Communications Surveys Tutorials*, 13(2):223–244, Second 2011.
- [12] I. Bueno, J.I. Aznar, E. Escalona, J. Ferrer, and J. Antoni Garcia-Espin. An Open-NaaS Based SDN Framework for Dynamic QoS Control. In *IEEE SDN for Future Networks and Services (SDN4FNS)*, pages 1–7, Nov 2013.
- [13] H.E.T. Carvalho, N.C. Fernandes, O.C.M.B. Duarte, and G. Pujolle. SLAPv: A service level agreement enforcer for virtual networks. In *International Conference on Computing, Networking and Communications (ICNC)*, pages 708–712, 2012.
- [14] S H. Cha, C. Tappert, and S. Yoon. Enhancing binary feature vector similarity measures. *Journal of Pattern Recognition Research*, pages 63–77, 2006.
- [15] C. Chaudet and Y. Haddad. Wireless Software Defined Networks: Challenges and opportunities. In *Microwaves, Communications, Antennas and Electronics Systems (COMCAS), 2013 IEEE International Conference on*, pages 1–5, Oct 2013.
- [16] Thomas M. Chen. Network Traffic Modeling. pages 326–339, 2007.
- [17] L. Chiaraviglio, M. Mellia, and F. Neri. Minimizing ISP Network Energy Cost: Formulation and Solutions. *Networking, IEEE/ACM Transactions on*, 20(2):463–476, 2012.
- [18] Woon Hau Chin, Zhong Fan, and Russell Haines. Emerging technologies and research challenges for 5G wireless networks. *Wireless Communications, IEEE*, 21(2):106–112, 2014.
- [19] S. S. Choi, S. H. Cha, and C. Tappert. A Survey of Binary Similarity and Distance Measures. *Journal on Systemics, Cybernetics and Informatics*, 8(1):43–48, 2010.
- [20] S. Davy, J. Famaey, J. Serrat-Fernandez, J.L. Gorricho, A. Miron, M. Dramitinos, P.M. Neves, S. Latre, and E. Goshen. Challenges to support edge-as-a-service. *Communications Magazine, IEEE*, 52(1):132–139, January 2014.
- [21] Michel Marie Deza and Elena Deza. *Encyclopedia of Distances*. Springer Berlin Heidelberg, 2009.
- [22] R. Doriguzzi Corin, M. Gerola, R. Riggio, F. De Pellegrini, and E. Salvadori. VeRTIGO: Network Virtualization and Beyond. In *Software Defined Networking (EWSDN), 2012 European Workshop on*, pages 24–29, Oct 2012.
- [23] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

- [24] Panagiotis Georgopoulos, Yehia Elkhatib, Matthew Broadbent, Mu Mu, and Nicholas Race. Towards Network-wide QoE Fairness Using Openflow-assisted Adaptive Video Streaming. In *Proceedings of the 2013 ACM SIGCOMM Workshop on Future Human-centric Multimedia Networking*, pages 15–20. ACM, 2013.
- [25] R. L. Gomes, L. Bittencourt, and E. Madeira. Supporting sla negotiation for vsdn based on similarity and price issues. In *Network Computing and Applications (NCA), 2014 IEEE 13th International Symposium on*, pages 287–290, Aug 2014.
- [26] R. L. Gomes, L. F. Bittencourt, and E. Madeira. A Framework for SLA Establishment of Virtual Networks based on QoS Classes. In *Proceedings of Fifth International Workshop on Management of the Future Internet (ManFI)*, 2013.
- [27] R. L. Gomes, L. F. Bittencourt, and E. R. M. Madeira. A Generic SLA Negotiation Protocol for Virtualized Environments. In *Proceedings of 18th IEEE International Conference On Networks (ICON 2012)*, 2012.
- [28] R. L. Gomes, L. F. Bittencourt, and E. R. M. Madeira. A Virtual Network Allocation Algorithm for Reliability Negotiation. In *22st International Conference on Computer Communications and Networks (ICCCN)*, 2013.
- [29] R. L. Gomes, L. F. Bittencourt, and E. R. M. Madeira. SLA Renegotiation According to Traffic Demand. In *2nd Workshop on Network Virtualization and Intelligence for the Future Internet (WNetVirt)*, 2013.
- [30] R. L. Gomes, L. F. Bittencourt, and E. R. M. Madeira. A Bandwidth-Feasibility Algorithm for Reliable Virtual Network Allocation. In *28th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, 2014.
- [31] R. L. Gomes, L. F. Bittencourt, and E. R. M. Madeira. A Similarity Model for Virtual Networks Negotiation. In *29th Symposium On Applied Computing (SAC)*, 2014.
- [32] R. L. Gomes, L. F. Bittencourt, E. R. M. Madeira, Eduardo C. Cerqueira, and M. Gerla. Energy-Aware Allocation of Reliable Virtual Software Defined Networks. In *12th IEEE Consumer Communications and Networking Conference (CCNC)*, 2015.
- [33] R. L. Gomes, L. F. Bittencourt, E. R. M. Madeira, Eduardo C. Cerqueira, and M. Gerla. QoE-Aware dynamic virtual network resource adaptation for EaaS environment. In *IEEE ICC 2015 - Communications Software, Services and Multimedia Applications Symposium (ICC'15 (10) CSSMA)*, pages 6861–6866, London, United Kingdom, June 2015.
- [34] R. L. Gomes, L. F. Bittencourt, E. R. M. Madeira, Eduardo C. Cerqueira, and M. Gerla. State-Aware allocation of reliable virtual software defined networks based on bandwidth and energy. In *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, USA, January 2016. (Accepted for publication).

- [35] R. L. Gomes, L. F. Bittencourt, E. R.M. Madeira, E. Cerqueira, and M. Gerla. An architecture for dynamic resource adjustment in VSDNs based on traffic demand. In *IEEE Global Communications Conference (GLOBECOM 2014)*, pages 2005–2010, Dec 2014.
- [36] R. L. Gomes and E. R. M. Madeira. A Traffic Classification Agent for Virtual Networks Based on QoS Classes. *Latin America Transactions, IEEE*, 10(3):1734–1741, april 2012.
- [37] Mithat Gonen. *Analyzing Receiver Operating Characteristic Curves With SAS (Sas Press Series)*. SAS Publishing, 2007.
- [38] F.E. Harrell. *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis*. Graduate Texts in Mathematics. Springer, 2001.
- [39] P.C.K. Hung, Haifei Li, and Jun-Jang Jeng. WS-Negotiation: an overview of research issues. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, page 10 pp., jan. 2004.
- [40] Nachikethas A. Jagadeesan and Bhaskar Krishnamachari. Software-Defined Networking Paradigms in Wireless Networks: A Survey. *ACM Comput. Surv.*, 47(2):27:1–27:11, November 2014.
- [41] Adam Kalai, Omer Tamuz, Ce Liu, Ohad Shamir, and Serge Belongie. Adaptively Learning a Similarity Model, 2012.
- [42] K. Kerpez and G. Ginis. Software-Defined Access Network (SDAN). In *48th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6, March 2014.
- [43] Hyojoon Kim and N. Feamster. Improving network management with software defined networking. *Communications Magazine, IEEE*, 51(2):114–119, February 2013.
- [44] Hyunmin Kim, Jaebeom Kim, and Young-Bae Ko. Developing a cost-effective Open-Flow testbed for small-scale Software Defined Networking. In *Advanced Communication Technology (ICACT), 2014 16th International Conference on*, pages 758–761, Feb 2014.
- [45] Dominik Klein, Thomas Zinner, Kathrin Borchert, Stanislav Lange, Vlad Singeorzan, and Matthias Schmid. Evaluation of Video Quality Monitoring Based on Pre-computed Frame Distortions. In *Advances in Communication Networking*, volume 8115 of *Lecture Notes in Computer Science*. 2013.
- [46] J.F. Kurose and K.W. Ross. *Computer Networking: A Top-down Approach*. Pearson Education, 2010.
- [47] Bob Lantz, Brandon Heller, and Nick McKeown. A network in a laptop: rapid prototyping for software-defined networks. In *Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks, Hotnets '10*, pages 19:1–19:6, New York, NY, USA, 2010. ACM.

- [48] Hyang-Won Lee, E. Modiano, and Kayi Lee. Diverse Routing in Networks With Probabilistic Failures. *IEEE/ACM Transactions on Networking*, 18(6), 2010.
- [49] M.;J. Lesot, M. Rifqi, and H. Benhadda. Similarity measures for binary and numerical data a survey. *Int. J. Knowl. Eng. Soft Data Paradigm.*, 1(1):63–84, December 2009.
- [50] Hongyun Li, Xirong Que, Yannan Hu, Xiangyang Gong, and Wendong Wang. An autonomic management architecture for SDN-based multi-service network. In *Workshops Proceedings of the Global Communications Conference, GLOBECOM 2013, Atlanta, GA, USA, December 9-13, 2013*, pages 830–835, 2013.
- [51] V. Li and J. Silvester. Performance Analysis of Networks with Unreliable Components. *IEEE Transactions on Communications*, 32(10):1105–1110, oct 1984.
- [52] T. Lin, Joon-Myung Kang, H. Bannazadeh, and A. Leon-Garcia. Enabling SDN applications on Software-Defined Infrastructure. In *Network Operations and Management Symposium (NOMS), 2014 IEEE*, pages 1–7, May 2014.
- [53] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan. Energy Aware Network Operations. In *IEEE INFOCOM Workshops 2009*, pages 1–6, 2009.
- [54] Isaias Martinez-Yelmo, Isaac Seoane, and Carmen Guerrero. Fair Quality of Experience (QoE) Measurements Related with Networking Technologies. In Evgeny Osipov, Andreas Kessler, ThomasMichael Bohnert, and Xavier Masip-Bruin, editors, *Wired/Wireless Internet Communications*, volume 6074 of *Lecture Notes in Computer Science*, pages 228–239. Springer Berlin Heidelberg, 2010.
- [55] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.
- [56] Deepankar Medhi and Karthikeyan Ramasamy. *Network Routing: Algorithms, Protocols, and Architectures*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007.
- [57] Marc Mendonca, Bruno Nunes Astuto, Xuan Nam Nguyen, Kattia Obraczka, Thierry Turetli, et al. A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. [http://hal.inria.fr/docs/00/82/50/87/PDF/SDN\\_survey.pdf](http://hal.inria.fr/docs/00/82/50/87/PDF/SDN_survey.pdf), 2013.
- [58] RE Munich. *Topics geo: natural catastrophes 2012: analyse, assessments, positions*. Munchener Ruckversichereungs-Gesellschaft, 2013.
- [59] Jad Naous, David Erickson, G. Adam Covington, Guido Appenzeller, and Nick McKeown. Implementing an OpenFlow Switch on the NetFPGA Platform. In *Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, ANCS '08, pages 1–9, New York, NY, USA, 2008. ACM.

- [60] Richard E. Neapolitan. *Learning Bayesian Networks*. Prentice-Hall, Inc., 2003.
- [61] Kien Nguyen, Quang Tran Minh, and S. Yamada. A Software-Defined Networking Approach for Disaster-Resilient WANs. In *22nd International Conference on Computer Communications and Networks (ICCCN)*, pages 1–5, 2013.
- [62] Elisabetta Nitto, Massimiliano Penta, Alessio Gambi, Gianluca Ripa, and Maria Luisa Villani. Negotiation of service level agreements: An architecture and a search-based approach. In *Proceedings of the 5th international conference on Service-Oriented Computing, ICSOC '07*, pages 295–306, Berlin, Heidelberg, 2007. Springer-Verlag.
- [63] G.P. O'Reilly, D.J. Houck, E. Kim, T.B. Morawski, D.D. Picklesimer, and H. Uzunalioglu. Infrastructure simulations of disaster scenarios. In *Telecommunications Network Strategy and Planning Symposium. NETWORKS 2004, 11th International*, pages 205–210, June 2004.
- [64] Panagiotis Papadimitriou, Olaf Maennel, Adam Greenhalgh, Anja Feldmann, and Laurent Mathy. Implementing Network Virtualization for a Future Internet. 20th ITC Specialist Seminar on Network Virtualization - Concept and Performance Aspects, 2009.
- [65] Jincheol Park, K. Seshadrinathan, Sanghoon Lee, and A.C. Bovik. Video Quality Pooling Adaptive to Perceptual Distortion Severity. *IEEE Transactions on Image Processing*, 22(2):610–620, Feb 2013.
- [66] H. Pouyllau and R. Douville. End-to-end QoS Negotiation in Network Federations. *3rd IEEE/IFIP International Workshop on Bandwidth on Demand and Federation Economics*, 2010.
- [67] A. Pras and G. Pavlou. Network and service management [Series Editorial]. *IEEE Communications Magazine*, 52(1):130–131, January 2014.
- [68] K. Pulakka. Controlling of satisfaction of the end-users and profits of the isps in the ds enabled internet. In *Communication Systems, 2002. ICCS 2002. The 8th International Conference on*, volume 1, pages 138–144 vol.1, Nov 2002.
- [69] S.R. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man and Cybernetics*, 21(3):660–674, may. 1991.
- [70] R. Serral-Gracià, E. Cerqueira, M. Curado, M. Yannuzzi, E. Monteiro, and X. Masip-Bruin. An Overview of Quality of Experience Measurement Challenges for Video Applications in IP Networks. In *Proceedings of the 8th International Conference on Wired/Wireless Internet Communications*, pages 252–263, Berlin, Heidelberg, 2010. Springer-Verlag.

- [71] M. Zubair Shafiq, Lusheng Ji, Alex X. Liu, and Jia Wang. Characterizing and Modeling Internet Traffic Dynamics of Cellular Devices. In *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '11, pages 305–316. ACM, 2011.
- [72] Rob Sherwood, Michael Chan, Adam Covington, Glen Gibb, Mario Flajslik, Nikhil Handigol, Te-Yuan Huang, Peyman Kazemian, Masayoshi Kobayashi, Jad Naous, Srinivasan Seetharaman, David Underhill, Tatsuya Yabe, Kok-Kiong Yap, Yiannis Yiakoumis, Hongyi Zeng, Guido Appenzeller, Ramesh Johari, Nick McKeown, and Guru Parulkar. Carving research slices out of your production networks with OpenFlow. *SIGCOMM Comput. Commun. Rev.*, 40:129–130, January 2010.
- [73] Pontus Skoldstrom and Kiran Yedavalli. Network virtualization and resource allocation in OpenFlow-based wide area networks. In *Proceedings of IEEE International Conference on Communications (ICC)*, 2012.
- [74] R. Soundararajan and A.C. Bovik. Video Quality Assessment by Reduced Reference Spatio-Temporal Entropic Differencing. *IEEE Transactions on Circuits and Systems for Video Technology*, 2013.
- [75] James P.G. Sterbenz, David Hutchison, Egemen K. Çetinkaya, Abdul Jabbar, Justin P. Rohrer, Marcus Schöller, and Paul Smith. Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Computer Networks*, 54(8):1245–1265, 2010.
- [76] Douglas D. Walker. Disaster Recovery Planning Inside General Electric. *Journal of Information Systems Management*, 2(4):25–33, 1985.
- [77] Shie-Yuan Wang. Comparison of SDN OpenFlow network simulator and emulators: EstiNet vs. Mininet. In *Computers and Communication (ISCC), 2014 IEEE Symposium on*, pages 1–6, June 2014.
- [78] Shie-Yuan Wang, Chih-Liang Chou, and Chun-Ming Yang. EstiNet openflow network simulator and emulator. *Communications Magazine, IEEE*, 51(9):110–117, September 2013.
- [79] S.H. Yeganeh, A. Tootoonchian, and Y. Ganjali. On scalability of software-defined networking. *Communications Magazine, IEEE*, 51(2):136–141, February 2013.
- [80] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3), 1965.
- [81] F. Zaheer, J. Xiao, and R. Boutaba. Multi-Provider Service Negotiation and Contracting in Network Virtualization. In *12th IEEE/IFIP Network Operations and Management Symposium*, pages 471–478, april 2010.
- [82] Geng Zhang, Lan Su, Yang Wang, Xi Liu, and Jie Li. Research on communication network architecture of energy internet based on sdn. In *Advanced Research and Technology in Industry Applications (WARTIA), 2014 IEEE Workshop on*, pages 316–319, Sept 2014.

- [83] Min Zhang, Chunming Wu, Yang Qiang, and Ming Jiang. Robust dynamic bandwidth allocation method for virtual networks. In *Proceedings of IEEE International Conference on Communications (ICC)*, 2012.
- [84] Shengli Zhang, Caihong Kai, and Lingyang Song. SDN based uniform network architecture for future wireless networks. In *Computing, Communication and Networking Technologies (ICCCNT), 2014 International Conference on*, pages 1–5. IEEE, 2014.
- [85] Wei Zhou, Li Li, Min Luo, and Wu Chou. Rest api design patterns for sdn northbound api. In *Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on*, pages 358–365, May 2014.