

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

LEANDRO ALEXANDRE FREITAS

**QoS-RRC: Um Mecanismo com
Orquestração de
Sobre-provisionamento de Recursos e
Balanceamento de Carga para
Roteamento Orientado a QoS na
Internet do Futuro**

Goiânia
2011

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR AS TESES E DISSERTAÇÕES ELETRÔNICAS (TEDE) NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

1. Identificação do material bibliográfico: ☒ **Dissertação** ☐ **Tese**

2. Identificação da Tese ou Dissertação

Autor (a):	Leandro Alexandre Freitas				
E-mail:	leandro.ccomputacao@gmail.com				
Seu e-mail pode ser disponibilizado na página? ☒ Sim ☐ Não					
Vínculo empregatício do autor			Servidor público federal – Docente IFG		
Agência de fomento:			UF:		Sigla:
País:	Brasil	UF:	GO	CNPJ:	937500991-20
Título:	QoS-RRC: Um Mecanismo com Orquestração de Sobre-provisionamento de Recursos e Balanceamento de Carga para Roteamento Orientado a QoS na Internet do Futuro				
Palavras-chave:	Sobre-provisionamento de recursos, Balanceamento de carga, Roteamento, Internet do Futuro				
Título em outra língua:	A Mechanism with Orchestration of the Overprovisioning of Resources and Load Balancing for QoS-Oriented Routing in Future Internet				
Palavras-chave em outra língua:	Overprovisioning of resources, Load balancing, Routing, Future Internet				
Área de concentração:	Ciência da Computação				
Data defesa: (dd/mm/aaaa)	18/02/2011				
Programa de Pós-Graduação:	Instituto de Informática				
Orientador (a):	Prof. Dr. Augusto José Venâncio Neto				
E-mail:	augusto.deti@ufc.br				
Co-orientador (a):*	Prof. Dr. Rui Aguiar				
E-mail:	ruilaa@ua.pt				

*Necessita do CPF quando não constar no SisPG

3. Informações de acesso ao documento:

Concorda com a liberação total do documento ☒ SIM ☐ NÃO¹

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF ou DOC da tese ou dissertação.

O sistema da Biblioteca Digital de Teses e Dissertações garante aos autores, que os arquivos contendo eletronicamente as teses e ou dissertações, antes de sua disponibilização, receberão procedimentos de segurança, criptografia (para não permitir cópia e extração de conteúdo, permitindo apenas impressão fraca) usando o padrão do Acrobat.

Assinatura do (a) autor (a)

Data: ____ / ____ / ____

¹ Neste caso o documento será embargado por até um ano a partir da data de defesa. A extensão deste prazo suscita justificativa junto à coordenação do curso. Os dados do documento não serão disponibilizados durante o período de embargo.

LEANDRO ALEXANDRE FREITAS

QoS-RRC: Um Mecanismo com Orquestração de Sobre-provisionamento de Recursos e Balanceamento de Carga para Roteamento Orientado a QoS na Internet do Futuro

Dissertação apresentada ao Programa de Pós-Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Mestre em Mestrado em Ciência da Computação.

Área de concentração: Redes de Computadores.

Orientador: Prof. Dr. Augusto José Venâncio Neto

Co-Orientador: Prof. Dr. Rui Aguiar

Goiânia
2011

Dados Internacionais de Catalogação na Publicação (CIP)

Alexandre, Leandro Alexandre

QoS-RRC: Um Mecanismo com Orquestração de Sobre-provisionamento de Recursos e Balanceamento de Carga para Roteamento Orientado a QoS na Internet do Futuro [manuscrito] / Leandro Alexandre Freitas. - 2011.

79 f. : il., figs., tabs.

Orientador: Prof^a. Dr. Augusto José Venâncio Neto: Co-Orientador: Prof^a. Dr. Rui Aguiar.

Dissertação (Mestrado) – Universidade Federal de Goiás, Instituto de Informática, 2011.

Bibliografia.

Inclui lista de tabelas e figuras.

1. Sobre-provisionamento de recursos. 2. Balanceamento de Carga. 3. Roteamento. 4. Internet do Futuro. 5. 4WARD. 6. Caminho Genérico. 7. Qualidade de Serviço. 8. Qualidade de Experiência. I Título

LEANDRO ALEXANDRE FREITAS

QoS-RRC: Um Mecanismo com Orquestração de Sobre-provisionamento de Recursos e Balanceamento de Carga para Roteamento Orientado a QoS na Internet do Futuro

Dissertação defendida no Programa de Pós-Graduação do Instituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Mestre em Mestrado em Ciência da Computação, aprovada em 18 de Fevereiro de 2011, pela Banca Examinadora constituída pelos professores:

Prof. Dr. Augusto José Venâncio Neto

Instituto de Informática – UFG

Presidente da Banca

Prof. Dr. Rui Aguiar

Instituto de Telecomunicações - Pólo Universidade de Aveiro – UA

Prof. Dr. Eduardo Coelho Cerqueira

EngComp – UFPA

Prof. Dr. Kleber Vieira Cardoso

Instituto de Informática – UFG

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador(a).

Leandro Alexandre Freitas

Mestre em Ciência da Computação pela Universidade Federal de Goiás (2010) e Bacharel em Ciência da Computação pela Universidade Federal de Goiás (2008). Participou durante sua graduação em Projetos de Pesquisa na área de Redes de Sensores sem fio e RFID (Radio-Frequency IDentification). Atualmente é professor em regime de dedicação exclusiva pelo Instituto Federal de Goiás e participa do projeto da Dell denominado Ubiquitous Computing for Learning Environment (UCLE), como Pesquisador Assistente. Suas áreas de interesse atuais são Redes de Sensores sem Fio, Internet do Futuro, Computação Ubíqua e sensível ao contexto para ambientes de aprendizagem e Qualidade de Serviço (QoS).

A Deus, meu Senhor e tudo, a Jesus, meu Salvador, a Maria, minha eterna mãe, minha família, presente do céu (meu pai Mário, minha mãe Maria das Graças e meu irmão Luciano), ao meu professor Augusto Neto, orientador e amigo. Por fim aos meus grandes amigos do Ministério Alfa Ômega.

Agradecimentos

A Deus meu criador e Maria sua Mãe, que me deram força e saúde para conclusão deste trabalho. A meus pais que sempre incentivaram meus estudos e me apoiaram em todos momentos. A meu irmão Luciano Alexandre, meu grande amigo, por todas palavras de conselho. A meu professor orientador Augusto Neto, pela orientação, pela amizade, paciência e por todos ensinamentos. Muito obrigado por tudo professor, mais que um orientador eu ganhei um grande amigo. A meus amigos de faculdade, em especial, André Coimbra, Thiago Rosa, Carine Rodrigues, Daniel Ferreira e Marcelo Quinta. A todos amigos do Ministério Alfa Ômega, cada um em especial. Ao Leandro Marçal, que colaborou efetivamente para conclusão deste trabalho e a todos aqueles que contribuíram de alguma forma para minha formação.

(...) Mas a vida, a vida, a vida,
a vida só é possível
reinventada. (...)

Cecília Meireles,
Poesia: Reinvenção.

Resumo

Alexandre Freitas, Leandro. **QoS-RRC: Um Mecanismo com Orquestração de Sobre-provisionamento de Recursos e Balanceamento de Carga para Roteamento Orientado a QoS na Internet do Futuro**. Goiânia, 2011. 79p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.

Os conceitos e modelos para Internet do Futuro no Projeto 4WARD abordam uma arquitetura *clean-slate* ("recomeçar a Internet do zero") com várias inovações na rede, incluindo um novo paradigma de conectividade, chamado Caminho Genérico (*Generic Path* - GP). Na arquitetura GP, várias facilidades foram projetadas para suportar eficientemente complexas aplicações de valor agregado e serviços com garantia de Qualidade de Serviço (*Quality of Service* - QoS).

Os GPs abstraem principalmente a heterogeneidade das redes e de qualquer entidade, independentemente de seu escopo (tecnologia, localização ou camada de arquitetural). Para isso, a cooperação da camada de rede com mecanismos de provisionamento é necessária, de modo a mapear as demandas dos recursos exigidos pela sessão (requisitos de QoS, como por exemplo largura de banda mínima e máximo atraso/perda) nos GPs adequados. Em contraste com o suporte atual, o provisionamento de QoS robusto e escalável é fortemente exigido para alocações eficientes de GPs.

Portanto, esta dissertação apresenta o QoS-Routing and Resource Control (QoS-RRC), um mecanismo de apoio a criação de GPs de modo a lidar com suas exigências. O QoS-RRC complementa arquitetura GP com roteamento orientado a QoS, auxiliado com balanceamento de carga, para selecionar os caminhos que vão ao encontro as demandas da sessão, enquanto mantém largura de banda residual para aumentar a experiência do usuário. Para obter escalabilidade, o QoS-RRC opera com base em uma abordagem centrada no provisionamento, que emprega baixo armazenamento de estado e poucas operações de rede.

A avaliação de desempenho do QoS-RRC foi realizada com o simulador *Network Simulator* v.2 (NS2), demonstrando drásticas melhorias da qualidade dos fluxos quanto a experiência de atraso e largura de banda, se comparado com as soluções do estado da arte. Além disso, o impacto do QoS-RRC em comparação com o atual QoS das redes IP e os mecanismos de roteamento padrão, sobre a experiência do usuário, também foi avaliado, analisando métricas objetivas e subjetivas de Qualidade da Experiência (*Quality of Experience* - QoE), ou seja, *Peak Signal to Noise Ratio* (PSNR), *Structural Similarity Index* (SSIM), *Video Quality Metric* (VQM) e *Mean Opinion Score* (MOS).

Palavras-chave

Sobre-provisionamento de recursos, Balanceamento de Carga, Roteamento, Internet do Futuro

Abstract

Alexandre Freitas, Leandro. **QoS-RRC: An Overprovisioning-centric and Load Balance-aided Solution for Future Internet QoS-oriented Routing.** Goiânia, 2011. 79p. MSc. Dissertation. Instituto de Informática, Universidade Federal de Goiás.

The Future Internet concepts and designs of 4WARD project concerns a clean-slate architecture with various networking innovations, including a new connectivity paradigm called Generic Path (GP). In GP architecture, several facilities are designed to efficiently support complex value-added applications and services with assured Quality of Service (QoS).

GPs mainly abstract underlying network heterogeneity, and any entity, regardless its scope (technology, location or architectural layer) communicate each other in a single way via a common interface. To that, cooperation with network-layer provisioning mechanisms is required in the sense to map data paths meeting session-demanded resources (QoS requirements - minimum bandwidth and maximum delay/loss experience) into appropriate GPs. In contrast as support today, robust and scalable QoS-provisioning facilities are strongly required for efficient GP allocations.

Therefore, this dissertation introduces the QoS-Routing and Resource Control (QoS-RRC), a set of GP-compliant facilities to cope with the hereinabove requirements. QoS-RRC complements GP architecture with QoS-oriented routing, aided with load balancing, to select paths meeting session-demands while keeping residual bandwidth to increase user experience. For scalability, QoS-RRC operates based on an overprovisioning-centric approach, which places low state storage and network operations.

Initial QoS-RRC performance evaluation was carried out in Network Simulator v.2 (NS2), demonstrating drastic improvements of flow delay experience and bandwidth use among a relevant state-of-the-art solution. Moreover, the impact of QoS-RRC compared to current IP QoS and routing standards on the user experience has been evaluated, by analysing main objective and subjective Quality of Experience (QoE) metrics, namely Peak Signal to Noise Ratio (PSNR), The Structural Similarity Index (SSIM), Video Quality Metric (VQM) and Mean Opinion Score (MOS).

Keywords

Sumário

Lista de Figuras	12
Lista de Tabelas	13
Lista de Algoritmos	14
Glossário	15
1 Introdução	17
1.1 Motivação	17
1.2 Objetivos Gerais	19
1.3 Objetivos Específicos	19
1.4 Estrutura da Dissertação	20
2 Trabalhos Relacionados	21
3 Visão Geral da Arquitetura 4WARD para Internet do Futuro	23
3.1 Arquitetura GP e seus elementos	23
3.2 Gerenciadores de Registro de Caminhos Genéricos (GPMR)	25
4 Descrição detalhada do QoS-RRC	28
4.1 Arquitetura QoS-RRC	29
4.1.1 Sentido <i>Up-stream</i> e <i>Down-stream</i> da rede	30
4.2 Interfaces Internas e Externas	31
4.3 QoS-RRC Protocol (QoS-RRC-P)	31
4.3.1 Response Identification Information (RII)	32
4.3.2 Reserved Path (RSVPATH)	33
4.3.3 Multicast Specification (MSPEC)	33
4.3.4 INFOSPEC	33
4.3.5 QoS Specification (QSPEC)	34
4.3.6 Mensagens QoS-RRC	36
4.3.7 Mensagem <i>RESERVE</i>	36
4.3.8 Mensagem <i>RESPONSE</i>	38
4.4 Agentes QoS-RRC	39
4.5 Tabelas de Estado (<i>State Table</i>)	40
4.5.1 Tabelas de estado QoS-RRC-E	41
4.5.2 Tabelas de estado QoS-RRC-C	42

5	Funcionalidades do QoS-RRC	44
5.1	Aprovisionamento de Recursos de QoS e Árvores Multicast	44
5.2	Roteamento no QoS-RRC	47
5.2.1	Composição das Tabelas de Roteamento	47
5.2.2	Métricas de Roteamento	47
5.2.3	Melhor Caminho: Fluxo Máximo Custo Mínimo	50
5.3	Detecção de Congestionamento ou Falha no Roteador para Apoio a Resiliência	52
5.3.1	Conceituação	53
5.3.2	Módulo DCRDrop	54
5.3.3	Detecção de Falha do Roteador - DDRouter	58
5.4	Balanceamento de Carga no QoS-RRC	60
5.4.1	Descrição do Algoritmo de Balanceamento de carga	61
6	Avaliação e Resultados	64
6.1	Simulador NS2	64
6.2	Evalvid - Uma Framework para Avaliação de Qualidade de Vídeo	65
6.3	Qualidade de Experiência (<i>Quality of Experience</i> - QoE)	66
6.4	Avaliação do QoS-RRC	66
6.4.1	Experimentos baseados na Rede	67
6.4.2	Experimentos baseados no Usuário	70
7	Conclusão e Trabalhos Futuros	74
7.1	Síntese e Contribuições da Dissertação	74
7.2	Trabalhos Futuros	75
	Referências Bibliográficas	76

Lista de Figuras

3.1	Arquitetura GP	25
3.2	Nós de Distribuição de Registro	26
3.3	Estrutura básica da GPMR e Classes de Recursos	27
4.1	Mecanismo QoS-RRC na Arquitetura GP	28
4.2	Arquitetura QoS-RRC	29
4.3	Especificação funcional do objeto RII	32
4.4	Especificação funcional do objeto RSVPATH	33
4.5	Especificação funcional do objeto MSPEC	33
4.6	Especificação funcional do objeto INFOSPEC	33
4.7	Especificação funcional do cabeçalho comum QSPEC	34
4.8	Especificação funcional do cabeçalho objeto QSPEC	34
4.9	Especificação funcional do QSPEC [21]	35
4.10	Especificação funcional do objeto <i>QoS Desired</i> [21]	35
4.11	Especificação funcional do objeto <i>QoS Available</i> [21]	36
4.12	Formato comum das mensagens utilizadas no <i>QoS-RRC-P</i>	36
4.13	Formato da mensagem <i>RESERVE</i>	38
4.14	Formato da mensagem <i>RESPONSE</i>	39
4.15	Agentes QoS-RRC	40
5.1	Diagrama de Eventos QoS-RRC	46
5.2	Fila RED	55
5.3	Cenário sem falha de roteador	59
5.4	Cenário com falha de roteador	60
6.1	Diagrama de Blocos da Simulação	65
6.2	Cenário de Simulação	67
6.3	Vazão: MARA (Sem balanceamento de carga) e QoS-RRC (Balanceamento de carga)	68
6.4	Atraso: MARA e QoS-RRC	69
6.5	PSNR	71
6.6	SSIM	71
6.7	VQM	72

Lista de Tabelas

4.1	<i>Códigos de erro especificados na suíte QoS-RRC</i>	34
4.2	<i>Descrição das flags especificadas no QoS-RRC</i>	37
4.3	<i>Tabela de Árvores QoS-RRC-E</i>	41
4.4	<i>Tabela de Sessões QoS-RRC-E</i>	41
4.5	<i>Tabela de Sessões QoS-RRC-E</i>	42
4.6	<i>Tabela de Roteamento QoS-RRC-C (Sentido Up-stream)</i>	42
4.7	<i>Tabela de Roteamento QoS-RRC-C (Sentido Down-stream)</i>	43
6.1	<i>Conversão de PSNR para MOS</i>	70
6.2	<i>Alguns frames do vídeo News com IP Standard e com QoS-RRC</i>	73

Lista de Algoritmos

5.1	Escolha do Caminho	51
5.2	Detecção de Congestionamento	58
5.3	Balanceamento de Carga	62

Glossário

AF	Assured Forward
BE	Best Effort
BGP	Border gateway Protocol
CR	Core Router
CT	Compartment
CTR	Compartment Record
DCRDrop	Detection Congestion or Router Drop
DCRQueues	Detection of Congestion based on RED Queues
EF	Ensured Forward
EGP	Exterior Gateway Protocol
EP	End-Point
ER	Egress Router
ET	Entities
GP	Generic Path
GPMR	GP Managements Records
IR	Ingress Router
MARA	Multiuser Aggregated Resource Allocation Mechanism
Maxth	Maximum Treshold
Minth	Minimum Treshold
MIRTO	Multi-path Iterative Routing Traffic Optimizer
MOS	Mean Opinion Score
MP	Mediation Point
MR	Master Record
MSPEC	Multicast Specification
NS-2	Network Simulator V2
OSPF	Quality Of Service extensions to Open Shortest Path First
OTCL	Object-oriented Tool Command Language

PSNR	Peak Signal to Noise Ratio
QoE	Quality of Experience
QoS-RRC-P	QoS-RRC Protocol
QoS-RRC-RC	QoS-RRC Resource Controller
QoS-RRC	QoS-Routing and Resource Control
QoS	Quality of Service
QSPEC	QoS Specification
RED	Random Early Detection
Routing-LB	Routing with Load Balanced
RSVPATH	Reserved Path
SSM	Structural Similarity Index
VoD	Video on Demand
VoIP	Voice on IP
VQM	Video Quality Metric

Introdução

Esta dissertação visa fornecer um mecanismo de roteamento orientado a QoS com orquestração de provisionamento de recursos e balanceamento de carga para Internet do Futuro. A solução proposta emprega também o controle de congestionamento das filas dos roteadores da rede, por meio de um gerenciador de filas ativas, o RED.

Este capítulo é estruturado como segue. A Seção 1.1 apresenta a motivação para emprego de um mecanismo com sobre-provisionamento de recursos e balanceamento de carga para roteamento orientado a QoS, o QoS-RRC. Na Seção 1.2 apresentamos os Objetivos Gerais e na Seção 1.3 os objetivos específicos. Por fim, a Seção 1.4 descreve a estrutura da dissertação.

1.1 Motivação

A tendência de permitir que os usuários acessem cada vez mais e mais novos serviços da Internet e aplicações ubíquas (a qualquer hora, em qualquer lugar e com qualquer dispositivo final) tem levantado questões-chave para cenários futuros, incluindo: i) conexões ininterruptas; ii) apoio a robustez; iii) propriedades de escalabilidade e; iv) confiabilidade nas comunicações. Nesse sentido, aplicações multimídia baseadas na Internet estão cada vez mais em foco, auxiliando em uma série de desafios da sociedade moderna (por exemplo, e-security, e-commerce, e-health e e-tourism). A lista das aplicações atualmente mais exigentes incluem também VoD, IPTV (Internet), VoIP, push-media, dentre outras.

Além disso, o interesse especial dos provedores de Internet móvel pode ser notado em serviços personalizados e imersivos (3D), destinados a fornecer acesso a vários usuários simultaneamente, independente da base de tecnologias de transporte suportada (também conhecido como multi-party). Neste âmbito, a integração de diferentes tipos de sessões do usuário tem um valor acrescentado por sua aplicabilidade social, tais como cuidados com saúde, serviços baseados em localização, monitoramento ambiental, atividade sísmica, ou gestão de energia. Tais sessões são atualmente sustentadas por

estruturas complexas, ao passo que poderiam ser muito mais eficientes e sustentáveis, se fossem integradas inteiramente à Internet como funções e processos básicos.

As características sensíveis ao tempo das sessões multimídia possuem diversos requisitos que não podem ser eficazmente suportados pela atual Internet. Entre outros aspectos, sessões requerem largura de banda suficiente garantida ao longo do tempo e robustez para propagações intermitentes de dados para além do paradigma tradicional de transporte *best-effort*. Assim, um mecanismo que ofereça Qualidade de Serviço (*Quality of Service* - QoS) é de suma importância de maneira que os recursos da rede possam ser configurados para que as sessões sofram baixas taxas de atraso de propagação (delay), variação do atraso (jitter) e perda de pacotes (loss) fim-a-fim. Além disso, o fornecimento de largura de banda com suporte multicast é necessário, o qual, vários usuários podem receber o conteúdo da mesma sessão simultaneamente, mantendo o desempenho da rede através da implantação de duplicação de pacotes somente em pontos estratégicos (não em toda a rede, como no unicast).

As tentativas de superar ou atenuar esta deficiência conduziram à proposta de algumas funções não suportadas nativamente pelo modelo convencional da Internet, de modo a fornecer QoS, multicast e suporte a mobilidade de acordo com as necessidades atuais. Apesar de não exigirem mudanças significativas nos elementos de rede, os mecanismos de sobreposição aumentam a complexidade do sistema e afetam drasticamente o desempenho da rede, principalmente em termos de escalabilidade. Neste último, adicione às implantações de sobreposição excessiva carga de sinalização, consumo de energia e sobrecarga de CPU/memória com funções de controle, a maioria delas de forma centralizada [19]. Além disso, acrescenta-se a heterogeneidade dos cenários atuais, motivada pelo *cross-layering* onde as informações podem ser acessadas por camadas não conectadas diretamente. Por exemplo, a qualidade da sessão pode ser adaptada com base no contexto de diferentes camadas: sessão (codec, destino e políticas); rede (largura de banda, atraso, jitter e perda de pacotes); MAC (erros de quadros, prioridade de enfileiramento, tamanho da rajada, etc) e; ambiental (velocidade, localização, umidade, etc.) No entanto, o *cross-layering* na Internet atual é muito complexo, uma vez que protocolos e aplicações, cada um com suas características e mecanismos próprios, devem ser usados.

Os problemas e limitações da atual Internet [27], motivaram a comunidade científica a avaliar o uso de uma nova arquitetura de rede, totalmente remodelada com grandes inovações para a implantação eficiente de cenários futuros. A Internet do Futuro foi concebida para permitir o apoio a diferentes tipos de sessões, que podem ser acessadas ubiquamente e intermitentemente propagadas com garantias de recursos ao longo do tempo através de conexões multi-party (um-para-um, um-para-muitos ou muitos-para-muitos).

O projeto FP7 4WARD [29][33] é uma proposta Européia que visa especificar

uma arquitetura nova para a Internet. A proposta do 4WARD é remodelar completamente a internet combinando múltiplas arquiteturas de rede complementares em um foco comum, orientado a objetos. Entre suas características (por exemplo, redes de informação, gerenciamento nativo e virtualização), um paradigma de conectividade novo chamado de Caminho Genéricos(GP) é proposto. O transporte de dados através dos GPs ocorre por meio do mapeamento das sessões em recursos físicos, tais como enlaces, interfaces de rede, roteadores, dentre outros. A arquitetura dos GPs visa otimizar a conectividade empregando um único caminho com suporte nativo a confiabilidade, robustez, QoS, QoE, multi-party e mobilidade, de modo a fornecer comunicação entre qualquer entidade, independentemente de sua localização ou camada arquitetural. Em outras palavras, as mesmas funções podem ser invocadas por quaisquer entidades para a comunicação local ou remota.

O sucesso da arquitetura GP depende do apoio de mecanismos externos que sejam eficientes e que forneçam capacidade de roteamento e alocação dos recursos físicos, para os caminhos de comunicação. Neste sentido, é possível mapear as sessões dos usuários de acordo com seus requisitos fornecendo transporte de dados *multi-party* fim-a-fim, com garantia de QoS.

1.2 Objetivos Gerais

Como objetivos gerais, propomos um mecanismo com orquestração de provisionamento de recursos e balanceamento de carga para roteamento orientado a QoS na Internet do Futuro, denominado QoS-RRC.

1.3 Objetivos Específicos

Como objetivos específicos, esta dissertação visa melhorar a arquitetura GP fornecendo suporte a partir dos serviços de provisionamento de recursos e balanceamento de carga para roteamento. O QoS-RRC permite a instalação de GPs dinamicamente para satisfazer as exigências da sessão, em termos de QoS e conectividade *multi-party*, sobre ambientes heterogêneos. O QoS-RRC usa uma abordagem mais centrada no provisionamento de recursos para enfrentar problemas de escalabilidade, por meio da tomada de decisões distribuídas, sem operações por-fluxo e com controle de recursos.

Os benefícios do QoS-RRC também são experimentados por usuários finais, onde as percepções visuais aumentam em comparação com as abordagens atuais. Avaliações de desempenho foram realizados com o QoS-RRC utilizando o simulador (*Network Simulator v.2 - NS2* [30]), que demonstrou os benefícios do mecanismo em termos de utilização de largura de banda (para a comprovação do comportamento da largura de banda

residual global) e os atrasos provocados pelos caminhos ao longo do tempo de simulação (para o impacto do controle de qualidade).

Além disso, o impacto do QoS-RRC quanto à experiência do usuário também foi medido através da análise de métricas objetivas e subjetivas de QoE, *Peak Signal to Noise Ratio* (PSNR), *Structural Similarity* (SSM), *Video Quality Metric* (VQM) e *Mean Opinion Score* (MOS), utilizando seqüências de um vídeo real. O PSNR é um termo da engenharia para a relação entre a potência máxima possível de um sinal e a potência gerada por um ruído. O PSNR é mais comumente usado como uma medida de qualidade de reconstrução de codecs de compressão (por exemplo, para a compressão de imagens). O SSM é um método para medir a similaridade entre duas imagens. O VQM é um método padronizado de medir objetivamente a qualidade de vídeo que melhor prediz a classificação da qualidade subjetiva, que poderia ser obtida a partir de espectadores humanos. Já a métrica MOS é uma média aritmética de todas as notas individuais atribuídas pelos telespectadores.

1.4 Estrutura da Dissertação

A dissertação está estruturada da seguinte forma. No Capítulo 2, apresentamos alguns trabalhos relacionados a nossa proposta. O Capítulo 3 apresenta uma visão geral da arquitetura 4WARD para Internet do Futuro. No Capítulo 4 fornecemos uma descrição detalhada do mecanismo, apresentando a arquitetura do QoS-RRC, suas interfaces, a especificação protocolar, seus agentes e as tabelas de estado. No Capítulo 5 apresentamos as funcionalidades do QoS-RRC, descrevendo todos seus módulos e, por fim, no Capítulo 6 descrevemos as avaliações e resultados coletados das experimentações com o QoS-RRC e outros mecanismos. O Capítulo 7 apresenta as conclusões e trabalhos futuros.

Trabalhos Relacionados

Os trabalhos relacionados apresentados nesta seção visam apontar alguns dos mecanismos que propõem serviços similares aos oferecidos pelo QoS-RRC. Dentre os mecanismos que identificamos temos: MARA, MIRTO e QOSPF.

O *Multiuser Aggregated Resource Allocation Mechanism* (MARA) [1] é um mecanismo de provisionamento de recursos para Internet do Futuro. Ele emprega sobreprovisionamento dinâmico da rede (largura de banda e árvores multicast) sem sinalização por-fluxo. A qualidade de serviço é assegurada com base nas sobre-reservas por-classe definidas a partir da arquitetura DiffServ [40], sendo reajustadas dinamicamente para mapear a utilização de largura de banda. As árvores multicast são provisionadas durante a inicialização da rede e dinamicamente instaladas, baseadas em agregação multicast. Quanto ao roteamento, o MARA utiliza o protocolo de roteamento padrão da internet, não garantindo QoS quanto a escolha das rotas, diferentemente do que ocorre com o QoS-RRC que escolhe as melhores rotas a partir de um algoritmo de fluxo máximo custo mínimo [13].

O *Multi-path Iterative Routing Traffic Optimizer* (MIRTO) [10] é um protocolo de roteamento distribuído, que explora a diversidade dos caminhos adaptativo. Um agente MIRTO é executado no roteador de entrada da rede e é responsável por dividir o tráfego de entrada em múltiplos caminhos em direção ao destino. O envio dos fluxos por mais de um caminho é atualizado a cada tempo, de acordo com um algoritmo especificado. O agente MIRTO pode ser associado a um fluxo ou a um tráfego agregado, de acordo com o contexto considerado na aplicação (por exemplo, multi-homing que é uma técnica para aumentar a confiabilidade da conexão com a Internet para redes IP). O MIRTO não realiza provisionamento de classes e árvores como o QoS-RRC.

O *Quality Of Service extensions to Open Shortest Path First* (QOSPF) [9] é uma extensão do OSPF, que fornece suporte aos fluxos com QoS. Quando suas operações não utilizam uma arquitetura de reserva de recursos, como por exemplo o RSVP [32], o provisionamento de recursos ocorre por meio de métodos estatísticos que medem a largura de banda e características de atraso do enlace. Caso não seja utilizado um mecanismo com reserva explícita de recursos, o QOSPF não poderá fornecer garantias de que as

restrições baseadas nas métricas de QoS sejam respeitadas. Apesar de fornecer suporte ele é baseado em um mecanismo de inundação, o que prejudica a rede. O QoS-RRC não realiza inundação frequentemente, mas apenas no *bootstrap* da rede, o que auxilia na redução de sobrecarga de processamento e de dados na rede. Além disso, o roteamento do QoS-RRC ocorre de maneira eficaz pois ele acontece de maneira a distribuir os fluxos de dados pelos caminhos.

Os trabalhos relacionados, não oferecem todos os serviços presentes no QoS-RRC. Sendo assim, fica evidente que, dentre todos os trabalhos relacionados, os mecanismos não garantem provisionamento de recursos e de árvores multicast, balanceamento de carga para roteamento orientado a QoS e mecanismo de detecção de congestionamento ou falha de roteador. Além disso, nenhuma destas propostas possuem interfaces que permitam o acoplamento a arquitetura GP, o que é um dos requisitos imprescindíveis para o mecanismo.

Visão Geral da Arquitetura 4WARD para Internet do Futuro

O objetivo geral da Arquitetura GP é generalizar e abstrair o modelo de comunicação de uma forma inovadora, sem utilizar o modelo arquitetural da Internet atual. Especificamente, os objetivos do framework GP são [33]: i) Desenvolver bases para descrever e prescrever qualquer comunicação de rede; ii) Prover um modelo de serviço de comunicação genérica que não seja limitada por nenhum paradigma de comunicação e; iii) Projetar primitivas e construtores arquiteturais a partir deste paradigma. Como a arquitetura GP é orientada a objetos, suas características incluem recursividade, opacidade controlada e virtualização, com estados e funcionalidades em qualquer nível e acessível a partir de qualquer nível e; iv) Ser agnóstica, relativo à tecnologia, plataforma e comunicação.

A combinação destas e outras características fazem do GP uma arquitetura muito poderosa para projeto e descrição de redes, ao mesmo tempo que seletivamente esconde a complexidade de gerenciamento das redes, se for desejado. As entidades e suas respectivas funções da arquitetura GP serão descritas nas seções seguintes.

O restante deste capítulo está estruturado da seguinte forma. Na Seção 3.1, descrevemos a Arquitetura dos Caminhos Genéricos juntamente com seus elementos. Na Seção 3.2, apresentamos os Gerenciadores de Registro de Caminhos Genéricos.

3.1 Arquitetura GP e seus elementos

Nessa arquitetura, nós (terminais ou de rede) são representados por *Nós Compartimento* (*Compartment* - CT). Dentro de um nó CT, existem as *Entidades* (*Entities* - ET), as quais modelam um serviço em execução em qualquer nível e generalizam uma função de processamento de dados para comunicação, podendo também se comunicarem com outras ETs, tanto horizontalmente quanto verticalmente. Entre suas funções incluem a resolução de nomes, roteamento e endereçamento dentro de seu escopo ou CT. Para estabelecer e manter uma comunicação entre si, eles configuram os GPs, uma abstração de um caminho percorrido pela informação, a qual generaliza o transporte de dados e/ou um

serviço de transformação através da rede. Um GP é acessado pelas ETs através de *End Points* (EPs), os quais compreendem o conjunto de recursos necessários para prover comunicação fim-a-fim entre duas ou mais ETs. Assim, as ETs podem ter um ou mais EPs. Tanto comunicações inter-serviços (comunicação inter-processos) quanto comunicações inter-tecnologias (vertical) são obtidas conectando os EPs através de GPs. As comunicações inter-processos, apesar de serem obtidas por meio dos GPs, recebem o nome de *Hooks*. Como uma diversidade de recursos podem ser representados por um GP é necessário que ele seja contextualizado, sendo este escopo delimitado pelo CT no qual o GP está sendo executado e pode, por exemplo, corresponder a uma rede, a uma aplicação ou a um procedimento (tanto local como remoto).

Um GP pode ser composto por outros GPs, cada um implementando um serviço de tecnologia diferente, permitindo desta forma uma visão ubíqua do cenário. Para se interconectar a GPs, um *Ponto de Mediação* (*Mediation Point* - MP) é utilizado, escoando os dados entre vários GPs. O MP associa EPs de GPs com outros EPs, compondo um caminho fim-a-fim entre duas entidades comunicantes. Cada GP é criado por uma GP Factory, e suas informações, relativas aos GPs, são armazenadas nos *Gerenciadores de Registro de Caminhos Genéricos* (*GP Managements Records* - GPMR). O GPMR é compreendido principalmente de uma lista de GPs e seus respectivos atributos.

Os GPs devem ser mapeados em recursos físicos (enlaces, interfaces, etc.) para permitir a propagação de dados, e um framework para coletar e controlar as informações relativas a rede é necessário. Sendo assim, existe a necessidade da utilização de um mecanismo que faça a coleta e controle destas informações, e que realize também, o roteamento baseado nestas informações, objetivo que será conseguido com a utilização de um mecanismo de roteamento com QoS.

A Figura 3.1 demonstra a arquitetura GP e seus elementos.

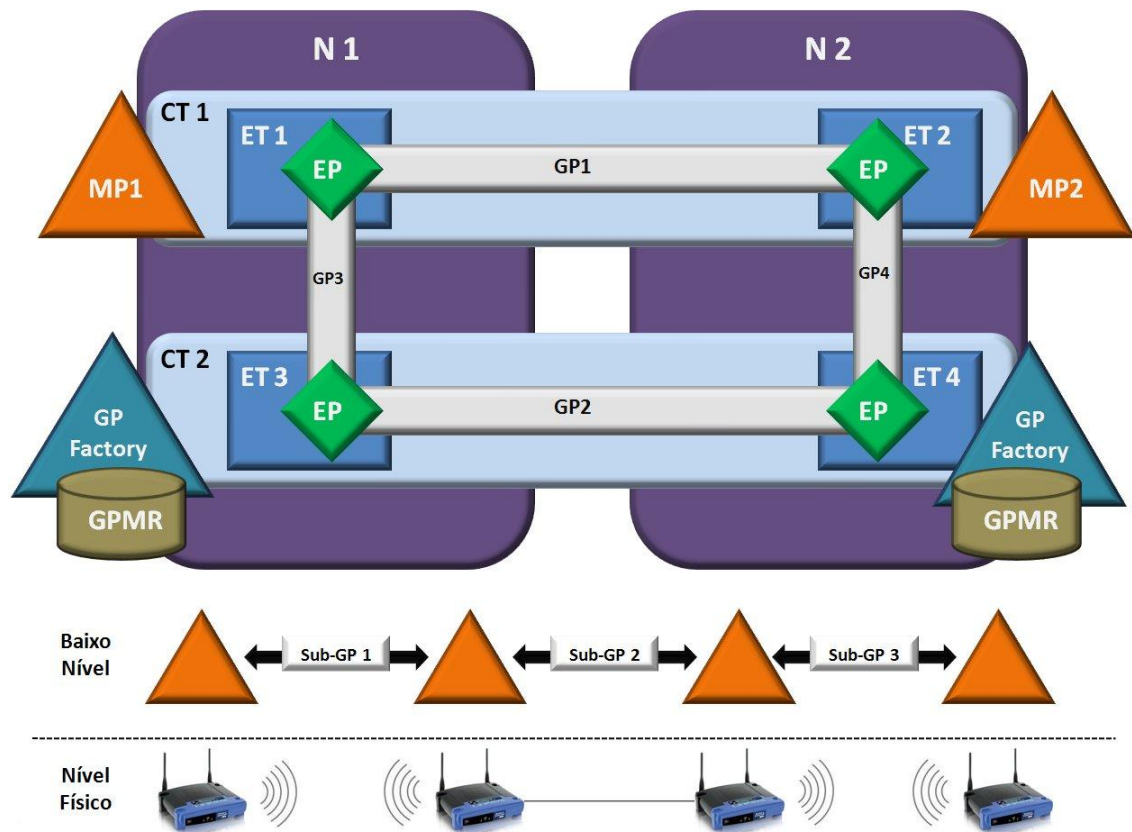


Figura 3.1: Arquitetura GP

3.2 Gerenciadores de Registro de Caminhos Genéricos (GPMR)

O Gerenciador de Registro de Caminhos Genéricos é um banco de dados repleto de GPs. Os recursos associados a um GP são organizados como informações de atributo em um objeto chamado GPIItem. O acesso às informações é possível através *Dials*, para Inspeção/Configuração das informações. As informações de controle presentes em um GPIItem incluem a lista de composição dos sub-GPs, parâmetros das aplicações multimídia como QoS/QoE e outras características como por exemplo, privacidade ou políticas relacionadas. Um registro que mantém um GP de alto-nível, mostrando um relacionamento entre os GPMR, é apresentado em [19][34], denominado Registro Mestre (*Master Record* - MR). Um MR tem uma visão geral de todos os GPs de baixo-nível em execução, associados a mesma comunicação. Ao adicionar o *Registro Compartimento* (*Compartment Record* - CTR) como um registro de anúncio do GPMR em execução no CT é possível estabelecer um sistema hierárquico distribuído. Desta forma, um MR aponta a uma lista de RCs, um para cada CT, no qual o nó compartimento participa.

Sendo assim, o GPMR traz uma abordagem inovadora para as redes atuais, pro-

porcionando uma estrutura unificada sobre os caminhos de comunicação. Nesse sentido, como os GPs precisam de um controle adequado, e os GPMRs armazenam todas as informações relevantes aos GPs, torna-se fácil fornecer qualquer dado necessário a criação de um novo GP.

A Figura 3.2 mostra a disposição lógica dos registros em nós.

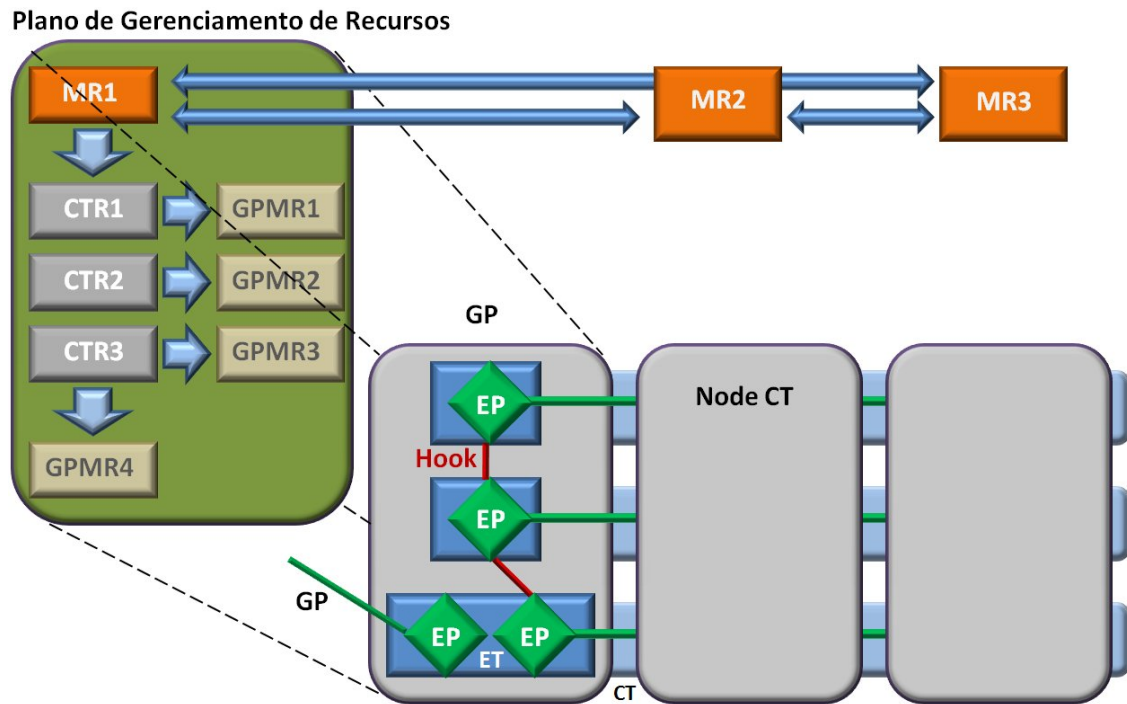


Figura 3.2: Nós de Distribuição de Registro

Em um nível de abstração mais elevado, um GP é resultado de uma agregação de GPs de nível inferior (sub-GPs), um para cada uma das tecnologias de comunicação (TCP, Ethernet, Fibra Óptica, etc). A gama de recursos geridos a partir de estatísticas, como vazão, SNR, PSNR, SSIM, VQM, MOS e atraso fim-a-fim, para as políticas de dados inclui, o nível de prioridade dos pacotes e a importância de cada frame de vídeo sobre a experiência do usuário. A natureza das informações armazenadas e a estrutura dos registros dependem da classe do GP, e portanto, uma classe GPMR existe para cada classe do GP. Todas as classes GPMR derivam de uma GPMR, que contém: uma lista de EPs, uma lista composta de GPs e um conjunto de GPItens. A classe GPMR portanto, depende diretamente da classe GPItem, que será sub-classificada de acordo com características como orientação à conexão (datagrama vs stream), propriedades físicas relacionadas (com fio vs sem fio) ou número de destinos (unicast vs multicast vs broadcast). Dessa forma, um GP terá um número muito específico de GPItens, que estão relacionados à tecnologia ou ao serviço, como mostrado na Figura 3.3.

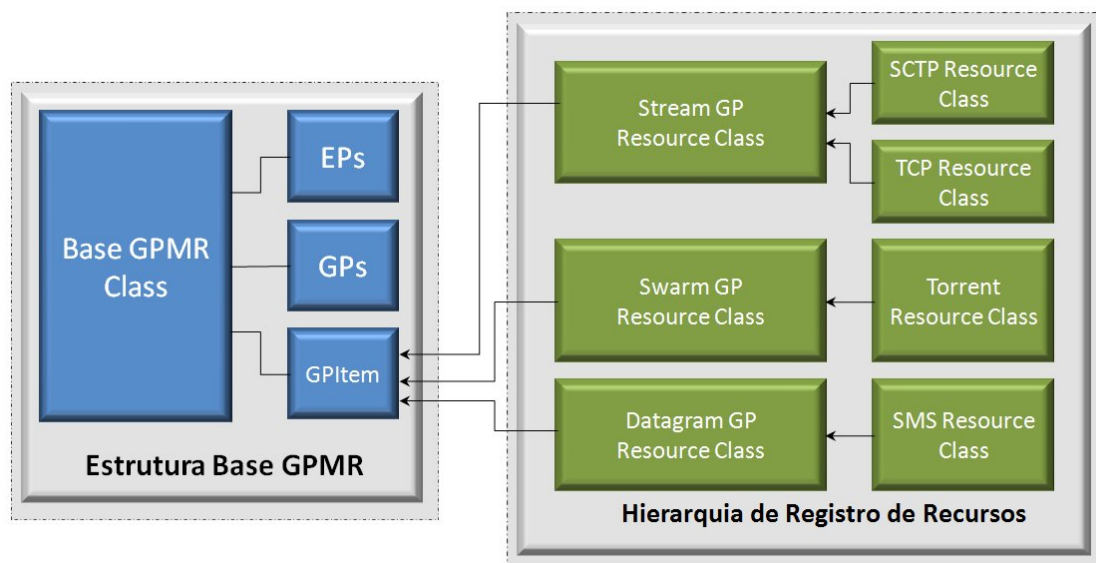


Figura 3.3: *Estrutura básica da GPMR e Classes de Recursos*

Descrição detalhada do QoS-RRC

A proposta do mecanismo QoS-RRC é fornecer suporte às fábricas de GPs, a fim de garantir com que os requisitos das entidades interessadas sejam cumpridos, evitando a degradação do sistema. Para isso, o QoS-RRC complementa a arquitetura GP fornecendo roteamento e QoS, operando dinamicamente sobre os recursos da rede e garantindo ao usuário final uma melhor experiência com o sistema utilizado.

Em termos de suporte a QoS, a abordagem de sobre-reserva dinâmica de recursos associada com controle de admissão é implantada para permitir a instalação da sessão sem sinalizações por fluxo. Além disso, o QoS-RRC mantém uma tabela de estado com informações sobre os melhores caminhos da rede para realizar o roteamento. Assim, as decisões de roteamento podem ser implementadas de forma mais eficiente por meio de uma visão geral da rede e das capacidades atuais de QoS, ao contrário de soluções [9] que mantêm apenas as informações do próximo salto com os recursos classificados por métricas ineficientes e mantido por operações constantes de inundações. A Figura 4.1 mostra a arquitetura GP com o mecanismo QoS-RRC acoplado:

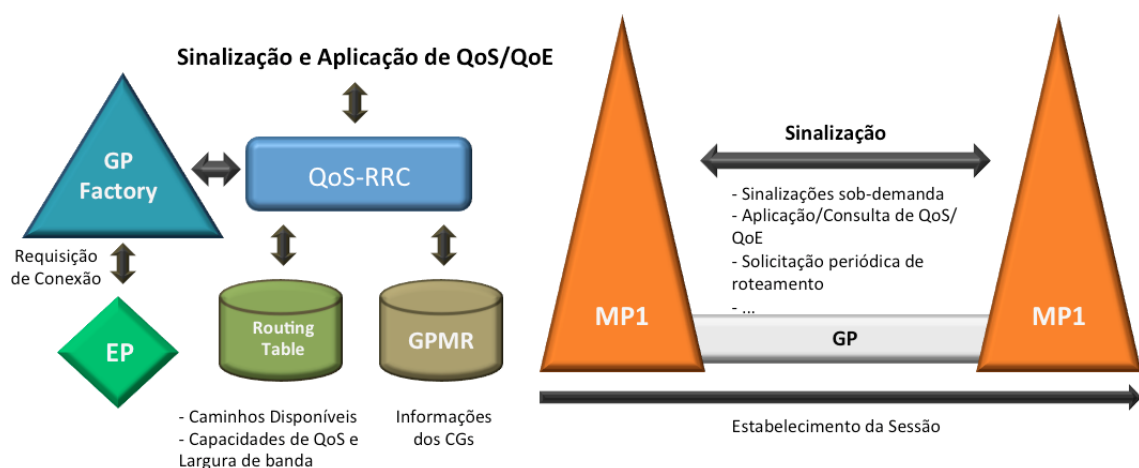


Figura 4.1: Mecanismo QoS-RRC na Arquitetura GP

O QoS-RRC encontra-se hospedado nos nós que implementam os MPs realizando roteamento e QoS das funcionalidades. Eventos de roteamento dinâmico podem ser fornecidos por sinalizações assíncronas, como quando uma interface de rede cai ou

torna-se ativa. Além disso, sinalizações periódicas também podem ocorrer uma vez que são necessárias atualizações dos parâmetros de QoS para escolha dos melhores caminhos para novas sessões e re-roteamento delas, no caso de queda de enlace ou roteador.

Este capítulo é estruturado da seguinte forma. Na Seção 4.1 demonstramos a organização da Arquitetura QoS-RRC. Na Seção 4.2 descrevemos as interfaces internas e externas utilizadas pelo mecanismo para fornecer conectividade com outros componentes. A Seção 4.3 especifica o protocolo do mecanismo QoS-RRC e na Seção 4.4 os agentes das bordas e do núcleo da rede. Por fim, a Seção 4.5 apresenta todas as tabelas de estado utilizadas para armazenar as informações necessárias ao funcionamento do QoS-RRC.

4.1 Arquitetura QoS-RRC

A arquitetura do QoS-RRC foi projetada de forma modular e flexível para estabelecer uma independência entre seus componentes e permitir com que seus módulos possam ser executados concorrentemente. Além disso, esta modularidade permite que outros componentes sejam inseridos de forma facilitada, uma vez que, o novo módulo só precisa implementar as interfaces de comunicação e configuração dos recursos do QoS-RRC. A Figura 4.2 apresenta todos os módulos da arquitetura QoS-RRC:

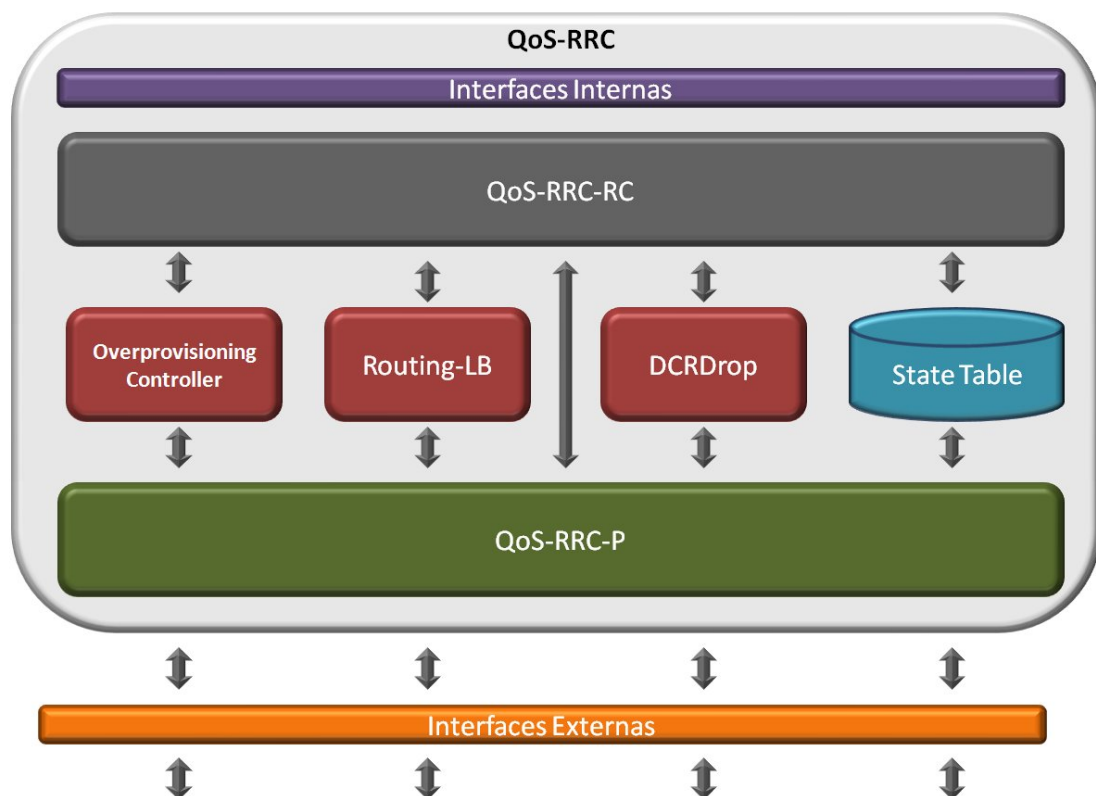


Figura 4.2: *Arquitetura QoS-RRC*

- **Interfaces Internas:** estas interfaces são aquelas especificadas para a comunicação interna entre módulos do QoS-RRC;
- **Interfaces Externas:** estas interfaces são aquelas especificadas para a comunicação com módulos externos ao mecanismo QoS-RRC. Elas permitem, por exemplo, o acoplamento do QoS-RRC à arquitetura GP;
- **QoS-RRC Protocol (QoS-RRC-P):** este módulo é responsável por fornecer informações de controle para que o QoS-RRC-RC defina o estado dos elementos da rede e configure os recursos da rede;
- **QoS-RRC-RC (QoS-RRC Resource Controller):** é o mecanismo responsável pela interação com outros elementos do sistema, visando a instalação correta de QoS (Reserva de largura de banda por classe de serviço) e conectividade (árvores de distribuição multicast), bem como, a manipulação de estado;
- **Overprovisioning Controller:** responsável por realizar o provisionamento de recursos e de árvores da rede. O QoS-RRC faz o provisionamento de largura de banda por classe de serviço no momento da inicialização da rede e caso seja necessário, reajustes dinâmico em tempo de execução também são permitidos. Ele também faz o provisionamento de árvores de distribuição multicast;
- **Routing with Load Balanced (Routing-LB):** o balanceamento de carga e o encaminhamento de todos os fluxos pertencentes a sessão é realizado por este módulo, fornecendo para cada sessão QoS. O Routing-LB é ativado: i) A cada nova sessão solicitada; ii) Em casos de detecção de congestionamento na rede e, iii) Falha nos roteadores;
- **Detection Congestion or Router Drop (DCRDrop):** este módulo foi desenvolvido para detectar congestionamento nas filas dos roteadores da rede. Para isso ele adota o mecanismo de gerenciamento ativo de filas RED (Random Early Detection) [35], que foi adaptado para atender aos requisitos de QoS-RRC;
- **State Table:** esta tabela contém todas as informações de estado da rede, incluindo tabelas de roteamento, métricas de QoS por classe de serviço e por árvore, sessões ativas, entre outros.

4.1.1 Sentido *Up-stream* e *Down-stream* da rede

Para uma melhor compreensão da troca de mensagens e envio de fluxos de dados à rede, resolvemos diferenciar os sentidos aos quais as informações trafegam na rede. O sentido *up-stream* corresponde ao envio de mensagens do *Core-Router* (CR) e *Egress Router* (ER) ao *Ingress Router* (IR). Estas mensagens podem relatar ao IR sucesso de alguma operação solicitada ou alertar sobre um congestionamento na rede. O sentido *down-stream* corresponde ao envio de mensagens do IR ao CR e ER. A partir deste

sentido, o IR pode encaminhar fluxos de dados a rede, enviar mensagens de configuração ou alteração dos parâmetros da rede, atualizar as tabelas de estado com as capacidades de QoS e detectar possíveis falhas na rede, como por exemplo queda de enlace ou falha no roteador.

4.2 Interfaces Internas e Externas

Para que todos os módulos do QoS-RRC possam se comunicar internamente e, mecanismos externos, possam utilizar os serviços oferecidos pelo QoS-RRC, definimos interfaces, interna e externa respectivamente, de acordo com o nível de comunicação. A definição de interfaces internas permite, unificar a troca de informações entre os módulos e facilitar a inserção de novos componentes, o que garante extensibilidade do mecanismo QoS-RRC. Quanto às interfaces externas, a padronização permite com que outros mecanismos que queiram ter acesso aos serviços oferecidos pelo QoS-RRC, apenas tenham que implementar as interfaces definidas. Isso agrega ainda mais valor ao mecanismo QoS-RRC, quanto a extensibilidade. Em nosso caso específico, arquitetura GP deverá implementar esta interface para alimentar a GPMR com informações necessárias a criação dos GPs.

Para especificação das interfaces, utilizamos a terminologia adotada no modelo *Open System Interconnection* (OSI)[6]. Neste modelo, as primitivas de serviços são divididas em quatro classes e o serviço é orientado a conexão:

- *Request*: pedido enviado pelo módulo que solicita uma operação;
- *Indication*: através dessa primitiva, o módulo é informado de uma solicitação de operação.
- *Response*: resposta de um módulo a um pedido de operação;
- *Confirmation*: o módulo solicitante é informado do resultado da operação.

4.3 QoS-RRC Protocol (QoS-RRC-P)

Esta seção introduz o QoS-RRC Protocol, que faz parte da *suíte* QoS-RRC. Nela, descreveremos os detalhes de suas funcionalidades bem como as mensagens que foram especificadas.

O QoS-RRC-P é usado para troca de informações de controle, permitindo sob-demanda, realizar a alocação dos recursos da rede, instalar as árvores multicast utilizadas para encaminhamento de dados das sessões e alertar sobre possíveis problemas ocorridos na rede. Além de oferecer suporte a QoS por-classe, o QoS-RRC-P permite fornecer informações de IP multicast para suporte as árvores de distribuição SSM. Portanto, o

caminho de comunicação é sinalizado quando uma requisição é recebida de modo a a fornecer informações de controle relacionadas a QoS (valor de largura de banda para reservas por-classe) e conectividade (endereço IP multicast para controlar a construção das árvores de distribuição).

O QoS-RRC-P suporta dois modos operacionais para controle de árvores de distribuição, o *Default* e o modo *Optional*. Quando configurado com o modo operacional default, o QoS-RRC dá suporte ao protocolo multicast disponível, por exemplo PIM-SSM, para construir as árvores de distribuição multicast. O PIM-SSM cria árvores de distribuição multicast baseado na tabela de roteamento chamada *Multicast Routing Information Base* (MRIB) [16]. Uma vez que cada roteador é visitado pela mensagem QoS-RRC-P, a MRIB é atualizada com o endereço IP do nó anterior. No fim da alocação de recursos, o QoS-RRC aciona o protocolo para criação das árvores de distribuição multicast. A informação populada pelo QoS-RRC nas MRIBs força as mensagens do protocolo multicast a seguirem o caminho de comunicação com capacidade de QoS obtidos anteriormente.

O QoS-RRC-P distingue-se de abordagens fim-a-fim (YESSIR [11] e RSVP [32]) operando nas bordas da rede. Além disso, ele aumenta a escalabilidade usando alocação de recursos por-classe, ao invés de por-fluxo (RSVP e YESSIR).

Para especificação das mensagens, utilizamos a especificação QoS-NSLP [21] e reutilizamos alguns campos para composição de nossa mensagem. Dentre eles destacamos *Response Identification Information* (RII), *INFOSPEC* e alguns recipientes especificados no objeto QoS Specification (QSPEC). Dois novos objetos foram definidos para utilizarmos nas mensagens QoS-RRC-P, o *Reserved Path* (RSVPATH) e o *Multicast Specification* (MSPEC). O protocolo de transporte utilizado foi o UDP. A especificação funcional do QoS-RRC-P, incluindo objetos, flags e mensagens é apresentada a seguir.

4.3.1 Response Identification Information (RII)

O objeto *RII* é um identificador único, dentro do contexto da sessão, o qual é usado para associar uma mensagem *RESPONSE* com uma mensagem antecedente *RESERVE*. O *RII* de cada mensagem *RESERVE* é gerado no IR ao receber um pedido de requisição. Logo, a mensagem *RESERVE* e sua mensagem associada, *RESPONSE*, devem ter o mesmo RII na suíte QoS-RRC. A Figura 4.3 mostra o formato do objeto *RII*.

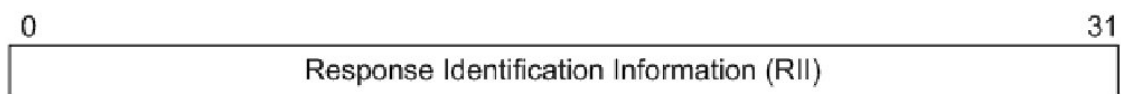


Figura 4.3: Especificação funcional do objeto *RII*

4.3.2 Reserved Path (RSVPATH)

O objeto *RSVPATH* carrega a lista de endereços IP dos roteadores visitados pela mensagem *RESERVE*, ao longo do caminho de comunicação. Cada endereço IP é uma palavra de 32 bits para IPv4 e 4 palavras de 32 bits para IPv6. Após o roteador ser visitado por uma mensagem de *RESERVE*, o roteador deve adicionar o endereço IP de sua interface de rede no final da lista criada no objeto *RSVPATH*. Portanto, o tamanho do objeto *RSVPATH* aumenta à medida em que se possui mais roteadores ao longo do caminho de comunicação. A Figura 4.4 mostra o formato do objeto *RSVPATH*.

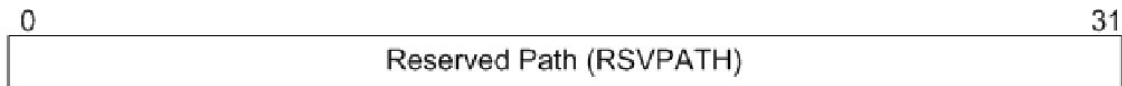


Figura 4.4: Especificação funcional do objeto *RSVPATH*

4.3.3 Multicast Specification (MSPEC)

O objeto *Multicast Specification* (MSPEC) carrega a identificação de um SSM da árvore de distribuição multicast (endereço IP da fonte e do grupo multicast). Cada endereço IP é uma palavra de 32 bits para IPv4 e 4 palavras de 32 bits para IPv6. A Figura 4.5 mostra o formato do objeto *MSPEC*.



Figura 4.5: Especificação funcional do objeto *MSPEC*

4.3.4 INFOSPEC

O objeto *INFOSPEC* é utilizado pelo QoS-RRC para fornecer feedback sobre a operação empregada (sucesso ou falha). Logo, o QoS-RRC utiliza esta informação para saber se a operação foi realizada ou não. O objeto possui dois campos de 16 bits, *error code* e *error class*, respectivamente. A Figura 4.6 mostra o formato do objeto *INFOSPEC*.



Figura 4.6: Especificação funcional do objeto *INFOSPEC*

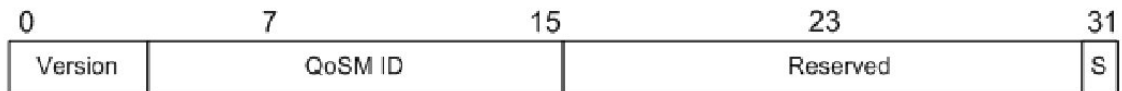
A partir da especificação QoS-NSLP, os códigos usados para fornecerem feedback sobre as operações são descritas na Tabela 4.1.

Tabela 4.1: Códigos de erro especificados na suíte QoS-RRC

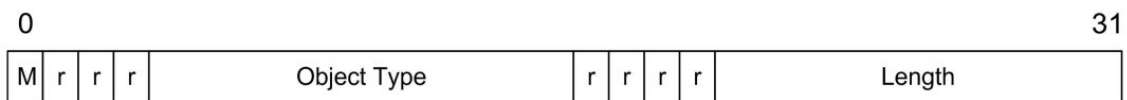
Error Code		E Class	
1	Informativa	1	Mudança de rota: mudança de rota ocorrida no caminho down-stream
2	Sucesso	1	Reserva com Sucesso
		2	Liberação dos recurso com sucesso
		3	Atualização com sucesso
		4	Modificação com sucesso
3	Falha	1	Falha na reserva: <i>Classe de Serviço (Class of Service - CoS)</i> indisponível
		2	Falha de reserva: largura de banda indisponível
		3	Falta de robustez: recursos indisponíveis
		4	Falha de modificação

4.3.5 QoS Specification (QSPEC)

O objeto *QoS Specification* (QSPEC) fornece informações requisitadas pelos agentes QoS-RRC para controle de estado dos recursos de QoS. O objeto QSPEC tem um cabeçalho comum e cada objeto que compõe o QSPEC, tem um cabeçalho objeto. A Figura 4.7 mostra o formato do cabeçalho comum QSPEC.

**Figura 4.7:** Especificação funcional do cabeçalho comum QSPEC

O campo *Version* é usado para identificar a versão do QSPEC atualmente usado. O campo *QoSM ID* é preenchido com o código de identificação do QoSM em uso, sendo o QoS-RRC neste caso. A Figura 4.8 mostra o formato do cabeçalho objeto QSPEC.

**Figura 4.8:** Especificação funcional do cabeçalho objeto QSPEC

A flag *M*, quando definida como *TRUE* = 1, indica que o parâmetro seguinte deve ser examinado. Caso contrário, o parâmetro pode ser ignorado. O *Object Type*, quando definido como *FALSE* = 0, indica que o cabeçalho está associado com o objeto *QoS Desired*, caso contrário (definido como *TRUE* = 1), o cabeçalho está associado com o objeto *QoS Available*. *Length* é a quantidade de parâmetros.

No QSPEC do QoS-RRC-P, o *QoS Description* é dividido em objetos *QoS Desired* e *QoS Available*, como mostrado na Figura 4.9. O primeiro é utilizado como suporte a alocação de recursos de QoS (reserva de banda), e o segundo para descrever a capacidade atual recursos suportados no gargalo do caminho de comunicação. A Figura 4.9 descreve os objetos que compõem o QSPEC do QoS-RRC.

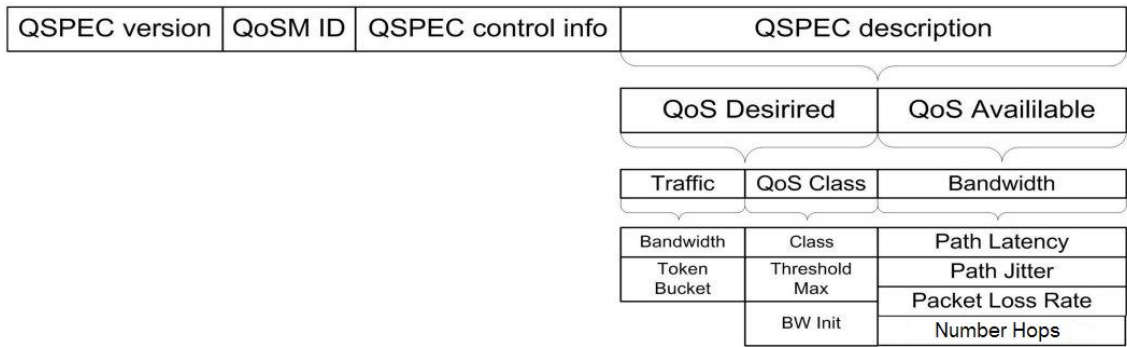


Figura 4.9: Especificação funcional do QSPEC [21]

O objeto *QoS Desired* carrega o tráfego (*Traffic*) e o objeto *QoS Class*. O objeto *Traffic* descreve a largura de banda exigida pelo fluxo e os parâmetros do *token bucket*. O objeto *Class* informa o código da CoS atribuída ao fluxo. O objeto *Class* pode ser preenchido com uma lista de códigos de CoS, no caso de uma operação de atualização. A Figura 4.10 descreve o formato do objeto *QoS Desired*.

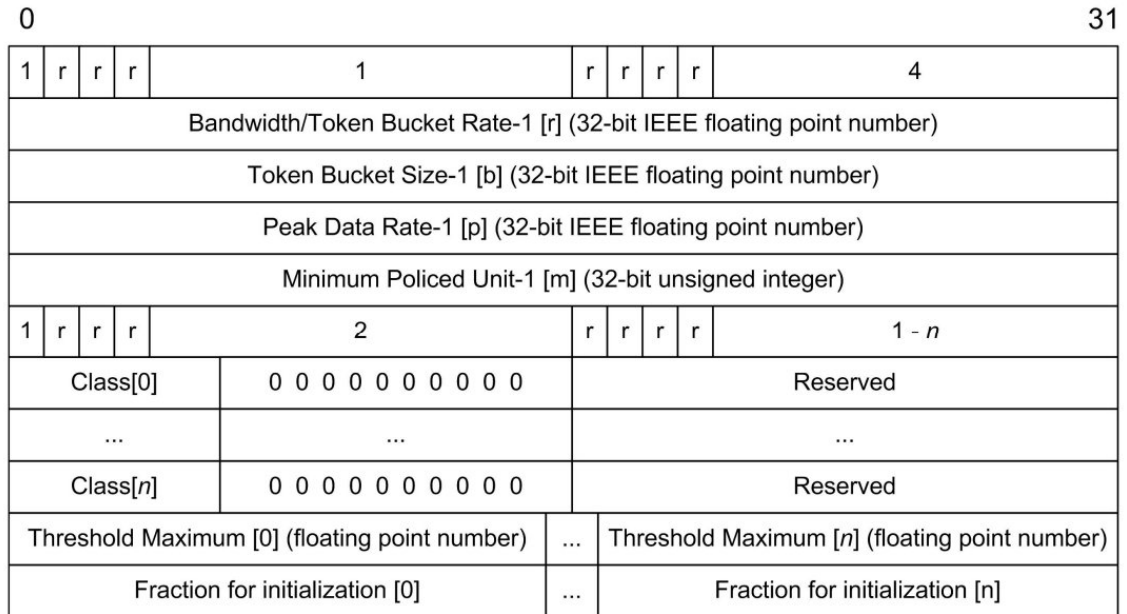


Figura 4.10: Especificação funcional do objeto *QoS Desired* [21]

O objeto *QoS Available* é mostrado na Figura 4.11. Este objeto descreve as capacidades QoS de um enlace no caminho de comunicação, em termos de largura de banda, latência caminho, variação de atraso do caminho, e a razão de perda de pacotes por CoS. Assim, um agente QoS-RRC somente atualiza as informações, adicionado-as ao objeto *QoS Available*, quando as capacidades locais de QoS, coletadas da *Management Information Base* (MIB), são menores.

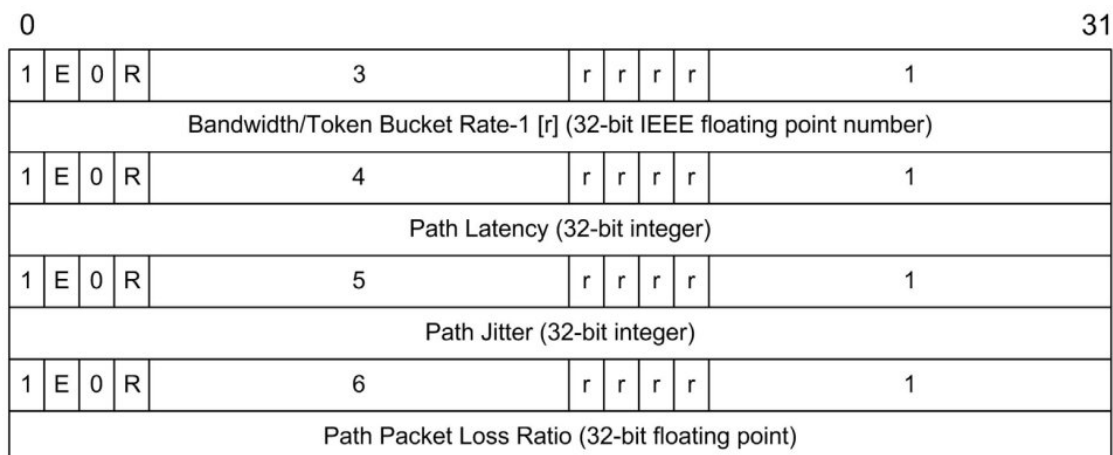


Figura 4.11: Especificação funcional do objeto QoS Available [21]

4.3.6 Mensagens QoS-RRC

A complexidade das operações realizadas pelo QoS-RRC é minimizada através da aplicação de apenas duas mensagens de sinalização: *RESERVE* e *RESPONSE*. Todas as mensagens QoS-RRC-P devem conter um cabeçalho comum, com os primeiros 32 bits da mensagem. A Figura 4.12 mostra o formato do cabeçalho comum.



Figura 4.12: *Formato comum das mensagens utilizadas no QoS-RRC-P*

Os campos do cabeçalho são:

- *Message Type*: campo de 8 bits, quando configurado com *00* indica uma mensagem *RESERVE*, e quando configurado com *01*, indica uma mensagem *RESPONSE*;
- *Flags específicas da mensagem*: campo de 9 bits, como mostrados na Tabela 4.2.

A seguir mostramos a especificação funcional das mensagens de sinalização utilizadas para alocação de recursos.

4.3.7 Mensagem *RESERVE*

A mensagem *RESERVE* consiste de uma sinalização unicast que transporta informações de controle para ativar o QoS-RRC-RC para instalar, modificar, lançar ou coletar o estado dos recursos de rede. A alocação de recursos é solicitada enviando a mensagem *RESERVE* down-stream, e assim, cada roteador visitado opera de acordo com

Tabela 4.2: Descrição das flags especificadas no QoS-RRC

Tipo de Flag	Descrição
INIT (I)	Quando configurada como <i>TRUE</i> = 1, indica uma mensagem solicitando a inicialização do sistema.
TEAR (T)	Quando configurado como <i>TRUE</i> = 1, indica que os valores de QoS carregados pela mensagem devem ser atualizados.
ALTERATE (A)	Quando configurado como <i>TRUE</i> = 1, indica que o valor do estado descrito no <i>QSPEC</i> , que já está presente no roteador, deve ser modificado.
REFRESH (R)	Quando configurado como <i>TRUE</i> = 1, indica que o valor dos estados dos recursos devem ser atualizados.
MULTICAST (M)	Quando configurado como <i>TRUE</i> = 1, indica a requisição de instalação de árvores multicast.
VERSION (V)	Quando configurado como <i>TRUE</i> = 1, indica que o IPv6 está sendo utilizado, caso contrário, o IPv4.
UPDATE (U)	Quando configurado como <i>TRUE</i> = 1, indica a coleta de parâmetros de QoS da rede, para atualização no IR.
CONGESTION (C)	Quando configurado como <i>TRUE</i> = 1, indica que existe um congestionamento na rede. O objeto <i>INFOSPEC</i> deverá conter o endereço IP do roteador com congestionamento.
LINK_DOWN (L)	Quando configurado como <i>TRUE</i> = 1, indica que ocorreu alguma falha na rede, sendo por exemplo, queda de enlace ou falha no roteador. O objeto <i>INFOSPEC</i> deverá conter o endereço IP do roteador com falha.

a funcionalidade indicada. Ao ativar algumas das flags especificadas anteriormente na Tabela 4.2, a mensagem *RESERVE* faz com que o QoS-RRC-RC realize algumas das ações:

- Alocar recursos de QoS e conectividade através do envio de mensagens *RESERVE* no sentido down-stream, após as verificações de controle de admissão implementadas, com as flags *R* e *T* definidas como *FALSE* = 0. No que diz respeito aos recursos de conectividade, a flag *M* definida como *TRUE* = 1 indica um pedido de multicast. Assim, uma árvore de distribuição multicast *SSM* deve ser construída;
- Modificar o valor de estado enviando mensagens *RESERVE* no sentido down-stream com a flag *A* configurada como *TRUE* = 1. Por exemplo, o QoS-RRC-RC pode modificar o estado associado a uma sessão, que pode ser necessário em condições de re-roteamento;
- Atualizar estado dos recursos através do envio de mensagens unicast *RESERVE* no sentido down-stream, com a flag *R* configurada como *TRUE* = 1. Além disso, QoS-RRC-RC verifica quando o caminho de comunicação atual está em conformidade

com a lista de roteadores que estão sendo transportados no objeto *RSVPATH*, o que aumenta a robustez do protocolo;

- Liberação dos recursos através do envio de uma mensagem unicast no sentido up-stream com a flag *T* definida como $TRUE = 1$. Os atuais valores dos recursos de QoS são transportados no objeto *QSPEC* e, quando aplicável, no *MSPEC*, para o estado de multicast;
- Coletar o estado dos recursos através do envio de mensagens unicast no sentido down-stream com a flag *C* definida como $TRUE = 1$. De acordo com o tipo de recurso, o valor do estado é preenchido nos objetos *QSPEC*, *MSPEC* e *RSVPATH*. A Figura 4.13 mostra a formato da mensagem *RESERVE*.

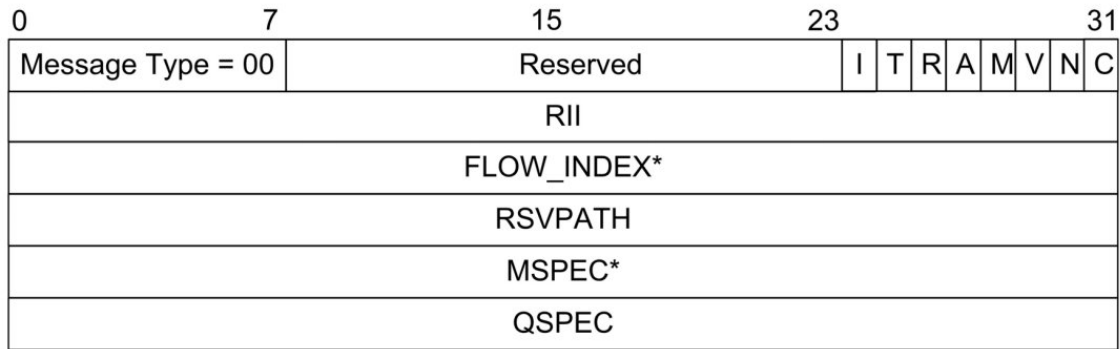


Figura 4.13: Formato da mensagem *RESERVE*

Além do objeto *RII*, uma mensagem *RESERVE* deve conter os objetos *RSVPATH* e *QSPEC*. O objeto *MSPEC* é opcional, sendo usado apenas em pedidos de fluxos multicast.

4.3.8 Mensagem *RESPONSE*

A mensagem *RESPONSE* consiste de uma sinalização unicast usada principalmente como feedback, sendo preenchida de informações no objeto *INFOSPEC*. Em contraste com o que é comumente aplicado pelos protocolos de sinalização disponíveis (por exemplo, RSVP e RMD-QoS), o QoS-RRC-RC também usa a mensagem *RESPONSE* para dar apoio à liberação de recursos imediatamente quando ocorre uma falha na alocação dos recursos. A mensagem *RESPONSE* visita os roteadores fazendo o percurso do caminho inverso ao percorrido pela mensagem *RESERVE*, indicando a quantidade de recursos liberados no *QSPEC*. A Figura 4.14 mostra o formato da mensagem *RESPONSE*.

0	7	15	23	31									
Message Type = 01		Reserved				I	T	R	A	M	V	N	C
RII													
RSVPATH													
INFOSPEC													
MSPEC*													
QSPEC*													

Figura 4.14: Formato da mensagem *RESPONSE*

Além disso, a mensagem *RESPONSE* transporta informações sobre a capacidade de recursos da rede atualmente disponíveis no caminho de comunicação, que é preenchido no objeto *QoS Available* (definido no *QSPEC*). A mensagem *RESPONSE* é enviada no sentido up-stream quando resultado de operação de alocação de recursos, coleta ou atualização e, down-stream quando resultado de uma operação de liberação de recursos. A mensagem *RESPONSE* deve levar um objeto *INFOSPEC*, que é preenchido com informações sobre o sucesso (*OK*) da instalação de uma reserva ou uma condição de erro (*NOK*) (de acordo com as informações fornecidas na Tabela 4.1). Dependendo das informações obtidas a partir do *INFOSPEC*, a mensagem *RESPONSE* pode também conter objetos *RSVPATH*, *MSPEC* e *QSPEC*.

4.4 Agentes QoS-RRC

O mecanismo QoS-RRC utiliza-se de agentes para executar suas operações. Estes agentes são instalados em todos os roteadores da rede. Para isso, desenvolvemos dois tipos diferentes de agentes: i) Agentes QoS-RRC-E (Edge - StateFull), utilizados na borda da rede e, ii) Agentes QoS-RRC-C (Core - LightState). Esse tratamento diferenciado, de acordo com a região do roteador na rede, permite com que os recursos destes dispositivos sejam utilizados da maneira mais adequada, trazendo a complexidade para as bordas da rede, e deixando o núcleo com menor carga de processamento. A Figura 4.15 apresenta a localização dos agentes na rede.

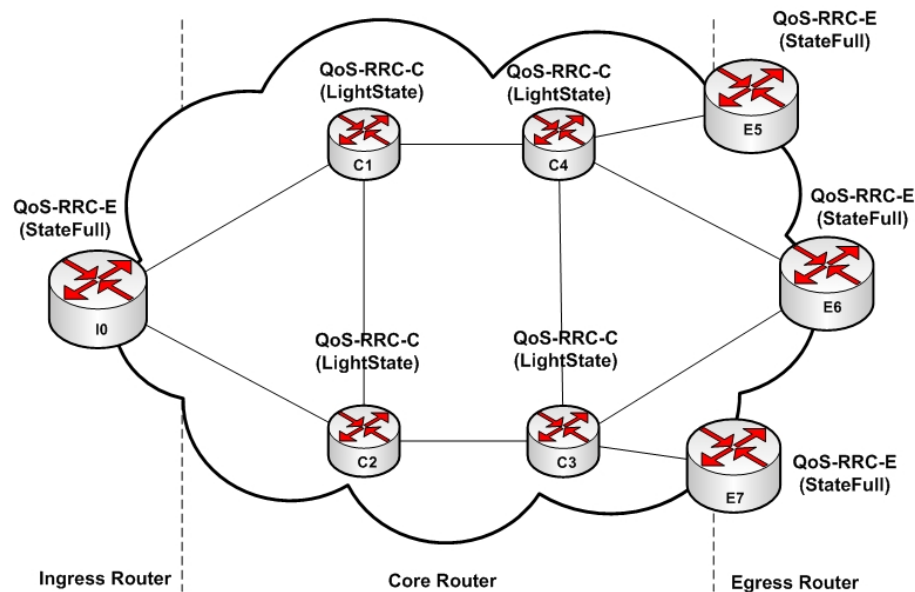


Figura 4.15: Agentes *QoS-RRC*

Os agentes QoS-RRC-E possuem as seguintes funcionalidades:

- Aprovisionamento de recursos da rede, quanto a QoS e árvores multicast;
- Seleção das melhores rotas a cada novo pedido de requisição de sessão;
- Cálculo de custo dos caminhos;
- Re-roteamento com balanceamento de carga em casos de detecção de congestionamento ou falhas na rede;
- Suporte a mecanismos externos, por exemplo, arquitetura GP;
- Armazenamento de tabelas de estado de toda rede, contendo, informações de QoS por CoS e por árvore, sessões ativas, árvores multicast instaladas, endereço IP de todos os roteadores por árvore, dentre outros.

Os agentes QoS-RRC-C possuem as seguintes funcionalidades:

- Encaminhamento dos fluxos de dados no sentido down-stream da rede;
- Notificação de congestionamento da rede através de mensagens de alerta, no sentido up-stream;
- Armazenamento de tabelas de estado locais contendo os níveis de QoS atuais por CoS, árvores multicast instaladas, dentre outros.

4.5 Tabelas de Estado (*State Table*)

As tabelas de estado armazenam informações dos roteadores da rede. Estas tabelas possuem conteúdos diferentes, dependendo do agente QoS-RRC instalado no roteador. Nesta seção, apresentaremos as informações armazenadas em cada uma destas

tabelas de estado. Na Sub-Seção 4.5.1 apresentamos as tabelas de estado dos roteadores da borda e na Sub-Seção 4.5.2 dos roteadores do núcleo.

4.5.1 Tabelas de estado QoS-RRC-E

Como relatado na Seção 4.4, os roteadores da borda possuem maior capacidade de processamento e execução das operações. Sendo assim, suas tabelas de estado são mais robustas de modo que o IR tenha ciência de toda rede. Abaixo, apresentamos todas as tabelas juntamente com uma breve descrição de todos seus campos. A Tabela 4.3 armazena todas as árvores da rede. Na Tabela 4.4, apresentamos as sessões ativas e por fim, na Tabela 4.5 os fluxos relacionados às sessões.

Tabela 4.3: *Tabela de Árvores QoS-RRC-E*

Tree Index*	IP Multicast	RSVPATH	Cost ClassX*	IP ER	BW Reserved
16 bits	32 bits (IPv4)	32 bits	32 bits	32 bits (IPv4)	16 bits
	4x32 bits (IPv6)			4x32 bits (IPv6)	

* Existem 5 custos diferentes para cada árvore, que podem ser estendidos para mais classes, de acordo com a necessidade da aplicação. Este custo é calculado levando em consideração a classe o tipo de aplicação.

- **Tree Index***: índice da árvore;
- **IP Multicast**: endereço IP da árvore multicast;
- **RSVPATH**: contém os endereços IPs dos roteadores ao longo do caminho de comunicação/árvore;
- **Cost Class1**: representa o custo da árvore para a aplicações da Classe 1;
- **Cost Class2**: representa o custo da árvore para a aplicações da Classe 2;
- **Cost Class3**: representa o custo da árvore para a aplicações da Classe 3;
- **Cost Class4**: representa o custo da árvore para a aplicações da Classe 4;
- **Cost Class5**: representa o custo da árvore para a aplicações da Classe 5;
- **IP ER**: endereço IP do ER.

Tabela 4.4: *Tabela de Sessões QoS-RRC-E*

Session Index**	Tree Index*	IP Destination
16 bits	16 bits	32 bits (IPv4) 4x32 bits (IPv6)

- **Session Index****: índice da sessão;
- **Tree Index***: chave secundária para Tabela 4.4;
- **IP Destination**: endereço IP de destino.

Tabela 4.5: *Tabela de Sessões QoS-RRC-E*

Flow Index	Session Index**	CoS
16 bits	16 bits	16 bits

- **Flow Index**: índice do fluxo;
- **Session Index****: chave secundária para Tabela 4.4;
- **CoS**: classe de serviço.

4.5.2 Tabelas de estado QoS-RRC-C

Nesta Sub-Seção, apresentamos as tabelas de estado utilizadas pelos CRs. Essas tabelas possuem menos informações que aquelas contidas na borda, de modo a não sobrecarregar o núcleo da rede. Abaixo, apresentamos os campos que as compõem juntamente com uma breve descrição deles. A Tabela 4.3 armazena todas as árvores da rede. Na Tabela 4.6 apresentamos a tabela de roteamento Up-stream e na Tabela 4.7, no sentido down-stream .

Tabela 4.6: *Tabela de Roteamento QoS-RRC-C (Sentido Up-stream)*

Network ID	IP Address	Network Mask	MAC Address	CoS	Delay	Jitter	Loss
16 bits	32 bits (IPv4) 4x32 bits (IPv6)	32 bits	48 bits	16 bits	32 bits	32 bits	32 bits

- **Network ID**: é responsável pela identificação do sistema autônomo a partir do qual a mensagem foi enviada;
- **IP Address**: este campo contém o endereço IP do destinatário, o qual os roteadores do núcleo utilizarão para enviar suas mensagens de alerta;
- **Network Mask**: usada para obter o IP correspondente à rede pública, à sub-rede e aos hosts;
- **MAC Address**: endereço MAC da interface de rede;
- **CoS**: possui a classe de serviço da sessão;
- **Delay**: contém o atraso da interface de rede;

- **Jitter:** contém a variação do atraso da interface de rede;
- **Loss:** contém a taxa de perda de pacotes da interface de rede.

Tabela 4.7: *Tabela de Roteamento QoS-RRC-C (Sentido Down-stream)*

IP Multicast	MAC Address
32 bits (IPv4)	48 bits
4x32 bits (IPv6)	

- **IP Multicast:** endereço IP da árvore multicast;
- **MAC Address:** endereço MAC da interface de rede.

Tendo estas informações, será possível aos roteadores do núcleo encaminhar os dados no sentido down-stream e enviar informações do contexto da rede no sentido up-stream.

Funcionalidades do QoS-RRC

O mecanismo QoS-RRC apresenta funcionalidades relacionadas aos módulos de sua arquitetura. Dentre elas, podemos destacar:

- Aprovisionamento de recursos de QoS e de árvores multicast;
- Roteamento de dados no sentido up-stream e down-stream;
- Balanceamento de carga;
- Detecção de congestionamento e falha na rede.

A partir de suas características, iremos descrever a seguir cada uma das funcionalidades destes módulos. Este capítulo divide-se da seguinte forma. A Seção 5.1 apresenta o módulo de provisionamento de recursos de QoS e de árvores multicast. A Seção 5.2 descreve o protocolo de roteamento e na Seção 5.3 o mecanismo de detecção de congestionamento. Por fim, na Seção 5.4 apresentamos o mecanismo de balanceamento de carga do QoS-RRC.

5.1 Aprovisionamento de Recursos de QoS e Árvores Multicast

O provisionamento de recursos de QoS e de árvores multicast do QoS-RRC é especificado com base na solução patenteada em [3] e em [2], respectivamente. A idéia principal consiste de, configurar as filas dos roteadores por classe de serviço e instalar as árvores de agregação multicast.

A suíte QoS-RRC fornece suporte a QoS através da implementação de controle de admissão e eficiente provisionamento da largura de banda para permitir roteamento com QoS. Isto permite manter a eficiência geral do sistema e melhorar a experiência do usuário com a aplicação. Nesse sentido, o controle/predição de largura de banda agregada é melhor que aqueles por-fluxo [32], pois estes são bem conhecidos como ineficientes devido a sobrecarga causada pelo controle/manutenção de um grande número de variáveis [22][26]. O QoS-RRC também controla largura de banda por meio de

um mecanismo de aprovisionamento, o qual inicializa as sobre-reservas por classe na inicialização da rede (baseado nos parâmetros atribuídos pelo operador da rede) e também faz um redimensionamento de forma dinâmica [1][4], levando em conta estatísticas das condições da rede, demanda de recursos e classes-políticas. A integração de controle de admissão e aprovisionamento de recursos por-classe provê melhorias de desempenho da rede, onde várias sessões e GPs podem ser instalados sem sinalização por-fluxo, tanto quanto possível.

Na inicialização da rede, o QoS-RRC primeiramente realiza um *flooding* (inundação) em todo o sistema através do envio de mensagens *RESERVE*, com a flag *I* ativada, realizando assim sobre-reservas por-classe, tendo em conta um fator de inicialização atribuído pelo operador de rede (por exemplo, $1/2$ ou $1/4$ da capacidade do enlace local). Para evitar operações redundantes, cada nó deve fazer algumas verificações antes de configurar os recursos como por exemplo, garantir que aprovisionamento por classe é inicializado apenas uma. Depois disso, cada roteador visitado pela mensagem acrescenta em seu campo *RSVPATH* seu endereço IP local. Além das informações sobre o caminho, os parâmetros de QoS também são analisados, sendo substituídos quando as informações locais são melhores que aquelas contidas no campo *QoS Available*. Estes parâmetros são largura de banda, delay, jitter e loss. Por fim, ao chegar ao ER uma nova mensagem é composta, denominada *RESPONSE*, contendo a flag *OK*. Posteriormente, ele envia esta mensagem para o IR correspondente, terminando assim com todas as informações do caminho de comunicação.

Com base nos caminhos unicast recebidos, o IR inicia a composição dos caminhos multicast. Uma vez que um grande número de árvores pode ser criado, o QoS-RRC filtra e mantém somente as melhores árvores segundo alguns critérios estabelecidos em [2]. Por exemplo, árvores que possuem IR no centro da rede são descartadas de maneira a manter somente fluxos no sentido down-stream. Após filtrar estas árvores, o QoS-RRC cria endereços IPs multicast para um conjunto deles e envia para dentro da rede a mensagem *RESERVE(M)*, responsável por criar as árvores multicast nos roteadores da rede. Para instalação destas árvores, utilizamos o protocolo PIM-SSM [16].

A fim de esclarecer as operações de aprovisionamento de recursos do QoS-RRC, a Figura 5.1 mostra um diagrama que resume a sequência de eventos introduzidas por uma sessão multimídia de tempo-real. Durante o processo de criação de um GP para o estabelecimento de uma sessão (multi-party), a GPMR deve acionar o QoS-RRC com um pedido de requisição de sessão contendo informações sobre a identificação da sessão e os requisitos de QoS (bit-rate, classe de serviço, atraso, variação do atraso e perda). Após isso, o QoS-RRC deve executar o algoritmo de roteamento, de modo a escolher o melhor caminho de dados. A decisão leva em consideração os nós que hospedam as entidades envolvidas na comunicação (para escolher os caminhos candidatos) e os requisitos de

QoS da sessão solicitada. Baseado no atual nível de qualidade de cada caminho candidato, o QoS-RRC escolhe o mais adequado (verificações de controle de admissão) e aceita a sessão multimídia de tempo real. Em seguida, ele aciona o MP para estabelecer a conexão. Depois de realizar todas as configurações, a GPMR é alimentada adequadamente para realizar a criação do GP que permita a propagação dos dados da sessão.

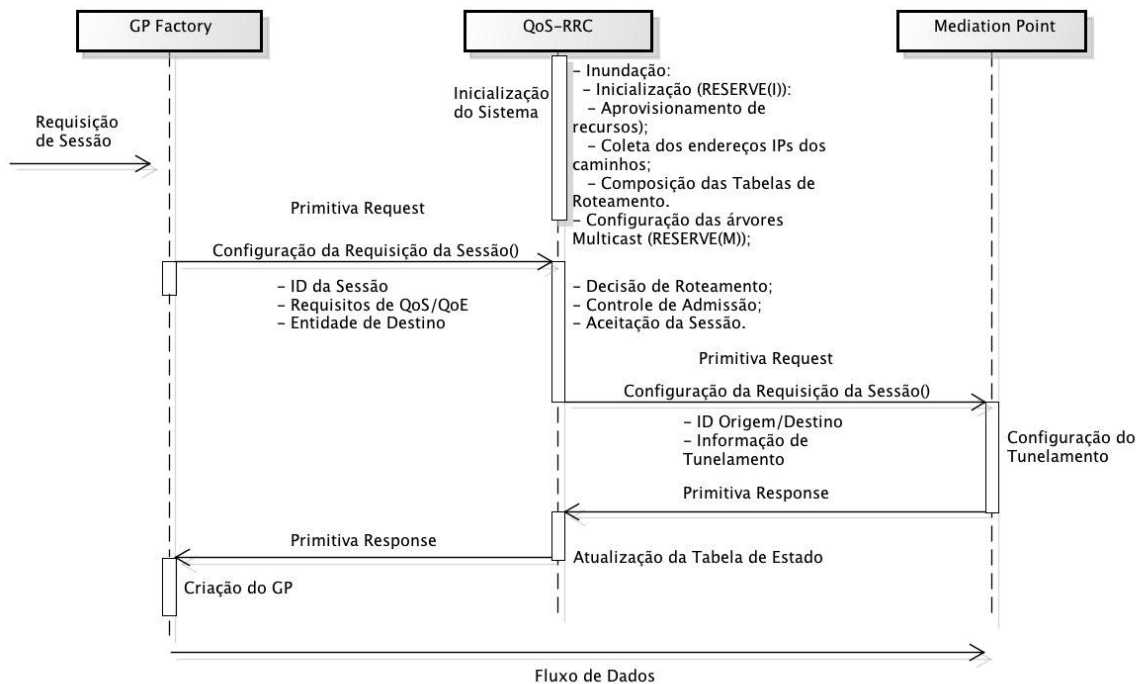


Figura 5.1: Diagrama de Eventos QoS-RRC

Um mecanismo de controle de conectividade também é suportado pelo QoS-RRC, o que permite agregar várias sessões para um mesmo GP. Esta técnica de agregação pretende reduzir a sobrecarga de estado e de processamento dentro da rede, um legado do multicast IP por fluxo, permitindo suporte escalável a multi-party. Além disso, o aprovisionamento de recursos fornece resiliência, o que é extremamente requerido em ambientes heterogêneos. Por exemplo, eventos de redes dinâmicas, tais como falha de enlaces ou *handovers* (vertical ou horizontal) exigem re-roteamento das sessões. O QoS-RRC implementa esse re-roteamento em diferentes níveis, onde, em primeiro lugar, tenta-se redirecionar os fluxos de um GP para outro caminho. Caso não seja bem sucedido (por exemplo, o novo caminho não suporta todas as sessões), o QoS-RRC pode usar outros GPs existentes (em outros caminhos) ou criar um outro GP para as sessões. Assim, o QoS-RRC permite rápido suporte a resiliência, a comunicações intermitentes e mantém sessões de multimídia em tempo real com bom nível de qualidade, mesmo durante anomalias na rede.

5.2 Roteamento no QoS-RRC

O QoS-RRC foi projetado com um módulo de roteamento chamado Routing-LB (Roteamento com Balanceamento de Carga), que apoia-se nos requisitos da sessão e no contexto atual da rede para escolher o melhor caminho. Juntamente com o roteamento é feito o balanceamento de carga, que distribui de maneira uniforme as sessões entre os caminhos de dados.

Nesta seção, mostraremos como é realizada a composição das tabelas de roteamento e como são tratados pedidos de requisição de novas sessões com a aplicação do algoritmo de escolha do melhor caminho que foi desenvolvido.

5.2.1 Composição das Tabelas de Roteamento

O primeiro passo para a realização de um roteamento é a composição de tabelas nos roteadores da rede. A composição destas tabelas tem seu início no momento em que a mensagem *RESERVE(I)* é enviada para dentro da rede. Neste momento, são compostas as tabelas de roteamento down-stream. Cada mensagem enviada contém o endereço IP do roteador remetente e, com base nisso, os roteadores do núcleo associam este endereço IP à interface de rede pela qual a mensagem foi recebida. Além disso, outros parâmetros são adicionados a tabela de roteamento e à mensagem, por exemplo, as métricas de QoS de cada interface (o que permitirá calcular o custo de envio da mensagem para cada uma das interfaces de rede locais) e o identificador do sistema autônomo do qual a mensagem foi enviada. Por fim, ao chegar no ER uma nova mensagem é composta, a *RESPONSE(OK)* que realizará a composição das tabelas de roteamento no sentido up-stream. Ela realiza as mesmas operações anteriores, atribuindo o endereço IP à interface de rede pela qual a mensagem foi recebida. Como já foi mostrado, na Seção 4.5 apresentamos as tabelas de roteamento compostas para realização do roteamento na rede.

5.2.2 Métricas de Roteamento

Com o intuito de se determinar os melhores caminhos de uma rede, protocolos de roteamento utilizam-se de métricas que auxiliam na avaliação de cada rota. Com isso, realizamos um estudo aprofundado [14][17][37] e selecionamos três métricas de roteamento que visam atender os requisitos da arquitetura GP [33] e auxiliar na escolha do melhor caminho. As métricas escolhidas foram delay, jitter e loss. A partir destas métricas, utilizamos a Recomendação ITU-T G.1010 [39] que sugere alguns parâmetros aceitáveis para elas e compomos funções que representam o custo de cada caminho. As funções foram criadas para uma variedade de aplicações que envolvem meios de comunicação como voz, vídeo, imagem, áudio e texto e, buscam levar satisfação ao usuário final. Nesse

sentido, apesar das métricas escolhidas serem em nível de QoS, os parâmetros aceitáveis para elas foram criados observando a perspectiva do usuário. Logo, temos também uma melhoria do mecanismo em nível de QoE. Abaixo apresentamos uma breve descrição das métricas de QoS utilizadas:

- **Delay (Atraso):** atraso é o tempo que um pacote de dados gasta para sair de uma origem e chegar a um determinado destino. Esse atraso pode ocorrer por diversos motivos, dentre os quais podemos citar: i) atraso de processamento; ii) atraso nas filas dos roteadores; iii) atraso para percorrer todo o enlace; dentre outros;
- **Jitter (Variação do atraso):** representa a variação do atraso e na maior parte dos casos é considerado um parâmetro de desempenho. Isto acontece pois ele é um importante parâmetro na camada de transporte, devido a variação da chegada dos pacotes individuais. Apesar de existirem mecanismos que tentam eliminar ou evitar esses atrasos ocorridos, por exemplo buffers, existem aplicações que são altamente sensíveis ao jitter, como por exemplo, aplicações de voz. Como a voz é codificada, comprimida e fragmentada em pequenos pacotes antes do envio dos dados, cada pacote pode seguir um caminho diferente na rede. Com isso, uma re-ordenação dos pacotes ou mesmo variações de atraso de um pacote para outro podem ocorrer, o que pode implicar na distorção da voz ou outra anomalia dos dados recebidos;
- **Loss (Perda de pacotes):** A perda de dados tem um impacto direto na qualidade do serviço oferecido aos usuários. Arquivos podem chegar ao destino corrompidos ou vídeos em tempo real podem perder frames que não deveriam faltar ao longo de uma transmissão. Além disso, a perda de dados não se limita aos efeitos de erros de bits ou perda de pacotes, mas também inclui todo e qualquer efeito causado pela degradação introduzida nos fluxos de dados. Isso pode ser observado na utilização de codecs que podem diminuir a qualidade do vídeo ao longo da transmissão.

Nesse sentido, sabendo que estas informações têm um impacto direto na satisfação do usuário, é necessária a escolha de um caminho que possua baixo atraso, baixa variação do atraso e pouca ou nenhuma perda de dados. A partir destas métricas, desenvolvemos funções de cálculo de custo dos caminhos baseadas no tipo de aplicação, de modo a auxiliar o algoritmo de roteamento na escolha do melhor caminho.

Funções de Custo

Atualmente, existem uma grande diversidade de aplicações que possuem diferentes características e requerem limites aceitáveis de delay, jitter e loss, para uma boa execução. Sendo assim, se faz necessário a criação de funções de cálculo de custo que sejam específicas quanto aos diferentes tipos de serviços que podem ser oferecidos aos usuários.

Baseado nessa necessidade e nos requisitos necessários à alimentação da GPMR para criação de novos GPs, desenvolvemos funções baseadas na Recomendação ITU-T G.1010 [39], que visam atender com especificidade alguns tipos de aplicação. Note que, todas essas operações são realizadas no IR, de modo que ele tenha ciência do comportamento global da rede quanto aos níveis de QoS. Abaixo, apresentamos as funções de cálculo de custo para alguns tipos de aplicações. Estas aplicações podem ser extendidas de acordo com a necessidade ou reconfiguradas pelo administrador da rede.

- Variáveis:

- D: Delay;
- J: Jitter;
- L: Loss.

- Classe 1: VoIP:

$$\text{Custo do caminho} = \sum_{i=1}^n (0,11 * L + 0,0825 * D + 0,33 * J) / n$$

n: número de roteadores ao longo do caminho.

- Classe 2: Web Browsing (HTML), Serviços de Transação Comercial (e-commerce) e e-mail:

Função de cálculo:

$$\text{Custo do caminho} = \sum_{i=1}^n L / n, \text{ se Delay igual ou próximo a Zero, ou}$$

$$\text{Custo do caminho} = \sum_{i=1}^n (2,5 * 10^{-4} * D) / n, \text{ se Loss igual a Zero.}$$

n: número de roteadores ao longo do caminho.

- Classe 3: Jogos Interativos e Telnet:

Função de cálculo:

$$\text{Custo do caminho} = \sum_{i=1}^n L / n, \text{ se Delay igual ou próximo a Zero, ou}$$

$$\text{Custo do caminho} = \sum_{i=1}^n (5 * 10^{-3} * D) / n, \text{ se Loss igual a Zero.}$$

n: número de roteadores ao longo do caminho.

- Classe 4: Alta Qualidade de Streaming de Áudio:

Função de cálculo:

$$\text{Custo do caminho} = \sum_{i=1}^n (0,45 * L + 1 * 10^{-5} * D + 0,45 * J) / n$$

n: número de roteadores ao longo do caminho.

- Classe 5: Transferência/Recuperação de Dados em massa (Bulk Data Transfer/Retrieval):

Função de cálculo:

$\text{Custo do caminho} = \sum_{i=1}^n L/n$, se Delay igual ou próximo a Zero, ou

$\text{Custo do caminho} = \sum_{i=1}^n (1,6 * 10^{-5} * D)/n$, se Loss igual a Zero.

n : número de roteadores ao longo do caminho.

5.2.3 Melhor Caminho: Fluxo Máximo Custo Mínimo

Para desenvolvimento de um algoritmo que determinasse as melhores rotas entre origem e destino, foi realizado um estudo aprofundado, e após essa revisão bibliográfica, constatamos que nosso trabalho se tratava de um problema de Otimização, mais especificamente de Fluxo em redes [20]. Com isso, criamos um algoritmo baseado no teorema clássico de Fluxo Máximo Custo Mínimo, baseado no método de *Ford-Fulkerson* e *Dijkstra* [13]. Para um melhor entendimento, algumas definições de redes no contexto de grafos serão apresentadas, além do teorema.

Definições e Teorema

1. **Grafo:** grafo dirigido ou grafo orientado é um par $(N; A)$, onde N é um conjunto finito e A é um conjunto de pares ordenados de elementos de N . Os elementos de N são chamados nós. Os elementos de A são chamados arcos.
2. **Rede:** é um grafo $(N; A)$ juntamente com uma ou mais funções que atribuem números aos arcos e/ou aos nós.
3. **Função capacidade:** uma função-capacidade em um grafo $(N; A)$ é qualquer função de A em $\mathbb{Z}_{\geq}$:

$$u : A \rightarrow \mathbb{Z}_{\geq}.$$

Para cada arco ij , um inteiro não-negativo u_{ij} é a capacidade do arco ij na rede $(N; A; u)$. Assim, dizemos que uma função x de A em $\mathbb{Z}_{\geq}$ respeita u se:

$$f \leq u;$$

ou seja, se x_{ij} para cada arco ij .¹

4. **Teorema do Fluxo Máximo Custo Mínimo:** Para quaisquer dois nós s e t em uma rede $(N; A; u)$ com função-capacidade u tem-se:

¹A letra 'u' é a inicial de 'upper bound'

$$\max_{x \leq u} \text{valor}(f) = \min_T f(\bar{T}, T),$$

onde o máximo é tomado sobre todos os fluxos que respeitam u e o mínimo é tomado sobre todos os custos.

logo, o valor fluxo não deve ser maior que o valor da função capacidade, para um custo mínimo.

Algoritmo de Escolha do Caminho

O Algoritmo 5.1 corresponde ao pseudocódigo de escolha do caminho de dados no qual a sessão será mapeada. O algoritmo tenta agrupar fluxos distintos de uma mesma sessão no mesmo caminho de modo a economizar recursos e proporcionar uma melhor utilização da largura de banda do caminho usado. No entanto, isso nem sempre é possível e em virtude disso, o algoritmo foi projetado para distribuir os fluxos de dados entre dois ou mais caminhos, para uma mesma sessão, caso seja necessário.

Algoritmo 5.1: Escolha do Caminho

Result: Caminhos Escolhidos

Declaração de variáveis:

BW $\leftarrow$ Bitrate solicitado;

cos $\leftarrow$ CoS da sessão;

ipDestino $\leftarrow$ Endereço IP do destino;

bwResidual $\leftarrow$ Largura de banda residual;

roteadorEgresso[] $\leftarrow$ Lista de roteadores egressos

arvoreMaxBWRes[] $\leftarrow$ Árvores com maior largura de banda residual;

arvoreMenorCusto[] $\leftarrow$ Árvores com menor custo;

begin

roteadorEgresso[] $\leftarrow$ Filtrar e selecionar os caminhos que possuem roteadores egressos que alcançam o host de destino. Para isso, executar algum algoritmo EGP;

arvoreMaxBWRes[] $\leftarrow$ Filtrar e selecionar as árvores que possuem maior BW Residual;

arvoreMenorCusto[] $\leftarrow$ Filtrar e selecionar as árvores que possuem menor custo, baseado nas métricas de roteamento definidas: delay, loss e jitter;

while (BW > 0) **do**

arvoreMaxBWRes $\leftarrow$ Selecionar de arvoreMaxBWRes[], a árvore que possui a maior BW Residual;

arvoreMenorCusto $\leftarrow$ Selecionar de arvoreMenorCusto[], a árvore que possui o menor custo;

caminhoEscolhido[] $\leftarrow$ caminho escolhido;

BW $\leftarrow$ bwResidual - BW;

end

Para se determinar uma rota que atenda às necessidades da sessão é preciso que sejam fornecidos seus requisitos (CoS e bit-rate desejado, delay, jitter e loss aceitáveis),

além do endereço IP do destino. Os requisitos da sessão permitem ao algoritmo verificar se o caminho tem capacidade de atender sua solicitação no que se diz respeito ao QoS, e o endereço IP do remetente, fará com que apenas os caminhos que possuem rotas em direção ao destino sejam os escolhidos. Para isso, um algoritmo de *Exterior Gateway Protocol* (EGP) é necessário, uma vez que ele determina possíveis caminhos da fonte ao nó de destino. Em nosso projeto, utilizamos o *Border Gateway Protocol* (BGP) [41] como protocolo para validar a solução proposta.

Depois de terem sido filtrados os caminhos que possuem essas características, o algoritmo avalia aqueles que tem maior largura de banda residual de modo a permitir a maior passagem de fluxos por uma mesma rota. Isso diminui a quantidade de sinalização causada na rede, além da taxa de utilização de CPU dos roteadores. Os caminhos candidatos são ordenados em ordem decrescente de largura de banda residual. Para estes caminhos uma função de cálculo do custo é executada, levando em consideração o tipo de aplicação da sessão e as métricas de roteamento. Tanto a largura de banda residual quanto o custo do caminho são calculados para cada rota em particular, isto é, para todos os caminhos origem-destino.

Por fim, o algoritmo seleciona a melhor rota, aquela que possui maior largura de banda residual e menor custo e, mapeia os fluxos da sessão no(s) caminho(s) de comunicação escolhido(s). A partir disso, os dados podem começar a ser transmitidos.

5.3 Detecção de Congestionamento ou Falha no Roteador para Apoio a Resiliência

Mecanismos de detecção de congestionamento ou detecção de falhas nos roteadores são utilizados com o intuito de fornecer suporte a rede para garantia de QoS. No que se refere a congestionamento da rede, pacotes de dados podem ser descartados a medida que se aumenta a disputa pelo mesmo enlace e o buffer dos roteadores acaba se esgotando, devido a formação de filas que ocorrem em suas portas. A medida que novos pacotes chegam ao roteador, o *buffer* dessas filas se torna cada vez mais cheio e, caso o roteador não consiga processar esses dados e encaminhá-los a interface de saída, o roteador começará a descartar estes pacotes. Em caso de queda de enlace ou falha do roteador da rede, é necessário que todas as sessões pertencentes àquele caminho sejam redirecionadas, de maneira a garantir a continuidade de transmissão dos fluxos.

A partir desse contexto, desenvolvemos para o QoS-RRC um módulo de detecção de congestionamento e detecção queda de enlace ou falha nos roteadores da rede para apoio a resiliência, denominado *Detection Congestion or Router Drop* (DCRDrop). Este mecanismo tenta evitar a perda de pacotes em situações de sobrecarga da rede ou queda

dos roteadores. Para tratar os casos de sobrecarga da rede, utilizamos uma abordagem proativa de modo que o mecanismo reaja antes do congestionamento da rede.

Inicialmente apresentaremos alguns conceitos relacionados aos mecanismos de controle de congestionamento e após isso detalharemos a solução proposta.

5.3.1 Conceituação

Os mecanismos de controle de congestionamento podem ser divididos em dois métodos, de acordo com a capacidade de assistência explícita à camada de transporte: Controle de congestionamento fim-a-fim e Controle de congestionamento assistido pela rede.

Controle de congestionamento fim-a-fim

Neste método, a camada de rede não fornece suporte explícito à camada de transporte. A detecção de congestionamento é diagnosticada por meio dos sistemas finais, a partir de observações do comportamento da rede através de perdas ou atrasos dos dados. O protocolo TCP adota este método, uma vez que, o protocolo IP (nível 3), não fornece informações de congestionamento da rede.

Controle de congestionamento assistido pela rede

No controle de congestionamento assistido pela rede, os roteadores fornecem informações ao remetente, alertando da eminência de congestionamento na rede. Neste método existem dois tipos de operação:

- **Realimentação direta:** neste caso, uma mensagem é composta pelo roteador da rede e enviada diretamente ao remetente, alertando sobre o congestionamento;
- **Realimentação indireta:** neste método, o roteador marca o cabeçalho do pacote para que o destinatário notifique o remetente sobre o congestionamento. No entanto, esta abordagem apresenta um problema, pois o remetente só receberá a notificação de congestionamento depois do tempo de ida e volta da mensagem, o que pode fazer com que diversos pacotes sejam descartados.

Dentre alguns dos mecanismos que utilizam o controle de congestionamento assistido pela rede, citamos:

- **DECbit[8]:** este mecanismo foi projetado para uso na arquitetura de rede digital (DNA - Digital Network Architecture) as quais são redes sem conexão que utilizam protocolos de transporte orientado a conexões. O DECbit utiliza agentes que

monitoram as cargas de dados nos roteadores e, caso seja detectado um congestionamento ele notifica explicitamente os nós finais. Sua notificação é implementada através de um bit de congestionamento nos pacotes que saem do roteador. Ao chegar ao host de destino, ele copia este bit na mensagem de ACK e envia de volta a origem, que ajusta a taxa de transmissão de dados (TCP) de modo a evitar o congestionamento;

- **RED (Random Early Detection)** [35]: Este mecanismo foi proposto por *Floyd e Jacobson* e é conhecido como detecção aleatória antecipada. Sua abordagem é semelhante a apresentada pelo mecanismo DECbit, no qual, cada roteador monitora o tamanho de sua fila. No entanto, sua diferença está no fato de que, ao invés de apenas fazer a marcação do pacote, o RED descarta as mensagens fazendo com que o protocolo TCP, reajuste sua janela deslizante diminuindo a taxa de transmissão dos fluxos. Além disso, outra diferença encontra-se no cálculo utilizado para descarte dos pacotes, que é baseado em funções probabilísticas.

De maneira a atender as necessidades do QoS-RRC, resolvemos utilizar o método de controle de congestionamento assistido pela rede, uma vez que os agentes estarão atuando neste nível da pilha de protocolos da rede. Para isso, realizaremos algumas adaptações do mecanismo RED de maneira que o QoS-RRC trate possíveis problemas relacionados a detecção de congestionamento na rede.

5.3.2 Módulo DCRDrop

Assim como os demais módulos do QoS-RRC, o DCRDrop trabalha de maneira desacoplada e independente, colaborando com as demais funcionalidades do sistema como um todo. Este módulo possui dois níveis de operações, para os quais é possível determinar congestionamento e falhas nos roteadores. Para isso, desenvolvemos dois componentes que tratam cada um destes problemas de maneira particular. Essa fragmentação em componentes permite com que sejam realizados tratamentos específicos para situações críticas diferentes:

- **DCRQueues (Detection of Congestion based on RED Queues)**: componente que detecta congestionamento baseado nas filas RED;
- **DDRouter (Down Detection Router)**: componente que detecta de quebra de enlaces ou falha nos roteadores da rede.

Detecção de congestionamento baseado nas filas RED - DCRQueues

De maneira a monitorar as filas dos roteadores da rede desenvolvemos um mecanismo de detecção baseado nas filas RED. Este mecanismo ao longo dos anos tem passado por

diversas propostas de adaptações [15][18][25][28][36], que sempre se preocuparam em melhorar o processo de medição das filas dos roteadores, otimizar a função probabilística de descarte de pacotes e garantir aos usuários finais um melhor *feedback* quanto ao QoS. Para esclarecer um pouco mais sobre o RED, daremos uma breve descrição de seu funcionamento e mostraremos as adaptações que foram realizadas de modo a acoplá-lo no componente DCRQueues.

Funcionamento do Mecanismo RED

A proposta original do mecanismo de gerenciamento de filas ativas RED visa trabalhar em conjunto com o protocolo TCP, da camada de transporte. A partir da detecção de congestionamento na rede, o RED solicita implicitamente ao protocolo TCP que reduza o tamanho da janela deslizante. Esta notificação implícita ocorre por meio do descarte de pacotes de dados.

Antes da execução do RED é necessário a configuração de alguns parâmetros que são utilizados pelo mecanismo. Estes parâmetros são configurados estaticamente e uma escolha eficaz pode levar o mecanismo a aumentar seu desempenho quanto a detecção de congestionamento. Os autores do mecanismo fazem sugestões de possíveis configurações destes parâmetros em [35]. Dentre alguns dos parâmetros existentes, dois são de extrema importância, o *Maxth* (*Maximum Threshold*) e o *Minth* (*Minimum Threshold*). Estes limites juntamente com a média móvel ponderada permitem ao RED determinar qual ação deverá ser executada pelo módulo de detecção de congestionamento. A Figura 5.2 apresenta a fila RED e estes patamares:

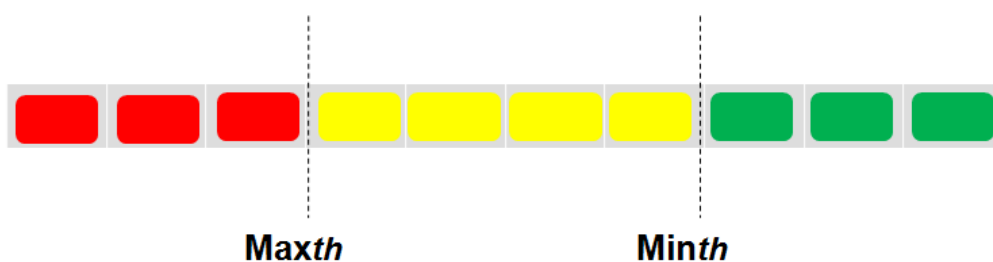


Figura 5.2: Fila RED

As cores representadas na fila RED indicam o tratamento ao qual o pacote será submetido após chegar ao roteador. De acordo com a especificação do RFC 2309 [35], os pacotes são marcados com as cores vermelho, amarelo ou verde e depois disso, o roteador optará por enfileirar ou descartar o pacote, segundo a cor contida em seu cabeçalho.

A medida com que os pacotes de dados chegam ao roteador, o mecanismo RED calcula o tamanho médio da fila (*avg*), usando uma função de média móvel ponderada. O cálculo do tamanho médio da fila deve ocorrer devido a natureza de rajada que a

Internet possui. Isso acontece pois as filas dos roteadores tornam-se cheias com grande rapidez e depois se esvaziam, não permitindo com que as reais condições da rede sejam monitoradas. A partir do momento que utilizamos uma função para capturarmos a média do comprimento da fila ao longo do tempo, detectamos o verdadeiro comportamento das filas dos roteadores.

Dado que o tamanho médio da fila foi calculado, a medida com que os novos pacotes de dados chegam ao roteador, o RED faz a seguinte verificação e aplica as regras:

- **Se $avg \leq Minth$**
Enfileira o pacote.
- **Se $Minth < avg < Maxth$**
O RED calcula a probabilidade P do pacote ser descartado. Se o pacote tiver a probabilidade P , o RED descarta. Caso contrário, enfileira.
- **Se $avg \geq Maxth$**
Descarta o pacote.

Caso o tamanho médio da fila seja inferior ao patamar mínimo, o pacote é colocado na fila e encaminhado para o enlace de saída. Se o tamanho médio da fila é superior ao patamar máximo, o pacote é automaticamente descartado. Por fim, se o tamanho médio da fila se encontra entre o patamar mínimo e máximo, uma função probabilística é utilizada. Esta função verifica a probabilidade de descarte de um pacote que chega a interface de rede. Caso essa probabilidade seja P , o pacote é descartado, senão, enfileirado e encaminhado a interface de saída.

A medida que os pacotes são descartados, o protocolo TCP reduz o tamanho da janela deslizante, reduzindo também a taxa de transmissão de dados. Por fim, temos uma significativa redução do congestionamento na rede.

Adaptação do Mecanismo RED

O mecanismo RED foi desenvolvido para trabalhar juntamente com o protocolo TCP. No entanto, como a maior parte das sessões requisitadas utilizam o protocolo UDP para transmissão de dados, dentre eles streams de vídeo, áudio e outros, adaptamos o RED de maneira que o protocolo de transporte não afetasse a eficácia de nosso mecanismo. Além disso, outra adaptação feita se remete ao descarte de pacotes. O QoS-RRC não descarta pacotes, pois possui controle de admissão e em casos de congestionamento da rede, realiza balanceamento de carga, o que permite aumentar o nível de satisfação dos usuários, além de aproveitar de maneira mais eficiente os recursos da rede.

Sendo assim, as modificações do RED foram:

- Independência do Protocolo TCP: ao invés do mecanismo RED descartar pacotes da rede, induzindo o protocolo TCP a reduzir a janela deslizante, o RED notifica localmente o agente QoS-RRC-C, alertando sobre um congestionamento;
- Não há descarte de pacotes: em vez de descartar pacotes, criamos uma função de notificação do IR que indica a eminência de congestionamento na rede. Depois disso, o QoS-RRC executa o algoritmo de balanceamento de carga, a fim de distribuir os fluxos uniformemente em toda a rede.

A detecção de congestionamento na rede ocorre por meio do componente DCR-Queues presente nos agentes QoS-RRC-C. Este componente pertence ao módulo DCR-Drop e fica em constante monitoramento das filas RED. Quando é detectada a eminência de um congestionamento, o componente DCRQueues notifica o módulo DCRDrop que compõe uma mensagem de notificação *RESERVE(C)* preenchendo o objeto *INFOSPEC* com o endereço IP do roteador degradado. Esta mensagem é enviada aos IR relacionados que analisam o conteúdo da mensagem e ativam o módulo de balanceamento de carga, Routing-LB, solicitando o re-roteamento das sessões degradadas. Este re-roteamento leva em consideração os requisitos iniciais de cada uma das sessões, bem como o estado atual da rede. Isto permitirá ao QoS-RRC escolher novos caminhos para as sessões degradadas.

A Figura 5.2 mostra o algoritmo de detecção de congestionamento da fila RED.

O funcionamento do algoritmo segue as bases do mecanismo RED. Após o pacote de dados chegar ao roteador, o algoritmo calcula o tamanho da fila RED e efetua uma decisão, dentre as seguintes:

- Se a média da fila for inferior ao parâmetro *Minth*, o pacote é enfileirado;
- Se a média estiver entre o *Minth* e o *Maxth*, o RED calcula a probabilidade do pacote ser descartado. Após isso, ele executa alguma das ações:
 - Se o pacote tiver probabilidade P de ser descartado, o agente QoS-RRC-C envia ao IR uma mensagem solicitando o re-roteamento com balanceamento de carga de alguns fluxos. A quantidade de fluxos a serem redirecionados será determinada por uma função presente no algoritmo de balanceamento de carga;
 - Se o pacote não tiver probabilidade P de ser descartado, o pacote é enfileirado.
- Se a média estiver acima do *Maxth*, o agente QoS-RRC-C envia ao IR uma mensagem alertando do congestionamento da rede e solicitando o re-roteamento com balanceamento de carga de alguns fluxos. A quantidade de fluxos a serem redirecionados será determinada por uma função presente no algoritmo de balanceamento de carga.

Algoritmo 5.2: Detecção de Congestionamento**Result:** Enfileiramento do pacote e Detecção de congestão**Declaração de variáveis:**avg $\leftarrow$ Média do tamanho da fila RED;Minth $\leftarrow$ Limite inferior da fila RED;Maxth $\leftarrow$ Limite superior da fila RED;**begin**

Calcular avg;

if (avg < Minth) **then**

| Enfileira o pacote

else **if** (avg > Minth) and (avg < Maxth) **then** | Calcula a probabilidade P do pacote ser descartado; **if** (*O pacote que chegou tem probabilidade P de ser descartado*) **then**

| Notificar agente QoS-RRC da eminência de um congestionamento;

| Enfileira o pacote;

else

| Enfileira o pacote;

else **if** (avg > Maxth) **then**

| Notificar agente QoS-RRC do congestionamento;

| Enfileira o pacote;

end**5.3.3 Detecção de Falha do Roteador - DDRouter**

O módulo DDRouter é utilizado para inferir falhas ocorridas nos roteadores da rede. Sendo detectada, o módulo Routing-LB será ativado para realizar re-roteamento de todos os fluxos que estão vinculados àquela rota, de modo a permitir com que as sessões ativas continuem em seu estado de execução.

Seu funcionamento ocorre da seguinte forma: a cada 30 segundos (por default, este parâmetro pode ser reajustado pelo administrador da rede), o QoS-RRC-E envia a partir do IR para dentro da rede a mensagem *RESERVE(U)* para coleta dos níveis de QoS. Estas mensagens são enviadas por um caminho unicast (uma para cada caminho) e em cada roteador visitado, os níveis atuais de QoS são adicionadas a mensagem (*QoS Available*).

Além disso, um contador de saltos é utilizado para detecção de falhas na rede. Ele funciona da seguinte maneira: a cada roteador visitado a variável *Number Hops* é adicionada em uma unidade. Ao chegar ao ER, a mensagem é reencaminhada ao IR e em cada roteador, o valor de *Number Hops* é decrescido em uma unidade. Ao chegar ao IR, o módulo DDRouter verificará se o valor de *Number Hops* é zero. Em caso afirmativo, temos a comprovação de que todos os roteadores estão ativos. Caso contrário, existe

algum roteador com falha na rede. Os roteadores do núcleo da rede repassam a mensagem *RESERVE(U)* e acionam um temporizador de 30 segundos. Se neste período a mensagem não retornar, ele cria uma nova mensagem *RESPONSE(L)* com o número do roteador do próximo salto. Ao chegar ao IR, o módulo DDRouter verificará que o contador não está zerado, constatando assim que um roteador parou de funcionar. Com isso, este módulo ativa o Routing-LB para realizar um re-direcionamento com balanceamento de carga de todas as sessões do caminho. As Figuras 5.3 e 5.4 ilustram um cenário de funcionamento normal e um com falha no roteador, respectivamente.

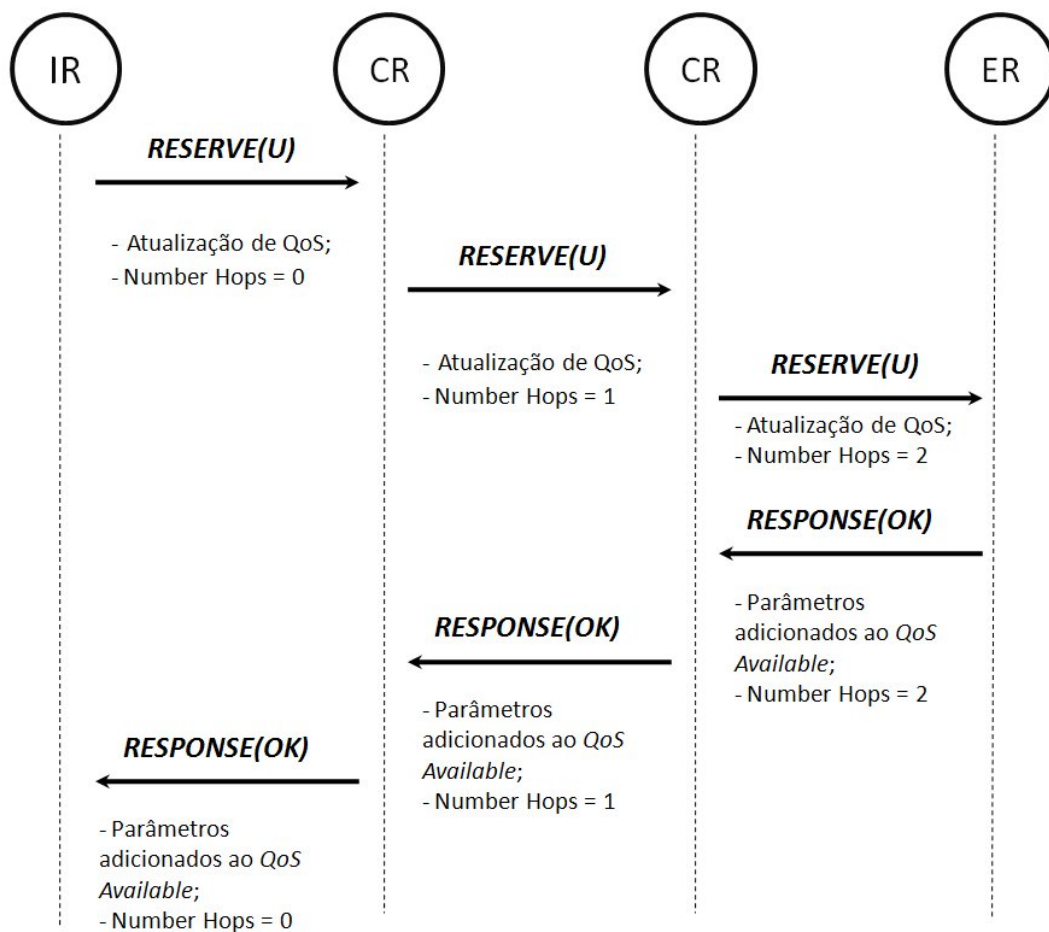


Figura 5.3: Cenário sem falha de roteador

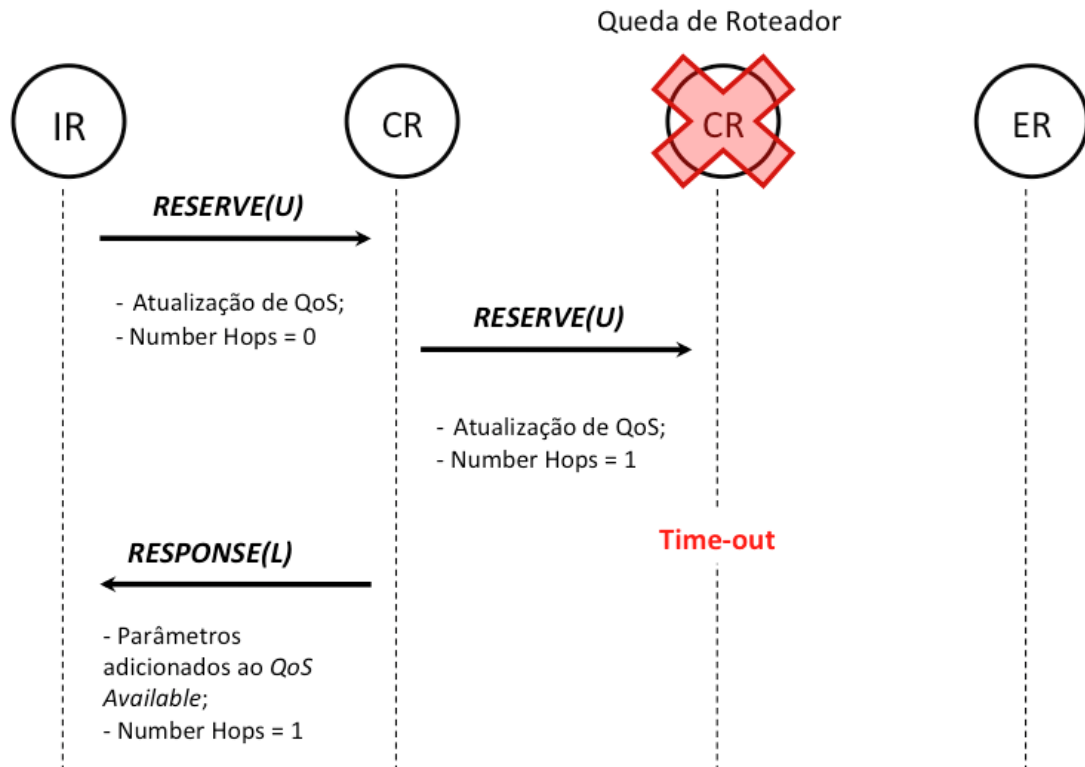


Figura 5.4: *Cenário com falha de roteador*

A Seção 5.4 apresenta o mecanismo de balanceamento de carga.

5.4 Balanceamento de Carga no QoS-RRC

O processo de distribuir carga entre os diversos recursos da rede é denominado balanceamento de carga. O desequilíbrio entre as cargas de trabalho dos computadores pode reduzir drasticamente o desempenho do sistema e assim, afetar seu desempenho. Com isso, é necessário que haja uma distribuição de tráfego entre os múltiplos caminhos físicos, com o intuito de atenuar os problemas de congestionamento da rede e de competição pelos mesmos recursos.

A fim de se levar vantagem da capacidade total do sistema, algoritmos de balanceamento de carga são altamente importantes às redes de computadores, de modo que, podemos melhorar o desempenho do sistema através da transferência apropriada das tarefas .

Nesse sentido, desenvolvemos um algoritmo de balanceamento de carga com capacidade para roteamento orientado a QoS para Internet do Futuro. Este algoritmo encontra-se no módulo Routing-LB e implementa balanceamento de carga durante períodos de congestionamento ocorridos no sistema como resultado de falhas na rede (uma vez que o controle de admissão impede que mais sessões que a capacidade atual dos

recursos sejam aceitas). O principal objetivo consiste em melhorar a alocação dos recursos da rede por meio da seleção de múltiplos caminhos de dados de maneira uniforme, evitando assim que, algumas rotas tornem-se ociosas e outras demasiadamente sobrecarregadas. Além disso, o desempenho pode ser maximizado, reduzindo o tempo de resposta das operações.

O balanceamento de carga ocorre:

- A cada novo pedido de sessão, o qual são solicitados requisitos mínimos necessários a execução da sessão. O QoS-RRC escolhe a melhor rota levando em consideração estes requisitos além dos custos de cada caminho;
- Em casos de degradação da rede, como por exemplo, congestionamento nas filas dos roteadores. Neste caso, o Routing-LB utiliza uma função que determina quantas sessões devem ser redirecionadas. Após isso, ele seleciona os fluxos mais recentes e faz um re-roteamento das sessões;
- Em casos de falhas de roteador. Neste caso, o Routing-LB faz um re-roteamento de todos os fluxos que passam por ele.

A Sub-seção 5.4.1 descreve o algoritmo de balanceamento de carga.

5.4.1 Descrição do Algoritmo de Balanceamento de carga

O algoritmo de balanceamento de carga tem como resultado um re-roteamento dos fluxos entre os diversos caminhos de dados contidos na rede. Para que isso ocorra, o módulo de detecção de congestionamento para apoio a resiliência (DCRDrop) fica em constante monitoramento nos roteadores do núcleo. Em caso de congestionamento ou falha nos roteadores, o DCRDrop envia uma mensagem de alerta ao IR que, a partir disso, aciona o algoritmo de balanceamento de carga. O Algoritmo 5.3, apresenta o pseudocódigo proposto.

A primeira ação do algoritmo de balanceamento de carga é verificar qual o problema ocorrido na rede: i) congestionamento das filas RED dos roteadores ou; ii) Falha no roteador.

Em caso de detecção de congestionamento, o algoritmo inicialmente irá calcular a taxa de utilização dos recursos de cada uma das árvores da rede e depois disso, a taxa de utilização global. Isso permitirá ao algoritmo verificar a quantidade de sessões que deverão ser re-direcionadas, uma vez que o balanceamento de carga é feito de modo a manter os fluxos das sessões distribuídos entre os caminhos da maneira mais equitativa possível. Um exemplo disso seria, a taxa de utilização dos recursos globais da rede estar em 50% e a taxa da árvore degradada estar em 80%. Neste caso, serão necessários o re-roteamento de 30% dos fluxos da árvore degradada, de modo que esta sobrecarga seja distribuída entre as demais árvores.

Algoritmo 5.3: Balanceamento de Carga**Result:** Re-roteamento de Fluxos**Declaração de variáveis:**

recursosArvore[] $\leftarrow$ Recursos utilizados por cada árvore, em porcentagem;
 recursosRede $\leftarrow$ Média global dos recursos utilizados por todas as árvores da rede, em porcentagem;

arvoreMaxTaxaUtilizacaoRecursos $\leftarrow$ Árvore com maior taxa de utilização de recursos;

selecionandoArvoresDegradadas[] $\leftarrow$ Árvores que passam pelo roteador degradado;

quantArvoresDegradadas $\leftarrow$ Quantidade de árvores degradadas;

begin

Seleção de árvores:

selecionandoArvoresDegradadas[] $\leftarrow$ Filtrar e selecionar as árvores que passam pelo roteador degradado;

quantArvoresDegradadas $\leftarrow$ quantidade de árvores degradadas;

if (*Houver degradação da rede*) **then**

Calcular a taxa de utilização dos recursos de cada árvore:

recursosArvore[] $\leftarrow \sum_{i=1}^n (FuncaoParaCalculodoCusto)/n$;

n: Número de roteadores da árvore.

Calcular a taxa de utilização dos recursos globais da rede:

recursosRede $\leftarrow \sum_{i=1}^n recursosArvore[]/n$;

n: Quantidade de árvores.

for ($i \leftarrow 0$ to $quantArvoresDegradadas - 1$) **do**

if (*Taxa de recursos utilizados pela árvore congestionada for > Média global dos recursos utilizados pela rede*) **then**

arvoreMaxTaxaUtilizacaoRecursos $\leftarrow$ Taxa de recursos utilizados pela árvore.

Fazer balanceamento de carga:

Qual a quantidade de fluxos que deverá ser roteada?

quantFluxosRoteados $\leftarrow$ arvoreMaxTaxaUtilizacaoRecursos - recursosRede;

Quais fluxos serão roteados?

Serão roteados os fluxos mais recentes.

Após isso, executar algoritmo para escolha do caminho: sua restrição está relacionada ao fato de que, ele deverá escolher somente as árvores que não passam pelo roteador degradado.

else

if (*Falha no Roteador*) **then**

Rotear todos os fluxos que passam pelo Roteador.

Após isso, executar algoritmo para escolha do caminho: sua restrição está relacionada ao fato de que, ele deverá escolher somente as árvores que não passam pelo roteador degradado.

end

Assim como no Algoritmo 5.1 de escolha do caminho, o redirecionamento das sessões leva em consideração o tipo de aplicação que está sendo utilizada. Outra ação levada em consideração pelo algoritmo está na escolha das sessões que deverão ser redirecionadas. O algoritmo de balanceamento de carga sempre escolhe as sessões mais recentes, uma vez que as sessões mais antigas tendem a estar mais próximas de sua finalização. Além disso, o re-roteamento pode causar algum tipo de atraso ou retardo no redirecionamento dos fluxos, uma vez que são necessárias reconfigurações das tabelas de roteamento locais do IR. Nesse sentido, não é conveniente redirecionar sessões que estão mais próximas de seu término e, portanto, possuem uma maior prioridade, já que foram inicializadas mais cedo.

Para casos de detecção de falhas no roteador, o algoritmo de balanceamento de carga realizará o redirecionamento de todas as sessões ativas daquele caminho de dados. Para cada uma destas sessões, o algoritmo de escolha do caminho é acionado, levando sempre em consideração os requisitos iniciais da sessão, o custo do caminho que sempre deve ser o mínimo e o QoS que deve ser garantida ao usuário. Ao final do re-roteamento, o IR registrará em sua tabela de estados todos os novos caminhos das sessões e continuará a transmissão dos dados.

Avaliação e Resultados

Os testes e avaliações realizados com o QoS-RRC foram desenvolvidos com o intuito de se comprovar a eficiência do mecanismo em comparação com outras soluções propostas. Para isso, estendemos as funcionalidades do simulador NS2 (Network Simulator) e fizemos experimentos baseados na rede (QoS) e nos usuários (QoE).

Este capítulo é composto da Seção 6.1 onde apresentamos o simulador NS2. Na Seção 6.2, descrevemos o framework Evalvid e na Seção 6.3, falamos a respeito de QoE. Por fim, na Seção 6.4, temos a avaliação do mecanismo QoS-RRC.

6.1 Simulador NS2

O *Network Simulator v2* (NS2) [30] é um simulador de eventos discretos, resultante de um projeto conhecido como VINT (Virtual InterNetwork Testbed). Dentre outros, compõem este projeto a DARPA, USC/ISI, Xerox PARC, LBNL e a Universidade de Berkeley. Este simulador oferece suporte à simulação de um grande número de tecnologias de rede com e sem fio, além de diferentes cenários baseados nos protocolos TCP e UDP, diversos escalonadores e políticas de fila e caracterização de tráfego com diversas distribuições estatísticas.

O NS2 utiliza duas linguagens: C++ para a estrutura básica (protocolos, agentes, etc) e *Object-oriented Tool Command Language* (OTCL) para uso como *frontend*. O OTCL é uma linguagem interpretada, desenvolvida pelo *Massachusetts Institute of Technology* (MIT). Nela serão efetivamente escritos os códigos das simulações. O motivo para se utilizar duas linguagens baseia-se em duas diferentes necessidades. De um lado existe a necessidade de uma linguagem mais robusta para a manipulação de bytes e pacotes, além de implementar algoritmos que executem um grande conjunto de dados. Nesse contexto, C++ que é uma linguagem compilada e de uso tradicional, mostrou-se a ferramenta mais eficaz. Por outro lado, é fato que, durante o processo de simulação são necessários vários ajustes com certa frequência. A cada ajuste de um enlace, atraso, ou nó, é necessária a realização de um novo teste. Com a utilização da linguagem OTCL, que é interpretada, há uma simplificação no processo iterativo de alteração e re-execução do modelo.

6.2 Evalvid - Uma Framework para Avaliação de Qualidade de Vídeo

Evalvid [23] é uma estrutura formada por um conjunto de ferramentas para avaliação da qualidade de fluxos de vídeo transmitidos em redes reais ou simuladas. O que torna a estrutura Evalvid atraente é a relativa simplicidade da interação entre os arquivos de trace usados na avaliação da qualidade de transmissão. Para isso são necessários apenas:

- Arquivos trace de entrada da rede: representam o fluxo de vídeo original;
- Arquivos trace de saída da rede: representam o fluxo de vídeo corrompido.

Desta forma, a rede pode ser encarada como uma "caixa preta" que provoca perdas, atrasos e um possível reordenamento dos pacotes. Assim como a "caixa" que representa a rede, as demais "caixas" que formam a estrutura da rede também podem ser substituídas. Essa característica é particularmente interessante para o caso da "caixa" que representa o codificador de vídeo.

Através da ferramenta *mp4Trace*, pertencente ao Evalvid, as sequências de vídeo codificadas e preparadas para transmissão foram usadas como entrada para a geração de um arquivo de texto do tipo trace, que contém todas as informações necessárias ao NS2. Com base nisso, o NS2 poderá criar um fluxo simulado que comporta exatamente como o fluxo codificado original, em termos de número de sequências e tipos de quadros, quantidade e tamanho dos segmentos de quadros (pacotes) enviados e, temporização da transmissão [38]. A Figura 6.1 mostra o diagrama de blocos da estrutura usada na simulação da rede.

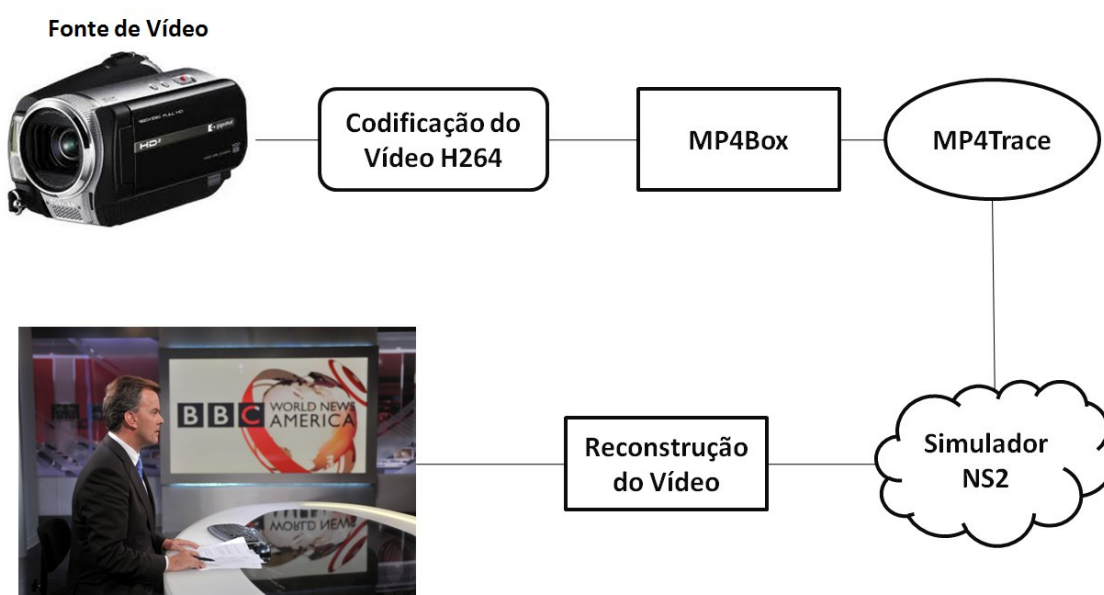


Figura 6.1: Diagrama de Blocos da Simulação

6.3 Qualidade de Experiência (*Quality of Experience* - QoE)

QoE é a percepção do usuário sobre a qualidade de um determinado serviço de rede ou aplicação. Como a experiência é algo subjetivo [24], pessoas com contextos de vida diferentes podem assistir ao mesmo filme ou lerem o mesmo livro e ao final, terem experiências e opiniões adversas sobre eles. Nesse sentido, são grandes as dificuldades para detectar o que realmente pode influenciar uma pessoa, pois aspectos que são bons para um, podem não ser para outro.

Como as métricas de QoS não conseguem capturar a experiência dos usuários finais com o serviço oferecido, pois dizem respeito apenas a capacidade da rede (largura de banda, bit-rate, delay, jitter, loss, dentre outros) e, tendo em vista a necessidade de se realizar avaliações que se atentassem também a experiência do usuário com o sistema, utilizamos métricas objetivas e subjetivas de QoE [5] para validação de nosso mecanismo. Para a avaliação subjetiva, utilizamos o *Mean Opinion Score* (MOS) que atribui uma nota a qualidade do vídeo ou áudio avaliado. Para avaliações objetivas, utilizamos algumas métricas que também avaliam a qualidade de vídeo e áudio, mas o fazem automaticamente. Estas métricas são *Peak Noise to Ratio* (PSNR), *Structural Similarity* (SSIM) e *Video Quality Metric* (VQM).

6.4 Avaliação do QoS-RRC

A avaliação da performance obtida pelo mecanismo QoS-RRC, juntamente com os módulos de roteamento e balanceamento de carga foi feita utilizando o simulador NS2 que foi estendido com as funcionalidades do QoS-RRC. O modelo de simulação utilizou uma topologia com 16 nós, largura de banda de 10Mbps e delay, jitter e loss flutuantes nos enlaces. Para a diferenciação das classes, utilizamos uma estrutura baseada no *DiffServ* [40] com escalonadores do tipo *Weighted Fair Queue* (WFQ), policiadores *Token Bucket* e filas do tipo *Random Early Detection* (RED). Temos um roteador de entrada de dados, chamado de *Ingress Router* (IR) e três roteadores de saída de dados, chamados *Egress Router* (ER). O mecanismo de escolha dos caminhos e balanceamento de carga fica situado no IR, e as requisições dos usuários são encaminhadas pelo ER até o IR. Utilizamos 250 sessões, cada uma contendo 4 fluxos, contemplando as classe de serviço *Expedited Forward* (EF), *Assured Forward 1* (AF1), *Assured Forward 2* (AF2) e *Best Effort* (BE), o que implica em 1000 fluxos. Os fluxos também variam e podem ser de 128Kbps e 256Kbps para cada uma das sessões, a uma taxa de envio constante de dados. Todos os pacotes tem um tamanho de 1000 bytes. Para que pudéssemos nos aproximar ainda mais do mundo real, inicializamos as sessões em intervalos de tempo

diferentes. O tempo total das simulações foi de aproximadamente 60 segundos. Por fim, para demonstrarmos o impacto gerado pelo QoS-RRC sobre a experiência do usuário, medições objetivas e subjetivas foram feitas utilizando-se métricas de QoE em um vídeo real. A ferramenta Evalvid foi utilizada para avaliar e validar nossos experimentos quanto a QoE.

Para realização de nossos experimentos, utilizamos o cenário apresentado na Figura 6.2.

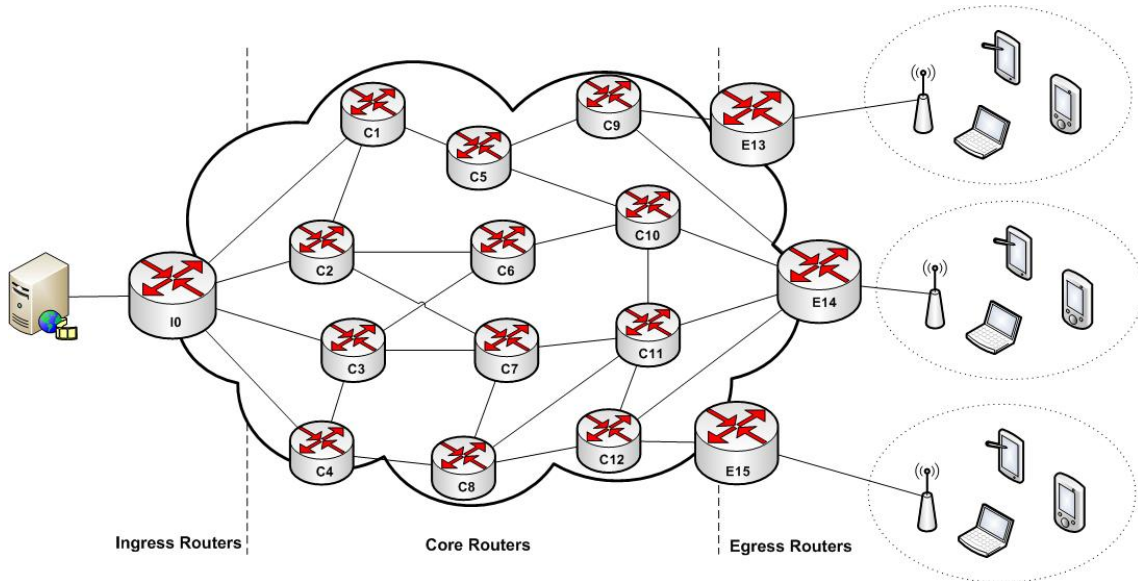


Figura 6.2: *Cenário de Simulação*

Para que pudéssemos verificar o desempenho do QoS-RRC, dividimos nossas avaliações em 2 experimentos. O primeiro dos experimentos é baseado na rede, realizando um comparativo entre o QoS-RRC e o MARA. Nele, avaliamos largura de banda, largura de banda residual e atraso gerado pelos mecanismos. O segundo experimento baseia-se na experiência do usuário, realizando um comparativo entre o QoS-RRC e uma configuração para Internet com IP padrão (IP Standard). Suas avaliações tratam a taxa de perda de dados, o número de sessões negadas e aspectos relacionados as medições objetivas e subjetivas de QoE com um vídeo real.

6.4.1 Experimentos baseados na Rede

O conjunto de experimentos baseados na rede analisou o comportamento do QoS-RRC e do MARA em termos de utilização de largura de banda (para a comprovação do comportamento da largura de banda residual global) e os atrasos provocados pelos caminhos ao longo do tempo de simulação (para o impacto do controle de qualidade). O objetivo principal de examinar tais medições, avaliar os benefícios do QoS-RRC com balanceamento de carga e roteamento orientado a QoS, o que não acontece com o MARA.

Em contraste com QoS-RRC, o mecanismo MARA preocupa-se principalmente em realizar provisionamento de recursos da rede dinamicamente, baseando-se no roteamento padrão da Internet, para a seleção dos caminhos candidatos.

Para avaliarmos a capacidade de re-roteamento e balanceamento de carga, simulamos uma quebra do enlace entre o roteador de entrada I0 e roteador do núcleo C1, no instante 41 segundos. A Figura 6.3 mostra os resultados da simulação de rendimento para cada um dos caminhos selecionados de maneira a se medir a taxa de utilização do enlace. É importante ressaltar que não há perda de dados em ambos os cenários, devido ao suporte de controle de admissão. Os caminhos especificados na Figura 6.2, passam pelos seguintes roteadores:

- **Caminho A:** I0 - C2 - C6 - C10 - E14.
- **Caminho B:** I0 - C4 - C8 - C11 - E14.
- **Caminho C:** I0 - C1 - C5 - C9 - E14.
- **Caminho D:** I0 - C3 - C7 - C11 - C12 - E14.

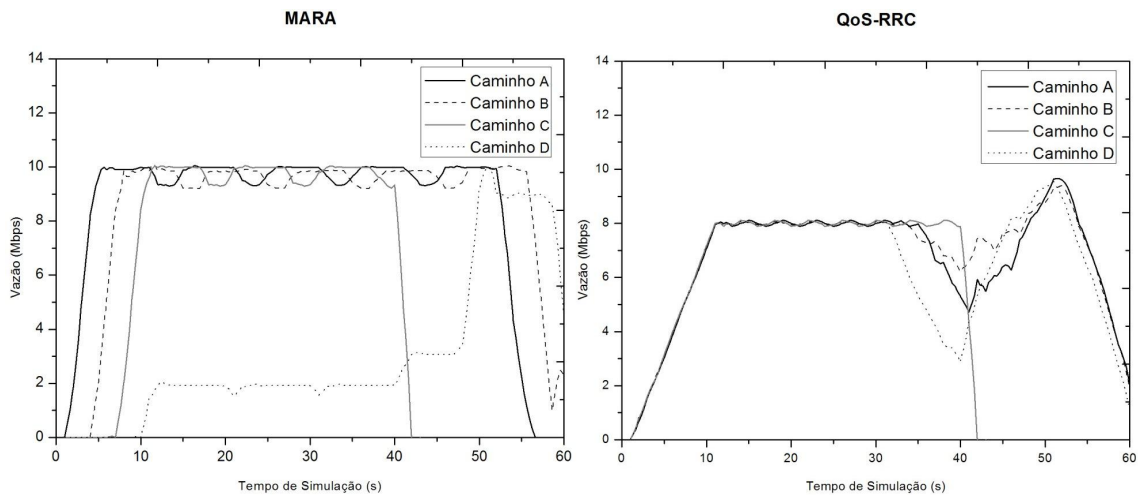


Figura 6.3: *Vazão: MARA (Sem balanceamento de carga) e QoS-RRC (Balanceamento de carga)*

Na Figura 6.3 percebemos na simulação a distribuição das sessões entre os caminhos de dados, os quais, ambos os mecanismos QoS-RRC e MARA, provisionam os mesmos caminhos (A, B, C e D, neste caso). Observando o gráfico do MARA, percebemos que o caminho B só é escolhido a partir do momento em que esgota-se a capacidade de largura de banda do enlace A. Isso acontece para os demais caminhos. Quando ocorre o evento de quebra do enlace, podemos observar que o MARA começa a redirecionar todas as sessões para o caminho D, não se preocupando se existe algum outro caminho com menor custo ou com maior largura de banda residual. O término aleatório de cada uma das sessões ao longo do tempo de simulação, fazem com que haja uma flutuação

nas taxas de dados de todas os caminhos, o que leva à uma instabilidade global das redes que utilizam o MARA.

Em contraste a isso, os resultados das simulações do QoS-RRC mostram um comportamento muito diferente para a largura de banda em todos os caminhos selecionados. Durante a simulação, todos os caminhos experimentaram uma utilização de recursos uniforme, com média de 8 Mbps por enlace, antes da quebra do enlace, indo atingir a capacidade máxima uma única vez, no instante 53 segundos, para compensar a perda do caminho C. Além disso, a influência do balanceamento de carga da carga com roteamento orientado a QoS permite ao QoS-RRC a alocação uniforme de largura de banda em todos os caminhos. Assim, a redução da largura de banda residual em todos os caminhos atesta a eficiência de QoS-RRC, em comparação com MARA, que pode ser executada por meio da análise dos gráficos de atraso, apresentados na Figura 6.4.

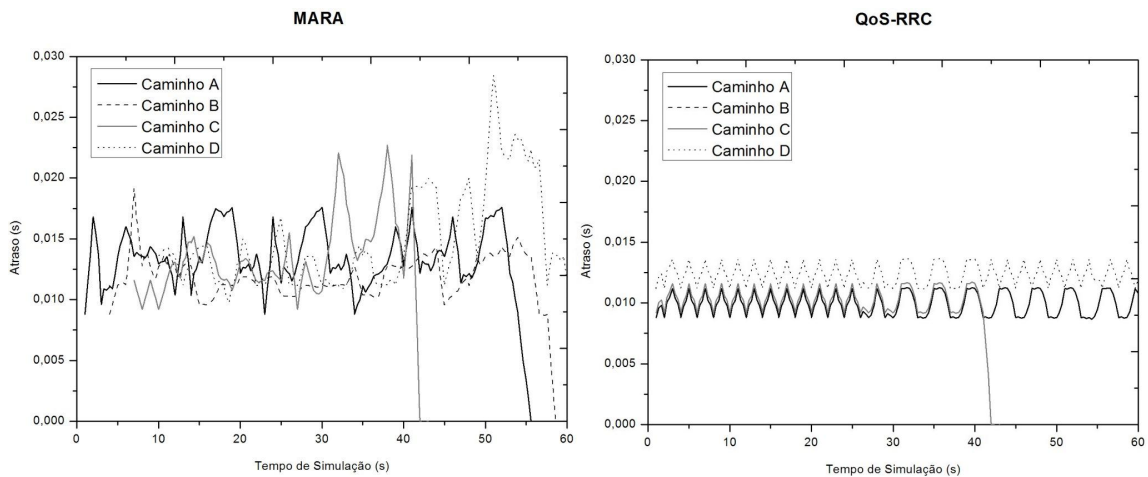


Figura 6.4: Atraso: MARA e QoS-RRC

Os resultados apresentados na Figura 6.4 mostram que o QoS-RRC QoS permitiu uma melhor experiência quanto aos níveis de atraso em todos os caminhos (mínimo de 9ms e máximo de 13ms, com desvios padrão de 0,0012; 0,0012; 0,0024 e; 0,0012 para os caminhos A, B, C e D, respectivamente), enquanto o MARA (atingindo 27ms de pico de atraso, com desvios padrão de 0,0049; 0,0041; 0,0072 e; 0,0065 para as árvores A, B, C e D, respectivamente). O balanceamento de carga com roteamento orientado a QoS permitiu ao QoS-RRC um comportamento uniforme quanto ao atraso de propagação, mesmo após a quebra do enlace. Além disso, também observamos variações de atraso pequenas (jitter) nas experiências com o QoS-RRC, o que é de suma importância para garantir a qualidade das sessões de multimídia. O jitter teve um aumento de 40%, o que ocorreu após a quebra do enlace, ficando na média de 0,9ms. Portanto, as conclusões que tiramos a respeito das experiências baseadas em rede é que a abordagem de provisionamento de recursos com balanceamento de carga com roteamento orientado a QoS do QoS-RRC é mais eficiente que o MARA (e conseqüentemente outras soluções).

6.4.2 Experimentos baseados no Usuário

Nos experimentos baseados no usuário, o QoS-RRC foi analisado em um cenário geral da Internet (IP Standard) configurado com um padrão de QoS por classe (*DiffServ*) e roteamento *Best-Effort* (OSPF). Assim, a configuração com IP Standard não aplicou controle de admissão e operações de reserva de recursos.

Em média, os resultados numéricos mostraram que o QoS-RRC não registrou perda de pacotes e bloqueou apenas 2,8% das sessões de vídeo. Na configuração IP Standard, 24% dos pacotes foram perdidos em todas as sessões de vídeo.

O *Peak Signal to Noise Ratio* (PSNR) é a métrica objetiva mais tradicional e compara quadro a quadro a qualidade do vídeo recebido pelo usuário, com o vídeo original. Além do fato de ser amplamente utilizado, devido sua baixa complexidade, o PSNR só fornece uma indicação da diferença entre o quadro recebido e o quadro original não levando em consideração o sistema de visão humano (*Human Visual System* - HVS), aspectos analisados durante nossa avaliação. Para assistir um vídeo considerado com boa qualidade, a partir da perspectiva do usuário, ele deve ter um PSNR médio de pelo menos 31dB. Isto baseia-se no mapeamento de valores PSNR para MOS, mostrado na Tabela 6.1 onde o MOS é considerada a mais popular métrica subjetiva.

Tabela 6.1: Conversão de PSNR para MOS

PSNR (dB)	MOS
> 37	5 (Excelente)
31 - 37	4 (Bom)
25 - 31	3 (Razoável)
20 - 25	2 (Pobre)
< 20	1 (Ruim)

Em nossas simulações, utilizamos o vídeo *News*, fornecido pelo próprio site do Evalvid [23]. Para o vídeo *News*, a média do PSNR para IP Standard foi de 19 dB (com desvio padrão de 4,6), conforme ilustrado na Figura 6.5. Além disso, o vídeo é considerado ruim de acordo com a experiência do usuário, tal como apresentado na Tabela 6.1. No entanto, quando o QoS-RRC é usado, as médias de PSNR passam para 45 dB (com desvio padrão de 1,9), mantendo assim o vídeo com excelente nível de qualidade, mesmo em períodos de congestionamento.

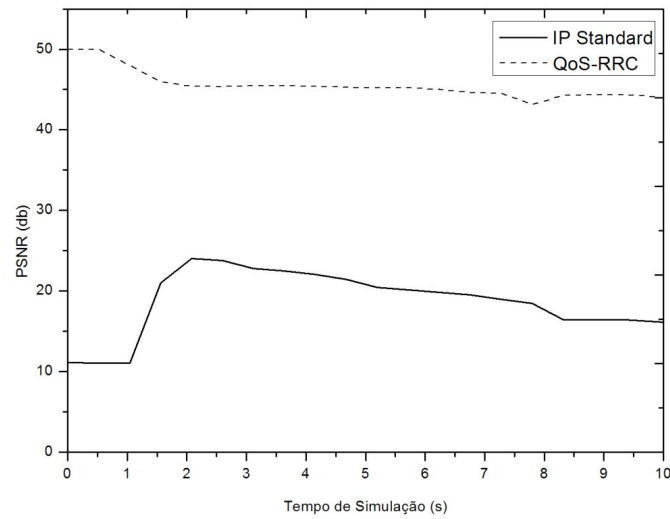


Figura 6.5: PSNR

A fim de se realizar uma comparação que leva em conta a estrutura dos objetos e fornecer uma melhor avaliação que o PSNR, a métrica *Structural Similarity* (SSIM) decompõe as imagens enviadas e recebidas, tendo em consideração 3 componentes HVS: a luminosidade, o contraste e as distorções estruturais.

O índice SSIM é um valor decimal entre 0 e 1, o qual 0 significa que não há nenhuma correlação com a imagem original, e 1 significa que é a mesma imagem. A Figura 6.6 mostra os valores SSIM para ambos os mecanismos, onde fica evidente que o QoS-RRC mantém o conteúdo de vídeo em tempo real com um excelente nível de qualidade durante todo experimento (SSIM de 0,99 em média), enquanto as médias da configuração IP Standard ficam em torno de 0,63 (com um desvio padrão de 0,12).

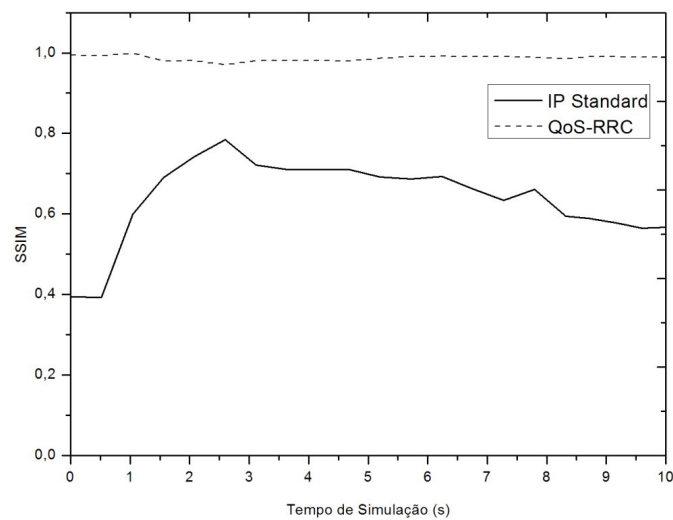


Figura 6.6: SSIM

A métrica *Video Quality Metric* (VQM) obtém como entrada o vídeo original e o vídeo processado, e verifica o nível de qualidade multimídia baseado na percepção

do olho humano e em aspectos de subjetividade, incluindo embassamento, ruído global, distorção de bloco e distorção de cor. Os resultados da avaliação do VQM variam entre valores de 0 a 5, onde 5 é a melhor pontuação possível. Conforme apresentado na Figura 6.7, em média, os valores VQM são de 2,57 (com desvio padrão de 0,73) e 4,69 (com desvio padrão de 0,96) para o IP Standard e QoS-RRC, respectivamente.

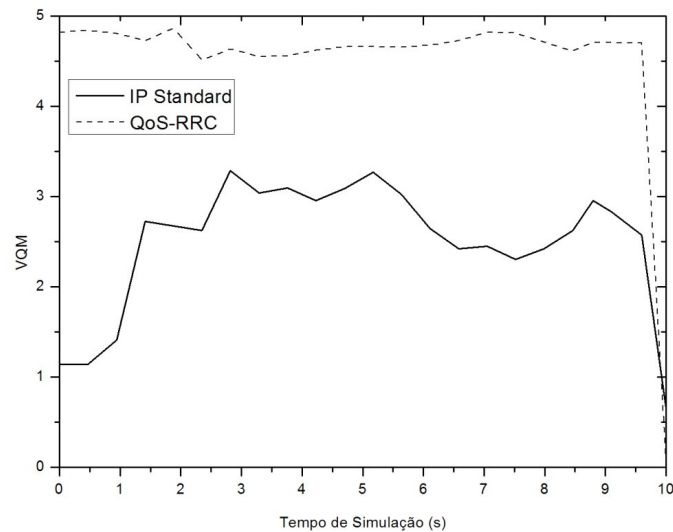


Figura 6.7: VQM

Para mostrar o impacto do QoS-RRC (em comparação com a configuração IP Standard) a partir do ponto de vista do usuário, quando o sistema está enfrentando 18% de congestionamento, alguns quadros de uma sessão de vídeo (selecionados aleatoriamente), foram capturados (Tabela 6.2). O benefício do QoS-RRC é visível nos frames de vídeo, em especial no bailarino.

Tabela 6.2: Alguns frames do vídeo News com IP Standard e com QoS-RRC

Mecanismo	Quadro número [07]	Quadro número [08]
<i>IP Standard</i>		
<i>QoS-RRC</i>		

As métricas objetivas e subjetivas de QoE demonstraram que o QoS-RRC manteve as sessões de vídeo em tempo real com um nível de qualidade excelente, mesmo em períodos de congestionamento.

Conclusão e Trabalhos Futuros

Esta dissertação apresenta o mecanismo QoS-RRC (QoS-Routing and Resource Control), uma proposta para dar suporte à arquitetura GP, um paradigma do projeto europeu 4WARD para Internet do Futuro.

Este capítulo organiza-se da seguinte maneira. Na Seção 7.1 apresentamos uma síntese de nosso trabalho bem como nossas contribuições. Na Seção 7.2, colocamos perspectivas para trabalhos futuros.

7.1 Síntese e Contribuições da Dissertação

O QoS-RRC emprega sobre-provisionamento de recursos e balanceamento de carga para roteamento orientado a QoS, permitindo assim que a arquitetura GP possa suportar eficientemente aplicações multimídia de tempo real e sessões multi-party. A abordagem de roteamento orientado a QoS do QoS-RRC, seleciona dinamicamente caminhos de dados indo ao encontro dos requisitos da sessão para garantir ao usuário uma melhor experiência, enquanto previne o sistema de degradações de operações por-fluxo e soluções baseadas em flooding (principalmente implementadas no estado da arte). O suporte a resiliência do QoS-RRC também melhorou a eficiência do mecanismo a partir de uma estratégia baseada nas filas RED para detecção de congestionamento na rede, sendo possível realizar re-roteamento de dados neste caso.

As simulações realizadas para avaliar o QoS-RRC permitiram demonstrar suas vantagens em comparação com os trabalhos relacionados relevantes (MARA para experimentos baseados em rede), bem como atual QoS da Internet e roteamento padrão (experiências baseados no usuário). Nos experimentos baseados em rede, o QoS-RRC otimizou drasticamente as experiências de atraso em todos os caminhos de dados selecionados (14ms de atraso de pico, enquanto MARA alcançou 27ms). Além disso, o QoS-RRC uniformemente equilibra a carga de trabalho nos caminhos roteados, permitindo uma maior quantidade de largura de banda residual. Nos experimentos baseados no usuário, o QoS-RRC não apresentou nenhuma perda de pacotes, enquanto na configuração do IP Standard 24% dos pacotes foram perdidos. Além disso, no que se trata as experiências do usuário,

ficou evidente que o conteúdo do vídeo de tempo real apresentado pelo QoS-RRC possui um excelente nível de qualidade, comprovado pelas medições de QoE, onde média do SSIM foi de 0,99, VQM de 4,69, e PSNR de 45 dB. Em contraste, com a configuração de IP Standard, obtivemos, SSIM com média de 0,63, VQM de 2,67 e PSNR de 19 dB.

Sendo assim, concluímos que o mecanismo proposto QoS-RRC apresenta características que vão ao encontro das reais necessidades das aplicações multimídia de tempo real, fornecendo suporte a sessões multi-party e integração a arquitetura CG, do Projeto 4WARD para Internet do Futuro.

7.2 Trabalhos Futuros

Os resultados deste trabalho apontaram como trabalhos futuros a avaliação do QoS-RRC através de prototipagem, a fim de chegar a conclusões ainda mais precisas. Além de conceber o QoS-RRC de acordo o paradigma GP do 4WARD, acreditamos que seus benefícios podem resolver problemas de desempenho atuais de redes em malha sem fio. Sendo assim, nossos trabalho futuros também incluirão a extensão do QoS-RRC para orquestrar sistemas de malha sem fio, tais como IEEE 802.11s.

Referências Bibliográficas

- [1] A. NETO, E. CERQUEIRA, A. R. E. E. A. **Scalable resource provisioning for multi-user communications in next generation networks.** *IEEE Globecom 2008 Next Generation Networks, Protocols e Services Symposium*, 2008.
- [2] A. NETO, E. CERQUEIRA, A. R. E. E. A. **Method e apparatus for multicast tree allocation.** *European Patent Officer EP2031796 (A1)*, 2009.
- [3] A. NETO, E. CERQUEIRA, A. R. E. P. M. **Method e apparatus for qos resource reservation e configuration of multicast network resources.** *European Patent Officer EP2031796 (A1)*, 2009.
- [4] A. NETO, E. CERQUEIRA, M. C. E. E. A. **Scalable multimedia group communications throught the over-provisioning of network resources.** *11th IFIP/IEEE International Conference on Management of Converged Multimedia Networks e Services*, 2008.
- [5] ANDREW PERKIS, S. M. E. O. I. H. **A model for measuring quality of experience.** *Center for Quantifiable Quality of Service in Communication Systems*, 2006.
- [6] C. SHUE, W. H. E. K. D. **Osi connectionless transport services on top of udp version: 1.** RFC 1240, Internet Engineering Task Force, Junho 1991.
- [7] E ANTONIO LIOTTA, F. A. **Qoe-aware qos management.** *Proceedings of MoMM*, Novembro 2008.
- [8] E BRUCE DAVIE, L. P. **Redes de Computadores.** Elsevier, 2004.
- [9] E. CRAWLEY, R. N. E. E. A. **A framework for qos-based routing in the internet.** RFC 2386, Internet Engineering Task Force, Agosto 1998.
- [10] E DIEGO PERINO, L. M. **Evaluating the performance of multi-path routing e congestion control in presence of network resource management.** *Ultra Modern Telecommunications e Workshops, 2009. ICUMT 09. International Conference on*, 2009.

- [11] E H. SCHULZRINNE, P. P. **Yessir: A simple reservation mechanism for the internet.** In *Proc. of the Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Julho 1998.
- [12] E JON CROWCROFT, Z. W. **Bandwidth-delay based routing algorithms.** *Global Telecommunications Conference, 1995. GLOBECOM '95., IEEE*, 3(5329097):2129–2133, Novembro 1995.
- [13] E KENNETH STEIGLITZ, C. H. P. **Combinatorial Optimization: Algorithms e Complexity.** Número 0-486-40258-4. General Publishing Company, Ltd. 30 Lesmill Road, Don Milss, Toronto, Ontario, Janeiro 1998.
- [14] E MARCO SCHNEIDER, M. G. G. **Maximizable routing metrics.** *IEEE/ACM Transactions on Networking (TON)*, 11(04), Agosto 2003.
- [15] E MARK CLAYPOOL, J. C. **Analysis of active queue management.** In *Proceedings of the 2nd IEEE International Symposium on Network Computing e Applications (NCA)*, Abril 2003.
- [16] E OTTO CARLOS, L. H. **Roteamento multicast na internet.** *Universidade Federal do Rio de Janeiro.*
- [17] E PETER E. LININGTON, M. R. **Distributed quality of service multicast routing with multiple metrics for receiver initiated joins.** p. 180, Washington, DC, USA, 2000. IEEE Computer Society.
- [18] E SHAOHUA YU, J. H. **An innovation three color marker in diffserv networks.** *Proceedings of the First International Multi-Symposiums on Computer e Computational Sciences (IMSCCS'06)*, IEEE Computer Society:192–198, Junho 2006.
- [19] E X. FU, J. M. **Analysis of existing quality-of-service signaling protocols.** RFC 4094, Internet Engineering Task Force, Maio 2005.
- [20] FEOFILOFF, P. **Fluxo em Redes.** Departamento de Ciência da Computação e Instituto de Matemática e Estatística, São Paulo, 2004.
- [21] G. ASH, A. BADER, C. K. E. D. O. **Qos nslp qspec template.** IETF Internet Draft 1240, Internet Engineering Task Force, Fevereiro 2008.
- [22] INC, A. C. **Ospf version 2.** RFC 2328, Internet Engineering Task Force, Abril 1998.
- [23] J. KLAUE, B. R. E. A. W. **Evalvid - a video quality evaluation tool-set**, 2008.
- [24] JAIN, R. **Quality of experience.** *IEEE Multimedia - IEEE Computer Society*, 11:95–96, Março 2004.

- [25] JAY KUMAR SUNDARARAJAN, FANG ZHAO, P. Y.-M. E. M. M. **A modification to red aqm for cioq switches**. *Laboratory for information e decision Systems, Massachusetts Institute of Technology, Cambridge*, 3:1708–1712, Novembro 2004.
- [26] KANNAN, G. **Adaptive selective flooding qos routing**. *Degree of Master Science in Computer Science*, 2000.
- [27] MARCELO D, N. F. L. H. E. E. A. **Internet do Futuro: Um Novo Horizonte**. Grupo de Teleinformática e Automação, Universidade Federal do Rio de Janeiro, 2009.
- [28] MIKKEL CHRISTIANSEN, KEVIN JEFFAY, D. O. E. F. D. S. **Tuning red for web traffic**. *IEEE/ACM Transactions on Networking*, 9, Junho 2001.
- [29] NORBERT NIEBERT, STEPHAN BAUCKE, K. W. E. H. W. **The way 4ward to the creation of a future internet**. *Personal, Indoor e Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, p. 1–5, Setembro 2008.
- [30] PAGE, T. N.-. H. **The network simulator - ns2**, Março 2007.
- [31] PETER BROOKS, BJORN HESTNES, S. H. E. C. A. **Communicating quality of experience data for the development of multimedia services**, 2003.
- [32] R. BRADES, L. Z. E. E. A. **Resource reservation protocol rsvp version 1 functional specification**. RFC 2205, Internet Engineering Task Force, Setembro 1997.
- [33] RANDRIAMASY, S. E. A. **Mechanisms for generic paths, the fp7 4ward project, deliverable d5.2**. 2009.
- [34] S. FIGUEIREDO, J. LOURENÇO, R. A. E. A. N. **Taxonomy for gp-aware mobility**. *Proc. of the First International ICST Conference on Mobile Networks e Management*, 2009.
- [35] S. FLOYD, V. J. E. E. A. **Recommendations on queue management e congestion avoidance in the internet**. RFC 2309, Internet Engineering Task Force, Abril 1998.
- [36] SALLY FLOYD, R. G. E. S. S. **Adaptive red: An algorithm for increasing the robustness of reds active queue management**. *AT&T Center for Internet Research at ICSI*, Agosto 2001.
- [37] SUJATHA P. TERDAL, V. M. E. A. D. **Multiple metrics based load balancing routing protocol for mobile ad hoc networks**. *Internet, 2009. AH-ICI 2009. First Asian Himalayas International Conference on*, p. 1–5, Novembro 2009.

- [38] TORRES, D. P. **Impacto do Gerenciamento Ativo de Filas em Redes IP Transportando Fluxos de Vídeo H.264/MPEG-4 AVC**. Master's thesis, Universidade Federal do Paraná, Curitiba, 2009.
- [39] UNION, I. T. **ITU-T Recommendation G.1010: End-user multimedia QoS categories (Quality of service e performance)**. ITU, Novembro 2001.
- [40] WANG, Z. **Internet QoS - Architectures e Mechanisms for Quality of Service**. Morgan kaufmann Publishers, USA, 2001.
- [41] Y. REKHTER, T. L. E. S. H. **A border gateway protocol 4 (bgp-4)**. RFC 4271, Internet Engineering Task Force, Janeiro 2006.