



**UNIFACS**

UNIVERSIDADE SALVADOR

LAUREATE INTERNATIONAL UNIVERSITIES®

**UNIFACS UNIVERSIDADE SALVADOR  
MESTRADO ACADÊMICO EM SISTEMAS E COMPUTAÇÃO**

**ANDRÉ LUIZ COSTA DE OLIVEIRA**

**MECANISMO DE NOTIFICAÇÃO BASEADO EM CONTEXTO PARA  
ENCAMINHAMENTO ADAPTATIVO**

Salvador  
2015

**ANDRÉ LUIZ COSTA DE OLIVEIRA**

**MECANISMO DE NOTIFICAÇÃO BASEADO EM CONTEXTO PARA  
ENCAMINHAMENTO ADAPTATIVO**

Dissertação apresentada ao Mestrado em Sistemas e  
Computação da UNIFACS Universidade Salvador,  
Laureate International Universities como requisito parcial  
para a obtenção do título de Mestre

Orientador: Prof. Dr. Paulo Nazareno Maia Sampaio.

Salvador  
2015

## FICHA CATALOGRÁFICA

(Elaborada pelo Sistema de Bibliotecas da UNIFACS Universidade Salvador,  
Laureate International Universities)

Oliveira, André Luiz Costa de

Mecanismo de notificação baseado em contexto para encaminhamento adaptativo/ André Luiz Costa de Oliveira. - 2015.

156 f. : il.

Dissertação Programa de Pós-Graduação em Sistemas e Computação de UNIFACS Universidade Salvador, Laureate International Universities como requisito parcial à obtenção do título de Mestre.

Orientador: Prof. Dr. Paulo Nazareno Maia Sampaio.

1. Redes de computadores. 2. Roteamento adaptativo. I. Sampaio, Paulo Nazareno Maia, orient. II. Título.

CDD:004. 6

ANDRÉ LUIZ COSTA DE OLIVEIRA

MECANISMO DE NOTIFICAÇÃO BASEADO EM CONTEXTO PARA  
ENCAMINHAMENTO ADAPTATIVO

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre, na UNIFACS Universidade Salvador, Laureate International Universities, pela seguinte banca examinadora:

Paulo Nazareno Maia Sampaio – Orientador \_\_\_\_\_  
Doutor em Informatique et Télécommunications Université Paul Sabatier - Toulouse  
III/LAAS-CNRS  
UNIFACS Universidade Salvador, Laureate International Universities

Artur Henrique Kronbauer \_\_\_\_\_

Doutor em Ciência da Computação, na área de Interação Humano-Computador pela  
Universidade Federal da Bahia (UFBA)  
UNIFACS Universidade Salvador, Laureate International Universities

Romildo Martins Bezerra \_\_\_\_\_  
Doutor em Ciência da Computação - Ufba - Unifacs  
Instituto Federal da Bahia – IFBA

Salvador, 23 de outubro de 2015.

## **AGRADECIMENTOS**

Agradeço a minha Família e aos amigos que me apoiaram durante este período que priorizei a realização deste objetivo, por vezes em detrimento da companhia deles.

Agradeço especialmente a meu Orientador, Dr. Paulo Sampaio, que por seus exemplos de disponibilidade, dedicação à pesquisa e melhoria continuada das ideias, do projeto e principalmente da formação dos pesquisadores sob sua tutela, foi peça-chave para o desenvolvimento deste trabalho.

Agradeço a meus colegas pesquisadores, Fred, Sergio e Jota, que mesmo cada um estando dedicado ao seu foco de estudo, pudemos todos dar contribuições significativas para o resultado global, não só nos pontos de interseção das pesquisas, mas principalmente no apoio nas horas difíceis, dividindo as incertezas e apoiando nos sucessos, fortalecendo muito o grupo como um todo.

Sem detrimento aos demais, cabe um agradecimento especial a Fred, parceiro constante de longas discussões produtivas de pontos de vista e argumentações que considero terem sido fundamentais para a evolução do trabalho de cada um de nós.

## RESUMO

Nos últimos anos vêm se intensificando o uso de redes de computadores junto ao crescimento na adoção de novas aplicações em tempo real e o aumento na concorrência de tráfegos distintos competindo por recursos de rede. Neste ambiente heterogêneo, as técnicas tradicionais de gestão de rede baseadas em parâmetros qualidade de serviço e engenharia de tráfego nem sempre conseguem atender aos reais requisitos das aplicações. Em paralelo, novos parâmetros vêm sendo pesquisados para auxiliar a avaliação de sistemas computacionais, tais como a qualidade dos dispositivos envolvidos, a qualidade de experiência percebida pelo usuário que interage com o sistema, e o próprio contexto que envolve as entidades relacionadas. Pesquisas que envolvem novos algoritmos de encaminhamento baseados em contexto vêm sendo realizadas, no entanto, focando principalmente nas redes sem fio e nas necessidades de mobilidade e redes oportunistas. Desta forma, este trabalho propõe uma abordagem que considera os parâmetros tradicionais de qualidade de serviço, considerando também os novos parâmetros recentes em áreas de pesquisa como a qualidade de dispositivo, a qualidade de experiência e o contexto, extrapolando seu uso para qualquer tipo de rede. É discutido e definido nesta dissertação um arcabouço para encaminhamento adaptativo em redes de computadores baseado em informações contextuais e, em particular, é proposto e modelado um mecanismo para a coleta e validação das informações de contexto antes de serem aplicadas para a tomada de decisão quanto ao encaminhamento na rede.

**Palavras-chave:** Redes Definidas por *Software*. Roteamento Adaptativo. Contexto. Qualidade de Contexto. Regras de Contexto. Redes Sensíveis ao Contexto.

## ABSTRACT

Along the last decades the utilization of computer networks has increased considerably, together with the adoption of emerging real-time applications and the raise in traffic concurrency competing for network resources. Within this heterogeneous environment, the traditional network management techniques based on quality of service metrics and network engineering are not able, in general, to meet the requirements of user' s applications. At the same time, research is being under way to identify new metrics in order to support the evaluation of computational systems, such as the quality of devices applied, the quality of experience noticed by the user that interacts with the system, and the context itself related to all the related entities. Research related to new routing algorithms based on context have been carried out. However, these research outcomes focus mainly on wireless networks, mobility issues and opportunistic networks. Therefore, this work proposes a generic approach for the collecting and validation of context information considering the traditional quality of service metrics as well as quality of device and quality of experience metrics, before this information being applied to support routing decisions.

**Keywords:** Software Defined Networks. Adaptive Routing. Context. Quality of Context. Context Rules. Context-aware networks.

## LISTAS DE FIGURAS

Figura 1 - Context-aware Adaptive Routing Framework (CAARF).....	18
Figura 2 - Foco do trabalho no CAARF.....	20
Figura 3 - The RFC 7426 SDN Architectural Model.....	26
Figura 4 - Exemplo de simulação NS2: OTcl.....	29
Figura 5 - Exemplo de simulação NS2: Traço.....	30
Figura 6 - Exemplo de simulação NS2: NAM.....	30
Figura 7 - Categorias fundamentais para Informação de Contexto.....	34
Figura 8 - <i>Context Controller</i> .....	41
Figura 9 - <i>Context Model</i> .....	45
Figura 10 - Funcionamento do CAARF.....	46
Figura 11 - Fluxo da simulação.....	66
Figura 12 - Divisão do processamento entre as abordagens.....	67
Figura 13 - Processamento da abordagem de contexto.....	67
Figura 14 - Processamento da abordagem de encaminhamento.....	67
Figura 15 - XML - Estrutura base das marcações.....	69
Figura 16 - XML - Exemplos de marcações de individualidade.....	70
Figura 17 - XML - Exemplos de marcações de tempo.....	70
Figura 18 - XML - Exemplos de marcações de localização.....	70
Figura 19 - XML - Exemplos de marcações de atividade.....	71
Figura 20 - XML - Exemplos de marcações de relações.....	71
Figura 21 - XML - Exemplos de marcações de QoE.....	72
Figura 22 - XML - Exemplos de marcações de QoD.....	72
Figura 23 - XML - Exemplos de marcações de QoS.....	73
Figura 24 - Relações da base de contexto.....	77
Figura 25 - Exemplo de código AWK para calculo da taxa de transferência.....	80
Figura 26 - Topologia base dos cenários.....	83
Figura 27 - Cadastro dos encaminhadores.....	86
Figura 28 - Cadastro dos parâmetros de um encaminhador.....	86
Figura 29 - Cadastramento dos links entre os encaminhadores.....	87
Figura 30 - Exemplo do cadastro dos parâmetros de cada enlace.....	87
Figura 31 - Exemplo da associação entre os enlaces e os encaminhadores.....	88
Figura 32 - Cadastramento de um caminho.....	88
Figura 33 - Associação dos enlaces de um caminho ao ID do caminho.....	88
Figura 34 - Cadastro de um encaminhamento.....	89



Figura 35 - Tempos do Cenário 1 .....	91
Figura 36 - Cenário 1 - Execução 1: NAM no momento 2,44 .....	91
Figura 37 - Cenário 1 - Execução 1: Encaminhamento melhor esforço.....	92
Figura 38 - Recorte do XML enviado por N9 no tempo 1,0 .....	94
Figura 39 - Consulta se um dispositivo já existe antes da inclusão.....	94
Figura 40 - Inclusão de novo dispositivo e suas características .....	95
Figura 41 - Cadastro da individualidade .....	95
Figura 42 - Cadastro da Relação.....	95
Figura 43 - Cadastro da Atividade.....	96
Figura 44 - Verificação de QoC da Identidade.....	96
Figura 45 - Atualização do estado de QoC aprovado para parâmetros da Individualidade .....	97
Figura 46 - Validação de QoC da Aplicação.....	97
Figura 47 - Atualização do estado de QoC aprovado para parâmetros da Atividade.....	98
Figura 48 - Validação de QoC do Dispositivo .....	98
Figura 49 - Atualização do estado de QoC aprovado para parâmetros do dispositivo...	99
Figura 50 - Definindo uma relação como pertencente ao Contexto .....	99
Figura 51 - Consulta do Contexto atual.....	99
Figura 52 - Exibição do novo Contexto com uma relação no Cenário 1.....	100
Figura 54 - Consulta de melhor caminho para o VoIP do Cenário 1 .....	100
Figura 55 - Novo contexto do Cenário 1 com duas relações.....	101
Figura 56 - Consulta de melhor caminho para o VoD do Cenário 1 .....	102
Figura 57 - Cenário 1 - Execução 2 : Configuração dos caminhos MPLS.....	102
Figura 58 - Cenário 1 - Execução 2: NAM momento 1,39 .....	103
Figura 59 - Cenário 1 - Execução 2: NAM momento 2,08 .....	103
Figura 60 - Cenário 1 - Execução 2: NAM momento 4,4 .....	103
Figura 61 - Cenário 1 - Execução 2: Encaminhamento baseado em contexto .....	104
Figura 62 - Tempos do Cenário 2.....	105
Figura 63 - Cenário 2 - Execução 1: Nam no momento 2,4.....	106
Figura 64 - Cenário 2 - Execução 1: Encaminhamento melhor esforço.....	106
Figura 65 - Recorte do XML enviado por N12 no tempo 1,5 .....	109
Figura 66 - Recorte do XML enviado por N14 no tempo 1,5 .....	109
Figura 67 - Novo contexto do Cenário 1 com duas relações.....	110
Figura 69 - Cenário 2 - Execução 2: NAM momento 3,4 .....	110
Figura 70 - Cenário 2 - Execução 2: Encaminhamento baseado em contexto .....	111

Figura 71 - Tempos do Cenário 3.....	112
Figura 72 - Cenário 3 - Execução 1: Nam no momento 2,4.....	113
Figura 73 - Cenário 3 - Execução 1: Nam no momento 3,7.....	113
Figura 74 - Cenário 3 - Execução 1: Nam no momento 4,3.....	113
Figura 75 - Cenário 3 - Execução 1: Nam no momento 5,9.....	114
Figura 76 - Cenário 3 - Execução 1: Nam no momento 6,1.....	114
Figura 77 - Cenário 3 - Execução 1: Encaminhamento melhor esforço.....	115
Figura 78 - Recorte do XML enviado por N16 no tempo 4,0.....	118
Figura 79 - Recorte do XML enviado por N11 no tempo 6,2.....	119
Figura 80 - Cenário 3: Configuração de caminhos MPLS no NS2.....	120
Figura 81 - Cenário 3 - Execução 2: Nam no momento 3,4.....	120
Figura 82 - Cenário 3 - Execução 2: Nam no momento 4,3.....	121
Figura 83 - Cenário 3 - Execução 2: Nam no momento 6,3.....	121
Figura 84 - Cenário 3 - Execução 2: Nam no momento 7,3.....	121
Figura 85 - Cenário 3 - Execução 2: Encaminhamento baseado em contexto.....	122

## LISTAS DE TABELAS

Tabela 1 - Comparação dos simuladores gratuitos de rede .....	27
Tabela 2 - Regras QoC .....	60
Tabela 3 - Descrição dos nós de rede .....	84
Tabela 4 - Descrição das conexões de dados.....	85
Tabela 5 - Caracterização do contexto das Estações .....	89
Tabela 6 - Caracterização dos fluxos.....	90
Tabela 7 - Cenário 1: Tempos dos fluxos.....	90
Tabela 8 - Resultados do Cenário 1 - Execução 1 .....	92
Tabela 9 - Resumo dos eventos de contexto no Cenário 1 - Execução 2 .....	93
Tabela 10 - Resultados do Cenário 1 - Execução 2.....	104
Tabela 11 - Cenário 2: Tempos dos fluxos.....	105
Tabela 12 - Resultados do Cenário 2 - Execução 1 .....	107
Tabela 13 - Resumo dos eventos de contexto no Cenário 2 - Execução 2 .....	108
Tabela 14 - Resultados do Cenário 2 - Execução 2.....	111
Tabela 15 - Cenário 3: Tempos dos fluxos.....	112
Tabela 16 - Resultados do Cenário 3 - Execução 1 .....	115
Tabela 17 - Resumo dos eventos de contexto no Cenário 3 - Execução 2 .....	116
Tabela 18 - Resultados do Cenário 3 - Execução 2.....	123

## LISTAS DE ABREVIATURAS E SIGLAS

ARPANET	Advanced Research Projects Agency Network
BGP	Border Gateway Protocol
CAARF	Context-aware Adaptive Routing Framework
CASs	Context-Aware Services
CBR	Constant Bit Rate
CEs	Contextual Elements
CPU	Central Processing Unit
CSV	Comma-Separated Values
FTP	File Transfer Protocol
GB	Gigabyte
Ghz	Gigahertz
GMPLS	Generalized MPLS
GPS	Global Positioning System
ID	Identity Document
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ITU	International Telecommunication Union
ITU-T	ITU Telecommunication Standardization Sector
JSON	JavaScript Object Notation
LAN	Local Area Network
MAC	Media Access Control
Mbps	Megabit por segundo
MOS	Mean Opinion Score
MOSu	Mean Opinion Score User
MOSs	Mean Opinion Score Systemic
MPLS	Multiprotocol Label Switching
Nam	Network Animator
NoSQL	non SQL / Not only SQL
NS	Network Simulator
NS2	Network Simulator version 2
NS3	Network Simulator version 3
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OTcl	MIT Object Tcl

QoC	Quality of Context
QoD	Quality of Device
QoE	Quality of Experience
QoS	Quality of Service
R-Factor	Rating Factor
RDF	Resource Description Framework
RIP	Routing Information Protocol
RSVP	Resource reSerVation Protoco
SDN	Software Defined Networks
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	Structured Query Language
TCL	Tool Command Language
TCP	TransmissionControlProtocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TOE	TCP offload engine
UDP	User Datagram Protocol
VoD	Video on Demand
VoIP	Voice over IP
WAN	Wide Area Network
Web	World Wide Web
WWW	World Wide Web
XML	eXtensible Markup Language
XSD	XML Schema Definition

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	16
1.1 CONTEXTUALIZAÇÃO .....	16
1.2 PROBLEMÁTICA .....	17
1.3 MOTIVAÇÃO .....	17
1.4 OBJETIVOS .....	19
1.5 METODOLOGIA .....	20
1.6 CONTRIBUIÇÕES E RESULTADOS ALCANÇADOS .....	21
1.7 ORGANIZAÇÃO DA DISSERTAÇÃO .....	22
<b>2 REVISÃO CONCEITUAL E DE LITERATURA</b> .....	<b>24</b>
2.1 INTRODUÇÃO .....	24
2.2 REVISÃO CONCEITUAL .....	24
<b>2.2.1 Redes de computadores</b> .....	<b>24</b>
<b>2.2.2 Simuladores de redes</b> .....	<b>26</b>
<b>2.2.3 Qualidade de serviço</b> .....	<b>31</b>
<b>2.2.4 Qualidade de experiência</b> .....	<b>31</b>
<b>2.2.5 Qualidade de dispositivo</b> .....	<b>32</b>
<b>2.2.6 Roteamento adaptativo</b> .....	<b>32</b>
<b>2.2.7 Engenharia de tráfego</b> .....	<b>33</b>
<b>2.2.8 Contexto</b> .....	<b>34</b>
2.3 ESTADO DA ARTE (REVISÃO DA LITERATURA) .....	35
<b>2.3.1 Engenharia de tráfego</b> .....	<b>35</b>
<b>2.3.2 Roteamento adaptativo baseado em contexto</b> .....	<b>36</b>
2.4 CONCLUSÃO .....	37
<b>3 ARQUITETURA BASEADA EM CONTEXTO</b> .....	<b>39</b>
3.1 INTRODUÇÃO .....	39
3.2 CAARF: CONTEXT-AWARE ADAPTIVE ROUTING FRAMEWORK .....	39
3.3 MODELO DE CONTEXTO .....	43
3.4 CAARF: ABORDAGEM OPERACIONAL .....	46
3.5 GESTÃO DE CONTEXTO .....	48
<b>3.5.1 Módulos da Gestão de Contexto</b> .....	<b>49</b>
<b>3.5.2 Eventos da Gestão de Contexto</b> .....	<b>52</b>
<b>3.5.3 Exemplos de informações coletáveis</b> .....	<b>57</b>
<b>3.5.4 Regras de QoS</b> .....	<b>59</b>
3.6 CONCLUSÃO .....	61

<b>4 SIMULAÇÃO: GESTÃO DE CONTEXTO</b> .....	<b>63</b>
4.1 INTRODUÇÃO.....	63
4.2 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS .....	63
4.3 MODELAGEM DA SIMULAÇÃO.....	64
4.4 MODELAGEM DA PERSISTÊNCIA.....	68
<b>4.4.1 Markup Description</b> .....	<b>68</b>
<b>4.4.2 Bases de Contexto</b> .....	<b>74</b>
4.5 LINGUAGENS E FERRAMENTAS DE IMPLEMENTAÇÃO.....	78
<b>4.5.1 Simulador</b> .....	<b>78</b>
<b>4.5.2 Rede baseada em contexto</b> .....	<b>78</b>
<b>4.5.3 Ferramenta de análise</b> .....	<b>79</b>
4.6 CONCLUSÕES .....	81
<b>5 ESTUDOS DE CASO</b> .....	<b>82</b>
5.1 INTRODUÇÃO.....	82
5.2 CARACTERIZAÇÃO DA TOPOLOGIA .....	83
5.3 CENÁRIO 1 .....	90
<b>5.3.1 Cenário 1 - Execução 1</b> .....	<b>91</b>
<b>5.3.2 Cenário 1 - Execução2</b> .....	<b>93</b>
5.4 CENÁRIO 2 .....	104
<b>5.4.1 Cenário 2 - Execução 1</b> .....	<b>105</b>
<b>5.4.2 Cenário 2 - Execução2</b> .....	<b>107</b>
5.5 CENÁRIO 3 .....	112
<b>5.3.1 Cenário 3 - Execução 1</b> .....	<b>113</b>
<b>5.3.2 Cenário 3 - Execução2</b> .....	<b>116</b>
5.4 DISCUSSÕES .....	123
5.5 CONCLUSÃO.....	124
<b>6 CONCLUSÕES E PERSPECTIVAS FUTURAS</b> .....	<b>125</b>
6.1 CONCLUSÕES .....	125
6.2 PERSPECTIVAS FUTURAS .....	128
<b>REFERÊNCIAS</b> .....	<b>130</b>
<b>APÊNDICE A - PUBLICAÇÕES DO AUTOR</b> .....	<b>135</b>
<b>ANEXO A - CÓDIGO</b> .....	<b>136</b>
<b>ANEXO B - TABELA COMPLEMENTAR DE REGRAS DE QOC</b> .....	<b>137</b>
<b>ANEXO C - INTEGRA DO ARQUIVO XML</b> .....	<b>139</b>

<b>ANEXO D - DEFINIÇÃO DE ESQUEMA, XSD, UTILIZADO PARA VALIDAR O XML.....</b>	<b>141</b>
<b>ANEXO E - CAMINHOS POSSÍVEIS ENTRE AS QUATRO REDES LOCAIS ..</b>	<b>144</b>
<b>ANEXO F - ÍNTEGRA DO ARQUIVO TCL DO CENÁRIO 1 - EXECUÇÃO</b>	<b>1145</b>
<b>ANEXO G- PROCESSAMENTO DO CÓDIGO AWK - EXEMPLO DO CÓDIGO PARA ATRASO FIM A FIM.....</b>	<b>151</b>
<b>ANEXO H - EXEMPLO DO CÓDIGO PARA BITS TRANSMITIDOS PARA LARGURA DE BANDA .....</b>	<b>152</b>
<b>ANEXO I - INTEGRA DO ARQUIVO XML ENVIADO NO INSTANTE (1,0)</b>	<b>153</b>
<b>ANEXO J - APLICATIVO CISCO WEBEX, NA PORTA REMOTA 9000.....</b>	<b>154</b>
<b>ANEXO K - PRINCIPAIS DIFERENÇAS NO ARQUIVO TCL DO CENÁRIO 1 - EXECUÇÃO 1 PARA O CENÁRIO 2 – EXECUÇÃO 2 .....</b>	<b>155</b>
<b>ANEXO L - DIFERENÇAS NO ARQUIVO TCL DO CENÁRIO 2 - EXECUÇÃO 1 PARA O CENÁRIO 3 – EXECUÇÃO 1 .....</b>	<b>156</b>



# 1 INTRODUÇÃO

## 1.1 CONTEXTUALIZAÇÃO

Com a intensificação no uso das redes de computadores, com o crescimento de uso dos computadores pessoais e com a intensificação da troca de informações instantaneamente nos negócios, na indústria, na educação, entre outros, rompendo barreiras geográficas, a carga das redes é cada vez maior (FOROUZAN, 2006).

Da mesma forma, o surgimento de novas aplicações e maior demanda por aplicações em tempo real levou à crescente coexistência de tráfegos distintos na mesma rede (TANENBAUM, 2003) (KUROSE e ROSS, 2010). Portanto, torna-se necessário buscar novas alternativas para gestão das redes e seus tráfegos.

Nos últimos 20 anos diferentes propostas tentam garantir justiça no uso de recursos de rede, entre estas, técnicas de Qualidade de Serviço (QoS) têm sido adotadas, como Serviços Integrados (IntServ) (RFC2205, 1997), Serviços Diferenciados (DiffServ) (RFC2475, 1998), encaminhamento baseado em etiquetas como o *MultiprotocolLabelSwitching* (MPLS) (RFC3031), e posteriormente sua evolução, a generalização do MPLS (*Generalized MPLS*) (RFC3945, 2004).

Considerando ainda que as soluções tradicionais de Engenharia de Tráfego (TE) (FORTZ, 2002) e Qualidade de Serviço (QoS) (TANENBAUM, 2003) são baseadas em métricas orientadas à rede, tais como, largura de banda (*throughput*), atraso (*delay*), variação do atraso (*jitter*), perda (*packetloss*) e confiabilidade (*reliability*) (EL-GENDY, 2003), tais métricas tornaram-se insuficientes para atender as expectativas destas novas aplicações e principalmente as expectativas dos usuários.

Em paralelo, novos parâmetros vêm sendo avaliados para auxiliar a tomada de decisão em ambientes computacionais, inclusive em redes de computadores, relacionados à Qualidade de Experiência (QoE) (ITU-T, 2015) (MITRA et al. 2011), Qualidade de Dispositivo (QoD) (BUCHHOLZ e SCHIFFERS, 2003) (NAZARIO et al. 2012) e principalmente ao Contexto (WEISER, 1999) (LAGHARI AND CONNELLY, 2012).

Essa nova abordagem de contexto, QoS, QoD e QoC já vem sendo explorada em alguns trabalhos de roteamento adaptativo baseado em contexto, principalmente para tomada de decisão nas próprias estações de trabalho, geralmente em redes se, fio móveis ou *Ad Hoc*,

como por exemplo nos trabalhos de (WENNING et al., 2009), (YASAR et al., 2010) e (PETZ et al., 2012) .

## 1.2 PROBLEMÁTICA

Considerando a contextualização apresentada, foram identificados os principais problemas que norteiam a proposta deste trabalho de mestrado:

- Intensificação no uso das redes de computadores;
- Tráfegos distintos coexistindo na mesma rede;
- Aumento dos tráfegos em tempo real;
- Nos últimos 20 anos diferentes propostas tentam garantir justiça no uso de recursos de rede;
  - Técnicas de QoS têm sido adotadas: IntServ, DiffServ, MPLS/GMPLS, Engenharia de Tráfego;
  - Tentam garantir requisitos para tráfegos em tempo real;
  - Em geral, tais técnicas focam na admissão, controle de congestionamento, gestão dos *buffers* e programação;
- No entanto, a utilização apenas de QoS mostra-se insuficiente para atender as expectativas do usuário;
- As abordagens tradicionais de QoS e Engenharia de Tráfego não contemplam a percepção dos usuários da rede (QoE), e;
- Parâmetros adicionais como QoD, QoE e Contexto já são explorados em alguns propostas, no entanto, voltadas às redes sem fio e à tomada de decisão local nos dispositivos.

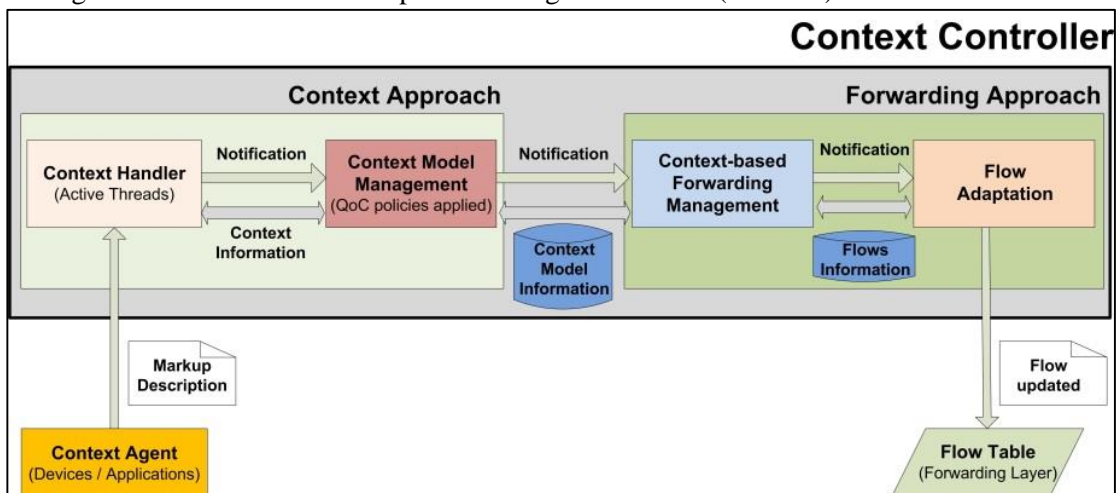
## 1.3 MOTIVAÇÃO

Buscando alternativas aos problemas elencados anteriormente, mostra-se necessária uma solução que integre as soluções tradicionais de gestão da rede e dos fluxos, QoS e ET, com a possibilidade de contemplar a percepção do usuário (QoE), e outros novos parâmetros já em estudo como QoD e principalmente Contexto.

Desta forma, o *Context-aware Adaptive Routing Framework* (CAARF) proposto mostra-se como uma alternativa para a tomada de decisão de encaminhamento em redes de computadores baseada em informações contextuais coletadas dos dispositivos envolvidos, assim como as informações dos próprios dispositivos (QoD) e da percepção dos usuários envolvidos (QoE), sem no entanto ignorar os parâmetros tradicionais de gestão de redes, QoS e ET (CAARF. 2014) .

O projeto do arcabouço CAARF está sendo proposto e desenvolvido como resultado da contribuição integrada de quatro alunos do Programa de Mestrado em Sistemas e Computação da Universidade Salvador (UNIFACS). Neste projeto, os diferentes alunos (André Oliveira, Carlos Muakad, Joatham Silva e Sérgio Spínola) estão realizando contribuições distintas e complementares de forma a proporcionar a modelagem, desenvolvimento e validação do arcabouço CAARF, apresentado na Figura 1.

Figura 1 - Context-aware Adaptive Routing Framework (CAARF)



Na Figura 1 podem ser observadas três estruturas, o controlador de contexto que abriga os módulos principais do arcabouço: *ContextController*, o agente responsável pela coleta de informações de contexto dos dispositivos de rede e dos usuários conectados; *Context Agent*, sendo deste agente a responsabilidade do envio das informações de contexto em arquivos; *MarkupDescription*, relacionada às tabelas de fluxos atualizadas nos dispositivos encaminhadores da rede com as atualizações provenientes do arcabouço.

Quanto ao controlador de contexto, *ContextController*, este está dividido em duas grandes abordagens operacionais, a abordagem de contexto e abordagem de encaminhamento, *Context Approach* e *Forwarding Approach* respectivamente. A abordagem de contexto é

responsável pela centralização das informações de contexto coletadas pelos agentes, validação das políticas e regras de qualidade de contexto (QoC) e disponibilização do contexto atual através de sua visão do modelo de contexto, *ContextModelView*. Já a abordagem de encaminhamento é responsável pela consulta ao contexto ativo e processamento das regras de encaminhamento para definição dos caminhos na rede, e posteriormente aplicação destas novas decisões nos dispositivos de encaminhamento.

Portanto, a principal motivação para a proposta deste trabalho foi contribuir com a definição e desenvolvimento de uma arquitetura para o arcabouço proposto, mais especificamente contribuindo com sua abordagem de contexto. O objetivo geral e objetivos específicos são apresentados na próxima seção.

#### 1.4 OBJETIVOS

O trabalho apresentado nesta dissertação de mestrado tem como objetivo geral a definição, simulação e validação de estratégias para gerenciamento do contexto, incluindo representação, coleta, persistência e validação das informações contextuais, sendo este parte de uma arquitetura para o desenvolvimento de uma solução de encaminhamento adaptativo baseado em contexto. Desta forma, o objetivo geral pode ser dividido nos seguintes objetivos específicos:

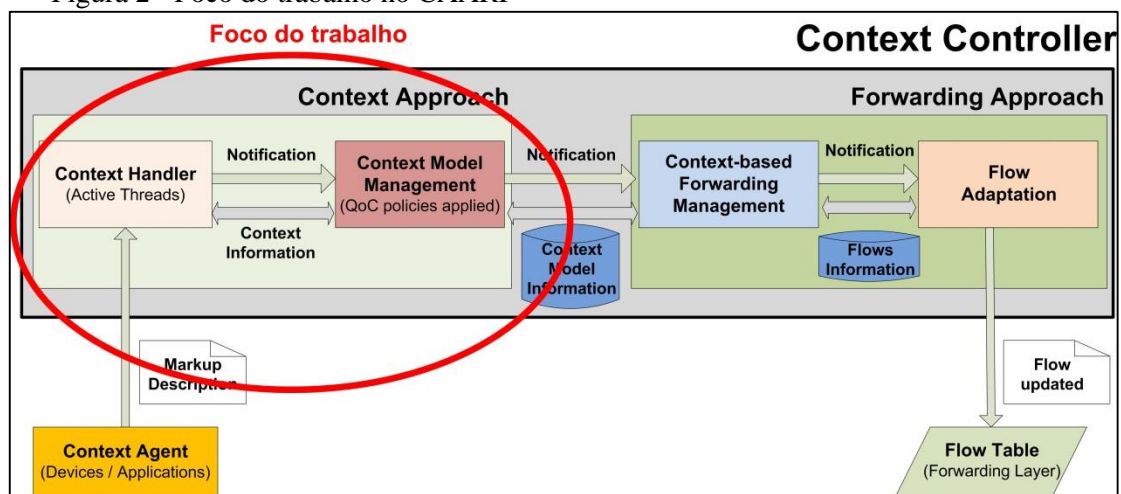
- **Propor um modelo de contexto** que represente dispositivos, aplicações e usuários de redes convergentes;
- **Elencar informações de contexto relevantes** para o encaminhamento de fluxos em redes convergentes;
- **Propor regras de Qualidade de Contexto (QoC)** a serem aplicadas nas informações contextuais a fim de melhorar os resultados obtidos pela solução de encaminhamento baseada em contexto;
- **Definir a abordagem de contexto**, sua estrutura interna, módulos, ações e notificações necessárias, desde a coleta das informações até a entrega via base de contexto;
- **Propor experimentalmente uma solução de encaminhamento adaptativo baseado em informações de contexto** para redes convergentes que suporte

novas abordagens e métricas de qualidade e a percepção do contexto que envolve as entidades que fazem uso da rede, e;

- **Validação da solução proposta** através da realização de estudos de casos para a otimização do encaminhamento baseado em informações de contexto e utilizando cenários simulados.

Para isso, o foco deste trabalho de mestrado está na abordagem de contexto (*ContextApproach*), mas especificamente no módulo de gerenciamento do modelo de contexto (*ContextModel Management*), conforme destacado na Figura 2.

Figura 2 - Foco do trabalho no CAARF



Conforme Figura 2, a validação do arcabouço envolve a execução do ciclo de processamento realizado pela abordagem de encaminhamento (*ForwardingApproach*), e para isso este trabalho disponibilizará as informações de contexto através de uma visão dos dados (*ContextModelInformation*), que poderá ser aplicado para a validação do módulo de abordagem de encaminhamento tratado em (MUAHAD, 2015) sobre os dados de contexto fornecidos.

## 1.5 METODOLOGIA

A seguir é apresentada a metodologia adotada para elaboração do trabalho proposto:

- Realização de estudos comparativos das soluções e modelos existentes para o encaminhamento adaptativo baseado em contexto;

- Estudo de trabalhos relacionados para auxiliar na proposta e posterior definição de um modelo de dados para representação das informações contextuais a serem coletadas dos dispositivos, usuários e elementos de rede;
- Estudo de trabalhos relacionados para auxiliar na proposta e posterior definição das relações entre os módulos do arcabouço para processamento das informações contextuais e tomada de decisão sobre o encaminhamento;
- Definição do modelo de dados a ser usado pelo arcabouço CAARF e a proposta dos mecanismos e regras necessários para o gerenciamento das informações de contexto;
- Definição e desenvolvimento do módulo de gerência de informações de contexto;
- Definição do arcabouço CAARF e relações entre os módulos e troca de notificações necessárias para seu funcionamento;
- Estudo e identificação de um conjunto de ferramentas necessárias para a implementação da solução proposta, tais como banco MySQL, arquivos de marcação XML, simulação em redes com o NS2, entre outros;
- Validação da solução implementada através de estudos de caso baseados em cenários simulados, usando técnicas com modificações do encaminhamento das informações.

## 1.6 CONTRIBUIÇÕES E RESULTADOS ALCANÇADOS

Como contribuições do trabalho realizado nesta dissertação de mestrado estão as discussões de propostas e soluções em áreas que envolvem roteamento adaptativo, Qualidade de Serviço (QoS), Qualidade de Dispositivo (QoD), Qualidade de Experiência (QoE), Contexto e Qualidade de Contexto (QoC). Em cada área dando um enfoque específico, como no caso, por exemplo, da qualidade de contexto, possibilitando o filtro das informações de contexto coletadas e validando para que apenas aquelas que possam ser utilizadas para tomada de decisão de encaminhamento sejam disponibilizadas para processamento dos demais módulos do arcabouço.

Contribuições da área da qualidade do dispositivo e qualidade de experiência também ocorreram, permitindo ampliar a visão das relações das características dos dispositivos influenciando a satisfação dos usuários, alocando recursos de rede apenas nos casos onde os

dispositivos envolvidos possam efetivamente fazer uso deles, portanto, evitando desperdício de recursos que poderiam ser alocados para outros que poderiam usufruir melhor deles. Ainda na qualidade de experiência, mostra uma possibilidade de uso da percepção do usuário quanto ao serviço que está sendo oferecido a ele, ampliando sua importância nos aspectos de avaliação da rede e seu encaminhamento de informações.

Outra contribuição foi definição de um arcabouço para encaminhamento adaptativo genérico que utiliza informações de contexto, devidamente qualificadas, para tomada de decisão. Sendo apresentados conceitos e definições relacionadas, bem como sua estrutura, funcionamento, interação entre os módulos e trocas de mensagens de notificações. Tendo sido o trabalho apresentado cabível de customização e extensão, permite que outros pesquisadores possam contribuir com ele, corrigindo e melhorando seu funcionamento.

E ainda, como contribuição direta, foi proposta a abordagem de contexto, mais especificamente o módulo de gerenciamento do modelo de contexto do arcabouço proposto e foram realizados três estudos de caso baseados em cenários simulados para ilustração do funcionamento do arcabouço. Estas simulações demonstram o gerenciamento das informações de contexto que permitem a avaliação do contexto atual e tomada de decisão de encaminhamento baseada em critérios e redefinição dos caminhos na rede.

## 1.7 ORGANIZAÇÃO DA DISSERTAÇÃO

Além deste capítulo introdutório, esta dissertação está organizada em mais cinco capítulos, conforme apresentados a seguir:

- Capítulo 2 – Revisão conceitual e de literatura: onde são apresentados os aspectos teóricos e conceituais utilizados ao longo da dissertação, além da identificação e estudo dos principais trabalhos relacionados existentes na literatura;
- Capítulo 3 – Arquitetura baseada em Contexto: onde são apresentados conceitos e características do *Context-aware Adaptive Routing Framework* (CAARF), seu modelo de dados, e mais especificamente sua abordagem de contexto;
- Capítulo 4 – Simulação: Gestão de Contexto: onde é apresentada a estruturação da simulação realizada para validação do arcabouço, os principais resultados da contribuição desta dissertação de mestrado, relacionados à definição de uma

arquitetura para o arcabouço CAARF, mais especificamente dos módulos da abordagem de contexto;

- Capítulo 5 – Estudos de caso: onde são apresentados alguns estudos de casos como forma de validar o funcionamento do arcabouço CAARF, proporcionando melhores soluções para o encaminhamento baseado em contexto e algumas conclusões acerca destes estudos;
- Capítulo 6 – Conclusões e perspectivas futuras: onde são discutidas as principais conclusões deste trabalho e de sua continuação através das perspectivas futuras.



## **2 REVISÃO CONCEITUAL E DE LITERATURA**

### **2.1 INTRODUÇÃO**

O trabalho realizado nesta dissertação de mestrado utiliza conceitos de redes de computadores, simulação em redes de computadores, qualidade de serviço e dispositivo, contexto, entre outros, além de criar algumas extensões e adaptações de alguns deles, como por exemplo a qualidade de contexto. A fim de proporcionar uma melhor compreensão desses conceitos, neste capítulo é realizada uma revisão conceitual e é apresentada uma revisão da literatura relacionada aos conceitos apresentados.

O capítulo está dividido em uma seção com a revisão conceitual sobre as redes de computadores, redes definidas por software, simuladores de redes, qualidade de serviço, qualidade de experiência, qualidade de dispositivo, engenharia de tráfego e contexto. E outra seção com o estado da arte sobre roteamento adaptativo, qualidade de serviço, qualidade de experiência, qualidade de dispositivo, engenharia de tráfego, aplicações e redes baseadas em contexto.

### **2.2 REVISÃO CONCEITUAL**

São definidos nesta seção os conceitos base que serão utilizados neste trabalho.

#### **2.2.1 Redes de computadores**

O uso das redes de computadores, juntamente com advento do computador pessoal interconectado nestas redes, tem possibilitado a troca de informações quase instantaneamente, rompendo barreiras geográficas, e mudando os negócios, a indústria, a ciência e a educação (FOROUZAN, 2006).

Segundo apresentado em (FOROUZAN, 2006), as redes de comunicação foram desenvolvidas para o envio de dados de um lugar ao outro, e para isso é necessário que estas redes saibam criar e gerenciar diversos tipos de fluxos de dados que serão transmitidos por elas. São componentes destas as conexões dos computadores, os ativos de redes tais como os meios de transmissão e dispositivos de redes, e seus protocolos.

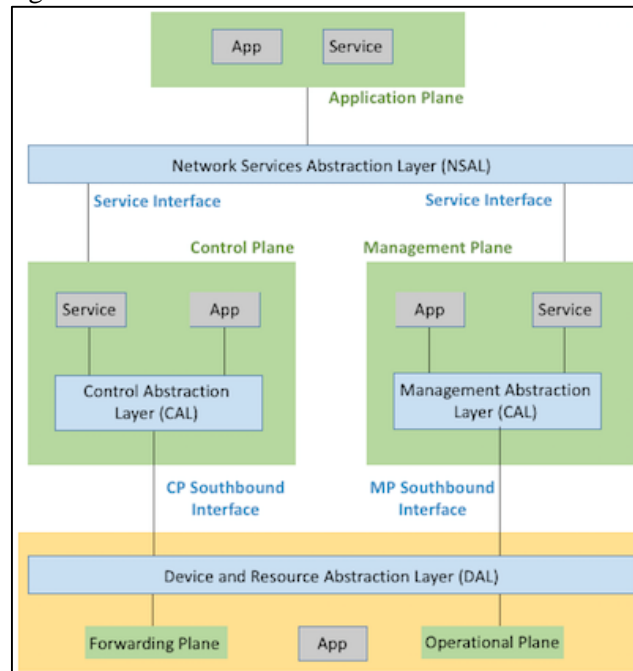
Em (TANENBAUM, 2003) as redes de computadores são definidas como a interconexão de um conjunto de computadores autônomos utilizando a mesma tecnologia. É feita a distinção do conceito de redes de computadores do conceito de sistemas distribuídos, sendo este último um conjunto de computadores independentes que parecem ser um único sistema para um usuário, em geral, tendo um único paradigma ou modelo aparente para este usuário.

O processo de escolher o caminho dentro da rede da origem até o destino é chamado roteamento, e a construção da tabela de caminhos pode ser feita de forma estática ou via algum algoritmo de roteamento (TANENBAUM, 2003). Dependendo dos protocolos de rede utilizados, a escolha de caminhos é feita a partir de informações existentes no cabeçalho do pacote, tais como endereço de destino do pacote ou alguma indicação da conexão a qual ele pertence (KUROSE ; ROSS, 2010) .

Entre os algoritmos de roteamento dinâmico, os mais conhecidos são o vetor distância e estado dos enlaces, sendo este primeiro o algoritmo de roteamento original da ARPANET, precursora da Internet. Seu algoritmo consiste em manter uma tabela, vetor, com a distância conhecida para chegar a qualquer destino, escolhendo o melhor caminho entre os disponíveis (TANENBAUM, 2003).

Saindo da escolha de caminhos baseada em informações de algoritmos de roteamento presentes dentro dos roteadores, existe a nova abordagem das Redes Definidas por *Software* (*Software Defined Networks* - SDN), onde se enfatiza o papel do software na tomada de decisão das redes, tornando- as programáveis (RFC7426, 2015). Os planos das Redes Definidas por *Software* podem ser vistos na Figura 3.

Figura 3 - The RFC 7426 SDN Architectural Model



Fonte: Sdnskills (2015).

Na Figura 3, pode-se identificar o desacoplamento do plano de Encaminhamento e Operação implementado nos dispositivos de rede do plano de Controle e Gerência da rede que passa a ser executado fora destes dispositivos. No plano de Aplicação estão as aplicações e serviços de rede. Entre eles camadas de abstração com interfaces de conexão entre os planos (RFC7426, 2015).

### 2.2.2 Simuladores de redes

A simulação de redes de computadores é uma alternativa a modelagem e validação de cenários de redes de computadores reais, reduzindo custos e tempo para a experimentação e estudos (SIRAJ et al, 2012). Baseado no estudo apresentado em (SIRAJ et al, 2012), foram avaliados os simuladores *Network Simulator*, NS2 (NS2, 2015), NS3 (NS3, 2015), OPNET (RIVERBED, 2015), NetSim (NETSIM, 2015), OMNeT++ (OMNeT, 2015), REAL (REAL, 2015), J-Sim (JSIM, 2015) e QualNet (QUALNET, 2015). São apresentadas suas características básicas e modos de operação, sem comparação direta entre eles.

Os principais simuladores gratuitos podem ser identificados no resumo adaptado da comparação apresentado em (GUPTA et al, 2013), conforme a Tabela 1.

Tabela 1 - Comparação dos simuladores gratuitos de rede

<b>Simulador</b>	<b>Custo e Código</b>	<b>Linguagem</b>	<b>Tipos de Redes</b>	<b>Pontos relevantes</b>
NS2	Gratuito; Código aberto	C++ OTcl	Redes cabeadas e sem fio	Ampla documentação e exemplos; Suporte pela comunidade; Desenvolvimento interrompido
NS3	Gratuito; Código aberto	C++ Python	Redes cabeadas, sem fio, sensores sem fio	Sucessor do NS2; Suporte novas rede; Pouca documentação e exemplos
OMNeT++	Gratuito não comercial; Código proprietário	C++	Redes cabeadas e sem fio estruturadas	Suporte gráfico aprimorado; Uso em expansão; Sem desenvolvimento pela comunidade
JSIM	Gratuito; Código aberto	Java Tcl	Redes cabeadas, sem fio, sensores sem fio	Maior escalabilidade para redes e eventos; Mais lento que outros simuladores

Fonte: Adaptado de Gupta et al (2013).

Os simuladores apresentados na tabela de (GUPTA et al, 2013) também são mencionados em (SIRAJ et al, 2012) e em (BORBORUAH e NANDI, 2014).

Em (GUPTA et al, 2013) são realizadas análises dos simuladores de código aberto NS2, NS3, JSIM e do simulador gratuito, para fins não comerciais, OMNeT++, sendo discutidas suas vantagens e desvantagens, considerando sua eficiência para uso acadêmico por estudantes, pesquisadores e pela indústria. Foram considerados seus recursos disponíveis na época do estudo e perspectivas de novas funcionalidades em desenvolvimento, além da facilidade de aprendizado e usabilidade. Na opinião dos pesquisadores, o NS2 foi considerado a melhor opção, ressalvando que para interessados em interface gráfica, a escolha mudaria para o OMNeT++, e que para simulações de redes grandes, também recomendam não escolher o NS2. Os pesquisadores ainda destacam o fato do desenvolvimento do NS2 ter sido interrompido, sendo seu substituído o NS3, embora os formatos não sejam compatíveis, e o fato do NS3 ainda estar em desenvolvimento, sem a popularidade e suporte a modelos do NS2.

Já em (BORBORUAH e NANDI, 2014), são analisados os simuladores proprietários OPNET, NetSim, e QualNet, e os de código aberto NS2 e o NS3, além do gratuito para fins não comerciais OMNet++. Considerando apenas os gratuitos, o NS2 é amplamente usado pela

abundância de pacotes e contribuições disponíveis. Já o NS3 é considerado o sucessor do NS2, com suporte a redes de novas tecnologias, embora ainda tenha desafios a superar, entre eles, maior participação da comunidade de pesquisa, pouca documentação e maior validação dos resultados obtidos. Quanto ao OMNet++, são destacadas sua interface gráfica, seu crescimento no meio acadêmico e na indústria, sendo destacado ainda seu crescimento, mas com falta de participação da comunidade em seu desenvolvimento.

Considerando os dados levantados, foram realizadas pesquisas com o NS3 para validação do trabalho realizado nesta dissertação, mas sua falta de documentação, poucas simulações disponíveis com códigos e resultados para comparação, e alto investimento em tempo para adaptação às linguagens C++ e Python, inviabilizou o desenvolvimento do experimento pretendido.

Considerando que o NS2 apresenta suporte para todas as tecnologias de redes abordadas nesta pesquisa, a ampla disponibilidade de documentação, códigos e resultados para análise comparativa, e ainda o conhecimento já existente em OTcl, optou-se por substituir o NS3 pelo NS2, sendo este último o simulador de redes utilizado neste trabalho.

## **Simulador NS2**

As simulações em NS2 são escritas em OTcl (NS2, 2015), uma extensão orientada a objetos da linguagem Tcl (OTCL, 2015). São parâmetros básicos de uma simulação:

- i. Criação do simulador;
- ii. Tempo de simulação;
- iii. Cor do fluxo (opcional);
- iv. Arquivo de traço;
- v. Arquivo de simulação NAM (opcional);
- vi. Definição dos nós;
- vii. Definição dos links;
- viii. Posicionamento dos nós para a simulação NAM (opcional);
- ix. Definição dos agentes;
- x. Definição das aplicações;
- xi. Processos a serem executados ao final da simulação.

O código resumido com a estrutura apresentada pode ser vista da Figura 4.

Figura 4 - Exemplo de simulação NS2: OTcl

```

# i
set ns [new Simulator]
# ii
set val(stop) 3.0
# iii
$ns color 0 Green
# iv
set tracefile [open out.tr w]
$ns trace-all $tracefile
# v
set namfile [open out.nam w]
$ns namtrace-all $namfile
# vi
set n0 [$ns node]
# (...) código resumido
# vii
$ns simplex-link $n0 $n1 1.0Mb 10ms DropTail
$ns queue-limit $n0 $n1 50
# (...) código resumido
# vii
$ns simplex-link-op $n0 $n1 orient right-up
# (...) código resumido
# ix
set udp0 [new Agent/UDP]
$udp0 set class_ 0
$udp0 set fid_ 0
$ns attach-agent $n0 $udp0
set null1 [new Agent/LossMonitor]
$ns attach-agent $n3 $null1
$ns connect $udp0 $null1
$udp0 set packetSize_ 1500
# x
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set rate_ 0.1Mb
$ns at 1.0 "$cbr0 start"
$ns at 2.0 "$cbr0 stop"
# xi
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam -a out.nam &
    exit 0
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run

```

A integra deste código encontra-se no Anexo A. Neste código é possível identificar uma simulação de 3 minutos (ii), com quatro nós (vi), interconectando por ligações de 1 Mbps com e filas de 50 pacotes (vii). Os arquivos de saída foram definidos em (iv e v), onde um fluxo CBR/UDP de 0,1 Mbps é transmitido do minuto 1 ao 2 (ix e x).

Uma amostra do arquivo de traço gerado pode ser vista na Figura 5.

Figura 5 - Exemplo de simulação NS2: Traço

```

+ 1 0 1 cbr 210 ----- 0 0.0 3.0 0 0
- 1 0 1 cbr 210 ----- 0 0.0 3.0 0 0
r 1.01168 0 1 cbr 210 ----- 0 0.0 3.0 0 0
+ 1.01168 1 3 cbr 210 ----- 0 0.0 3.0 0 0
- 1.01168 1 3 cbr 210 ----- 0 0.0 3.0 0 0

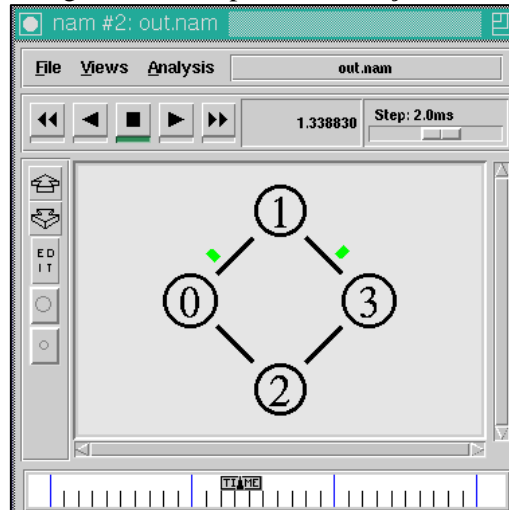
```

Na Figura 5 é possível ver um exemplo dos registros das informações dos tempos dos eventos, geração e consumo de pacotes por cada nó da rede, entre outros dados. (NS2, 2015)

Outros arquivos de traços podem ser gerados com informações de, por exemplo, atraso, perda de pacotes e largura de banda utilizada (NS2, 2015), servindo como dados resumidos auxiliares à análise do arquivo de traço.

Uma visualização gráfica deste arquivo de traço pode ser vista na ferramenta *Network Animator* (NAM) (NAM, 2015), conforme Figura 6.

Figura 6 - Exemplo de simulação NS2: NAM



Na Figura 6 o fluxo gerado pelo nó 0 é enviado pelos link 0-1 e 1-3 até o nó 3 onde é consumido.

A topologia do exemplo apresentado é extremamente simples, existindo apenas dois caminhos entre a origem e o destino. Entretanto, mesmo em redes tão pequenas, a escolha por um caminho ou por outro pode influenciar as características dos resultados obtidos. Esta

mudança de caminhos e seus resultados serão demonstrados mais adiante nos cenários propostos no capítulo 5.

### 2.2.3 Qualidade de serviço

Segundo (TANENBAUM, 2003), as redes de computadores têm registrado crescimento de uso e variação cada vez maior dos tipos de aplicações que transitam sobre elas, principalmente aplicações multimídia. Estas aplicações necessitam de parâmetros de rede distintos, tais como de confiabilidade, retardo, flutuação e largura de banda, e juntos estes parâmetros ajudam a definir os requisitos de Qualidade de Serviço (*Quality of Service - QoS*) necessários para o funcionamento adequado desta aplicação de rede. Já em (EL-GENDY, 2003) são considerados como parâmetros que definem o QoS a largura de banda (*throughput*), o atraso (*delay*), a variação do atraso (*jitter*), a perda (*loss*) e a confiabilidade (*reliability*).

Diferentes propostas foram desenvolvidas, segundo (TANENBAUM, 2003) e (EL-GENDY, 2003), os principais paradigmas são os Serviços Integrados (*Integrated Services - IntServ*) e sua implementação do protocolo de reserva de recursos (Resource reSerVation Protocol – RSVP) (RFC2205, 1997), os Serviços Diferenciados (*Differentiated Services - DiffServ*) (RFC2475, 1998), as redes (*Multiprotocol Label Switching – MPLS*) (RFC3031, 2001). E posteriormente evolução da generalização do MPLS (*Generalized MPLS*) (RFC3945, 2004). Estas técnicas focam em estratégias para tentar determinar configurações ou provisionar recursos para os serviços ou grupos de serviços que fazem uso da rede.

O uso de técnicas de QoS permite que os usuários das redes de computadores, os clientes, e seus prestadores de serviços definam acordos com as expectativas da qualidade do serviço a ser prestada, os acordos de nível de serviço (*Service Level Agreement - SLA*) que vão nortear o que é esperado da rede e os indicadores de desempenho que serão utilizados e as formas de medição (TANENBAUM, 2003).

### 2.2.4 Qualidade de experiência

Com o crescimento e diversificação dos meios de telecomunicação tornou-se necessário medir a qualidade das chamadas de voz realizadas, e para isso são definidos critérios a serem respondidos pelo usuário do sistema de telecomunicações. Entre outros, um



dos critérios mais utilizados é a média do escore de opinião (*Mean Opinion Score – MOS*) (ITU-T, 2015).

Esta métrica é adotada desde 1996 pela União Internacional das Telecomunicações (*International Telecommunication Union – ITU*) e define cinco níveis de satisfação quanto a qualidade da ligação, sendo cinco (5) para excelente, quatro (4) para bom, três (3) para regular, dois (2) para pobre e um (1) para ruim (ITU-T, 2015).

Segundo (MITRA et al. 2011), essa avaliação da percepção da qualidade pelo ponto de vista do usuário define a Qualidade de Experiência (*Quality of Experience – QoE*), que compreende fatores cognitivos, comportamentais e psicológicos. Em (LAGHARI AND CONNELLY, 2012) são consideradas como influenciadoras do QoE as áreas: contextuais, tecnológicas, comerciais e humanas.

### **2.2.5 Qualidade de dispositivo**

Sendo mais um fator relevante a ser analisado para uso adequado das redes de computadores, e que pode influenciar a percepção de qualidade do usuário (QoE) ou mesmo o desempenho da rede (QoS), é importante avaliar a Qualidade do Dispositivo (*Quality of Device - QoD*). A QoD irá prover informações e características técnicas de cada dispositivo e suas capacidades (NAZARIO et al. 2012).

Em (BUCHHOLZ e SCHIFFERS, 2003) é exemplificada a característica de um dispositivo que tem precisão na coleta do sistema de posicionamento global (*Global Positioning System- GPS*) inferior ao desejável para um uso específico, influenciando no resultado esperado deste sistema computacional. Pode-se ainda através do QoD identificar incompatibilidades entre um dispositivo em particular em relação a outro equipamento, o que poderia torná-lo incompatível para determinado uso.

### **2.2.6 Roteamento adaptativo**

Segundo apresentado em (FOROUZAN, 2006), roteamento é o processo de escolha de caminhos dentro da rede, decidindo entre as alternativas existentes para onde encaminhar os pacotes. Tipicamente usa-se o roteamento estático, quando os caminhos são configurados manualmente pelo administrador da rede, ou o roteamento dinâmico, quando protocolos de

roteamento em execução nos roteadores trocam informações para construção automatizada da tabela de roteamento (FOROUZAN, 2006) (TANENBAUM, 2003) (KUROSE; ROSS, 2010). São exemplos de protocolos de roteamento dinâmico *Routing Information Protocol*(RIP), *Open Shortest Path First* (OSPF), *Border Gateway Protocol* (BGP). (KUROSE; ROSS, 2010)

Considerando a dinamicidade das redes, com mudanças de topologia e tráfego que podem ocorrer durante sua operação, é desejável utilizar algoritmos que se adaptem junto, algoritmos de roteamento adaptativos que mudem suas decisões de roteamento junto com as mudanças da rede (TANENBAUM, 2003).

### **2.2.7 Engenharia de tráfego**

A engenharia de tráfego visa a busca pela adaptação do roteamento para as condições pelas quais a rede passe em um determinado momento, melhorando o desempenho e a eficiência da rede para o usuário (FORTZ, 2002) . Segundo a (RFC3272, 2002), engenharia de tráfego é:

- i – Um conjunto de políticas, objetivos e requisitos para avaliação do desempenho das redes, que pode ou não incluir dependências de contexto, e ainda mecanismos para otimização de seu desempenho;
- ii – Uma coleção de ferramentas e mecanismos de medição, caracterização, modelagem, controle do tráfego e alocação de recursos de rede, bem como o controle sobre o mapeamento e a distribuição dos tráfegos pela infraestrutura de rede;
- iii – Um conjunto de restrições sobre o ambiente operacional, os protocolos de rede e o próprio sistema de engenharia de tráfego;
- iv – Um conjunto de técnicas e metodologias quantitativas e qualitativas para abstrair, formular e resolver problemas de engenharia de tráfego;
- v – Um conjunto de parâmetros de controle administrativos que podem ser manipulador através de sistemas de gerenciamento de configurações, podendo incluir subsistemas e repositórios para medição e auditoria, e;
- vi – Um conjunto de diretrizes para avaliação do desempenho da rede, otimização e melhoria de desempenho.

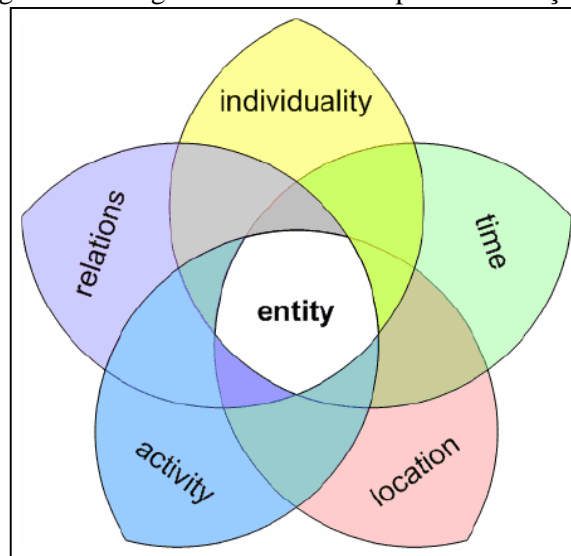
### 2.2.8 Contexto

Contexto pode ser definido como qualquer informação que possa ser usada para caracterizar uma situação referente a uma entidade, seja uma pessoa, grupos de pessoas, identidades, um lugar ou um objeto físico ou computacional (DEY et al., 2001). Os Serviços Sensíveis ao Contexto (*Context-Aware Services - CASs*) foram definidos originalmente em (WEISER, 1999) e visam melhorar a experiência do usuário e os resultados obtidos da interação deste com os serviços computacionais.

A definição de contexto está também relacionada com a descrição de Elementos Contextuais (*Contextual Elements - CEs*), pois, enquanto o contexto refere-se à interação entre um agente e uma aplicação, CEs caracterizam o domínio de contexto no qual aquele agente está inserido (VIEIRA et al. 2007).

Em (ZIMMERMANN et al., 2007) é proposto um relacionamento das informações contextuais e uma definição de seus elementos, conforme Figura 7.

Figura 7 - Categorias fundamentais para Informação de Contexto



Fonte: Zimmenmann et al, (2007).

Na Figura 7 pode-se identificar a entidade como sendo uma intersecção das informações contextuais que a caracterizam, sendo elas:

- Individualidade (*Individuality*): representando informações que permitam identificar de forma única uma entidade;

- Tempo (*Time*): representando informações temporais que permitam avaliar quando uma informação de contexto foi gerada ou, ou o tempo no qual alguma atividade relacionada ao contexto foi realizada;
- Localização (*Location*): representando elementos que permitam identificar a localização da entidade, podendo ser uma localização precisa de onde se encontra, ou se esta próximo a algum outro elemento contextual ou mesmo outra entidade;
- Atividade (*Activity*): representando as atividades sendo realizadas pela entidade;
- Relações (*Relations*): representando as relações entre a entidade e outras entidades.

Podem ser consideradas informações relevantes da interação entre o usuário e uma aplicação, incluindo o próprio usuário e aplicação associados, seu uso desta aplicação, e informações contextuais que influenciem suas interações e os resultados que a aplicação possa apresentar (DEY et al., 2001).

Desta forma, aplicações sensíveis ao contexto são capazes de propiciar serviços com tarefas baseadas em assistência, ações baseadas em contexto e adaptação do comportamento do sistema de acordo com informações contextuais (VIEIRA et al., 2007) (NAZARIO et al., 2012).

Saindo do foco nas aplicações, programas, e considerando o uso mais amplo em sistemas computacionais, em (MUSOLESI, 2005), (WENNING et al., 2009) e (PETZ, 2012) o contexto é utilizado para tomada de decisão na comunicação em redes de computadores.

## 2.3 ESTADO DA ARTE (REVISÃO DA LITERATURA)

Nesta seção é apresentada uma revisão da literatura associada a temática correlata ao foco deste trabalho.

### 2.3.1 Engenharia de tráfego

Estudos avaliando o crescimento das redes de computadores e seus fluxos vêm sendo realizados buscando otimização através de análises dinâmicas, previsão de crescimento e regulação dos estados dos enlaces (KARTHIGA; BALAMURUGAN, 2013) . São considerados para isso parâmetros de QoS como perda de pacotes, atraso, taxa de utilização dos enlaces.

Sendo a engenharia de tráfego focada principalmente em QoS, em (ALRESHOODI; WOODS, 2013) é possível acompanhar a análise de trabalhos que relacionam a influência de fatores relacionados a qualidade de serviço (QoS) e a qualidade de experiência (QoE), e a consequente hipótese de que quando a experiência do usuário esta satisfatória, provavelmente a qualidade de serviço da rede esta em níveis melhores.

Em (CHIOCCHETTI et. al, 2013) é discutido o encaminhamento de requisições dinâmicas para o novo paradigma de distribuição de conteúdos em redes baseados em cópias de informações disponibilizadas por servidores de conteúdos permanentes, ou cópias temporariamente disponíveis.

O encaminhamento adaptativo entre múltiplas topologias de redes é discutido em (CHENNAKESHAVULU e PRASANTH, 2013), com recursos de monitoramento de redes, otimização da topologia e recursos de controle e adaptação dos tráfegos de rede.

Uma alternativa para as técnicas de Engenharia de tráfego é a utilização do contexto de forma a dar suporte ao roteamento adaptativo pois amplia as possibilidade de análises de informações relacionadas necessárias para tomada de decisão.

### **2.3.2 Roteamento adaptativo baseado em contexto**

Considerando o potencial da utilização de contexto para fornecimento de informações que tornem os sistemas mais assertivos, necessitando de menor esforço por parte dos usuários no fornecimento de informações, são apresentadas algumas iniciativas para utilização de contexto para tomada de decisão no encaminhamento de dados na rede.

Os trabalhos são apresentados em ordem cronológica para permitir o acompanhamento de algumas iniciativas e seu foco com o passar do tempo. Sendo um dos trabalhos iniciais nesta área, em (MUSOLESI et al., 2005) o contexto foi utilizado para a tomada de decisão no encaminhamento de redes móveis sem uso de equipamentos centralizadores de conectividade, *Ad Hoc*. As métricas consideraram a conectividade do nó e a situação da bateria. O algoritmo conseguiu aumentar a escalabilidade do ambiente, com diminuição das cópias mensagens sendo transmitidas, e menos uso das memórias de transmissão, *buffers*.

Em (MASCOLO et al., 2006) é apresentado um algoritmo de roteamento adaptativo para redes de sensores sem fio que avalia o estado atual de seus parceiros de comunicação e

prioriza a transmissão para os que tiverem mais parceiros disponíveis para encaminhar a informação, e com melhor situação de bateria. Entre os resultados, diminuição do atraso para chegada dos dados e maior tolerância a falhas.

Também considerando o estado das baterias, em (WENNING et al., 2009) a situação do nó é avaliada de forma global, inclusive a qualidade dos sinais sem fio captados e a quantidade de saltos necessários para chegar ao destino. O estudo focou em redes de sensores sem fio, mas o arcabouço proposto permite generalização para outros cenários. Foram obtidos melhor qualidade dos enlaces usados para comunicação e menos consumo de energia.

O arcabouço proposto em (YASAR et al., 2010) focou em redes móveis com parceiros ponto a ponto. Tinha como métricas o posicionamento do nó, sua direção de movimento, tempo de atividade e interesses envolvidos. Conseguiu diminuir os problemas de confiabilidade nos nós parceiros e uma melhor disponibilidade do ambiente.

Em (PETZ et al., 2012) foi proposto um arcabouço para redes sem fio que avaliava métricas relacionadas a taxas de transmissão alcançadas, contadores de pesos das taxas e equilíbrio dos vizinhos conectados. Alcançou menos latência de rede e menos sobrecarga de processamento no ambiente.

Em (MU et al., 2014) é destacado o fato das ligações em redes ZigBee serem fixadas durante a inicialização ou quanto os caminhos são definidos, e propõe uma otimização adaptativa do roteamento para gerar balanceamento do consumo de energia entre os nós parceiros, evitando a diminuição do tempo de vida dos equipamentos com menor capacidade energética.

É possível observar o foco dos trabalhos em redes sem fio, com uso de informações de QoD para melhorar seu desempenho de comunicação (QoS), e considerando uma melhor experiência para os usuários (QoE).

## 2.4 CONCLUSÃO

Foram apresentados neste capítulo definições que são utilizadas neste trabalho, com destaque para Qualidade de Serviço, Qualidade de Dispositivo, Qualidade de Experiência, Contexto e Roteamento Adaptativo.

Trabalhos correlatos que utilizam estes temas também foram apresentados. Alguns destes trabalhos servem como comparação para o trabalho em desenvolvimento.

## 3 ARQUITETURA BASEADA EM CONTEXTO

### 3.1 INTRODUÇÃO

Neste capítulo é apresentado o *Context-aware Adaptive Routing Framework (CAARF)*, arcabouço de roteamento adaptativo baseado em contexto, seus módulos, o modelo de contexto que norteia sua estrutura de dados, uma abordagem operacional de toda a estrutura, especificamente, o módulo de gestão de contexto é detalhado.

Conforme apresentado no capítulo 2, o roteamento tradicionalmente é baseado em técnicas de engenharia de tráfego e parâmetros de qualidade de serviço para realizar a tomada de decisão. Pelo contrário, o roteamento proporcionado pelo CAARF foge deste paradigma tradicional e se baseia em parâmetros de contexto, de qualidade de experiência, qualidade de dispositivo, levando também em consideração os parâmetros de qualidade de serviço.

O CAARF também foge do uso tradicional do contexto em redes. Como apresentado no capítulo 2, o uso de contexto em sistemas computacionais esta focado nas aplicações, e quando seu uso chega até as redes de comunicação de dados, foca nas redes oportunistas, ou redes ad hoc ou em redes sem um nó em particular fazendo algum papel de controle sobre a rede. De forma contrária, o CAARF está baseado em redes estruturadas, com equipamentos realizando funções específicas de comutação ou roteamento.

### 3.2 CAARF: CONTEXT-AWARE ADAPTIVE ROUTING FRAMEWORK

O *Context-aware Adaptive Routing Framework (CAARF)* foi projetado para funcionar como um *middleware* para a tomada de decisão na rede, aderente ao paradigma das redes definidas por software (*Software Defined Networks - SDN*) (RFC7426, 2015). O seu funcionamento deve ser transparente, sendo possível a rede funcionar com ou sem seu uso, sendo que os fluxos tratados podem se beneficiar de seu encaminhamento aprimorado baseado no contexto.

A ideia é que uma vez definido um domínio de rede, tipicamente equipamentos desse domínio possam ser configurados com seu plano de roteamento tradicional, por exemplo, usando técnicas de rotas padrões ou roteamento dinâmico baseado em vetor distância. Em um segundo momento, esses equipamentos podem ser reconfigurados dinamicamente para



atender a características específicas baseadas no contexto atual dos usuários, das aplicações ou dos dispositivos envolvidos em comunicações específicas monitoradas pela implementação do sistema baseado em contexto.

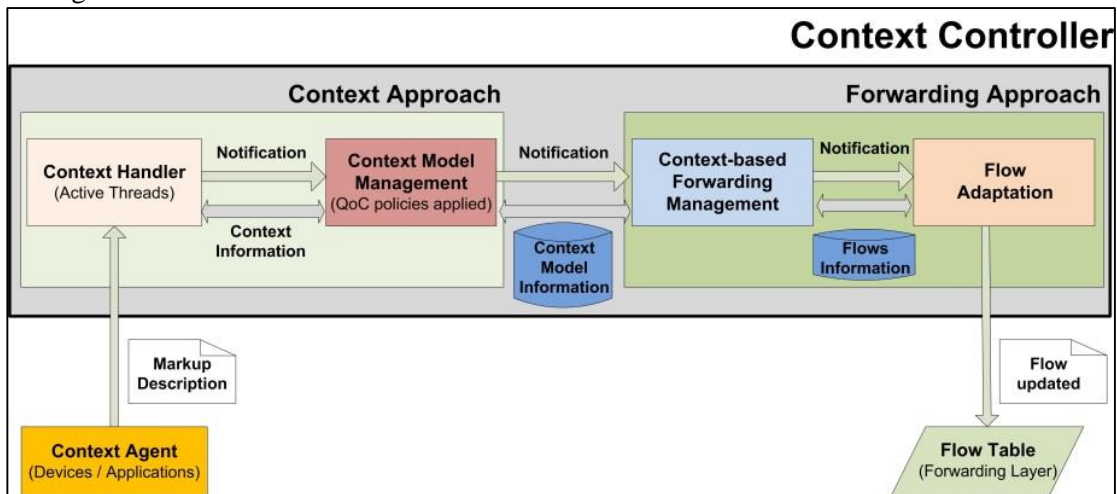
Para viabilizar essa reconfiguração, são necessários:

- Elementos que colem dados dos dispositivos de rede, das aplicações em execução e dos usuários que estão fazendo uso destas;
- Elementos que centralizem e tratem dados de contexto, disponibilizando apenas os relevantes para análise;
- Elementos que realizem a tomada de decisão quando ao encaminhamento baseado no contexto, e;
- Elementos que apliquem as novas regras definidas nos dispositivos de comutação e roteamento.

Considerando tais necessidades, o arcabouço CAARF estrutura os seguintes componentes:

- ***Context Agent***: agente de contexto responsável pela coleta de informações de contexto e envio destas para o armazenamento centralizado;
- ***Context Approach***: módulo responsável pelos dados de contexto, realizando a recepção, centralização, tratamento e disponibilização das informações de contexto;
- ***Forwarding Approach***: módulo responsável pelo encaminhamento baseado nas informações de contexto, e;
- ***Flow Table***: regras de encaminhamento resultantes aplicadas nos dispositivos de rede.

O arcabouço e suas relações são ilustrados na Figura 8:

Figura 8 - *Context Controller*

A nomenclatura dos módulos de gestão de contexto (*Context Approach*) e de gestão de encaminhamento (*Forwarding Approach*), assim como dos seus respectivos submódulos, foi definida na língua inglesa por questões de generalização e utilização nas publicações científicas. Por essa razão, esses termos são utilizados em inglês nesta dissertação.

Na Figura 8 pode-se observar o *Context Controller*, que abriga os módulos responsáveis pela abordagem de contexto e abordagem de encaminhamento, *Context Approach* e *Forwarding Approach* respectivamente. O *Context Agent* localizado fora do controle e responsável pelo envio de informações via arquivos em linguagem de marcação de descrições, *Markup Descriptions*, e o *Flow Table* que recebe os fluxos atualizados, *Flow updated*.

Tanto a abordagem de contexto quanto a abordagem de encaminhamento tem submódulos definidos para especializar funções internas, separando atividades de complexidades distintas, permitindo maior modularização de suas definições e flexibilizando implementações futuras. São submódulos do *Context Approach*:

- **Context Handler:** onde é realizado o tratamento das informações de contexto:
  - Recebe as informações do *Context Agent*;
  - Verifica se o arquivo recebido está estruturalmente integro, seguindo a estrutura de marcação adequada, isto ainda sem checar as informações contidas em seus descritores;
  - Realiza a inserção das informações no banco de dados do *Context Model Management*, e;

- Notifica o *Context Model Management* sobre a existência de novas informações para processamento.
- ***Context Model Management***: o gerenciamento do modelo de contexto, que:
  - Centraliza as informações de contexto;
  - Aplica as políticas de qualidade de contexto;
  - Disponibiliza as informações de contexto tratadas através de sua base de informações do modelo de contexto, *Context Model View*, e;
  - Notifica o *Context-based Forwarding Management* sobre a existência de novas informações para processamento.

Já os submódulos de *Forwarding Approach* são:

- ***Context-based Forwarding Management***: o gerenciamento de encaminhamento baseado em contexto, que:
  - Cria regras de encaminhamento baseadas em contexto;
  - Realiza a inserção destas regras na base de informações de fluxo, *Flow Information*, e;
  - Notifica o *Flow Adaptation* sobre a existência de novas regras a serem aplicadas.
- ***Flow Adaptation***: o adaptador de fluxo, que:
  - Carrega as novas informações de fluxo existentes no *Flow Information*, e;
  - Aplica as novas regras de encaminhamento nos dispositivos de encaminhamento e comutação da rede.

Representando o ponto de troca de informações entre as duas abordagens, o modelo de informações de contexto, ***Context Model View***, que é um repositório de informações de contexto tratadas e acessível para tomada de decisão de encaminhamento. Este repositório representa as informações conforme o modelo de contexto que será apresentado a seguir.

Nas próximas subseções deste trabalho são apresentados o modelo de dados, requisitos e especificações que contribuirão ao arcabouço apresentado.

### 3.3 MODELO DE CONTEXTO

O modelo de contexto, *Context Model*, consiste na estruturação das informações de contexto a serem utilizadas pelo arcabouço, definindo: seus parâmetros de representação, grupos de parâmetros de qualidade existentes e políticas de validação a serem aplicadas.

Conforme definido no capítulo 2, as informações de contexto são utilizadas para caracterizar uma entidade qualquer, podendo ser aplicadas a pessoas, lugares, objetos, ou qualquer outro tipo de entidade. Desta forma, esta definição é estendida neste trabalho para caracterizar elementos de uma rede de computadores, seus dispositivos de rede, suas aplicações de rede e seus usuários.

A caracterização das informações de contexto segue a estruturação das cinco categorias fundamentais definidas por (ZIMMERMANN et al, 2007), apresentadas no capítulo 2, tendo suas definições estendidas neste trabalho já considerando o arcabouço CAARF:

- ***Individuality Context***: individualidade do contexto, representando informações que permitam identificar de forma única um equipamento de rede, usuário ou assinatura de serviço sendo utilizada por uma aplicação de rede;
- ***Time Context***: tempo do contexto, representando informações temporais que permitam avaliar quando uma informação de contexto foi gerada ou, pelo menos, quando foi coletada pelo arcabouço;
- ***Location Context***: localização do contexto, representando elementos que permitam identificar a localização da entidade, um dispositivo de rede ou de um usuário, elementos tais como a rede em que se encontra, se esta próximo a algum equipamento específico, ou mesmo coordenadas de posicionamento global (GPS);
- ***Activity Context***: atividade do contexto, representando as atividades sendo realizadas, tais como aplicações de rede em execução;
- ***Relations Context***: relações do contexto, representando as relações entre as entidades do contexto, tais como a conexão de uma aplicação de rede a um *host* remoto.

São ainda partes do modelo de contexto as representações das informações de QoS, QoD e QoE, apresentadas no capítulo 2, tendo suas definições estendidas neste trabalho já considerando o arcabouço CAARF:

- **Qualidade de Serviço (QoS):** informações sobre os requisitos das aplicações de rede para funcionamento adequado, informações sobre as atuais conexões existentes entre as aplicações de redes e outros dispositivos de rede, e informações sobre os dispositivos de comutação e roteamento;
- **Qualidade de Dispositivo (QoD):** informações que caracterizem os dispositivos que fazem uso da rede, capacidade computacional dos dispositivos de rede, e níveis de precisão dos sensores;
- **Qualidade de Experiência (QoE):** informações coletadas ou calculadas que possam avaliar a percepção do usuário de rede sobre sua experiência de uso da própria rede e das aplicações de rede, grau de satisfação, e fatores correlatos a experiência do usuário como fatores que possam alterar sua própria percepção tais como ruído no ambiente no qual realiza camadas de voz (VoIP).

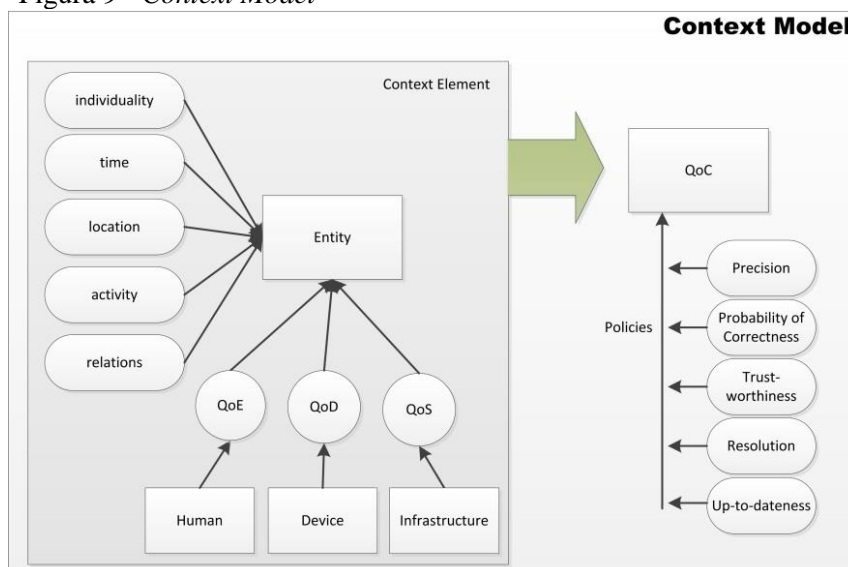
Por fim, considerando as diferentes fontes de informações de contexto, é importante que o modelo de contexto e as informações de contexto possam ser validadas por políticas de **Qualidade de Contexto (QoC)**, conforme definição original de (BUCHHOLZ; SCHIFFERS, 2003), apresentada no capítulo 2, e estendidas neste trabalho já considerando o arcabouço CAARF:

- **Precisão:** política que avalia o nível de precisão da informação de contexto, podendo ignorar informações com níveis fora do estabelecido, por exemplo, podendo ignorar informações provenientes de um GPS com nível de precisão abaixo do patamar especificado;
- **Probabilidade de acerto:** política que avalia a probabilidade de uma informação estar correta, podendo ignorar informações fora de faixas possíveis, por exemplo, podendo ignorar informações sobre velocidades de transmissão acima da velocidade máxima da interface de conexão que esta sendo utilizada;

- **Confiabilidade:** política que avalia se a fonte é confiável para fornecer informações, por exemplo, podendo ignorar informações coletadas de um agente instável que se conecta sucessivas vezes reportando localizações distintas em um intervalo curto de tempo;
- **Resolução:** política que avalia a granularidade da informação coletada e o uso que se deseja dar a estas informações, por exemplo, podendo ignorar dados com granularidade muito grande e focando em dados com menos granularidades, mais relevantes para o uso pretendido;
- **Atualização:** política que avalia se a informação esta atualizada, por exemplo, podendo ignorar dados de uma comunicação que não é atualizada dentro do período esperado, podendo indicar que o agente não esta mais reportando por alguma falha, não sendo mais a informação fornecidas por este agente relevantes para análise.

Desta forma o modelo de contexto, *Context Model*, definido para o arcabouço CAARF considera os elementos de contexto com as definições das entidades de contexto, as características de qualidades específicas dos usuários, dispositivos e infraestrutura, e realiza a validação destas informações com as políticas de qualidade de contexto, conforme representado na Figura 9:

Figura 9 - *Context Model*

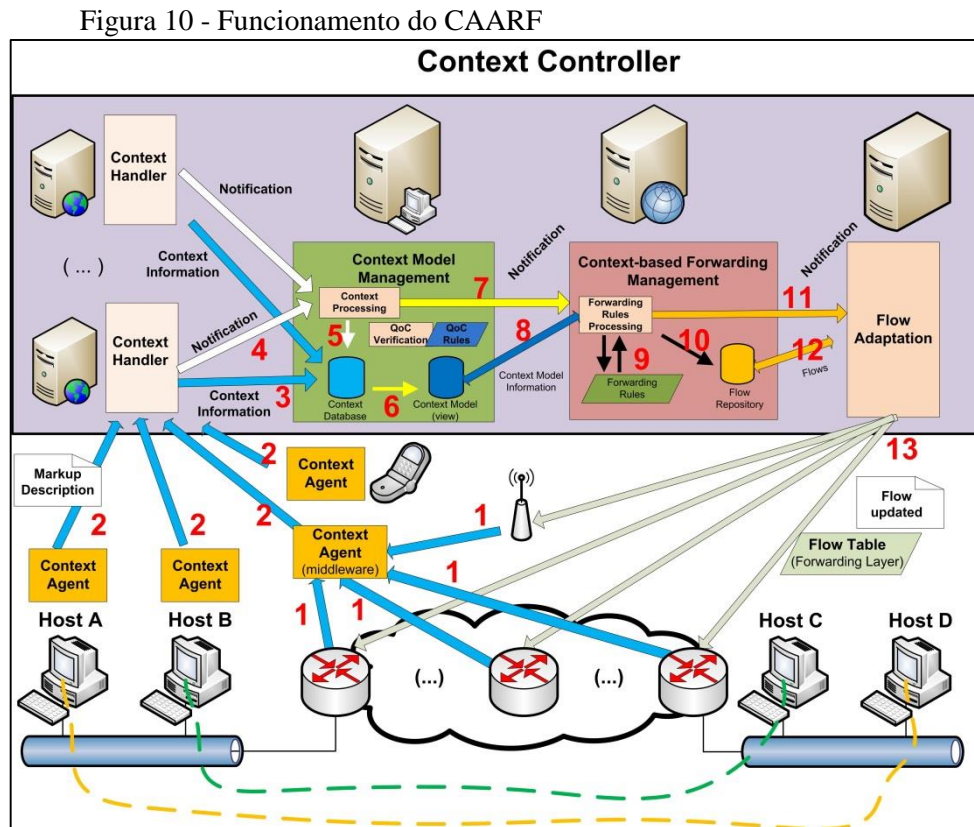


Conforme ilustrado na Figura 9, o arcabouço CAARF pode utilizar informações provenientes de diversas fontes, devidamente relacionadas entre si, e com filtros para

viabilizar que apenas informações realmente úteis sejam utilizadas para tomada de decisão de encaminhamento, como será mostrado mais adiante neste trabalho.

### 3.4 CAARF: ABORDAGEM OPERACIONAL

Visando facilitar o entendimento do funcionamento da implementação do arcabouço CAARF, a seguir é apresentado um cenário exemplo fictício com quatro estações (*hosts*) nomeados de “*Host A*” até “*Host D*”, sendo que os dois primeiros estão em uma rede local (LAN) e os dois últimos em outra LAN, estando estas LANs interconectadas por uma rede privada gerida pela organização, conforme ilustrado na Figura 10.



Neste cenário, ilustrado na Figura 10, o *Host A* está se comunicando com o *Host C*, e o *Host B* está se comunicando com o *Host D*, ambos utilizando aplicações de rede suportadas pelo *middleware* baseado no CAARF.

Então, os *Hosts* A e B, assim como os equipamentos responsáveis pelo encaminhamento, estão reportando informações de contexto para o *middleware*, sendo que (1) as informações dos encaminhadores são coletadas pelo *Context Agent Middleware* que pode estar em execução em um *host* ou em um equipamento dedicado, por exemplo, um *appliance*, enquanto (2) os *hosts* reportam diretamente através do *Context Agent* em execução em seus sistemas operacionais. Estas informações são estruturadas em arquivos *Markup Description*, por exemplo, arquivos XML (XML, 2015) ou JSON (JSON, 2015), e são transmitidas para o serviço Web, *webservice*, do *Context Handler*. Enquanto nos *hosts* A e B são coletadas informações do próprio *host*, do usuário e da aplicação de rede em execução, nos encaminhadores são coletadas informações sobre o estado das conexões, capacidade computacional, entre outras.

No *ContextHandler* estas informações são recebidas, é verificada a consistência da estrutura do arquivo *Markup Description* recebido, e se o arquivo estiver integro, será processado pelo submódulo (3) *Buffer Management* que realiza a inserção das informações diretamente na base *Context Database* do *Context Model Management*. E ainda no *ContextHandler* o submódulo (4) *Notification Scheduling* notifica a chegada de novas informações de contexto para o *Context Model Management*.

Estando as informações já na base do *Context Model Management*, seu submódulo (5) *Context Processing* recebe notificações de novas informações de contexto, recupera informações de contexto do *Context Database* e as submete ao submódulo *QoC Verification*, que irá aplicar as políticas de QoC pré-definidas pelo administrador da rede. Sendo que, após validação das políticas de QoC, o (6) *Context Processing* registra o novo contexto e o torna disponível através de uma visão do banco, a *Context Model View*, e o (7) *Context Processing* notifica a modificação do contexto para o módulo *Context-based Forwarding Management*.

No *Context-based Forwarding Management*, o submódulo (8) *Forwarding Rules Processing* recebe as notificações de novo contexto e realiza consultar na base *Context Model View*, e, através do submódulo (9) *Forwarding Rules Verification* aplica as regras de encaminhamento pré-definidas pelo administrador da rede. Logo após este processamento, o (10) *Forwarding Rules Processing* registra os fluxos atualizados no *Flow Repository* e (11) notifica o *Flow Adaptation* da atualização de novas regras de fluxo a serem aplicadas nos encaminhadores.



Quando o (12) *Flow Adaptation* consulta as novas regras de fluxo no *Flow Repository* através de sua visão de regras ativas, a *Current Flow View*, realiza o escalonamento para (13) aplicar as novas regras de fluxo contidas no *Current Flow View* nos encaminhadores da rede.

Desta forma, o *middleware* coleta informações de contexto, valida, processa o encaminhamento baseado nas informações de contexto, e reconfigura os encaminhadores para repercutir estas atualizações nos fluxos.

Após a aplicação de rede encerrar sua comunicação ou se o agente para de reportar seu estado por um tempo predeterminado, o *Context Model Management* realiza a remoção destas informações do *Context Model*, por conseguinte, o *Context-based Forwarding Management* vai remover estas regras de encaminhamento do *Flow Repository*, causando a sua remoção dos encaminhadores pelo *Flow Adaptation*.

Na próxima seção é detalhado o funcionamento da abordagem de encaminhamento e das políticas de QoS.

### 3.5 GESTÃO DE CONTEXTO

Para o funcionamento adequado do *middleware*, a coleta de informações é um ponto chave, precisando ser realizada em todos os *hosts* que se deseje ter o tratamento diferenciado da rede baseada em seus contextos. Para tanto, o *Context Agent* precisa estar em execução em seus sistemas operacionais e reportando suas informações para o ambiente. Neste trabalho são abstraídos os detalhes da implementação do agente, consideram-se apenas os eventos internos destes agentes que geram envios de mensagens para o *Handler* e os dados enviados por este. Considera-se ainda que a abordagem de contexto finaliza suas funcionalidades ao entregar os dados de contexto coletados, centralizados, devidamente verificados pelas políticas de QoS e estruturados no *Context Model View*, sendo abstraídos os detalhes da implementação da abordagem de encaminhamento.

Vale destacar que os módulos do agente, *Context Agent*, e da abordagem de encaminhamento, *Forwarding Approach*, estão em andamento em trabalhos de Mestrado de outros pesquisadores, estendendo o arcabouço CAARF.

Nesta subseção é apresentada a abordagem de contexto, *Context Approach*, do arcabouço CAARF, seus módulos internos de gestão de contexto, os eventos do arcabouço relacionados ao contexto, e é apresentada uma visão geral dos repositórios de contexto.

### 3.5.1 Módulos da Gestão de Contexto

Abstraindo a definição dos *Context Agent*, agentes de coleta que são executados no sistema operacional do dispositivo a ser monitorado, os considerando que estes agentes enviam arquivos com informações de contexto organizados em linguagem de marcação, *Markup Description*, a ser descrita adiante neste trabalho.

Considerando ainda que quando não for possível executar o agente diretamente no sistema a ser monitorado, agentes intermediários em execução em outros dispositivos podem ser utilizados para realizar tal coleta, os *Context Agent Middleware*.

Sabendo que, tanto os arquivos *Markup Description* enviados pelo o *Context Agent*, quanto os enviados pelo *Context Agent Middleware*, são processados da mesma forma pelo arcabouço.

O módulo de gestão de contexto visa a gestão das informações de contexto recebidas a partir do ambiente de rede. Foram definidos dois submódulos para a abordagem de contexto, o manipulador de informações coletadas e centralizador destas informações em um repositório único no ambiente, o *Context Handler*, que pode ser único no ambiente, ou podem existir outros para dividir a carga operacional e um validador das informações de contexto, que aplica os filtros de QoC definidos e gerencia o repositório centralizado de informações de contexto, o *Context Model Management*. São submódulos do *Context Handler*:

- ***WebService Notification Management:***
  - Serviço Web responsável por receber os arquivos Markup Description dos agentes e encaminha-los internamente para o Buffer Management realizar seu processamento. Considerando a escalabilidade do ambiente, deve trabalhar com múltiplas linhas de atendimento concorrentes e multithreading;
  
- ***Buffer Management:***
  - Responsável por verificar se o arquivo *Markup Description* recebido está íntegro, conforme arquivo de definição em uso pelo ambiente. Caso não esteja, ele deve ser descartado uma vez que será possível identificar que blocos de dados compõem cada informação de contexto contida nele;

- Após verificar a integridade do arquivo, deve realizar a leitura de suas informações de contexto e gravação destas diretamente no repositório de contexto existente no *Context Database*;
- Após registro no repositório, internamente deve solicitar ao *Notification Scheduling* para que notifique ao *Context Model Management* sobre a existência de novos dados a serem processados, ficando livre o *Buffer Management* para processar um novo arquivo, independente da notificação já ter sido enviada ou não.
- ***Notification Scheduling:***
  - Após solicitação interna do *Buffer Management*, o *Notification Scheduling* deve se conectar ao *Context Processing*, submódulo do *Context Model Management*, e notificar sobre a existência de novas informações de contexto a serem verificadas;
  - Em casos em que existam mensagens de notificação enfileiradas, opcionalmente pode-se realizar a priorização das notificações a serem enviadas, cabendo estudo adicional dos critérios a serem utilizados.

Especificados os componentes do manipulador que vai receber os dados dos agentes, segue a especificação dos submódulos do *Context Model Management*, responsável pelas verificações e repositório, são eles:

- ***Context Processing:***
  - Responsável pela recepção das notificações provenientes do *Notification Scheduling*, informando que existem novas informações a serem processadas;
  - Após receber a notificação, aciona o *QoC Verification* para verificar as informações de contexto;
  - Ao final da execução da verificação pelo *QoC Verification*, se existirem modificações nas informações existentes no repositório *Context Database*, notifica o *Context-based Forwarding Management* para que realize seu processamento.
- ***QoC Verification:***

- Responsável por aplicar as políticas de qualidade de contexto (QoC), pré-definidas pelo administrador de rede;
  - Após aplicar as políticas, retorna ao *Context Processing* informando se existiram ou não modificações no repositório *Context Model View*.
- **QoC Rules:**
    - Conjunto de regras de QoC pré-definidas pelo administrador de rede, e que serão utilizadas pelo *QoC Verification*;
    - Pode conter regras genéricas, validando informações genéricas usáveis por qualquer tipo de encaminhamento de rede, mas considera-se que sua eficiência será maior se forem utilizadas regras específicas para o tipo de rede e de análise de encaminhamento que vai fazer uso de suas informações resultantes. Regras de QoC são discutidas adiante deste trabalho.
- **Context Database:**
    - Repositório responsável por armazenar as informações de contexto;
    - Recebe inserções de informações de contexto diretamente do *Buffer Management* existente no *Context Handler*;
    - Após análise do *QoC Verification*, as informações de contexto validadas pelas políticas de QoC passam a fazer parte de uma visão deste repositório. Este repositório virtual resultante passa a ser identificado no arcabouço como *Context Model View*, e é este que é acessível pelo *Context-based Forwarding Management* para realizar suas análises de encaminhamento baseado em contexto;
      - O uso de uma visão das informações já existente no repositório minimiza as interações de inclusão de dados, ficando estas restritas a marcadores que identificam se uma informação foi ou não aprovada nas políticas de QoC e marcadores que identificam se um conjunto de informações de contexto vai ou não constar no *Context Model View*;
      - Operações de análise dos arquivos *Markup Description* e de inclusão de suas informações em repositório são executadas pelo *Context Handler*, diminuindo a carga do *Context Model Management*.

Com estas especificações, o *Context Handler* e o *Context Model Management* executam a gestão de informações na abordagem de contexto, sendo suas notificações internas apresentadas na próxima seção.

### 3.5.2 Eventos da Gestão de Contexto

Segundo (PRESSMAN, 2006) , em geral, os eventos ocorrem sempre que um sistema ou um ator trocam informações, sendo o evento o fato de trocar informações, e não as informações trocadas.

Para o CAARF, alguns eventos ocorrem e são tratados internamente em um mesmo componente do arcabouço, enquanto outros podem desencadear ações em outros componentes, conforme descrito a seguir.

- **São eventos do *Context Agent*:**
  - **Ativação de um agente:**
    - Fato do processo do agente ser inicializado no sistema operacional a ser monitorado;
    - Realiza uma chamada para o evento de coleta de informações;
    - Envia as informações coletadas para o *Context Handler*.
  - **Início de uma relação:**
    - Fato de uma aplicação de rede iniciar uma conexão de rede com um *host* remoto, na nomenclatura do arcabouço, definindo uma relação;
    - Realiza uma chamada para o evento de coleta de informações;
    - Envia as informações coletadas para o *Context Handler*;
    - Se a aplicação for inicializada no sistema operacional, mas não realizar conexões remotas, não ativa este evento.
  - **Finalização de uma relação:**
    - Considerando que a aplicação de rede esteja sendo previamente monitorada pelo agente, envia uma notificação ao *Context Handler* informando que a relação foi finalizada.

- **Atualização das relações:**
  - Periodicamente, conforme configuração do arcabouço, o agente deve chamar o evento de coleta de informações;
  - Se as informações coletadas forem distintas das enviadas anteriormente, envia as informações atualizadas para o *Context Handler*;
  - Se as informações coletadas forem iguais às enviadas anteriormente, envia uma notificação para o *Context Handler* informando estar operando normalmente e sem alterações, isto é, uma mensagem tipo *Keep Alive*.
  
- **Coleta de informações:**
  - Se o sistema operacional suportar, coleta informações sobre o *host*, o usuário corrente, parâmetros do dispositivo e informações sobre as aplicações de rede que estejam realizando conexões remotas, isto é, relações com outros *hosts*;
  - Se existirem aplicações de rede compatíveis, coleta informações sobre a aplicação;
  - Se existir uma interface de interação direta com o usuário, coleta informações fornecidas pelo usuário nesta interface.
  
- **Interação do usuário:**
  - Considerando que existe uma interface de interação direta com o usuário no agente, e que esta foi acionada por ele;
  - Aciona o evento de coleta de informações;
  - Envia as informações coletadas para o *Context Handler*.
  
- **Desligamento do agente:**
  - Quando for acionado o encerramento do agente, este deve notificar o *Context Handler* que esta sendo encerrado, conseqüentemente não vai mais reportar ao *middleware*.
  
- **São eventos do *Context Handler*:**
  - **Ativação de um agente:**

- Verifica se o agente que esta reportando ativação tinha relações ativas no *Context Model View*, caso positivo, desabilita estas relações no repositório;
  - Caso existam informações sobre relações no evento de ativação do agente, as cadastra no repositório *Context Database*;
  - Se uma das situações acima descritas ocorrer, chama o evento de processamento do contexto.
- **Início de uma relação:**
    - Cadastra as informações da nova relação no repositório *Context Database*;
    - Chama o evento de processamento do contexto.
- **Finalização de uma relação:**
    - Consulta se a relação finalizada consta no repositório *Context Model View*, caso positivo, desabilita esta relação;
    - Chama o evento de processamento do contexto.
- **Atualização das relações:**
    - Consulta se as relações atualizadas constam no repositório *Context Database*, caso positivo, atualiza estas relações;
    - Caso não constem no repositório *Context Database*, cadastra as informações das reações;
    - Chama o evento de processamento do contexto.
- **Interação do usuário:**
    - Se a interação do usuário foi para uma aplicação de rede específica, consulta se a relação consta no repositório *Context Database*, caso positivo, atualiza as informações da interação do usuário nesta relação;
    - Se a interação não foi para uma relação específica, consulta todas as relações ativas deste usuário e atualiza em todas as informações da interação do usuário;

- Caso não constem relações no repositório *Context Database*, cadastra as informações das relações reportadas e da interação do usuário em todas elas;
  - Aciona o evento de processamento do contexto.
- **Desligamento do agente:**
    - Consulta no repositório *Context Model View* se existem relações ativas para o agente sendo finalizado, caso existam, desativa todas.
- **Processamento do Contexto:**
    - Notifica ao *Context Model Management* a necessidade de processar as alterações do contexto existentes no *Context Database*.
- **Context Agent sem atualizar:**
    - Considerando que o *Context Handler* deve manter um controle de quando foi a ultima vez que um agente enviou informações ou notificações para ele;
    - Se este intervalo de tempo for maior que o pré-definido pelo sistema, deve chamar o evento Desligamento do agente, removendo assim as informações de todas as relações deste agente, desta forma, evitando que informações possivelmente desatualizadas sejam utilizadas.
- **Atualização do Context Handler:**
    - Caso o *Context Handler* não tenha chamado o evento de Processamento do Contexto há um intervalo de tempo maior que o pré-definido pelo sistema, conseqüentemente estando sem demonstrar ao *Context Model Management* que está ativo, o *Context Handler* deve enviar uma mensagem informando estar operando normalmente e sem alterações, isto é, uma mensagem tipo *Keep Alive*.

Como apresentado nos eventos do *Context Handler*, quase todos os eventos foram originados no *Context Agent*, apenas o evento Coleta de informações que se originou e encerrou no próprio agente. Além desta diferença, foram inseridos três novos eventos no



*Context Handler*, os eventos Processamento do Contexto, *Context Agent* sem atualizar e Atualização do *Context Handler*.

- **São eventos do *Context Model Management*:**
  - **Processamento do Contexto:**
    - Executa o submódulo *QoC Verification* do *Context Model Management* para checar se existem novas informações no repositório *Context Database* para aplicar as políticas do *QoC Rules*;
    - Verifica se o *Context Model View* foi alterado desde a última execução, caso positivo, chama o evento notifica novo contexto.
  - **Atualização do *Context Handler*:**
    - Considerando que o *Context Model Management* deve manter um controle de quando foi a última vez que um *Context Handler* enviou informações ou notificações para ele, se este intervalo de tempo for maior que o pré-definido pelo sistema, deve desabilitar do repositório *Context Model View* todas as relações provenientes deste *Context Handler*, desta forma, evitando que informações possivelmente desatualizadas sejam utilizadas.
  - **Notifica novo contexto:**
    - Notifica ao *Context-based Forwarding Management* sobre a existência de um novo contexto para processamento.

Como apresentado nos eventos do *Context Model Management*, apenas dois eventos vieram do *Context Handler*, os eventos Processamento do Contexto e Atualização do *Context Handler*, os demais se encerraram no próprio *Context Handler*, e introduziu um novo evento denominado Notifica novo contexto.

Como apresentado anteriormente, o foco deste trabalho está na gestão do contexto, entretanto, para definir os eventos desta gestão foi necessário especificar os eventos do *Context Agent* devido à sua influência direta nos eventos aqui relacionados, cabendo a outros pesquisadores dar ênfase ao agente, fazendo correções e melhorias deste trabalho.

Estando os eventos definidos, na próxima seção é feita uma apresentação de exemplos de informações coletáveis para manipulação pela abordagem de contexto.

### 3.5.3 Exemplos de informações coletáveis

Nesta seção são apresentados exemplos de informações de contexto que podem ser coletadas pelos agentes e posteriormente utilizadas pelo arcabouço para tomada de decisão de encaminhamento. Vale destacar que a definição de quais informações são mais relevantes para coleta, e se na coleta cabe ou não criar interdependência entre algumas destas informações, é um estudo que deve ser aprofundado em trabalhos futuros, sendo as coletas propostas neste trabalho utilizadas para demonstração do arcabouço, sendo apenas uma primeira discussão deste tema.

Outro ponto que deve ser destacado é a relação das informações com o tipo de rede, por exemplo, em alguns tipos de redes *Openflow* (ONF, 2015) pode não ser relevante coletar dados de endereços IP, ou portas de comunicação, ou outros dados específicos que podem inclusive não estar disponíveis. Uma abordagem pode ser a coleta global de dados e tratar estas diferenças dentro do arcabouço, mas isso também pode ter um custo computacional relevante que deve ser avaliado.

As informações coletáveis apresentadas a seguir estão agrupadas conforme estruturação do modelo de dados discutido anteriormente, sendo elas:

- Informações que individualizam a entidade:
  - Usuário;
  - Nome do equipamento;
  - Domínio do equipamento;
  - Endereço físico da interface de rede padrão do equipamento;
  - Endereço lógico da interface de rede padrão do equipamento, e;
  - Assinaturas de serviços associadas ao usuário.
- Informações que auxiliem na identificação temporal:
  - Hora local do equipamento;
  - Uso ou não de ajustes de tempo, e;
  - Última atualização de tempo.
- Informações que auxiliem na localização:

- Coordenadas GPS;
- Endereço de rede do equipamento;
- Cidade relatada, e;
- Locais próximos relatados.
- Informações que descrevam as atividades em curso:
  - Identificação das relações ativas, e;
  - Correlação das relações ativas com assinaturas de serviços existentes.
- Informações que detalhem as relações (conexões remotas de rede):
  - Identificadores únicos no agente das relações monitoradas;
  - Nomes informados pelas aplicações;
  - Nomes nos executáveis das aplicações;
  - Versões dos executáveis das aplicações;
  - Endereços IPs dos equipamentos remotos;
  - Portas origem das conexões;
  - Portas remotas das conexões;
  - Protocolos de transporte utilizados nas conexões, e;
  - Protocolos de aplicação utilizados nas conexões.
- Informações sobre o QoE:
  - Identificação das relações que tem dados de QoE disponíveis;
  - Média dos escores de opinião calculados (*R-Factor*);
  - Média dos escores de opinião informados (MOS), e;
  - Nível de ruído no ambiente.
- Informações sobre QoD:
  - Uso ou não de dados GPS;
  - Nível de precisão do GPS;
  - Quantidade de processadores do equipamento;
  - Velocidade dos processadores do equipamento;
  - Quantidade de memória do equipamento;
  - Identificadores únicos no agente das interfaces de rede;
  - Velocidade das interfaces de rede;
  - Endereços físicos das interfaces de rede;
  - Endereços lógicos das interfaces de rede, e;
  - Suporte ou não a descarregar de processamento da pilha TCP/IP (TOE).

- Informações sobre QoS:
  - Identificação das interfaces de rede que tem dados de QoS disponíveis;
  - Média de vazão de cada interface;
  - Perda de pacotes das interfaces;
  - Média dos atrasos de cada interface, e;
  - Média das variações de atraso de cada interface.

Destas informações relacionadas, nem todas podem estar disponíveis em todos os equipamentos para coleta, podendo algumas delas existir apenas em equipamentos específicos, por exemplo, os dados de QoS possivelmente só serão coletados de encaminhadores de rede e não de equipamentos de usuários.

Como mencionado anteriormente, parte destas informações só faz sentido quando agrupadas, por exemplo, endereço IP, porta de origem e porta de destino de uma conexão de rede. A forma como são estruturados estes agrupamentos é apresentada mais adiante neste trabalho.

Já tendo definidas as informações de contexto que exemplificarão o funcionamento do arcabouço, no próximo tópico são exemplificadas as validações de QoC a serem aplicadas sobre estas coletas.

### **3.5.4 Regras de QoC**

São apresentadas aqui exemplos de regras de QoC utilizadas no arcabouço CAARF que, conforme apresentado no capítulo 2, tem por objetivo verificar as informações contextuais antes de serem utilizadas na tomada de decisão dos sistemas sensíveis ao contexto, permitindo por exemplo, ignorar informações com valores impossíveis ou incoerentes se comparados com outras informações de contexto relevantes.

Os exemplos de regras de QoC utilizadas para demonstração do arcabouço podem ser vistas na Tabela 2 a seguir (As demais regras de QoC definidas são apresentadas no Anexo II):

Tabela 2 - Regras QoC

Nº	Regra	Motivo	Ação
1	Verifica se foi especificado um endereço lógico (IP) válido.	Indispensável para construção posterior de regras de encaminhamento.	Descarta todas as informações deste agente.
2	Verifica se foi especificado um endereço físico (MAC) válido.	Opcionalmente necessário para regras de encaminhamento.	Se for inválido, descarta; Se não foi especificado, ignora.
3	Verifica se foi especificado um endereço de domínio válido.	Opcionalmente necessário para regras de encaminhamento.	Se for inválido, descarta; Se não foi especificado, ignora.
4	Verifica se existem informações temporais.	Necessário para verificar atualização das informações.	Se existirem, verifica se estão dentro da tolerância pré-definida no arcabouço; Se não foi especificado, ignora.
5	Verifica se existe informação de servidor de tempo (NTP).	Indica maior confiabilidade das informações temporais.	Se existir, verifica regra 6; Se não existir, ignora.
6	Acionada se passar na regra 5; Verifica se existe informação de atualização de tempo (NTP).	Indica quando foi a última atualização de tempo.	Se existir, verifica se esta dentro da tolerância pré-definida no arcabouço, estando, ignora; Não estando, destaca as informações deste agente; Se não existir, ignora.
7	Verifica se foi especificado endereço lógico (IP) válido do equipamento remoto de uma relação.	Desejável para construção posterior de regras de encaminhamento.	Se existir, ignora; Se não existir ou não for válido, descarta esta informação.
8	Verifica se foi especificado endereço de porta local válida na relação.	Desejável para construção posterior de regras de encaminhamento.	Se existir, ignora; Se não existir ou não for válido, descarta esta informação.
9	Verifica se foi especificado endereço de porta remota válida na relação.	Desejável para construção posterior de regras de encaminhamento.	Se existir, ignora; Se não existir ou não for válido, descarta esta informação.
10	Verifica se foi especificado protocolo de transporte válido na relação.	Desejável para construção posterior de regras de encaminhamento.	Se existir, ignora; Se não existir ou não for válido, descarta esta informação.
11	Verifica se foi especificado protocolo de aplicação válido na relação.	Desejável para construção posterior de regras de encaminhamento.	Se existir, ignora; Se não existir ou não for válido, descarta esta informação.
12	Verifica se a vazão da interface é superior à largura de banda da interface.	Indica coleta de dados incorreta.	Ignora esta informação.

Como pode ser observado na Tabela 2, as regras de 1 até 3 focam questões de individualização da entidade, as regras de 4 até 6 focam em questões ligadas a gestão dos

tempos, de 7 até 11 focam nas relações, a regra 12 trata de um parâmetro de QoS e sua validação para ser coerente para uso posterior no QoS. Estes agrupamentos das regras de QoS discutidos ilustram o relacionamento entre as regras e a característica sendo avaliada da entidade.

Não fazem parte das regras relacionadas, as validações quanto ao formato das informações coletadas, por exemplo, se o endereço IP segue o formato IPv4 ou IPv6 correto. Algumas destas validações podem ser realizadas pelas definições das estruturas dos arquivos *Markup Descriptions*, por exemplo, se forem utilizados arquivos XML (XML, 2015), pode-se utilizar as definições de esquema, arquivos XSD (XSD, 2015). Se estas validações não forem realizadas nos arquivos *Markup Descriptions*, devem ser feitas pelas regras de QoS. No entanto, o endereço de difusão (*broadcast*) 255.255.255.0, tem o formato IPv4 correto, embora não seja um endereço válido para identificar um equipamento da rede.

Deve-se notar que mesmo tendo, as informações de contexto e as regras de QoS, relação com o tipo de rede e o tipo de encaminhamento que esta sendo utilizado no arcabouço, o foco destas regras QoS não é tomar decisões quando ao encaminhamento, sendo este papel função exclusiva da abordagem de encaminhamento, a função da abordagem de contexto é entregar informações validadas por critérios que possam torná-las possivelmente mais úteis.

### 3.6 CONCLUSÃO

Neste capítulo foi apresentado o *Context-aware Adaptive Routing Framework* e seu posicionamento como um nível adicional na tomada de decisão no encaminhamento em redes de computadores baseando suas escolhas não só nos critérios tradicionais de QoS, mas também em informações de contexto coletadas dos dispositivos, sistemas e usuários que fazem uso da rede, sendo, portanto, aderente aos princípios das redes SDN.

Foram apresentadas ainda descrições gerais do *ContextAgent*, da abordagem de contexto, foco deste trabalho com o *Context Handler* o *Context Model Management*, e descrições gerais da abordagem de encaminhamento. Quanto a abordagem de contexto, foram descritos seus requisitos, módulos internos de gestão, eventos e exemplos de informações de contexto e de regras de QoS.

Através das validações realizadas é possível qualificar as informações de contexto antes de serem disponibilizadas para a abordagem de encaminhamento, ignorando

informações com valores impossíveis, ou fora de faixas configuráveis pelo administrador do arcabouço, e ainda, testando interdependências entre informações que se processadas separadamente não teriam utilidade. Desta forma, o posterior processamento de encaminhamento será realizado baseado em informações realmente úteis, diminuindo desvios nos resultados obtidos e aumentando a precisão do encaminhamento.

Nos próximos capítulos serão apresentadas as simulações e cenários utilizados para validação do arcabouço proposto.

## 4 SIMULAÇÃO: GESTÃO DE CONTEXTO

### 4.1 INTRODUÇÃO

Neste capítulo é apresentada a estruturação da simulação realizada para validação do arcabouço e da abordagem de contexto foco deste trabalho. Para isso é apresentada a modelagem do experimento realizado, mostrando suas fases intermediárias de coleta e processamento de dados, os passos para a manipulação das entradas e saídas, e como são analisados seus resultados obtidos.

Fazem parte desta apresentação, os requisitos funcionais e não funcionais, a estruturação de persistência de dados dos arquivos *Markup Description* e das bases de informações de contexto, as ferramentas e linguagens utilizadas para executar a simulação e analisar os resultados obtidos.

### 4.2 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

Conforme (PRESSMAN, 2006), os requisitos Funcionais definem as funções do sistema, entradas e saídas, processamentos que devem ser realizados, entre outras funcionalidades. Já os requisitos Não Funcionais definem questões relacionadas à usabilidade, confiabilidade, disponibilidade, desempenho, portabilidade, entre outras características. Tendo estes pontos em vista, para a gestão de contexto, que inclui os dados recebidos pelos agentes, são requisitos:

- **Requisitos Funcionais:**
  - Deve ser possível realizar a coleta de informações nos dispositivos que façam uso da rede, independente de serem estações de usuários ou dispositivos de encaminhamento;
  - As informações coletadas devem ser centralizadas no ambiente;
  - Após coleta, as informações devem ser validadas pelas políticas de QoC do ambiente;
  - Informações previamente coletados devem ser atualizados periodicamente para validar se continuam pertinentes para o ambiente;



- Quando não houver atualização periódica das informações, estas devem ser removidas no ambiente;
- As informações coletadas, centralizadas e validadas, devem estar disponíveis para consulta pela abordagem de encaminhamento;
- **Requisitos Não Funcionais:**
  - A coleta deve ser realizada por entidades fora do ambiente, não afetando as entidades que estejam sendo monitoradas;
  - Em casos onde a entidade a ser monitorada não suporte execução do coletor diretamente em seu sistema, podem ser utilizados mecanismos de coleta indireta das informações, sendo este coletor intermediário o responsável pelo envio das informações ao ambiente;
  - O envio de informações das entidades coletoras para o ambiente deve utilizar linguagem de marcação comum;
  - Deve-se evitar a duplicação de informações em repositório, com conseqüente crescimento desnecessário destes repositórios e perda de desempenho, podendo utilizar estruturas lógicas, tais como filtros ou visões das informações já validadas;
  - Informações que não façam mais parte do contexto atual podem ser indisponibilizadas para a abordagem de encaminhamento, como por exemplo relações de um dispositivo que não esteja mais ativo na rede, mas é desejável que estas informações estejam temporariamente preservadas no repositório para análises futuras do ambiente, .

Com estes requisitos definidos, é possível especificar as características internas da abordagem de contexto que serão exploradas no próximo item.

### 4.3 MODELAGEM DA SIMULAÇÃO

O experimento foi realizado em um simulador de eventos discretos, *Network Simulator* (NS2), que permite programar o momento dos eventos e suas características. Nele,

os eventos são inicializados e os resultados são gravados em arquivos de traço que podem ser posteriormente analisados (NS2, 2015).

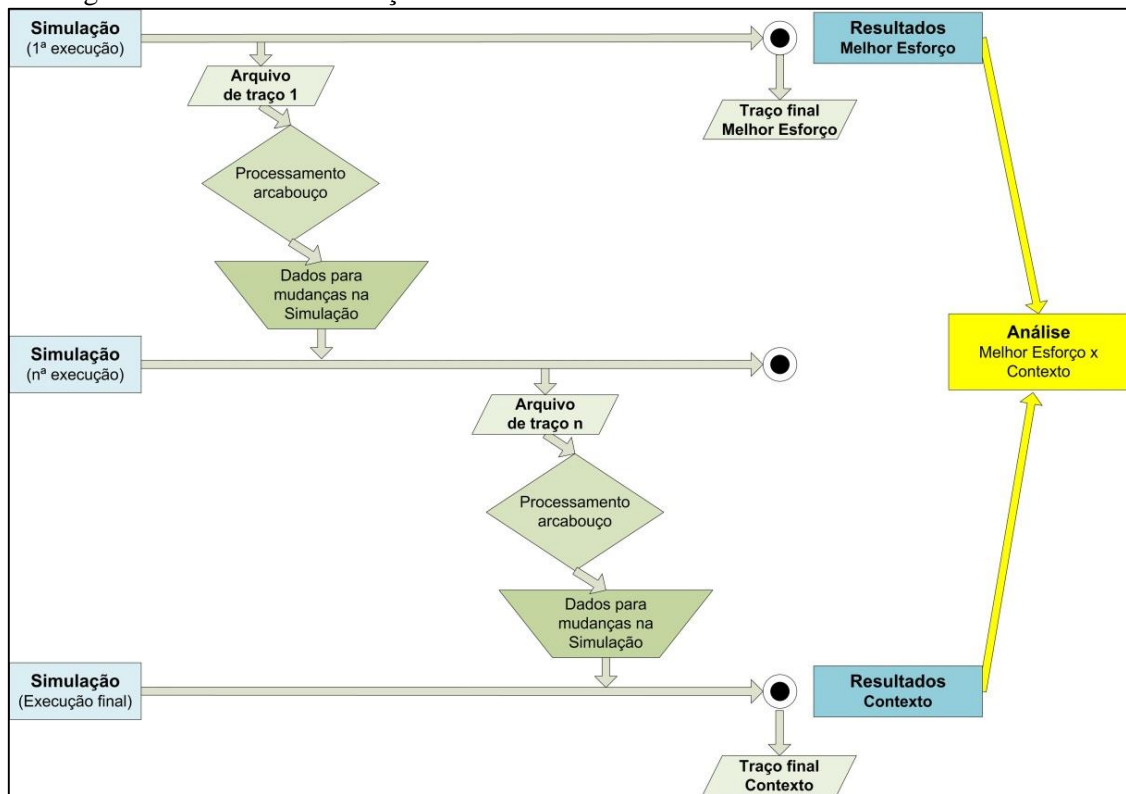
Considerando que o arcabouço deve funcionar como uma camada adicional na tomada de decisão da rede, os eventos devem iniciar sua transmissão de dados normalmente, e com a posterior coleta e processamento dos dados de contexto o arcabouço deve indicar os ajustes no encaminhamento para atender as necessidades do contexto.

Desta forma, a rede inicial da simulação é configurada para funcionar como uma rede baseada no melhor esforço, sendo que o encaminhamento (e a construção de caminhos) é realizado utilizando o algoritmo de roteamento dinâmico vetor distância, onde o menor caminho entre dois pontos é calculado considerando a menor quantidade de saltos (*hops*), entre a origem e o destino (TANENBAUM, 2003). Ao final do experimento é coletado um arquivo de traço, *log*, com todos os eventos ocorridos, incluindo, geração de pacotes, tamanho, enfileiramento para transmissão, consumo da fila para transmissão, recepção no nó adjacente, até a chegada ao nó destino, registrando ainda as perdas e os tempos decorridos nestes eventos (NS2, 2015).

Com o arquivo de traço coletado, *log*, e analisando os eventos programados na simulação, é possível realizar as entradas de informações requeridas pelo arcabouço e posterior retorno deste com modificações do encaminhamento. Sendo, estas modificações, configuradas em uma nova simulação no instante na qual seriam realizadas pelo arcabouço. Por exemplo, um fluxo que se inicie no instante ( $t$ ) e que se considere levar um minuto para ser processado, será programado na simulação para repercutir seus ajustes no instante ( $t + 1$ ). Como consequência, esta simulação terá resultados finais diferentes na anterior, repercutindo os ajustes realizados.

O fluxo exposto pode ser visto da Figura 11 onde é representada a primeira simulação executada e sua geração de traço para análise. A partir destes dados é possível identificar os eventos monitorados pelo arcabouço e realizar a entrada de dados equivalente, executar sua tomada de decisão baseada no contexto, incluindo decisões de validação de contexto e decisões de encaminhamento baseado no contexto, e gerar os dados necessários para mudança da simulação. A reprogramação da simulação é realizada conforme indicações do arcabouço e realizada uma nova execução, gerando novo arquivo de traço.

Figura 11 - Fluxo da simulação



A quantidade de interações depende da análise dos arquivos de traços gerados, seguindo a lógica dos eventos descritos no capítulo 3.

Quando não existem mais eventos a serem processados, esta simulação é considerada final e seus arquivos de traços contém os resultados de uma execução equivalente ao uso do arcabouço em toda sua execução.

Os arquivos de traço, *log*, da primeira execução e da última execução da simulação permitem realizar uma análise comparativa de uma rede em operação baseada em melhor esforço e o que seria esta mesma rede executada considerando o contexto.

Na representação de processamento do arcabouço ilustrado na Figura 11 são realizados dois processamentos distintos: um da abordagem de contexto e outro da abordagem de encaminhamento. Esta divisão está representada na Figura 12:

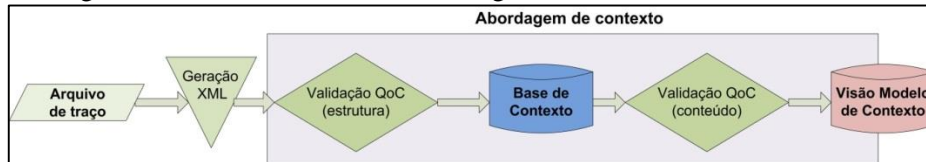
Figura 12 - Divisão do processamento entre as abordagens



Na Figura 12 é possível identificar que o arquivo de traço é processado, gera entradas para a abordagem de contexto, e suas entradas relevantes são disponibilizadas na Visão do Modelo de Contexto que servirá como ponto de partida para a abordagem de encaminhamento realizar seu processamento e indicar as mudanças na simulação necessárias.

Este processamento da abordagem de contexto está representado na Figura 13.

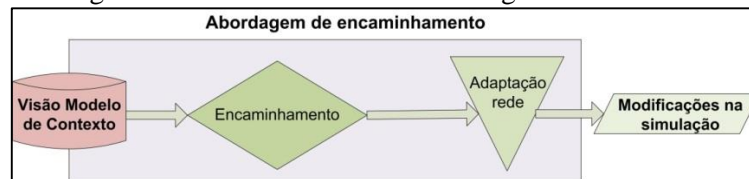
Figura 13 - Processamento da abordagem de contexto



Na Figura 13 está representado o arquivo de traço gerado, a extração dos dados necessários para geração dos arquivos XML, sua posterior validação estrutural que será discutida mais adiante neste trabalho, e sua carga na base de contexto. Uma vez que os dados estejam em base, são executadas rotinas para validação das informações de contexto e criação da visão do Modelo de Contexto.

Estando os dados disponíveis através do Modelo de Contexto, a abordagem de encaminhamento realiza seu processamento, conforme representado na Figura 14.

Figura 14 - Processamento da abordagem de encaminhamento



Conforme ilustrado na Figura 14, com a base de dados de contexto já qualificada pelas regras de QoC, a abordagem de encaminhamento realiza a tomada de decisão e gera as alterações que devem ser feitas na rede. Estas são adaptadas para o tipo de regra de rede a ser configurada. Então, é realizada a reconfiguração da rede e executada uma nova simulação com posterior coleta de novos arquivos de traço.

Na próxima sessão será apresentada a persistência de dados em arquivo XML e em base de dados.

#### 4.4 MODELAGEM DA PERSISTÊNCIA

No arcabouço proposto foram tratados dois pontos chaves para persistência de dados, o arquivo *Markup Description* enviado do *Context Agent* para o *Context Handler*, e as bases de contexto e de modelo de contexto, esta última acessível pela abordagem de encaminhamento para tomada de decisão.

Serão apresentadas aqui suas escolhas e estruturas internas.

##### 4.4.1 Markup Description

Para os dados coletados nos agentes, foi considerada a necessidade destes enviarem dados em formato aberto, flexível e passível de tratamento e validação de suas informações.

Foram avaliados os seguintes formatos de arquivos (LOSCIO, 2015) (OPENKNOWLEDGE, 2015) (MODEL, 2015) :

- ***Comma-Separated Values (CSV)***: Formato compacto; Utiliza separadores entre os campos; Estrutura dos dados depende de documentação da aplicação (CSV, 2015) (LOSCIO, 2015) (MODEL, 2015);
- ***JavaScript Object Notation (JSON)***: Leve; Flexível; Não extensível, já que pode representar qualquer tipo de dado não recorrente; Não tem validador em seus descritores, problema que pode ser resolvido por validação realizada pela aplicação; Usado também em bancos noSQL (JSON, 2015) (LOSCIO, 2015) (MODEL, 2015) ;
- ***Resource Description Framework (RDF)***: Internamente pode utilizar formato JSON ou XML, entre outros; Maior complexidade de definição; Defendido por alguns autores como formato desejável para troca de dados abertos (RDF, 2015) (LOSCIO, 2015) (MODEL, 2015) , e;
- ***eXtensible Markup Language (XML)***: Flexível; Extensível; Tem estrutura de validação fora da aplicação, através dos arquivos de definição de esquemas, *XML Schema Definition (XSD)*; Considerado por alguns autores como o atual padrão para troca de dados na Web (XML, 2015) (LOSCIO, 2015) (MODEL, 2015) .

Outros formatos de arquivos como planilhas, documentos texto, texto puro ou formatos proprietários não foram avaliados por não atenderem as necessidades elencadas para coleta.

Entre os formatos avaliados, foi definida a utilização do formato XML pela existência de estrutura de validação independente da interpretação da aplicação, através de seus arquivos de esquema (XSD) e possibilidade de representar dados recorrentes.

Desta forma, aderente a estruturação do Modelo de Contexto apresentado no capítulo 3, a estrutura das marcações base do arquivo XML ser vista na Figura 15.

Figura 15 - XML - Estrutura base das marcações



Podem ser vistos na Figura 15 a representação dos dados da entidade, entre estes, o marcador *ContextVersion*, o único não representado no Modelo de Contexto. *ContextVersion* é necessário para indicar ao *Handler* qual versão de arquivo XSL deve ser usado para validar os dados e como deve realizar a inserção na base de contexto.

Quanto aos demais marcadores, exemplos de marcações de individualidade estão presentes na Figura 16.

Figura 16 - XML - Exemplos de marcações de individualidade

```

<Individuality>
  <Collected_Username>Joe</Collected_Username>
  <Collected_Hostname>wks1</Collected_Hostname>
  <Subscribe>
    <Provided_Subscribe_ID>11</Provided_Subscribe_ID>
    <Provided_Optional_String>Text</Provided_Optional_String>
    <Provided_Optional_Integer>1000</Provided_Optional_Integer>
    <Provided_Optional_Real>1000</Provided_Optional_Real>
  </Subscribe>
  <Collected_Domainname>lab.unifacs.br</Collected_Domainname>
  <Collected_MAC>84-8F-69-CA-72-CE</Collected_MAC>
  <Collected_IPv4>10.0.0.1</Collected_IPv4>
</Individuality>

```

Na notação ilustrada na Figura 16 estão as informações que identificam o usuário, o equipamento na rede, e assinaturas de serviços que este usuário possua.

Exemplos de marcações de tempo estão presentes na Figura 17.

Figura 17 - XML - Exemplos de marcações de tempo

```

<Time>
  <Collected_NTP>Yes</Collected_NTP>
  <Collected_LastNTPupdate>1</Collected_LastNTPupdate>
  <Collected_LocalTime>2014-07-09T19:20-03:00</Collected_LocalTime>
</Time>

```

Na Figura 17 podem ser observadas informações sobre o tempo atual, e a existência ou não de informações NTP e tempo de atualização.

Exemplos de marcações de localização estão presentes na Figura 18.

Figura 18 - XML - Exemplos de marcações de localização

```

<Location>
  <Collected_GPScoordinates>12°58'47.75"S 38°27'31.37"O</Collected_GPScoordinates>
  <Collected_Network>10.0.0.0/24</Collected_Network>
  <Provided_City>Salvador</Provided_City>
  <Provided_Near>Caminho das Árvores</Provided_Near>
  <Provided_Near>Unifacs</Provided_Near>
</Location>

```

Na Figura 18 podem ser observadas informações de coordenadas GPS, endereço de rede padrão do dispositivo, e informações auxiliares de localização.

Exemplos de marcações de atividade estão presentes na Figura 19.

Figura 19 - XML - Exemplos de marcações de atividade

```

<Activity>
  <Act>
    <Collected_rID>1</Collected_rID>
    <Provided_Subscribe_ID>11</Provided_Subscribe_ID>
  </Act>
  <Act>
    <Collected_rID>2</Collected_rID>
  </Act>
</Activity>

```

Na Figura 19 podem ser observadas informações das atividades em curso, neste exemplo, duas atividades. A primeira atividade está em uso com a relação 1, que será apresentada a seguir, e utiliza a assinatura de serviço 11, descrita nos campos da individualidade apresentados na Figura 16. A segunda atividade não tem informações adicionais, apenas o identificador da relação que faz uso.

Exemplos de marcações de relações estão presentes na Figura 20.

Figura 20 - XML - Exemplos de marcações de relações

```

<Relations>
  <Rel>
    <Collected_rID>1</Collected_rID>
    <Collected_Conected>172.16.0.2</Collected_Conected>
    <Collected_ApplicationName>Skype</Collected_ApplicationName>
    <Collected_ApplicationExecutable>skype.exe</Collected_ApplicationExecutable>
    <Collected_ApplicationVersion>7.0</Collected_ApplicationVersion>
    <Collected_SourcePort>3032</Collected_SourcePort>
    <Collected_DestinationPort>80</Collected_DestinationPort>
    <Collected_TransportProtocol>TCP</Collected_TransportProtocol>
    <Collected_ApplicationProtocol>HTTP</Collected_ApplicationProtocol>
  </Rel>
  <Rel>
    <Collected_rID>2</Collected_rID>
    <Collected_Conected>10.0.0.33</Collected_Conected>
    <Collected_ApplicationName>Viber</Collected_ApplicationName>
    <Collected_ApplicationExecutable>viber.exe</Collected_ApplicationExecutable>
    <Collected_ApplicationVersion>5.0</Collected_ApplicationVersion>
    <Collected_SourcePort>3033</Collected_SourcePort>
    <Collected_DestinationPort>443</Collected_DestinationPort>
    <Collected_TransportProtocol>TCP</Collected_TransportProtocol>
    <Collected_ApplicationProtocol>HTTPS</Collected_ApplicationProtocol>
  </Rel>
</Relations>

```

Na Figura 20 são descritas duas relações ativas, sendo os endereços de origem não explicitamente especificados por já terem sido informados na individualidade e incluindo as portas de origem respectivas. São ainda definidas as informações sobre os destinos das relações, suas aplicações em uso e informações sobre as respectivas aplicações.



Exemplos de marcações de QoE estão presentes na Figura 21.

Figura 21 - XML - Exemplos de marcações de QoE

```

<QoE>
  <Peer>
    <Collected_rID>1</Collected_rID>
    <Collected_MOSs>4.4</Collected_MOSs>
    <Provided_MOSu>2</Provided_MOSu>
    <Collected_RFactor>96</Collected_RFactor>
    <Collected_Noise>100</Collected_Noise>
  </Peer>
  <Peer>
    <Collected_rID>2</Collected_rID>
    <Collected_MOSs>5</Collected_MOSs>
  </Peer>
</QoE>

```

Na Figura 21 são exemplificadas duas situações de coletas distintas: a primeira com o QoE, pertinente a relação 1 apresentada anteriormente, tendo dois valores, um informado pelo usuário (MOSu), e outro calculado pelo sistema (MOSs), entre elas o ruído do ambiente, (*noise*). Neste exemplo, o sistema calculou o MOS da ligação entre boa e excelente (MOSs=4.4). Entretanto, provavelmente por causa do ruído do ambiente, o usuário informou uma qualidade ruim de ligação (MOSu=2) (ITU-T, 2015). Esta discrepância pode, ou não, ser levada em conta na posterior análise da abordagem de encaminhamento. Já na segunda coleta de QoE, a da relação 2 apresentada anteriormente, consta apenas o MOS calculado, igual a cinco, de uma ligação excelente (ITU-T, 2015).

Exemplos de marcações de QoD estão presentes na Figura 22.

Figura 22 - XML - Exemplos de marcações de QoD

```

<QoD>
  <Collected_HasGPS>Yes</Collected_HasGPS>
  <Collected_GPSprecision>10</Collected_GPSprecision>
  <Collected_ProcessorCores>4</Collected_ProcessorCores>
  <Collected_ProcessorClock>2,6</Collected_ProcessorClock>
  <Collected_Memory>6</Collected_Memory>
  <Nic>
    <Collected_NIC_ID>1</Collected_NIC_ID>
    <Collected_Bandwidth>10000</Collected_Bandwidth>
    <Collected_HasToE>Yes</Collected_HasToE>
    <Collected_MAC>84-8F-69-CA-72-CE</Collected_MAC>
    <Collected_IPv4>10.0.0.1</Collected_IPv4>
  </Nic>
</QoD>

```

São especificadas na Figura 22 informações do dispositivo, entre elas, a presença de um GPS, seu nível de precisão e características do processador e memória do equipamento em uso. São ainda fornecidas informações complementares sobre a placa de rede do dispositivo, por exemplo, sua velocidade máxima (10000) expressa em Mbps.

Exemplos de marcações de QoS estão presentes na Figura 23.

Figura 23 - XML - Exemplos de marcações de QoS

```

<QoS>
  <Nic>
    <Collected_NIC_ID>1</Collected_NIC_ID>
    <Collected_Throughput>0,7</Collected_Throughput>
    <Collected_Loss>3</Collected_Loss>
    <Collected_Latency>3</Collected_Latency>
    <Collected_Jitter>2</Collected_Jitter>
  </Nic>
</QoS>

```

Tais informações (Figura 23) são mais comuns em dispositivos de comutação da rede, mas podem também estar disponíveis em sistemas de usuários finais devido a existência de coletores auxiliares, sondas de rede (*network probes*) ou informações trocadas entre as aplicações origem e destino das relações. Neste caso, são exemplificadas a vazão atual da placa de rede, perda de pacotes, latência e variação do atraso.

A íntegra do arquivo XML do exemplo está disponível no Anexo C e o arquivo de definição de esquema, XSD, utilizado para validar este XML, e todos os outros utilizados pelo arcabouço, estão disponíveis no Anexo D.

Quanto aos exemplos de coletas apresentados, vale ressaltar que:

- Nem todas as informações exemplificadas precisam estar presentes, podendo parte destas não ser coletadas pelo agente, e simplesmente serem suprimidas do arquivo XML a ser enviado, e;
- Para facilitar a identificação de que informações estão especificadas, os nomes dos campos não foram abreviados, e é explicitamente identificada a origem dos dados, se informados (*provided*) ou coletados (*collected*). Esta organização pode aumentar muito o tamanho dos arquivos em trânsito e foi utilizada desta forma para facilitar a apresentação neste trabalho. Torna-se necessário avaliar em trabalhos futuros se o impacto é relevante, podendo buscar alternativas para abreviações destas estruturas.

Quanto à questão do impacto no desempenho devido carga de arquivos XML em bases relacionais, principalmente pela necessidade de validação do esquema (NICOLA; JOHN, 2003), a estruturação do arcabouço permite minimizar este impacto devido ao fato destas tarefas serem executadas pelos *Context Handler* que já entregam as informações inseridas na base de contexto, retirando assim esta carga do núcleo do arcabouço. Gargalos no processamento dos arquivos XML podem ser mitigados acrescentando outros *Handlers* ao

ambiente, aumentando sua escalabilidade. Estudos complementares podem ser realizados em trabalhos futuros para quantificar este impacto e verificar se realmente foi resolvido com esta abordagem.

Uma vez definida a persistência em XML, a seguir será apresentada a persistência da base de dados.

#### 4.4.2 Bases de Contexto

Considerando que os dados de contexto já foram coletados pelos agentes e estão disponíveis para manipulação pelo arcabouço, será agora especificada a persistência interna definida em base de dados em Sistema de Gerenciamento de Banco de Dados (SGBD).

A escolha do SGBD foi focada em bancos relacionais gratuitos, sendo a escolha definida pelo *MySQL Community Edition*, que além de ser gratuito, pode ser executado em diversos sistemas operacionais e suporta as características consideradas essenciais para este trabalho, entre elas: replicação, *stored procedures*, *triggers* e *views* (MYSQL, 2015).

O banco reflete a estruturação do Modelo de Contexto, tendo que ser ajustada a fim de permitir a flexibilidade do ambiente, entre elas, a coleta opcional de dados, o que poderia resultar em campos em branco nas tabelas. Para isso, foi definido que nas tabelas principais estariam apenas os dados chaves, índices, e outras informações obrigatórias, ficando os dados auxiliares em tabelas de parâmetros, permitindo um crescimento dinâmico de sua alocação conforme necessário. São elas:

- Tabela Individualidade (*individuality*): Armazena as informações da individualidade e seus vínculos com as tabelas de dispositivos (*devices*) e relações (*relations*), apresentadas adiante. Considerando que alguns parâmetros podem se armazenados e outros não, a flexibilização do armazenamento desta tabela é necessária, conforme apresentado anteriormente, para evitar campos em branco. Esta flexibilização foi conseguida com a criação de uma tabela auxiliar de parâmetros (*parameter*) e as informações são armazenadas na tabela auxiliar (*individualityparameter*), fazendo referência a estes parâmetros previamente cadastrados e ao identificados da individualidade da tabela principal;

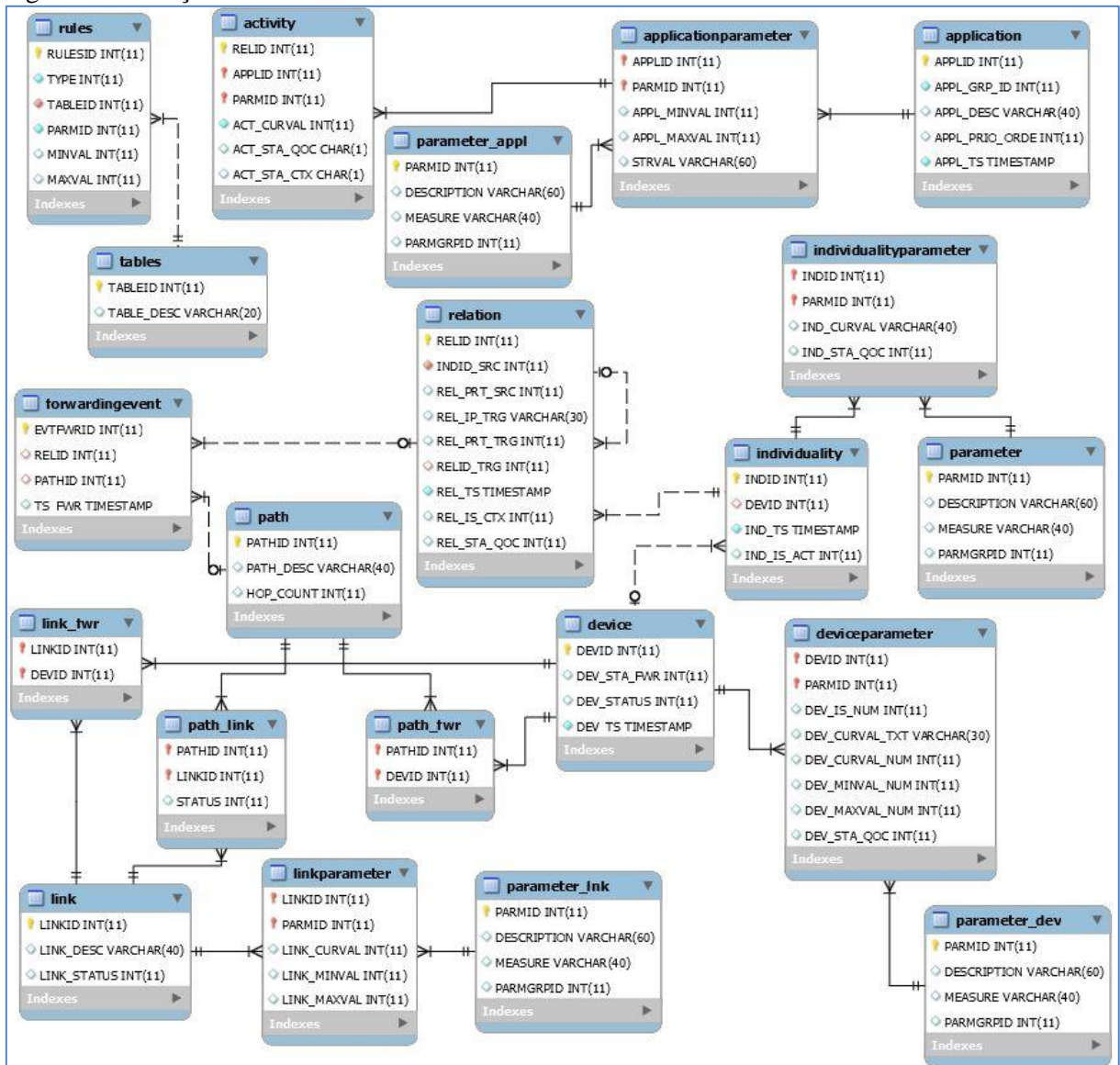
- Tabela Dispositivos (***devices***): Armazena as informações do dispositivo em uso pelo indivíduo, ou informações sobre os dispositivos de encaminhamento existentes na rede. Utiliza a tabela auxiliar de parâmetros (***parameter\_dev***) para identificar que parâmetros são suportados, e a tabela auxiliar (***deviceparameter***) para armazenar seus dados opcionais, conforme identificadores de parâmetros suportados. Uma vez cadastrado o dispositivo, sua identificação é atualizada na tabela da individualidade. A diferenciação dos dispositivos de usuários e dispositivos de encaminhamento é feita pelo seu campo de estado;
- Tabela de Relações (***relation***): Armazena as informações das relações detectadas pelo agente, sendo cada relação um registro independente, mesmo que proveniente do mesmo agente. Não utiliza tabelas auxiliares, contendo todos os dados chaves na própria tabela. Esta tabela pode apresentar auto relacionamento para os casos em que a origem e o destino da comunicação sejam equipamentos dentro do domínio do arcabouço. No cadastro da relação são especificadas a individualidade origem da relação, e a individualidade destino da relação, sendo que esta última apenas em relações cujo destino sejam indivíduos no domínio do arcabouço;
- Tabela Aplicações (***application***): Armazena informações sobre as aplicações suportadas pelo arcabouço, seguindo a lógica de flexibilização adotada anteriormente, os parâmetros suportados são cadastrados em tabela auxiliar (***applicationparameter***), e as informações cadastradas ficam armazenadas na tabela auxiliar (***parameter\_appl***). Entre outras informações, são registrados os requisitos mínimo e desejável de banda para a aplicação, nome do executável e portas de comunicação, entre outros;
- Tabela Atividade (***activity***): Vincula as relações ativas da tabela (***relations***) com as características da aplicação em uso através da tabela (***applicationparameter***), permitindo consulta aos requisitos das aplicações das reações ativas, e outras informações correlacionadas;
- Tabela Regras (***rules***): Armazena informações sobre parte das regras de validação de QoC e regras de contexto, além de informações sobre as regras de encaminhamento. Sua estrutura permite validações através de comparações de informações das tabelas Individualidade, Dispositivos, Aplicações e Ligações apresentadas anteriormente. Sua estruturação em banco permite a inclusão de novas regras em tempo de execução do

controlador de contexto, visto que basta a inclusão destas regras, que nas próximas interações já estarão em uso pelo controlador. Utiliza a tabela auxiliar (*tables*) que cria os vínculos com as demais tabelas da base nas quais estas regras serão aplicadas;

- Tabela Ligações (*link*): Armazena as informações das conexões entre os comutadores e encaminhadores do domínio do arcabouço e suas redes conectadas. Inclui estado das conexões, se ativas ou inativas. Utiliza as tabelas auxiliares de parâmetros das ligações (*linkparameter*) e as vincula a tabela dos dados das conexões (*parameter\_lnk*), que contem ainda as origens e destinos das conexões. Utiliza ainda a tabela auxiliar de ligações e encaminhadores (*link\_fwr*) para criar vínculos entre as ligações e os dispositivos cadastrados, e;
- Tabela Caminho (*path*): Armazena as informações dos caminhos previamente cadastrados pelo administrador, servindo como um cadastro inicial que contem além do identificador, a quantidade de saltos (*hops*), do caminho. Através da tabela auxiliar de caminhos e encaminhadores (*path\_fwr*), cria vínculos entre os encaminhadores do caminho e os dispositivos cadastrados, e através da tabela auxiliar de caminhos e ligações (*path\_link*), cria vínculos entre os caminhos e as ligações existentes. Por fim, através da tabela auxiliar de encaminhadores e eventos (*forwardingevent*) cria os vínculos entre o caminho escolhido e a relação que fará uso deste.

As tabelas e suas relações podem ser vistas na Figura 24.

Figura 24 - Relações da base de contexto



Desta forma, a base de contexto ilustrada na Figura 24 consegue ser flexível para armazenar as informações coletadas, evitando registros em branco desnecessariamente. Permite ainda que as informações possam ser validadas individualmente, tendo campos de estado que indicam se as informações passaram ou não pela validação de QoC, e se após esta validação, são ou não relacionadas ao contexto.

Apenas as informações que tenham seu indicador de estado como ativo para o contexto serão acessíveis através da visão do Modelo de Contexto.

Com a utilização de visões é possível evitar a duplicação de informações em bases, e redução de ações de inserção, mais lentas, optando ações mais rápidas, às consultas.

Após apresentação da persistência, serão apresentadas as linguagens utilizadas neste trabalho.

## 4.5 LINGUAGENS E FERRAMENTAS DE IMPLEMENTAÇÃO

Nesta seção é apresentado o simulador de redes usado neste trabalho, a arquitetura de rede escolhida como solução para implementação dos caminhos baseados em contexto, e a linguagem para avaliação dos resultados.

### 4.5.1 Simulador

Conforme apresentado no capítulo 2, o simulador o NS2 apresenta suporte para as características de redes necessárias para este trabalho, aceitando encaminhamento baseado em melhor esforço, tipicamente vetor distância, e encaminhamento com circuitos virtuais, através de seu suporte a MPLS (NS2, 2015). Junto a isso, a ampla disponibilidade de documentação, códigos e resultados para análise comparativa, e ainda o conhecimento já existente em OTcl (OTCL, 2015), tornam o NS2 a escolha para simulador de redes utilizado neste trabalho.

### 4.5.2 Rede baseada em contexto

Na simulação do encaminhamento baseado em contexto foi avaliada a rede de comutação de rótulos multiprotocolo, *MultiProtocol Label Switching* (MPLS). Estas redes realizam a comutação baseada nas etiquetas (*tags*), para definir o caminho dos pacotes na rede (TANENBAUM, 2003) (OLIVEIRA et al, 2012).

Na convivência com redes baseadas nos cabeçalhos IPs, os pacotes são classificados na entrada da rede MPLS, etiquetados, e suas etiquetas são removidas na saída da rede. Para os pacotes não explicitamente marcados, o algoritmo permite utilizar etiquetas genéricas de caminhos predefinidas pela rede, baseada no algoritmo vetor distância, enquanto que os pacotes com fluxos explicitamente identificados, podem seguir por caminhos pré determinados pelo administrador da rede (OLIVEIRA et al, 2012).

Na implementação do MPLS no NS2, conforme documentação e modelos disponíveis em MPLS-sim-template.txt (NSNAM, 2015) e (JAWHAR, 2015), entre outros, é possível configurar a rede MPLS com suas características citadas acima, com caminhos genéricos ou

com caminhos pré definidos por fluxo de dados, desta forma, é possível simular a rede baseada em contexto neste trabalho.

No NS2, as configurações básicas para MPLS incluem (NSNAM, 2015) (JAWHAR, 2015):

- **Criação dos nós MPLS:**

```
$ns node-config -MPLS ON
set n0 [$ns node]
#(...)
set n3 [$ns node]
$ns node-config -MPLS OFF
```
- **Configuração dos nós MPLS e criação de caminhos:**

```
for {set i 1} {$i < 12} {incr i} {
  set a LSR$i
  for {set j [expr $i+1]} {$j < 12} {incr j} {
    set b LSR$j
    eval $ns LDP-peer $$a $$b
  }
  set m [eval $$a get-module "MPLS"]
  $m enable-reroute "new"
```
- **Configuração da remoção de etiquetas no destino:**

```
$node ldp-trigger-by-withdraw FEC -1
```
- **Criação de um caminho explícito:**

```
$node make-explicit-route FEC no1_no2_..._noN LSPid -1
```
- **Agregação de um fluxo a um caminho existente:**

```
$node flow-aggregation TRAFFIC_CLASS -1 FEC -1
```
- **Instalação, ativação, de um caminho criado:**

```
$node flow-erlsp-install FEC -1 LSPid
```
- **Remoção de um caminho explícito existente:**

```
$node ldp-trigger-by-release LSPid FEC
```

Com estes comandos NS2, é possível reprogramar a simulação para repercutir as mudanças de caminhos definidas pelo arcabouço CAARF para avaliação das mudanças baseadas no contexto.

Em seguida será apresentada a ferramenta de análise dos arquivos de traço gerados.

#### 4.5.3 Ferramenta de análise

Considerando que os arquivos de traço contêm todos os eventos ocorridos na simulação, incluindo, geração e consumo de pacotes, seus tempos, nós de rede envolvidos em



cada comunicação, perdas de pacotes e outras ocorrência (NS2, 2015), o foco das análises deste trabalho é realizado nos arquivos de traço coletados.

Para esta análise, foi escolhida a linguagem AWK que já tem comprovada eficiência na leitura e processamento dos arquivos de traço e geração de dados para análises (AWK, 2015) (ULTIMATE, 2015).

O nome da linguagem, AWK, é proveniente das iniciais de seus criadores, Alfred V. Aho, Peter J. Weinberger, e Brian W. Kernighan. Sendo esta uma linguagem de programação de propósito especial que trabalho com reformatação de dados simples e permite fazer operações sobre grandes volumes de dados com poucas linhas de código (CLOSE et al, 1995). Funciona processando linha a linha os arquivos de traço, e, com a manipulação de marcadores, *tags*, vai separando as informações tabuladas para posterior processamento. Um programa simples para cálculo da taxa de transferência de uma conexão pode ser visto na Figura 25.

Figura 25 - Exemplo de código AWK para calculo da taxa de transferência

```

BEGIN {
  fromNode=5; toNode=6;
  lineCount = 0; totalBits = 0;
}

/^r/&&$3==fromNode&&$4==toNode {
  totalBits += 8*$6;
  if ( lineCount==0 ) {
    timeBegin = $2; lineCount++;
  } else {
    timeEnd = $2;
  };
};

END{
  duration = timeEnd-timeBegin;
  print "Number of records is " NR;
  print "Output: ";
  print "Transmission: N" fromNode "->" toNode;
  print " - Total transmitted bits = " totalBits " bits";
  print " - duration = " duration " s";
  print " - Throughput = " totalBits/duration/1e3 " kbps.";
};

```

Na Figura 25 é possível observar a estruturação da linguagem em três blocos, um bloco inicial que define as variáveis, um bloco intermediário que é processado para cada linha do arquivo de traço a ser analisado, gerando contabilização de resultados e cálculos diversos, e um bloco final que faz a consolidação e exibe os resultados.

## 4.6 CONCLUSÕES

Neste capítulo foi apresentado o passo a passo para a simulação de funcionamento do arcabouço e das regras de encaminhamento utilizadas nele. Foram ainda apresentadas estruturas para exemplificar o encaminhamento das informações.

A modelagem da base de dados representou um ponto de especial atenção, visto que precisava ter a flexibilidade proposta pelo ambiente, com informações que poderiam ou não ser coletadas, e mesmo que fossem coletadas, poderiam não ser disponibilizadas para o encaminhamento a depender se sua aprovação nas regras de QoC. Para resolver isso foi necessário utilizar estruturas altamente flexíveis com campos de controle auxiliares para ativar seletivamente a validação e posterior disponibilização destas informações.

Foram ainda encontradas dificuldades na definição das redes MPLS e seus fluxos, tendo que realizar ajustes na topologia para resolver os problemas. Basicamente, foi necessário isolar os roteadores com função de marcação de pacotes de entrada, ILSR, dos roteadores de encaminhamento do núcleo LSR, evitando que os pacotes não marcados entrantes seguissem equivocadamente por caminhos preexistentes de outros fluxos. Com isso, o MPLS conseguiu atender aos objetivos esperados de tratamento das informações por fluxo, quando configurado, e funcionamento similar as redes melhor esforço para todos os demais casos não explicitados.

A análise posterior dos dados representou um esforço de interpretação das informações coletadas, pois, informações coletadas como perda, atraso ou largura de banda representam uma interpretação resumida de todos os eventos que aconteceram durante todo o caminho dos pacotes. Efetivamente só foi possível analisar os dados de forma adequada quando os resultados foram analisados diretamente nos arquivos de traço de toda a simulação, acompanhando o pacote desde sua geração até seu consumo final, permitindo análises fim a fim e por trecho. Neste ponto, a simplicidade da linguagem AWK facilitou muito a manipulação das informações e geração de dados para análise.

No próximo capítulo são apresentados os cenários utilizados nas simulações, suas coletas de dados e seus resultados obtidos.

## 5 ESTUDOS DE CASO

### 5.1 INTRODUÇÃO

Neste capítulo são apresentados os cenários de simulação para validação do arcabouço CAARF e da abordagem de contexto, sua execução seguindo a modelagem definida no capítulo 4, e características de carga de cada um deles e seus resultados obtidos.

A simulação foco deste trabalho objetiva a análise comparativa e qualitativa dos resultados da execução de uma rede de computadores baseada em encaminhamento IP comum, tratado aqui como rede melhor esforço, e uma rede cujo encaminhamento siga indicações do arcabouço CAARF que considera o contexto atual das entidades envolvidas.

A validação proposta utiliza três cenários com gradação de fluxos e complexidade crescente, sendo cada cenário testado em melhor esforço e depois em encaminhamento baseado em contexto, totalizando portando seis execuções de simulações, seguindo a seguinte estrutura:

Cenários:

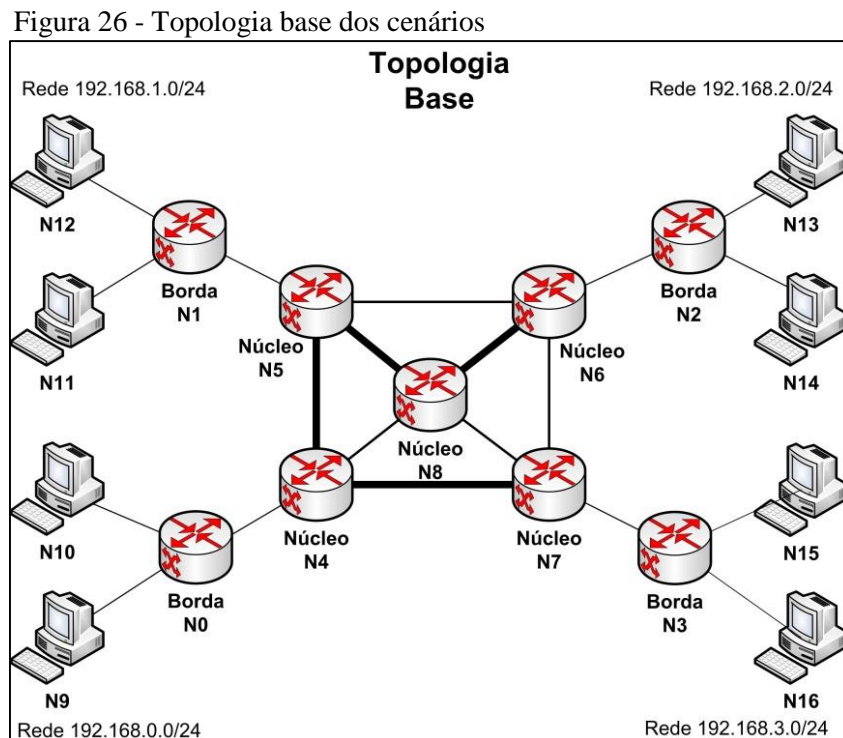
- **Cenário 1:** Carga e complexidade Baixa
  - **Objetivo:** Demonstrar o funcionamento básico do arcabouço, das regras de QoC e do encaminhamento baseado em contexto;
  - **Execuções:**
    - Execução 1: Baixa - Melhor Esforço;
    - Execução 2: Baixa– Contexto;
- **Cenário 2:** Carga e complexidade Média
  - **Objetivo:** Demonstrar o funcionamento do arcabouço com aumento das informações tratadas pelo contexto, incluindo reprovações nas regras de QoC;
  - **Execuções:**
    - Execução 1: Média- Melhor Esforço;
    - Execução 2: Média– Contexto, e;

- **Cenário 3:** Carga e complexidade Alta
  - **Objetivos:** Demonstrar o funcionamento do arcabouço com um volume maior de fluxos e situações variadas de não aprovação nas regras de QoC, inclui validação de QoD e QoE;
  - **Execuções:**
    - Execução 1: Alta- Melhor Esforço;
    - Execução 2: Alta- Contexto.

O ambiente comum aos Cenários e suas caracterizações gerais são apresentadas na seção 5.2, os cenários 1, 2 e 3 são apresentados nas seções 5.3, 5.4 e 5.5, respectivamente, e as conclusões são apresentadas na seção 5.6.

## 5.2 CARACTERIZAÇÃO DA TOPOLOGIA

Dado que a topologia física será a mesma para todos os cenários, e que as variações ficam a cargo da quantidade e características dos fluxos, e dos pares envolvidos, a topologia comum aos cenários pode ser vista na Figura 26.



Na Figura 26 podem ser identificados os encaminhadores com nomes que vão de N0 até N8, e os dispositivos de usuários que vão de N9 até N16. Os encaminhadores são divididos em dois grupos, encaminhadores de borda (*edge*) e encaminhadores de núcleo (*core*).

Ainda conforme a Figura 26, as estações que originam e consomem fluxos, estão organizadas da seguinte forma:

- Nós 9 e 10 tem interface conectada a rede local 192.168.0.0/24, conectada a N0;
- Nós 11 e 12 tem interface conectada a rede local 192.168.1.0/24, conectada a N1;
- Nós 13 e 14 tem interface conectada a rede local 192.168.2.0/24, conectada a N2, e;
- Nós 15 e 16 tem interface conectada a rede local 192.168.3.0/24, conectada a N3.

A nomenclatura dos dispositivos segue a numeração que estes têm na simulação NS2, que foi mantida inalterada para facilitar a identificação posterior das origens e destinos nas informações apresentadas. Estes nomes e números podem ser vistos na Tabela 3.

Tabela 3 - Descrição dos nós de rede

<b>Nº do nó NS2</b>	<b>Nome na rede</b>	<b>Tipo de dispositivo</b>	<b>Especialização</b>
0	N0	Encaminhador	Borda
1	N1	Encaminhador	Borda
2	N2	Encaminhador	Borda
3	N3	Encaminhador	Borda
4	N4	Encaminhador	Núcleo
5	N5	Encaminhador	Núcleo
6	N6	Encaminhador	Núcleo
7	N7	Encaminhador	Núcleo
8	N8	Encaminhador	Núcleo
9	N9	Estação	---
10	N10	Estação	---
11	N11	Estação	---
12	N12	Estação	---
13	N13	Estação	---
14	N14	Estação	---
15	N15	Estação	---
16	N16	Estação	---

A Tabela 3 destaca a especialização dos encaminhadores de núcleo e borda, pois nas redes MPLS os encaminhadores de borda realizam a classificação dos fluxos e inclusão das etiquetas de caminhos, enquanto os encaminhadores de núcleo realizam apenas a leitura das etiquetas e encaminhamento dos pacotes conforme caminhos pré definidos (OLIVEIRA et al, 2012).

Todas as conexões entre os nós permitem envio de dados em ambos os sentidos, *full duplex*, sendo diferenciados por seu tipo e características de largura de banda e atraso, conforme Tabela 4.

Tabela 4 - Descrição das conexões de dados

<b>Tipo</b>	<b>Nó origem</b>	<b>Nó destino</b>	<b>Largura de Banda</b>	<b>Atraso</b>
Acesso	N0	N4	10 Mbps	1 ms
Acesso	N0	N9	10 Mbps	1 ms
Acesso	N0	N10	10 Mbps	1 ms
Acesso	N1	N5	10 Mbps	1 ms
Acesso	N1	N11	10 Mbps	1 ms
Acesso	N1	N12	10 Mbps	1 ms
Acesso	N2	N6	10 Mbps	1 ms
Acesso	N2	N13	10 Mbps	1 ms
Acesso	N2	N14	10 Mbps	1 ms
Acesso	N3	N7	10 Mbps	1 ms
Acesso	N3	N15	10 Mbps	1 ms
Acesso	N3	N16	10 Mbps	1 ms
Núcleo Banda	N4	N5	2 Mbps	50 ms
Núcleo Rápido	N4	N8	1 Mbps	5 ms
Núcleo Banda	N4	N7	2 Mbps	50 ms
Núcleo Rápido	N5	N6	1 Mbps	5 ms
Núcleo Banda	N5	N8	2 Mbps	50 ms
Núcleo Rápido	N6	N7	1 Mbps	5 ms
Núcleo Banda	N6	N8	2 Mbps	50 ms
Núcleo Rápido	N7	N8	1 Mbps	5 ms

Na Tabela 4, as conexões são divididas em três grupos:

- **Acesso:** conexões que representam as ligações de acesso das bordas ao núcleo da rede, operando todas a 10 Mbps e com atraso de 1 ms, caracterizadas desta forma para que suas características causem pouca influência nos resultados das coletas;
- **Núcleo Banda:** conexões que operam todas a 2 Mbps e atraso de 50 ms, representando ligações com maior disponibilidade de banda, entretanto também com maior atraso, e;
- **Núcleo Rápido:** conexões que operam todas a 1 Mbps e atraso de 5 ms, representando ligações mais rápidas, que os pacotes sofrerão menos com os atrasos que outras conexões de núcleo, entretanto, com menor largura de banda disponível.

A topologia base dos cenários apresentada na Figura 26 e a escolha dos valores para Largura de Banda e Atraso das conexões de dados apresentadas na Tabela 4 teve por objetivo permitir que cada rede de borda tivesse no mínimo um caminho com maior largura de banda e

maior atraso e ainda no mínimo um caminho com menor largura de banda e menor atraso para qualquer outra rede de borda existente na topologia.

Neste trabalho não foi previsto mecanismo de descoberta automatizada de caminhos, sendo o cadastramento dos encaminhadores, suas ligações e mapeamento dos caminhos possíveis entre os nós cadastrados previamente pelo administrador do ambiente.

O cadastro dos encaminhadores pode ser visto da Figura 27.

Figura 27 - Cadastro dos encaminhadores

```
INSERT INTO DEVICE (DEVID, DEV_STA_FWR, DEV_STATUS) VALUES (0, 1, 1);
INSERT INTO DEVICE (DEVID, DEV_STA_FWR, DEV_STATUS) VALUES (1, 1, 1);
INSERT INTO DEVICE (DEVID, DEV_STA_FWR, DEV_STATUS) VALUES (2, 1, 1);
INSERT INTO DEVICE (DEVID, DEV_STA_FWR, DEV_STATUS) VALUES (3, 1, 1);
INSERT INTO DEVICE (DEVID, DEV_STA_FWR, DEV_STATUS) VALUES (4, 1, 1);
INSERT INTO DEVICE (DEVID, DEV_STA_FWR, DEV_STATUS) VALUES (5, 1, 1);
INSERT INTO DEVICE (DEVID, DEV_STA_FWR, DEV_STATUS) VALUES (6, 1, 1);
INSERT INTO DEVICE (DEVID, DEV_STA_FWR, DEV_STATUS) VALUES (7, 1, 1);
INSERT INTO DEVICE (DEVID, DEV_STA_FWR, DEV_STATUS) VALUES (8, 1, 1);
```

Na Figura 27 foram cadastrados na tabela DEVICE os dispositivos de 0 até 8, o campo que se trata de um encaminhador foi ativado, valor 1 em DEV\_STA\_FWR, e também foi marcado como ativo o campo que indica estar operacional, valor 1 em DEV\_STATUS. Além destes campos, automaticamente o banco será realiza a inclusão do campo com a hora de inclusão.

Os cadastros de alguns parâmetros de um encaminhador são mostrados na Figura 28.

Figura 28 - Cadastro dos parâmetros de um encaminhador

```
INSERT INTO DEVICEPARAMETER (DEVID, PARMID, DEV_IS_NUM, DEV_CURVAL_TXT) VALUES (0, 12, 0, 'N0');
INSERT INTO DEVICEPARAMETER (DEVID, PARMID, DEV_IS_NUM, DEV_CURVAL_TXT) VALUES (0, 22, 0, '192.168.1.0');
INSERT INTO DEVICEPARAMETER (DEVID, PARMID, DEV_IS_NUM, DEV_CURVAL_TXT) VALUES (0, 26, 0, '192.168.1.1');
INSERT INTO DEVICEPARAMETER (DEVID, PARMID, DEV_IS_NUM, DEV_CURVAL_TXT) VALUES (0, 47, 0, '255.255.255.0');
```

Na Figura 28 é exemplificado o cadastro na tabela DEVICEPARAMETER dos parâmetros do encaminhador N0, sendo informado seu ID, conforme cadastro prévio dos dispositivos na tabela DEVICES, e parâmetros como nome no nó, endereços da interface 1 conectada às estações, com endereço de rede, endereço IP, e mascara de rede e percentual de uso, respectivamente parâmetros 12, 26, 47 e 49 da tabela relacionada PARAMETER\_DEV.

Parâmetros adicionais como percentual de CPU em uso e percentual de memória em uso, respectivamente parâmetros 49 e 50, podem ser cadastrados, ou quais quer outro desejados pelo administrador do ambiente.

O cadastramento dos enlaces, *links*, entre os encaminhadores é mostrado na Figura 29.

Figura 29 - Cadastramento dos links entre os encaminhadores

```

INSERT INTO LINK (LINKID, LINK_DESC, LINK_STATUS) VALUES (1, 'LINK ENTRE NO 1 E NO 4', 1);
INSERT INTO LINK (LINKID, LINK_DESC, LINK_STATUS) VALUES (2, 'LINK ENTRE NO 1 E NO 5', 1);
INSERT INTO LINK (LINKID, LINK_DESC, LINK_STATUS) VALUES (3, 'LINK ENTRE NO 2 E NO 6', 1);
INSERT INTO LINK (LINKID, LINK_DESC, LINK_STATUS) VALUES (4, 'LINK ENTRE NO 3 E NO 7', 1);
INSERT INTO LINK (LINKID, LINK_DESC, LINK_STATUS) VALUES (5, 'LINK ENTRE NO 4 E NO 5', 1);
INSERT INTO LINK (LINKID, LINK_DESC, LINK_STATUS) VALUES (6, 'LINK ENTRE NO 4 E NO 7', 1);
INSERT INTO LINK (LINKID, LINK_DESC, LINK_STATUS) VALUES (7, 'LINK ENTRE NO 4 E NO 8', 1);
INSERT INTO LINK (LINKID, LINK_DESC, LINK_STATUS) VALUES (8, 'LINK ENTRE NO 5 E NO 6', 1);
INSERT INTO LINK (LINKID, LINK_DESC, LINK_STATUS) VALUES (9, 'LINK ENTRE NO 5 E NO 8', 1);
INSERT INTO LINK (LINKID, LINK_DESC, LINK_STATUS) VALUES (10, 'LINK ENTRE NO 6 E NO 7', 1);
INSERT INTO LINK (LINKID, LINK_DESC, LINK_STATUS) VALUES (11, 'LINK ENTRE NO 6 E NO 8', 1);
INSERT INTO LINK (LINKID, LINK_DESC, LINK_STATUS) VALUES (12, 'LINK ENTRE NO 7 E NO 8', 1);

```

Na Figura 29 são cadastrados na tabela LINK os doze enlaces existentes entre os encaminhadores, sendo seus campos o ID do enlace, sua descrição, e a definição de enlace ativo (1), respectivamente. Não é necessário cadastrar os enlaces entre as estações e seus respectivos encaminhadores de borda já que esta associação é obtida através de outras tabelas do modelo, e para efeito de caminho, são considerados apenas os encaminhadores.

O exemplo de cadastramento dos parâmetros dos enlaces pode ser visto na Figura 30.

Figura 30 - Exemplo do cadastro dos parâmetros de cada enlace

```

# Enlace ID 1 (entre N0 e N4), com banda de 10 Mbps e Delay 1 ms
INSERT INTO LINKPARAMETER (LINKID, PARMID, LINK_CURVAL, LINK_MINVAL, LINK_MAXVAL) VALUES (1, 1, 0, 0, 10000);
INSERT INTO LINKPARAMETER (LINKID, PARMID, LINK_CURVAL, LINK_MINVAL, LINK_MAXVAL) VALUES (1, 2, 1, 0, 100);
INSERT INTO LINKPARAMETER (LINKID, PARMID, LINK_CURVAL, LINK_MINVAL, LINK_MAXVAL) VALUES (1, 5, 10000, 0, 10000);

# Enlace ID 5 (entre N4 e N5), com banda de 2 Mbps e Delay 50 ms
INSERT INTO LINKPARAMETER (LINKID, PARMID, LINK_CURVAL, LINK_MINVAL, LINK_MAXVAL) VALUES (5, 1, 0, 0, 2000);
INSERT INTO LINKPARAMETER (LINKID, PARMID, LINK_CURVAL, LINK_MINVAL, LINK_MAXVAL) VALUES (5, 2, 50, 0, 100);
INSERT INTO LINKPARAMETER (LINKID, PARMID, LINK_CURVAL, LINK_MINVAL, LINK_MAXVAL) VALUES (5, 5, 2000, 0, 2000);

# Enlace ID 7 (entre N4 e N8), com banda de 1 Mbps e Delay 5 ms
INSERT INTO LINKPARAMETER (LINKID, PARMID, LINK_CURVAL, LINK_MINVAL, LINK_MAXVAL) VALUES (7, 1, 0, 0, 1000);
INSERT INTO LINKPARAMETER (LINKID, PARMID, LINK_CURVAL, LINK_MINVAL, LINK_MAXVAL) VALUES (7, 2, 5, 0, 100);
INSERT INTO LINKPARAMETER (LINKID, PARMID, LINK_CURVAL, LINK_MINVAL, LINK_MAXVAL) VALUES (7, 5, 1000, 0, 1000);

```

Na Figura 30, são realizados os cadastros na tabela LINKPARAMETER de três enlaces com características distintas, um tipo Acesso entre N0 e N4, um tipo Núcleo Banda entre N4 e N5, e um tipo Núcleo Rápido entre N4 e N8, respectivamente. São mostrados os campos ID do enlace, previamente cadastrado na tabela LINK, ID do parâmetro, da tabela PARAMETER\_LNK, neste campo variando entre os parâmetros banda em uso, atraso e largura de banda total, respectivamente 1, 2 e 5. São cadastrados seus valores atuais, mínimo e máximo respectivamente.

O exemplo da associação entre os enlaces e seus encaminhadores das extremidades é mostrada na Figura 31.



Figura 31 - Exemplo da associação entre os enlaces e os encaminhadores

```
INSERT INTO LINK_FWR (LINKID, DEVID) VALUES (1,0);
INSERT INTO LINK_FWR (LINKID, DEVID) VALUES (1,4);
```

Na Figura 31 esta sendo feita na tabela LINK\_FWR criação da associação entre o nó N0 e o nó N4 ou link de ID 1, conforme ID da tabela LINK previamente cadastrada.

A tabela de caminhos possíveis entre as quatro redes locais do diagrama pode ser vista no Anexo V, e o cadastramento de um caminho desta tabela pode ser visto na Figura 32.

Figura 32 - Cadastramento de um caminho

```
INSERT INTO PATH (PATHID, PATH_DESC, HOP_COUNT) VALUES (1, 'CAMINHO DE NO 0 A NO 3', 3);
```

Na Figura 32, esta sendo cadastrado na tabela PATH o caminho de ID 1, sua descrição e a quantidades de saltos até o destino, respectivamente. O cadastramento do campo descrição exemplificado não tem efeito operacional, sendo realizado apenas para facilitar consultas durante os testes.

A associação dos enlaces de um caminho ao ID deste caminho pode ser vista na Figura 33.

Figura 33 - Associação dos enlaces de um caminho ao ID do caminho.

```
INSERT PATH_LINK (PATHID, LINKID, STATUS) VALUES (1,1,1);
INSERT PATH_LINK (PATHID, LINKID, STATUS) VALUES (1,6,1);
INSERT PATH_LINK (PATHID, LINKID, STATUS) VALUES (1,4,1);
```

Na Figura 33 esta sendo realizado o cadastramento na tabela PATH\_LINK do ID do caminho, previamente cadastrado na tabela PATH, e dos enlaces que fazem parte deste caminho, e esta sendo definido seu estado como ativo (1), respectivamente. Para o caminho de ID 1 são realizadas as associações dos três enlaces que fazem parte deste caminho, 1 6 e 4, respectivamente.

Considerando já estarem todos os encaminhadores, enlaces, associação de encaminhadores e enlaces cadastrados, caminhos e associação dos enlaces aos caminhos, um exemplo de cadastramento de um encaminhamento é mostrado na Figura 34.

Figura 34 - Cadastro de um encaminhamento

```

INSERT INTO PATH_FWR (PATHID,DEVID) VALUES (1,0);
INSERT INTO PATH_FWR (PATHID,DEVID) VALUES (1,4);
INSERT INTO PATH_FWR (PATHID,DEVID) VALUES (1,7);
INSERT INTO PATH_FWR (PATHID,DEVID) VALUES (1,3);

```

Na Figura 34 esta sendo inclusa na tabela PATH\_FRW o ID do caminho previamente cadastrado na tabela PATH e o ID do dispositivo de encaminhamento previamente cadastrado na tabela DEVICE. A cadastro esta associando ao caminho de ID 1 os dispositivos N0, N4, N7 e N3 respectivamente.

Após todos os cadastros concluídos, quando uma estação cadastrada no arcabouço e informar que pertence a rede 192.168.0.0/24, sua origem, e que esta realizando uma atividade com uma estação que pertença a rede 192.168.3.0/24, destino da conexão, o arcabouço pode traçar os caminhos possíveis entre estas estações.

Considerando que o experimento contempla informações de contexto, sua caracterização influencia nos resultados da simulação, portanto, são definidas suas informações contextuais básicas na Tabela 5.

Tabela 5 - Caracterização do contexto das Estações

Nº NS2	Hostname	IP	Rede	Marcara	CPU	Memória
9	N9	192.168.0.9	192.168.0.0	/24	3 Ghz	6 GB
10	N10	192.168.0.10	192.168.0.0	/24	1 Ghz	1 GB
11	N11	192.168.1.11	192.168.1.0	/24	2 Ghz	4 GB
12	N12	192.168.1.12	192.168.1.0	/24		
13	N13	192.168.2.13	192.168.2.0	/24	2 Ghz	4 GB
14						
15	N14	192.168.3.15	192.168.3.0	/24	2 Ghz	4 GB
16	N16	192.168.3.16	192.168.3.0	/24	1 Ghz	1 GB

Na Tabela 5 são apresentadas informações básicas sobre a individualidade e a localização das entidades, tais como nome da estação de rede, endereço IP, endereço de rede, marcara de rede, assim como informações básicas sobre as características de dos dispositivos (QoD) de cada entidade. Cada cenário a ser apresentado pode propor informações contextuais adicionais, sendo as listadas na tabela anterior apenas o ponto de partida a ser considerado. As células em branco representam informações não coletadas pelo agente.

Os fluxos utilizados nos cenários estão caracterizados conforme Tabela 6.

Tabela 6 - Caracterização dos fluxos

<b>Nome</b>	<b>Transporte</b>	<b>Codificação</b>	<b>Vazão</b>
VoIP	UDP	G911	64 Kbps
VoD	UDP	H264 (VGA)	520 Kbps
VoD-HD	UDP	H264 (HD)	1,8 Mbps
Backup	TCP	não se aplica	a disponível

Os fluxos da Tabela 6 utilizam tamanho de pacote padrão do simulador, 1500 Kbps para os agentes e 1000 Kbps para as aplicações (NS2, 2015). Apenas a fluxo VoIP utiliza 200 Kbps para o agente e 160 Kbps para a aplicação, conforme definido no padrão VoIP G911 (G711, 2015) adotado como referência em (CISCO, 2015). Os fluxos VoD foram definidos conforme implementação adotada na solução CISCO WebEx (CISCO2, 2015) do codificador H.264 (H264, 2015). Já o fluxo de Backup considerou os requisitos de fluxo de FTP genérico (KUROSE; ROSS, 2010).

### 5.3 CENÁRIO 1

Como já citado, este cenário tem carga e a complexidade baixas, o objetivo é demonstrar o funcionamento básico do arcabouço, das regras de QoS e do encaminhamento baseado em contexto. Desta forma, nenhum fluxo terá informações de contexto rejeitadas pelas regras de QoS.

São considerados dois fluxos, um VoIP e um VoD-HD, sendo suas origens, destinos e tempos conforme Tabela 7.

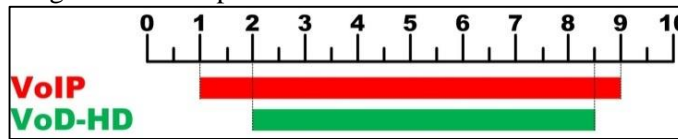
Tabela 7 - Cenário 1: Tempos dos fluxos

<b>Tipo</b>	<b>Origem</b>		<b>Destino</b>		<b>Tempo</b>	
	<b>Nó</b>	<b>Identificador</b>	<b>Nó</b>	<b>Identificador</b>	<b>Início</b>	<b>Fim</b>
VoIP	N9	CBR0/UDP0	N15	Null1	1,0	9,0
VoD-HD	N11	CBR1/UDP2	N13	Null3	2,0	8,5

Pode-se identificar na Tabela 7 que o nó 9 iniciará um fluxo UDP de VoIP com destino ao nó 15 no intervalo de 1,0 até 9,0, enquanto o nó 11 iniciará um fluxo UDP do VoD-HD com destino ao nó 13 no intervalo de 2,0 até 8,5.

Os fluxos VoIP e VoD-HD tem origens e destinos distintos, e sobreposição de tempos, conforme pode ser visto na Figura 35.

Figura 35 - Tempos do Cenário 1

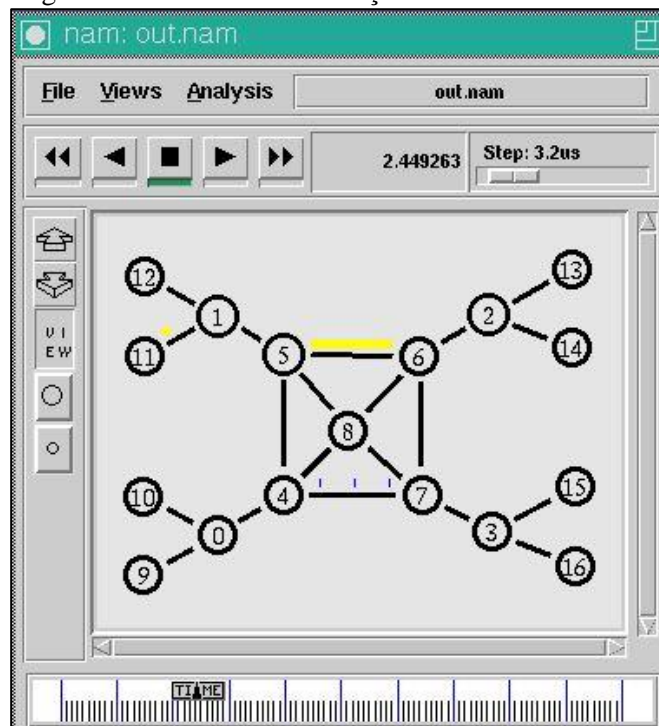


A sobreposição dos tempos da Figura 35 ocorre no intervalo 2,0 até 8,5.

### 5.3.1 Cenário 1 - Execução 1

Uma primeira execução foi realizada para coleta dos resultados de uma rede baseada em melhor esforço comum. O encaminhamento padrão pode ser visto no NAM da Figura 36.

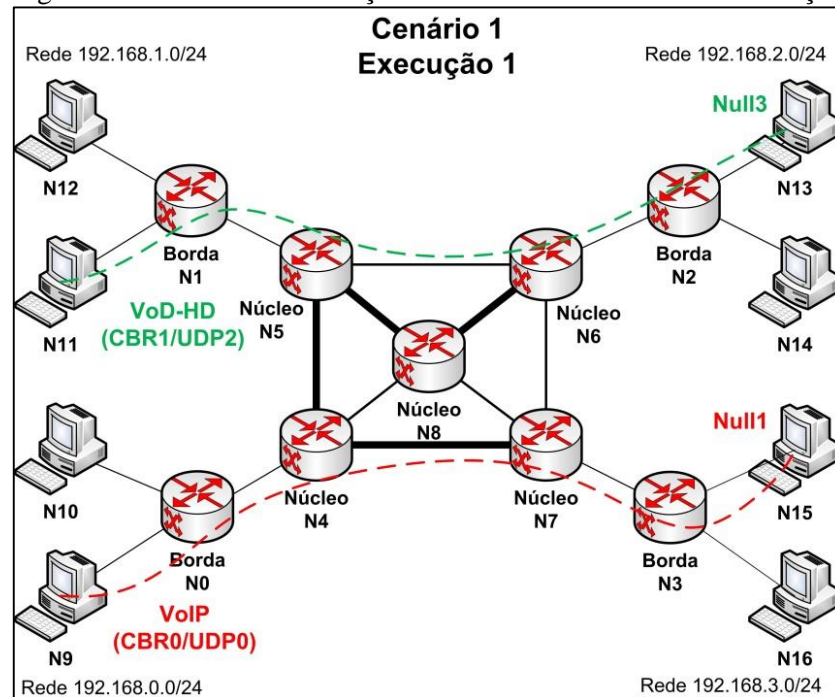
Figura 36 - Cenário 1 - Execução 1: NAM no momento 2,44



A Figura 26, representa o momento 2,4, sendo possível identificar o fluxo de VoIP em azul seguindo entre os nós 4 e 7, e o fluxo de VoD-HD em amarelo seguindo pelos nós 5 e 6. Os fluxos seguiram a menor contagem de saltos para seus destinos, entretanto, o fluxo VoIP seguiu por um caminho com maior largura de banda e maior atraso, N0-N4-N7-N3, e o fluxo VoD-HD seguiu por um caminho com menor largura de banda e menor atraso, N1-N5-N6-N2. Vale destacar que estes caminhos foram escolhidos pela rede apenas pelo critério de contagem de saltos, sem considerações de banda e atraso.

O caminho seguido pelos fluxos esta representado na Figura 37.

Figura 37 - Cenário 1 - Execução 1: Encaminhamento melhor esforço



Na Figura 37, pode-se ver uma ilustração representando os fluxos existentes no NAM da Figura 36 da primeira execução, que devido à representação do instante da informação não permite visualização da totalidade do caminho.

A íntegra do arquivo TCL do Cenário 1 - Execução 1, pode ser visto no Anexo F. Nele estão caracterizados os nós, os links e suas propriedades, as conexões entre, os nós, e os agentes de conexão, conforme definido nas seções 5.2 e 5.3.1 deste trabalho.

Os resultados desta primeira execução são apresentados na Tabela 8.

Tabela 8 - Resultados do Cenário 1 - Execução 1

Fluxo	Atraso Fim a Fim	Bits Transmitidos	Largura de Banda
VoIP	0,0110304 s	513280 bits	64,16kbps
VoD-HD	0,0599102 s	6714720 bits	1001,16 kbps

Na Tabela 8 o VoIP teve largura de banda dentro do configurado no TCL, de 64 Kbps, enquanto o VoD-HD ficou com largura de banda de 1001,16 kbps, inferior ao configurado de 1,8Mbps no TCL devido a restrição de banda máxima do caminho escolhido. Os resultados foram gerados através do processamento do código AWK, um exemplo do código para atraso fim a fim pode ser visto no Anexo G, enquanto que o exemplo do código para bits transmitidos e para largura de banda pode ser visto no Anexo H.

### 5.3.2 Cenário 1 - Execução2

Uma segunda execução do cenário definido na seção 5.3.1 foi realizada baseada no mesmo arquivo TCL base, com a adição neste arquivo de modificações de encaminhamento indicadas pelo arcabouço. Os eventos do arcabouço ocorridos nesta segunda execução podem ser vistos no resumo da Tabela 9.

Tabela 9 - Resumo dos eventos de contexto no Cenário 1 - Execução 2

<b>Tempo (m)</b>	<b>Fluxo</b>	<b>Evento</b>	<b>Ação</b>
1,0	VoIP	Nova relação	Envio XML para arcabouço (mudança de contexto)
1,7	VoIP	Retorno arcabouço	Ativação de novo encaminhamento
2,0	VoD-HD	Nova relação	Envio XML para arcabouço (mudança de contexto)
2,7	VoD-HD	Retorno arcabouço	Ativação novo encaminhamento
8,5	VoD-HD	Fim Relação	Envio XML para arcabouço (mudança de contexto)
9,0	VoD-HD	Retorno arcabouço	Remoção de caminho
9,0	VoIP	Fim Relação	Envio XML para arcabouço (mudança de contexto)
9,5	VoIP	Retorno arcabouço	Remoção de caminho

Na Tabela 9 podem ser identificados os eventos de nova relação, que resultam no envio de arquivos XML para o arcabouço, e também os retornos do arcabouço indicando mudanças de encaminhamento e sua efetiva ativação na simulação, assim como os eventos de fim de relação e sua consequente remoção de caminhos.

Para guiar o entendimento dos acontecimentos relevantes desta simulação, serão explicados os passos seguidos por estes eventos que afetaram a simulação, e seus procedimentos internos realizados pelo arcabouço.

A integra do arquivo XML enviado no instante (1,0) encontra-se no Anexo I.

Um recorte das informações mais relevantes do XML enviado pelo N9 no tempo 1,0 pode ser visto na Figura 38.

Figura 38 - Recorte do XML enviado por N9 no tempo 1,0

```

<Individuality>
  <Collected_Username>N9</Collected_Username>
  <Collected_Hostname>N9</Collected_Hostname>
  <Collected_Domainname>lab.unifacs.br</Collected_Domainname>
  <Collected_MAC>00-53-00-00-00-09</Collected_MAC>
  <Collected_IPv4>192.168.0.9</Collected_IPv4>
</Individuality>

<Location>
  <Collected_Network>192.168.0.0/24</Collected_Network>
</Location>

<Activity>
  <Act>
    <Collected_rID>1</Collected_rID>
  </Act>
</Activity>

<Relations>
  <Rel>
    <Collected_rID>1</Collected_rID>
    <Collected_Conected>192.168.3.15</Collected_Conected>
    <Collected_ApplicationName>Viber</Collected_ApplicationName>
    <Collected_ApplicationExecutable>viber.exe</Collected_ApplicationExecutable>
    <Collected_ApplicationVersion>5.0</Collected_ApplicationVersion>
    <Collected_SourcePort>50000</Collected_SourcePort>
    <Collected_DestinationPort>5243</Collected_DestinationPort>
    <Collected_TransportProtocol>UDP</Collected_TransportProtocol>
  </Rel>
</Relations>

```

Na Figura 38 podem-se identificar as informações da individualidade, da localização, da atividade e da relação. Foi identificada uma relação de N9 com o IP 192.168.3.15, utilizando o aplicativo Viber, na porta remota 5243.

O cadastro dos dispositivos é feito pelo *Handler* à medida que chegam arquivos XML com informações destes, conforme sequencia de atividades.

I – No exemplo, a consulta para verificar se já existe o nó N9 pode ser vista na Figura 39.

Figura 39 - Consulta se um dispositivo já existe antes da inclusão

```

SELECT
  DE.DEVID AS DEVICE
FROM
  device de inner join
  (select devid from deviceparameter dp inner join parameter_dev pd where `pd`.`DESCRIPTION` LIKE 'HOSTNAME'
   AND `dp`.`DEV_CURVAL_TXT` = 'N9') a   on (de.devid = a.devid and dev_status = 1 and dev_sta_fwr = 0)
inner join (select devid from deviceparameter dp inner join parameter_dev pd where `pd`.`DESCRIPTION` LIKE 'IPV4 IF 1'
   AND `dp`.`DEV_CURVAL_TXT` = '192.168.0.9') c   on (a.devid = c.devid)
inner join (select devid from deviceparameter dp inner join parameter_dev pd where `pd`.`DESCRIPTION` LIKE 'NETWORK IPV4 IF 1'
   AND `dp`.`DEV_CURVAL_TXT` = '192.168.0.0') d   on (a.devid = d.devid)
inner join (select devid from deviceparameter dp inner join parameter_dev pd where `pd`.`DESCRIPTION` LIKE 'NETWORK MASK IF 1'
   AND `dp`.`DEV_CURVAL_TXT` = '255.255.255.0') e   on (a.devid = e.devid)

```

Na Figura 39, esta sendo consultado se o nó N9, que não é um encaminhador, esta ativo com as informações de contexto coletadas. Se a resposta for diferente de *null*, significa

que já existe este dispositivo ativo e não é necessário incluí-lo, se for *null*, será realizada sua inclusão.

**II**– Neste caso, considerando ser um novo dispositivo, sua inclusão é realizada conforme consulta da Figura 40.

Figura 40 - Inclusão de novo dispositivo e suas características

```
INSERT INTO DEVICE (DEVID, DEV_TS, DEV_STA_FWR, DEV_STATUS) VALUES (9, CURRENT_TIMESTAMP, 0, 1); # Novo Dispositivo
INSERT INTO DEVICEPARAMETER (DEVID, PARMID, DEV_CURVAL_TXT ) VALUES (9, 12, 'N9'); # Hostname
INSERT INTO DEVICEPARAMETER (DEVID, PARMID, DEV_CURVAL_TXT ) VALUES (9, 26, '192.168.0.9'); # NET IPv4 IF1
INSERT INTO DEVICEPARAMETER (DEVID, PARMID, DEV_CURVAL_TXT ) VALUES (9, 22, '192.168.0.0'); # IPv4 IF1
INSERT INTO DEVICEPARAMETER (DEVID, PARMID, DEV_CURVAL_TXT ) VALUES (9, 47, '255.255.255.0'); # NETMASK IF1
INSERT INTO DEVICEPARAMETER (DEVID, PARMID, DEV_CURVAL_NUM ) VALUES (9, 8, '3000'); # CPU Total (MHz)
INSERT INTO DEVICEPARAMETER (DEVID, PARMID, DEV_CURVAL_NUM ) VALUES (9, 49, '10'); # % uso CPU
INSERT INTO DEVICEPARAMETER (DEVID, PARMID, DEV_CURVAL_NUM ) VALUES (9, 9, '6000'); # Memória Total (MB)
INSERT INTO DEVICEPARAMETER (DEVID, PARMID, DEV_CURVAL_NUM ) VALUES (9, 50, '25'); # % uso Memória
```

Na Figura 40, é realizada a inclusão do dispositivo na tabela DEVICE, e são inseridos seus atributos na tabela DEVICEPARAMETER, sendo eles, *hostname*, endereço de rede, endereço IP, máscara de rede, características da CPU e memória e percentual de uso.

**III**– Já com o dispositivo cadastrado, é cadastrada a individualidade, conforme Figura 41.

Figura 41 - Cadastro da individualidade

```
INSERT INTO INDIVIDUALITY (INDID, IND_TS, IND_IS_ACT) VALUES (1, CURRENT_TIMESTAMP, 1); # Cadastro da Individualidade
INSERT INTO INDIVIDUALITYPARAMETER (INDID, PARMID, IND_CURVAL ) VALUES (1, 14, 'N9 User'); # USERNAME
INSERT INTO INDIVIDUALITYPARAMETER (INDID, PARMID, IND_CURVAL ) VALUES (1, 13, 'lab.unifacs.br'); # DOMAIN
INSERT INTO INDIVIDUALITYPARAMETER (INDID, PARMID, IND_CURVAL ) VALUES (1, 26, '192.168.0.9'); # IPV4 IF 1
INSERT INTO INDIVIDUALITYPARAMETER (INDID, PARMID, IND_CURVAL ) VALUES (1, 22, '192.168.0.0'); # NETWORK IPV4 IF 1
INSERT INTO INDIVIDUALITYPARAMETER (INDID, PARMID, IND_CURVAL ) VALUES (1, 47, '255.255.255.0'); # NETMASK IF 1
```

Na Figura 41 pode ser que cadastro na tabela INDIVIDUALITY da individualidade 1, e parâmetros *username*, *domain*, endereço IP, endereço de rede, máscara de rede, respectivamente 14, 13, 26, 22 e 47, cadastrados na tabela INDIVIDUALITYPARAMETER.

**IV**– A partir do momento que tanto o dispositivo quanto a individualidade estão cadastrados, é cadastrada a nova relação, conforme Figura 42.

Figura 42 - Cadastro da Relação

```
INSERT INTO RELATION (RELID, INDID_SRC, REL_PRT_SRC, REL_IP_TRG, REL_PRT_TRG, REL_TS, REL_IS_CTX, REL_STA_QOC)
VALUES ( 1, 1, 50000, '192.168.3.15', 5243, CURRENT_TIMESTAMP, 0, 0);
```

Na Figura 42 a relação é cadastrada na tabela RELATION, novo RELID, e associada à individualidade origem previamente cadastrada, INDID\_SRC, com seus parâmetros de porta de origem (50000), endereço de rede destino, porta de destino (5243), e hora de cadastro são



registrados. Inicialmente os parâmetros de teste de QoC, REL\_STA\_QOC, e se faz parte do contexto, REL\_IS\_CTX ficam em zero, indicando que a relação não foi verificada pelo QoC e ainda não faz parte do contexto.

V– Cadastro da Atividade, conforme Figura 43.

Figura 43 - Cadastro da Atividade

```
INSERT INTO ACTIVITY (RELID, APPLID, PARMID, ACT_CURVAL, ACT_STA_QOC, ACT_STA_CTX) VALUES (1, 2, 24, 50000, 0, 0);
INSERT INTO ACTIVITY (RELID, APPLID, PARMID, ACT_CURVAL, ACT_STA_QOC, ACT_STA_CTX) VALUES (1, 2, 25, 5243, 0, 0);
INSERT INTO ACTIVITY (RELID, APPLID, PARMID, ACT_CURVAL, ACT_STA_QOC, ACT_STA_CTX) VALUES (1, 2, 31, 1, 0, 0);
INSERT INTO ACTIVITY (RELID, APPLID, PARMID, ACT_CURVAL, ACT_STA_QOC, ACT_STA_CTX) VALUES (1, 2, 33, 1, 0, 0);
```

Na Figura 43 são cadastrados na tabela ACTIVITY os parâmetros da atividade associada à relação de ID 1, sendo o parâmetro 24 a porta de origem, o parâmetro 25 a porta de destino, o parâmetro 31 a associação com o nome da aplicação Viber, e o parâmetro 33 a associação aos executável do Viber.

VI– Após a inclusão da relação e da atividade no banco, a verificação de QoC interna ao banco é realizada para a individualidade, conforme Figura 44.

Figura 44 - Verificação de QoC da Identidade

```
select count(*), a.parmid
  from individualityparameter a
     inner join (select minval, maxval, parmid from rules where type = 1 and tableid = 1) b
     on (a.parmid = b.parmid)
     inner join parameter c
     on (a.parmid = c.parmid)
 where a.ind_curval is null
    and ( c.description like 'NETWORK IPV4%'
        or c.description like 'NETWORK MASK IPV4%'
        or c.description like 'IPV4%')
```

Na Figura 44 são verificados se os parâmetros de rede, IP e mascara foram preenchidos, sendo estes os parâmetros mínimos necessários para identificação de um fluxo.

O conjunto de regras definidas como tipo 1, regras de QoC, e consultado da tabela RULES e aplicado na tabela 1, da individualidade, sendo exemplos de checagem:

- Existência de IP;
- Existência de rede IP, e;
- Existência de mascara.

Se estes parâmetros não tiverem sido informados, o QoC dos parâmetros da identidade não será validado, sendo os parâmetros ind\_sta\_qoc da tabela individualityparameter definidos como 0, caso aprovados no QoC, definidos para 1, conforme Figura 45.

Figura 45 - Atualização do estado de QoC aprovado para parâmetros da Individualdiade

```
UPDATE `contextdatabase`.`individualityparameter` SET `IND_STA_QOC`='1' WHERE `INDID`='1' and `PARMID`='13';
UPDATE `contextdatabase`.`individualityparameter` SET `IND_STA_QOC`='1' WHERE `INDID`='1' and `PARMID`='14';
UPDATE `contextdatabase`.`individualityparameter` SET `IND_STA_QOC`='1' WHERE `INDID`='1' and `PARMID`='22';
UPDATE `contextdatabase`.`individualityparameter` SET `IND_STA_QOC`='1' WHERE `INDID`='1' and `PARMID`='26';
UPDATE `contextdatabase`.`individualityparameter` SET `IND_STA_QOC`='1' WHERE `INDID`='1' and `PARMID`='47';
```

Na Figura 45 são definidos como QoC aprovado todos os parâmetros validados da individualidade. Destaca-se o fato do QoC ser validado e aprovado por parâmetro, e não pelo conjunto de parâmetros.

## VII– Verificação de QoC da Aplicação, conforme Figura 46.

Figura 46 - Validação de QoC da Aplicação

```
select a.parmid from activity a
  inner join (select minval, maxval, parmid from rules where type = 1 and tableid = 2) b
    on (a.parmid = b.parmid)
   where a.act_curval is null
      and a.parmid in (29)
union
select a.parmid from activity a
  inner join (select minval, maxval, parmid from rules where type = 1 and tableid = 2) b
    on (a.parmid = b.parmid)
   where a.act_curval = 0
      and a.parmid in (31,33)
```

Na Figura 46 é realizada a verificação se a porta da aplicação foi inserida em branco, parâmetro 29, e se foi identificado o ID do programa e da aplicação, parâmetros 31 e 33 respectivamente.

O conjunto de regras definidas como tipo 1, regras de QoC, e consultado da tabela RULES e aplicado na tabela 2, da atividade, sendo exemplos de checagem:

- Se a porta origem da aplicação esta no intervalo de 1 até 65365;
- Se a porta destino da aplicação esta no intervalo de 1 até 65365;
- Se o rede destino pode ser identificada;
- Se o executável da aplicação foi identificado, permitindo distinguir a aplicação, e;
- Se o valor do MOS esta na faixa válida, 0 até 5.

Se a porta ou a identificação da aplicação não existirem, o QoC não será validado para os parâmetros da Aplicação, e os parâmetros act\_sta\_qoc da tabela activity serão definidos para 0, caso aprovados no QoC, definidos para 1, conforme Figura 47.

Figura 47 - Atualização do estado de QoC aprovado para parâmetros da Atividade

```
UPDATE `contextdatabase`.`activity` SET `ACT_STA_QOC`='1' WHERE `RELID`='1' and `APPLID`='1' and `PARMID`='1';
UPDATE `contextdatabase`.`activity` SET `ACT_STA_QOC`='1' WHERE `RELID`='1' and `APPLID`='1' and `PARMID`='2';
UPDATE `contextdatabase`.`activity` SET `ACT_STA_QOC`='1' WHERE `RELID`='1' and `APPLID`='1' and `PARMID`='3';
UPDATE `contextdatabase`.`activity` SET `ACT_STA_QOC`='1' WHERE `RELID`='1' and `APPLID`='2' and `PARMID`='24';
UPDATE `contextdatabase`.`activity` SET `ACT_STA_QOC`='1' WHERE `RELID`='1' and `APPLID`='2' and `PARMID`='25';
UPDATE `contextdatabase`.`activity` SET `ACT_STA_QOC`='1' WHERE `RELID`='1' and `APPLID`='2' and `PARMID`='31';
UPDATE `contextdatabase`.`activity` SET `ACT_STA_QOC`='1' WHERE `RELID`='1' and `APPLID`='2' and `PARMID`='33';
```

Na Figura 47 são definidos como QoC aprovado todos os parâmetros validados da atividade. Novamente, destaca-se o fato do QoC ser validado e aprovado por parâmetro, e não pelo conjunto de parâmetros.

#### VIII– Verificação de QoC do Dispositivo, conforme Figura 48.

Figura 48 - Validação de QoC do Dispositivo

```
select count(*), a.parmid into cont3, parm from deviceparameter a
  inner join (select minval, maxval, parmid from rules where type = 1 and tableid = 3) b
 on (a.parmid = b.parmid)
  where a.dev_curval_num <= b.minval
 or a.dev_curval_num >= b.maxval
```

Na Figura 48 é realizada a verificação se os parâmetros cadastrados para o dispositivo estão dentro das faixas de tolerância definidas nas regras como válidas.

O conjunto de regras definidas como tipo 1, regras de QoC, e consultado da tabela RULES e aplicado na tabela 3, dos dispositivo, sendo exemplos de checagem:

- Se o endereço IP foi informado;
- Se a rede foi informada;
- Se a mascara foi informada, e;
- Se a capacidade da CPU esta na faixa aceitável, 0 até 10000 (considerando Khz).

Se a porta ou a identificação da aplicação não existirem, o QoC não será validado para os parâmetros da Aplicação, e os parâmetros act\_sta\_qoc da tabela activity serão definidos para 0, caso aprovados no QoC, definidos para 1, conforme Figura 49.

Figura 49 - Atualização do estado de QoC aprovado para parâmetros do dispositivo

```
UPDATE `contextdatabase`.`deviceparameter` SET `DEV_STA_QOC`='1' WHERE `DEVID`='9' and `PARMID`='8';
UPDATE `contextdatabase`.`deviceparameter` SET `DEV_STA_QOC`='1' WHERE `DEVID`='9' and `PARMID`='9';
UPDATE `contextdatabase`.`deviceparameter` SET `DEV_STA_QOC`='1' WHERE `DEVID`='9' and `PARMID`='12';
UPDATE `contextdatabase`.`deviceparameter` SET `DEV_STA_QOC`='1' WHERE `DEVID`='9' and `PARMID`='22';
UPDATE `contextdatabase`.`deviceparameter` SET `DEV_STA_QOC`='1' WHERE `DEVID`='9' and `PARMID`='26';
UPDATE `contextdatabase`.`deviceparameter` SET `DEV_STA_QOC`='1' WHERE `DEVID`='9' and `PARMID`='47';
UPDATE `contextdatabase`.`deviceparameter` SET `DEV_STA_QOC`='1' WHERE `DEVID`='9' and `PARMID`='49';
```

Na Figura 49, se os valores testados estiverem fora das faixas definidas, o parâmetro `dev_sta_qoc` da tabela `deviceparameter` é definido como 0 e a informação não passou no QoC, caso contrário, definido como 1 e está validado pelo QoC.

Desta forma, os grupos de informações são validados pelo QoC do banco.

**IX**– Após validação dos grupos de informações, a relação é atualizada como validada no QoC e sendo integrante do Contexto, conforme Figura 50.

Figura 50 - Definindo uma relação como pertencente ao Contexto

```
UPDATE `contextdatabase`.`relation` SET `REL_IS_CTX`='1', `REL_STA_QOC`='1' WHERE `RELID`='1';
```

Na Figura 50 são atualizados tabela `RELATION`, em `REL_STA_QOC` indicando que a relação teve suas regras de QoC validadas, e em `REL_IS_CTX` indicando que esta relação agora faz parte do contexto.

**X**– Após atualização do Contexto, considera-se que uma notificação é enviada a abordagem de encaminhamento para processamento do novo encaminhamento.

**XI**– A consulta do Contexto atual pode ser vista na Figura 51.

Figura 51 - Consulta do Contexto atual

```
select * from context_model_view
```

Na Figura 51 o procedimento armazenado, *stored procedure*, `CONTEX_MODEL_VIEW` é selecionada.

O novo Contexto pode ser visto na Figura 52.

Figura 52 - Exibição do novo Contexto com uma relação no Cenário 1

HOST_ORIGEM	IP_ORIGEM	REDE_ORIGEM	MASK_ORIGEM	PORTA_ORIGEM	IP_DESTINO
N15	192.168.0.9	192.168.0.0	255.255.255.0	10000	192.168.3.15
PORTA_DESTINO	APLICACAO	APPL_VAZAO	ORDEM_VAZAO	APPL_ATRASO	ORDEM_ATRASO
5243	VIBER	200	2	50	1

Na Figura 52 é exibida a relação que faz parte do contexto atual e seus dados básicos. A imagem foi dividida em duas linhas para melhor visualização. Informações adicionais do contexto podem ser selecionadas pela abordagem de encaminhamento através de consultas as tabelas relacionadas.

Segundo (MUAKAD, 2015) a consulta de melhor caminho realizada pela abordagem de encaminhamento exibe o resultado da Figura 53.

Figura 54 - Consulta de melhor caminho para o VoIP do Cenário 1

```

1 • select a.pathid, a.path_delay_sum, a.path_hop_count from path a
2   inner join (select pf.pathid from (select pathid from path_fwr where devid = 0) pf
3   inner join (select pathid from path_fwr where devid = 3) pa
4   on (pf.pathid = pa.pathid)
5   group by pf.pathid) b
6   on (a.pathid = b.pathid)
7   order by a.path_delay_sum, a.path_hop_count
8   limit 1
9

```

pathid	path_delay_sum	path_hop_count
2	12	4

A Figura 53 mostra a ultima consulta executada pela abordagem de encaminhamento, indicando o caminho de ID 2, isto é, **N0-N4-N8-N7-N3**, que tem atraso total de 12 ms e vai passar por quatro saltos.

Baseado no retorno da abordagem de encaminhamento, a simulação foi reprogramada para definir o caminho do fluxo VoIP conforme indicado, entrando esta alteração em vigor no momento 1,7 minuto.

No momento 2 minutos, o fluxo VoD-HD foi iniciado e gerou novo XML par o arcabouço, com uma nova relação de N11 para 192.168.2.13, utilizando o aplicativo CISCO WebEx, na porta remota 9000, conforme Anexo J, e que processado seguindo os mesmos passos definidos **I a XI**, com as seguintes diferenças:

- Em I, é consultado se o nó N11 já existe e está ativo, com resposta nula, não existe ou não está ativo;
- Em II, é realizada a inclusão do dispositivo e dos parâmetros do dispositivo;
- Em III, é verificado se a individualidade já existe, com resposta nula, cadastra a individualidade;
- Em IV, a nova relação é cadastrada;
- Em V, a atividade é cadastrada;
- Em VI, é verificado o QoC para a individualidade;
- Em VII, é verificado o QoC para a aplicação;
- Em VIII, é verificado o QoC para o dispositivo;
- Em IX, a relação é atualizada para Contexto;
- Em X, a consulta do novo contexto é exibida conforme Figura 54.

Figura 55 - Novo contexto do Cenário 1 com duas relações

HOST_ORIGEM	IP_ORIGEM	REDE_ORIGEM	MASK_ORIGEM	PORTA_ORIGEM	IP_DESTINO	PORTA_DESTINO		
N15	192.168.0.9	192.168.0.0	255.255.255.0	10000	192.168.3.15	5243		
N15	192.168.0.9	192.168.0.0	255.255.255.0	11321	192.168.2.13	9000		
				APLICACAO	APPL_VAZAO	ORDEM_VAZAO	APPL_ATRASO	ORDEM_ATRASO
				VIBER	200	2	50	1
				Cisco WebEX	800	1	400	2

Na Figura 54 é possível identificar as duas relações ativas no contexto com suas informações básicas. A imagem está dividida em dias para melhor visualização. Outras informações do contexto podem ser consultadas pela abordagem de encaminhamento através das relações das tabelas.

Segundo (MUAKAD, 2015) a consulta de melhor caminho realizada pela abordagem de encaminhamento para a relação entre 192.168.0.9 e 192.168.2.13 é exibida na Figura 55.

Figura 56 - Consulta de melhor caminho para o VoD do Cenário 1

```

1 • select a.pathid, a.path_bw_min, a.path_hop_count from path a
2   inner join (select pf.pathid from (select pathid from path_fwr where devid = 1) pf
3   inner join (select pathid from path_fwr where devid = 2) pa
4   on (pf.pathid = pa.pathid)
5   group by pf.pathid ) b
6   on (a.pathid = b.pathid)
7   order by a.path_bw_min desc, a.path_hop_count
8   limit 1

```

pathid	path_bw_min	path_hop_count
21	2000.00	4

A Figura 55 mostra a ultima consulta executada pela abordagem de encaminhamento, indicando o caminho de ID 21, isto é, **N1-N5-N8-N6-N2**, que tem banda de 2 Mbps e vai passar por quatro saltos.

Baseado no retorno da abordagem de encaminhamento, a simulação foi reprogramada para definir o caminho do fluxo VoD-HD conforme indicado, entrando esta alteração em vigor no momento 2,7 minutos.

A configuração dos caminhos MPLS baseados no arcabouço podem ser vista na Figura 56.

Figura 57 - Cenário 1 - Execução 2 : Configuração dos caminhos MPLS

```

# MPLS Lables
$ns at 0.5 "[$n2 get-module MPLS] ldp-trigger-by-withdraw 13 -1"
$ns at 0.5 "[$n3 get-module MPLS] ldp-trigger-by-withdraw 15 -1"

# VoIP
$ns at 1.5 "[$n0 get-module MPLS] make-explicit-route 3 0_4_8_7_3 1015 -1"
$ns at 1.7 "[$n0 get-module MPLS] flow-erlsp-install 15 -1 1015"
$ns at 9.0 "[$n0 get-module MPLS] ldp-trigger-by-release 13 1015"

# VoD-HD
$ns at 2.5 "[$n1 get-module MPLS] make-explicit-route 2 1_5_8_6_2 1013 -1"
$ns at 3.0 "[$n1 get-module MPLS] flow-erlsp-install 13 -1 1013"
$ns at 9.5 "[$n1 get-module MPLS] ldp-trigger-by-release 13 1001"

# MPLS Lables End

```

Na Figura 56 são configurados os caminhos explícitos do VoIP por **N0-N4-N8-N7-N3**e do VoD-HD por **N1-N5-N8-N6-N2**.

A execução pode ser vista nas Figuras 57, 58 e 59.

Figura 58 - Cenário 1 - Execução 2: NAM momento 1,39

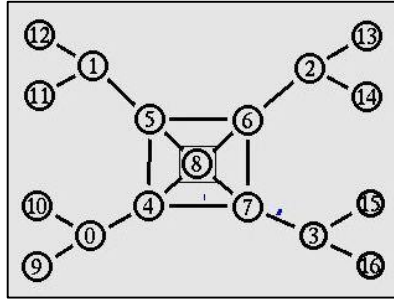


Figura 59 - Cenário 1 - Execução 2: NAM momento 2,08

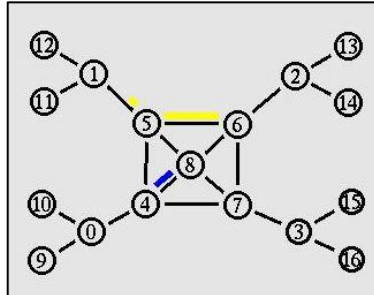
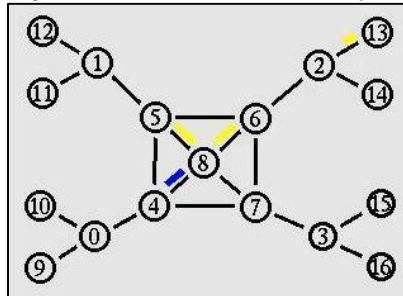


Figura 60 - Cenário 1 - Execução 2: NAM momento 4,4

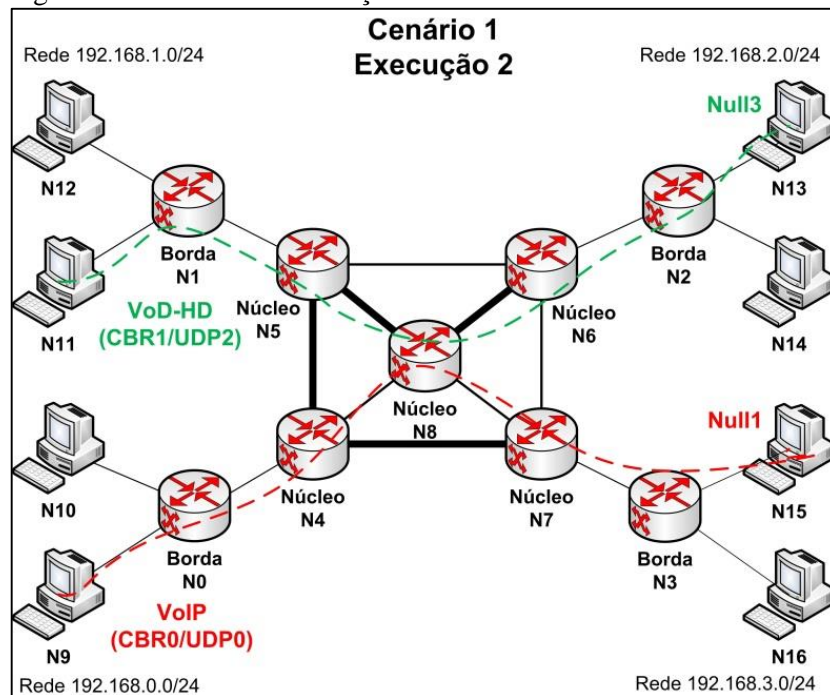


Na Figura 57 a simulação está no momento 1,39 e o fluxo VoIP segue pelos caminhos originais baseado em melhor esforço. Na Figura 58 a simulação está no momento 2,08 e já se identifica o fluxo VoIP seguindo pelo caminho alternativo baseado em contexto, enquanto o fluxo de VoD-HD continua no caminho original. Na Figura 59 a simulação está no momento 4,4 e ambos os fluxos já seguem pelos caminhos alternativos baseados em contexto.

O resultado desta execução pode ser visto na Figura 60.



Figura 61 - Cenário 1 - Execução 2: Encaminhamento baseado em contexto



A Figura 60 representa os fluxos seguindo pelos caminhos após modificação baseada em contexto. Vale lembrar que, por exemplo, o fluxo VoIP seguiu o caminho N0-N4-N7-N3 até o instante 2,5, mudando neste instante para o caminho N0-N4-N8-N7-N3.

Os resultados desta execução 2 estão disponíveis na Tabela 10.

Tabela 10 - Resultados do Cenário 1 - Execução 2

Fluxo	Atraso Fim a Fim	Transmitidos	Largura de Banda
VoIP	0,00223492 s	513280bits	64,16kbps
VoD-HD	0,0228405 s	11296000bits	1712,35kbps

Na Tabela 10 o fluxo VoIP mantém a largura de banda configurada, 64 kbps, enquanto o fluxo VoD-HD aumenta seu uso de largura de banda, se aproximando da configuração definida na simulação de 1,8 Mbps. Quanto ao atraso fim a fim, o fluxo VoIP tem uma leve redução de tempos em relação a coleta anterior.

## 5.4 CENÁRIO 2

Como já citado, este cenário tem carga e a complexidade médias, e o objetivo é demonstrar o funcionamento do arcabouço com aumento das informações sendo tratadas pelo contexto, incluindo reprovações nas regras de QoC.

Para tanto, são mantidos os dois fluxos do cenário 1, e são incluídos mais dois fluxos que terão seu QoC reprovado, conforme Tabela 11.

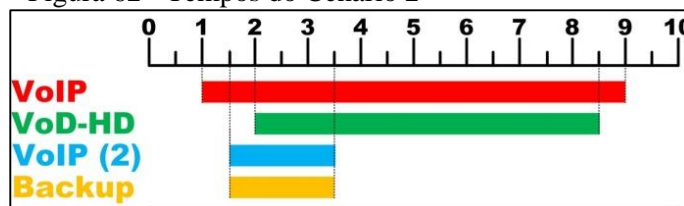
Tabela 11 - Cenário 2: Tempos dos fluxos

Tipo	Origem		Destino		Tempo	
	Nó	Identificador	Nó	Identificador	Início	Fim
VoIP	N9	CBR0/UDP0	N15	Null1	1,0	9,0
VoD-HD	N11	CBR1/UDP2	N13	Null3	2,0	8,5
VoIP (2)	N12	CBR2/UDP4	N10	Null5	1,5	3,5
Backup	N14	FTP3/TCP6	N16	Sink7	1,5	3,5

Além dos fluxos do cenário 1, pode-se identificar na Tabela 11 que o nó 12 iniciará um fluxo UDP de VoIP (2) com destino ao nó 10 no intervalo de 1,5 até 3,5, enquanto o nó 14 iniciará um fluxo TCP de Backup com destino ao nó 16 no intervalo de 1,5 até 3,5.

A sobreposição de tempos pode ser identificada, conforme pode ser visto na Figura 61.

Figura 62 - Tempos do Cenário 2

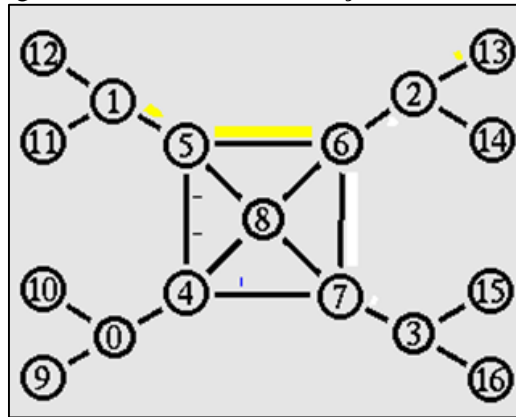


Na da Figura 61, a sobreposição dos tempos dos fluxos ocorre no intervalo 2,0 até 3,5.

#### 5.4.1 Cenário 2 - Execução 1

Uma primeira execução foi realizada para coleta dos resultados de uma rede baseada em melhor esforço comum. O encaminhamento padrão pode ser visto no NAM da Figura 62.

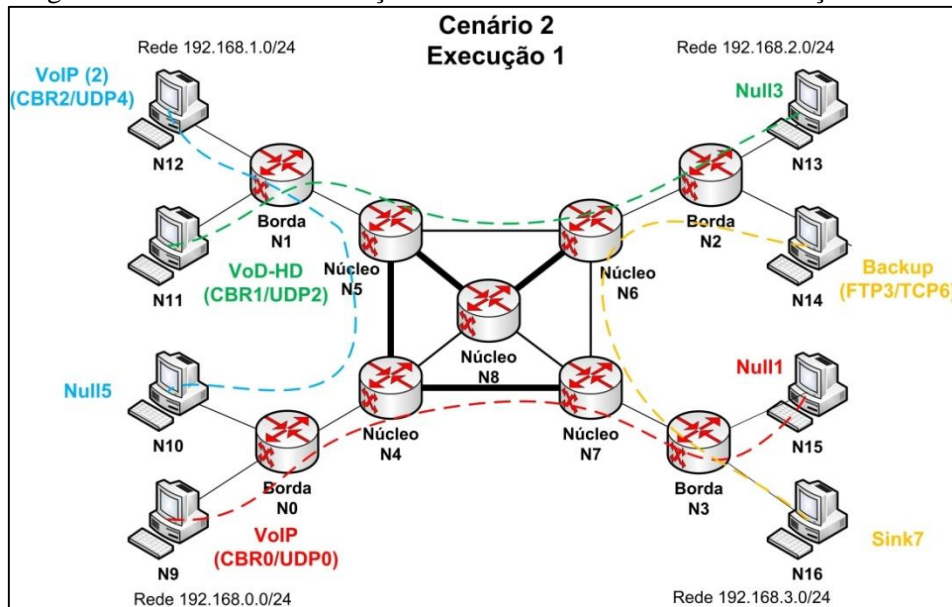
Figura 63 - Cenário 2 - Execução 1: Nam no momento 2,4



A Figura 62 representa o momento 2,4, sendo possível identificar, além dos fluxos do cenário 1, o fluxo de VoIP (2) em preto seguindo entre os nós 5 e 4 e o fluxo de Backup em branco seguindo pelos nós 6 e 7. Assim como no cenário 1, os fluxos seguiram a menor contagem de saltos para seus destinos, sendo que o fluxo VoIP (2) seguiu por um caminho com maior largura de banda e maior atraso, N1-N5-N4-N0, e o fluxo de Backup seguiu por um caminho com menor largura de banda e menor atraso, N2-N6-N7-N3.

O caminho seguido pelos fluxos esta representado na Figura 63.

Figura 64 - Cenário 2 - Execução 1: Encaminhamento melhor esforço



Na Figura 63, pode-se ver uma ilustração representando os fluxos existentes no NAM da Figura 62 da primeira execução, que devido à representação do instante da informação não permite visualização da totalidade do caminho.

As principais diferenças no arquivo TCL do Cenário 1 - Execução 1 para o Cenário 2 – Execução 2, podem ser vistas no Anexo K. Nele estão caracterizados os agentes e as aplicações dos fluxos VoIP (2) e Backup.

Os resultados desta primeira execução são apresentados na Tabela 12.

Tabela 12 - Resultados do Cenário 2 - Execução 1

<b>Fluxo</b>	<b>Atraso Fim a Fim</b>	<b>Bits Transmitidos</b>	<b>Largura de Banda</b>
VoIP	0,0110339 s	513280 bits	64,16kbps
VoD-HD	0,0599102 s	6896000 bits	1001,16 kbps
VoIP (2)	0,0110304 s	128000bits	64,6465 kbps
Backup	0,0430292 s	2205600 bits	991,855 kbps

Na Tabela 12 os valores coletados via AWK dos fluxos VoIP e VoD-HD se mantêm similares aos do Cenário 1. Quando aos fluxos introduzidos no Cenário 2, o VoIP (2) teve atraso de 0,0110304 s e largura de banda de 64,6465 kbps, dentro do esperado, mantendo o limite máximo da aplicação configurada, o destaque fica com o Backup que teve largura de banda de 991,855 kbps devido a limitação da banda do enlace pelo qual ele passou.

#### **5.4.2 Cenário 2 - Execução2**

Uma segunda execução do cenário definido na seção 5.4.1 foi realizada baseada no mesmo arquivo TCL base, com a adição neste arquivo de modificações de encaminhamento indicadas pelo arcabouço.

Os eventos do arcabouço ocorridos nesta segunda execução podem ser vistos no resumo da Tabela 13.

Tabela 13 - Resumo dos eventos de contexto no Cenário 2 - Execução 2

<b>Tempo (m)</b>	<b>Fluxo</b>	<b>Evento</b>	<b>Ação</b>
1,0	VoIP	Nova relação	Envio XML para arcabouço (mudança de contexto)
1,7	VoIP	Retorno arcabouço	Ativação de novo encaminhamento
1,5	VoIP (2)	Nova relação	Envio XML para arcabouço (mudança de contexto)
---	VoIP (2)	Sem arcabouço para a simulação (apenas interno)	Informações enviadas não passaram no QoC definido, e a relação não entrou no novo Contexto, sendo ignorada pela abordagem de encaminhamento
1,5	Backup	Nova relação	Envio XML para arcabouço (mudança de contexto)
---	Backup	Sem arcabouço para a simulação (apenas interno)	Informações enviadas não passaram no QoC definido, e a relação não entrou no novo Contexto, sendo ignorada pela abordagem de encaminhamento
2,0	VoD-HD	Nova relação	Envio XML para arcabouço (mudança de contexto)
2,7	VoD-HD	Retorno arcabouço	Ativação novo encaminhamento
8,5	VoD-HD	Fim Relação	Envio XML para arcabouço (mudança de contexto)
9,0	VoD-HD	Retorno arcabouço	Remoção de caminho
9,0	VoIP	Fim Relação	Envio XML para arcabouço (mudança de contexto)
9,5	VoIP	Retorno arcabouço	Remoção de caminho

Na Tabela 13 , além dos eventos já descritos no Cenário 1, podem ser identificados os eventos de novas relações VoIP (2) e Backup, que resultam no envio de arquivos XML para o arcabouço. Destaca-se o fato que ao o arcabouço verificar o QoC das informações enviadas, estar não foram aprovadas, não entrando no Contexto ativo, e portanto, não sendo utilizadas pela abordagem de contexto para tomada de decisão.

Um recorte das informações mais relevantes do XML enviado pelo N12 no tempo 1,5 pode ser visto na Figura 64.

Figura 65 - Recorte do XML enviado por N12 no tempo 1,5

```

<Relations>
  <Rel>
    <Collected_rID>1</Collected_rID>
    <Collected_Conected>192.168.0.10</Collected_Conected>
    <Collected_ApplicationName>Novo programa VoIP</Collected_ApplicationName>
    <Collected_ApplicationExecutable>novovoip.exe</Collected_ApplicationExecutable>
    <Collected_ApplicationVersion>1.0</Collected_ApplicationVersion>
    <Collected_SourcePort>60000</Collected_SourcePort>
    <Collected_DestinationPort>5555</Collected_DestinationPort>
    <Collected_TransportProtocol>UDP</Collected_TransportProtocol>
  </Rel>
</Relations>

```

Na Figura 64 são exibidas apenas as informações da relação ativa por estar nelas a não conformidade do QoC. No exemplo, foi identificada a aplicação, suas portas de comunicação, mas por se tratar de uma nova aplicação VoIP, ela ainda não faz parte da tabela APPLICATION, não sendo, portanto, possível ao arcabouço identificar que tipo de fluxo é este, e portanto não podendo tomar decisões sobre ele.

O cadastro deste XML é feito pelo *Handler* normalmente, mas devido a não existência desta aplicação, estas relações internas do banco não são criadas, sendo recusadas pela checagem na da tarefa VII, verificado o QoC para a aplicação, e conseqüentemente a tarefa IX, a relação é atualizada para Contexto, também não é concluída, estando este fluxo fora do contexto ativo.

Quanto ao fluxo de Backup, um recorte das informações mais relevantes do XML enviado pelo N14 no tempo 1,5 pode ser visto na Figura 65.

Figura 66 - Recorte do XML enviado por N14 no tempo 1,5

```

<Individuality>
  <Collected_Username>N14 User</Collected_Username>
  <Collected_Hostname>N14</Collected_Hostname>
  <Collected_Domainname>lab.unifacs.br</Collected_Domainname>
  <Collected_MAC>00-53-00-00-00-14</Collected_MAC>
  <Collected_IPv4>169.254.0.1</Collected_IPv4>
</Individuality>

```

Na Figura 65 são exibidas apenas as informações da identidade por estar nelas a não conformidade do QoC. No exemplo, o endereço IP informado pela entidade é um endereço de autoconfiguração da interface (RFC5735, 2010) que coletado por engano pelo agente, e portanto não é útil para encaminhamento da rede por não poder sofrer encaminhamento.

Devido as não validações de QoC dos fluxos VoIP (2) e Backup, a consulta no Contexto ativo permanece conforme Figura 66.

Figura 67 - Novo contexto do Cenário 1 com duas relações

HOST_ORIGEM	IP_ORIGEM	REDE_ORIGEM	MASK_ORIGEM	PORTA_ORIGEM	IP_DESTINO	PORTA_DESTINO
N15	192.168.0.9	192.168.0.0	255.255.255.0	10000	192.168.3.15	5243
N15	192.168.0.9	192.168.0.0	255.255.255.0	11321	192.168.2.13	9000

APLICACAO	APPL_VAZAO	ORDEM_VAZAO	APPL_ATRASO	ORDEM_ATRASO
VIBER	200	2	50	1
Cisco WebEX	800	1	400	2

Na Figura 66 é possível identificar apenas os dois fluxos do Cenário 1 que passaram pela validação do QoC e entraram no Contexto Ativo, sendo os dois novos fluxos, VoIP (2) e Backup totalmente ignorados pela abordagem de encaminhamento.

As reconfigurações dos caminhos MPLS seguem conforme procedimentos exemplificados no Cenário 1.

A segunda execução do Cenário 2 pode ser vista nas Figuras 67 e 68.

Figura 68 - Cenário 2 - Execução 2: NAM momento 2,2

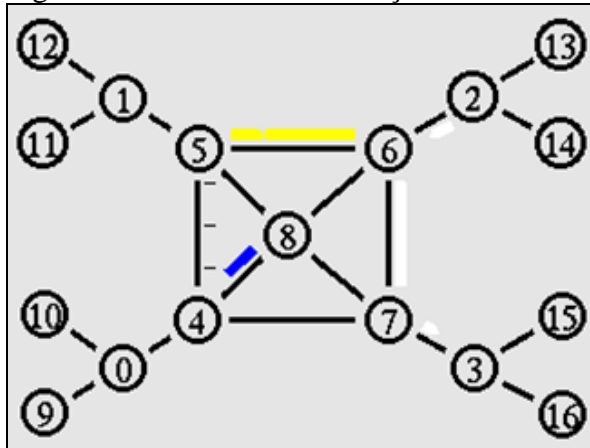
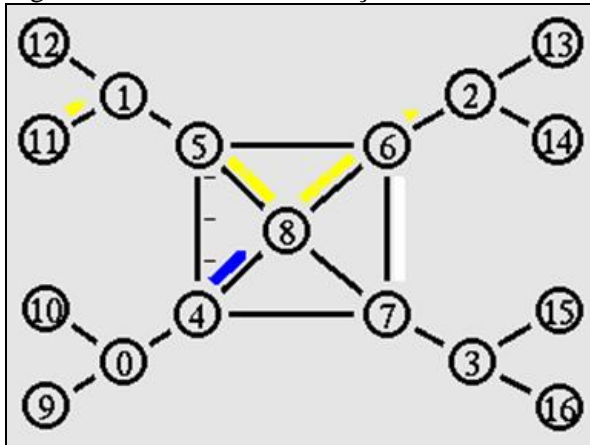


Figura 69 - Cenário 2 - Execução 2: NAM momento 3,4

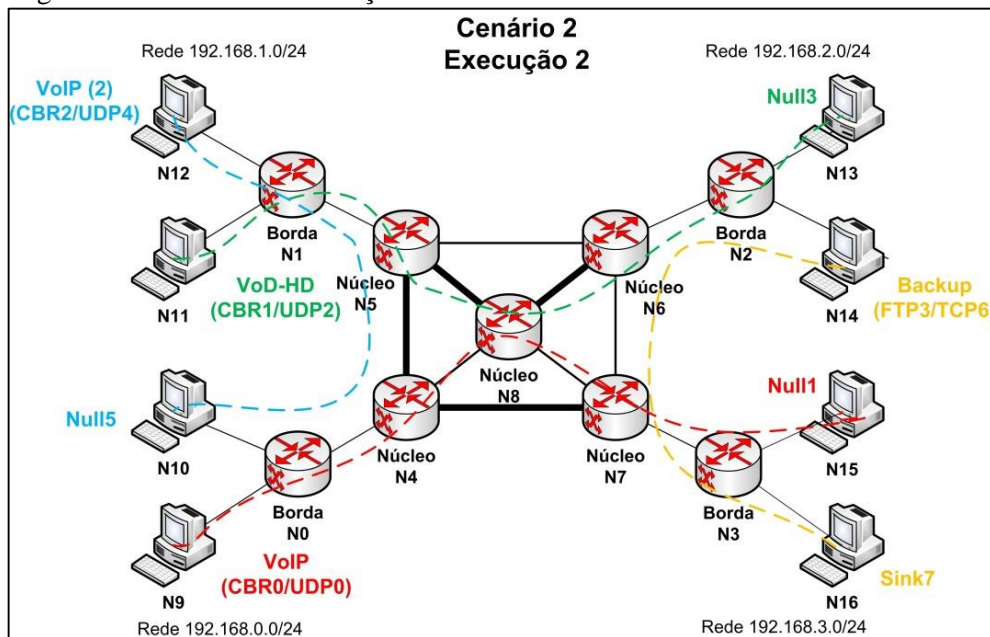




Na Figura 67 a simulação esta no momento 2,2, e é possível identificar os fluxos VoIP já no caminho indicado pelo arcabouço, o fluxo VoD-HD em seu caminho original antes da mudança, e os fluxos VoIP (2) seguindo também pelo caminho original N1\_N5\_N4\_N0 e o fluxo de Backup no caminho N140N2-N6-N7-N3. Na Figura 68 a simulação esta no momento 3,4 e já se identifica os fluxos VoIP e VoD-HD seguindo pelo caminho alternativo baseado em contexto, enquanto os fluxos de VoIP (2) e Backup continua no caminho original.

O resultado desta execução pode ser visto na Figura 69.

Figura 70 - Cenário 2 - Execução 2: Encaminhamento baseado em contexto



A Figura 69 representa os fluxos VoID e VoD seguindo pelos fluxos após modificação baseada em contexto e os fluxos VoIP (2) e Backup nos caminhos originais.

Os resultados desta execução 2 estão disponíveis na Tabela 14.

Tabela 14 - Resultados do Cenário 2 - Execução 2

<b>Fluxo</b>	<b>Atraso Fim a Fim</b>	<b>Transmitidos</b>	<b>Largura de Banda</b>
VoIP	0,00223764 s	513280 bits	64,16kbps
VoD-HD	0,0228405 s	11296000 bits	1712,35 kbps
VoIP (2)	0,0110304 s	128000 bits	64,6465 kbps
Backup	0,0430267 s	2205600 bits	991,855 kbps

Na Tabela 14 os fluxos VoIP e VoD apresentam comportamento de melhora do atraso e da banda, respectivamente, similar ao ocorrido no cenário 1. Divergente disso, os fluxos VoIP (2) e Backup não apresentaram melhoras devida a não modificação de seus caminhos pela não aprovação nas regras de QoS. O fluxo mais prejudicado é o Backup, que por suas



características de uso de banda variável (TANENBAUM, 2003), não pode utilizar a maior banda disponível em outros caminhos disponíveis.

### 5.5 CENÁRIO 3

Como já citado, este cenário tem carga e a complexidade altas, e o objetivo é demonstrar o funcionamento do arcabouço com um volume maior de fluxos, além das não aprovações de QoC, inclui encaminhamentos baseados em QoD e QoE. Para tanto, são mantidos os dois fluxos do cenário 2, e são incluídos mais dois fluxos que terão seu encaminhamento tratado baseado em QoD e QoE, conforme Tabela 15.

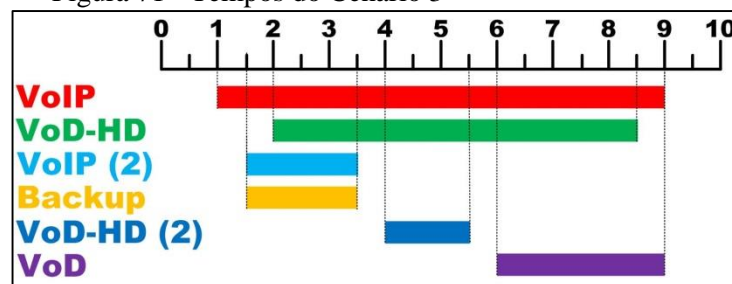
Tabela 15 - Cenário 3: Tempos dos fluxos

Tipo	Origem		Destino		Tempo	
	Nó	Identificador	Nó	Identificador	Início	Fim
VoIP	N9	CBR0/UDP0	N15	Null1	1,0	9,0
VoD-HD	N11	CBR1/UDP2	N13	Null3	2,0	8,5
VoIP (2)	N12	CBR2/UDP4	N10	Null5	1,5	3,5
Backup	N14	FTP3/TCP6	N16	Sink7	1,5	3,5
VoD-HD (2)	N16	CBR4/UDP8	N14	Null9	4,0	5,5
VoD	N11	CBR5/UDP10	N10	Null11	6,0	9,0

Além dos fluxos do cenário 2, pode-se identificar na Tabela 15 que o nó 16 iniciará um fluxo UDP de VoD-HD (2) com destino ao nó 14 no intervalo de 4,0 até 5,5, enquanto o nó 11 iniciará um fluxo UDP de VoD com destino ao nó 10 no intervalo de 6,0 até 9,0.

A sobreposição de tempos pode ser identificada, conforme pode ser visto na Figura 70.

Figura 71 - Tempos do Cenário 3



Conforme ilustrado na Figura 70, as sobreposições de tempos inseridas neste cenário 3 ocorrem entre o momento 4 e 5,5 com os fluxos VoIP e VoD-HD com o novo fluxo VoD-HD (2), e entre o momento 6 e 9 também com os fluxos VoIP e VoD-HD, agora com o novo fluxos VoD.

### 5.3.1 Cenário 3 - Execução 1

Uma primeira execução foi realizada para coleta dos resultados de uma rede baseada em melhor esforço comum. O encaminhamento padrão pode ser visto no NAM das Figuras 71 até 75.

Figura 72 - Cenário 3 - Execução 1: Nam no momento 2,4

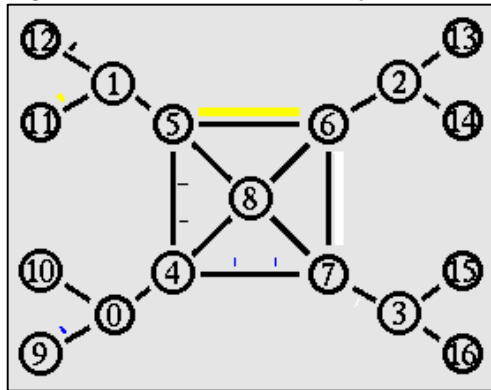


Figura 73 - Cenário 3 - Execução 1: Nam no momento 3,7

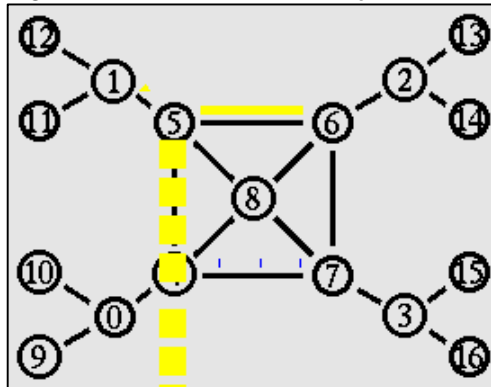


Figura 74 - Cenário 3 - Execução 1: Nam no momento 4,3

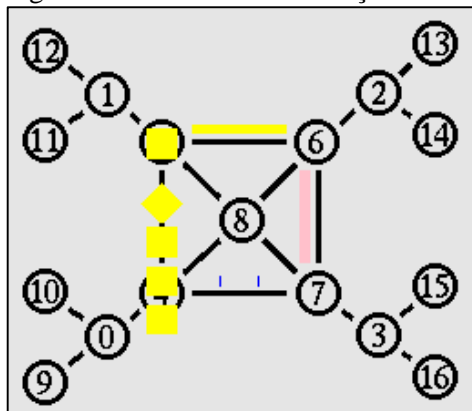


Figura 75 - Cenário 3 - Execução 1: Nam no momento 5,9

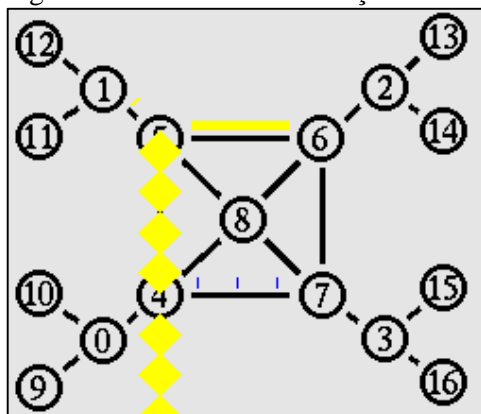
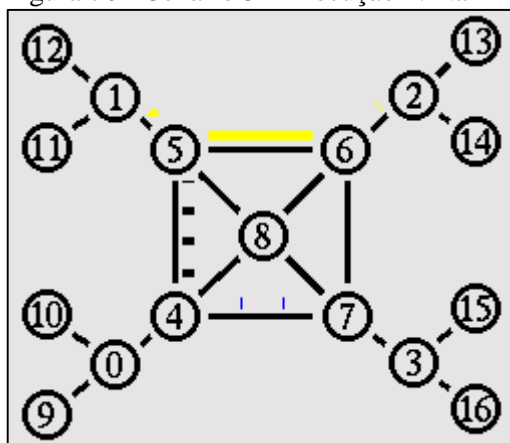


Figura 76 - Cenário 3 - Execução 1: Nam no momento 6,1



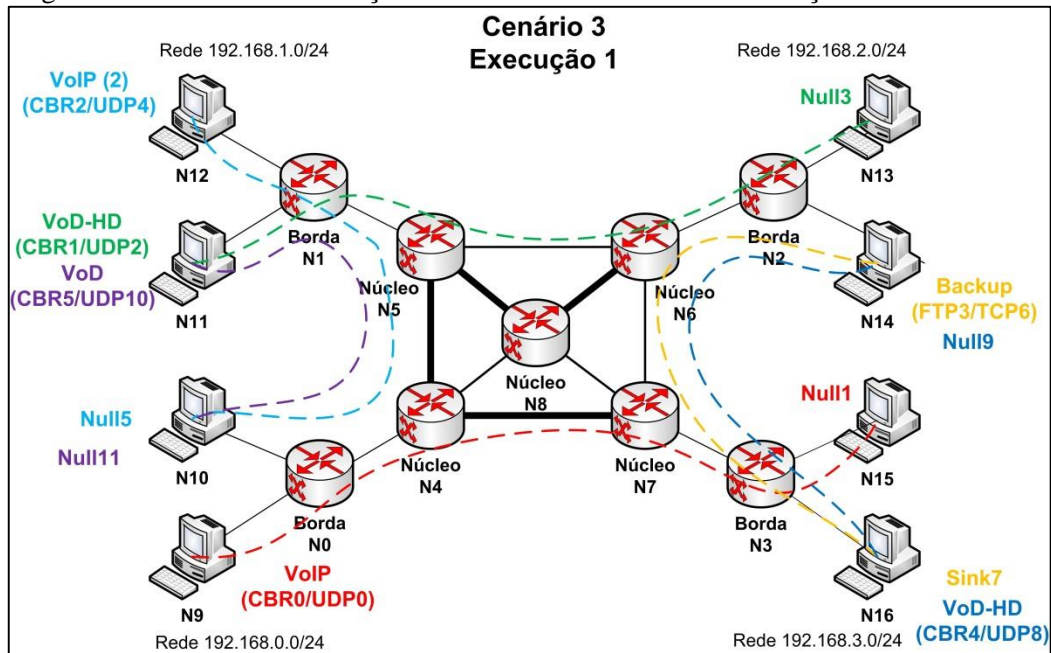
A Figura 71 ilustra o momento 2,4 sendo possível identificar os fluxos do cenário 2, VoIP (azul), VoD-HD (amarelo), VoIP (2) (preto) e Backup (branco). Enquanto a Figura 72 ilustra o momento 3,7, já com os fluxos VoIP (2) e Backup encerrados, apenas com os fluxos VoIP e VoD-HD em execução.

Na Figura 73 ilustra o momento 4,3 mantendo os fluxos VoIP e VoD-HD, acrescentando o novo fluxo VoD-HD(2) (rosa), iniciando no nó 16 com destino ao nó 14, passando pelo caminho N3-N7-N6-N2. Seguindo para Figura 74, é ilustrado o momento 5,9 com o fluxo VoD-HD (2) já encerrado não é mais exibido.

Por fim, na Figura 75 além dos fluxos VoIP e VoD-HD, é acrescentado o novo fluxo VoD (preto), iniciado no nó 11 com destino ao nó 10, passando pelo caminho N1-N5-N4-N0.

Os caminhos seguidos pelos fluxos estão representados na Figura 76.

Figura 77 - Cenário 3 - Execução 1: Encaminhamento melhor esforço



Na Figura 76, é apresentada uma ilustração descrevendo os fluxos existentes no NAM das Figuras 71 até 75 da primeira execução, que devido à representação do instante da informação não permite visualização da totalidade do caminho.

As diferenças no arquivo TCL do Cenário 2 - Execução 1 para este Cenário 3 – Execução1, podem ser vistas no diferenças no arquivo TCL do Cenário 2 - Execução 1 para este Cenário 3 – Execução1. Nele estão caracterizados os agentes e as aplicações dos fluxos VoD-HD (2) e VoD.

Os resultados desta primeira execução são apresentados na Tabela 16.

Tabela 16 - Resultados do Cenário 3 - Execução 1

Fluxo	Atraso Fim a Fim	Transmitidos	Largura de Banda
VoIP	0.0110339 s	513280 bits	64.16 kbps
VoD-HD	0.0599459 s	6896000 bits	1001.16 kbps
VoIP (2)	0.0110304 s	128000 bits	64.6465 kbps
Backup	0.0430292 s	2205600 bits	991.855 kbps
VoD-HD (2)	0.0540494 s	1896000 bits	1004.24 kbps
VoD	0.0122576 s	1568000 bits	522.806 kbps

Na Tabela 16 os valores coletados via AWK dos fluxos VoIP, VoD-HD, VoIP (2) e Backup se mantêm similares aos do Cenário 2. Quando aos fluxos introduzidos no Cenário 3, o VoHD (2) obteve atraso de 0.0540494 s e largura de banda de 1004.24 kbps, dentro do esperado, mantendo o limite máximo da disponibilidade de banda existente, não atingindo o

valor máximo configurado de 1,8 Mbps. Enquanto o VoD obteve atraso de 0.0122576 s e atingiu o máximo configurado para a aplicação, 520 Kbps.

### 5.3.2 Cenário 3 - Execução2

Uma segunda execução do cenário definido na seção 5.5.1 foi realizada baseada no mesmo arquivo TCL, com a adição neste arquivo de modificações de encaminhamento indicado pelo arcabouço.

Os eventos do arcabouço ocorridos nesta segunda execução podem ser vistos no resumo da Tabela 17.

Tabela 17 - Resumo dos eventos de contexto no Cenário 3 - Execução 2

Tempo (m)	Fluxo	Evento	Ação
1,0	VoIP	Nova relação	Envio XML para arcabouço (mudança de contexto)
1,5	VoIP (2)	Nova relação	Envio XML para arcabouço (mudança de contexto)
---	VoIP (2)	Sem arcabouço para a simulação (apenas interno)	Informações enviadas não passaram no QoC definido, e a relação não entrou no novo Contexto, sendo ignorada pela abordagem de encaminhamento
1,5	Backup	Nova relação	Envio XML para arcabouço (mudança de contexto)
---	Backup	Sem arcabouço para a simulação (apenas interno)	Informações enviadas não passaram no QoC definido, e a relação não entrou no novo Contexto, sendo ignorada pela abordagem de encaminhamento
1,7	VoIP	Retorno arcabouço	Ativação de novo encaminhamento
2,0	VoD-HD	Nova relação	Envio XML para arcabouço (mudança de contexto)
2,7	VoD-HD	Retorno arcabouço	Ativação novo encaminhamento
3,5	VoIP (2)	Fim Relação	Envio XML para arcabouço (mudança de contexto)
---	VoIP (2)	Sem mudança no Contexto ativo	Como esta relação não estava no Contexto ativo, a relação é desativada sem mudança no contexto ativo
3,5	Backup	Fim Relação	Envio XML para arcabouço (mudança de contexto)
---	Backup	Sem mudança no Contexto ativo	Como esta relação não estava no Contexto ativo, a relação é desativada sem mudança no contexto ativo
4,0	VoD-HD (2)	Nova relação	Envio XML para arcabouço (mudança de contexto)

Tempo (m)	Fluxo	Evento	Ação
---	VoD-HD (2)	Mudança do Contexto ativo, sem mudança de encaminhamento	Informações enviadas passaram no QoC, a relação entrou no Contexto ativo, mas no processamento do encaminhamento os requisitos de QoD não foram cumpridos e não foi gerado encaminhamento específico para este fluxo.
5,5	VoD-HD (2)	Fim Relação	Envio XML para arcabouço (mudança de contexto)
---	VoD-HD (2)	Mudança do Contexto ativo, sem mudança de encaminhamento	Remoção da relação do contexto ativo, sem remoção de regras de encaminhamento por não existirem.
6,0	VoD	Nova relação	Envio XML para arcabouço (mudança de contexto)
6,2	VoD	Usuário informa MOS	Envio XML para arcabouço (mudança de contexto)
7,0	VoD	Retorno arcabouço	Ativação novo encaminhamento
8,5	VoD-HD	Fim Relação	Envio XML para arcabouço (mudança de contexto)
9,0	VoD-HD	Retorno arcabouço	Remoção da relação do Contexto ativo, e remoção do caminho
9,0	VoIP	Fim Relação	Envio XML para arcabouço (mudança de contexto)
9,0	VoD	Fim Relação	Envio XML para arcabouço (mudança de contexto)
9,5	VoIP	Retorno arcabouço	Remoção da relação do Contexto ativo, e remoção do caminho
9,5	VoD	Retorno arcabouço	Remoção da relação do Contexto ativo, e remoção do caminho

Na Tabela 17, além dos eventos já descritos no Cenário 2, podem ser identificados os eventos de novas relações VoD-HD (2) e VoD, que resultam no envio de arquivos XML para o arcabouço. Ambos os fluxos foram aprovador no QoC e entraram no Contexto ativo. No entanto, no processamento do encaminhamento, como o fluxo VoD-HD (2) não cumpria os requisitos de QoD de sua aplicação, o mesmo não teve seu encaminhamento específico gerado, continuando no encaminhamento melhor esforço. Já o fluxo VoD, recebeu uma informação do MOS do usuário identificando uma percepção de qualidade de experiência (QoE) insatisfatória, indicando ao processamento de encaminhamento a necessidade de ajustes específicos.

Um recorte das informações mais relevantes do XML enviado pelo N16 no tempo 4,0 pode ser visto na Figura 77.

Figura 78 - Recorte do XML enviado por N16 no tempo 4,0

```

<Relations>
  <Rel>
    <Collected_rID>1</Collected_rID>
    <Collected_Conected>192.168.2.14</Collected_Conected>
    <Collected_ApplicationName>Cisco WebEX</Collected_ApplicationName>
    <Collected_ApplicationExecutable>webex.exe</Collected_ApplicationExecutable>
    <Collected_SourcePort>11321</Collected_SourcePort>
    <Collected_DestinationPort>9000</Collected_DestinationPort>
    <Collected_TransportProtocol>UDP</Collected_TransportProtocol>
  </Rel>
</Relations>

<QoD>
  <Collected_ProcessorCores>1</Collected_ProcessorCores>
  <Collected_ProcessorClock>1</Collected_ProcessorClock>
  <Collected_Memory>1</Collected_Memory>
</QoD>

```

Na Figura 77 são exibidas informações da relação ativa entre N16 e N14, sua aplicação ativa e características do dispositivo (QoD). Neste exemplo, a CPU tem um core e velocidade de processamento de 1 Ghz, e o dispositivo tem 1 Gb de memória.

O cadastro deste XML é feito pelo *Handler* normalmente, passando pelas fazer I até XI descritas no Cenário 1. Por se tratar de uma aplicação cadastrada do ambiente, esta relação entrará no Contexto ativo.

Conforme descrito no exemplo, o fluxo VoD-HD (2) foi originado em um dispositivo com 1 Ghz de CPU e 1 Gb de memória, e conforme especificação cadastrada da tabela APPLICATIONPARAMETER relacionada a aplicação WebEx, ela exige no mínimo velocidade de 2 Ghz de CPU (*clock*) e 2 GB de memória (CISCO2, 2015) , parâmetros 36 e 9 respectivamente da tabela PARAMETER\_APPL.

Quanto ao fluxo de VoD, um recorte das informações mais relevantes do XML enviado pelo N11 no tempo 6,2 pode ser visto na Figura 78.

Figura 79 - Recorte do XML enviado por N11 no tempo 6,2

```

<Relations>
  <Rel>
    <Collected_rID>1</Collected_rID>
    <Collected_Conected>192.168.0.11</Collected_Conected>
    <Collected_ApplicationName>Cisco WebEX</Collected_ApplicationName>
    <Collected_ApplicationExecutable>webex.exe</Collected_ApplicationExecutable>
    <Collected_SourcePort>11321</Collected_SourcePort>
    <Collected_DestinationPort>9000</Collected_DestinationPort>
    <Collected_TransportProtocol>UDP</Collected_TransportProtocol>
  </Rel>
</Relations>

<QoE>
  <Peer>
    <Collected_rID>1</Collected_rID>
    <Provided_MOSu>3</Provided_MOSu>
  </Peer>
</QoE>

<QoS>
  <Nic>
    <Collected_NIC_ID>1</Collected_NIC_ID>
    <Collected_Throughput>0,5</Collected_Throughput>
  </Nic>
</QoS>

```

Na Figura 78 são exibidas apenas as informações da relação ativa, do QoE informado pelo usuário, MOS=3, e de QoS, com largura de banda em uso de 0,5 Mbps.

Neste exemplo o XML também é cadastrado pelo *Handler* normalmente, passando pelas fazer I até XI descritas no Cenário 1. Também por se tratar de uma aplicação cadastrada do ambiente, esta relação entrará no Contexto ativo.

Destaca-se neste caso a existência da informação MOS=3, uma indicação do usuário (QoE) que sua experiência esta regular (ITU-T, 2015). Segundo análise de encaminhamento realizada (MUAKAD, 2015) o caminho indicado sai do requisito de banda de situações anteriores, e considera também a necessidade de melhoria do atraso.

As reconfigurações dos caminhos MPLS definidas pelo arcabouço para o Cenário 3 estão ilustradas na Figura 79.



Figura 80 - Cenário 3: Configuração de caminhos MPLS no NS2

```

# MPLS Tables
$ns at 0.5 "[$n2 get-module MPLS] ldp-trigger-by-withdraw 13 -1"
$ns at 0.5 "[$n3 get-module MPLS] ldp-trigger-by-withdraw 15 -1"
$ns at 0.5 "[$n0 get-module MPLS] ldp-trigger-by-withdraw 11 -1"

# VoIP
$ns at 1.5 "[$n0 get-module MPLS] make-explicit-route 3 0_4_8_7_3 1015 -1"
$ns at 1.7 "[$n0 get-module MPLS] flow-erlsp-install 15 -1 1015"
$ns at 9.0 "[$n0 get-module MPLS] ldp-trigger-by-release 13 1015"

# VoD-HD
$ns at 2.5 "[$n1 get-module MPLS] make-explicit-route 2 1_5_8_6_2 1013 -1"
$ns at 3.0 "[$n1 get-module MPLS] flow-erlsp-install 13 -1 1013"
$ns at 9.5 "[$n1 get-module MPLS] ldp-trigger-by-release 13 1001"

# VoD
$ns at 6.5 "[$n1 get-module MPLS] make-explicit-route 0 1_5_6_7_8_4_0 1010 -1"
$ns at 7.0 "[$n1 get-module MPLS] flow-erlsp-install 10 -1 1010"
$ns at 9.5 "[$n1 get-module MPLS] ldp-trigger-by-release 10 1010"

```

Na Figura 79 esta ilustrada a configuração dos caminhos MPLS da simulação, tendo os fluxos VoIP encaminhado fixado por N0-N4-N8-N7-N3, o fluxo VoD-HD seguindo pelo caminho N1-N5-N8-N6-N2, e o fluxo VoD seguindo pelo caminho N1-N5-N6-N7-N8-N4-N0.

A execução após reconfiguração do MPLS pode ser vista nas Figuras 80 até 83.

Figura 81 - Cenário 3 - Execução 2: Nam no momento 3,4

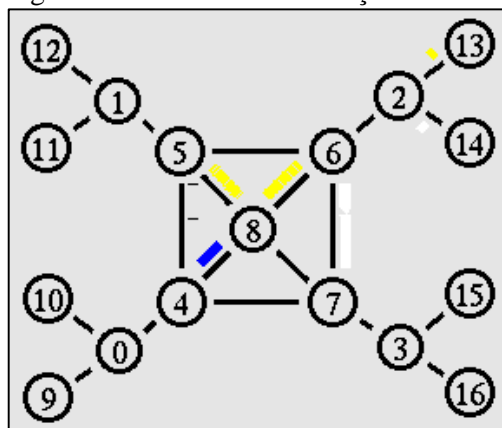


Figura 82 - Cenário 3 - Execução 2: Nam no momento 4,3

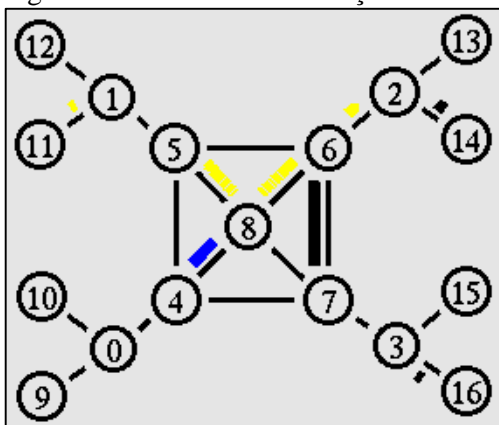


Figura 83 - Cenário 3 - Execução 2: Nam no momento 6,3

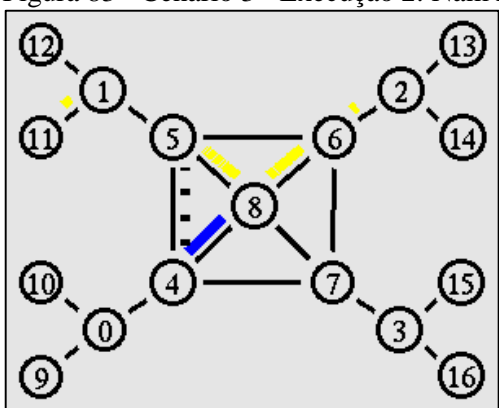
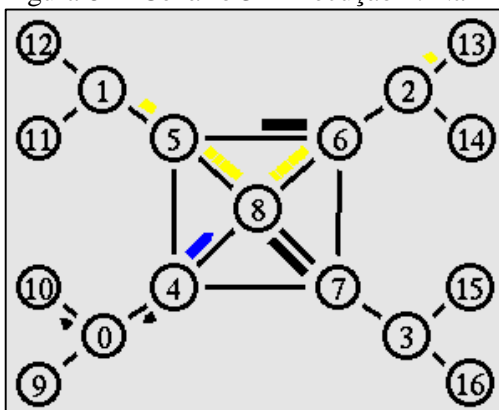


Figura 84 - Cenário 3 - Execução 2: Nam no momento 7,3



A Figura 80 ilustra o momento 3,4 e identifica os fluxos VoIP em amarelo e VoD-HD em azul, seguindo os caminhos definidos pelo arcabouço, e os fluxos VoIP (2) em preto e Backup em braço, ambos seguindo caminhos baseados em melhor esforço, conforme explicado no Cenário 2 – Execução 2.

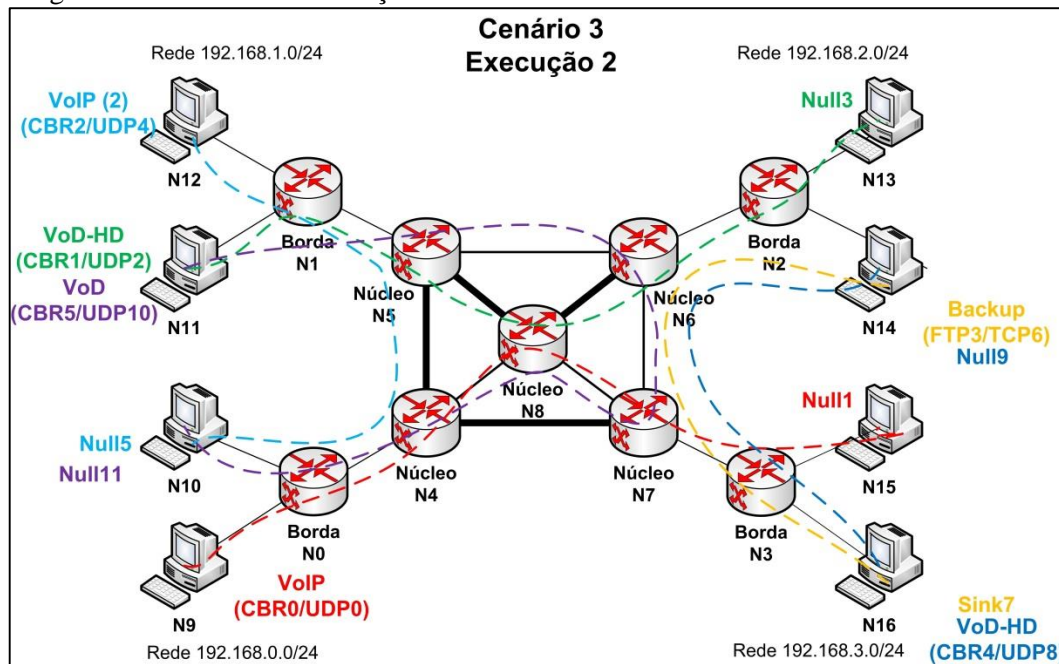
A Figura 81 ilustra o momento 4,3 e exibe os fluxos VoIP e VoD-HD , e apresenta o novo fluxo VoD-HD (2) entre os nós N16 e N14 seguindo pelo caminho baseado em melhor esforço por N3-N7-N6-N2.

Na Figura 82 o fluxo VoD-HD (2) já não existe, e pode-se identificar o novo fluxo VoD, também em preto, entre N11 e N10 seguindo seu caminho baseado em melhor esforço por N1-N5-N4-N0.

Por fim, na Figura 83 esta ilustrando o novo caminho seguido pelo fluxo VoD baseado na recomendação do arcabouço, seguindo pelo caminho N1-N5-N6-N7-N8-N4-N0.

O resultado desta execução pode ser visto na Figura 84.

Figura 85 - Cenário 3 - Execução 2: Encaminhamento baseado em contexto



A Figura 84 representa os fluxos ilustrados pelo NAM das Figura 80 até 83, que devido à representação do instante da informação não permite visualização da totalidade do caminho.

Os resultados desta execução 2 estão disponíveis na Tabela 18.

Tabela 18 - Resultados do Cenário 3 - Execução 2

<b>Fluxo</b>	<b>Atraso Fim a Fim</b>	<b>Transmitidos</b>	<b>Largura de Banda</b>
VoIP	0,00223764 s	513280 bits	64,16 kbps
VoD-HD	0,022844 s	11296000 bits	1712,35 kbps
VoIP (2)	0,0110304 s	128000 bits	64,6465 kbps
Backup	0,0430267 s	2205600 bits	991,855 kbps
VoD-HD (2)	0,0540577 s	1896000 bits	1004,24 kbps
VoD	0,00703167 s	1568000 bits	525,963 kbps

Na Tabela 18 os fluxos VoIP, VoD-HD, VoIP (2) e Backup apresentam resultados similares a execução 2 do cenário 2 conforme. Destaca-se na tabela o resultado VoD-HD (2) que manteve resultados similares à execução 1 do cenário 3 devido ao arcabouço não ter sugerido sua mudança de encaminhamento, mantendo o caminho melhor esforço anterior. Quanto ao fluxo VoD, este foi direcionado para um caminho com largura de banda suficiente para seu fluxo, e com menor atraso, possibilitando a manutenção de sua largura de banda em patamares próximos, mas melhorando seu atraso fim a fim que saiu de aproximadamente 0,0122s para aproximadamente 0,0070s.

#### 5.4 DISCUSSÕES

As simulações realizadas neste capítulo demonstram o funcionamento da solução proposta, mais especificamente o processamento pela abordagem de contexto das informações coletadas pelos agentes, sua validação com as regras e políticas de QoC e disponibilização ou não das informações para a abordagem de encaminhamento através de sua visão de contexto ativo.

Foram exemplificadas a representação de informações das entidades que permitiram sua individualização, identificação das relações e atividades ativas e características de QoS, QoD e QoE enviadas pelos agentes para o arcabouço.

Foi demonstrado também o funcionamento da abordagem de encaminhamento do arcabouço proposto, considerando seus resultados para definição do encaminhamento, sem entretanto detalhar seu funcionamento.

Os resultados coletados comparam o funcionamento das redes baseadas em encaminhamento melhor esforço, tipicamente vetor distância, com o funcionamento das mesmas redes considerando as indicações de encaminhamento fornecidas pelo arcabouço.

Nas coletas de resultados dos fluxos contemplados pelo encaminhamento do arcabouço pode-se comprovar a diminuição do atraso fim a fim para as aplicações que necessitam de menor atraso e aumento da largura de banda utilizada para a aplicações que se beneficiam desta característica da rede.

## 5.5 CONCLUSÃO

Neste capítulo foi demonstrado que o arcabouço proposto é capaz de armazenar e processar informações de contexto customizáveis e aplicar regras de validação para disponibilizar apenas as que atendam a critérios de qualidade que possibilitem a abordagem de encaminhamento processar a escolha de caminhos na rede.

A demonstração do arcabouço considerou escolhas de caminhos com características distintas de largura de banda e atraso, demonstrando nos resultados coletados a mudança nos valores conforme a característica priorizada, maior banda ou menos atraso.

## 6 CONCLUSÕES E PERSPECTIVAS FUTURAS

Foi apresentada nesta dissertação a proposta de um arcabouço que objetiva viabilizar o encaminhamento adaptativo baseado em contexto para redes de computadores, o *Context-aware Adaptive Routing Framework* (CAARF), mas especificamente seu módulo de gerência de contexto.

As conclusões gerais deste trabalho são apresentadas na seção 6.1, onde é apresentado um resumo do trabalho realizado, seus objetivos propostos e alcançados, alguns aspectos do desenvolvimento, vantagens e desvantagens da solução proposta, a relevância do trabalho realizado e uma análise do aprendizado obtido. E, na seção 6.2 são apresentadas as perspectivas futuras.

### 6.1 CONCLUSÕES

Esta dissertação propôs a discussão do uso de informações de contexto para tomada de decisão no encaminhamento adaptativo em redes de computadores. Para tanto, foi discutida a necessidade de qualificar as informações de contexto submetidas ao arcabouço, disponibilizando apenas as aprovadas em validações da qualidade das informações de contexto (QoC).

Para isso, foi proposto um arcabouço de encaminhamento adaptativo baseado em contexto, o *Context-aware Adaptive Routing Framework* (CAARF), e foram definidas as funcionalidades dos módulos *ContextHandler* e do *ContextModel Management*, ambos integrantes da abordagem de contexto, o *Context Approach*. Considerando ainda que estes módulos recebem arquivos *Markup Descriptions* provenientes do *Context Agent* em desenvolvimento, foram definidos os eventos a ser enviados pelo *Agent* que repercutem em interações com o *Handler*. Na outra extremidade do uso das informações de contexto, foram definidas as notificações a ser enviadas para tratamento da abordagem de encaminhamento, *Forwarding Approach*, não sendo detalhado seu funcionamento neste trabalho.

Em seguida, foi definido o modelo de contexto, *ContextModel*, que relaciona os atributos das entidades envolvidas, seus parâmetros de qualidade de serviço (QoS), qualidade de dispositivo (QoD), qualidade de experiência (QoE), e qualifica estes dados baseados em políticas e regras de qualidade de contexto (QoC). Desta forma, a abordagem de

encaminhamento só visualiza informações já validadas e consideradas úteis para tomada de decisão de encaminhamento.

Quanto à persistência das informações de contexto, foram definidas neste trabalho a estrutura dos arquivos *Markup Descriptions* baseados em XML enviados do *Agent* para o arcabouço, e a modelagem do bando de dados de informações de contexto, *ContextDatabase*, usado pelo arcabouço para armazenamento das informações de contexto em uso pelo ambiente, e ponto de interconexão entre as abordagem de contexto e abordagem de encaminhamento através de sua visão de dados, a *ContextModelView*.

Com o objetivo de validar o arcabouço proposto e o tratamento das informações de contexto, foram realizadas simulações da execução de uma rede com encaminhamento melhor esforço e comparados aos resultados obtidos na execução da mesma rede com encaminhamento baseado na tomada de decisão de contexto proposta pelo arcabouço CAARF. Os cenários utilizados nas simulações tiveram sua complexidade gradada para demonstrar o funcionamento básico do arcabouço focado em contexto e QoS, depois para demonstrar a recusa de informações de contexto baseadas em políticas de QoC, e por fim o uso de informações contextuais para encaminhamento utilizando critérios de QoD e QoE com MOS.

Nos resultados, demonstrou-se melhora de *performance* com o uso do encaminhamento baseado em contexto proposto pelo arcabouço CAARF, sendo os resultados mais expressivos quando a aplicação em avaliação faz uso elevando de banda, e com resultados mais discretos quando a aplicação requisita menor atraso de rede, ambos focados em QoS. Esta diferença de resultados pode estar ligada ao fato da topologia proposta para a simulação ter apenas oito nós em sua rede de encaminhamento, não conseguindo evidenciar de forma expressiva o ganho com a redução do atraso.

Outro ponto que mostrou resultado positivo foi na avaliação do encaminhamento quando o critério utilizado foi baseado em QoD. No caso avaliado, seguindo apenas os parâmetros de encaminhamento da aplicação, o arcabouço deveria sugerir um caminho com maior largura de banda disponível, permitindo a vazão máxima requerida pela aplicação. No entanto, na avaliação dos parâmetros de QoD, constatou-se que os requisitos mínimos do dispositivo não atendiam a aplicação, portanto não sendo necessário modificar o encaminhar pois considera-se que sem atender aos requisitos de QoD a aplicação não atinja seu desempenho adequado.

O último exemplo de encaminhamento apresentado deu ênfase à experiência do usuário (QoE), sendo uma determinada aplicação encaminhada por padrão por um caminho com perfil de maior largura de banda, e o atraso um fator secundário em sua avaliação. No entanto, depois de coleta de avaliação do usuário (MOS), considerando a qualidade obtida inadequada, o encaminhamento mudou o caminho padrão e colocou como critério mais relevante o atraso, deixando a largura de banda como avaliação acessória.

Foram os objetivos iniciais, propor experimentalmente uma solução de encaminhamento adaptativo baseado em informações de contexto para redes convergentes, um modelo de contexto que represente dispositivos, aplicações e usuários de redes convergentes, e elencar informações de contexto relevantes para o encaminhamento de fluxos. E ainda, como objetivos específicos, propor regras de Qualidade de Contexto (QoC) a serem aplicadas nas informações contextuais a fim de melhorar os resultados obtidos pela solução de encaminhamento baseada em contexto, definir a estrutura interna, módulos, ações e notificações necessárias para o módulo de contexto da solução proposta, desde a coleta até a entrega via base de contexto.

Desta forma, conforme apresentado nesta conclusão, os objetivos propostos foram alcançados, com a definição da arquitetura, definição dos módulos de contexto, definição da persistência em XML e da persistência do repositório em banco relacional, o *ContextModelDatabase*, definição das regras de QoC, simulações e análise dos resultados obtidos demonstrando o funcionamento do arcabouço proposto.

No entanto, foram encontradas dificuldades durante o desenvolvimento deste trabalho, principalmente na definição do ambiente de simulação e nas dificuldades de automatização total das simulações devido à necessidade de desenvolvimento de pilhas de protocolos completas. Tais dificuldades tornaram necessária a abordagem adotada nas simulações deste trabalho, utilizando os arquivos de *log* gerados pelo simulador para análise dos dados e programação de eventos que alterassem a simulação conforme indicações do arcabouço, gerando resultado similar a sua utilização.

São consideradas vantagens do arcabouço proposto sua flexibilidade para extensões do modelo de dados e consequentes extensões de sua persistência de dados. Da mesma forma, a sua separação entre os módulos que o compõem, sendo possível o desenvolvimento de *Agents* ou *Handlers* variados que se comunicam com o ambiente através de arquivos configuráveis e entradas em base de dados. A separação das funcionalidades do *ContextModel Management*



do restante da abordagem de encaminhamento é outra vantagem, tendo como ponto em comum apenas a visão do modelo de contexto, *ContextModelView*.

Porém, esta separação entre os módulos e esta possibilidade de extensão de sua persistência de dados pode criar um ambiente complexo e com muitas variáveis a serem tratadas para a obtenção de resultados úteis, necessitando atenção em manter seu mapeamento de funções, eventos e estruturas de informação devidamente atualizadas entre os desenvolvedores envolvidos.

Analisando o trabalho aqui apresentado, considera-se sua relevância pela discussão do encaminhamento em redes de computadores baseado em informações de contexto, e propor formas de gerenciar estas informações desde sua coleta, passando pela validação com políticas e regras que visam manter a qualidade das informações contextuais em uso, até sua disponibilização para tomada de decisão.

Por fim, o aprendizado obtido neste trabalho ampliou horizontes quanto às possibilidades diversas de encaminhamento em redes de computadores, saindo dos tradicionais métodos e algoritmos existentes, e considerando um leque maior de possibilidades a serem levadas em consideração baseadas no contexto das entidades envolvidas na comunicação.

## 6.2 PERSPECTIVAS FUTURAS

Esta seção identifica alguns trabalhos futuros sugeridos para dar continuidade à pesquisa aqui apresentada, visando propor ajustes e extensões as suas funcionalidades e recursos:

- Conforme mencionado no capítulo 3, podem-se avaliar critérios para priorização das notificações a serem enviadas do *NotificationScheduling* ao *ContextProcessing*;
- Como discutido no capítulo 3, no item que trata das regras de QoC, pode-se avaliar a eficiência de se utilizar regras de QoC genéricas para qualquer tipo de rede comparando a utilização de regras de QoC específicas para o tipo de rede e da análise de encaminhamento que vai fazer uso de suas informações resultantes;
- Apesar do *Context Agent* não ser foco deste trabalho, são relacionados eventos deste que podem influenciar na abordagem de contexto pesquisada aqui. Desta forma, no item Eventos da Gestão de Contexto, são relacionados aos eventos do agente que indicam a coleta de informações sobre conexões remotas de rede. Cabe avaliar se

todas as conexões remotas devem ser analisadas, ou se é válido prover inteligência adicional ao agente para só coletar informações de algumas aplicações específicas, diminuindo assim sua carga de trabalho, e conseqüentemente gerando menos informações de contexto que não serão utilizadas. Além de ajustes nos demais eventos do agente;

- Outro ponto relevante para novas pesquisas são as informações das coletas e suas relações, conforme discutido no Capítulo 3, e ainda sua relação de relevância com o tipo de tomada de decisão de encaminhamento que fará uso destas informações;
- Existe ainda a necessidade de inclusão no arcabouço de mecanismos de descoberta automatizada de caminhos existentes entre as redes interconectadas por ele, diminuindo a atuação do administrador de redes, e evitando erros nas configurações;
- Por fim, uma extensão do arcabouço que aborda apenas enfileiramento já está em desenvolvimento (SILVA, 2015), mas a extensão para tratar encaminhamento e enfileiramento simultaneamente pode resultar em uma solução mais ampla no tratamento dos fluxos e mais aderente as necessidades das redes convergentes atuais.

## REFERÊNCIAS

- ALRESHOODI, M.; WOODS, J. Survey on QoE\ QoS Correlation Models for Multimedia Services. **International Journal of Distributed and Parallel Systems (IJDPS)**, v.4, n.3, may 2013. arXiv:1306.0221.
- AWK Scripts for NS2 to process data from Trace Files. 2015. Disponível em: <<http://www.nsnam.com/2013/03/awk-scripts-for-ns2-to-process-data.html>>. Acesso em: 3 jun. 2015.
- BORBORUAH G. ; NANDI, G. A Study on Large Scale Network Simulators. **International Journal of Computer Science and Information Technologies (IJCSIT)**, v. 5, n.6, 2014, 7318-7322. ISSN: 0975-9646
- BUCHHOLZ, T.; SCHIFFERS, M. Quality of context: what it is and why we need it. In: WORKSHOP OF THE OPENVIEW UNIVERSITY ASSOCIATION (OVUA'03), 3., 2003. **Paper...** 2003.
- OLIVEIRA, A. L. C. et al. Context-aware framework for adaptive routing. In: INTERNATIONAL WORKSHOP ON ADVANCES IN ICP INFRASTRUCTURES AND SERVICES, 3., 2014, Miami, EUA. **Proceedings...** 2014.
- CHENNAKESHAVULU, K.; PRASANTH, S. K. Handling Traffic Dynamics in Multipath Routing Topology. **International Journal of Computer Trends and Technology**, v.4, n.3, 2013. ISSN: 2231-2803
- CHIOCCHETTI, R. et. al. **INFORM**: a dynamic INterest FORwarding Mechanism for Information Centric Networking. Hong Kong, China: ICN'13, 2013.
- CISCO. Cisco. **Voice Over IP - Per Call Bandwidth Consumption**. Disponível em: <<http://www.cisco.com/c/en/us/support/docs/voice/voice-quality/7934-bw-width-consume.pdf>>. Acesso em: 3 jun. 2015.
- CISCO2. Cisco. **Cisco WebEx Network Bandwidth - White Paper**. Disponível em: <[http://www.cisco.com/c/en/us/products/collateral/conferencing/webex-meeting-center/white\\_paper\\_c11-691351.pdf](http://www.cisco.com/c/en/us/products/collateral/conferencing/webex-meeting-center/white_paper_c11-691351.pdf)>. Acesso em: 1 jul. 2015.
- CLOSE, D. B. et al. **The AWK Manual**. Edition 1.0 (1995). Disponível em: <<http://www.staff.science.uu.nl/~oostr102/docs/nawk/nawkA4.pdf>>. Acesso em: 3 jun. 2015.
- CSV. **Common Format and MIME Type for Comma-Separated Values (CSV) Files**. Disponível em: <<https://tools.ietf.org/html/rfc4180>>. Acesso em: 3 jun. 2015.
- DEY, A. K.; ABOWD, G. D.; SALBER, D. A Conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. **Journal Human-Computer Interaction archive**, v.16, n.2, p. 97-166, 2001.
- EL-GENDY, M.A.; BOSE, A.; SHIN, K.G. Evolution of the internet qos and support for soft real-time applications. **Proceedings of the IEEE**, v.91, n. 7, p.1086 – 1104, 2003.
- FOROUZAN, B. A. **Comunicação de dados e redes de computadores**. São Paulo: Bookman, 2006. ISBN 978-85-363-0614-8.

FORTZ, B.; REXFORD, J.; THORUP, M. Traffic engineering with traditional IP routing protocols. **IEEE Communications Magazine**, 2002.

G711. **ITU-T Recommendation G.711. G.711 : Pulse code modulation (PCM) of voice frequencies.** Disponível em: <<https://www.itu.int/rec/T-REC-H.264-201402-I/en>>. Approved in 1988-11. Acesso em: 3 jun. 2015.

GUPTA, S. G., et al. Open-source network simulation tools: an overview. **International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)**, v. 2, n.4, apr. 2013. ISSN: 2278 – 1323

H264. ITU-T Recommendation H.264. SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS. Disponível em: <<https://www.itu.int/rec/T-REC-H.264-201402-I/en>>. Approved in 2014-02. Acesso em: 3 jun. 2015.

ITU-T Recommendation P.800. Methods for subjective determination of transmission quality. 2015. Disponível em: <<https://www.itu.int/rec/T-REC-P.800-199608-I/en>>. Acesso em: 3 jun. 2015.

JAWHAR, C. How MPLS is Implemented in NS. IRISA 2002. 2015. Disponível em: <<http://boudania.free.fr/research/mmt/stages/mplspimsm/1.%20Repport/PIMSM-MPLS-oNS.doc>>. Acesso em: 3 jun. 2015.

JSIM. [Home Page]. Disponível em: <<http://www.physiome.org/jsim/>>. Acesso em: 3 jun. 2015.

JSON. **Introducing JSON.** Disponível em: <<http://json.org/>>. Acesso em: 3 jun. 2015.

KARTHIGA, S; BALAMURUGAN, M. S. Traffic engineering system based on adaptative multipath routing. **International Journal of Emerging Technology and Advanced Engineering**, v. 3, n.2, 2013.

KUROSE, J.; ROSS, K. W. Redes de computadores e a internet - uma abordagem top-down. 5.ed. São Paulo: Pearson, 2010. ISBN 978-85-88639-97-3.

LAGHARI, K. U. R.; CONNELLY, K. Toward total quality of experience: A QoE model in a communication ecosystem. **Communications Magazine IEEE**, v.50, n.4, p.58-65, 2012.

LÓSCIO, B. F. **Armazenamento de dados.** Disponível em: <[http://www.w3c.br/pub/Agenda/PalestraCursoParaEntidadesPublicasDaAmericaLatinaECaribeSobreDadosAbertos2013PalestraSobreArmazenamentoDeDados/Armazenamento\\_de\\_dado\\_s.pdf](http://www.w3c.br/pub/Agenda/PalestraCursoParaEntidadesPublicasDaAmericaLatinaECaribeSobreDadosAbertos2013PalestraSobreArmazenamentoDeDados/Armazenamento_de_dado_s.pdf)>. Acesso em: 3 jun. 2015.

MASCOLO, C.; MUSOLESI, M. SCAR: Contextaware Adaptive Routing in Delay Tolerant Mobile Sensor Networks. In: INTERNATIONAL CONFERENCE ON WIRELESS COMMUNICATIONS AND MOBILE COMPUTING. IWCMC '06. 2006. **Proceedings...** 2006.

MITRA, K.; ZASLAVSKY, A.; ÅHLUND, C. A probabilistic context-aware approach for quality of experience measurement in pervasive systems. In: ACM SYMPOSIUM ON APPLIED COMPUTING, 2011. **Proceedings...** 2011.

MODEL for Tabular Data and Metadata on the Web. Disponível em <<http://www.w3.org/TR/tabular-data-model/>>. Acesso em: 16 jul. 2015.

MU et al. An adaptive routing optimization and energy-balancing algorithm in zigbee hierarchical networks. **EURASIP Journal on Wireless Communications and Networking**, v.43, 2014.

MUAKAD, C. F. J. **Gerencia de encaminhamento adaptativo baseado em contexto**. 2015. Dissertação (Mestrado)- Universidade Salvador – UNIFACS. Salvador, 2015.

MUSOLESI, M.; HAILES, S.; MASCOLO, C. Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks. Published in World of Wireless Mobile and Multimedia Networks. (WoWMoM 2005). In: IEEE INTERNATIONAL SYMPOSIUM, 6., 2005, Taormina, Giardini Naxos. **Proceedings...** 2005. ISBN 0-7695-2342-0.

MYSQLCommunity Edition. Disponível em: <<http://www.mysql.com/products/community/>>. Acesso em: 3 jun. 2015.

NAM: Network Animator. Disponível em: <<http://www.isi.edu/nsnam/nam/>>. Acesso em: 3 jun. 2015.

NAZARIO, D. C.; DANTAS, M. A. R.; TODESCO, J. L. Taxonomia das publicações sobre Qualidade de Contexto. **Sustainable Business International Journal**, n.20, 2012.

NETSIM. TETCOS. NetSim: Network Simulator and Emulator. Disponível em:<<http://tetcos.com/>>. Acesso em: 3 jun. 2015.

NICOLA, M.; JOHN, J. XML parsing: a threat to database performance. Pin: INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT,12., 2003, New Orleans, LA, USA. **Proceedings...** 2003

NS2. The Network Simulator - ns-2. Disponível em: <<http://www.isi.edu/nsnam/ns/>>. Acesso em: 3 jun. 2015.

NS3. Disponível em:<<https://www.nsnam.org/>>. Acesso em: 3 jun. 2015.

NSNAM. Disponível em: <<http://sourceforge.net/projects/nsnam/>>. Acesso em: 3 jun. 2015.

OLIVEIRA, J. M.; LINS, R. D.; MENDONÇA, R. **Redes MPLS: fundamentos e aplicações**. [S.l.]: Brasport, 2012. ISBN:9788574525396.

OMNeT++. Disponível em: <<https://omnetpp.org/>>. Acesso em: 3 jun. 2015.

ONF - OPEN NETWORKING FOUNDATION. Disponível em: <<https://www.opennetworking.org/>>. Acesso em: 3 jun. 2015.

OPEN KNOWLEDGE.Formatos de Arquivo. Disponível em: <[http://opendatahandbook.org/guide/pt\\_BR/appendices/file-formats/](http://opendatahandbook.org/guide/pt_BR/appendices/file-formats/)>. Acesso em: 3 jun. 2015.

OTCL. Disponível em: <<http://otcl-tclcl.sourceforge.net/otcl/>>. Acesso em: 3 jun. 2015.

PETZ, A. et al. An Architecture for Context-Aware Adaptation of Routing in Delay-Tolerant Networks. In: EXTREME CONFERENCE ON COMMUNICATION, 4., 2012. Zurich, Switzerland. **Proceedings...** 2012.

PRESSMAN, R. S. **Engenharia de Software**. 6. Ed. São Paulo; Editora McGraw-Hill, 2006, ISBN: 8586804576

QUALNET. **QualNet | SCALABLE Network Technologies**. Disponível em: <<http://web.scalable-networks.com/content/qualnet>>. Acesso em: 3 jun. 2015.

RDF. **Resource Description Framework (RDF)**. Disponível em: <<http://www.w3.org/RDF/>>. Acesso em: 3 jun. 2015.

REAL. **The REAL Network Simulator**. Disponível em: <<http://www.cs.cornell.edu/skeshav/real/>>. Acesso em: 3 jun. 2015.

RFC2205. **Network Working Group. Resource ReSerVation Protocol (RSVP)**. RFC-2205. September 1997.

RFC2475. **Network Working Group. An Architecture for Differentiated Services**. RFC-2475. December 1998.

RFC3031. **Network Working Group. Multiprotocol Label Switching Architecture**. RFC-3031. January 2001.

RFC3272. **Network Working Group. Overview and Principles of Internet Traffic Engineering**. RFC-3272. May 2002.

RFC3945. **Network Working Group. Generalized Multi-Protocol Label Switching (GMPLS) Architecture**. RFC-3945. October 2004.

RFC5735. **Internet Research Task Force (IRTF). Special Use IPv4 Addresses**. RFC-5735. January 2010. ISSN: 2070-1721

RFC7426. **Internet Research Task Force (IRTF). Software-Defined Networking (SDN): Layers and Architecture Terminology**. RFC-7426. January 2015. ISSN: 2070-1721

RIVERBED. **Riverbed. Riverbed Modeler**. 2015. Disponível em: <<http://www.riverbed.com/products/performance-management-control/network-performance-management/network-simulation.html>>. Acesso em: 3 jun. 2015.

SDNSKILLS. **The RFC 7426 SDN Architectural Model**. 2015. Disponível em: <<http://www.sdnskills.com/learn/sdn-terms-01/>>. Acesso em: 3 jun. 2015.

SILVA, J. P. S. **Implementação de uma arquitetura para o arcabouço Context-Aware Adaptative Routing Framework (CAARF)**. 2015. Dissertação (Mestrado)- Universidade Salvador – UNIFACS. Salvador, 2015.

SIRAJ, S.; GUPTA, A. K.; RINKU-BADGUJAR. **Network Simulation Tools Survey**. **International Journal of Advanced Research in Computer and Communication Engineering**, v. 1, n 4, jun. 2012. ISSN : 2278 – 1021

TANENBAUM, A. S. **Redes de computadores**. Tradução da 4. ed. Americana. São Paulo: Editora: Campus, 2003. ISBN: 8535211853.

ULTIMATE NS2. **Post processing NS2 Result using NS2 Trace**. 2015. Disponível em: <<http://ns2ultimate.tumblr.com/post/2803718245/post-processing-ns2-result-using-ns2-trace-awk>>. Acesso em: 3 jun. 2015.

VIEIRA, V. et al. **Investigating the specificities of contextual elements management: the CEManTIKA Approach**. In: INTERNATIONAL AND INTERDISCIPLINARY

CONFERENCE ON MODELING AND USING CONTEXT (CONTEXT'07), LNAI 4635, 6., 2007, Roskilde, Denmark. **Proceedings...** 2007.

WEISER, M. The Computer for the 21st Century. **ACM SIGMOBILE Mobile Computing and Communications Review**, Special issue dedicated to Mark Weiser, v. 3, n.3, p. 3-11, 2009.

WENNING, B.; TIMM-GIEL, A.; GÖRG, C. **A generic framework for context-aware routing and its implementation in wireless sensor networks**. ITG-Fachbericht-Mobilkommunikation-Technologien und Anwendungen. [S.l.]: [s.n.], 2009.

XML. **XML Tutorial**. 2015. Disponível em: <<http://www.w3schools.com/xml/>>. Acesso em: 3 jun. 2015.

XSD. **XML Schema Tutorial**. 2015. Disponível em: <<http://www.w3schools.com/schema/default.asp>>. Acesso em: 3 jun. 2015.

YASAR, A.; PREUVENEERS, D.; BERBERS, Y. Evaluation framework for adaptive context-aware routing in large scale mobile peer-to-peer systems. **Peer-to-Peer Networking and Applications**, v. 4, n.1, p. 37-49, 2010.

YERIMA, S. Implementation and evaluation of measurement based admission control schemes within a converged networks QoS management framework. **International Journal of Computer Networks & Communications**, v. 3, n. 4, p. 137-152, 2011.

ZIMMERMANN, A.; LORENZ, A; OPPERMANN, R. An operational definition of context. In: INTERNATIONAL AND INTERDISCIPLINARY CONFERENCE ON MODELING AND USING CONTEXT (CONTEXT'07), 6., 2007. **Proceedings...** 2007.

**APÊNDICE A - PUBLICAÇÕES DO AUTOR**

**OLIVEIRA, A. L. C., MUAKAD, C. F. J., SPÍNOLA, S. S., SILVA, J. P. S., SAMPAIO, P. N. M.** (2014). *Context-aware framework for adaptive routing*. In: 3rd International Workshop on ADVANCEs in ICP Infrastructures and Services. Miami, EUA, December, 2014.

**OLIVEIRA, A. L. C., MUAKAD, C. F. J., SPÍNOLA, S. S., SILVA, J. P. S., SAMPAIO, P. N. M.,** *Providing adaptive traffic routing based on user and network context*. Submetido para conferência.

**OLIVEIRA, A. L. C., MUAKAD, C. F. J., SPÍNOLA, S. S., SILVA, J. P. S., SAMPAIO, P. N. M.,** *Adaptive Traffic Routing Based on Context*. Submetido para conferência.



## ANEXO A - CÓDIGO

```

# i
set ns [new Simulator]
# ii
set val(stop) 3.0
# iii
$ns color 0 Green
# iv
set tracefile [open out.tr w]
$ns trace-all $tracefile
# v
set namfile [open out.nam w]
$ns namtrace-all $namfile
# vi
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
# vii
$ns simplex-link $n0 $n1 1.0Mb 10ms DropTail
$ns queue-limit $n0 $n1 50
$ns simplex-link $n1 $n3 1.0Mb 10ms DropTail
$ns queue-limit $n1 $n3 50
$ns simplex-link $n0 $n2 1.0Mb 10ms DropTail
$ns queue-limit $n0 $n2 50
$ns simplex-link $n2 $n3 1.0Mb 10ms DropTail
$ns queue-limit $n2 $n3 50
# viii
$ns simplex-link-op $n0 $n1 orient right-up
$ns simplex-link-op $n1 $n3 orient right-down
$ns simplex-link-op $n0 $n2 orient right-down
$ns simplex-link-op $n2 $n3 orient right-up
# ix
set udp0 [new Agent/UDP]
$udp0 set class_ 0
$udp0 set fid_ 0
$ns attach-agent $n0 $udp0
set null1 [new Agent/LossMonitor]
$ns attach-agent $n3 $null1
$ns connect $udp0 $null1
$udp0 set packetSize_ 1500
# x
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set rate_ 0.1Mb
$ns at 1.0 "$cbr0 start"
$ns at 2.0 "$cbr0 stop"
# xi
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
exec nam -a out.nam &
exit 0
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run

```

## ANEXO B - TABELA COMPLEMENTAR DE REGRAS DE QOC

Tabela 1 - Tabela complementar de Regras de QoC

Nº	Regra	Motivo	Ação
1	Verifica se existe <i>ContextVersion</i> no XML, e se é um número inteiro positivo	Necessário para definir a versão do XSD a ser usado	Se não identificado <i>ContextVersion</i> , ou valor inválido, usa a versão padrão do <i>Handler</i>
2	Verifica se <i>Username</i> é alfanumérico	Auxiliar na identificação da entidade	Se não for, descarta
3	Verifica se <i>Hostname</i> é alfanumérico	Auxiliar na identificação da entidade	Se não for, descarta
4	Verifica se <i>Domainname</i> é alfanumérico	Auxiliar na identificação da entidade	Se não for, descarta
5	Verifica se <i>MAC</i> é um endereço válido, contendo doze caracteres hexadecimais, dois a dois, separados por dois pontos ou hífen (TANENBAUM, 2003)	Auxiliar na identificação da entidade	Se não for, ou o formato for inválido, descarta
6	Verifica se <i>IPv4</i> é um endereço válido, alfanumérico, contendo até doze caracteres numéricos em grupos de três, separados por pontos (TANENBAUM, 2003)	Auxiliar na identificação da entidade	Se não for, ou o formato for inválido, descarta
7	Verifica se <i>SubscribeID</i> é inteiro positivo	Necessário na identificação de serviços assinados	Se não for, descarta todo este grupo de informações de assinatura
8	Verifica se <i>Subscribe Optional String</i> é alfanumérico	Opcional na identificação de serviços assinados	Se não for, descarta
9	Verifica se <i>Subscribe Optional Integer</i> é um número inteiro	Opcional na identificação de serviços assinados	Se não for, descarta
10	Verifica se <i>Subscribe Optional Real</i> é um número real	Opcional na identificação de serviços assinados	Se não for, descarta
11	Verifica se <i>NTP</i> é booleano	Opcional para identificação temporal	Se não for, descarta
12	Verifica se <i>LastNTPupdate</i> é um número inteiro positivo	Opcional para identificação temporal	Se não for, descarta
13	Verifica se <i>LocalTime</i> é do tipo data	Opcional para identificação temporal	Se não for, descarta
14	Verifica se <i>GPSCordinates</i> é alfanumérico	Opcional para identificações temporais	Se não for, descarta

Nº	Regra	Motivo	Ação
15	Verifica se <i>Network</i> é endereço IP válido, alfanumérico, contendo até doze caracteres numéricos em grupos de três, separados por pontos (TANENBAUM, 2003)	Auxiliar na identificação da localização da entidade	Se não for, descarta
16	Verifica se <i>City</i> é alfanumérico	Auxiliar na identificação da localização da entidade	Se não for, descarta
17	Verifica se <i>Nearé</i> alfanumérico	Auxiliar na identificação da localização da entidade	Se não for, descarta
18	Verifica se <i>rID</i> é um número inteiro positivo	Necessário para identificar as atividades sendo realizadas	Se não for, descarta todo este grupo de informações de atividade
19	Verifica se <i>Subscribe_ID</i> da <i>Activity</i> é um número inteiro positivo	Opcional para identificar o serviço assinado vinculado a esta atividade	Se não for, descarta
20	Verifica se <i>MAC</i> é diferente de: FF:FF:FF:FF:FF:FF ou FF-FF-FF-FF-FF-FF	Endereço reservado (TANENBAUM, 2003)	Se não for diferente, descarta
21	Verifica se <i>IPv4</i> é diferente de: 0.0.0.0/8 , 127.0.0.0/8 , 169.254.0.0/16, 240.0.0.0/4 e 255.255.255.255/32	Endereços reservados (RFC5735, 2010)	Se não for diferente, descarta
22	Verifica se pelo menos um dos parâmetros opcionais do <i>Subscribe</i> existe ( <i>String</i> , <i>Integer</i> ou <i>Real</i> )	Se não existir, o <i>Subscribe</i> não poderá ser identificado	Se não existir pelo menos um, descarta
23	Verifica se a capacidade da CPU esta na faixa aceitável pelo administrador, de 0 até 10000 Khz	Opcional na correta identificação das características do dispositivo	Se não for, descarta
24	Verifica se a porta de origem da relação esta no intervalo de 1 até 65365	Auxiliar na identificação da aplicação em uso	Se não for, descarta
25	Verifica se a porta de destino da relação esta no intervalo de 1 até 65365	Auxiliar na identificação da aplicação em uso	Se não for, descarta
26	Verifica se o executável da aplicação foi identificado	Auxiliar na identificação da aplicação em uso	Se não for, descarta
27	Verifica se a rede destino foi identificada	Necessária na identificação do destino da relação	Se não existir, descarta
28	Verifica se o valor do MOS esta na faixa válida, 0 até 5	Necessário na identificação da expectativa do usuário	Se não existir, descarta

## ANEXO C - INTEGRA DO ARQUIVO XML

```

<?xml version="1.0" encoding="UTF-8"?>
<Entity>
<ContextVersion>
  1
</ContextVersion>
<Individuality>
<Collected_Username>Joe</Collected_Username>
  <Collected_Hostname>wks1</Collected_Hostname>
  <Subscribe>
    <Provided_Subscribe_ID>11</Provided_Subscribe_ID>
    <Provided_Optional_String>Text</Provided_Optional_String>
    <Provided_Optional_Integer>1000</Provided_Optional_Integer>
    <Provided_Optional_Real>1000</Provided_Optional_Real>
  </Subscribe>
  <Collected_Domainname>lab.unifacs.br</Collected_Domainname>
  <Collected_MAC>84-8F-69-CA-72-CE</Collected_MAC>
  <Collected_IPv4>10.0.0.1</Collected_IPv4>
</Individuality>
<Time>
<Collected_NTP>Yes</Collected_NTP>
  <Collected_LastNTPupdate>1</Collected_LastNTPupdate>
<Collected_LocalTime>2014-07-09T19:20-03:00</Collected_LocalTime>
</Time>
<Location>
  <Collected_GPScoordinates>12°58'47.75"S
38°27'31.37"O</Collected_GPScoordinates>
  <Collected_Network>10.0.0.0/24</Collected_Network>
  <Provided_City>Salvador</Provided_City>
  <Provided_Near>Caminho das Árvores</Provided_Near>
  <Provided_Near>Unifacs</Provided_Near>
</Location>
<Activity>
<Act>
  <Collected_rID>1</Collected_rID>
<Provided_Subscribe_ID>11</Provided_Subscribe_ID>
  </Act>
  <Act>
    <Collected_rID>2</Collected_rID>
  </Act>
</Activity>
<Relations>
  <Rel>
    <Collected_rID>1</Collected_rID>
    <Collected_Conected>172.16.0.2</Collected_Conected>
    <Collected_ApplicationName>Skype</Collected_ApplicationName>
    <Collected_ApplicationExecutable>skype.exe</Collected_ApplicationExec
utable>
    <Collected_ApplicationVersion>7.0</Collected_ApplicationVersion>
    <Collected_SourcePort>3032</Collected_SourcePort>
    <Collected_DestinationPort>80</Collected_DestinationPort>
    <Collected_TransportProtocol>TCP</Collected_TransportProtocol>
    <Collected_ApplicationProtocol>HTTP</Collected_ApplicationProtocol>
  </Rel>
  <Rel>
    <Collected_rID>2</Collected_rID>
    <Collected_Conected>10.0.0.33</Collected_Conected>
    <Collected_ApplicationName>Viber</Collected_ApplicationName>

```

```

    <Collected_ApplicationExecutable>viber.exe</Collected_ApplicationExec
utable>
    <Collected_ApplicationVersion>5.0</Collected_ApplicationVersion>
    <Collected_SourcePort>3033</Collected_SourcePort>
    <Collected_DestinationPort>443</Collected_DestinationPort>
    <Collected_TransportProtocol>TCP</Collected_TransportProtocol>
    <Collected_ApplicationProtocol>HTTPS</Collected_ApplicationProtocol>
    </Rel>
</Relations>
<QoE>
<Peer>
    <Collected_rID>1</Collected_rID>
    <Collected_MOSs>4.4</Collected_MOSs>
    <Provided_MOSu>2</Provided_MOSu>
    <Collected_RFactor>96</Collected_RFactor>
    <Collected_Noise>100</Collected_Noise>
</Peer>
<Peer>
    <Collected_rID>2</Collected_rID>
    <Collected_MOSs>5</Collected_MOSs>
</Peer>
</QoE>
<QoD>
<Collected_HasGPS>Yes</Collected_HasGPS>
    <Collected_GPSprecision>10</Collected_GPSprecision>
    <Collected_ProcessorCores>4</Collected_ProcessorCores>
    <Collected_ProcessorClock>2,6</Collected_ProcessorClock>
<Collected_Memory>6</Collected_Memory>
<Nic>
    <Collected_NIC_ID>1</Collected_NIC_ID>
    <Collected_Bandwidth>10000</Collected_Bandwidth>
    <Collected_HasToE>Yes</Collected_HasToE>
    <Collected_MAC>84-8F-69-CA-72-CE</Collected_MAC>
    <Collected_IPv4>10.0.0.1</Collected_IPv4>
</Nic>
</QoD>
<QoS>
    <Nic>
        <Collected_NIC_ID>1</Collected_NIC_ID>
        <Collected_Throughput>0,7</Collected_Throughput>
        <Collected_Loss>3</Collected_Loss>
        <Collected_Latency>3</Collected_Latency>
        <Collected_Jitter>2</Collected_Jitter>
    </Nic>
</QoS>
</Entity>

```

## ANEXO D - DEFINIÇÃO DE ESQUEMA, XSD, UTILIZADO PARA VALIDAR O XML

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="Entity">
<xs:complexType>
<xs:sequence>
<xs:element type="xs:float" name="ContextVersion"/>
<xs:element name="Individuality">
<xs:complexType>
<xs:sequence>
<xs:element type="xs:string" name="Collected_Username"/>
<xs:element type="xs:string" name="Collected_Hostname"/>
<xs:element name="Subscribe">
<xs:complexType>
<xs:sequence>
<xs:element type="xs:byte" name="Provided_Subscribe_ID"/>
<xs:element type="xs:string" name="Provided_Optional_String"/>
<xs:element type="xs:short" name="Provided_Optional_Integer"/>
<xs:element type="xs:short" name="Provided_Optional_Real"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element type="xs:string" name="Collected_Domainname"/>
<xs:element type="xs:string" name="Collected_MAC"/>
<xs:element type="xs:string" name="Collected_IPv4"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Time">
<xs:complexType>
<xs:sequence>
<xs:element type="xs:string" name="Collected_NTP"/>
<xs:element type="xs:byte" name="Collected_LastNTPupdate"/>
<xs:element type="xs:string" name="Collected_LocalTime"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Location">
<xs:complexType>
<xs:sequence>
<xs:element type="xs:string" name="Collected_GPScoordinates"/>
<xs:element type="xs:string" name="Collected_Network"/>
<xs:element type="xs:string" name="Provided_City"/>
<xs:element type="xs:string" name="Provided_Near" maxOccurs="unbounded"
minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Activity">
<xs:complexType>
<xs:sequence>
<xs:element name="Act" maxOccurs="unbounded" minOccurs="0">
<xs:complexType>
<xs:sequence>
<xs:element type="xs:byte" name="Collected_rID"/>
<xs:element type="xs:byte" name="Provided_Subscribe_ID" minOccurs="0"/>
</xs:sequence>

```

```

</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Relations">
<xs:complexType>
<xs:sequence>
<xs:element name="Rel" maxOccurs="unbounded" minOccurs="0">
<xs:complexType>
<xs:sequence>
<xs:element type="xs:byte" name="Collected_rID"/>
<xs:element type="xs:string" name="Collected_Conected"/>
<xs:element type="xs:string" name="Collected_ApplicationName"/>
<xs:element type="xs:string" name="Collected_ApplicationExecutable"/>
<xs:element type="xs:float" name="Collected_ApplicationVersion"/>
<xs:element type="xs:short" name="Collected_SourcePort"/>
<xs:element type="xs:short" name="Collected_DestinationPort"/>
<xs:element type="xs:string" name="Collected_TransportProtocol"/>
<xs:element type="xs:string" name="Collected_ApplicationProtocol"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="QoE">
<xs:complexType>
<xs:sequence>
<xs:element name="Peer" maxOccurs="unbounded" minOccurs="0">
<xs:complexType>
<xs:sequence>
<xs:element type="xs:byte" name="Collected_rID"/>
<xs:element type="xs:float" name="Collected_MOSs"/>
<xs:element type="xs:byte" name="Provided_MOSu" minOccurs="0"/>
<xs:element type="xs:byte" name="Collected_RFactor" minOccurs="0"/>
<xs:element type="xs:byte" name="Collected_Noise" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="QoD">
<xs:complexType>
<xs:sequence>
<xs:element type="xs:string" name="Collected_HasGPS"/>
<xs:element type="xs:byte" name="Collected_GPSprecision"/>
<xs:element type="xs:byte" name="Collected_ProcessorCores"/>
<xs:element type="xs:string" name="Collected_ProcessorClock"/>
<xs:element type="xs:byte" name="Collected_Memory"/>
<xs:element name="Nic">
<xs:complexType>
<xs:sequence>
<xs:element type="xs:byte" name="Collected_NIC_ID"/>
<xs:element type="xs:short" name="Collected_Bandwidth"/>
<xs:element type="xs:string" name="Collected_HasToE"/>
<xs:element type="xs:string" name="Collected_MAC"/>
<xs:element type="xs:string" name="Collected_IPv4"/>
</xs:sequence>
</xs:complexType>

```

```

</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="QoS">
<xs:complexType>
<xs:sequence>
<xs:element name="Nic">
<xs:annotation>
<xs:documentation>In milliseconds</xs:documentation>
</xs:annotation>
<xs:complexType>
<xs:sequence>
<xs:element type="xs:byte" name="Collected_NIC_ID"/>
<xs:element type="xs:string" name="Collected_Throughput"/>
<xs:element type="xs:byte" name="Collected_Loss">
<xs:annotation>
<xs:documentation>In Gbps</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element type="xs:byte" name="Collected_Latency">
<xs:annotation>
<xs:documentation>In packets</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element type="xs:byte" name="Collected_Jitter">
<xs:annotation>
<xs:documentation>In milliseconds</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```



## ANEXO E - CAMINHOS POSSÍVEIS ENTRE AS QUATRO REDES LOCAIS

Tabela 2 - Caminhos possíveis entre as quatro redes locais

<b>ID Caminho</b>	<b>Caminho</b>
1	N0-N4-N7-N3
2	N0-N4-N8-N7-N3
3	N0-N4-N5-N8-N7-N3
4	N0-N4-N5-N6-N7-N3
5	N0-N4-N8-N6-N7-N3
6	N0-N4-N5-N8-N6-N7-N3
7	N0-N4-N5-N1
8	N0-N4-N8-N5-N1
9	N0-N4-N7-N8-N5-N1
10	N0-N4-N7-N6-N5-N1
11	N0-N4-N8-N6-N5-N1
12	N0-N4-N7-N8-N6-N5-N1
13	N0-N4-N7-N6-N8-N5-N1
14	N0-N4-N8-N6-N2
15	N0-N4-N5-N6-N2
16	N0-N4-N7-N6-N2
17	N0-N4-N7-N8-N6-N2
18	N0-N4-N5-N8-N6-N2
19	N0-N4-N7-N8-N5-N6-N2
20	N1-N65-N6-N2
21	N1-N5-N8-N6-N2
22	N1-N5-N8-N7-N6-N2
23	N1-N5-N4-N8-N6-N2
24	N1-N5-N4-N7-N8-N6-N2
25	N1-N5-N4-N8-N7-N6-N2
26	N3-N7-N6-N2
27	N3-N7-N8-N6-N2
28	N3-N7-N4-N8-N6-N2
29	N3-N7-N8-N5-N6-N2
30	N3-N7-N4-N5-N8-N6-N2

## ANEXO F - ÍNTEGRA DO ARQUIVO TCL DO CENÁRIO 1 - EXECUÇÃO 1

```

#=====
#      Simulation parameters setup
#=====
set val(stop)    10.0                                ;# time of simulation end
#=====
#      Initialization
#=====
#Create a ns simulator
set ns [new Simulator]
$ns color 0 Blue
$ns color 1 Red
$ns color 2 yellow
$ns color 3 green
$ns color 4 black
$ns color 5 pink
$ns color 6 white
$ns color 7 black
set cir 1000000
set cbs 3000
set pir 3000
set ebs 3000
set pbs 3000
set packetSize 1000
set testTime 10.0;
#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile
#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
set f1 [open lb1.tr w]
set f2 [open lost2.tr w]
set f3 [open delay3.tr w]
set f4 [open lb4.tr w]
set f5 [open lost5.tr w]
set f6 [open delay6.tr w]
#=====
#      Nodes Definition
#=====
#Create 17 nodes
$ns node-config -MPLS ON
# Edge Routers
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
# Edge Routers End
# Core Routers
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]
set n7 [$ns node]
set n8 [$ns node]
# Core Routers End
$ns node-config -MPLS OFF
# Hosts
set n9 [$ns node]

```

```

set n10 [$ns node]
set n11 [$ns node]
set n12 [$ns node]
set n13 [$ns node]
set n14 [$ns node]
set n15 [$ns node]
set n16 [$ns node]
#Hosts End
#=====
#           Routers Definition
#=====
#Create 0 routers
#=====
#           Links Definition
#=====
#Createlinks between nodes or
#Createlinks between routers or
#Createlinks between routers and nodes
# Edge Links
$ns duplex-link $n9 $n0 10.0Mb 1ms DropTail
$ns queue-limit $n9 $n0 50
$ns duplex-link $n10 $n0 10.0Mb 1ms DropTail
$ns queue-limit $n10 $n0 50
$ns duplex-link $n0 $n4 10.0Mb 1ms DropTail
$ns queue-limit $n0 $n4 50
$ns duplex-link $n11 $n1 10.0Mb 1ms DropTail
$ns queue-limit $n11 $n1 50
$ns duplex-link $n12 $n1 10.0Mb 1ms DropTail
$ns queue-limit $n12 $n1 50
$ns duplex-link $n1 $n5 10.0Mb 1ms DropTail
$ns queue-limit $n1 $n5 50
$ns duplex-link $n13 $n2 10.0Mb 1ms DropTail
$ns queue-limit $n13 $n2 50
$ns duplex-link $n14 $n2 10.0Mb 1ms DropTail
$ns queue-limit $n14 $n2 50
$ns duplex-link $n2 $n6 10.0Mb 1ms DropTail
$ns queue-limit $n2 $n6 50
$ns duplex-link $n15 $n3 10.0Mb 1ms DropTail
$ns queue-limit $n15 $n3 50
$ns duplex-link $n16 $n3 10.0Mb 1ms DropTail
$ns queue-limit $n16 $n3 50
$ns duplex-link $n3 $n7 10.0Mb 1ms DropTail
$ns queue-limit $n3 $n7 50
# Edge Links End
# Links Core type 1
$ns duplex-link $n5 $n4 2.0Mb 50ms DropTail
$ns queue-limit $n5 $n4 50
$ns duplex-link $n7 $n4 2.0Mb 50ms DropTail
$ns queue-limit $n7 $n4 50
$ns duplex-link $n5 $n8 2.0Mb 50ms DropTail
$ns queue-limit $n5 $n8 50
$ns duplex-link $n6 $n8 2.0Mb 50ms DropTail
$ns queue-limit $n6 $n8 50
# End Links Core type 1
# Links Core type 2
$ns duplex-link $n5 $n6 1.0Mb 1ms DropTail
$ns queue-limit $n5 $n6 50
$ns duplex-link $n6 $n7 1.0Mb 1ms DropTail
$ns queue-limit $n6 $n7 50
$ns duplex-link $n4 $n8 1.0Mb 1ms DropTail
$ns queue-limit $n4 $n8 50

```

```

$ns duplex-link $n7 $n8 1.0Mb lms DropTail
$ns queue-limit $n7 $n8 50
# End Links Core type 1
#Give node position (for NAM)
$ns duplex-link-op $n9 $n0 orient right-up
$ns duplex-link-op $n10 $n0 orient right-down
$ns duplex-link-op $n0 $n4 orient right-up
$ns duplex-link-op $n11 $n1 orient right-up
$ns duplex-link-op $n12 $n1 orient right-down
$ns duplex-link-op $n1 $n5 orient right-down
$ns duplex-link-op $n13 $n2 orient left-down
$ns duplex-link-op $n14 $n2 orient left-up
$ns duplex-link-op $n2 $n6 orient left-down
$ns duplex-link-op $n15 $n3 orient left-down
$ns duplex-link-op $n16 $n3 orient left-up
$ns duplex-link-op $n3 $n7 orient left-up
$ns duplex-link-op $n5 $n4 orient left-down
$ns duplex-link-op $n5 $n6 orient right
$ns duplex-link-op $n6 $n7 orient left-down
$ns duplex-link-op $n7 $n4 orient left
$ns duplex-link-op $n4 $n8 orient right-up
$ns duplex-link-op $n5 $n8 orient right-down
$ns duplex-link-op $n6 $n8 orient left-down
$ns duplex-link-op $n7 $n8 orient left-up
for {set i 0} {$i < 9} {incr i} {
    set a n$i
    for {set j [expr $i+1]} {$j < 9} {incr j} {
        set b n$j
        eval $ns LDP-peer $$a $$b
    }
    set m [eval $$a get-module "MPLS"]
    $m enable-reroute "new"
}
Classifier/Addr/MPLS set control_driven_1
# $ns enable-data-driven
Classifier/Addr/MPLS enable-data-driven
# $ns enable-on-demand
Classifier/Addr/MPLS enable-on-demand
# $ns enable-ordered-control
Classifier/Addr/MPLS enable-ordered-control
#=====
#           Agents Definition
#=====
#Setup a UDP connection
set udp0 [new Agent/UDP]
$udp0 set class_ 0
$udp0 set fid_ 0
$ns attach-agent $n9 $udp0
set null1 [new Agent/LossMonitor]
$ns attach-agent $n15 $null1
$ns connect $udp0 $null1
$udp0 set packetSize_ 200

#Setup a UDP connection
set udp2 [new Agent/UDP]
$udp2 set class_ 2
$udp2 set fid_ 2
$ns attach-agent $n11 $udp2
set null3 [new Agent/LossMonitor]
$ns attach-agent $n13 $null3
$ns connect $udp2 $null3

```

```

$udp2 set packetSize_ 1500

#=====
#           Applications Definition
#=====
# VoIP
#Setup a CBR Application over UDP connection
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set packetSize_ 160
$cbr0 set interval_ 0.005
$cbr0 set codePt_ 0
$cbr0 set rate_ 64Kb
$ns at 1.0 "$cbr0 start"
$ns at 9.0 "$cbr0 stop"

# VoD-HD
#Setup a CBR Application over UDP connection
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp2
$cbr1 set packetSize_ 1000
$cbr1 set interval_ 0.005
$cbr1 set codePt_ 0
$cbr1 set rate_ 1.8Mb
$ns at 2.0 "$cbr1 start"
$ns at 8.5 "$cbr1 stop"

#$ns at 0.0 "$n0 label Node"
#=====
#           Termination
#=====
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile f1 f2 f3 f4 f5 f6 namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    close $f1
    close $f2
    close $f3
    close $f4
    close $f5
    close $f6
    exec nam -a out.nam &
    exec xgraph lb1.tr -geometry 800x400 -t "Bandwidth of the application
    attach to agent UDP0 " -x "Simulation Time" -y "Bandwidth" &
        exec xgraph lost2.tr -geometry 800x400 -t "Lost of the application
    attach to agent UDP0 " -x "Simulation Time" -y "Lost" &
        exec xgraph delay3.tr -geometry 800x400 -t "Delay of the application
    attach to agent UDP0 " -x "Simulation Time" -y "Delay" &
        exec xgraph lb4.tr -geometry 800x400 -t "Bandwidth of the application
    attach to agent UDP2 " -x "Simulation Time" -y "Bandwidth" &
        exec xgraph lost5.tr -geometry 800x400 -t "Lost of the application
    attach to agent UDP2 " -x "Simulation Time" -y "Lost" &
        exec xgraph delay6.tr -geometry 800x400 -t "Delay of the application
    attach to agent UDP2 " -x "Simulation Time" -y "Delay" &
    exit 0
}
set holdPkttime1 0
set holdNrPktReceiv1 0
set holdPkttime2 0

```

```

set holdNrPktReceiv2 0
set holdrate 0

proc record {} {

global null1 null3 f1 f2 f3 f4 f5 f6 holdPkttime1 holdNrPktReceiv1
holdPkttime2 holdNrPktReceiv2
set ns [Simulator instance]
set time 0.2

set bw1 [$null1 set bytes_]
set drop1 [$null1 set nlost_]
set lstPtime1 [$null1 set lastPktTime_]
set nrpkts1 [$null1 set npkts_]

set bw2 [$null3 set bytes_]
set drop2 [$null3 set nlost_]
set lstPtime2 [$null3 set lastPktTime_]
set nrpkts2 [$null3 set npkts_]

set now [$ns now]

puts $f1 "$now [expr $bw1/$time*8/1000000]"
puts $f2 "$now [expr $drop1]"

if { $nrpkts1 > $holdNrPktReceiv1 } {

puts $f3 "$now [expr ($lstPtime1 - $holdPkttime1)/($nrpkts1 -
$holdNrPktReceiv1)]"
} else {

puts $f3 "$now [expr ($nrpkts1 - $holdNrPktReceiv1)]"
}

puts $f4 "$now [expr $bw2/$time*8/1000000]"
puts $f5 "$now [expr $drop2]"

if { $nrpkts2 > $holdNrPktReceiv2 } {

puts $f6 "$now [expr ($lstPtime2 - $holdPkttime2)/($nrpkts2 -
$holdNrPktReceiv2)]"
} else {

puts $f6 "$now [expr ($nrpkts2 - $holdNrPktReceiv2)]"
}

>null1 set bytes_ 0
>null1 set nlost_ 0
>null3 set bytes_ 0
>null3 set nlost_ 0
set holdPkttime1 $lstPtime1
set holdNrPktReceiv1 $nrpkts1
set holdPkttime2 $lstPtime2
set holdNrPktReceiv2 $nrpkts2
$ns at [expr $now+$time] "record"
}
$ns at 0.0 "record"
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"

```

\$ns run

## ANEXO G- PROCESSAMENTO DO CÓDIGO AWK - EXEMPLO DO CÓDIGO PARA ATRASO FIM A FIM

# Arquivo: delay-e2e-voip.awk

```
BEGIN {
  src="9.0"; dst="15.0";
  num_samples = 0;
  total_delay = 0;
}

/^\+/\&&$9==src&&$10==dst {
  t_arr[$12] = $2;
};

/^r/\&&$9==src&&$10==dst{
  if (t_arr[$12] > 0) {
    num_samples++;
    delay = $2 - t_arr[$12];
    total_delay += delay;
  };
};

END{
  avg_delay = total_delay/num_samples;
  print "Average end-to-end transmission delay is " avg_delay " seconds";
  print "Measurement details:";
  print " - Since packets are created from the address " src;
  print " - Until the packets are destroyed at the address " dst;
};
```



## ANEXO H - EXEMPLO DO CÓDIGO PARA BITS TRANSMITIDOS PARA LARGURA DE BANDA

# Arquivo: tp-e2e-voip.awk

```
BEGIN {
fromNode=3; toNode=15;
src1 = 9.0; dst1 = 15.0;
lineCount1 = 0;totalBits1 = 0;
}

/^r/&&$3==fromNode&&$4==toNode&&$9==src1&&$10==dst1 {
totalBits1 += 8*$6;
if ( lineCount1==0 ) {
timeBegin1 = $2; lineCount1++;
} else {
timeEnd1 = $2;
};
};

END{
duration = timeEnd1-timeBegin1;
print "Transmission: source " src1 "->Destination" dst1;
print " - Total transmitted bits = " totalBits1 " bits";
print " - duration = " duration " s";
print " - Throughput = " totalBits1/duration/1e3 " kbps.";
print " ";
};
```

## ANEXO I - INTEGRA DO ARQUIVO XML ENVIADO NO INSTANTE (1,0)

```

<?xml version="1.0" encoding="UTF-8"?>
<Entity>

<ContextVersion>
  1
</ContextVersion>

<Individuality>
<Collected_Username>N9 User</Collected_Username>
  <Collected_Hostname>N9</Collected_Hostname>
  <Collected_Domainname>lab.unifacs.br</Collected_Domainname>
  <Collected_MAC>00-53-00-00-00-09</Collected_MAC>
  <Collected_IPv4>192.168.0.9</Collected_IPv4>
</Individuality>

<Location>
  <Collected_Network>192.168.0.0/24</Collected_Network>
</Location>

<Activity>
<Act>
  <Collected_rID>1</Collected_rID>
</Act>
</Activity>

<Relations>
  <Rel>
    <Collected_rID>1</Collected_rID>
    <Collected_Conected>192.168.3.15</Collected_Conected>
    <Collected_ApplicationName>Viber</Collected_ApplicationName>
    <Collected_ApplicationExecutable>viber.exe</Collected_ApplicationExec
utable>
    <Collected_ApplicationVersion>5.0</Collected_ApplicationVersion>
    <Collected_SourcePort>50000</Collected_SourcePort>
    <Collected_DestinationPort>5243</Collected_DestinationPort>
    <Collected_TransportProtocol>UDP</Collected_TransportProtocol>
  </Rel>
</Relations>

</Entity>

```

## ANEXO J - APLICATIVO CISCO WEBEX, NA PORTA REMOTA 9000

```

<?xml version="1.0" encoding="UTF-8"?>
<Entity>

<ContextVersion>
  1
</ContextVersion>

<Individuality>
<Collected_Username>N11-User</Collected_Username>
  <Collected_Hostname>N11-Host</Collected_Hostname>
  <Collected_Domainname>lab.unifacs.br</Collected_Domainname>
  <Collected_MAC>00-53-00-00-00-11</Collected_MAC>
  <Collected_IPv4>192.168.1.11</Collected_IPv4>
</Individuality>

<Time>
<Collected_NTP>No</Collected_NTP>
  <Collected_LastNTPupdate>0</Collected_LastNTPupdate>
<Collected_LocalTime>2015-07-30T08:02:0-03:00</Collected_LocalTime>
</Time>

<Location>
  <Collected_Network>192.168.1.0/24</Collected_Network>
</Location>

<Activity>
<Act>
  <Collected_rID>1</Collected_rID>
</Act>
</Activity>

<Relations>
  <Rel>
    <Collected_rID>1</Collected_rID>
    <Collected_Conected>192.168.2.13</Collected_Conected>
    <Collected_ApplicationName>Cisco WebEX</Collected_ApplicationName>
    <Collected_ApplicationExecutable>webex.exe</Collected_ApplicationExec
utable>
    <Collected_SourcePort>11321</Collected_SourcePort>
    <Collected_DestinationPort>9000</Collected_DestinationPort>
    <Collected_TransportProtocol>UDP</Collected_TransportProtocol>
  </Rel>
</Relations>
</Entity>

```

## ANEXO K - PRINCIPAIS DIFERENÇAS NO ARQUIVO TCL DO CENÁRIO 1 - EXECUÇÃO 1 PARA O CENÁRIO 2 – EXECUÇÃO 2

```

#=====
#           Agents Definition
#=====
#Setup a UDP connection
set udp4 [new Agent/UDP]
$udp4 set class_ 4
$udp4 set fid_ 4
$ns attach-agent $n12 $udp4
set null5 [new Agent/LossMonitor]
$ns attach-agent $n10 $null5
$ns connect $udp4 $null5
$udp4 set packetSize_ 200

#Setup a TCP connection
set tcp6 [new Agent/TCP]
$tcp6 set class_ 6
$tcp6 set fid_ 6
$ns attach-agent $n14 $tcp6
set sink7 [new Agent/TCPSink]
$ns attach-agent $n16 $sink7
$ns connect $tcp6 $sink7
$tcp6 set packetSize_ 1500

#=====
#           Applications Definition
#=====
# VoIP (2)
#Setup a CBR Application over UDP connection
set cbr2 [new Application/Traffic/CBR]
$cbr2 attach-agent $udp4
$cbr2 set packetSize_ 160
$cbr2 set interval_ 0.005
$cbr2 set codePt_ 0
$cbr2 set rate_ 0.064Mb
$ns at 1.5 "$cbr2 start"
$ns at 3.5 "$cbr2 stop"

# Backup / FTP
#Setup a FTP Application over TCP connection
set ftp3 [new Application/FTP]
$ftp3 attach-agent $tcp6
$ftp3 set packetSize_ 1000
$ftp3 set interval_ 0.003
$ftp3 set codePt_ 0
$ns at 1.5 "$ftp3 start"
$ns at 3.5 "$ftp3 stop"

```

## ANEXO L - DIFERENÇAS NO ARQUIVO TCL DO CENÁRIO 2 - EXECUÇÃO 1 PARA O CENÁRIO 3 – EXECUÇÃO 1

```

#=====
#           Agents Definition
#=====
#Setup a UDP connection
set udp8 [new Agent/UDP]
$udp8 set class_ 8
$udp8 set fid_ 8
$ns attach-agent $n16 $udp8
set null9 [new Agent/LossMonitor]
$ns attach-agent $n14 $null9
$ns connect $udp8 $null9
$udp8 set packetSize_ 1500

#Setup a UDP connection
set udp10 [new Agent/UDP]
$udp10 set class_ 10
$udp10 set fid_ 10
$ns attach-agent $n11 $udp10
set null11 [new Agent/LossMonitor]
$ns attach-agent $n10 $null11
$ns connect $udp10 $null11
$udp10 set packetSize_ 1500

#=====
#           Applications Definition
#=====
# VoD-HD (2)
#Setup a CBR Application over UDP connection
set cbr4 [new Application/Traffic/CBR]
$cbr4 attach-agent $udp8
$cbr4 set packetSize_ 1000
$cbr4 set interval_ 0.005
$cbr4 set codePt_ 0
$cbr4 set rate_ 1.8Mb
$ns at 4.0 "$cbr4 start"
$ns at 5.5 "$cbr4 stop"

# VoD
#Setup a CBR Application over UDP connection
set cbr5 [new Application/Traffic/CBR]
$cbr5 attach-agent $udp10
$cbr5 set packetSize_ 1000
$cbr5 set interval_ 0.005
$cbr5 set codePt_ 0
$cbr5 set rate_ 520Kb
$ns at 6.0 "$cbr5 start"
$ns at 9.0 "$cbr5 stop"

```