

UNIVERSIDADE FEDERAL FLUMINENSE

PAULO DAVID SCHVARTZ MARIBONDO

**CAoS:
Evitando Degradação de Tráfego de Voz em
Redes Corporativas usando Redes Definidas por
Software**

NITERÓI

2015

UNIVERSIDADE FEDERAL FLUMINENSE

PAULO DAVID SCHVARTZ MARIBONDO

CAoS:
Evitando Degradação de Tráfego de Voz em
Redes Corporativas usando Redes Definidas por
Software

Dissertação de Mestrado apresentada ao
Programa de Pós-Graduação em Enge-
nharia de Telecomunicações da Universi-
dade Federal Fluminense como parte dos
requisitos necessários à obtenção do tí-
tulo de Mestre em Engenharia de Teleco-
municações. Área de concentração:
Sistemas de Telecomunicações

Orientador:
Natalia Castro Fernandes

NITERÓI

2015

PAULO DAVID SCHVARTZ MARIBONDO

CAoS: Evitando Degradação de Tráfego de Voz em Redes Corporativas usando
Redes Definidas por Software

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Telecomunicações da Universidade Federal Fluminense como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Telecomunicações. Área de concentração: Sistemas de Telecomunicações

Aprovado em 03 de Agosto de 2015.

BANCA EXAMINADORA

Prof^a. Natalia Castro Fernandes, D.Sc.
Universidade Federal Fluminense (UFF)
Orientadora

Prof^a. Débora Christina Muchaluat Saade, D. Sc.
Universidade Federal Fluminense (UFF)

Prof^o. Rodrigo de Souza Couto, D. Sc.
Universidade do Estado do Rio de Janeiro (UERJ)

NITERÓI
2015

*Dedico este trabalho principalmente aos meus familiares, em especial à
minha amada avó Suzana Schwartz Maribondo da Trindade (in memoriam).*

Agradecimentos

Primeiramente gostaria de agradecer a Petrobras pela oportunidade de realizar este Mestrado, pois sem o incentivo desta Empresa, certamente não seria possível completar esta trajetória.

Agradeço muito a Professora Natália Castro Fernandes que, além de ministrar magistralmente duas das mais importantes cadeiras do programa que eu optei fazer, me orientou com muita atenção nestes últimos anos de estudos e testes pelos quais passei. Um dos meus principais receios em entrar para o Mestrado era justamente a questão da escolha do orientador, e felizmente acabei por ser orientando de uma das mais dedicadas profissionais de educação que tive nos últimos anos. Muito obrigado.

Agradeço imensamente à minha Família por entender minhas diversas ausências por causa de estudos, provas, simulações, etc. Em especial a minha esposa Christiane Fialho Ribeiro, pela força que me deu nas horas de cansaço e que permitiu que eu criasse meu laboratório de simulações em casa, bagunçando a sala por diversas vezes. Mais do que especialmente, não só agradeço como também dedico esta Pós-Graduação a minha avó, a Engenheira e Professora Suzana Schvartz Maribondo da Trindade (*in memoriam*), que foi a minha mentora intelectual desde meu nascimento, quem me guiou até aqui.

Meus atenciosos agradecimentos a esta banca examinadora, pelas excelentes contribuições e pelo tempo dispendido na leitura, formulação dos comentários e apresentação de minha defesa de dissertação.

Resumo

Nesta dissertação propõe-se um novo método para Adaptação de Codec em Redes Definidas por Software, nomeado *Codec Adaptation over SDN* (CAoS). Este método é baseado na arquitetura *Software Defined Network* (SDN), implementando uma eficiente e inovadora abordagem para evitar a degradação de voz em redes corporativas. Através de elementos de borda, enlaces *Wide Area Network* (WAN) enfileiram o tráfego de voz e também limitam o número de chamadas simultâneas passantes ao implementar o *Quality of Service* (QoS) e o Controle de Admissão de Chamadas (CAC). Esta proposta utiliza o controlador SDN centralizado para monitorar em tempo real o tráfego de voz e evitar seu congestionamento, através de uma adaptação de codec automática em clientes *Voice over IP* (VoIP) que estão competindo por banda escassa. Diferentemente de outras abordagens que utilizam hardwares específicos para a realização da adaptação de codec ou monitoração da rede, ao utilizar as redes SDN, muitas funcionalidades de rede podem ser implementadas aproveitando-se o controlador centralizado. Com isso o CAoS pôde ser elaborado sem a adição de hardware extra à topologia, otimizando os custos de *Capital Expenditure* (CAPEX) de uma companhia. Os testes de emulação foram realizados usando o Mininet, o controlador POX e clientes VoIP programáveis. O módulo CAoS foi adicionado ao controlador OpenFlow e dividido em três sub-módulos: monitoramento, decisão e *Session Initiation Protocol* (SIP), este último também presente nos clientes VoIP. Os resultados mostram que a qualidade da voz melhora significativamente quando um congestionamento ocorre no enlace e o CAoS está ativo. O método também permite o aumento do número de chamadas simultâneas que passam pelo enlace WAN por causa da redução do *payload* do codec.

Palavras-chave: OpenFlow, SDN, *coder-decoder* (CODEC), SIP, VoIP.

Abstract

In this dissertation we propose a new method for Codec Adaptation Over Software-defined networks, named *Codec Adaptation over SDN* (CAoS). This method is based on a software-defined network architecture and implements an efficient and innovative approach for voice degradation avoidance in enterprise networks. Enterprise scarce *Wide Area Network* (WAN) links queue voice traffic and also limit it to a certain number of calls by performing *Quality of Service* (QoS) and *Call Admission Control* (CAC) on border elements. This study proposes the use of an existing centralized *Software Defined Network* (SDN) controller to monitor real-time voice traffic and to avoid voice congestion, by performing an automatic codec adaptation to *Voice over IP* (VoIP) peers that are competing for scarce throughput. Differently from other approaches which use dedicated hardwares for codec adaptation or network monitoring, by using the SDN, many network features can be implemented taking the advantage of the SDN centralized controller. Therefore, CAoS could be labored without any extra hardware addition, optimizing company *Capital Expenditure* (CAPEX). Emulation tests were performed using Mininet, POX controller and programmable VoIP clients. CAoS module was added to the OpenFlow controller and it was divided in three sub-modules: monitoring, decision and *Session Initiation Protocol* (SIP) modules, this last one also present in VoIP clients. The results show that quality of voice increases significantly when congestion occurs in a link and CAoS is running. The method also allows the increase of number of calls, due to codec payload reduction.

Keywords: OpenFlow, SDN, *coder-decoder* (CODEC), SIP, VoIP.

Lista de Figuras

2.1	Extensão de ramais analógicos utilizando VoIP.	10
2.2	Topologia da telefonia IP.	11
2.3	Registro na base de dados da central IP.	12
2.4	Mobilidade na telefonia IP.	13
2.5	Jitter: a variação do atraso entre pacotes.	15
2.6	Mensagens SIP durante registro, iniciação e término de uma chamada.	21
2.7	Rota de menor custo.	31
2.8	Técnicas de QoS e CAC.	34
3.1	Conceitos básicos do SDN implementado em OpenFlow.	39
3.2	Atuação do plano de controle.	40
3.3	Entradas da tabela de fluxos [1].	41
3.4	Simples exemplo de fluxos OpenFlow.	42
3.5	Ambiente Mininet interagindo com “rede viva”.	43
4.1	Arquitetura do NCVoIP [2].	47
5.1	Fases do CAoS.	53
5.2	Alteração de Codec	58
6.1	Ambiente de emulação - controlador POX e Mininet rodando em equipamentos distintos, e VoipMonitor capturando na interface WAN.	66

6.2	Comparação de Codecs no Teste #1.	69
6.3	Desempenho do CAoS para múltiplas chamadas.	72
6.4	Permormance do CAoS variando período entre medições (<i>Polling</i>).	74
6.5	Mensagens de requisição e resposta relativas às estatísticas de fluxo e portas enviadas do comutador ao controlador.	75
6.6	Desempenho do CAoS - TCP x UDP	78

Lista de Tabelas

2.1	Codecs típicos.	24
2.2	Codecs e suas taxas.	26
2.3	Fator R e MOS.	30
3.1	Tabela de fluxos do comutador da Figura 3.4.	42
6.1	Parâmetros do Teste #1	67
6.2	Parâmetros do Teste #2	71
6.3	Parâmetros do Teste #3	73
6.4	Tráfego simétrico de controle.	75
6.5	Parâmetros do Teste #4	76

Lista de Abreviaturas e Siglas

API	<i>Application Programming Interface</i>	44
ATA	Adaptador para Telefone Analógico	9
CAC	Controle de Admissão de Chamadas	31
CAoS	<i>Codec Adaptation over SDN</i>	4
CAPEX	<i>Capital Expenditure</i>	v
CBWFQ	<i>Class-Based Weighted Fair Queuing</i>	32
CODEC	<i>coder-decoder</i>	v
CPDs	Centros de Processamento de Dados	35
CSMA/CD	<i>Carrier Sense Multiple Access with Collision Detection</i>	27
CSMA/CA	<i>Carrier Sense Multiple Access With Collision Avoidance</i>	27
DDD	Discagem Direta à Distância	2
DDI	Discagem Direta Internacional	31
DSCP	<i>Differentiated Services Code Point</i>	32
EF	<i>Expedited Forwarding</i>	55
ETSI	<i>European Telecommunications Standards Institute</i>	29
FIB	<i>Forward Information Base</i>	38
HMM	Hora de Maior Movimento	32
HTTP	<i>Hypertext Transfer Protocol</i>	14
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos	22

IETF	<i>Internet Engineering Task Force</i>	17
IP	<i>Internet Protocol</i>	12
ISDN	<i>Integrated Services Digital Network</i>	16
ITU	<i>International Telecommunication Union</i>	16
LAN	<i>Local Area Network</i>	32
LLQ	<i>Low-Latency Queuing</i>	32
MEGACO	<i>Media Gateway Control</i>	17
MGCP	<i>Media Gateway Control Protocol</i>	17
MOS	<i>Mean Opinion Score</i>	5
MPLS	<i>Multi-Protocol Label Switching</i>	23
MPLS-VPN	<i>MPLS Virtual Private Network</i>	23
NB	<i>Narrow Band</i>	25
NCVoIP	<i>Nunes Costa VoIP</i>	46
NGN	<i>Next-Generation Networks</i>	3
ONF	<i>Open Networking Foundation</i>	37
PCM	<i>Pulse Code Modulation</i>	25
PESQ	<i>Perceptual Evaluation of Speech Quality</i>	29
PSTN	<i>Public Switched Telephony Network</i>	7
QoS	<i>Quality of Service</i>	21
RDSI	<i>Rede Digital de Serviços Integrados</i>	16
RIB	<i>Routing Information Base</i>	38
RPTC	<i>Rede Pública de Telefonia Comutada</i>	7
RTCP	<i>RTP Control Protocol</i>	13
RTP	<i>Real-time Transport Protocol</i>	13

SDN	<i>Software Defined Network</i>	4
SDP	<i>Session Definition Protocol</i>	18
SIP	<i>Session Initiation Protocol</i>	6
SMTP	<i>Simple Mail Transport Protocol</i>	14
TC	<i>Traffic Control</i>	64
TCP	<i>Transmission Control Protocol</i>	12
TDM	<i>Time division multiplex</i>	2
TI	<i>Tecnologia da Informação</i>	35
TOS	<i>Type of Service</i>	22
UDP	<i>User Datagram Protocol</i>	13
VoIP	<i>Voice over IP</i>	5
WAN	<i>Wide Area Network</i>	3
WB	<i>Wide Band</i>	25

Sumário

1	Introdução	1
1.1	Motivação e objetivos	1
1.2	Organização do trabalho	5
2	Voz sobre IP e telefonia IP	7
2.1	VoIP - <i>Voice over IP</i>	7
2.2	Telefonia IP	8
2.3	Vantagens	9
2.4	Protocolos	12
2.4.1	Protocolos de transporte	12
2.4.1.1	<i>Real-time Transport Protocol</i> (RTP)	14
2.4.1.2	<i>RTP Control Protocol</i> (RTCP)	15
2.4.2	Protocolos de sinalização	16
2.4.2.1	H.323	16
2.4.2.2	<i>Media Gateway Control Protocol</i> (MGCP)	17
2.4.2.3	<i>Media Gateway Control</i> (MEGACO/H.248)	17
2.4.2.4	<i>Session Initiation Protocol</i> (SIP)	17
2.4.2.5	Estabelecimento da chamada - <i>Call-Setup</i>	19

2.5	Qualidade	20
2.5.1	Qualidade de serviço (QoS)	20
2.5.1.1	QoS de camada 2	22
	802.1p	22
	802.1e	22
2.5.1.2	QoS de camada 3	22
	<i>Differentiated Services</i> (DiffServ)	22
	<i>Multi-Protocol Label Switching</i> (MPLS)	23
2.5.2	Qualidade de voz	23
	G.711 (<i>Pulse Code Modulation</i>)	24
	G.729	25
	SPEEX	25
	G.722	26
2.5.3	Atraso e <i>jitter</i>	26
2.5.4	<i>Mean Opinion Score</i> (MOS) e o Modelo E do <i>International Telecommunication Union</i> (ITU)	28
2.6	Motivação do trabalho	30
2.6.1	Vantagens da telefonia IP	30
	Rotas de menor custo	30
	Comunicações unificadas	31
2.6.2	Melhorias aplicáveis	32
	QoS - qualidade de serviço	32

Controle de Admissão de Chamadas (CAC)	32
3 Redes Definidas por Software: SDN	35
3.1 Histórico	35
3.2 Conceito	37
3.3 OpenFlow	38
3.4 Implementações SDN	42
Mininet	42
NOX/POX	43
4 Trabalhos relacionados	45
4.1 <i>Dynamic Codec with Priority for Voice over IP in WLAN</i>	45
4.2 Nunes Costa VoIP (NCVoIP)	46
4.3 <i>QoE-based Adaptive mVoIP Service Architecture in SDN Networks</i> .	47
4.4 <i>Improving network management with software defined networking</i> .	47
4.5 <i>Improving QoS in real-time internet applications</i>	48
4.6 Comparativos	48
5 Adaptação de codec em redes definidas por software - CAoS	50
5.1 CAoS - <i>Codec Adaptation over SDN</i>	51
5.1.1 Módulo de monitoração	52
5.1.2 Módulo de decisão	56
5.1.3 Módulo SIP	61

6	Implementação e resultados	62
6.1	Implementação	62
6.1.1	Medição	62
6.1.2	Cliente VoIP	63
6.1.3	Cenário de emulação	64
6.1.4	Resultados	66
	Teste #1	67
	Teste #2	70
	Teste #3	73
	Teste #4	76
7	Conclusão	79
7.1	Trabalhos futuros	81
	Referências	85

Capítulo 1

Introdução

1.1 Motivação e objetivos

A comunicação é um dos principais pilares para o sucesso de empresas, pois estas são formadas essencialmente por pessoas, as quais devem trocar ideias, discutir temas, aprimorar procedimentos, todos com o mesmo fim: garantir a melhora dos resultados da corporação. Quanto mais eficaz for a comunicação interna e externa de uma empresa, mais dinâmica será a busca por esses melhores resultados.

As comunicações empresariais evoluíram muito no último século com a utilização dos microcomputadores e das redes de dados. Antigamente, para se encaminhar uma simples mensagem de um setor para outro de uma mesma companhia, por menor que fosse essa mensagem, dependia-se de uma datilógrafa, mensageiro, e mais algum tempo para receber a resposta pelo mesmo processo. Somente à partir da década de 80 que os enlaces de dados começaram a ser instalados no Brasil, primeiramente através da Rede Nacional de Comunicação de Dados por Comutação de Pacotes (RENPAC) lançada pela Embratel [3].

Por outro lado, os sistemas de telefonia no Brasil são mais antigos, datando da época do Império, quando D. Pedro II ordenou a instalação das primeiras linhas de telégrafo entre o Palácio da Quinta da Boa Vista e as residências de seus ministros,

evoluindo até a criação da Discagem Direta à Distância (DDD) em 1969 e expansão do número de terminais telefônicos, que chegou a 2,5 milhões em 1977 [4].

Desde então, as comunicações ganharam muita agilidade no mundo corporativo, bastando que os usuários simplesmente discassem para o número correspondente ao usuário que desejasse comunicação e a conversação era estabelecida.

Até o início dos anos 2000, as redes telefônicas eram essencialmente determinísticas, baseadas na tecnologia *Time division multiplex* (TDM), onde cada conversação ocupa um canal de voz bem determinado, ou seja, enlaces exclusivos de voz eram estabelecidos entre centrais telefônicas locais das empresas e as provedoras de telefonia fixa. Cada circuito determinístico provia canais limitados para as conversações, e durante os períodos de baixa utilização, esses canais ficavam ociosos.

Com a evolução dos enlaces de dados, particularmente com a ampliação de suas capacidades, que agora migravam para fibras ópticas, vislumbrou-se a otimização desses circuitos fazendo-os trafegar tanto as comunicações de dados quanto de voz.

Com isso, desde o início dos anos 2000, as redes corporativas estão evoluindo seus sistemas telefônicos tradicionais para a telefonia IP - Voz sobre IP, onde não há a determinação de “canal de voz”. A migração se mostra muito vantajosa principalmente para ambiente corporativo, onde o volume de ligações é muito alto. Com a Telefonia IP, as chamadas telefônicas são roteadas pela própria rede de dados da companhia, não necessitando a utilização de circuitos exclusivos para a voz. Outra grande vantagem ao se utilizar o serviço de voz sobre redes IP é a otimização de chamadas de longa distância, pois usando a inteligência da controladora da Telefonia IP, chamadas que antes seriam tarifadas como longa distância, podem agora escoar localmente por onde a empresa possuir presença. Uma terceira vantagem é a convergência de diversos serviços que podem integrar o conceito de Comunicação Unificada, onde há a possibilidade de junção de diversos meios de comunicação [5], como o telefone, *smartphone*, *softphone*, e-mail, entre outros, em um só sistema de comunicação corporativo.

Porém, migrando o serviço de telefonia tradicional para as redes IP, aderindo ao conceito de *Next-Generation Networks* (NGN) [6], todos os problemas existentes que afetam os pacotes no mundo IP afetarão o serviço de telefonia. Como o sistema de comunicação por voz é um serviço de tempo real, qualquer *perda, atraso ou variação de atraso (jitter)* na entrega da informação, ou seja, dos pacotes de voz, entre os interessados na comunicação, afetará demasiadamente o serviço, fazendo com que a qualidade da rede como um todo se torne muito relevante para este tipo de aplicação [7].

Em companhias que optaram por migrar sua rede de voz para IP, a telefonia se torna depende dos enlaces *Wide Area Network* (WAN) para que usuários possam falar com os demais escritórios da corporação. No entanto, se o enlace não está disponível, ou não opera satisfatoriamente, o serviço de telefonia será diretamente comprometido. Bastaria uma simples sobrecarga neste circuito WAN para que o serviço de voz fosse degradado, uma vez que este serviço é muito sensível a variações bruscas de atraso, jitter e perda de pacotes, parâmetros diretamente ligados a “saúde” do enlace. A sobrecarga é um tipo de falha que qualquer companhia está sujeita a passar, pois os circuitos não são superdimensionados devido ao alto custo de aluguel. Dessa forma, durante uma sobrecarga, para que o serviço de voz seja menos afetado, os equipamentos da rede usam artifícios de classificação de serviço para que haja a priorização daqueles mais críticos. De fato, os serviços de voz precisam de garantias de baixa perda de pacotes, baixos atrasos e baixa variação do atraso. Contudo, o aumento desses parâmetros são os primeiros efeitos de um congestionamento no enlace.

Por não conseguir encaminhar pela interface WAN aqueles pacotes que chegam em sua interface de entrada, um equipamento de borda utiliza sua memória interna, o *buffer*, para armazenar alguns pacotes e enviá-los posteriormente. Este artifício, quando lida-se com tráfego de dados, é muito eficaz, porém o tráfego de voz exige que estes pacotes sejam entregues ao seu destino imediatamente, pois trata-se de um serviço de interlocução em tempo real. Assim como este atraso na

entrega dos pacotes, ocorre também a variação desta entrega, o que é chamado de *jitter*, que pode ser causado pelo tempo variado que diferentes pacotes permanecem na memória *buffer*, ou por diferentes percursos que dois ou mais pacotes percorreram até seu destino. A variação desses atrasos faria com que a voz remontada a partir dos pacotes recebidos ficasse “fora de ritmo”, o que é inaceitável. Considerando estes dois parâmetros afetados pelo congestionamento de um enlace, atraso e variação do atraso, é razoável que o receptor aguarde os pacotes por um determinado tempo, e que se expirado o limite máximo de espera, o descarte daqueles que estavam atrasados ocorre. Com isso, em um enlace congestionado, também temos o parâmetro de perda de pacotes.

Com o passar dos anos e a contínua evolução tecnológica, foi criado recentemente um novo conceito de redes: as *Software Defined Network* (SDN) [8]. Estas redes trazem de volta a hierarquização do plano de controle, ou seja, a inteligência da rede é centralizada em equipamentos definidos como controladores SDN. Estes controladores permitem a criação de módulos programáveis que podem servir como base de criação de funcionalidades extras à operação e monitoração da rede. Com este conceito, esta dissertação vislumbrou a criação de um novo módulo para redes de voz que usam arquitetura SDN, visando minimizar os sintomas de degradação relatados anteriormente.

Portanto, nesta dissertação é proposto o *Codec Adaptation over SDN* (CAoS), que monitora e controla o tráfego de voz em Redes Definidas por Software. Este trabalho inova ao abordar um novo método que minimiza os sintomas de congestionamento de rede, reduzindo a qualidade da conversação ao modificar o codec em uso por um de menor consumo de banda antes de um congestionamento ser detectado. A principal vantagem do CAoS é conseguir manter os fluxos de voz com boa qualidade mesmo quando a rede está congestionada. A outra vantagem deste trabalho é o uso de somente equipamentos existentes na telefonia IP e em redes SDN, fazendo o controlador OpenFlow monitorar e interagir diretamente com o *softswitch* ou com clientes VoIP para os devidos ajustes de codec, diferentemente

de outras abordagens.

Na análise do CAoS, foram emulados cenários com conversações reais. Nestes cenários, o congestionamento do enlace é forçado e o novo módulo programado para um controlador OpenFlow atua verificando o status do congestionamento, solicitando alteração da qualidade da conversação para o equipamento de voz de acordo com o estado da rede. Durante os testes, pôde-se mostrar como o *Mean Opinion Score* (MOS) foi suficientemente mantido a níveis muito bons quando comparado ao cenário sem uso do CAoS.

1.2 Organização do trabalho

Esta dissertação está organizada como descrito a seguir. O Capítulo II detalha o conceito de voz sobre IP, fazendo um apanhado dos protocolos utilizados, codificadores de áudio, vantagens da migração da telefonia tradicional para redes IP e alguns melhoramentos vislumbrados nas redes de voz sobre IP, o que motivou este trabalho.

O Capítulo III aborda o tema das Redes Definidas por Software - um novo conceito em redes que vem ganhando atenção no mercado e na comunidade acadêmica. Este capítulo introduz os conceitos básicos das redes SDN, o plano de controle e o plano de dados, assim como a característica de programabilidade e interoperabilidade com diversos sistemas. Ao final desta parte, aborda-se também o principal protocolo de comunicação em redes SDN: o OpenFlow.

No Capítulo IV trazemos alguns trabalhos relacionados com os temas abordados nesta dissertação, que obviamente também foram motivadores para o estudo. Adaptação dinâmica de codecs para usuários *Voice over IP* (VoIP) em redes sem fio [9], aplicação que utiliza um fluxograma de decisões [2], medição da qualidade de voz em redes SDN [10] e o uso da SDN como uma nova possibilidade de introduzir mecanismos para melhorar a gerência da rede [11] são alguns temas de trabalhos motivadores.

No Capítulo V este trabalho apresenta a nova proposta que visa melhoramentos nas redes de voz sobre IP em ambientes SDN. Mostramos como a “Adaptação de Codec em Redes Definidas por Software”, ou simplesmente CAoS, age nas redes através de seus 3 módulos - Monitoração, Decisão e *Session Initiation Protocol* (SIP).

O Capítulo VI detalha como a implementação dos testes de emulação foram realizados, quais equipamentos foram empregados, como foram interligados e ao final apresentamos os resultados das emulações em seus diversos cenários.

A dissertação é encerrada no Capítulo VII, onde concluímos que a migração para a Telefonia IP é uma realidade, trazendo diversos benefícios para os usuários, principalmente em ambientes corporativos. Também apresentamos pontos que podem ser melhorados quando falamos de qualidade do serviço de voz em redes IP, especificamente em redes SDN, onde a integração, programabilidade e gerenciamento são pontos fortes. Baseado nos resultados, afirmamos que a proposta do uso do CAoS traz ótimos benefícios quando comparamos com situações críticas e o mecanismo proposto não é usado. E por fim apresentamos algumas ideias para aprofundar os testes sobre este tema.

Capítulo 2

Voz sobre IP e telefonia IP

2.1 VoIP - *Voice over IP*

VoIP é o acrônimo para Voz sobre IP, que é uma tecnologia que permite, por exemplo, realizar ligações telefônicas utilizando uma conexão com a Internet ao invés da Rede Pública de Telefonia Comutada (RPTC), ou *Public Switched Telephony Network* (PSTN) em inglês. O tráfego VoIP pode ser realizado não somente através da Internet, mas de qualquer rede IP, como uma rede local de uma empresa.

Tradicionalmente, uma conversação telefônica tem as falas de seus interlocutores (sinais sonoros) convertidas em sinais elétricos analógicos e então estes convertidos em sinais digitais, ou seja, bits. Estes bits percorrem todo o “caminho” entre um e outro interlocutor, e são novamente convertidos em sinais sonoros na outra extremidade. Este percurso fica dedicado àquela conversação durante o todo o tempo que a mesma durar [12].

Já nas chamadas VoIP os sinais anteriormente convertidos em bits são colocados em pacotes de dados IP e trafegam pela rede como um datagrama típico, assim como e-mails, páginas *web*, *streaming* multimídia, entre outras aplicações conhecidas. Devido à voz ser passada em fluxos de datagramas, uma chamada VoIP

não requer exclusividade do “caminho” entre seus interlocutores, bastando apenas que os pacotes de voz cheguem ao seu destino na ordem e no tempo previsto, ou que o receptor consiga colocá-lo na ordem necessária. Para isso, a identificação e a priorização dos pacotes de voz se faz necessária [12].

2.2 Telefonia IP

Muitas pessoas confundem VoIP com Telefonia IP. Primeiramente, VoIP é apenas uma nomenclatura, a qual está intimamente ligada a Telefonia IP. Ela se refere à tecnologia de se utilizar a rede IP para transporte da voz. O VoIP é comumente usado como referência a qualquer tipo de ligação telefônica feita através de uma rede de dados. Como dito anteriormente, exemplos são a Internet ou uma rede local. Dizer que um sistema telefônico é VoIP não garante que o mesmo seja Telefonia IP.

A Telefonia IP é algo mais completo onde temos a garantia de que todos os equipamentos são realmente IP, capazes de interagir através de protocolos específicos da Telefonia IP, obedecendo todos os requisitos de qualidade de serviço (QoS) que são empregados para ter-se realmente uma telefonia confiável e de qualidade. [12]

Na telefonia IP as comutações das chamadas são realizadas por centrais IP ou *softswitches*. Estes equipamentos são nós da rede IP assim como um roteador, comutador ou um ponto de acesso sem fio. Sua função básica é a de manter um banco de dados de registro dos assinantes de determinada rede de voz trocando sinalização específica para o estabelecimento, controle e finalização de chamadas telefônicas. Esta sinalização trocada nada mais é do que o protocolo que os telefones IP utilizam para se “entenderem”. Uma vez registrado no banco de dados da central IP, o terminal obedecerá a regras bem definidas, como por exemplo as regras de discagem, assim como em uma central tradicional. Mais detalhes sobre os principais protocolos utilizados serão discutidos mais à frente.

Os ramais IP são equipamentos IP que codificam e decodificam a voz transfor-

mando-a em pacotes que trafegam na rede IP e trocam sinalização com o *softswitch*. Além dos ramais, existem também equipamentos que convertem a sinalização IP para a sinalização tradicional de um aparelho analógico, permitindo que aparelhos antigos possam também se registrar diretamente na central IP. Estes equipamentos são chamados de Adaptador para Telefone Analógico (ATA). Outras formas que os assinantes podem se registrar na rede de telefonia IP seriam através de *softphones* instalados em computadores pessoais ou em *smartphones*.

Além dos ramais IP, ATAs e *softphones*, encontramos também na rede de telefonia IP os *gateways*, que são equipamentos que fazem a integração da telefonia IP com a telefonia tradicional TDM. Estes *gateways* convertem os protocolos usados pela central IP em protocolos usados pelas centrais tradicionais, exercendo o papel de “portal” entre a telefonia IP e a telefonia legada. Quando um assinante da rede IP deseja estabelecer uma chamada com um usuário da rede legada, a central IP, através de sua tabela de encaminhamentos (rotas), encaminha a chamada para o *gateway* que está conectado àquela central legada cujo assinante chamado está conectado. A Figura 2.1 mostra uma topologia onde temos um serviço de voz tradicional utilizando a rede de dados IP para estender ramais telefônicos, onde o VoIP é empregado. Já a Figura 2.2 traz outra topologia onde se exemplifica uma rede de telefonia IP mais completa.

2.3 Vantagens

As três maiores vantagens da telefonia IP são funcionalidade, custo e mobilidade [12]. Tudo o que se precisa para seu funcionamento é uma infraestrutura de dados confiável. Dentro de um ambiente IP, além dos serviços tradicionais como chamadas em espera, conferências e transferências de chamadas, tem-se maior facilidade de interação com qualquer rede de dados existente, seja a própria Internet ou alguma rede privada, caracterizando maior funcionalidade da telefonia IP em comparação com a telefonia tradicional. Exemplo disto é o caso de escolas ame-

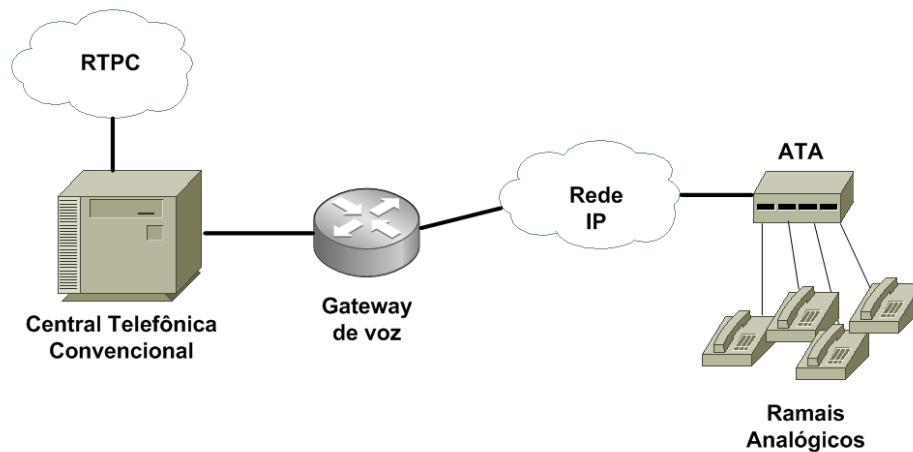


Figura 2.1: Extensão de ramais analógicos utilizando VoIP.

ricanas que já utilizam telefonia IP agregando controle de frequência dos alunos diretamente nos ramais dos professores [12].

A Telefonia IP traz consigo a capacidade do uso de rotas de menor custo usando a inteligência dos *softswitches* para rotear chamadas através da própria rede IP da companhia, evitando custos desnecessários com ligações de longa-distância, terminando, assim, as chamadas em *gateways* de borda mais próximos às cidades de destino. Estes *gateways* são posicionados estrategicamente nas cidades onde a empresa tem maior interesse de comunicação, com isso, todas as chamadas para aquela cidade de destino onde há um *gateway* instalado, passam a ter custo de chamadas locais, ao invés de chamadas interurbanas.

O usuário da telefonia IP pode ser alcançado em qualquer lugar do mundo utilizando um único número da rede pública (ou quantos achar necessário). Isso ocorre porque a central IP cria um vínculo entre o número da rede pública e um assinante de sua base de dados (Figura 2.3). Uma vez que este usuário pode estar em qualquer local dentro da rede IP da companhia, ao receber uma chamada através do número da rede pública, a central IP estabelece a chamada aonde quer que o assinante esteja, caracterizando assim enorme mobilidade. Este mesmo usuário ao originar uma chamada para algum telefone da rede pública, será visto pelo re-

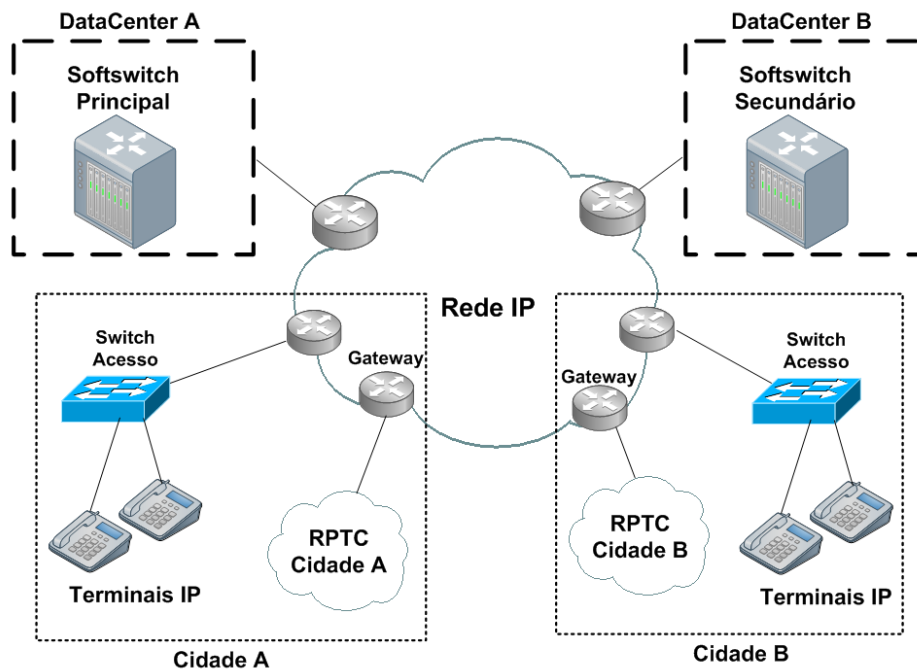


Figura 2.2: Topologia da telefonia IP.

cebedor como um outro usuário da rede pública, justamente por causa do vínculo que o primeiro recebeu. Por exemplo, ao ser vinculado com um número global de rede pública da cidade do Rio de Janeiro (+55212610MCDU), este usuário poderá realizar chamadas de qualquer lugar da rede IP de sua companhia para a rede pública, e isto ocorrendo, esta chamada ocorrerá como se o mesmo estivesse no Rio de Janeiro. Uma ilustração da mobilidade pode ser vista na Figura 2.4, onde um usuário do Rio de Janeiro se encontra na filial de Xangai e outro usuário de São Paulo encontra-se na filial de Nova York. Estes dois usuários são alcançados através de seus números globais de telefonia pública originais, mesmo estando longe de suas cidades de origem.

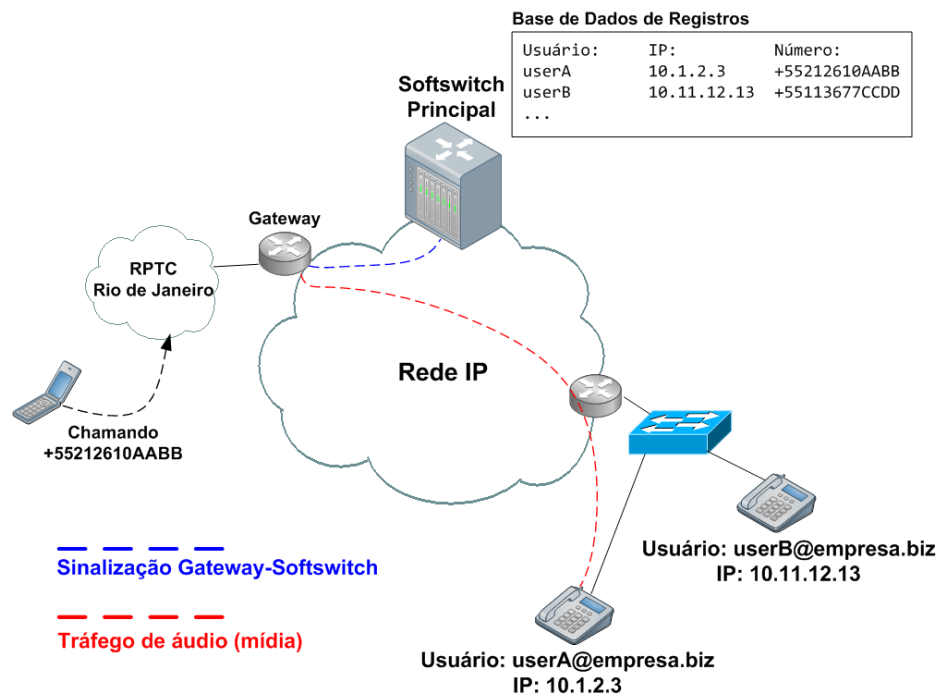


Figura 2.3: Registro na base de dados da central IP.

2.4 Protocolos

O serviço de voz sobre IP envolve dois tipos de protocolos, um para transportar os pacotes de voz, onde há a preocupação de como e por onde os pacotes trafegarão na rede, e outro para estabelecer a conexão suportando funcionalidades extras das chamadas. Estes dois tipos de protocolos são abordados abaixo.

2.4.1 Protocolos de transporte

Nas redes IP nos deparamos frequentemente diante dos dois protocolos de camadas 4 e 3, o *Transmission Control Protocol* (TCP) e o *Internet Protocol* (IP) respectivamente. Estes foram, originalmente, criados para suportar aplicações sem requisitos específicos de perdas ou atraso. Porém, hoje já existem protocolos de nível superior a estes, na camada 7, que dão melhor suporte ao tráfego de

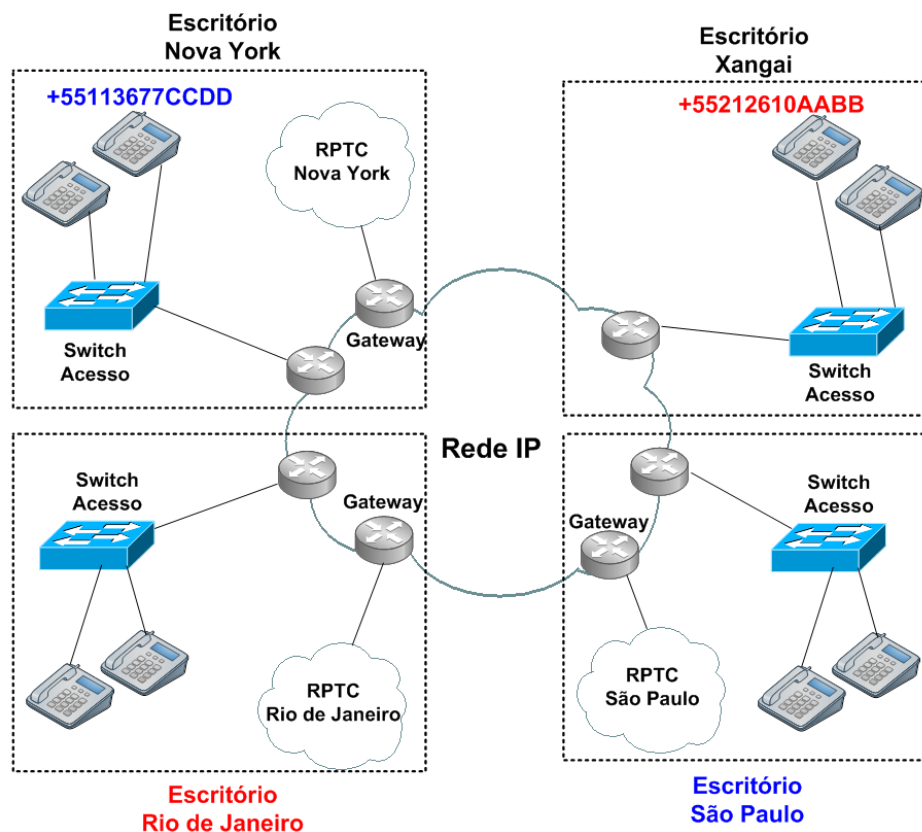


Figura 2.4: Mobilidade na telefonia IP.

mídia, como por exemplo o *Real-time Transport Protocol* (RTP) e o *RTP Control Protocol* (RTCP), que possuem mecanismos específicos para lidar bem com este tipo de informação pouco tolerante a atrasos. Do mesmo jeito que o IP, que é o protocolo de rede responsável pelo endereçamento e encaminhamento dos pacotes, dá suporte ao TCP para aplicações que necessitam de confiabilidade, este mesmo protocolo de camada 3 também dá suporte ao *User Datagram Protocol* (UDP). O UDP foi criado para realizar uma conexão sem confiabilidade, com pacotes de pequeno cabeçalho e conexões de pequena latência, características estas que se fazem necessárias quando temos na rede aplicações em tempo real [13], como por exemplo a telefonia.

Como dito, o UDP suporta melhor protocolos superiores que sofrem com atrasos da rede, como por exemplo o RTP e o RTCP. Estes protocolos são classificados

como protocolos de aplicação no Modelo TCP/IP, assim como o *Hypertext Transfer Protocol* (HTTP) [14], o *Simple Mail Transport Protocol* (SMTP) [15], entre outros.

Os dois protocolos utilizados para transportar voz através de redes IP são o RTP, responsável pelo transporte da voz, e o RTCP, responsável pelo desempenho do serviço da rede. Ambos os protocolos trabalham em conjunto, usualmente em cima do UDP [16].

2.4.1.1 *Real-time Transport Protocol* (RTP)

Os mecanismos de “bufferização” dos equipamentos utilizados nas redes IP trazem consigo o *jitter*, que é uma variação dos retardos sofridos pelos diversos pacotes enviados pela origem. A voz é tipicamente transportada em pacotes que contêm amostras de 20 a 40 ms de voz [12] [17]. Cada rajada de pacotes, em seu somatório, pode conter segundos de fala, e isso requer a característica isócrona dos circuitos comutados para que o reconhecimento da voz pelo receptor seja correto, ou seja, os pacotes podem até percorrer múltiplos percursos, porém devem chegar ao destino na mesma ordem que foram enviados e com a mesma relação de tempo que foram gerados. O RTP é utilizado para assegurar o sequenciamento dos pacotes e manter a relação temporal contínua, ou seja, evitar o *jitter*.

Durante uma conversação RTP, a origem coloca “selos de tempo” e uma sequência numérica no cabeçalho de cada pacote de voz antes que o mesmo seja enviado para a rede. Este cabeçalho também inclui um campo que indica o tipo de codificador de voz ou vídeo que está sendo utilizado. É esperado que os *buffers* dos elementos de rede pelos quais os pacotes passarão interrompam o sequenciamento e continuidade do fluxo dos pacotes. No receptor, os pacotes então são “bufferizados” pela última vez, porém agora, o RTP refaz o sequenciamento e os envia ao ouvinte com base nos “selos de tempo” [12]. Este processo efetivamente mascara o *jitter* e restabelece a continuidade temporal (Figura 2.5). O lado ruim do processo é que a “bufferização” final aumenta o atraso total da conexão (conversação).

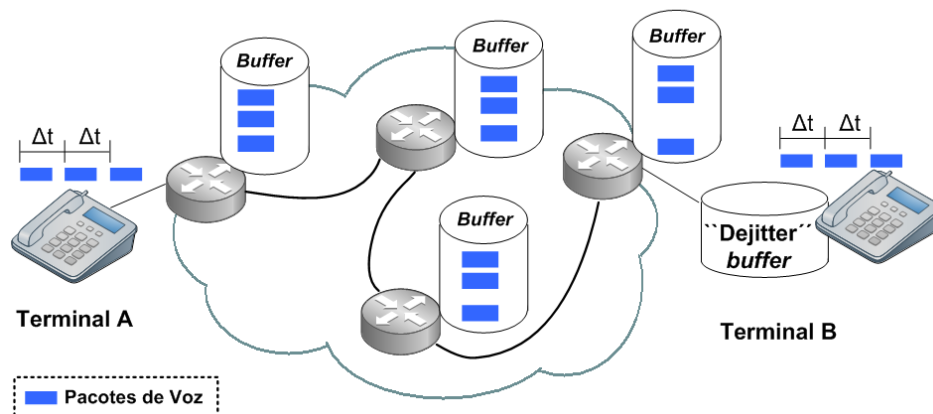


Figura 2.5: Jitter: a variação do atraso entre pacotes.

Pacotes em tempo real (voz e vídeo) são comumente encaminhados via UDP sobre o IP. Este protocolo de camada 3 pode perder pacotes ou entregá-los fora de ordem, e como o UDP não provê nenhum mecanismo para refazer o sequenciamento, o RTP é encarregado de tal função [12]. Porém, nem o UDP nem o RTP recuperam pacotes perdidos e, mesmo se o fizessem utilizando solicitação de retransmissão, esses pacotes retransmitidos chegariam ao destino muito tarde para ainda serem colocados na ordem correta. Com isso, a perda muito grande de pacotes degrada a qualidade de voz da conversação.

2.4.1.2 *RTP Control Protocol (RTCP)*

O RTCP é utilizado pelo RTP para monitorar a qualidade do serviço. O RTCP provê relatórios de monitoramento que ajudam o processo RTP. Esses relatórios são tipicamente gerados por cada nó da rede a cada 5 segundos, e são trocados entre os pares da conversação também via UDP, porém por portas diferentes das de tráfego de voz. As mensagens RTCP também podem ser monitoradas por uma ferramenta de monitoramento/desempenho e até mesmo pela própria operadora de acesso ou *backbone*. Usando pacotes RTCP como insumo para monitoramento das chamadas, a gerência da rede poderá ter uma visão em tempo real da qualidade

das chamadas em curso, além de poder criar um histórico dessas informações.

2.4.2 Protocolos de sinalização

Os protocolos de sinalização de telefonia tradicionais foram desenvolvidos para redes baseadas em circuitos determinísticos e assumiam que seus terminais não possuíam nenhum poder de processamento, quaisquer que fossem.

Estas sinalizações tradicionais envolviam desde fechamento do contato elétrico através da retirada do telefone do gancho pelo usuário, passando pelo chaveamento do circuito quando se discava o número do destinatário, até o envio do sinal de chamada para o assinante destino, e então a chamada era estabelecida. Mais tarde, a Rede Digital de Serviços Integrados (RDSI), em inglês *Integrated Services Digital Network* (ISDN), introduziu a ideia de que as trocas de mensagens de sinalização entre as centrais telefônicas e o assinante fossem realizadas utilizando-se um canal separado do canal da fala, criando assim uma rede de sinalização separada. Isso foi possível devido ao avanço das centrais e dos terminais, os quais passaram a ser digitais.

Existem 4 principais protocolos de sinalização utilizados em sistemas de telefonia IP.

2.4.2.1 H.323

Desenvolvido pelo *International Telecommunication Union* (ITU) para suportar videoconferência sobre a rede de pacotes, este protocolo foi atualizado para suportar também os requisitos de telefonia IP. Existem 4 versões de H.323 e todas trabalham em conjunto com o TCP. Cada fabricante criou uma versão proprietária do H.323, por isso, muitas vezes terminais de um fabricante não se comunicam com *softswitch* de outro fabricante [12] [18].

2.4.2.2 *Media Gateway Control Protocol* (MGCP)

Este protocolo de sinalização é usado principalmente em gateways, em um sistema de telefonia distribuído, visando a integração de redes de telefonia IP com redes de arquitetura SS7 (telefonia tradicional) [12] [19]. A evolução do *Media Gateway Control Protocol* (MGCP) resultou no *Media Gateway Control* (MEGACO), através da RFC 5125 [20].

2.4.2.3 *Media Gateway Control* (MEGACO/H.248)

Para redes onde há muitos gateways, uma boa prática é o uso de um controlador de gateway. O MEGACO ou H.248 é justamente o protocolo de controle utilizado exclusivamente para a comunicação entre o controlador e os gateways da rede, sendo um protocolo complementar ao SIP e H.323, os quais são utilizados para comunicação do gateway com a central IP [12] [20].

2.4.2.4 *Session Initiation Protocol* (SIP)

O SIP [21] foi desenvolvido pela *Internet Engineering Task Force* (IETF) como um padrão IP. As principais vantagens do SIP são a interoperabilidade entre fabricantes, fácil implementação e operação similar a outros protocolos IP. Este protocolo opera tanto sobre o TCP quanto sobre o UDP.

As possibilidades de integração do SIP não se restringem somente à telefonia IP propriamente dita, podendo também prover a interoperabilidade de forma geral, através das comunicações unificadas [22].

A maioria dos sistemas de telefonia IP já está colocando o SIP como protocolo de sinalização padrão. O SIP suporta cinco funções básicas para o estabelecimento e terminação de uma comunicação multimídia.

- Localização de usuário: determina localização do terminal;

- Disponibilidade do usuário: determina disponibilidade dos tipos de serviço que o usuário chamado tem (ex. voz, vídeo, mensagem instantânea);
- Capacidade do usuário: determina disponibilidade dos tipos de mídia aceitas pelo usuário (ex. tipo de codec);
- Estabelecimento da sessão: estabelece parâmetros da sessão para os terminais utilizando o *Session Definition Protocol* (SDP); Toda mensagem SIP que necessita definir os requerimentos de capacidade de uma chamada vai anexar em seu corpo uma mensagem SDP. Esta mensagem vai, por exemplo, informar quais *codecs* o chamador suporta.
- Gerenciamento da sessão: gerencia funcionalidades durante a chamada (ex. transferência, conferência).

O protocolo SIP descreve elementos de rede que desempenham funções específicas durante uma sessão, sendo eles:

- *User Agent* (UA): este é o terminal SIP. Telefones IP, telefones WLAN, softphones e ATA's são exemplos de UA's.
- *Registrar*: é um servidor que rastreia onde o terminal está localizado. Ele também permite o registro da disponibilidade de cada usuário em determinado momento e reconhece quais terminais podem ser acessados para conectar em determinado serviço.
- *Redirect Server*: redirecionam o terminal para conectar determinados serviços que não estão disponíveis localmente.
- *Proxy Server*: é o primeiro elemento envolvido no estabelecimento das trocas de mensagens entre os terminais. É o *softswitch* da Telefonia IP usando SIP.

Os três últimos elementos são integrados no Proxy Server, com isso, geralmente só existem UA's e Proxy Servers em uma rede SIP.

A sinalização SIP depende do uso de mensagens baseadas em textos, que facilitam a descoberta de eventuais problemas que podem ocorrer. Existem vários tipos de mensagens, das quais as principais são (Figura 2.6):

- OK: mensagem usada na confirmação de um solicitação;
- REGISTER: mensagem usada para o registro do UA no Proxy Server;
- INVITE: mensagem usada para iniciar uma sessão ou chamada SIP;
- UPDATE: mensagem usada para atualizar a característica de uma chamada SIP;
- BYE: mensagem usada para terminar uma sessão ou chamada SIP;

2.4.2.5 Estabelecimento da chamada - *Call-Setup*

Como sabemos, o SIP foi projetado para suportar diversos tipos de serviços, como voz, vídeo, multimídia, mensagens instantâneas, entre outros. Para o início da sessão de qualquer um desses serviços é necessária a realização do *Call Setup*, que nada mais é do que a troca de sinalização prévia dos terminais entre si e com o *softswitch* (central telefônica IP).

A seguir estão os passos do processo de *Call Setup* em uma rede SIP:

1. Para iniciar a conexão, o terminal origem envia uma mensagem INVITE para o terminal destino. O endereço chamado recebe o nome de *Uniform Resource Identifier* (URI) que se parece com um endereço de e-mail. (ex. sip:usuario@servidor.com). O SIP endereçará esta mensagem para o destino através de um ou diversos SIP Proxy's.
2. O terminal destino responde ao INVITE com uma mensagem "180 RINGING" destinada ao SIP Proxy, e este encaminha-a para o terminal origem.

3. Quando o terminal destino atende a chamada, gerando uma mensagem “200 OK” que é enviada para a origem pelo SIP Proxy, o tráfego de voz ou vídeo, então, se inicia. Os pacotes da chamada trafegam diretamente entre os terminais, não passando pelo SIP Proxy.
4. Caso um dos UA’s desejar alterar a característica da chamada em curso, este envia para seu par uma mensagem de UPDATE, que é respondida através de outra mensagem “200 OK” ou “488 NOT ACCEPTABLE”.
5. Ao término da chamada (*hang-up*), qualquer um dos terminais gera uma mensagem “BYE”, a qual é respondida com a mensagem “200 OK”, e então a chamada é desconectada.

2.5 Qualidade

O uso da tecnologia de “voz sobre pacotes” insere algumas grandes questões de qualidade: qualidade de serviço em geral, qualidade da voz (propriamente dita), atraso/*jitter* dos pacotes e controle de eco da conversação.

2.5.1 Qualidade de serviço (QoS)

Enquanto temos uma estrutura de protocolos para suportar a voz, a rede IP trabalha no modo *best effort* (onde usuários não têm garantia de taxa), significando que a rede não provê nenhuma garantia quanto a atraso, *jitter*, perda, sequência de pacotes, entre outros.

Como abordado, o RTP tem a função de ressequenciar os pacotes de voz e restabelecer a continuidade dos mesmos [12], porém não recupera pacotes perdidos. Portanto, quanto melhor e mais consistente os serviços das camadas inferiores, melhor será a qualidade do sistema de uma forma geral. Como a rede IP é uma rede compartilhada, pacotes não prioritários, como os de *e-mail*, acesso *web*, etc, não

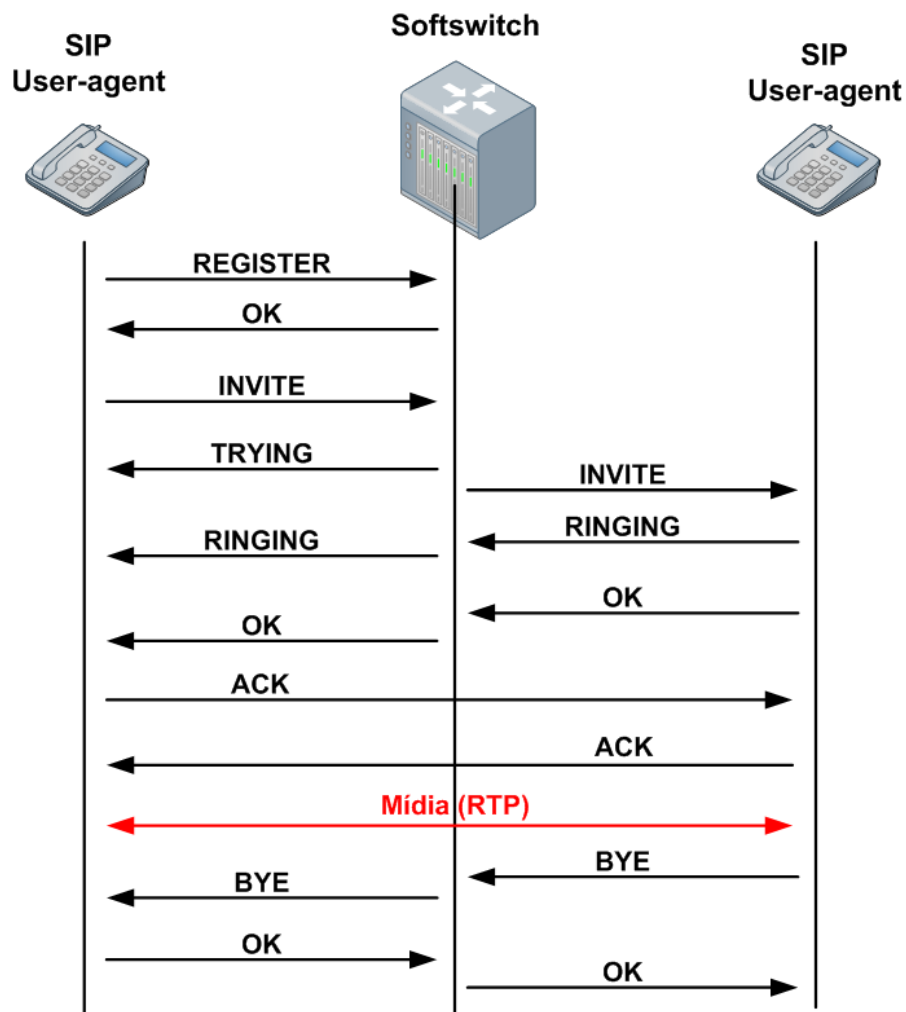


Figura 2.6: Mensagens SIP durante registro, iniciação e término de uma chamada.

possuem a mesma necessidade de serem encaminhados tão rapidamente quanto os pacotes de voz, então por este motivo existe a necessidade do *Quality of Service* (QoS), que dá a consistência necessária aos serviços prioritários, evitando o aumento dos parâmetros de atraso, *jitter* e perda de pacotes.

2.5.1.1 QoS de camada 2

A camada 2 define as funções dos protocolos de redes locais como Ethernet e há diferentes formas de definir a qualidade de serviço para este ambiente. As principais formas são:

802.1p O Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE) desenvolveu um mecanismo para redes locais cabeadas em seu padrão 802.1p. Usados nos switches das redes locais, o padrão insere um campo adicional ao cabeçalho do quadro ethernet, o qual inclui um número de 12 bits para identificação de VLAN's e um número de 3 bits indicando prioridade daquele quadro. A ideia é que cada switch da rede possa reorganizar o encaminhamento dos quadros que passam por ele se baseando na prioridade que aquele quadro se encaixou ao passar pelo primeiro switch desde sua origem [12].

802.1e Enquanto o 802.1p provê a prioridade através dos switches, em uma rede sem fio, os usuários competem pelo acesso em um único canal de rádio compartilhado. O padrão 802.11e determina que os quadros marcados com alta prioridade “aguardam” por uma disponibilidade de envio durante um tempo menor, e assim, têm acesso preferencial ao canal de rádio compartilhado [12].

2.5.1.2 QoS de camada 3

Existem dois principais métodos para se priorizar um pacote IP:

Differentiated Services (DiffServ) Neste método, pacotes exclusivos são encarregados de fazer a indicação de prioridade. Esta indicação reside no campo *Type of Service* (TOS) do cabeçalho IP. Roteadores habilitados para trabalhar com o *diffserv* reconhecem esta prioridade e mantêm filas separadas para cada categoria de tráfego. Quando pacotes chegam ao roteador, eles são classificados e

colocados nestas filas baseando-se no indicador de prioridade *diffserv*. Os pacotes de cada fila são encaminhados conforme a prioridade atribuída àquela fila [23].

Multi-Protocol Label Switching (MPLS) MPLS é um método de QoS mais abrangente que foi sendo implementado primeiramente em *backbones* de provedores de Internet, porém atualmente também está sendo configurado em roteadores de algumas empresas. Os provedores de Internet utilizam o *Multi-Protocol Label Switching* (MPLS) como motivo para venderem serviços especiais diferenciados e, logicamente, mais caros. Estes serviços são chamados de *MPLS Virtual Private Network* (MPLS-VPN). A chave deste serviço é que ele provê garantia para atraso, *jitter* e perda de pacotes. As operadoras que possuem redes MPLS oferecem múltiplos níveis de prioridade para suportar voz, vídeo e dados. O serviço MPLS pode garantir o desempenho porque o recurso para cada usuário é reservado no início do provimento de serviços, então um provedor de Internet poderia oferecer, por exemplo, dois acessos a Internet, o usual *best effort* e outro baseado em MPLS-VPN com desempenho garantido. É importante frisar que as garantias de qualidade somente cobrem tráfego na rede da operadora. Por exemplo, ao contratar um circuito entre dois sites de uma empresa, é essencial que a operadora contratada atue em todo o trajeto do circuito entre as localidades, evitando, assim, a terceirização de última milha, o que impossibilitaria o uso do serviço garantido [24].

2.5.2 Qualidade de voz

A voz produz um sinal que é naturalmente análogo às ondas mecânicas geradas pelas cordas vocais. Quando falamos, nossa voz cria um padrão de vibração e pressão no ar que é essencialmente um sinal analógico. Antes de ser transportado numa rede de pacotes, este sinal precisa ser convertido em bits. O papel de conversor é exercido pelo codec, abreviação para *coder* (no emissor) e *decoder* (no receptor).

Existem três questões a serem consideradas para a seleção do tipo de sistema

de codificação da voz:

- A taxa de transmissão;
- O atraso inserido pelo processo de codificação;
- A tolerância à perda de pacotes.

Qualquer técnica que converta voz em bits introduz alguma degradação na qualidade do som. Em geral, estas degradações são indistinguíveis ao ouvido humano. Entretanto, caso alguns destes bits sejam modificados devido a erros de transmissão, ou estejam faltando devido à perda de pacote, a qualidade da voz poderá ser afetada. Há muitas técnicas de codificação de voz e elas variam levando-se em consideração sua eficiência, robustez e atraso (de processamento).

Uma codificação de voz eficiente reduz o número de bits necessários por canal de voz, e, devido a isto, aumenta o número de canais de voz simultâneos que podem ser utilizados na rede. Como regra geral, um codec mais eficiente traz maior atraso e qualquer descarte de bits afeta mais a inteligibilidade da conversação.

Alguns típicos codificadores de voz são mostrados na Tabela 2.1 [25].

Codec	Taxa de Bits (kbps)	Atraso de Codificação (ms)	Tolerância a Perda (%)	Aplicações
G.711 (PCM)	64	0,13	10	Rede pública e PABX's
G.726 (ADPCM)	24, 32 ou 40	0,4	5	T1 e aparelhos telefônicos domésticos sem fio
G.729a	8	25	<2	VoIP via WAN
G.723.1	5,3 ou 6,4	67	<1	Limitado por causa do atraso
G.722	64	0,4	5	Radiodifusão e sistemas de conferência

Tabela 2.1: Codecs típicos.

G.711 (*Pulse Code Modulation*) As redes telefônicas tradicionais utilizam esta técnica de codificação descrita pela recomendação ITU G.711. Nesta técnica, um canal de voz é representado por um *stream* de bits de 64 kbps. Esta é uma alta

taxa comparativamente aos outros códigos, e, por causa disto, esta técnica suporta até 10% de perda de pacotes e, ainda assim, provê qualidade de som aceitável.

O processo de codificação *Pulse Code Modulation* (PCM) envolve 2 passos: amostragem e codificação.

O primeiro passo é feito retirando-se 8000 amostras por segundo do sinal analógico da voz, medindo-se a amplitude deste sinal.

No segundo passo o algoritmo de codificação arredonda o valor obtido anteriormente e o converte para um novo valor binário de 8 dígitos.

Com isso, tem-se 8000 amostras de 8 bits cada no período de 1 segundo. Isso leva a uma taxa de transmissão binária de 64000 bits por segundo para cada canal de voz.

As próximas técnicas de codificação da voz também são categorizadas como técnicas de compressão. Os resultados do uso destas técnicas, como já dito anteriormente, são a diminuição da taxa por canal de voz e o aumento do número de canais simultâneos na rede.

G.729 Esta é uma técnica patenteada e é uma atualização do padrão ITU. O G.729 define um mecanismo de compressão do PCM a uma taxa de 8 kbps. Esta compressão dura, em média, 25 ms e produz um sinal com mínima tolerância a perda de pacotes. A qualidade de voz cai abaixo do aceitável se apenas 1% dos pacotes forem perdidos.

SPEEX - Este é um codec de código aberto, baixas taxas (*bitrate*) e similar ao G.729, porém possuindo taxas que variam de 2 a 44 kbps, dependendo da faixa de frequência de áudio que estiver captando – *Narrow Band* (NB) ou *Wide Band* (WB).

G.722 O G.722 provê altas taxas de bits, por isso a baixa eficiência de sua técnica. Um típico codec (G.711) captura frequências de até 3.1 kHz. O G.722 utiliza um mecanismo de codificação aprimorado que pode capturar frequências de até 7 kHz no mesmo *stream* de 64 kbps. Isso significa mais fidelidade do som com a mesma taxa de bits. Esta codificação é utilizada onde a banda da rede não é fator escasso.

Analisando os requisitos de transmissão é importante ressaltar o cabeçalho dos pacotes. Cada pacote receberá um cabeçalho RTP, UDP, IP e mais um cabeçalho de camada 2, como por exemplo PPP (WAN) ou ethernet (LAN). Para exemplificar isto, a Tabela 2.2 [25] mostra o total de taxa de bits produzida por cada técnica de codificação considerando-se diferentes tamanhos de amostragem e protocolos de transporte.

CODEC	Taxa de bits (kbps)	Largura de banda de voz (kHz)	Tamanho das amostras (ms)	Tamanho do pacote (bytes)	Pacotes por segundo	Taxa em Ethernet (kbps)
G.711 (PCM)	64	3	20	160	50	87,2
G.711 (PCM)	64	3	30	240	33,3	79,4
G.711 (PCM)	64	3	40	320	25	75,6
G.722	64	7	20	160	50	87,2
G.722	64	7	30	240	33,3	79,4
G.722	64	7	40	320	25	75,6
G.726 (ADPCM)	32	3	20	160	50	55,2
G.726 (ADPCM)	32	3	30	240	33,3	47,4
G.726 (ADPCM)	32	3	40	320	25	43,6
G.729a	8	3	20	160	50	31,2
G.729a	8	3	30	240	33,3	23,4
G.729a	8	3	40	320	25	19,6
Speex NB	24	8	—	—	—	40,0

Tabela 2.2: Codecs e suas taxas.

2.5.3 Atraso e *jitter*

Estes dois itens são os principais vilões quando trabalha-se com tráfego em tempo real sobre redes IP.

O atraso é o somatório de tempo que o pacote leva percorrendo desde o seu terminal de origem até o terminal de destino, passando por todos os elementos de rede no meio do caminho. Isto é o que chamamos de “*mouth-to-ear delay*”

(atraso boca-ouvido). Sistemas de voz comutados por circuitos introduzem atrasos insignificantes, tipicamente abaixo de 10 ms. Já na telefonia IP esses atrasos podem variar dada a característica não determinística dos enlaces.

O atraso é uma combinação de vários fatores, dentre os quais se destacam:

1. Codificação da Voz: Como sabe-se, quanto mais eficiente o codec de voz, menor a taxa de bits será exigida por canal de voz, e, conseqüentemente, mais processamento este codec demandará, acrescentando mais atraso ao sistema.
2. Geração do pacote: Durante a codificação da voz o codec retira uma amostra da voz analógica e a codifica. Esta amostra varia normalmente entre 20 e 40 ms, porém recomenda-se utilizar amostragens de 20 ms para minimizar o atraso na geração do pacote de voz. O lado ruim de utilizar amostras menores é que o cabeçalho do pacote não diminui em nada e isso leva ao aumento da relação cabeçalho/conteúdo.
3. Atraso de transmissão: Quando um quadro é enviado através de uma rede (cabeadada ou sem-fio), o mesmo obedece ao protocolo de acesso ao canal: *Carrier Sense Multiple Access with Collision Detection* (CSMA/CD) em cabeamento metálico e *Carrier Sense Multiple Access With Collision Avoidance* (CSMA/CA) em enlaces de rádio. Isto significa que há uma pequena espera antes de o quadro ser realmente enviado, e em havendo uma colisão ou outro problema de transmissão, o mesmo terá de ser retransmitido. Após isto, o equipamento de rede receptor precisa certificar-se de que o quadro foi recebido sem erros para então enviá-lo a seu próximo destino. O objetivo do QoS de rede, como visto, é de minimizar este atraso gerado para aplicações de tempo real. Outro importante fator para o atraso de transmissão é a taxa do link, pois esta também determinará o quão rápido o *bit* será “lançado” ao meio de transmissão.
4. Atraso de propagação: Este atraso é relativo às distâncias físicas durante o

percurso do sinal elétrico/eletromagnético. Em redes locais onde este atraso é mínimo, teremos, no pior caso (enlace cabeado de 100 m), um atraso de aproximadamente 500 ns. Porém, atrasos de propagação se tornam significantes quando estes enlaces constituem redes WAN ou WLAN.

5. Atraso de switches e roteadores: Tipicamente este é o maior gerador de atrasos variáveis (*jitter*), e são compostos de dois tipos:

- Atraso por comutação: cada quadro/pacote que passa por switches/roteadores precisa ter seu cabeçalho checado. Posteriormente estes nós da rede analisarão suas tabelas de encaminhamento (tabela MAC nos switches e tabela de roteamento nos roteadores) e então encaminharão o quadro/pacote para o correto destino.
- Atraso por “bufferização”: este é o atraso que gera o *jitter*. O nó de rede processa um quadro/pacote por vez, devido a isso, os pacotes que chegam ao nó entram em uma fila de espera sendo armazenados em uma memória chamada de *buffer* e esperam a sua vez de serem processados. Este tempo de espera depende de quão cheio o buffer se encontra e qual a prioridade foi atribuída ao mesmo, visto que existirá um *buffer* para cada fila de prioridade.

6. Processo de retirada do *jitter* (*dejitter*): Uma vez que os pacotes chegam ao seu destino, o protocolo RTP do receptor há de restabelecer a ordem e continuidade temporal dos mesmos [12]. Isso requer também uma “bufferização” dos mesmos para que o RTP analise a marcação dos pacotes (“selos de tempo”) e os coloque na ordem certa [12]. Este é o último gerador de atraso antes do usuário terminar de receber a voz em seu ouvido.

2.5.4 MOS e o Modelo E do ITU

Um dos mais importantes parâmetros para se medir a qualidade da fala em uma chamada é o MOS, uma pontuação que varia de 1 a 5, sendo 5 relativa a

uma conversação perfeita. Porém este é um método subjetivo por essência, pois sua forma de pontuação é dada por ouvintes que analisam a amostra de áudio atribuindo uma nota. Duas formas de medição são atribuídas ao MOS: o MOS-LQ (*listening quality*) e o MOS-CQ (*conversational quality*) [12]. O MOS-LQ mede a qualidade do áudio somente para a escuta, não levando em conta aspectos bidirecionais, o que caracteriza a conversação, como eco e atraso, que é algo que o MOS-CQ faz.

Por outro lado, existe um segundo tipo de análise do áudio chamado “Teste Objetivo Baseado em Amostra”. Desenvolvidas pelo ITU, as recomendações P.861 e P.862 utilizam uma amostra de áudio perfeita que é passada por algum codificador (PCM, G.729, etc) e então a saída é comparada com uma versão distorcida da mesma amostra obtida de uma transformada de Fourier, quando então uma nova pontuação é obtida. Denomina-se essa pontuação de *Perceptual Evaluation of Speech Quality* (PESQ). Como esta medição não envolve uma conversação, sua pontuação é parecida com o MOS-LQ [12].

Como o PESQ envolve certo processamento, o *European Telecommunications Standards Institute* (ETSI) desenvolveu o Modelo E, que também considera aspectos conversacionais em sua fórmula, resultando no Fator R. Este modelo foi adotado pela recomendação G.107 [26] do ITU e agora é o padrão de medição de qualidade de voz em redes VoIP [26].

O Modelo E utiliza uma equação que leva em consideração o nível de ruído (R_0), atenuação do sinal (I_s), deterioração pelo atraso e eco (I_d), deterioração pela transmissão dos sistemas VoIP (I_e) e a expectativa do usuário (A).

$$R = R_0 - I_s - I_d - I_e + A$$

Com uma escala variando de 0 a 120, o ITU criou a Tabela 2.3 de equiparação entre o valor de R e o valor de MOS.

Para obter computacionalmente o MOS, aplicações de monitoramento da qua-

	Fator R	MOS (Escala do ITU)
Máximo da escala	120	5.0
Máximo atingível (G.711)	93	4.4
Qualidade boa	>80	>4.0
Qualidade aceitável	>70	>3.6
Qualidade baixa	>50	>2.6
Não recomendado	<50	<2.6

Tabela 2.3: Fator R e MOS.

lidade de voz, baseando-se na RFC 3611 [27], podem converter os parâmetros obtidos através dos pacotes RTCP em um Fator R do Modelo E, e este, através da tabela do ITU, em um valor de MOS.

2.6 Motivação do trabalho

O sistema de telefonia é um serviço básico em qualquer companhia, e tê-lo indisponível por um curto período de tempo, por menor que seja, pode ser crucial para o setor afetado, podendo até gerar prejuízos financeiros para a empresa como um todo.

Atualmente, grandes companhias estão migrando seus sistemas de telefonia tradicionais para voz sobre IP. Redes VoIP trazem a vantagem de ter um comutador centralizado, a central telefônica IP ou simplesmente *softswitch*, que oferece serviços de telefonia com algumas vantagens, como rota de menor custo (a qual é difícil de se implementar em redes de telefonia tradicionais) e economias operacionais na redução de circuitos dedicados de voz, uma vez que os circuitos de dados transportarão o serviço de voz sobre IP.

2.6.1 Vantagens da telefonia IP

Rotas de menor custo As rotas de menor custo usam a inteligência das centrais IP para rotear chamadas através da própria rede IP da companhia, evitando custos desnecessários com ligações longa-distância, terminando as chamadas em gateways de borda mais próximos das cidades de destino. Exemplo disto é mos-

trado na figura 2.7, onde usuários que desejam realizar ligações para outras cidades - DDD - ou até mesmo para outro país - Discagem Direta Internacional (DDI) - usam a infraestrutura IP da própria companhia para que a chamada seja efetivamente realizada como uma chamada local.

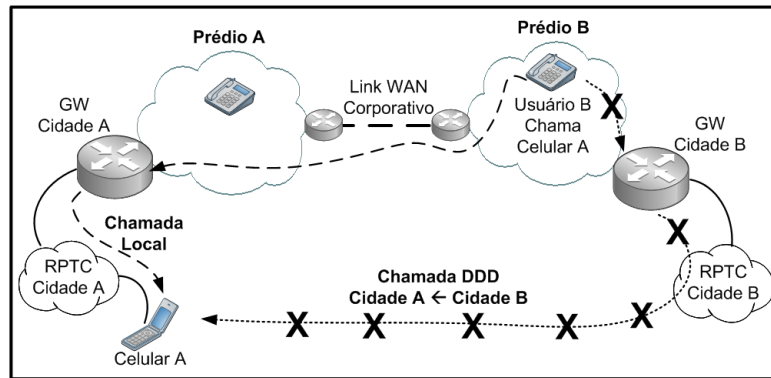


Figura 2.7: Rota de menor custo.

Contudo, como a largura de banda é algo limitado, e usando a infraestrutura IP também para telefonia, esta banda se torna item crucial, em contraste com a telefonia tradicional, onde circuitos determinísticos são usados exclusivamente para voz.

Por exemplo, usando rede IP para tráfego de voz, se um enlace WAN está congestionado durante uma chamada de voz importante ou uma audioconferência, a chamada simplesmente pode cair ou apresentar falhas na conversação. Por este motivo são usadas técnicas como melhores práticas para implantação do serviços de voz sobre a rede de dados IP. Essas técnicas reduzem a degradação do serviço de voz em redes IP, das quais destacamos duas: QoS e Controle de Admissão de Chamadas (CAC) [28].

Comunicações unificadas Outro importante benefício que a telefonia IP traz é a possibilidade de unificação de diversos meios de comunicação [5], como o telefone, *smartphone*, *softphone*, e-mail, entre outros.

De forma centralizada, os componentes que controlam o serviço de telefonia IP,

integrados a outros controladores do sistema de comunicações unificadas, permitem que os usuários sejam alcançados por diversos meios, sempre através do mesmo contato: seja um endereço de e-mail ou *chat*, seja um número de ramal interno da empresa ou externo da RPTC. Este benefício pode ainda ser incorporado através de servidores de código aberto [29], otimizando ainda mais os custos de investimento da companhia.

2.6.2 Melhorias aplicáveis

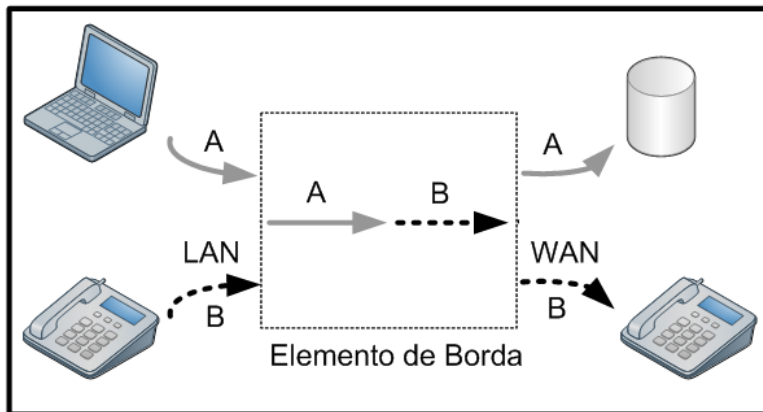
QoS - qualidade de serviço Para garantir que o serviço de voz não seja degradado durante um congestionamento do enlace, o QoS é comumente implementado em comutadores ou roteadores de borda. Basicamente, esses equipamentos checam quais tipos de pacotes estão chegando em suas interfaces de rede local *Local Area Network* (LAN) e que estão competindo pela banda da interface de saída WAN. A técnica de QoS mais comum implementada para aplicações de tempo real é a *Low-Latency Queuing* (LLQ) - enfileiramento de baixa latência [30] - em filas baseadas em peso *Class-Based Weighted Fair Queuing* (CBWFQ). Essas filas encaminham os pacotes identificados como aplicações de tempo real, passando-as na frente dos outros para serem entregues sem atraso. Por exemplo, um telefone IP deve marcar seus pacotes de voz com um valor específico para o campo *Differentiated Services Code Point* (DSCP) [23] do cabeçalho IPv4. Baseado neste valor, o equipamento comutador/roteador pode facilmente priorizar pacotes de voz em detrimento de outros pacotes.

Controle de Admissão de Chamadas (CAC) Mesmo que as técnicas de QoS sejam eficientes para garantir que os pacotes de voz sejam encaminhados antes de outros pacotes, isso não é o suficiente para assegurar que o atraso e as condições de banda para o tráfego de voz sejam atendidos. Caso um enlace WAN tenha, por exemplo, 30% de sua banda dedicada para tráfego de voz, durante a Hora de Maior Movimento (HMM), existe a possibilidade de que os 30% não sejam

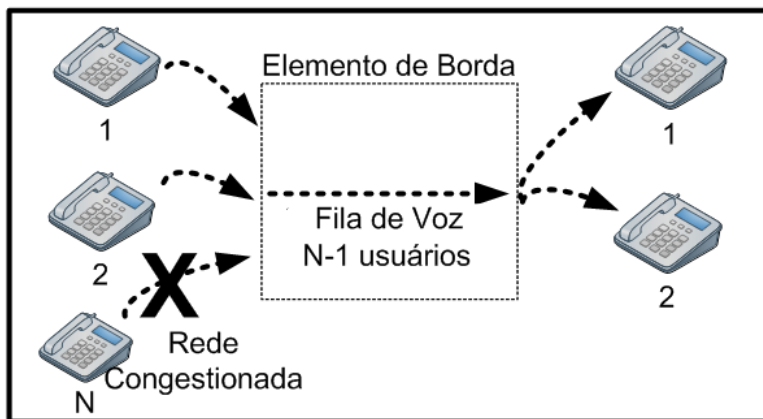
suficientes para a quantidade de chamadas, e então somente parte dos pacotes de voz serão encaminhados sem atraso, resultando em queda da qualidade de voz para todas aquelas chamadas que passam por aquela fila de voz congestionada.

O CAC limita a um número máximo a quantidade de chamadas simultâneas que passa por determinado enlace WAN, baseando-se na banda da fila de voz daquele enlace. Com isso, garante-se que 100% dos pacotes de voz sejam encaminhados sem atraso, pois se a banda da fila de voz chega ao seu limite, o CAC atua e não permite que uma nova chamada seja estabelecida até que outra seja desconectada.

Ambas as técnicas de QoS e CAC são úteis e comuns à maioria das configurações de rede das companhias, mas estas implementações ainda são limitadas, pois o CAC poderá, por exemplo, bloquear chamadas durante a HMM (Figura 2.8). Contornar esse problema seria uma grande vantagem para empresas que possuem alto tráfego de chamadas inter-sítios na HMM. Este trabalho vislumbra a otimização do uso do link WAN realizando um controle de taxa dos *streams* de voz para dar robustez ao enlace ora congestionado.



(a) Técnica de QoS: tráfego priorizado alcança interface de saída primeiro.



(b) Técnica de CAC: número de possíveis chamadas adequado com total de banda.

Figura 2.8: Técnicas de QoS e CAC.

Capítulo 3

Redes Definidas por Software: SDN

3.1 Histórico

Há pouco tempo atrás os recursos de rede, processamento e armazenamento eram todos dispostos de forma separada em termos físicos e operacionais, tudo em prol das políticas de segurança. Esse era o modo que as equipes de Tecnologia da Informação (TI) entendiam ser o correto, e somente quando a forma de processamento, armazenamento e rede se tornou muito mais eficiente sendo instaladas dentro de Centros de Processamento de Dados (CPDs) centralizados, a mentalidade dos profissionais de TI mudou. Os CPDs foram originalmente desenvolvidos para separar os servidores dos usuários, facilitando o gerenciamento e permitindo o compartilhamento dos recursos desses servidores entre os clientes [8].

Há mais de 10 anos uma transformação ocorreu. Uma empresa chamada *VMWare* inventou uma tecnologia que permitia que um computador rodando um sistema operacional pudesse executar um ou mais sistemas operacionais distintos. Essa empresa na verdade criou um pequeno aplicativo de supervisão que sintetiza um real ambiente computacional, permitindo que outros sistemas operacionais compartilhem CPU, memória, disco, placa de rede, etc. do hardware hospedeiro. Esse *software* foi batizado de *hypervisor*.

Com isso, servidores que tipicamente rodavam um único e dedicado sistema operacional para uma determinada aplicação, agora poderiam ser executados em um hardware mais poderoso que fosse capaz de supervisionar diversos sistemas operacionais executando diferentes aplicações. Com os avanços da informática, os processadores, memórias, discos rígidos, etc. ficaram muito mais poderosos, e então a *VMWare* aprimorou sua solução para que várias máquinas hospedeiras rodassem centenas de máquinas virtuais a partir de uma única *cluster* de *hypervisor*. Assim começou a era da computação elástica (*elastic computing*).

A computação elástica permite que qualquer máquina virtual possa ser movida de um CPD para outro. Isso é uma vantagem devido ao fato de que se um grupo de usuários acessa mais uma determinada aplicação em um CPD distante, essa aplicação que está instalada em uma máquina virtual da *cluster* pode ser movida com alguns cliques do mouse para um CPD mais próximo daqueles usuários, otimizando assim recursos de rede.

Ao passo que os *datacenters* evoluíam, os equipamentos de rede pareciam estagnar em termos de inovações. Apesar do crescimento de capacidade de processamento e de velocidade de interfaces de rede, as comunicações de dados nada evoluíram desde a chegada do IP, MPLS e tecnologias de mobilidade. O IP e o MPLS permitiram que os operadores de rede virtualizassem várias redes privadas dentro de um único equipamento de rede, seguindo a ideia da virtualização da *VMWare* em CPDs, e com isso, poderia isolar o tráfego de rede de diferentes empresas usando um mesmo equipamento [8].

Os equipamentos de rede tipicamente são fornecidos com capacidades de gerenciamento que permitem que o operador configure os recursos do hardware de forma limitada, por exemplo, configurando rotas estáticas, habilitando processos de roteamento, lista de acessos em interfaces, etc. Mas se por um motivo o administrador de rede deseja testar um novo tipo de protocolo usando aquele hardware, nada poderá ser feito, uma vez que o *firmware* do equipamento não permite a adição de tal protocolo a ser experimentado.

Ao mesmo tempo, o conceito de plano de controle distribuído vem à tona. Um equipamento de rede é composto de um plano de dados que conecta várias interfaces de rede a um barramento de comutação de quadros ou pacotes, e de um plano de controle que é a inteligência do equipamento. Por exemplo, equipamentos de rede que possuem o protocolo *spanning-tree*, rodam tal software em seus planos de controle, trocam mensagens específicas com o mesmo objetivo: evitar loops de rede. Reparem que para que o loop de rede não ocorra, os equipamentos adjacentes **devem** trocar mensagens, portanto se trata de uma forma distribuída do controle da rede.

3.2 Conceito

Todas as características mencionadas acima são importantes, pois criaram a motivação para o que evoluiu no que chamamos de Redes Definidas por Software (*Software Defined Networks* - SDN). Pela definição de Nadeau, “Redes Definidas Por Software são uma abordagem de arquitetura que otimiza e simplifica as operações de rede por proporcionar a interação mais próxima entre as aplicações, serviços e equipamentos, sejam reais ou virtualizados.” [8]

Essas redes formam um novo conceito para o desenvolvimento de pesquisas que vem ganhando muita atenção de grande parte da comunidade acadêmica e da indústria na área da computação e redes. Atualmente, o SDN tomou grande importância no mercado, dado que empresas como Google, Facebook, Yahoo, Microsoft, Verizon e Deutsche Telekom criaram a *Open Networking Foundation* (ONF), com o objetivo principal de adotar o SDN como padrão aberto para novos desenvolvimentos na área de *networking* [31].

Apesar de amplamente usadas no mercado, as redes IP tradicionais são complexas e de difícil gerenciamento. Estas são integradas verticalmente, ou seja, são compostas de equipamentos distribuídos que executam tanto o plano de controle como o plano de dados, conforme exposto anteriormente. As redes SDN quebram

este paradigma, separando estes planos, permitindo assim a criação centralizada de módulos lógicos programáveis no plano de controle para modificar o comportamento do plano de dados - comutadores da rede - em respostas a falhas, sobrecargas ou qualquer modificação necessária [8] [32]. Trazendo a inteligência para o plano de controle, as redes SDN desoneram os equipamentos do plano de dados em termos de processamento, permitindo mais fidelidade a sua principal função que é a de encaminhar os pacotes.

O plano de controle organiza um conjunto de informações para criar as tabelas de encaminhamento, baseando-se nas informações topológicas. Com a mesma ideia das redes TCP/IP, as redes SDN armazenam a informação topológica em uma base de dados chamada *Routing Information Base* (RIB), que se hospeda neste plano de controle. Já as tabelas de encaminhamento criadas a partir da RIB são chamadas de *Forward Information Base* (FIB). A FIB é criada uma vez que a RIB esteja consistente, sendo uma tabela que o plano de controle e o plano de dados mantém sincronizados [8], ou seja, cada comutador SDN possui sua própria FIB e uma cópia desta é encontrada no controlador. Portanto o plano de controle possui uma cópia de cada FIB de todos os planos de dados que controla. O plano de dados assegura que os pacotes sejam recebidos corretamente por uma interface e encaminhados por outra, baseado em uma regra da tabela FIB que foi previamente enviada pelo plano de controle.

3.3 OpenFlow

Uma das principais implementações das SDNs é o OpenFlow [33] [34] [35]. Este é um protocolo aberto idealizado pela Universidade de Stanford visando padronizar a forma que controladores se comunicam com os comutadores em uma arquitetura SDN, e um de seus principais objetivos iniciais foi permitir que equipamentos comerciais fossem utilizados em pesquisas com as redes SDN, haja visto que computadores usados nas universidades possuem menor processamento que

os equipamentos de mercado [36] [33] [34] [8], e também que testes com a SDN pudessem ser realizados em redes reais.

A Figura 3.1 resume os planos de controle e de dados. Em termos práticos, o plano de dados é realizado pelos equipamentos controlados, os comutadores, que obedecem ao plano de controle que é executado em um servidor chamado Controlador SDN.

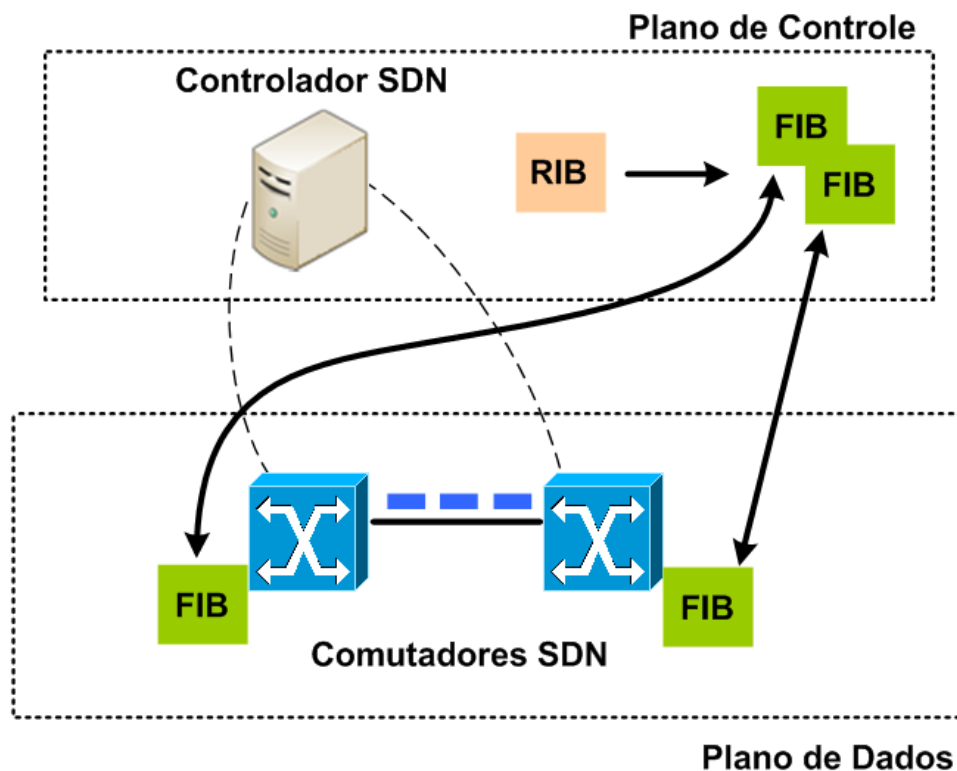


Figura 3.1: Conceitos básicos do SDN implementado em OpenFlow.

O conceito de programabilidade é um dos pontos principais das redes SDN [35] [37] [8]. Este conceito não é novo, haja visto que o gerenciamento de redes existe desde a criação das redes tradicionais. O ponto que difere a programabilidade das redes SDN é que os equipamentos, reais ou virtualizados, não somente são programáveis, mas interagem com outros tipos de softwares, criando um canal bidirecional de comunicação entre a rede SDN e o mundo exterior. Com isso pode

ser elaborado um módulo para interação da rede SDN com elementos de voz, por exemplo (Figura 3.2).

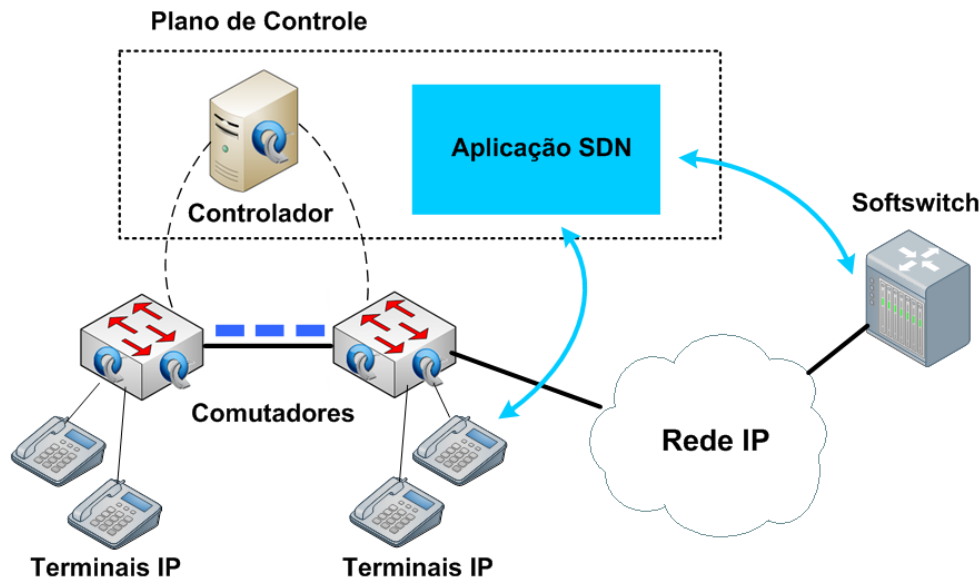


Figura 3.2: Atuação do plano de controle.

Dentro do modelo OpenFlow, com a migração do plano de controle para equipamentos exclusivos conforme rege a arquitetura SDN, fica sob responsabilidade destes equipamentos a interpretação de fluxos e regras associadas aos comutadores administrados por aquele plano. Com isso, detendo a inteligência da rede, tais equipamentos controladores permitem também a programação de novas entradas da tabela FIB baseando-se em algoritmos que entendam a topologia, e qualquer alteração que nesta ocorra, usando a RIB como fonte de informações topológicas. Normalmente, o controlador é representado por um módulo de software implementado de forma independente em algum ponto da rede.

Já o plano de dados trata do encaminhamento dos pacotes baseando-se em regras simples, chamadas de “ações”, associadas a cada entrada na tabela FIB. Os comutadores OpenFlow são equipamentos “burros”, encaminhadores de pacotes. É papel do controlador informar ao comutador o que fazer com determinado pacote. Como dito anteriormente, isto é feito através dos fluxos que a tabela FIB possui.

Cada fluxo desta tabela permitirá que o comutador execute uma ação ao pacote, tais como:

- Encaminhamento de pacote por porta específica;
- Alteração de cabeçalho do pacote;
- Descarte de pacote;
- Encaminhamento do pacote para inspeção do controlador.

Em uma rede OpenFlow os fluxos são definidos com uma entrada na tabela FIB. Cada entrada na tabela FIB possui três informações: (i) um cabeçalho que define o fluxo (regra) com base em campos das camadas de enlace, rede e transporte, (ii) estatísticas e (iii) uma ação (Figura 3.3). Estas estatísticas mantêm informações como número de pacotes que passaram por aquela regra ou o tempo desde a última vez que o fluxo foi usado, para que seja possível a sua exclusão por *timeout*.

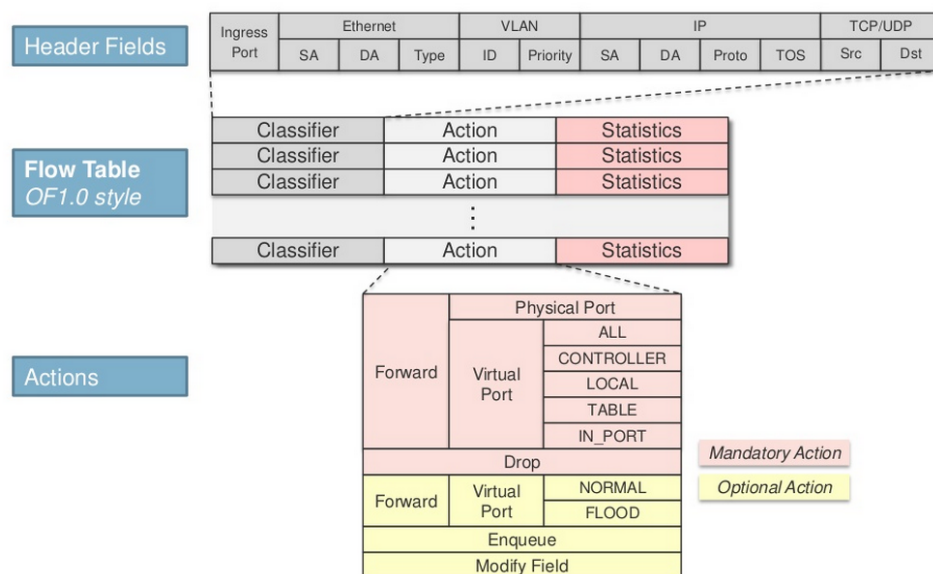


Figura 3.3: Entradas da tabela de fluxos [1].

Um simples exemplo do funcionamento dos fluxos está representado na Figura 3.4. Para o exemplo, clientes localizados na rede local desejam acessar duas aplicações nos servidores A ou B. Essas aplicações diferem-se pelo IP do servidor, que é dinâmico, e pela porta TCP que cada aplicação usa de forma fixa. Para que o comutador entregue os pacotes nos dois sentidos de forma correta, somente 3 fluxos seriam necessários na tabela FIB deste comutador, conforme exposto na Tabela 3.1.

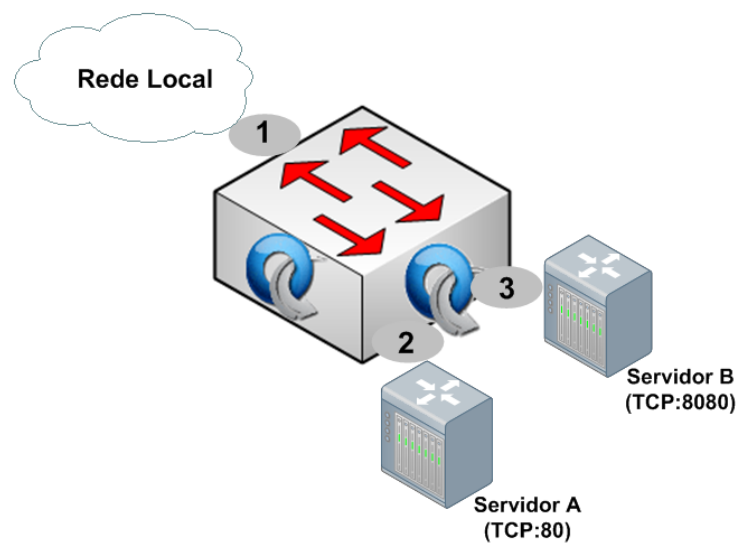


Figura 3.4: Simples exemplo de fluxos OpenFlow.

Regra	Ação
SW_Port_In = 1 e Prot_Dest = "TCP:80"	SW_Port_Out = 2
SW_Port_In = 1 e Prot_Dest = "TCP:8080"	SW_Port_Out = 3
SW_Port_In = 2 ou SW_Port_In = 3	SW_Port_Out = 1

Tabela 3.1: Tabela de fluxos do comutador da Figura 3.4.

3.4 Implementações SDN

Mininet Mininet [38] é um aplicativo emulador de rede que simula todo o ambiente OpenFlow, desde os *hosts*, comutadores, controlador e enlaces entre estes

elementos. Este aplicativo roda em ambiente Linux. Essa ferramenta é muito importante para a comunidade do OpenFlow, visto que é muito usada como forma de simular, testar e verificar o comportamento de novos cenários de rede SDN baseadas em OpenFlow.

Esta ferramenta permite ao desenvolvedor experimentar diversas topologias customizadas, bem realísticas e muitas vezes bastante complexas. O Mininet age exatamente como máquinas reais: o controlador OpenFlow nativo do Mininet roda o mesmo código comparado a controladores comerciais. A última versão do Mininet (v2.2.1) é baseada na mais recente versão do OpenFlow (v1.3) [38]. Os comutadores também trocam mensagens OpenFlow reais e são chamados de “*Open vSwitch*” [39]. Por isso, outra excelente característica do Mininet é a possibilidade de interação com as redes reais, ou seja, a máquina que está executando o Mininet permite que a aplicação use a sua interface de rede real para que os pacotes do Mininet trafeguem pela “rede viva” - Figura 3.5.

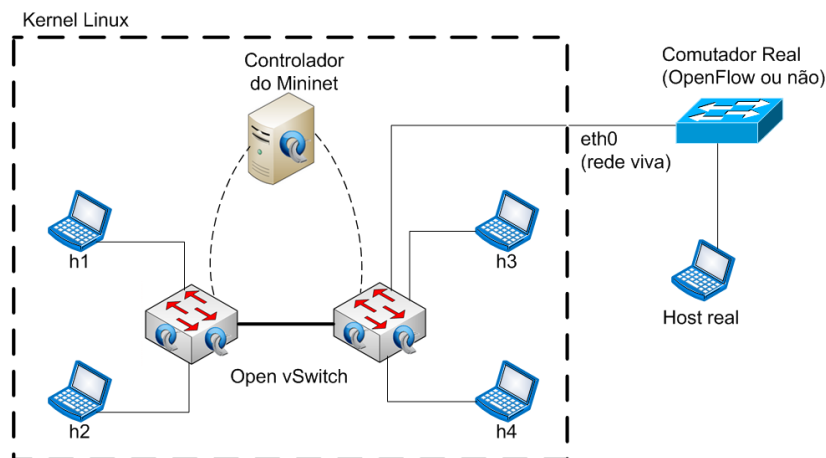


Figura 3.5: Ambiente Mininet interagindo com “rede viva”.

NOX/POX Uma das formas mais práticas de se implementar um controlador OpenFlow em ambientes reais é usando a distribuição NOX ou POX. O primeiro foi o precursor da implementação de um controlador OpenFlow, desenvolvido pela

empresa Nicira Networks [40], que doou este desenvolvimento para a comunidade de pesquisa, que agora vem sendo mantido e atualizado pela NOXRepo.org [41]. O NOX original é baseado em C++ e Python, com uma *Application Programming Interface* (API) para o OpenFlow v1.0, e é o controlador usado frequentemente em estudos acadêmicos que visam o desenvolvimento de aplicações SDN.

POX é o “primo” mais novo do NOX, sendo uma versão puramente em Python de seu precursor (*POX = NOX in Python* [8]). O POX traz algumas vantagens sobre NOX original, dentre as quais destacam-se a compatibilidade com Linux, Mac OS e Windows, e melhor desempenho das aplicações escritas sobre o POX em Python comparativamente às escritas no NOX híbrido¹ (C++/Python).

¹Já foi lançada uma nova versão do NOX puramente em C++ que possui melhor desempenho que o POX.

Capítulo 4

Trabalhos relacionados

Estudos já mostram que há a necessidade de uma “inteligência artificial” para que os recursos escassos de banda não atrapalhem a qualidade das chamadas que utilizam infraestrutura IP.

4.1 *Dynamic Codec with Priority for Voice over IP in WLAN*

Em [9], Stoeckigt et al estudaram aplicar uma adaptação dinâmica de codecs para usuários VoIP em redes sem fio. Neste trabalho, os clientes wi-fi (*VoIP users*) monitoram a qualidade de sua própria chamada e alteram o codec quando acham apropriado. Somente esta ação levaria estes usuários a terem uma percepção muito ruim de sua qualidade de voz naquela chamada, pois o codec de menor taxa tem menor robustez a qualquer deterioração de rede. Porém, paralelamente a isto, o AP (*access point*) ao qual aquele cliente está conectado usa o artifício do padrão IEEE 802.11e para melhor classificar o tráfego de voz do cliente que usa um codec de qualidade pior (e menor consumo de banda), ou seja, os pacotes daquele cliente passarão a ter mais prioridade do que aqueles que não sofreram alteração. O aspecto positivo deste trabalho é o fato de não haver nenhum equipamento novo na solução, dependendo apenas de códigos que rodam exclusivamente nos clientes

VoIP.

4.2 Nunes Costa VoIP (NCVoIP)

Em [2] os autores criaram o NCVoIP, uma aplicação que utiliza um fluxograma de decisões para analisar e modificar parâmetros de rede em uma chamada VoIP ativa. Esta aplicação utiliza 5 componentes para as funções necessárias do NCVoIP. São eles:

- GuiManager: interface entre o NCVoIP e o usuário (cliente VoIP);
- SipManager: implementa e gerencia o estado SIP, criando todas as mensagens SIP messages que permitem a estabilização, atualização e encerramento das chamadas;
- MediaManager: cria, transmite e recebe os pacotes RTP e RTCP. Também captura e executa os dados de multimídia.
- EvalQoS: interage com o MediaManager para poder periodicamente calcular o MOS através do modelo E do ITU-T;
- DoQoS: este componente implementa o algoritmo NCVoIP, avaliando quando a troca do codec ocorrerá. Ele recebe informações do EvalQoS e interage com o SipManager para realizar a alteração.

A figura 4.1 apresenta a arquitetura do modelo proposto por Costa e Nunes.

Dois fatores dificultam a implementação do modelo NCVoIP em cenário corporativo: um é o número de componentes, visto que são necessários vários nós, dificultando a operação do serviço. O segundo é que o componente *MediaManager* concentra o tráfego RTP, o que é um dificultador para o crescimento em escala do serviço.

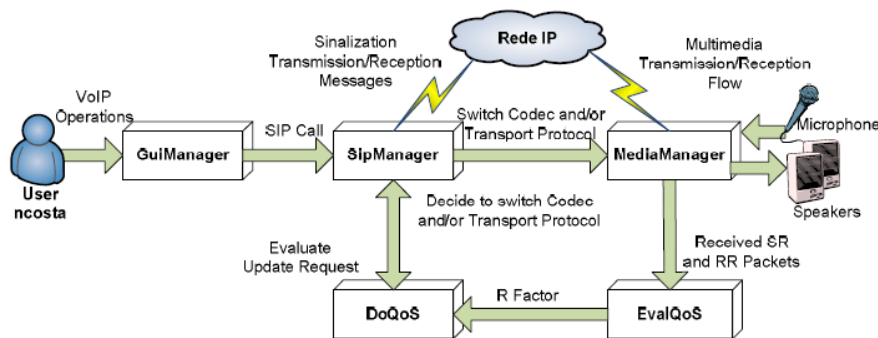


Figura 4.1: Arquitetura do NCVoIP [2].

4.3 *QoE-based Adaptive mVoIP Service Architecture in SDN Networks*

Kwon et al [10] também utilizam novos hardwares em uma rede sem fios definida por software para adaptação de codec dos terminais *wireless*: um agente de medição de QoS analisa valores de banda, *jitter* e atraso das interfaces aéreas dos APs e os utiliza para medição do MOS. Além disso, eles propõem o uso de uma gerência de QoS para VoIP, que analisa as condições da rede baseada nas informações dos agentes de medição e solicitam a adaptação de parâmetros de voz para os clientes VoIP.

4.4 *Improving network management with software defined networking*

Em [11], os autores identificaram o SDN como uma nova possibilidade de introduzir mecanismos para melhorar vários aspectos de gerência de rede. Eles focaram em três problemas: permitir frequentes alterações dadas as mudanças nas condições de rede, prover suporte para a configuração de rede em linguagem alto nível, e prover melhor visibilidade e controle de tarefas para executar diagnósticos de problemas em redes.

Sob o ponto de vista operacional, este é um excelente argumento para que o controlador SDN possa, através de sua programabilidade, adicionar recursos de gerenciamento de rede sem adição de novos componentes.

4.5 *Improving QoS in real-time internet applications*

Gorlatch et al [42] criaram uma API SDN que lida com ambientes *Real-Time Online Interactive Applications* (ROIA) - Aplicações Online em Tempo Real e Interativas. ROIA são aplicações de rede que conectam um grande número de usuários que, além de interagir com as próprias aplicações ROIA, também interagem com outros usuários em tempo real. A ideia de Gorlatch et al foi utilizar as capacidades de programação do controlador SDN e sua visão global da topologia para modificar em tempo real parâmetros de QoS de acordo com o comportamento e alterações da rede.

4.6 Comparativos

[2] e [10] são trabalhos que apresentam uma boa vantagem, que é o uso de um elemento específico para o cálculo do MOS, que pode trazer mais qualidade na adaptação do codec. Porém, incluindo mais elementos na rede proposta, eles aumentam consideravelmente o custo de implantação da solução.

Utilizando como base a ideia de central de Stoeckigt et al [9], esta dissertação aborda o tema de adaptação dinâmica de codecs em ambientes SDN, também usando como base as ideias de Hyojoon et al [11] e Gorlatch et al [42].

Porém, diferentemente de [2] e [10], essa dissertação aborda um modo mais simples de implementação da adaptação de codec. O CAoS é focado nos requisitos de uma rede corporativa, tratando de enlaces WAN. Usando as características do controlador SDN existente para lidar com variações de rede como Hyojoon et

al [11] e Gorlatch et al [42] fizeram.

Com isso, a partir de ideias encontradas nestes trabalhos relacionados, a dissertação apresenta o CAoS que executa uma rápida resposta a variações de rede, monitorando e enviando solicitações de alteração de codec durante uma chamada de voz ativa. Assim, a solução proposta alia eficiência e baixo custo de implementação. Além disso, CAoS pode ser usado em paralelo às implementações de QoS e CAC. Com isso a proposta apresentada permite que mais chamadas sejam realizadas usando os mesmos recursos de banda, sem a queda de chamadas ativas nem o bloqueio de novas chamadas.

Capítulo 5

Adaptação de codec em redes definidas por software - CAoS

Como visto, a implementação de QoS e CAC ajudará na redução da degradação da voz, mas este último artifício proporcionará uma situação inconveniente: como o CAC bloqueia chamadas quando a fila de voz chega ao seu limite, esta técnica possivelmente pode bloquear uma chamada importante.

Aumentar o número máximo de chamadas simultâneas em HMM significaria diminuir a qualidade das chamadas para que estas ocupassem menor banda total. Porém, fora da HMM, o uso de um codec de alta banda traz qualidade às conversações.

Considerando esta dicotomia, esta dissertação apresenta um trabalho onde é vislumbrada a alternância destes dois cenários, introduzindo na arquitetura SDN a capacidade de ser verificado o momento de congestionamento e então a atenuação da carga na rede através da modificação do codec em uso pelas chamadas de voz ativas. A escolha do ambiente SDN é devido a possibilidade de monitorar e reconfigurar a rede de forma simples e automática, simplificando a gerência da rede. Usa-se então um controlador OpenFlow para controlar a rede e também controlar as chamadas ativas, disparando as requisições de troca de codec de acordo com o cenário observado pelo próprio controlador.

Equipamentos específicos de monitoração de tráfego são elementos muito caros [43] [44], assim como aqueles que monitoram redes VoIP especificamente. Utilizando um elemento já existente em redes SDN, o controlador OpenFlow, com toda sua capacidade de programabilidade, elimina-se a necessidade de aquisição de elemento específico para desempenhar tais funções. Tais vantagens não poderiam ser aproveitadas em redes não SDN, visto que as funções de monitoramento e controle devem ser executadas em equipamentos específicos. Por isso opta-se pela utilização das redes SDN.

5.1 CAoS - *Codec Adaptation over SDN*

A proposta desta dissertação é usar a SDN como base para solucionar o problema descrito durante o congestionamento de um enlace, atuando no tráfego durante a HMM. Assim, é proposto um módulo executado na controladora SDN, nomeado de CAoS, que detecta quando o limite de tráfego da fila de voz atinge seu limiar, e então envia um pedido de mudança de codec para o elemento de controle de telefonia.

A detecção do limiar de tráfego de determinada interface de um equipamento não é algo complicado de se obter, e qualquer software de monitoração consegue ler este valor de taxa de um enlace através de mensagens de monitoramento divulgadas pelo próprio elemento monitorado, assim como ocorre em redes SDN. Porém fazer desta leitura um artifício para ajustar a rede é algo que foi pensado de forma desassociada por outros autores.

Da mesma forma, a adaptação da taxa de uma chamada VoIP também é algo factível, porém não implementado pelas companhias. Outros estudos [9] [2] [10] provam que tais adaptações são plenamente possíveis, porém utilizam elementos distintos para realizar o monitoramento e adaptação do codec de voz das chamadas. A principal inovação deste trabalho está em aproveitar o ambiente SDN para trazer ambas funcionalidades, monitoramento e controle, para o controlador OpenFlow

existente.

A interação do controlador com a rede de telefonia IP pode ocorrer de duas formas: através de uma API com o *softswitch* ou diretamente com o cliente VoIP (Figura 5.1). Tais interfaces programáveis estão disponíveis nos *softswitches* de mercado, porém a um alto custo, pois há o licenciamento de tais aplicações. Por outro lado, há no mercado diversos clientes VoIP de código aberto, sendo então possível a elaboração de tal API a um custo infinitamente menor, se resumindo em horas de trabalho ou, no pior dos casos, horas de consultoria. Por isso o trabalho seguiu na linha de desenvolvimento da interface programável no cliente VoIP, visando mais uma vez a desoneração financeira.

Com isso, detectado o momento que a adaptação é necessária, o controlador interage com os terminais VoIP, que possuem tal API desenvolvida especificamente neste trabalho, e então executam a adaptação do codec durante o curso da chamada utilizando mensagem de *UPDATE* do protocolo SIP (sem a desconexão/reconexão da chamada), aliviando o tráfego e permitindo que mais chamadas possam ser realizadas com a mesma banda total do enlace. A Figura 5.1 resume a solução apresentada.

A arquitetura do CAoS está subdividida em 3 sub-módulos:

- Módulo de Monitoração;
- Módulo de Decisão;
- Módulo SIP.

5.1.1 Módulo de monitoração

O módulo de monitoração do CAoS é executado no controlador SDN e utiliza 2 grupos de estatísticas vindas dos comutadores OpenFlow: *flowstats* e *ports-tats*. Esses grupos de estatísticas são solicitados pelo controlador aos comutadores através de mensagens definidas no protocolo OpenFlow.

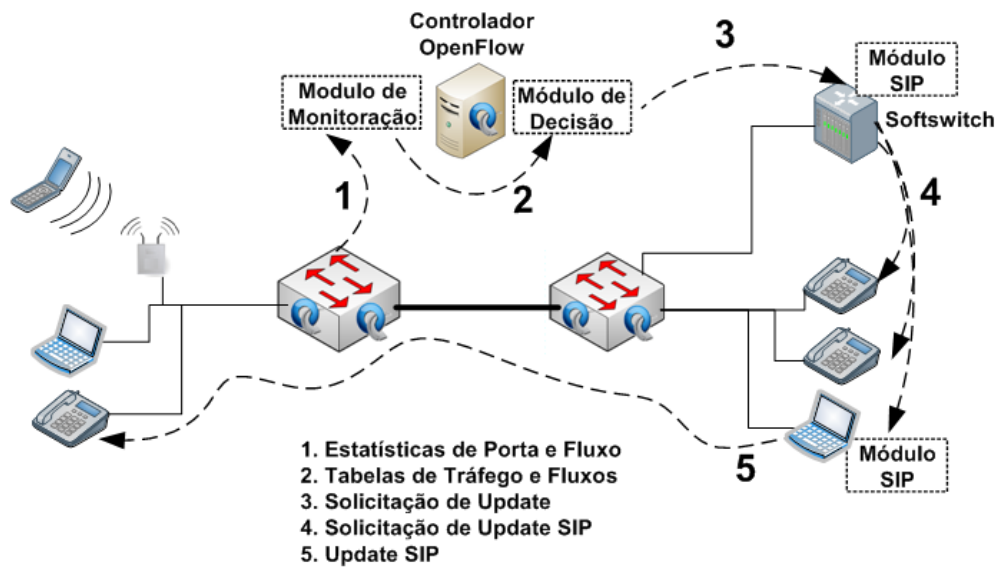


Figura 5.1: Fases do CAoS.

Inicialmente, ao ser executado, o CAoS invoca uma função de “Timer” do controlador. Esta função é a responsável por chamar as funções de solicitação de estatísticas em períodos recorrentes. É denominado de *Polling* (P) o intervalo de tempo que o controlador recebe as estatísticas. Caso este tempo seja curto, o módulo medirá o tráfego mais precisamente, permitindo o mecanismo ter uma ótima resposta a variações de rede. Entretanto, mais mensagens de controle sobrecarregarão a rede, além de maior processamento por parte do controlador. Por outro lado, caso este período seja mais longo, o módulo terá uma resposta mais lenta, porém a sobrecarga de mensagens de controle será baixa.

Cada vez que a monitoração é ativada e dados são requisitados aos comutadores da rede, as informações obtidas das mensagens de *FlowStats* e *PortStats* são:

1. *FlowStats*

- **sw_id**: identificação do comutador.
- **sw_in_port**: número da porta de entrada pela qual os pacotes de determinado fluxo chegam ao comutador.

- `packet_count`: contagem de pacotes do fluxo.
- `idle_timeout`: tempo que determinado fluxo está ocioso.
- `byte_count`: contagem de bytes do fluxo.
- `duration_sec`: tempo de atividade do fluxo.
- `protocol_type`: tipo de protocolo do fluxo.
- `dst_protocol_port`: porta de destino relativa ao fluxo.
- `src_ip_addr`: endereço IP de origem.
- `dst_ip_addr`: endereço IP de destino.
- `src_mac_addr`: endereço físico de origem.
- `dst_mac_addr`: endereço físico de destino.
- `tos`: campo de tipo de serviço.

2. *PortStats*

- `port_no`: número da porta de entrada pela qual os pacotes de determinado fluxo chegam ao comutador.
- `rx_packets`: quantidade de pacotes recebidos.
- `tx_packets`: quantidade de pacotes enviados.
- `rx_bytes`: quantidade de bytes recebidos.
- `tx_bytes`: quantidade de bytes enviados.
- `rx_dropped`: quantidade de pacotes recebidos descartados.
- `tx_dropped`: quantidade de pacotes enviados descartados.
- `rx_errors`: quantidade de pacotes recebidos com erro.
- `tx_errors`: quantidade de pacotes enviados com erro.

Partindo do princípio que o trabalho está focado no tratamento das chamadas de voz, o módulo de monitoração necessita saber a taxa efetiva dos somatório

de fluxos de voz que percorrem o comutador em cada momento. Como em cada comutador trafegam vários tipos de informações, pacotes de diferentes serviços, precisa-se diferenciar os pacotes de voz. Para isso, o módulo de monitoração utiliza a primeira parte dessas estatísticas (***FlowStats***), com as quais consegue discernir os fluxos de voz que trafegam na porta monitorada do comutador, filtrando pacotes por:

- (i) sua origem, ou seja, identificando o comutador monitorado através do campo ***sw_id*** da mensagem OpenFlow relativa aquele pacote;
- (ii) pelo tipo de protocolo camada 4 do pacote, ou seja, identificando o campo ***protocol_type*** como sendo UDP ou TCP;
- (iii) pela porta para a qual o pacote foi encaminhado. O campo ***dst_protocol_port*** identificará a porta de destino UDP ou TCP que aquele pacote foi direcionado;

Parametrizando o filtro e buscando pacotes **UDP** destinados a portas definidas para o tráfego de voz, entre 16.384 e 32.767¹, o módulo de monitoração somente tratará de pacotes RTP. Ainda assim poderia ser otimizada esta filtragem, fazendo com que o comutador pudesse ler o campo DSCP dos pacotes, uma vez que a marcação deste campo é feita na origem, ou seja, no próprio cliente VoIP. Assim, o comutador poderia criar uma fila exclusiva de voz e fazer o monitoramento baseando-se somente neste campo DSCP. Para aplicações de voz, este campo do cabeçalho IPv4 recebe o valor *Expedited Forwarding* (EF) [45].

O módulo de monitoração mede o tráfego usando as informações os pacotes OpenFlow ***PortStats*** - ***rx_bytes*** e ***tx_bytes*** - relativos àquela porta do comutador em questão – porta WAN. Essas informações trazem a quantidade total de bytes entrando e saindo naquela porta, portanto, uma vez que duas medições são

¹Por definição, (RFC 3550 [16]), estas portas não são registradas para o RTP, porém esta numeração é mais usada nas aplicações comerciais.

realizadas durante um intervalo “P”, pode-se assim aferir a taxa dados que aquela interface está trafegando nos dois sentidos.

$$\begin{aligned} B_{\text{in}} &= \frac{(rx_bytes_1 - rx_bytes_0) * 8 / 1000}{P} [kbps] \\ B_{\text{out}} &= \frac{(tx_bytes_1 - tx_bytes_0) * 8 / 1000}{P} [kbps] \\ P &= P_1 - P_0 [s] \end{aligned}$$

- rx_bytes_0 e tx_bytes_0 são medidos no tempo P_0 ;
- rx_bytes_1 e tx_bytes_1 são medidos no tempo P_1 ;

5.1.2 Módulo de decisão

Ao receber as informações fluxos RTP do módulo de monitoração, *FlowStats*, o módulo de decisão cria uma tabela interna (**tabela de pares**) com todos os pares de conversações que estão em curso. Essa tabela é um reflexo (enxuto) da tabela de fluxo do comutador que está sendo monitorado, com um atraso causado pelo intervalo de tempo entre as estatísticas. Uma vez que dois nós iniciam uma conversação, o módulo de decisão, ao perceber um *stream* RTP bidirecional e síncrono, adiciona uma entrada na tabela de pares, informando os endereços dos pares e o tipo de codec² que está sendo usado.

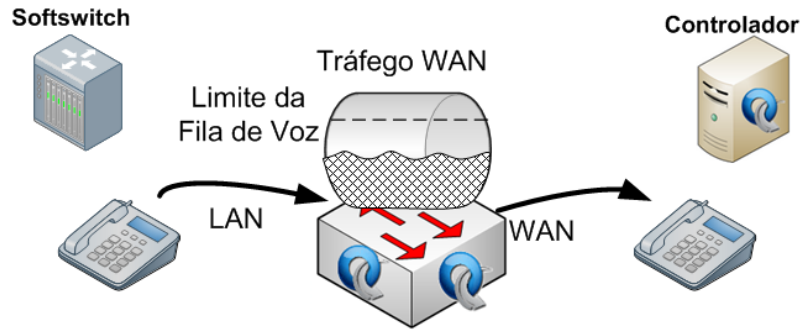
Paralelamente o Módulo de Decisão também recebe a cada “P” segundos as taxas da interface monitorada calculadas pelo Módulo de Monitoração. Com a informação deste tráfego, o módulo de decisão irá solicitar a atualização de codec quando um limiar predeterminado é atingido. Este limiar é determinado de acordo com o valor máximo de vazão do enlace que está conectado à porta monitorada.

Definindo um limite superior (α) de banda, quando o módulo constata que esse valor foi alcançado, ele verifica na tabela de pares e confirma se há alguma

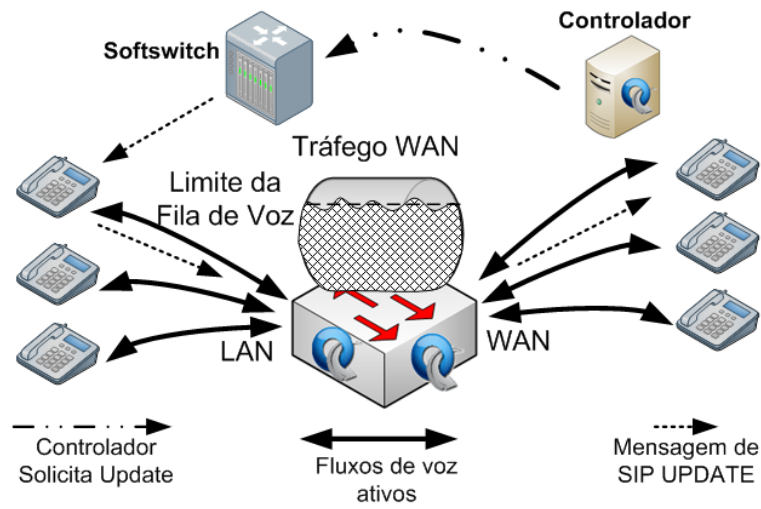
²O módulo de decisão desenvolvido não lê o pacote de invite SIP nem o pacote RTP, por isso não é possível identificar qual o codec em uso. Assume-se então que há um codec padrão para novas chamadas, pois se trata de ambiente corporativo.

conversação ativa usando um codec de alto consumo de banda. Encontrando a entrada, o módulo solicita ao elemento de voz a alteração para um codec de baixo consumo, o qual é um codec conhecido por se tratar de uma rede corporativa, portanto suportado por todos os terminais.

Por outro lado, definindo um limite inferior (β) de banda, o módulo também realiza o caminho inverso: quando este constata que o valor inferior foi atingido, ele confirma na tabela de pares se há alguma conversação ativa usando um codec de baixo consumo de banda e, encontrando, solicita a alteração para o codec de melhor qualidade de voz e alto consumo de banda. Este caminho inverso é realizado para que o melhor cenário quanto a qualidade da fala seja usado. A figura 5.2 demonstra o processo.



(a) Condição normal: sem sobrecarga na fila.



(b) Fila sobrecarregada: controlador OpenFlow interage requisitando mudança de codec.

Figura 5.2: Alteração de Codec

O módulo de decisão funciona sob demanda, ou seja, toda vez que o limite superior (α) ou o limite inferior (β) são alcançados, a função de troca de codec é invocada para uma conversação por vez, como indicado no Algoritmo 1.

A entrada do algoritmo de seleção é a ocupação do enlace (L_{occ}); N é o tamanho da tabela de pares (L_p). Quando L_{occ} atinge α , o limite superior de consumo de banda, o algoritmo inicia uma busca de baixo para cima na tabela de pares por uma conversação usando um codec diferente de *LowCodec*. Este tipo de busca

Algoritmo 1: Algoritmo de seleção do Módulo de Decisão.

Entrada: $L_{occ}, Lp, \alpha, \beta, LowCodec, HighCodec$

```

1  $N \leftarrow size(Lp)$ 
2  $Flag_{codec} \leftarrow 0$ 
3 if ( $L_{occ} \geq \alpha$ ) then
4   while  $Flag_{codec} = 0$  do
5     if ( $N \geq 1$ ) then
6       if ( $Lp[N]_{StreamCodec} \neq LowCodec$ ) then
7          $call : ChangeToLowCodec$ 
8          $Flag_{codec} \leftarrow 1$ 
9       else
10         $N \leftarrow N - 1$ 
11      end
12    else
13       $Flag_{codec} \leftarrow 1$ 
14    end
15  end
16 end
17 if ( $L_{occ} < \beta$ ) then
18   while  $Flag_{codec} = 0$  do
19     if ( $N \geq 1$ ) then
20       if ( $Lp[N]_{StreamCodec} \neq HighCodec$ ) then
21          $call : ChangeToHighCodec$ 
22          $Flag_{codec} \leftarrow 1$ 
23       else
24         $N \leftarrow N - 1$ 
25      end
26    else
27       $Flag_{codec} \leftarrow 1$ 
28    end
29  end
30 end

```

permite que o código altere primeiramente o codec da última chamada iniciada, favorecendo as chamadas mais antigas. Ao encontrar a entrada na tabela que necessite da alteração, o procedimento *ChangeToLowCodec* é iniciado. Porém se (L_{occ}) atinge β , limite inferior de consumo de banda, o algoritmo também inicia

uma busca por uma conversação usando um codec diferente de *HighCodec*, e envia então uma solicitação para o uso do codec de alto consumo novamente. Denifir o mesmo valor para α e β pode acarretar em uma instabilidade quando L_{occ} varia frequentemente ao redor do valor escolhido.

Ambos *ChangeToHighCodec* e *ChangeToLowCodec* são procedimentos que enviam mensagens em texto puro usando um *socket* UDP ou TCP direcionadas a um elemento de voz. Essas mensagens são enviadas através da simples abertura de uma porta e envio de uma mensagem de texto para um destino determinado: ou o *softswitch* ou um cliente VoIP. Os elementos de destino recebem a mensagem, a qual será usada pelo módulo SIP.

Como forma de manter a **tabela de pares** sempre atualizada, é ainda responsabilidade do módulo de decisão excluir qualquer entrada nesta tabela de fluxos que já não existe mais, verificando em cada “*polling*” se as entradas da tabela equiparam-se aos fluxos RTP.

Como pode ser observado, o algoritmo do CAoS é muito simples e eficaz, por isso é esperado que seu consumo de processamento no controlador SDN seja baixo e sua convergência eficaz. Entende-se por convergência a rapidez com que a rede responde a uma requisição de alteração por parte do algoritmo. Como explicado, a principal entrada do CAoS é a leitura da taxa efetiva do enlace monitorado (L_{occ}), e o que vai ditar o quão rápido o algoritmo vai interferir no ajuste do codec das chamadas é o valor predefinido de P (*polling*) e de quantas chamadas necessitam ser alteradas para o ajuste da rede. O módulo CAoS faz o ajuste sob demanda: por exemplo, se o algoritmo fez a leitura da taxa do enlace e verificou que há a necessidade de alterar o codec para 1 chamada, a requisição é enviada e outra leitura do enlace é feita, então outra requisição é enviada pelo CAoS se necessário. Por isso, o tempo de convergência vai depender da própria dinâmica de chamadas que estão ocorrendo na rede. É também esperado que quanto menor o valor de P, ou seja, leituras mais frequentes, mais eficaz é a convergência e maior é o processamento do controlador.

5.1.3 Módulo SIP

Como exposto anteriormente, o SIP [21] é um protocolo de sinalização para aplicações de voz sobre IP que lida facilmente com atualização de codec, através da mensagens que podem ser trocadas por usuários (*user-agents*). Toda vez que uma chamada SIP inicia, a origem envia uma mensagem de “invite” (convite) contendo uma lista de prioridades de codecs aceitos por aquela origem. Uma vez que o destino recebe este “invite”, ele responde informando qual o codec escolhido, baseado em sua lista de codecs preferidos, e então a chamada é estabelecida.

Durante o curso de uma chamada, se um dos nós envia uma mensagem de “update” (atualização) [46] com uma nova lista de codecs, então o outro nó verifica a sua disponibilidade para mudança de codec dentro da nova lista, e a alteração do codec é feita durante a chamada, sem qualquer perda de pacote de voz.

O módulo SIP é a última etapa da arquitetura do CAoS, recebendo o input do módulo de decisão para disparar uma mensagem para o *softswitch* ou para o *user-agent* fazer a alteração do codec.

Este módulo pode tanto ser executado usando o protocolo TCP quanto UDP. Este não refere-se a mensagem SIP propriamente dita, mas sim a uma mensagem própria do CAoS, em texto puro. Ao ser invocado, o módulo SIP simplesmente abre um *socket* TCP ou UDP no controlador (*socket* modo cliente) e envia uma *string* para o equipamento de voz efetivamente realizar a alteração do codec. O equipamento de voz recebe esta mensagem através do mesmo *socket* TCP ou UDP, que deve ser aberto em modo servidor, na mesma porta que o modo cliente enviar.

Pelo fato de existir uma interação com software externo ao controlador, que é uma das vantagens das redes SDN, este módulo também deve co-existir no equipamento de voz tendo sua função limitada a abrir um *socket* em uma porta pré-definida e receber a mensagem de texto do controlador SDN via TCP ou UDP, e então efetivamente alterar o codec em uso, usando para isso as trocas de mensagens de “update”.

Capítulo 6

Implementação e resultados

6.1 Implementação

A análise de desempenho da proposta CAoS foi feita por meio de uma emulação, tendo por objetivo simular um cenário próximo aos existentes em redes corporativas, onde chamadas entre prédios (*inter-sites*) ocorrem atravessando um enlace WAN. A emulação demonstrará como o CAoS pode melhorar a qualidade de voz durante um congestionamento de enlace, sem restringir o número total de chamadas como o CAC faz.

O melhor jeito de se aferir a qualidade de uma chamada é medindo-se a qualidade da fala da mesma. Durante um congestionamento, os resultados obtidos com o CAoS mostram que o mesmo número de chamadas, e até mais, podem ser realizadas se o codec utilizado for adaptado para atender os recursos de banda mais restritos.

6.1.1 Medição

Durante um congestionamento de tráfego de voz, é esperado que os valores de perda de pacote, atraso e *jitter* aumentem, tornando o valor do MOS muito baixo. Isto é algo que pode facilmente ocorrer se o número de chamadas excede o limite da

fila de voz para determinado enlace. O trabalho proposto mostra que aliviando o tráfego através da redução do *payload* dos pacotes de voz em trânsito, é esperado que a perda de pacote, atraso e *jitter* diminuam, ou até mesmo não ocorram, melhorando o valor do MOS, ou seja, a qualidade da fala daquela chamada.

Para avaliar o MOS durante o experimento, uma aplicação chamada VoipMonitor [47] foi utilizada. Este software mede o MOS usando o modelo E do ITU-T, capturando o tráfego de voz em uma determinada interface.

Neste trabalho, o VoipMonitor recebe pacotes da interface virtual de uma instância do emulador que corresponde a interface WAN da topologia. Capturando pacotes RTP e RTCP, esta aplicação mede o *jitter*, atraso e perda de pacotes, e então aplica o modelo E do ITU G.107, apresentando o MOS ao final de cada chamada.

6.1.2 Cliente VoIP

O cliente VoIP usado no trabalho foi o PJSIP [48], que é um agente SIP escrito em *python*. PJSIP é um projeto de código aberto e permite a manipulação de mensagens SIP durante o curso de uma chamada. Cada cliente VoIP executa um script que faz parte do CAoS (“módulo SIP”). Esta rotina escuta uma porta UDP ou TCP pré-definida esperando a mensagem do controlador POX. Uma vez que o cliente SIP recebe esta mensagem durante uma chamada estabelecida, este, então, envia uma mensagem de atualização (*SIP update*) para seu par priorizando o codec definido pelo controlador OpenFlow. O controlador indica qual codec usar baseado no limiar que foi atingido pelo módulo de decisão: α ou β . Uma característica importante deste cliente SIP é que ele não necessita de um SIP proxy (*softswitch*) para o gerenciamento da chamada. Mensagens são trocadas diretamente entre clientes PJSIP.

6.1.3 Cenário de emulação

Nesta análise, propõe-se um cenário onde comutadores OpenFlow funcionam como elementos de borda, conectando 2 sítios adjacentes. O controlador OpenFlow para esta emulação é o POX v0.3.0 (dart), que usa versão 1.0 do OpenFlow, e é o responsável por controlar essa rede, analisando o tráfego WAN (“módulo de monitoração”). Portanto, o CAoS foi implementado como uma aplicação no POX. Conectados a cada comutador estão os nós onde rodam os clientes VoIP (Figura 6.1).

O Mininet v2.0.0 [38] é a ferramenta de emulação, e é executado em duas instâncias. A primeira roda em um hardware com Ubuntu v12.04LTS @ Intel Core 2 Duo 1.9GHz com 3GB de RAM, onde são virtualizados 1 comutador OpenFlow mais 10 nós (hosts) que estão conectados ao comutador. A segunda instância também é executado em hardware similar, com Ubuntu v12.04LTS @ Intel Core 2 Duo 2.1GHz com 4GB de RAM, onde também são executados 1 comutador mais 10 nós. Nesta segunda instância do Mininet também é executado o “medidor” de MOS (VoipMonitor).

Cada instância do Mininet no ambiente de teste proposto cria um comutador OpenFlow (*Open vSwitch*) que, além dos hosts conectados às suas portas, possui uma porta ethernet do hardware hospedeiro conectada a uma interface virtual do comutador. Com isso, as instâncias do Mininet estão conectadas como se fossem dois equipamentos adjacentes. Para emular uma conexão WAN, ambas as máquinas hospedeiras executam o *Traffic Control* (TC) do Linux. Com isso é possível controlar a taxa efetiva do enlace e seu atraso bidirecional.

A ferramenta de Controle de Tráfego no Linux [49] está disponível desde o *kernel* 2.2. Basicamente, as interfaces ethernet dos computadores são baseadas em FIFO (*first in first out*), ou seja todos os pacotes saem pela interface na mesma ordem que chegaram. O Controle de Tráfego do Linux possibilita que essa regra seja modificada, criando outros mecanismos para o enfileiramento e envio dos pacotes

pela interface de saída. Exemplo disto é o artifício usado neste trabalho, onde o TC limita a taxa da interface ethernet a um valor menor que o seu valor máximo, descartando os pacotes excedentes àquela taxa configurada. Além do limite de taxa, o TC também permite que os pacotes sofram um atraso predeterminado. Isto é feito desviando esses pacotes para um buffer onde o tempo de armazenamento no mesmo é constante, com isso, um atraso pode ser simulado entre dois hosts que estão lado a lado.

O POX utiliza um terceiro hardware exclusivo e similar aos dois anteriores (Ubuntu v12.04 LTS @ Intel Core 2 Duo 2.0GHz com 4GB de RAM). A opção de se ter equipamentos separados é para que a performance do POX e de cada instância do Mininet seja a melhor possível. Como o POX periodicamente recebe e analisa estatísticas vindas dos comutadores, o desempenho deste é crucial para o bom funcionamento do CAoS, além de facilitar a análise do consumo de CPU em um dos testes realizados.

O controlador POX executa o módulo básico de encaminhamento em camada 2 (“*L2 Learning Switch*”) para realizar a comutação dos quadros. Adicionalmente é executado o CAoS, que foi escrito em *python* e que escuta as mensagens de estatísticas do OpenFlow a cada P segundos, e as usa para medir o tráfego na interface WAN. Uma vez que este tráfego chega a um valor específico, o módulo inicia um cliente UDP ou TCP que envia uma mensagem contendo a palavra “SPEEX” ou “G711” diretamente para o cliente VoIP, definindo assim o codec que este e seu par devem usar. Para os cenários testados, nenhum *softswitch* foi considerado para simplificar a emulação, uma vez que o cliente PJSIP permite que as mensagens de UPDATE sejam enviadas diretamente entre os *endpoints VoIP*.

Durante a estabilização da chamada, a origem oferta o melhor codec como prioridade 1, o G.711 [50], que consome por volta de 85 kbps de banda. O segundo codec ofertado na emulação é o SPEEX [51], que consome aproximadamente 40 kbps, pois utilizou-se o SPEEX NB [51]. O codec SPEEX foi escolhido dada sua similaridade com o G.729 [52], que é um codec licenciado e amplamente usado

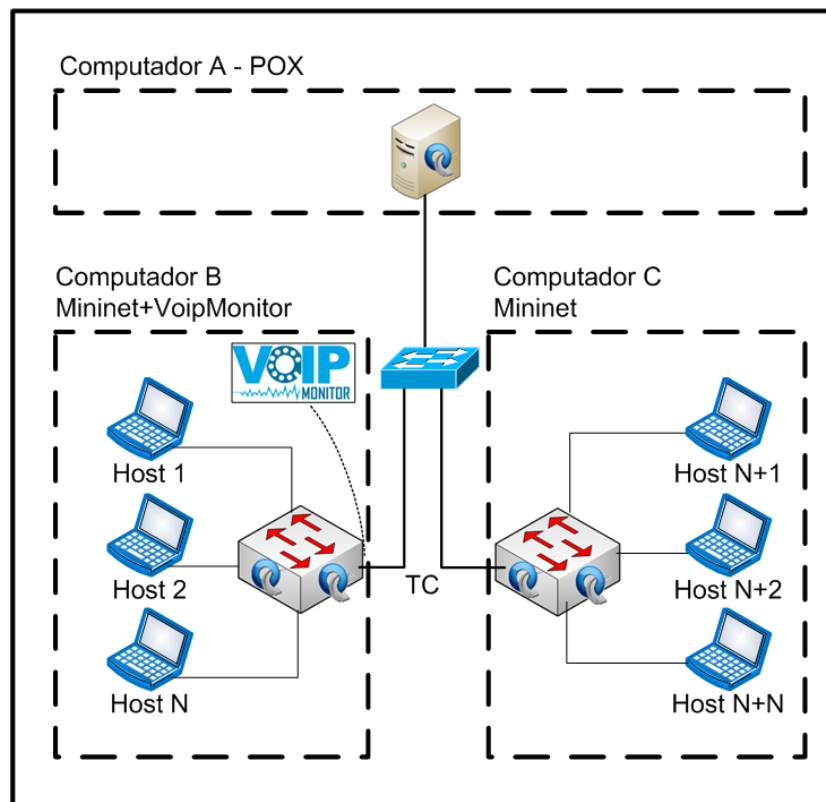


Figura 6.1: Ambiente de emulação - controlador POX e Mininet rodando em equipamentos distintos, e VoipMonitor capturando na interface WAN.

como codec de baixo consumo de banda no mercado.

Nos testes apresentados a seguir, todas as chamadas são iniciadas utilizando-se o codec G.711. Atingindo o valor de α , o módulo de decisão aciona o módulo SIP fazendo-o alterar o codec para SPEEX, e atingindo o valor de β , o módulo executa o caminho reverso, retornando o codec das chamadas para G.711.

6.1.4 Resultados

Quatro testes foram executados:

1. Comparativo do uso de codecs de alto e baixo consumo de banda;

2. Performance do CAoS para múltiplas chamadas;
3. Performance do CAoS variando-se o *polling* (P) de medições;
4. Performance do CAoS variando-se o protocolo de comunicação usado no módulo SIP.

Teste #1 Os parâmetros ajustados para o cenário do teste 1 estão apresentados na Tabela 6.1. Escolheu-se por fazer este teste como forma de validação básica da proposta, uma vez que o mesmo somente usa uma ou duas chamadas concorrentes, usando os diferentes codecs da emulação. O uso de 160 kbps de banda do enlace objetiva limitar o uso do codec G.711 para uma chamada apenas, portanto a segunda chamada em G.711 ultrapassa este valor do enlace. O valor de α indica que ao ser iniciada a segunda chamada em G.711, o codec será alterado caso o CAoS esteja sendo executado. O valor de β foi definido tendo em vista que se somente uma chamada estiver em curso usando o codec SPEEX (40 kbps), o módulo SIP será invocado para retornar a chamada para o G.711. Já o valor do atraso foi escolhido para emular um enlace de longa distância. O *polling* para este teste foi arbitrado.

Variável	Significado	Valor
B	Taxa do link	160 kbps
Δt	Latência unidirecional do enlace	5 ms
α	Limite superior de consumo de banda	140 kbps
β	Limite inferior de consumo de banda	60 kbps
P	<i>Polling</i>	2 s

Tabela 6.1: Parâmetros do Teste #1

Analizou-se o comportamento das chamadas em quatro cenários:

1. Para o primeiro cenário, somente uma chamada foi estabelecida em G.711, desde seu início até o término, sem intervenção do CAoS.

2. No segundo cenário, foi repetido o primeiro cenário, modificando o codec para SPEEX, ou seja, a chamada iniciou e terminou usando o codec de menor consumo de banda, também sem ação do CAoS.
3. No terceiro cenário, duas chamadas em G.711 foram iniciadas em sequência, e mais uma vez o módulo CAoS não foi executado.
4. Já no último cenário deste teste, também foram iniciadas duas chamadas em G.711 em sequência, porém desta vez com o módulo CAoS em execução. Com isso, ao início da segunda chamada em G.711, a taxa do enlace superou o valor de α , indicando que o enlace estava prestes a ficar congestionado, então o CAoS agiu requisitando a troca de codec da última chamada iniciada para o SPEEX, culminando no decréscimo da taxa do enlace para aproximadamente 125 kbps, valor inferior a α .

Para cada um dos quatro cenários foram analisadas a **perda de pacotes**, ***jitter*** e **MOS**. As medições foram realizadas em 20 rodadas (**n**), estabelecendo-se chamadas de 20 segundos cada. Os resultados na Figura 6.2 são representados pelos valores médios ($\bar{x}$). As barras de erro são representadas pelo erro padrão (SE), onde σ é o desvio padrão.

$$SE = \frac{\sigma}{\sqrt{n}}$$

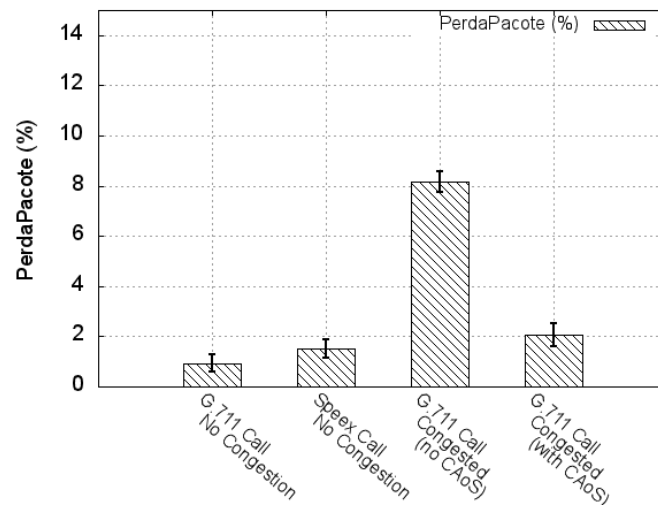
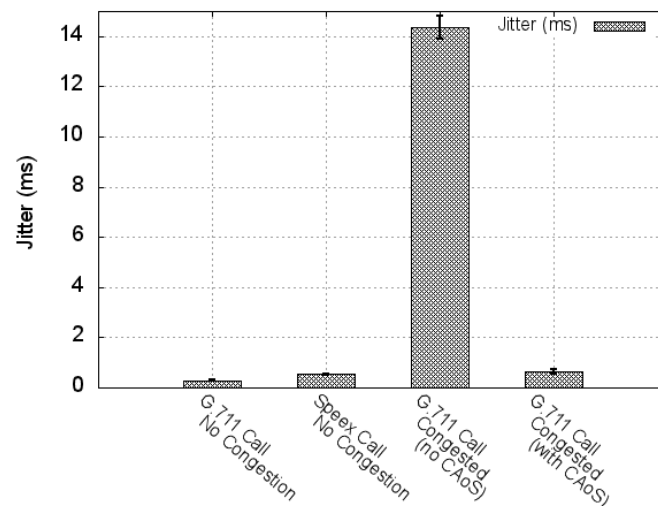
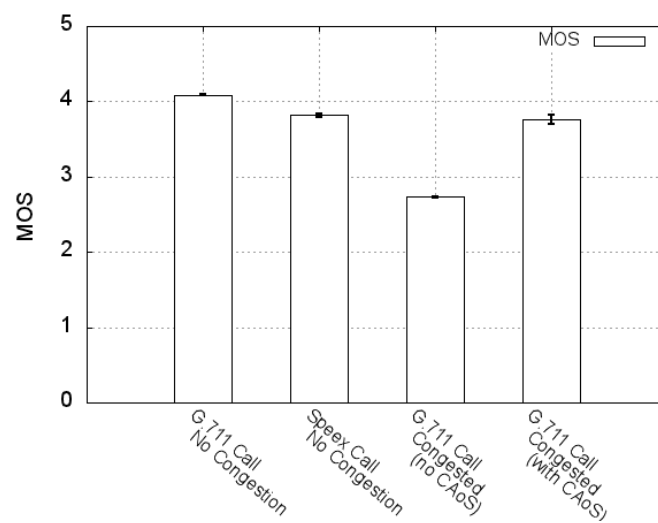
(a) Análise da *perda de pacotes* entre os 4 cenários.(b) Análise do *jitter* entre os 4 cenários.(c) Análise do *MOS* entre os 4 cenários.

Figura 6.2: Comparação de Codecs no Teste #1.

O primeiro teste mostrou que tanto usando o G.711 ou o SPEEX (primeiro e segundo cenários), há uma pequena diferença nos três parâmetros medidos quando não há necessidade do uso do CAoS. Para estes dois primeiros cenários, a diferença do valor de MOS (Figura 6.2(c)) é resultado da característica de cada codec. Como o MOS aferido pela aplicação VoipMonitor é baseado nos parâmetros de rede medidos (modelo E do ITU-T em conjunto com a RFC 3611 [27]), este software não apresenta o valor do MOS correto quando o SPEEX é usado, pois não há a opção de escolha do codec em uso e esta aplicação não faz a leitura do codec das chamadas, assumindo que o codec é o G.711. Por isso, uma correlação teve de ser feita de acordo com valores máximos do MOS encontrados em [53], assumindo que o MOS máximo para o SPEEX é o mesmo do G.729, dadas suas similaridades. Devido a isso, o valor do MOS é mais baixo, mesmo com os parâmetros de perda de pacotes e jitter sendo muito parecidos.

Os resultados do terceiro e quarto cenários, onde existe congestionamento, mostram uma diferença significativa nos valores de perda de pacotes, jitter e MOS. Isso confirma que o CAoS minimizou a deterioração da qualidade da chamada em um cenário que haveria congestionamento, atuando na troca do codec usado em uma das chamadas no momento que α é atingido, mantendo assim a qualidade equivalente a do uso do codec SPEEX em um cenário sem congestionamento.

Teste #2 Os parâmetros ajustados para o cenário do teste 2 estão apresentados na Tabela 6.2. No segundo teste modificou-se a banda do enlace WAN para 650 kbps tendo em vista que esse enlace deveria suportar algumas chamadas em G.711 e outras em SPEEX, não possibilitando que todas as chamadas fossem realizadas em G.711, que ocuparia 850 kbps de banda. Analisou-se então o comportamento de **perda de pacotes**, **jitter** e **MOS** em 2 cenários:

1. No primeiro cenário foram estabelecidas 10 chamadas simultâneas em G.711, sem que o CAoS estivesse em execução. Cada uma dessas chamadas inicia-se 2 segundos após a anterior, com isso a 10ª chamada foi estabelecida 18

segundos após a 1^a. Como os 10 fluxos em G.711 ocupam aproximadamente 850 kbps, garantiu-se o congestionamento do enlace para este cenário.

2. No segundo cenário, também foram iniciadas 10 chamadas em G.711, porém desta vez com o CAoS sendo executado no controlador. Portanto, quando o enlace atingiu o limite superior (α), o CAoS interagiu com os nós, solicitando a adaptação do codec para o SPEEX. Com isso, notou-se a redução da taxa no enlace para 500 kbps, valor menor que o limite α . Assim como no primeiro teste, para cada cenário de medições foram realizadas em 20 rodadas ($\mathbf{n}$) de 60 segundos cada. Resultados na Figura 6.3 são representados pelos valores médios ($\bar{x}$). As barras de erro são representadas pelo erro padrão (SE).

Variável	Significado	Valor
B	Taxa do link	650 kbps
Δt	Latência unidirecional do enlace	5 ms
α	Limite superior de consumo de banda	550 kbps
β	Limite inferior de consumo de banda	200 kbps
P	<i>Polling</i>	2 s

Tabela 6.2: Parâmetros do Teste #2

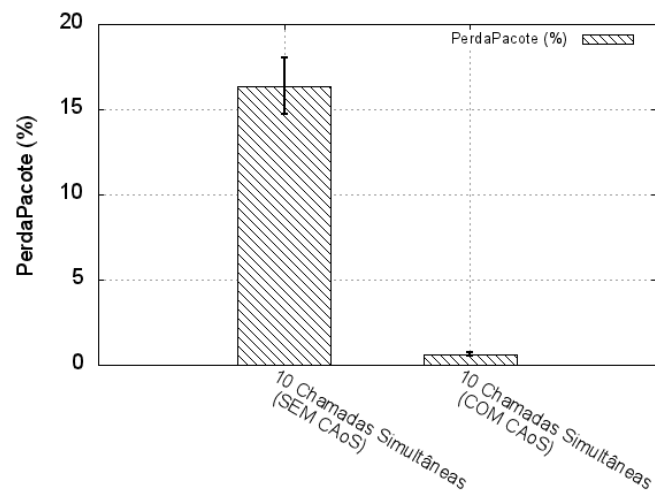
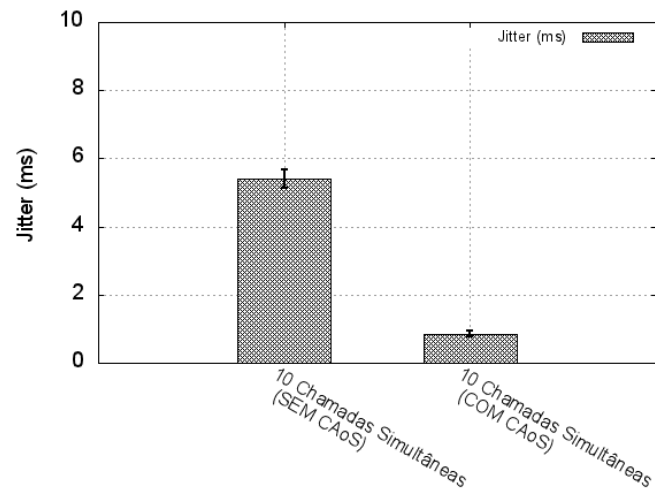
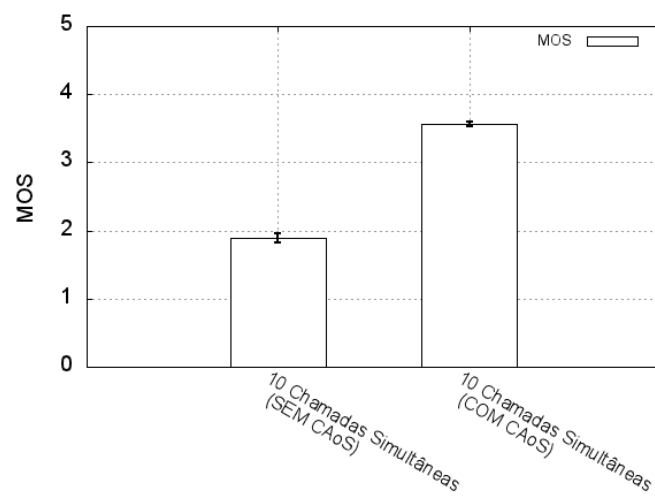
(a) Análise da *perda de pacotes*.(b) Análise do *jitter*.(c) Análise do *MOS*.

Figura 6.3: Desempenho do CAoS para múltiplas chamadas.

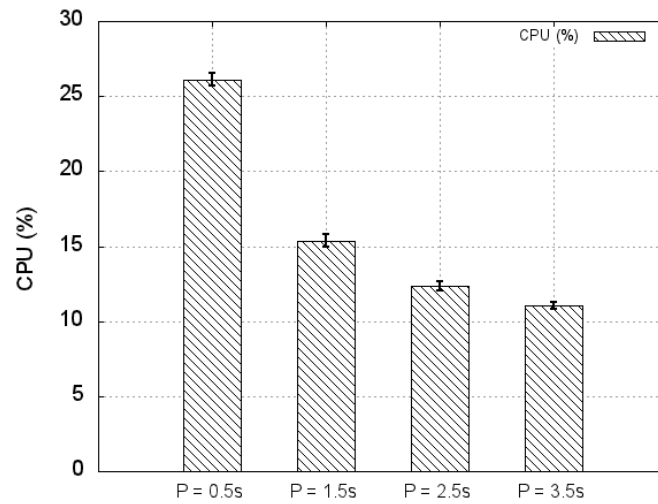
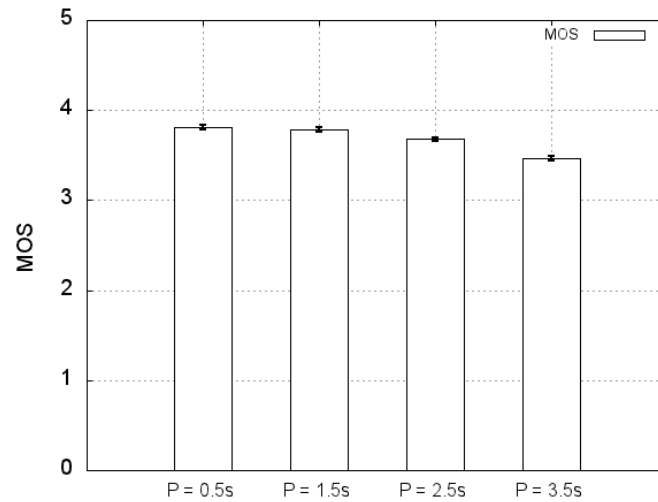
O segundo teste mostrou uma considerável melhora quando usa-se o CAoS para as 10 chamadas. Vemos clara melhora nos valores de perda de pacotes e *jitter*, resultando em excelentes valores de MOS, o qual atinge índices entre 3 e 4, que representam valores para uma qualidade de conversação razoável/boa [54] [55].

O uso de 10 pares de clientes VoIP na emulação reflete um bom cenário real, pois em ambiente corporativo onde cada usuário em média gera 0,03 E de tráfego de voz, ratificamos pela tabela B de Erlang [56] que 10 canais de voz estariam bem dimensionados para suportar 4,461 E na hora de maior movimento (com 1% de taxa de bloqueio). Com isso, 4,461 E condiz com um enlace bem dimensionado para 148 usuários (0,03 E / usuário). Ou seja, um enlace que suporte 10 chamadas simultâneas está bem dimensionado para atender um escritório ou departamento com 148 usuários de voz.

Teste #3 No terceiro teste, variou-se o valor do período entre as medições das estatísticas (P - *polling*), ainda com 10 chamadas simultâneas, enlace com taxa de 650 kbps e CAoS ativo. Foram comparados 4 cenários com valores de P distintos, medindo-se o consumo de CPU do hardware que executa o controlador POX e o MOS das chamadas. Para cada cenário de medições foram realizadas em 20 rodadas (**n**) de 60 segundos. Resultados na Figura 6.4 são representados pelos valores médios ($\bar{x}$). As barras de erro são representadas pelo erro padrão (SE).

Variável	Significado	Valor
B	Taxa do link	650 kbps
Δt	Latência unidirecional do enlace	5 ms
α	Limite superior de consumo de banda	550 kbps
β	Limite inferior de consumo de banda	200 kbps
P	<i>Polling</i>	Variável

Tabela 6.3: Parâmetros do Teste #3

(a) Análise de *consumo de CPU*.(b) Análise do *MOS*.Figura 6.4: Performance do CAoS variando período entre medições (*Polling*).

Este teste mostrou que seguramente o impacto do período entre medições está no consumo de processamento da controladora, comprovando o esperado. Conclui-se também que quanto menor o *polling*, o MOS sempre apresentará valores melhores. Isto é evidente dado que a rede responderá mais rapidamente às alterações de

taxa do enlace, fazendo com que a perda de pacotes e *jitter* sejam pouco significantes. Para o resultado deste teste, as diferenças das pontuações do MOS foram pequenas, não justificando o uso de valores pequenos para P, o que acarreta alto consumo de processamento do controlador com pouco ganho efetivo na qualidade das chamadas. O bom valor do MOS para P=3,5s deveu-se ao fato de que as chamadas foram iniciadas de 2 em 2 segundos, portanto pouca degradação ocorreu até o CAoS atuar. Caso fossem arbitrados valores mais espaçados para o *polling*, certamente haveria maior degradação do MOS para valores mais altos de P.

Adicionalmente ao terceiro teste foi medida a quantidade de bytes de cada mensagem openflow de *request* e *reply* relativas às estatísticas de fluxo e portas, solicitadas a cada P segundos (Figura 6.5), as quais totalizam 676 bytes, que é o total transferido nos dois sentidos da comunicação.

Protocol	Length	Info
OpenFlow	86	Type: OFPT_STATS_REQUEST
OpenFlow	390	Type: OFPT_STATS_REPLY
OpenFlow	122	Type: OFPT_STATS_REQUEST
OpenFlow	78	Type: OFPT_STATS_REPLY

Figura 6.5: Mensagens de requisição e resposta relativas às estatísticas de fluxo e portas enviadas do comutador ao controlador.

Com isso, para cada P apresentado na Figura 6.4(b), temos o equivalente tráfego de controle (coleta de estatísticas) apresentado na Tabela 6.4, sendo a taxa¹ de controle (Tx_{ctrl}) dada pela fórmula abaixo.

$$Tx_{ctrl} = \frac{676*8}{2*1000*P} [kbps]$$

Polling	Tx_{ctrl}
0,5s	5,4 kbps
1,5s	1,8 kbps
2,5s	1,1 kbps
3,5s	0,8 kbps

Tabela 6.4: Tráfego simétrico de controle.

¹Taxa simétrica: por isso a fórmula apresenta divisão por 2.

Pode-se concluir que o valor da taxa de controle para $P = 0,5s$ é relativamente alta (5,4 kbps) se considerarmos poucas chamadas em curso², fato este que pode ser desconsiderado se tratando de uma rede corporativa.

Teste #4 No quarto e último teste, foram comparados os protocolos usados no módulo SIP. Até o momento, nos testes anteriores, verificou-se a eficácia do CAoS usando o UDP, protocolo que não é orientado a conexão, ou seja, o módulo SIP residente no controlador simplesmente envia a mensagem de alteração de codec para o cliente SIP (PJSIP) e acredita que este tenha recebido a mensagem. Porém como não é orientado a conexão e não provê confiabilidade, os pacotes da mensagem podem sofrer qualquer degradação durante o trajeto até o cliente SIP e simplesmente não serem recebidos e, com isso, o CAoS perde sua eficácia.

Dada essa característica do UDP, resolveu-se implementar o TCP como protocolo de camada 4 para a comunicação $POX \leftrightarrow PJSIP$. Sendo orientado a conexão e provendo confiabilidade, o TCP garantirá que a mensagem enviada pelo CAoS é recebida pelo PJSIP, mesmo sendo necessária a retransmissão da mesma.

Comparou-se então 4 cenários, cada um com valores de taxa de enlace (B), α e β distintos, como apresentados acima. Respectivamente a cada cenário, foram realizadas 2, 5, 7 e 10 chamadas simultâneas, respeitando-se 2 segundos entre o estabelecimento de cada uma.

Variável	Significado	Valor
B	Taxa do link	160; 360; 480; 650 kbps
Δt	Latência unidirecional do enlace	5 ms
α	Limite superior de consumo de banda	140; 260; 380, 550 kbps
β	Limite inferior de consumo de banda	$B * 30\%$ kbps
P	<i>Polling</i>	2 s

Tabela 6.5: Parâmetros do Teste #4

O teste mostra (figura 6.6) que a utilização do TCP torna o CAoS mais confiá-

²Exemplo: para 1 chamada em G.711, teríamos $5,4/85 = 6\%$ de taxa de controle.

vel, ou seja, os valores obtidos em perda de pacote, *jitter* e MOS são ligeiramente melhores do que quando utilizamos o UDP. Isto prova que em certas rodadas usando o módulo SIP com o protocolo UDP, a troca de codec não ocorreu da melhor forma: provavelmente por uma questão de congestionamento a mensagem UDP não alcançou o destino, o que acarretou em medições inferiores para todos os quesitos dos cenário em que o UDP foi usado.

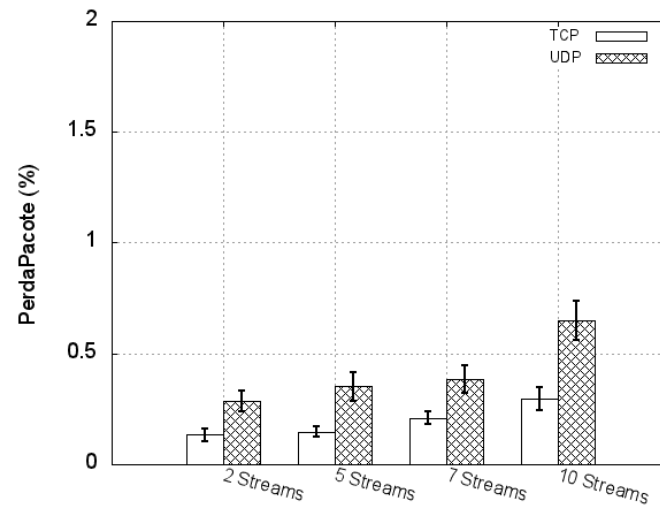
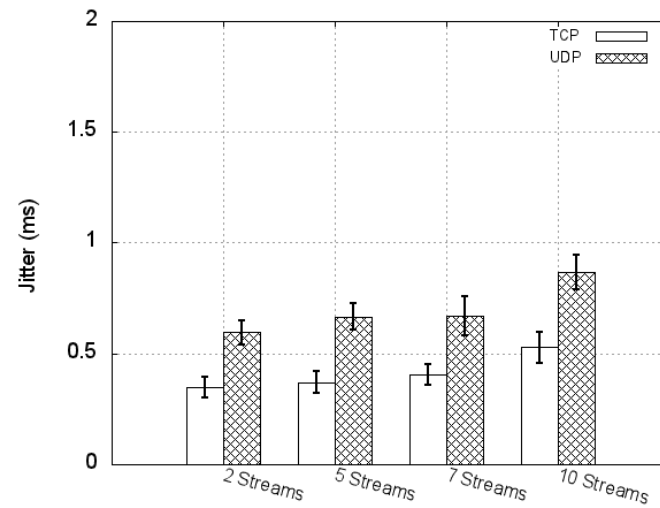
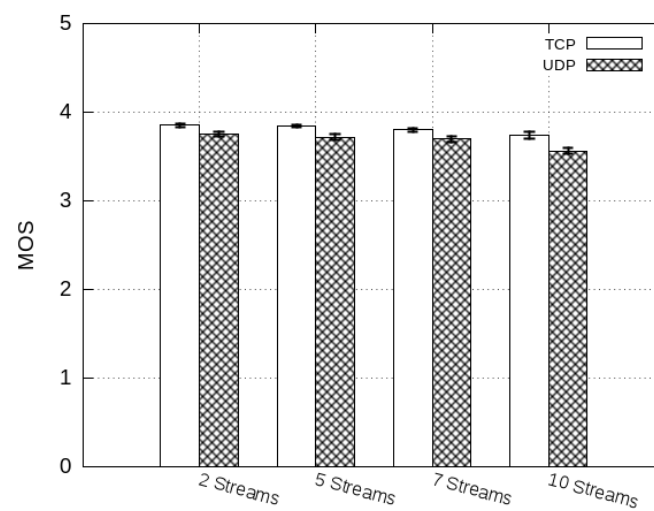
(a) Análise da *perda de pacotes*.(b) Análise do *jitter*.(c) Análise do *MOS*.

Figura 6.6: Desempenho do CAoS - TCP x UDP.

Capítulo 7

Conclusão

A Telefonia IP é uma realidade para o ambiente corporativo. Muitas empresas de pequeno, médio e grande porte estão migrando ou já migraram seus serviços de telefonia tradicional para IP, dadas as vantagens observadas por este trabalho, como as **rotas de menor custo, comunicações unificadas e gerenciamento centralizado**. Utilizando as redes IP, o serviço de telefonia precisa ser tratado de forma diferenciada, e para isso algumas importantes técnicas são usadas para que os pacotes que contém as informações da voz não sejam degradados, haja visto que a telefonia é um serviço de tempo real. Técnicas como o QoS (qualidade de serviço) e o CAC (controle de admissão de chamadas) são comumente utilizadas em equipamentos de borda para que o serviço de voz seja respectivamente *priorizado* e *limitado* para que o nível de serviço equivalha ao da telefonia tradicional (legada).

Este estudo apresentou que as técnicas de QoS e CAC, amplamente usadas em redes corporativas, são bastante úteis para o correto tratamento dos pacotes de voz, porém vislumbrou-se um melhor aproveitamento dos recursos dinâmicos de rede, principalmente nos enlaces WAN entre sítios de uma empresa, os quais são mais restritos em termos de disponibilidade de banda, e seu mau aproveitamento causa desperdícios financeiros operacionais.

As Redes SDN têm sido tema de muitos trabalhos no meio acadêmico nos

últimos anos, e atualmente algumas grandes empresas já possuem redes SDN implantadas em seu parque, como por exemplo a Google [57]. Outra grande empresa das telecomunicações já até comercializa controladores OpenFlow, como a Cisco Systems [58]. As redes SDN trazem de volta o conceito de centralização do gerenciamento da rede associado a programabilidade do controlador SDN através do uso de uma interface aberta com os comutadores e roteadores.

De forma a associar o gerenciamento centralizado da Telefonia IP e do ambiente SDN, este trabalho provou que a programabilidade do controlador SDN pode ser muito útil quando variações da rede podem afetar na qualidade do serviço ofertado. Exemplo disto é evidenciado quando o CAC é usado e o limite de chamadas na hora de maior movimento (HMM) é atingido, ou seja, novas chamadas são bloqueadas naquele horário. Para contornar esse problema, são usualmente propostas duas soluções: ampliar o enlace WAN ou aumentar a fila de voz para que mais chamadas possam ser realizadas no horário de pico de chamadas. A primeira solução possui custo muito elevado, enquanto a segunda implica maiores atrasos para as aplicações que cederam banda para o aumento da fila de voz.

Esta dissertação elaborou uma proposta chamada CAoS: *Codec Adaptation over SDN*. O CAoS é um módulo de redes SDN, elaborado exclusivamente em software executado no controlador da rede. Sua função é puramente destinada à rede de voz, onde o CAoS monitora, decide e invoca a adaptação do codec de voz de uma chamada em curso quando a fila de voz do enlace por onde aquela chamada está trafegando chega próximo a sua carga máxima. Basicamente o módulo monitora o enlace através da troca de mensagens OpenFlow entre o controlador e o comutador de borda, cuja interface é destinada ao tráfego WAN. Uma vez atingido o limiar pré-definido pelo administrador, o controlador envia uma mensagem de notificação aos clientes de voz para que estes alterem seus codecs por outros que consumam menos banda. Com isso, ao invés do CAC agir bloqueando chamadas por estarem ferindo o limite máximo da banda da fila de voz, o CAoS permite que mais chamadas possam ser realizadas usando-se menos recursos de rede.

Os resultados apresentados demonstram que a qualidade das chamadas, evidenciadas pelas leituras do MOS (*Mean Opinion Score*), são mantidas em níveis bastante aceitáveis quando o CAoS está ativo, em comparação quando este não atua. Testes foram realizados para diversos cenários, e em todos eles os resultados foram satisfatórios, aumentando a robustez do sistema de voz durante a simulação de HMM. Esta simulação foi realizada efetuando-se mais chamadas do que o enlace suportaria. Quando o CAoS não atuou, viu-se valores de *jitter*, atraso e perda de pacotes elevados, e quando o módulo foi habilitado no controlador, esses parâmetros permaneceram em níveis aceitáveis.

Portanto podemos afirmar que o presente estudo mostra que o método CAoS melhora os valores de MOS através da alteração do codec em uso durante chamadas ativas e permite que um maior número de chamadas possam ser estabelecidas durante a HMM, não degradando a qualidade das mesmas. Além disso, a dissertação mostra que a funcionalidade de adaptação do codec de voz, CAoS, pode ser implementada simultaneamente aos métodos tradicionais de prevenção de congestionamento, em uma única entidade, o controlador SDN.

7.1 Trabalhos futuros

Esta dissertação evidencia que a ideia proposta de adaptação do codec de voz a partir de uma solicitação oriunda do controlador OpenFlow v1.0 durante uma chamada é totalmente viável.

Testes mais aprofundados podem ser realizados na tentativa de ter-se um valor variável dinamicamente do intervalo das medições (P). Por exemplo, caso o tráfego na interface WAN seja alto, este período pode ser automaticamente ajustado para um valor curto, com isso o CAoS dará à rede uma rápida resposta, porém aumentando o tráfego de controle. Por outro lado, para um baixo tráfego, o período pode ser longo, aliviando a rede das mensagens de controle. Ainda referindo-se às medições para variados valores de *polling* (teste 3), rodadas com valores de P

mais espaçados podem ser realizadas, tendo em vista que o resultado do teste 3 pouco alterou para o valor de MOS devido ao fato das chamadas iniciarem a cada 2 segundos, o que é próximo ao máximo valor arbitrado para P naquele teste.

Outrossim, trabalhos futuros podem focar o uso de comutadores reais OpenFlow controlados por um controlador OpenFlow em versão mais recente, como a v1.3, e com hardware mais avançado, pois o ambiente Mininet é muito limitado. Os clientes VoIP podem ser executados por *softphones* instalados em laptops conectados aos comutadores ou mesmo usando infraestrutura de redes sem fios (WLAN). Utilizando redes reais, acredita-se que a escalabilidade dos testes pode ser alcançada, e para maiores cenários de estresse pode-se utilizar a aplicação SIPp [59]. Esta aplicação de código aberto é um gerador de tráfego SIP, estabelecendo e finalizando múltiplas chamadas SIP a partir de um único servidor, usando as trocas de mensagens padrão do SIP, como INVITE, UPDATE, BYE, OK, etc.

A versão 1.3 do OpenFlow traz nova funcionalidade que pode ser muito útil para testes futuros: *as medições de tráfego por fluxo*. Ou seja, ao invés do controlador OpenFlow medir o tráfego da interface, a medição poderia ser feita diretamente em relação aos fluxos RTP, com isso acredita-se que uma maior acurácia poderia ser alcançada para os valores de tráfego medidos.

Esta dissertação não mostrou o funcionamento do CAoS com outras técnicas que evitam a degradação do serviço de voz, e futuros testes poderiam ser feitos agregando-se funções como QoS, CAC e CAoS em um mesmo controlador.

O CAoS apresentado interagiu especificamente com os clientes SIP, mas em nenhum momento esta interação (controlador-cliente) usou o protocolo SIP em si. Os clientes sofreram uma adaptação para receber e ler a mensagem do controlador, e então iniciar as trocas de mensagens SIP entre si (*update*), mas o protocolo de sinalização de telefonia poderia ser outro, como por exemplo o H.323. Testes futuros podem diversificar o tipo de protocolo de sinalização criando diferentes API's nos clientes para suportar diferentes protocolos.

Os resultados apresentaram uma suave melhora quando o socket usado entre a comunicação controlador-cliente foi o TCP, dada sua confiabilidade na conexão. Porém, como o cabeçalho deste protocolo é maior, acaba-se desperdiçando banda. Uma proposta para futuros trabalhos é a evolução desta comunicação controlador-cliente sobre o socket UDP, criando uma resposta deste cliente ao controlador, obtendo-se assim a confiabilidade no protocolo de camada superior usando o UDP, cujo cabeçalho é menor.

A API desenvolvida nos clientes torna a solução presa, pois qualquer atualização de funcionalidade envolve também a atualização de centenas ou milhares de dispositivos. Vislumbra-se que futuramente pode ser desenvolvida uma forma mais transparente para o cliente de voz, como por exemplo o cliente receber o “comando” de troca de codec nativamente ao protocolo que esteja usando, como por exemplo uma mensagem SIP real vinda do controlador. Mas para isso ocorrer, o cliente deverá previamente “confiar” no controlador como sendo uma entidade da rede de voz. Com isso, talvez seja necessário desenvolvimento de uma nova API no controlador para emular um SIP proxy confiável.

Viu-se que ao início de cada chamada o CAoS assume que o par de clientes esteja usando um determinado codec de voz, pois em uma rede corporativa isto seria fácil de determinar. Porém para que o controlador realmente saiba qual codec está sendo usado, este precisaria receber o pacote de convite (*invite*) e da resposta “OK” entre os pares de clientes, para ler o campo SDP. Para isso, o comutador OpenFlow deve ter uma regra para que a cópia destas mensagens seja enviada para o controlador. Outra abordagem seria fazer o controlador receber alguns exemplares de pacotes RTP, cuja informação de codec também está inserida.

Assim como esta dissertação apresentou a alternância de apenas 2 codecs, pois tratamos de ambiente corporativo, o qual pode ser controlado, pode-se também aplicar esta ideia para o tratamento de mais de 2 codificadores, fazendo uma segmentação dos valores limiares da vazão do enlace, definindo-se assim um codec para cada segmento. Ou seja, teríamos não somente α e β , mas como também μ ,

ϕ , λ , etc, como limiares para definição do melhor codec a ser usado. Similarmente poderia ser testado a atuação do CAoS em uma rede de videoconferência, alterando codecs de vídeo para determinada vazão do link - VAoS (*Video Adaptation over SDN*).

Além do POX, outros controladores OpenFlow como o NOX, Beacon, Maestro, Jaxon, podem ser utilizados nos cenários emulados. Com isso pode-se comparar novos resultados àqueles apresentados nesta dissertação, principalmente no que diz respeito ao desempenho do controlador.

Referências

- [1] Juniper Networks, *Introduction to OpenFlow.*, <http://pt.slideshare.net/junosdeveloper/openflow-overview>, Acessado em Julho 2015.
- [2] N. Costa and M. Nunes, “Adaptive quality of service in voice over IP communications,” in *Fifth International Conference on Networking and Services (ICNS '09)*, April 2009, pp. 19–24.
- [3] M. S. R. M. de Carvalho, “A trajetória da internet no brasil: do surgimento das redes de computadores à instituição dos mecanismos de governança,” Master’s thesis, Universidade Federal do Rio de Janeiro, RJ, Brasil, Sep. 2006.
- [4] Ministério da Comunicações, *História da Telefonia*, <http://www.mc.gov.br/acoes-e-programas/redes-digitais-da-cidadania/44-historia-das-comunicacoes/22463-historia-da-telefonica>, Acessado em Junho 2015.
- [5] H. J. Wang and R. H. Katz, “Mobility support in unified communication networks,” in *Proceedings of the 4th ACM International Workshop on Wireless Mobile Multimedia*, ser. WOWMOM '01. New York, NY, USA: ACM, 2001, pp. 95–102. [Online]. Available: <http://doi.acm.org.ez24.periodicos.capes.gov.br/10.1145/605991.606004>
- [6] J. Kozik, W. A. Montgomery, and J. J. Stanaway, “Voice services in next-generation networks: The evolution of the intelligent network and its role in generating new revenue opportunities,” *Bell Labs Technical Journal*, vol. 3, no. 4, pp. 124–143, Oct 1998.
- [7] K. Park, Y. Oh, K. Lim, and K. Cho, “A real-time voice service with the adaptive packet loss recovery scheme in the hybrid residential gateway system,” *Consumer Electronics, IEEE Transactions on*, vol. 49, no. 2, pp. 359–366, May 2003.
- [8] T. D. Nadeau and K. Gray, *SDN: Software Defined Networks*. Sebastopol, CA: O’Reilly, 2013.

- [9] K. O. Stoeckigt, H. L. Vu, and P. Branch, “Dynamic codec with priority for voice over ip in wlan,” in *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, ser. MMSys '11. New York, NY, USA: ACM, 2011, pp. 199–210. [Online]. Available: <http://doi.acm.org.ez24.periodicos.capes.gov.br/10.1145/1943552.1943579>
- [10] D. Kwon, R. Thay, H. Kim, and H. Ju, “QoE-based adaptive mVoIP service architecture in SDN networks,” in *The Thirteenth International Conference on Networks (ICN '14)*, 2014, pp. 62–67.
- [11] H. Kim and N. Feamster, “Improving network management with software defined networking,” *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, February 2013.
- [12] M. F. Finneran, *VoWLAN: Voice Over WLANs: The Complete Guide*. Burlington, MA: Newnes, 2007.
- [13] M. A. Filippetti, *CCNA 5.0: Guia Completo de Estudo*. Florianópolis, SC: Visual Books, 2014.
- [14] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, “Hypertext transfer protocol - HTTP/1.1,” RFC 2616, Jun. 1999.
- [15] J. Klensin, “Simple mail transfer protocol,” RFC 2821, Apr. 2001.
- [16] R. F. H. Schulzrinne, S. Casner and V. Jacobson, “RTP: A transport protocol for real-time applications,” RFC 3550, Jul. 2003.
- [17] Cisco Systems Inc., *Voice Over IP - Per Call Bandwidth Consumption*, <http://www.cisco.com/c/en/us/support/docs/voice/voice-quality/7934-bwidth-consume.html>, Acessado em Junho 2015.
- [18] ITU, *Series H: Audiovisual and Multimedia Systems*, <https://www.itu.int/rec/T-REC-H.323-200912-I/en>, Acessado em Julho 2015.
- [19] F. Andreassen and B. Foster, “Media gateway control protocol (mgcp),” RFC 3435, Jan. 2003.
- [20] T. Taylor, “Reclassification of rfc 3525 to historic,” RFC 5125, Feb. 2008.
- [21] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, “SIP: Session initiation protocol,” RFC 3261, Jun. 2002.

- [22] M. J. Steinmann, “Unified communications with sip,” *Queue*, vol. 5, no. 2, pp. 50–55, Mar. 2007. [Online]. Available: <http://doi.acm.org.ez24.periodicos.capes.gov.br/10.1145/1229899.1229912>
- [23] K. Nichols, S. Blake, F. Baker, and D. Black, “Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers,” RFC 2474, Dec. 1998.
- [24] A. V. E. Rosen and R. Callon, “Multiprotocol label switching architecture,” RFC 3031, Jan. 2001.
- [25] H. Schulzrinne and S. Casner, “Rtp profile for audio and video conferences with minimal control,” RFC 3551, Jul. 2003.
- [26] ITU, *The E-model, a computational model for use in transmission planning*, <http://www.itu.int/rec/T-REC-G.107-201402-P/en>, Acessado em Julho 2014.
- [27] R. C. T. Friedman and A. Clark, “Rtp control protocol extended reports RTCP XR,” RFC 3611, Nov. 2003.
- [28] C. Olariu, J. Fitzpatrick, P. Perry, and L. Murphy, “A QoS based call admission control and resource allocation mechanism for LTE femtocell deployment,” in *2012 IEEE Consumer Communications and Networking Conference (CCNC)*, Jan 2012, pp. 884–888.
- [29] F. Almeida and J. Cruz, “Open source unified communications: The new paradigm to cut costs and extend productivity,” in *Proceedings of the Workshop on Open Source and Design of Communication*, ser. OSDOC ’12. New York, NY, USA: ACM, 2012, pp. 11–12. [Online]. Available: <http://doi.acm.org.ez24.periodicos.capes.gov.br/10.1145/2316936.2316939>
- [30] G. Su, T. O. Pham, and S. He, “QoS guarantee for IPTV using low latency queuing with various dropping schemes,” in *2012 International Conference on Systems and Informatics (ICSAI)*, May 2012, pp. 1551–1555.
- [31] D. Kreutz, F. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.
- [32] K. Bakshi, “Considerations for software defined networking (SDN): Approaches and use cases,” in *2013 IEEE Aerospace Conference*, March 2013, pp. 1–9.

- [33] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: Enabling innovation in campus networks,” *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008. [Online]. Available: <http://doi.acm.org/10.1145/1355734.1355746>
- [34] C. E. Rothenberg, M. R. Nascimento, and M. R. Salvador, “OpenFlow e redes definidas por software : um novo paradigma de controle e inovação em redes de pacotes,” *Control*, vol. 7, pp. 65–75, 2011.
- [35] Y. Watashiba, S. Hirabara, S. Date, H. Abe, K. Ichikawa, Y. Kido, S. Shimojo, and H. Takemura, “OpenFlow network visualization software with flow control interface,” in *2013 IEEE 37th Annual Computer Software and Applications Conference (COMPSAC)*, July 2013, pp. 475–477.
- [36] A. Lara, A. Kolasani, and B. Ramamurthy, “Network innovation using open-flow: A survey,” *Communications Surveys Tutorials, IEEE*, vol. 16, no. 1, pp. 493–512, First 2014.
- [37] D. Macedo, D. Guedes, L. Vieira, M. Vieira, and M. Nogueira, “Programmable networks: From software defined radio to software defined networking,” *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, pp. 1–1, 2015.
- [38] B. Lantz, B. Heller, N. Handigol, and V. Jeyakumar, *Mininet*, <http://www.mininet.org>, Acessado em Fevereiro 2014.
- [39] S.-Y. Wang, “Comparison of sdn openflow network simulator and emulators: Estinet vs. mininet,” in *Computers and Communication (ISCC), 2014 IEEE Symposium on*, June 2014, pp. 1–6.
- [40] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, “Nox: Towards an operating system for networks,” *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 3, pp. 105–110, Jul. 2008. [Online]. Available: <http://doi.acm.org.ez24.periodicos.capes.gov.br/10.1145/1384609.1384625>
- [41] NOX Repo Community, *NOX Repo*, <http://www.noxrepo.org/>, Acessado em Outubro 2014.
- [42] S. Gorlatch, T. Humernbrum, and F. Glinka, “Improving QoS in real-time internet applications: from best-effort to software-defined networks,” in *2014 International Conference on Computing, Networking and Communications (ICNC)*, Feb 2014, pp. 189–193.

- [43] Cisco Systems Inc., *Lista de Preços do Cisco Prime Collaboration*, <http://www.zones.com/site/product/index.html?id=102230431>, Acessado em Junho 2015.
- [44] Paessler AG, *Lista de Preços do PRTG - Paessler Router Traffic Grapher*, https://www.br.paessler.com/prtg/price_list, Acessado em Junho 2015.
- [45] K. C. J. Babiarz and F. Baker, “Configuration guidelines for diffserv service classes,” RFC 4594, Aug. 2006.
- [46] J. Rosenberg, “The session initiation protocol (SIP) update method,” RFC 3311, Sep. 2002.
- [47] VoipMonitor, *Monitoring VoIP Parameters*, <http://www.voipmonitor.org>, Acessado em Julho 2014.
- [48] Teluu Ltd., *PJSIP*, <http://www.pjsip.org>, Acessado em Maio 2014.
- [49] J. Boxman, *A Practical Guide to Linux Traffic Control*, http://blog.edseek.com/~jasonb/articles/traffic_shaping/index.html, Acessado em Junho 2015.
- [50] ITU, *Recommendation G.711*, <http://www.itu.int/rec/T-REC-G.711-198811-I/en>, Acessado em Julho 2014.
- [51] G. Herlein, J. Valin, A. Heggstad, and A. Moizard, “RTP payload format for the speex codec,” RFC 5574, Jun. 2009.
- [52] ITU, *Recommendation G.729*, <http://www.itu.int/rec/T-REC-G.729-201206-I/en>, Acessado em Julho 2014.
- [53] Cisco Systems Inc., *Understanding Codecs: Complexity, Hardware Support, MOS, and Negotiation*, http://www.cisco.com/en/US/tech/tk1077/technologies_tech_note09186a00800b6710.shtml#mos, Acessado em Junho 2014.
- [54] W.-E. Chen, P.-J. Lin, and Y.-B. Lin, “Real-time VoIP quality measurement for mobile devices,” *IEEE Systems Journal*, vol. 5, no. 4, pp. 538–544, Dec 2011.
- [55] A.-N. Moldovan, I. Ghergulescu, and C. Muntean, “A novel methodology for mapping objective video quality metrics to the subjective MOS scale,” in *2014 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, June 2014, pp. 1–7.

-
- [56] R. L. Freeman, *Fundamentals of Telecommunications*. New York, NY: John Wiley and Sons, Inc, 1999.
 - [57] Cade Metz, *Google Foretells Big Switch to Networks of the Future*, <http://www.wired.com/2012/11/big-switch/>, Acessado em Maio 2015.
 - [58] Cisco Systems, Inc., *Cisco Open SDN Controller*, <http://www.cisco.com/c/en/us/products/cloud-systems-management/open-sdn-controller/index.html>, Acessado em Maio 2015.
 - [59] R. Gayraud and O. Jacques, *SIPp*, <http://sipp.sourceforge.net/>, Acessado em Junho 2015.