UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Gerenciamento Adaptativo da Qualidade da Fala entre Terminais VoIP

Leandro Silva Galvão de Carvalho

Manaus
outubro de 2011

UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Leandro Silva Galvão de Carvalho

Gerenciamento Adaptativo da Qualidade da Fala entre Terminais VoIP

Tese apresentada ao Programa de Pós Graduação em Informática da Universidade Federal do Amazonas, como requisito parcial para a obtenção do título de *Doutor em Informática*, área de concentração em Redes de Computadores e Telecomunicações.

orientador:
Prof. Dr.-Ing. Edjair de Souza Mota

Manaus
outubro de 2011

Agradecimentos

Antes de tudo, agradeço a Deus, por me permitir concluir este trabalho.

Agradeço também ao meu orientador, Edjair de Souza Mota, pelo estímulo contínuo e paciência. Um muito obrigado também ao professor Edjard de Souza Mota e ao colega Alexandre Passito pelas discussões e insights sobre adaptabilidade e sistemas autonômicos durante os cafés da manhã que antecediam as nossas aulas.

Agradeço toda a ajuda dos meus alunos no Laboratório de Redes de Computadores e Multimídia (GRCM), especialmente ao Thiago Vianna, pela ajuda na implementação do código das estratégias adaptativas. Obrigado também ao Bruno Auzier e Higo Ferreira, pela aplicação dos testes subjetivos, e à Daniele Seixas e ao Diogo Soares, pela ajuda na montagem do ambiente de experimentos.

Agradeço profundamente o apoio de todos os meus colegas do Instituto de Computação na Universidade Federal do Amazonas. Muitíssimo obrigado à Elienai Nogueira, por todo o apoio e suporte não apenas como Coordenadora Administrativa, mas também como uma sincera amiga.

Finalmente, obrigado à minha mãe por todo o amor e apoio, e ao meu pai (*in memoriam*) pelo seu encorajamento e exemplo de vida.

Resumo

Chamadas de voz baseadas na tecnologia VoIP (Voice over Internet Protocol) estão suscetíveis a degradações diversas, provenientes tanto da camada de aplicação, como da camada de rede, tais como compressão do codec, atraso fim a fim e perda de pacotes. Durante anos, esse problema tem desafiado pesquisadores e profissionais, que têm concebido e melhorado mecanismos de controle de QoS para aplicações VoIP. Tais mecanismos visam otimizar a utilização dos recursos da rede e do terminal VoIP de modo a minimizar os efeitos deletérios da rede subjacente sobre a qualidade de voz. Entre as várias propostas de mecanismos de controle de QoS para VoIP, alguns deles procuram *adaptar* o fluxo de voz ou outros parâmetros VoIP de acordo com mudanças significativas na rede, preferências de usuário, ou requisitos dos provedores de serviços VoIP.

Sistemas VoIP particularmente exigem soluções de adaptação dinâmica para lidar com a complexa relação de compromisso entre qualidade de voz e fatores de degradação, por causa da natureza descentralizada e estocástica das redes IP na entrega de pacotes de voz. Embora as soluções adaptativas existentes para controle de QoS em VoIP mostrem alguma melhora de desempenho e apresentem algum tipo de *feedback*, elas não fornecem foco explícito na ciclo de controle (*control loop*).

Este documento mostra o progresso atual da nossa tese, que aborda o ajuste de parâmetros internos de terminais VoIP (camada de aplicação) que afetam o fluxo de voz, com o objetivo de melhorar a qualidade da fala em resposta a mudanças nas condições da rede. Não faz parte do escopo da tese abordar soluções adaptativas que se concentram exclusivamente em sinalização, bilhetagem, problemas de segurança, ou que operam no nível da camada de rede.

Portanto, esta tese aborda o problema da concepção e avaliação de estratégias adaptativas que explorem as relações de compromisso entre qualidade da fala e os seguintes fatores de degradação: compressão do codec, atraso fim a fim e perda de pacotes. A finalidade é otimizar atributos da fala perceptíveis aos usuário, sob a perspectiva de sistemas de software auto-adaptativo. A ênfase não reside em desenvolver novos codecs de áudio, mas sim em desenvolver um ciclo de controle como entidade central de um terminal VoIP, que possa adaptar as configurações do fluxo de voz de acordo com as condições da rede.

As principais contribuições desta tese são as seguintes: determinação da percepção do usuário durante a comutação de codec; parametrização de precedência de codecs para suporte de decisão de comutação de codec; enfoque no ciclo de controle baseado nas atividades de monitoramento–análise–planejamento–execução como núcleo do processo de adaptação; e análise de eficiência de troca de mensagens de *feedback*.

Palavras chave: Voz sobre Protocolo de Internet (VoIP), adaptação de qualidade da fala, controle de Qualidade de Serviço (QoS), ciclo de realimentação.

Abstract

Voice calls based on Voice over Internet Protocol (VoIP) technology are liable to several impairments from both application and network layer, such as codec compression, end-to-end delay, and packet loss. For years, this problem has been challenging researchers and practitioners, who have been designing and improving QoS control mechanisms for VoIP applications. Such mechanisms aim to make optimum use of network and terminal resources so as to minimize the effects of network impairments on voice quality. Among the several proposed QoS control mechanisms for VoIP, some of them seek to *adapt* the voice flow or other VoIP-related parameters in accordance with significant changes in the network, end users' preferences, or service providers' requirements.

VoIP systems are particularly likely to require a dynamic adaptation solution for dealing with the complex trade-off between speech quality and impairments, because of the decentralized control nature of IP networks and the stochastic nature of data packet delivery. Although the existing adaptive solutions for QoS control of VoIP show some performance improvement and exhibit some sort of feedback, they do not provide explicit focus on the control loop.

This document shows the current progress of our thesis, which addresses the adjustment of internal parameters of VoIP terminals (at application layer) that affect the voice flow, with the aim of improving speech quality in response to changes in network conditions. It is not in the scope of the thesis to propose adaptive solutions that focus exclusively on signaling, billing, security issues, or operate at the network layer.

Therefore, this thesis addresses the problem of how adjust encoding parameters in response to variations in delay and packet loss, in order to optimize speech quality. The objective is to optimize user-perceptible attributes of speech, under the perspective of self-adaptive software systems. The emphasis is not to develop new audio codecs, but to build a control loop in the core of sender and receiver terminals to adapt voice flow settings according to network conditions.

The main contributions of this thesis are the following: determination of user's perception during codec switching; parametrization of codec precedence for supporting codec switching decision; explicit design of a monitoring–analysis–planning–execution control loop as the core of the adaptation process; and efficiency analysis of feedback message exchanging.

Key Words: Voice over IP, speech quality adaptation, QoS control, feedback loop.

Contents

List of Figures

List of Tables

List of Acronyms and Variables

ACELP	Algebraic Code Excited Linear Prediction
ACK	Acknowledge
ACR	Absolute Category Rating
ADPCM	Adaptive Differential Pulse Code Modulation
AIMD	Additive Increase Multiplicative Decrease
AMR	GSM Adaptive Multi-Rate
AMR-WB	AMR Wideband
ANOVA	Analysis of Variance
AP	Access Point
BER	Bit Error Rate
BV	Broad Voice
CAC	Call Admission Control
CBR	Constant Bitrate
CCR	Comparison Category Rating
CELP	Code Excited Linear Prediction
CHT	Call Holding Time
CMOS	Comparison Mean Opinion Score
CMR	Code Mode Request
CNG	Comfort Noise Generation
CS-ACELP	Conjugate Structure – Algebraic Code Excited Linear Prediction
DCR	Degradation Category Rating
DLSR	Delay since Last Sender Report
DMOS	Degradation Mean Opinion Score
DTX	Discontinuous Transmission
ETSI	European Telecommunications Standards Institute
FEC	Forward Error Correction
GSM	Global System for Mobile Communications
GSM-EFR	GSM – Enhanced Full Rate
GSM-FR	GSM – Full Rate
GSM-HR	GSM – Half Rate
HTTP	Hipper Text Transport Protocol
iLBC	Internet Low Bit Rate Codec
IM	Instant Message
IP	Internet Protocol
iSAC	Internet Speech Audio Codec
ITBC	Interarrival Time Between Calls
ITU	International Telecommunications Union
ITU-T	ITU Telecommunication Standardization Sector
LAN	Local Area Network
LBR	Low Bitrate Redundancy
LD-CELP	Low Delay – Code Excited Linear Prediction
LPC	Linear-Predictive Coding

MAC	Medium Access Control
MAPE	Monitoring, Analysis, Planning, Execution
MDCX	Modify Connection (MGCP command)
MGCP	Media Gateway Control Protocol
MLT	Modulated Lapped Transform
MOS	Mean Opinion Score
MP-MLQ	Multi-Pulse – Maximum Likelihood Quantizer
OPAL	Open Phone Abstraction Library
PCM	Pulse Code Modulation
PER	Packet Error Rate
PESQ	Perceptual Evaluation of Speech Quality
PLB	Packet Loss Behavior
PLC	Packet Loss Concealment
POTS	Plain Old Telephone Service
PSTN	Public Switched Telephone Network
QoE	Quality of Experience
QoS	Quality of Service
RFC	Request for Comments
RPE-LTP	Regular Pulse Excitation – Long Term Prediction
RTP	Real-time Transport Protocol
RTCP	RTP Control Protocol
RTCP SR	RTCP Sender Report
RTCP RR	RTCP Receiver Report
RTCP XR	RTCP eXtended Reports
RTT	Round-Trip Time
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SVOPC	Sinusoidal Voice Over Packet Coder
TCP	Transmission Control Protocol
TFRC	TCP-Friendly Rate Control
TSNFC	Two-Stage Noise Feedback Coding
UDP	User Data Protocol
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial Radio Access Network
VAD	Voice Activity Detector
VBR	Variable Bitrate
VCELP	Vector Code Excited Linear Prediction
VoIP	Voice over Internet Protocol
WLAN	Wireless Local Access Network
XCP	eXplicit Control Protocol
XOR	eXclusive OR

Chapter 1

Introdução

Computer science is no more about computers than astronomy is about telescopes, biology is about microscopes or chemistry is about beakers and test tubes. Science is not about tools. It is about how we use them, and what we find out when we do.
Edsjer W. Dijkstra (1930–2002)

Contents

Aplicações e serviços baseados na tecnologia de voz sobre protocolo de Internet (*Voice over IP* – VoIP), têm experimentado um crescimento rápido nas duas últimas décadas. Inicialmente, o custo-benefícios da convergência impulsionou esse crescimento. Mas agora os usuários e operadores de serviços estão cada vez mais preocupados na qualidade e dependência de serviços VoIP, porque o aumento da escalabilidade também trouxe alguns obstáculos, tais como a complexidade de configuração e esforço de gerência.

No serviço de telefonia comutado tradicional (*Plain Old Telephone Service* – POTS), cada chamada utiliza um circuito dedicado. Já em redes IP, todas as chamadas compartilham os mesmos recursos. Devido à incerteza e não-determinismo inerente a este ambiente, fluxos de voz são prejudicados pela perda de pacotes, atraso e jitter, que afetam diretamente a percepção do usuário a respeito da qualidade da fala. Durante anos, este problema tem sido desafiado por pesquisadores e profissionais, os quais têm concebido e melhorado os mecanismos de controle de QoS para aplicações de VoIP. Tais mecanismos visam otimizar a utilização dos recursos de rede e terminal para minimizar os efeitos de deficiências da rede sobre a qualidade da voz. Entre as várias propostas de mecanismos de controle de QoS para VoIP, algumas delas procuram adaptar o fluxo de voz ou outros parâmetros relacionados com VoIP de acordo com mudanças significativas na rede, preferências dos usuários, ou requisitos dos provedores de serviços.

Sistemas adaptativos em geral respondem a mudanças no seu estado interno ou ambiente externo com orientação de um sistema de controle subjacente. Sistemas de VoIP são particularmente suscetíveis de exigir adaptação dinâmica por causa da natureza descentralizada das redes IP e à natureza estocástica da emtrega de pacotes de dados. Embora as soluções adaptativas existentes para controle de QoS em VoIP mostrem alguma melhoria de desempenho e apresentem algum tipo de realimentação (*feedback*), elas não fornecem um foco explícito no laço de controle. Por exemplo, a maioria dessas soluções não consideram o desempenho em estado de transiente ou estacionário, o que se põe como obstáculo fundamental para a validação e verificação de tais soluções [141].

Além de teoria de controle adaptativo, mecanismos de gerenciamento de QoS para VoIP compartilha analogia com a teoria de confiabilidade e a iniciativa de comunicação autonômica (AutoComm). Portanto, explorar e aprofundar o formalismo dessas áreas poderia ajudar na concepção de serviços VoIP mais robustos, confiáveis e escaláveis.

1.1 Escopo de Pesquisa

O termo *adaptação VoIP* pode soar impreciso. Assim, para definir o nosso escopo de pesquisa, devemos explicar o que queremos dizer com essa expressão. De acordo com Salehie & Tahvildari [169], softwares auto-adaptativos visam, em modo geral, ajustar vários artefatos ou atributos em resposta a mudanças internas (*self*) e a mudanças no *contexto* de um sistema de software. Em um serviço VoIP em particular, consideramos o *self* como sendo o par de terminais emissor e receptor agindo como um indivíduo inteiro, e o *contexto* como sendo a rede de dados subjacente, propenso a falhas de congestionamento e de enlace.

Como será detalhado na Seção B.5.3, quando falamos de *adaptação*, duas perguntas estão implícitas: *como* e *onde*. Em relação à questão de *como adaptar*, a nossa preocupação está focada em soluções adaptativas que exigem uma interação dinâmica entre os terminais emissor e receptor. Note-se que este requisito aumenta a complexidade, se comparado com soluções adaptativas *stand-alone*, tais como (1) alguns tipos de empacotamento adaptativa [170], que é baseada unicamente na autocorrelação dos blocos consecutivos do sinal, independentemente das condições da rede ou realimentação do receptor e (2) buffer de compensação de jitter adaptáveis [77, 114, 163], que não consideram qualquer mensagem do emissor, embora sentidos informações de contexto de rede.

Em relação à questão *onde adaptar*, consideremos as três dimensões onde a adaptação pode ser aplicada em sistemas VoIP:

1. *Fluxo de dados.* Uma chamada VoIP é composta de três fluxos de dados distintos: sinalização, mídia codificada e controle. Sinalização refere-se às mensagens trocadas para estabelecer, manter e terminar uma chamada. A voz do emissor (mídia) é codificada e enviada através do Protocolo de Tempo Real (*Real-time Transport Protocol* – RTP) em pacotes através da rede subjacente em direção ao receptor. Mensagens de controle compreendem diversas informações tais como faturamento, feedback, e bilhetagem. Nesta tese, vamos nos concentrar apenas na adaptação do fluxo RTP (mídia codificada).

2. *Plano de dados.* Como apontado por Chen et al. [34], os mecanismos de QoS de gerenciamento no plano de dados podem ser mapeados para as camadas correspondentes da arquitetura TCP/IP: física, acesso à rede, inter-rede, transporte e aplicação. Primeiramente, não consideramos parâmetros nas camadas física e acesso à rede. Os termos *gerenciamento* e *adaptação* implicam leitura e alteração de parâmetros, mas, por razões de padronização, os parâmetros das camadas física e de acesso à rede não devem ser modificados ao longo da chamada. Caso contrário, os terminais não seriam capazes de se comunicarem.

   Segundo, também desconsideramos parâmetros na camada de inter-rede. De acordo com Myakotnykh [142], métodos situados na camada de inter-rede muitas vezes não resolvem completamente o problema por duas razões: (1) nem todos os equipamentos suportam múltiplos protocolos de QoS; e (2) a Internet é um ambiente dinâmico, e a maior parte das tecnologias não pode reagir às condições mutáveis da rede e gerenciar a qualidade de todas as chamadas em tempo real.

Finalmente, soluções situadas na camada de aplicação, as quais são exploradas nesta tese, são projetadas para minimizar o efeito de perda de pacotes e latência sobre a qualidade da fala e são executadas pelo emissor ou receptor do fluxo RTP [7]. Tais mecanismos, também conhecidos como baseados no sistema final, incluem controle de taxa de codificação, empacotamento, controle antecipado de erro (*Forward Error Control* – FEC), ajuste de buffer de compensação de jitter, e compensação de perda de pacotes (*Packet Loss Concealment* – PLC).

3. *Aspecto gerenciado.* Como apontado por Gokhale & Lu [61], gerenciamento VoIP pode abranger dois aspectos principais: *desempenho* e *segurança*. Desempenho está relacionado com questões não-maliciosas, tais como sobrecarga de tráfego na rede ou falhas de recursos. Segurança está relacionada a causas mal-intencionadas, tais como *phishing*, *man-in-the-middle*, e negação de serviço, entre outros. Nesta tese, vamos nos concentrar apenas no aspecto *desempenho*, em termos de qualidade da fala.

Portanto, nesta tese, usaremos o termo *adaptação VoIP* para nos referir ao ajuste de parâmetros internos de terminais VoIP (camada de aplicação) que afetam o fluxo de voz, com o objectivo de melhorar a qualidade da fala, em resposta a mudanças nas condições de rede. Não exploraremos nesta tese soluções adaptativas que gerenciam exclusivamente a sinalização de chamada ou informações de tarifação, ou que são orientadas para questões de segurança do fluxo de voz, ou que se concentram no controle de QoS na camada de rede.

Figura 1.1 fornece uma idéia conceitual do escopo de pesquisa desta tese. Os eixos representam as três dimensões onde a adaptação pode ser aplicada em sistemas de VoIP, como explicado anteriormente. A disposição dos rótulos ao longo dos eixos não reflete um significado matemático ou de ordenação, mas implica apenas que qualquer tripla tomada do diagrama é também um escopo de pesquisa válido. Por exemplo, pode-se investigar os problemas de segurança de VoIP durante sinalização na camada de rede, ou avaliar o desempenho na entrega de informações de tarifação através de alguma arquitetura de rede.

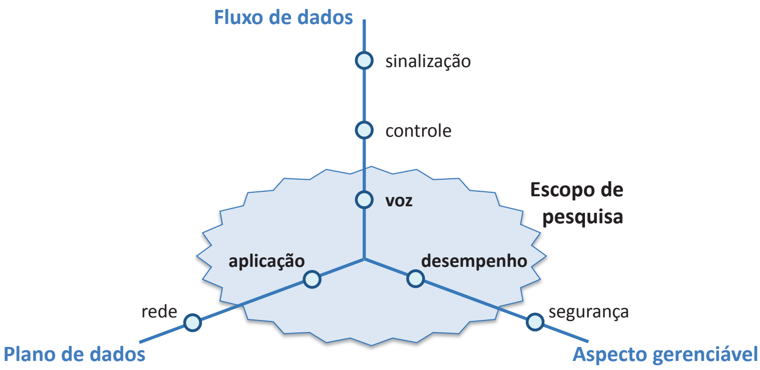


Figure 1.1: Identificação do escopo de pesquisa desta tese em meio as dimensões relacionadas com a tecnologia VoIP.

1.2 Definição do Problema

A Figura 1.2 mostra um sistema de VoIP tradicional, sem qualquer comportamento adaptativo. Ele consiste em três componentes principais: um emissor (fonte de tráfego VoIP), um receptor

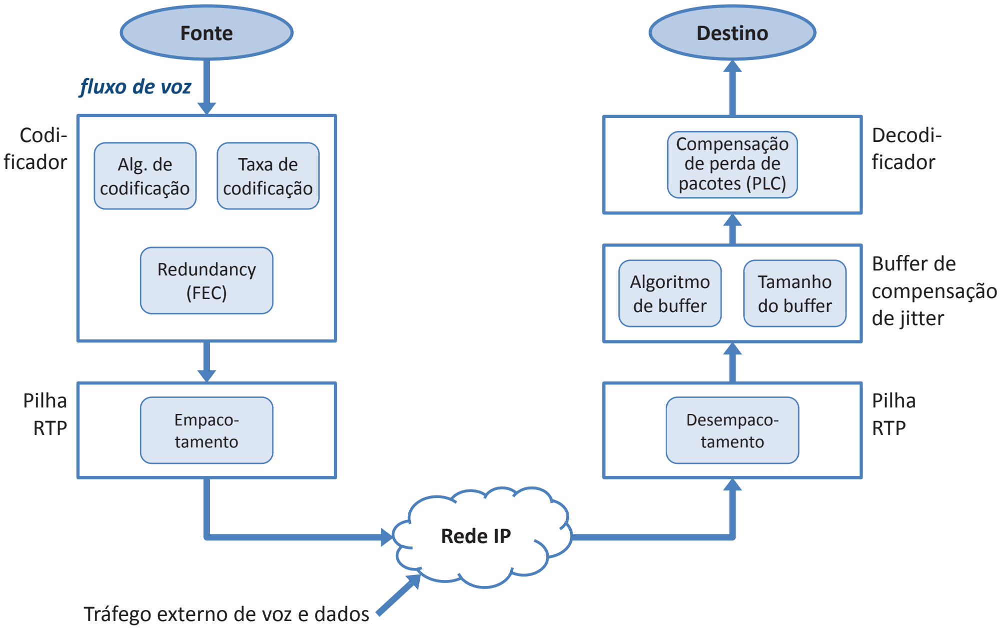


Figure 1.2: Examplo de sistema VoIP não-adaptativo.

(ponto de destino do fluxo de voz) e a rede IP subjacente. No lado do emissor, a fala é codificada em quadros de voz, que são agrupados em pacotes (empacotamento). Então, o fluxo de voz é tipicamente transportado através da rede IP juntamente com outros tipos de tráfego ou através de uma rede VoIP dedicada. No destino, o buffer de compensação de jitter elimina variações de atraso, e o decodificador compensa a pacotes perdidos antes de reproduzir o fluxo de voz reconstituído para o receptor.

De acordo com Myakotnykh [142], a fim de tornar este sistema adaptativo (ou seja, para gerenciar a qualidade da fala em tempo de execução segundo algumas políticas), é necessário

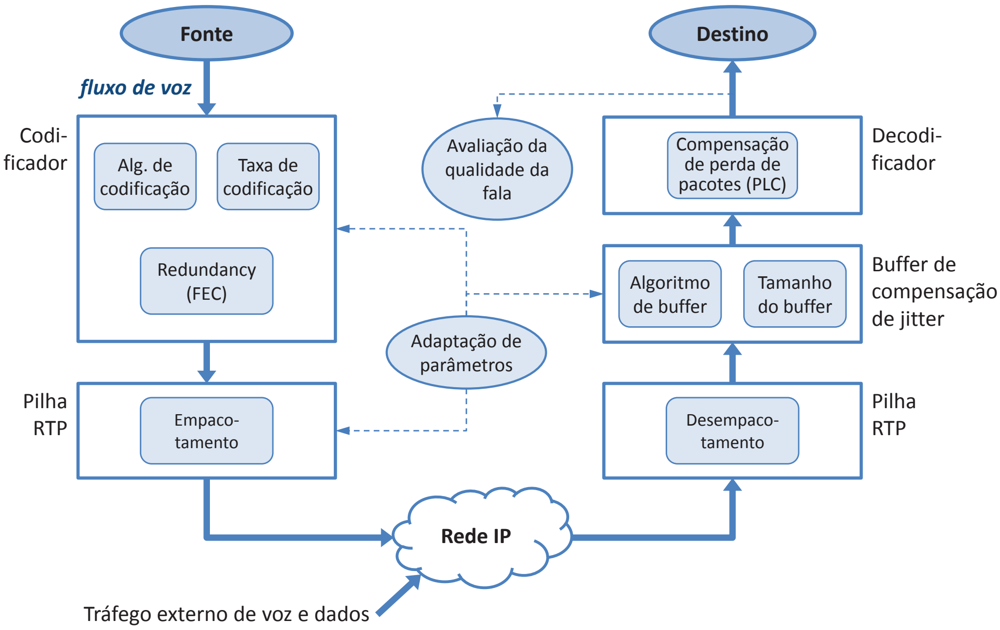


Figure 1.3: Exemplo de um sistema VoIP adaptativo, com a inclusão de dois componentes: avaliação da qualidade da fala, e adaptação de parâmetros.

projetar dois componentes: (1) mecanismos objetivos de avaliação da qualidade da fala em tempo real, e (2) algoritmos de adaptação de parâmetros, como mostrado na Figura 1.3.

Em um trabalho anterior [28], projetamos e implementamos um mecanismo de avaliação objetiva da qualidade da fala baseado no Modelo E [94]. Mais tarde, ele foi melhorado para trabalhar em tempo de execução [196]. Agora, o código correspondente se tornou parte da versão atual da biblioteca *Open Library Telefone Abstraction* (OPAL) [153] e também foi modificado para a biblioteca PJSIP [155]. Tanto na OPAL como na PJSIP, métricas de VoIP são transmitidas entre os terminais através de relatórios RTCP XR (*RTP Control Protocol eXtended Reports*) [56].

Portanto, resta-nos resolver o problema de adaptar parâmetros de terminais VoIP em resposta a mudanças percebidas por realimentação da avaliação da qualidade da fala. Este problema tem sido abordado pela comunidade científica desde a década de 1990, e uma variedade de soluções podem ser encontradas, como detalhado no Capítulo C. Para citar apenas alguns exemplos, Myakotnykh & Thompson [143] e Sfairopoulou et al. [175] propuseram uma solução adaptativa baseada em comutação de codec de acordo com as alterações detectadas na qualidade da fala. Qiao et al. [158] propuseram outro mecanismo de adaptação com base na variação da taxa de transmissão de bits para lidar com restrições de largura de banda em redes de acesso local sem fio (WLANs). Finalmente, a solução adaptativa de Ngamwongwattana [148] depende de ajuste de empacotamento para melhorar a qualidade da fala.

No entanto, existem muitas limitações e questões em aberto para ser investigadas sobre adaptatividade VoIP na camada de aplicação, apesar da intensa pesquisa sobre o tema (ver tabelas C.2–C.6 para uma lista completa). Uma das limitações mais importantes é que a maioria desses trabalhos não se consideram formalmente como uma implementação de um ciclo fechado de realimentação para controle de qualidade da fala. Outras limitações ou questões em aberto estão listadas abaixo:

- Algumas soluções adaptativas usam comutação de codec para melhorar a qualidade da fala durante uma chamada. Elas sustentam sua decisão sobre uma lista restrita de codecs, construída a partir de medidas off-line e estabelece relações de precedência entre seus elementos. Se um novo codec for incluído nesta lista, novos testes off-line devem ser executados para reorganizar a precedência entre as configurações de codificação. Assim, é necessário projetar uma estrutura flexível que explore a solução de compromisso entre codificação, perda de pacotes, atraso e qualidade da fala, com base em parametrização de algoritmos de codificação.

- Adaptação pode ser comparada a uma cirurgia, a qual visa curar um paciente, mas não deve matá-lo. Da mesma forma, os efeitos colaterais da adaptação não deve agravar o problema que ela tenta superar. Por exemplo, não se sabe ainda o que os usuários realmente percebem durante o momento exato em que um codec é trocado visando melhorar a qualidade da fala [143].

- Não há nenhum estudo que avalie o melhor posicionamento dos mecanismos de avaliação de qualidade da fala e de adaptação de parâmetros em um sistema VoIP, visando minimizar o fluxo de troca de mensagens de feedback entre emissor e receptor de endpoints.

- Há também uma falta de trabalhos experimentais que abordem aspectos de implementação de sistemas adaptativos VoIP. Por exemplo, durante o momento em que a adaptação é aplicada, o terminal receptor lida temporariamente com dois fluxos de voz: um codificado com parâmetros antigos, e outro codificado com novos parâmetros. Um requisito básico é que a transição durante a reprodução dos fluxos de voz deve ser tão suave quanto possível, mas tal exigência deve ser expressa por meio de parâmetros processáveis pelo mecanismo de adaptação.

Motivado pelas limitações descritas acima, recorremos aos princípios de software auto-adaptativo [169] para explorar o formalismo destas áreas de pesquisa com a finalidade de conceber serviços VoIP mais robustos, confiáveis e escaláveis. A mudança de foco de aspectos pontuais na solução de compromisso entre codificação e qualidade para o ciclo de controle como o núcleo do mecanismo de adaptação é uma das principais contribuições desta tese. A Figura 1.4 ilustra como nossas contribuições estão posicionados entre três campos de pesquisa: sistemas de VoIP em geral, controle de QoS, e software auto-adaptativo.

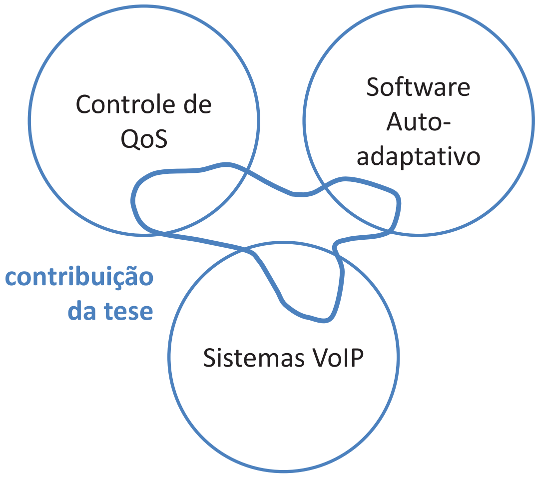


Figure 1.4: Posicionamento conceptual da contribuição desta tese entre três campos de pesquisa.

Portanto, esta tese aborda o problema de como ajustar os parâmetros de codificação em resposta a variações de atraso e perda de pacotes, a fim de otimizar a qualidade da fala. Iremos investigar como os princípios dos sistemas de software auto-adaptativo podem ser aplicados para a concepção e avaliação de terminais VoIP adaptáveis que possam lidar com este tipo de solução de compromisso. A ênfase não é desenvolver novos codecs de áudio, mas sim construir um ciclo de controle no núcleo de terminais emissor e receptor de modo a adaptar as configurações de fluxo de voz de acordo com as condições da rede.

1.3 Objetivos da Tese

O objetivo desta tese é fornecer uma solução em tempo de execução para o gerenciamento de qualidade da fala em chamadas VoIP por meio de princípios de software auto-adaptativo. Os objetivos específicos têm estrita relação com as limitações mencionadas anteriormente a respeito das atuais soluções adaptativas para VoIP, como segue:

1. Rever as soluções atuais para adaptação VoIP, analisando suas realizações e suas limitações, e identificar os pontos de contato com os princípios de software auto-adaptativo explorados ao longo desta tese.

2. Determinar como a comutação de codec afeta a percepção de qualidade da fala do usuário no exato momento da transição (perspectiva de curto prazo), e usar esse conhecimento quando ponderando sobre a melhoria da qualidade da fala em uma perspectiva de longo prazo.

3. Parametrizar a solução de compromisso entre as configurações de codificação, atraso fim a fim, perda de pacotes e de qualidade, de modo a criar dinamicamente uma lista de precedência dos codecs, independente de exaustivos testes *off-line* entre codecs, a fim de

apoiar o mecanismo de adaptação no planejamento de ajustes necessários.

4. Tornar explícito o ciclo de controle de realimentação como núcleo da atividade de adaptação dos terminais VoIP em resposta a mudanças nas condições de rede.

5. Determinar o melhor posicionamento dos mecanismos de avaliação de qualidade da fala e de adaptação de parâmetros, a fim de otimizar a troca de mensagens de realimentação entre terminais emissor e receptor através da rede IP.

6. Avaliar a eficiência da solução proposta em termos de propriedades auto-adaptativas desejadas, tais como precisão, estabilidade, tempo de estabilização de curto, pequeno *overshoot*, robustez e escalabilidade.

Os objetivos específicos supramencionados foram agrupados em três etapas principais, cuja inter-relação é mostrada na Figura 1.5. A primeira etapa do nosso estudo serve de base para as outras duas etapas e corresponde ao primeiro objetivo (Obj1), que envolve a revisão e análise do atual estado-da-arte nas três áreas de pesquisa, a partir do qual emerge a nossa contribuição. A segunda etapa corresponde ao segundo objetivo (Obj2), que envolve testes subjetivos *off-line* para determinar o que os usuários percebem no momento exato da mudança de codec. Finalmente, a terceira etapa corresponde aos quatro últimos objetivos (Obj3, Obj4, Obj5 e Obj6), e engloba tarefas relacionadas com a concepção e avaliação de um sistema VoIP dotado de princípios auto-adaptativos. Note-se que o quarto objetivo – explicitação do ciclo de controle de realimentação – está no centro desta etapa, beneficiando-se de decisões de projeto de outros objetivos na terceira fase e também ditando diretrizes para a realização dos mesmos.

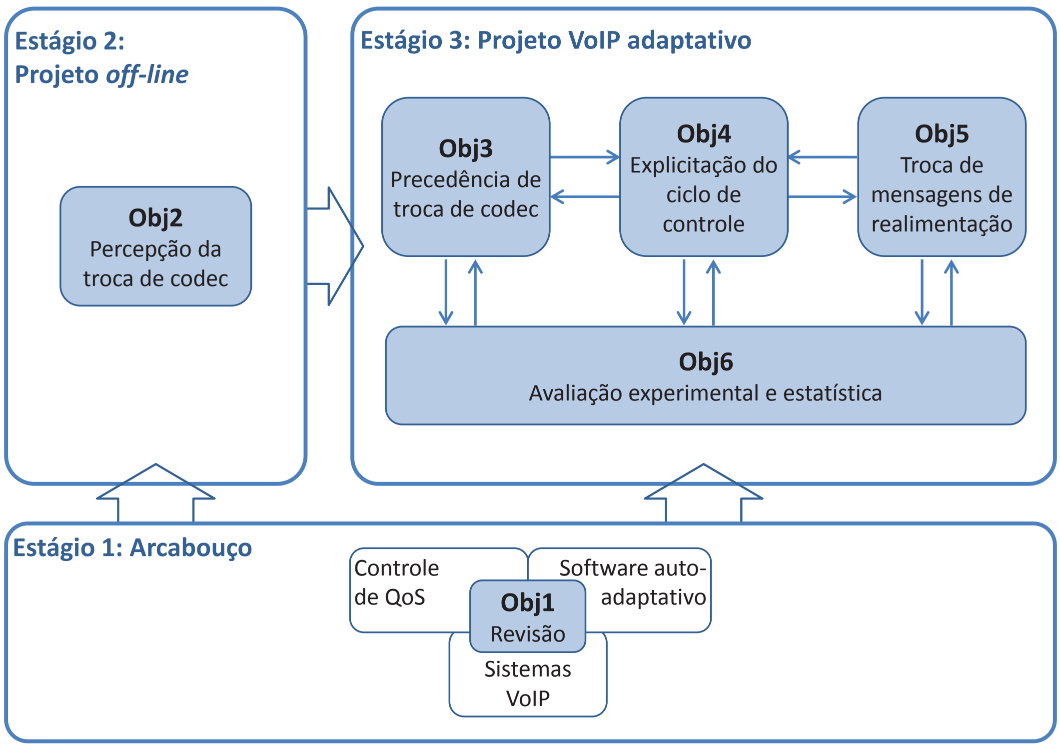


Figure 1.5: Inter-relação entre os objectivos específicos desta tese, que consiste em background (fase 1), off-line de design e avaliação (fase 2) e desenho de arquitetura adaptativa VoIP (etapa 3).

1.4 Visão Geral da Tese

As considerações teóricas, a revisão bibliográfica, a metodologia proposta, a apresentação e análise de resultados e as conclusões estão explicadas nos apêndices. Estes estão organizados conforme os objetivos descritos na Figura 1.5, como se segue.

No Capítulo B, analisamos algumas informações básicas sobre codificadores de voz, avaliação da qualidade da fala, gerenciamento de QoS em VoIP, e os princípios de software auto-adaptativo.

No Capítulo C, fornecemos uma ampla revisão da literatura sobre adaptação da qualidade da fala em chamadas VoIP usando parâmetros disponíveis na camada de aplicação. Também identificamos algumas deficiências e limitações das soluções disponíveis, justificando as contribuições desta tese.

O Capítulo D aborda o problema de determinar como a mudança de codec afeta a percepção do usuário no momento exato da comutação, o segundo objetivo desta tese. Apresentamos o projeto experimental e análise estatística de testes subjetivos.

Os Capítulos E, F and G constituem a investigação principal desta tese. O Capítulo E desenvolve três dos quatro objetivos agrupados na Fase 3 desta tese: (1) parametrização da precedência de comutação de codec, (2) projeto explícito do laço de controle MAPE (monitoramento – análise – planejamento – execução), e (3) análise da eficiência de troca de mensagens de realimentação. O Capítulo F apresenta a metodologia, o ambiente experimental e o planejamento de experimentos que realizamos para avaliar e comparar alguns candidatos de terminais VoIP adaptativos. O Capítulo G completa a investigação, apresentando a análise dos experimentos.

Finalmente, no Capítulo H, resumimos os resultados e apresentamos alguns trabalhos futuros.

Appendix A

Introduction

The aim of science is not to open the door to infinite wisdom,
but to set a limit to infinite error.
Bertold Brecht (1898–1956)

Contents

Applications and services based on Voice over Internet Protocol (VoIP) technology have been experiencing a rapid growth for the last two decades. Initially, cost-efficiency and convergence benefits stimulated such a growth. But now users and service operators are even more concerned in quality of service and dependability of VoIP services, because the increase in scalability has brought together some hurdles, such as configuration complexity and management effort.

Whereas in the Plain Old Telephone Service (POTS) each call uses a dedicated circuit, in IP networks, all calls share the same resources. Because of the uncertainty and nondeterminism inherent in this environment, voice streams are impaired by packet loss, delay, and jitter, which directly affect user perception of speech quality. For years, this problem has been challenging researchers and practitioners, who have been designing and improving QoS control mechanisms for VoIP applications. Such mechanisms aim to make optimum use of network and terminal resources to minimize the effects of network impairments on voice quality. Among the several proposed QoS control mechanisms for VoIP, some of them seek to *adapt* the voice flow or other VoIP-related parameters in accordance with significant changes in the network, end users' preferences, or service providers' requirements.

Adaptive systems in general respond to changes in their internal state or external environment with guidance of an underlying control system. VoIP systems are particularly likely to require dynamic adaptation because of the decentralized control nature of IP networks and the stochastic nature of data packet delivery. Although the existing adaptive solutions for QoS control of VoIP show some performance improvement and exhibit some sort of feedback, they do not provide an explicit focus on the control loop. For instance, most of them do not consider transient and steady-state performance, which poses as a critical obstacle to the validation and verification of such solutions [141].

Besides adaptive control theory, QoS management mechanisms for VoIP share analogy to reliability theory and autonomic communications initiative (AutoComm). Therefore, exploiting and deepening the formalism of these areas could help in designing more robust, reliable and scalable VoIP services.

A.1 Research Scope

VoIP adaptation can be an imprecise term. So, to define our research scope, we should explain what we mean by this expression. According to Salehie & Tahvildari [169], self-adaptive software, in general, aims to adjust various artifacts or attributes in response to changes in the *self* and in the *context* of a software system. In a real-time VoIP system, in particular, we regard the *self* as being the sender and receiver terminals acting as a whole individual, and the *context* as being the underlying network, prone to congestion and link failures.

As it will be discussed in detail in Section B.5.3, when we talk about *adaptation*, this entails at least two questions: *how* and *where*. Regarding to the *how*-question, our concern is focused on adaptive solutions that demand a dynamic interaction between sender and receiver endpoints. Note that this requirement increases complexity, if compared with stand-alone adaptive solutions, such as (1) some types of adaptive packetization [170], which is based solely on the autocorrelation of consecutive chunks of the signal, regardless network conditions or receiver's feedback; and (2) adaptive playout buffer scheduling [77, 114, 163], which does not take into account any message from the sender, although it senses context information from network.

Regarding to the *where*-question, consider the three dimensions where adaptation can be applied in VoIP systems:

1. *Data flow.* A VoIP call is made up of three distinct data flows: *signaling*, *encoded media* and *control*. Signaling refers to those messages exchanged for establishing, maintaining and terminating a call. The sender's voice (media) is encoded and sent in Real-time Transport Protocol (RTP) packets throughout the underlying network toward the destination. Control messages comprise another information such as billing, accounting, and feedback. In this thesis, we focus only on the adaptation of *RTP flow* (encoded media).

2. *Data plane.* As pointed by Chen et al. [34], QoS management mechanisms in data plane can be mapped to the corresponding layers of TCP/IP protocol architecture: *physical*, *medium access*, *internet*, and *application*. Firstly, we did not considered parameters at the physical and medium access layers. The terms *management* and *adaptation* imply the *reading* and *modification* of parameters, but, for standardization reasons, physical and medium access parameters should not be modified along the call. Otherwise, endpoints would not be able to communicate.

   Second, we disregarded also parameters at the network layer. As pointed by Myakotnykh [142], network-layer-based methods often do not solve the problem completely for two reasons: (1) not all equipment supports multiple QoS protocols; and (2) the Internet is a dynamic media, and most of the time technologies cannot react to changing network conditions and manage the quality of every call in real-time.

   Finally, application-layer-based solutions, which were actually explored in this thesis, are designed to minimize the effect of network loss and delay on speech quality and are performed by the sender or receiver of the RTP flow [7]. Also known as end-system-based, such mechanisms include encoding rate control, packetization, forward error control (FEC), playout buffer adjustment, and packet loss concealment (PLC).

3. *Managed aspect.* As pointed by Gokhale & Lu [61], VoIP management can cover two

main aspects: *performance* and *security*. Performance is related to nonmalicious issues, such as high network load or resource failure. Security is related to malicious causes, such as phishing, man-in-the-middle attack, and denial of service, among others. In this thesis, we focus only on the *performance* aspect, in terms of speech quality.

Therefore, in this thesis, we use the term *VoIP adaptation* to address the adjustment of internal parameters of VoIP terminals (application layer) that affect the voice flow, with the aim of improving speech quality, in response to changes in network conditions. We do *not* explore in this thesis adaptive solutions that exclusively manage call signaling or billing information, or that are oriented to security issues of the voice flow, or that concentrate in QoS control at the network layer.

Figure A.1 gives a conceptual idea of the research scope of this thesis. The axes represent the three dimensions where adaptation can be applied in VoIP systems, as explained earlier. The disposition of the labels along the axes does not reflect a mathematical or a ranking meaning, but it only implies that any triple taken from the diagram is also a valid research scope. For example, one can investigate security problems of VoIP signaling at the network layer, or evaluate the performance of billing information delivery through some network architecture.

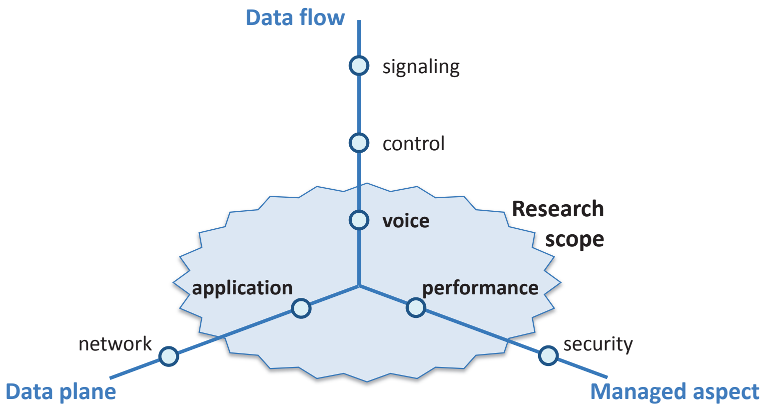


Figure A.1: Identification of the research scope of this thesis amidst the dimensions related to VoIP technology.

A.2 Problem Statement

Figure A.2 shows a traditional VoIP system, without any adaptive behavior. It consists of three main components: a sender (source of VoIP traffic), a receiver (destination point of voice flow) and the underlying IP network. At the sender side, speech is encoded into voice frames, which are grouped into packets (packetization). Then, the voice flow typically goes through the IP network with other types of traffic or through a dedicated VoIP-only network. At the destination side, the jitter buffer eliminates delay variations, and the decoder conceals missing packets before playing out the reconstituted voice flow to the receiver.

According to Myakotnykh [142], in order to make this system adaptive (i.e., to manage speech quality at run-time according to some policy), it is required to design two components: (1) objective mechanisms of real-time speech quality assessment, and (2) parameter adaptation algorithms, as shown in Figure A.3.

In an earlier work [28], we have already designed and implemented a non-real-time mechanism of objective speech quality evaluation based on the E-model [94]. We later improved it to work at run-time [196]. Now, the correspondent code became part of the current version of the Open

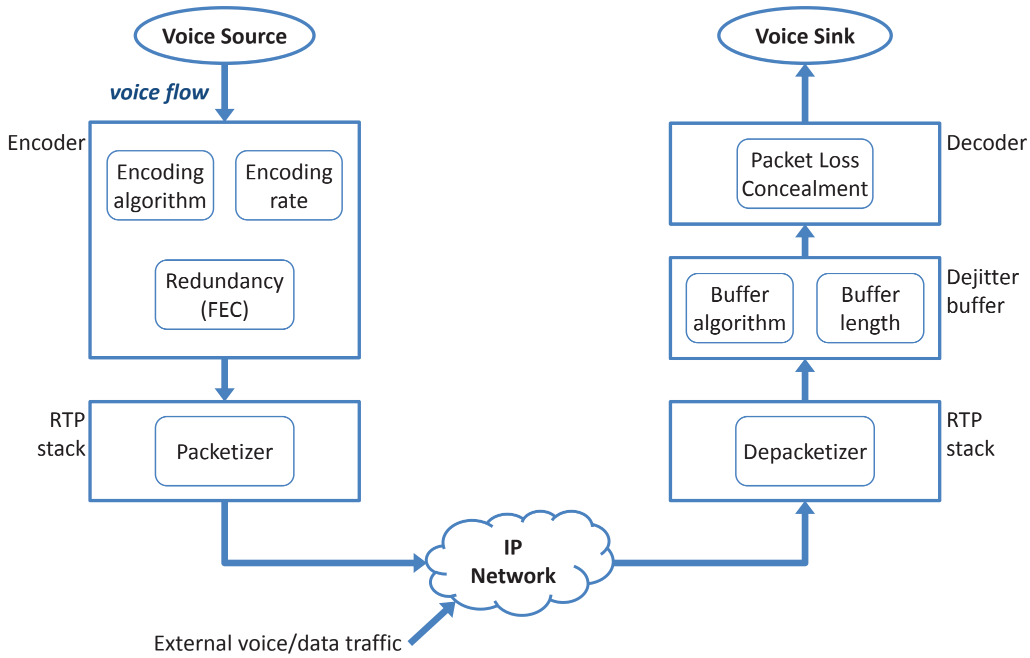


Figure A.2: Example of a nonadaptive VoIP system.

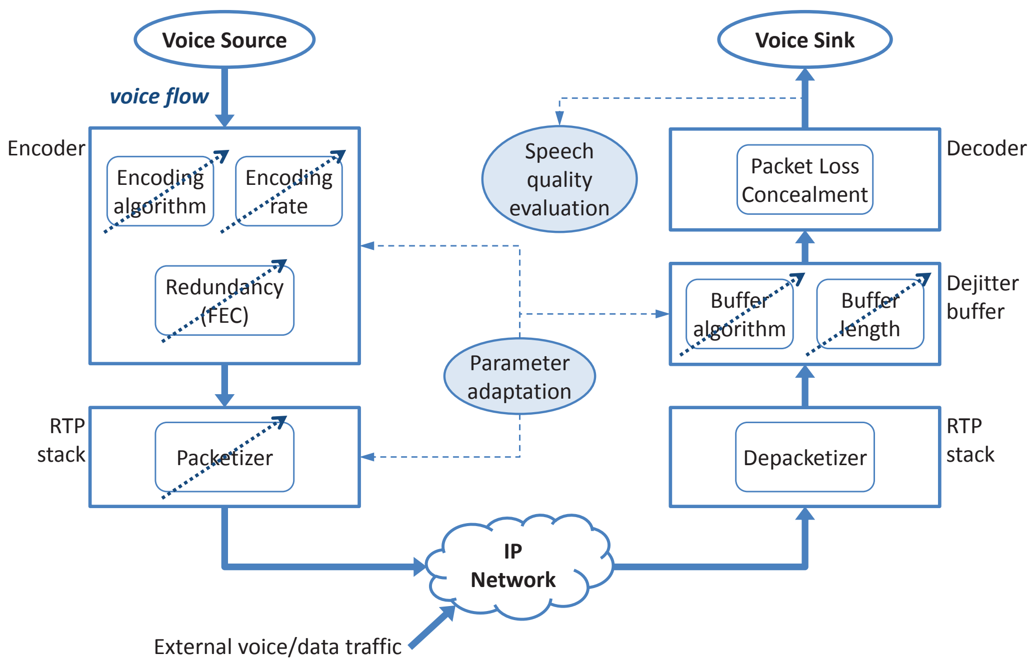


Figure A.3: Example of an adaptive VoIP system, with the inclusion of two components: speech quality evaluation and parameter adaptation.

Phone Abstraction Library (OPAL) [153] and has been submitted to the PJSIP platform [155]. In both OPAL and PJSIP, VoIP metrics are conveyed between endpoints through RTP Control Protocol eXtended Reports (RTCP XR) [56].

Therefore, it is left to us to tackle the problem of adapting parameters of VoIP terminals in response to changes fed back by the mechanism of speech quality assessment. This problem has been addressed by the research community since the 1990s, and a variety of solutions can be found, as detailed in Chapter C. To cite only a few examples, Myakotnykh & Thompson [143] and Sfairopoulou et al. [175] proposed an adaptive solution based on codec switching in accordance to detected changes in speech quality. Qiao et al. [158] proposed another adaptive mechanism based on variation of codec bitrate to cope with bandwidth restrictions in wireless local access networks (WLANs). Finally, Ngamwongwattana's adaptive solution [148] relies on packetization adjustment for improving speech quality.

Yet, there are many limitations and open issues to be investigated about VoIP adaptivity at the application layer, despite the intensive research on this subject (see tables C.2–C.6 for a comprehensive list). One of the most crucial limitations is that the majority of these works do not formally regard themselves as implementing a closed control loop for speech quality control. Other limitations or open issues are listed below:

- Some adaptive solutions use codec switching for improving the speech quality of a call. They underpin their decision on a restrict list of codecs, which is built from off-line measurements and settles precedence relations among its elements. If a new codec should have to be included in this list, off-line tests must be rerun to rearrange the precedence among encoding configurations. Thus, it is necessary to design a flexible framework that explores the trade-offs among encoding, end-to-end delay, packet loss and speech quality, based on standard parametrization of encoding algorithms.

- Adaptation can be compared to a surgery, which aims to heal a patient, but should not kill her. Likewise, the side effects of adaptation should not worsen the problem that it tries to overcome. For example, it is not known yet what users actually hear at the exact moment when the codec is switched in order to improve speech quality [143].

- There is no study that evaluates the best placing of speech quality assessment and parameter adaptation mechanisms in a VoIP system, in order to minimize the exchange flow of feedback messages between sender and receiver endpoints.

- There is also a lack of experimental works that tackle implementation aspects of adaptive VoIP. For example, during the moment in which adaptation is applied, the receiver endpoint temporary deals with two voice flows: one encoded with the old parameters, and another encoded with the new ones. An obvious requisite is that the transition in the playout of such flows should be as smooth as possible, but this requirement must be parameterized into the adaptation mechanism.

Motivated by the previous limitations, we resorted to the principles of self-adaptive software [169] to explore the formalism of these research areas for designing more robust, reliable

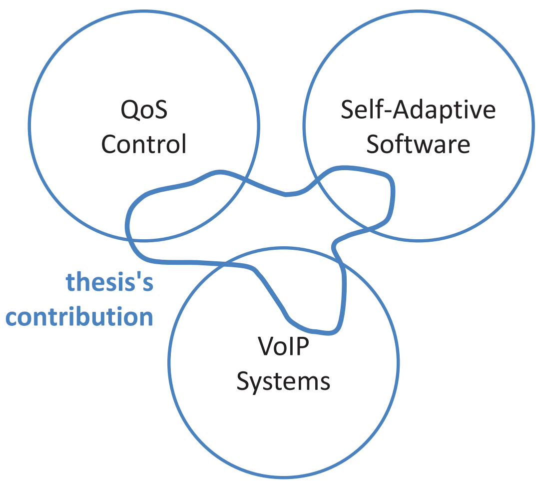


Figure A.4: Conceptual positioning of this thesis' contribution among three research fields.

and scalable VoIP services. The focus shift from punctual aspects of encoding/quality trade-off to the control loop in the core of the adaptation mechanism is one of the major contributions of this thesis. Figure A.4 illustrates how our contributions are positioned among three broad research fields: VoIP systems in general, QoS control, and self-adaptive software.

Therefore, this thesis addresses the problem of how adjust encoding parameters in response to variations in delay and packet loss, in order to optimize speech quality. We are going to investigate how the principles of self-adaptive software systems can be applied for designing and evaluating adaptive VoIP terminals that can deal with such a trade-off. The emphasis is not to develop new audio codecs, but to build a control loop in the core of sender and receiver terminals to adapt voice flow settings according to network conditions.

A.3 Aim and Objectives

The aim of this thesis is to provide a run-time solution for speech quality management of VoIP calls by means of self-adaptive software principles. The specific objectives have strict relation to the mentioned limitations of current adaptive VoIP solutions, as follows:

1. To review the current solutions to VoIP adaptation, analyzing their achievements and their limitations, and identifying the touching points with the self-adaptive software principles explored along this thesis.
2. To determine how codec switching affects user perception of speech quality at the exact moment of the transition (short-term perspective), and use this knowledge when pondering about speech quality improvement at a long-term perspective.
3. To parameterize the trade-offs among encoding settings, end-to-end delay, packet loss and quality, so that a precedence list of codecs can be dynamically created, independent of exhaustive off-line tests among punctual codecs, in order to support the adaptive mechanism for planning the necessary adjustments.
4. To make explicit the feedback control loop as the core of the adaptation activity of the VoIP terminals in response to changes in network conditions.
5. To determine the best placement of speech quality evaluation and adaptation parameter mechanisms, in order to optimize the exchanging of feedback messages between sender and receiver terminals across the IP network.
6. To evaluate the efficiency of the proposed solution in terms of desired self-adaptive properties, such as stability, accuracy, short settling time, small overshoot, robustness, and scalability.

The above-mentioned specific objectives were grouped into three main stages, whose interrelationship is depicted in Figure A.5. The first stage of our study serves as the basis to the other two stages and corresponds to the first objective (Obj1), which involves the review and analysis of current state-of-the-art in the three research fields from which emerges our contribution. The second stage corresponds to the second objective (Obj2), which involves off-line subjective tests for determining what human users perceive in the exact moment of codec switching. Finally, the third stage corresponds to the last four objectives (Obj3, Obj4, Obj5, and Obj6), and encompasses tasks related to the design and evaluation of a VoIP system endowed with self-adaptive principles. Note that the fourth objective – explicitness of the feedback control loop – lies in the core of this stage, benefiting from design decisions of other objectives in the third stage and also dictating guidelines to them.

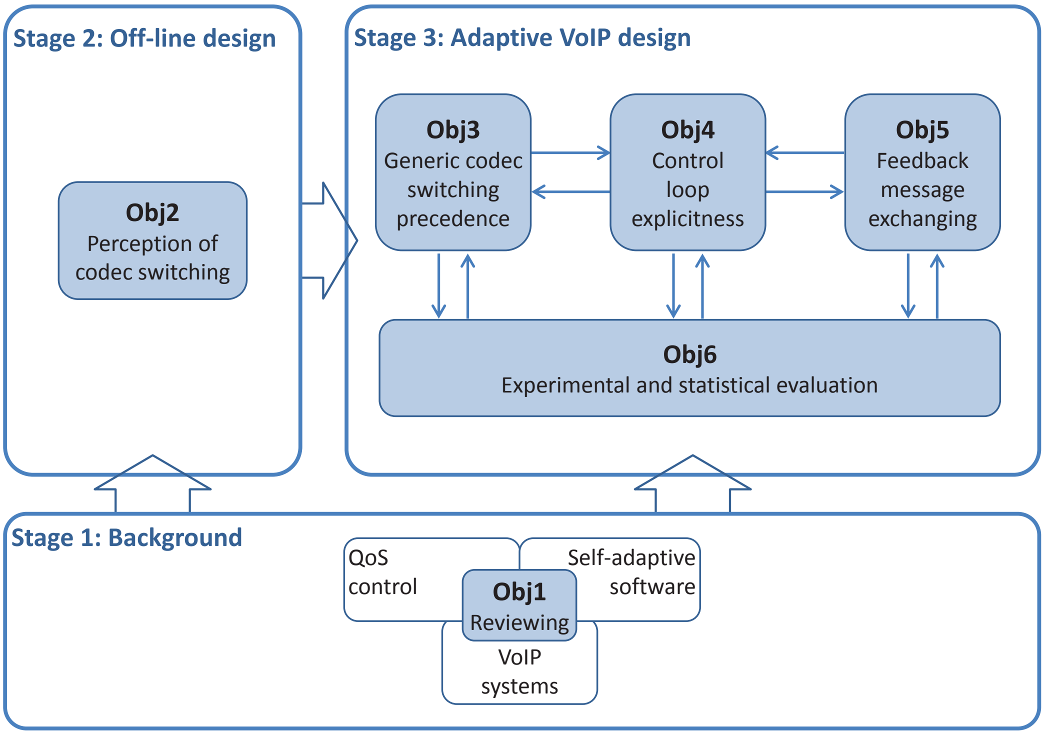


Figure A.5: Interrelationship among the specific objectives of this thesis, which consists of background (stage 1), off-line design and evaluation (stage 2) and design of adaptive VoIP architecture (stage 3).

A.4 Thesis Outline

This thesis is organized in guidance with the objectives depicted in Figure A.5, as follows.

In Chapter B, we review some background information about voice codecs, speech quality evaluation, QoS management of VoIP, and self-adaptive software principles.

In Chapter C, we provide a comprehensive literature review about speech quality adaptation of VoIP calls using parameters available at the application layer. We also identify some shortcomings and limitations of available solutions, justifying the contributions of this thesis.

Chapter D addresses the problem of determining how codec switching affects user perception at the exact moment of codec switching, the second objective of this thesis. It presents the experimental design and the statistical analysis of subjective experiments.

Chapters E, F and G constitute the main investigation of this thesis. Chapter E develops three of the four objectives grouped in Stage 3 of this thesis: (1) parametrization of codec switching precedence, (2) explicit design of the monitoring–analysis–planning–execution (MAPE) control loop, and (3) efficiency analysis of feedback message exchanging. Chapter F presents the methodology, the experimental environment and the design of experiments that we carried out for evaluating and comparing some candidates of adaptive VoIP terminal. Chapter G completes the investigation by presenting the analysis of the experiments.

Finally, in Chapter H, we summary the results and present some future work.

Appendix B

Background

Knowing a great deal is not the same as being smart;
intelligence is not information alone but also judgment,
the manner in which information is collected and used.
Carl Sagan (1934–1996)

Contents

In this chapter, we review some background information about VoIP fundamentals, voice codecs, speech quality evaluation, QoS management in IP networks, and principles of self-adaptive software. This will provide us a common basis of understanding for organizing the current research work on VoIP adaptation mechanisms presented in Chapter C, for assessing human perception during codec switching in Chapter D, for designing the adaptive VoIP system in Chapter E, and for planning the experimental environment in Chapter F.

This chapter is organized as follows. In Section B.1, the basic components of a typical VoIP call are described. Then, Section B.2 gives some important characteristics of voice codecs, such as bitrate, silence suppression, error correction and concealment, among others. Section B.3 describes the most used methods for evaluating speech quality and in which cases they are most suitable to be applied. Next, an overview of QoS management mechanisms of VoIP is given in Section B.4, focusing on those located at the application layer. Finally, Section B.5 gives a general view of self-adaptive software systems.

B.1 Voice Flow in a VoIP System

A typical VoIP call comprises three types of data flow:

1. *Signaling flow.* It enables VoIP endpoints to communicate with each other, set up and tear down calls, and renegotiate session parameters during the call.
2. *Media flow.* It transports the sender's speech content, encoded by some codec algorithm. It is conveyed by the Real-time Transport Protocol (RTP) [171].
3. *Media control flow.* Carried by the Real-Time Control Protocol (RTCP), it provides control and quality monitoring service for RTP transport. Three RTCP reports are of major importance for adaptive purposes: *sender report* (SR) and *receiver report* (RR), which carry the basic transmission and reception statistics from the active sender(s) and receiver(s) in an RTP session; and *extended report* (XR), which carries quality information, such as MOS and delay at dejitter buffer.

Figure B.1 focuses on the media flow path. The sender's voice is captured by a microphone, where it is converted into an analogue signal. Next, this signal is digitized by an A/D converter. The resultant discrete signal is then encoded and compressed by a codec (COmpressor/DECompressor) into voice *frames.* One or more frames are encapsulated into a voice *packet* by adding RTP, UDP, and IP headers. Next, the voice packets are dispatched to the IP network, where they are subjected to some impairment, such as transmission delay, jitter, and loss owing to congestion or transmission errors.

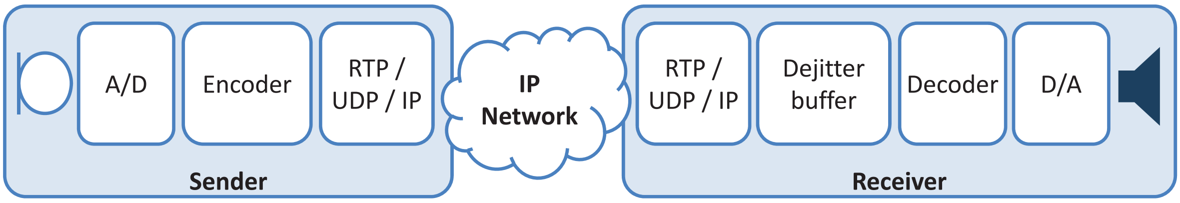


Figure B.1: Logical components of the VoIP transmission path.

At the receiver, the arriving packets are inserted into the *dejitter buffer* (or *playout buffer*), where they are temporarily stored to be played out isochronously. If packets are too late, then they are usually discarded and thus considered as lost by the application. After leaving the dejitter buffer, the speech frames are decoded. If a frame is lost, then the decoder fills the gap by applying some algorithm of Packet Loss Concealment (PLC). Finally, the digital signal is transformed into an acoustic signal, which is played out to the listener.

B.2 Voice Codecs

Codec is an algorithm used for reducing the bitrate required to describe a wide range of signals: video, audio and speech. In this thesis, we are concerned in only *voice codecs* or *speech codecs*, which are implemented in VoIP applications (softphones). The most important attributes of the voice codecs are presented in the following subsections.

B.2.1 Bitrate

The bitrate of a codec is the rate at which the bits that represent the speech input are generated by the encoder, usually expressed in bits per second (bit/s). This rate depends on the compression efficiency of the encoder algorithm and the use of silence suppression. During a conversation, we talk during 35% of the time on average. Therefore, silence suppression is an important feature, which includes three major components:

1. *Voice Activity Detector (VAD).* It is responsible for switching the codec from *active speech mode* to *background noise transmission mode.* If improperly designed, VAD techniques may clip parts of active speech periods, such as beginning and ending of sentences.

2. *Discontinuous Transmission (DTX).* It refers to the ability of a codec to stop transmitting frames when the VAD has detected a silence period.

3. *Comfort Noise Generator (CNG).* Some codecs do not stop transmission completely when DTX mechanism is on. Instead they send small packets containing parameters that allow the receiver to regenerate the background noise at the source.

B.2.2 Codec Delay

Coding algorithms generally require some amount of samples to perform good compression [42]. For real-time speech, this requires longer delays, which affects the quality of the conversation. Table B.1 (p. 35) lists the delay that some codecs take to generate a voice frame (T_{frame}). In VoIP systems, a number of voice frames (N_{frame}) are aggregated into one single RTP packet, before being sent through the network. Such a procedure is called *packetization.* Furthermore, some encoders need to accumulate additional signal samples before encoding the current frame. The time required for doing this is called look-ahead delay (T_{look}). Hence, the minimum delay owing to codec-related processing in VoIP is given by the following expression [88]:

$$T_{\text{codec}} = 2 \times (N_{\text{frame}} + T_{\text{frame}}) + T_{\text{look}}. \tag{B.1}$$

B.2.3 Robustness to Channel Imperfections

Bit errors and packet loss are inherent to communication infrastructures. The performance of a speech coder over such channels can efficiently be improved by (1) adding redundancy at the encoder and using the redundancy at the decoder to recover lost packets, or by (2) producing a replacement for a lost packet at the decoder, similar to the original one [179, 152]. Sender- and receiver-based repair are complementary techniques, and applications can use both methods to achieve the best possible performance. They are outlined below, but a throughout discussion can be found in [152].

1. *Sender-based repair.* These techniques, also known as Forward Error Correction (FEC), rely on adding redundant data to a stream, from which the contents of lost packets may be recovered in a bit-exact form at the receiver side. In Section C.2.2 (p. 60), we provide more details about these techniques.

2. *Receiver-based repair.* These techniques, also known as Packet Loss Concealment (PLC), use only the information of previously received packets to replace the missing packets [74]. They take advantage that speech signals exhibit large amounts of short-term self-similarity. The replacement can be done by simply inserting zeros, repeating signals, or by some more sophisticated method utilizing features of the speech signal [179].

Table B.1: Characteristics of the most well-known voice codecs.

Codec	**Bitrate** (kbit/s)	$\mathbf{T}_{\text{frame}}$ (ms)	$\mathbf{T}_{\text{look}}$ (ms)	**Compression type**	**Licence**
Narrowband codecs					
G.711	64	0.125[a]	—	PCM	Free
G.723.1	6.3	30	7.5	MP-MLQ	Proprietary
G.723.1	5.3	30	7.5	ACELP	Proprietary
G.726	16, 24, 32, 40	0.125[a]	—	ADPCM	Free
G.728	16	0.625[a]	—	LD-CELP	Proprietary
G.729	8	10	5	CS-ACELP	Proprietary
G.729A	8	10	5	CS-ACELP	Proprietary
G.729D	6.4	10	5	CS-ACELP	Proprietary
G.729E	11.8	10	5	CS-ACELP LPC	Proprietary
GSM-FR (6.10)	13	20	—	RPE-LTP	Free
GSM-HR (6.20)	5.6	20	4.4	VSELP	Proprietary
GSM-EFR (6.60)	12.2	20	—	ACELP	Proprietary
AMR-NB	4.75–12.2	20	5	ACELP	Proprietary
Speex (NB)	2.15–24.6	20	10	CELP	Free (open-source)
iLBC	13.33	30	10	LPC	Free
iLBC	15.2	20	5	LPC	Free
BV16	16	5	—	TSNFC	Proprietary
Wideband codecs					
G.722	48, 56, 64	0.0625[a]	1.5	SB-ADPCM	Free
G.722.1 (Siren 7)	16, 24, 32	20	20	MLT	Proprietary
G.722.1 (Siren 14)	24, 32, 48[b]	20	20	MLT	Proprietary
	48, 64, 96[c]	20	20	MLT	Proprietary
G.719 (Siren 22)	32, 48, 64[b]	20	20	MLT	Proprietary
	64, 96, 128[c]	20	20	MLT	Proprietary
G.722.2 (AMR-WB)	6.6–23.85	20	5	ACELP	Proprietary
Speex (WB)	4–44.2	20	14	CELP	Free (open-source)
iSAC	10–32	30–60	3	MLT	Proprietary
SVOPC	20	20	10	LPC	Proprietary
SILK	5–20	20–100	3	LTP	Free
BV32	32	5	—	TSNFC	Proprietary

[a] Usually, implementations of such codecs generate packets with a minimum length of 10ms.
[b] Mono audio.
[c] Stereo audio.

B.2.4 Classification

Figure B.2 outlines the three most-used criteria for classifying codecs: sampling frequency, packet rate, and binary representation. Regarding to the *sampling frequency* of speaker's voice, codecs can be classified into three groups:

1. *Narrowband.* It comprises the codecs that operate on audio signals filtered to a frequency range from 300 Hz to 3400 Hz and sampled at 8 kHz.

2. *Wideband (or broadband).* It comprises the codecs that operate on audio signals filtered to a wider frequency range, producing more natural, comfortable and intelligible speech [179]. Typical sampling frequencies are 16 kHz and 24 kHz.

3. *Multimode.* It comprises the codecs that operate on both narrowband or wideband.

Regarding to the *packet rate*, codecs can be classified into two groups:

1. *Constant Bitrate (CBR).* Codecs of this type send a bit stream of constant rate, independently of the voice input.

2. *Variable Bitrate (VBR).* They choose the most appropriate encoding bitrate from a pre-

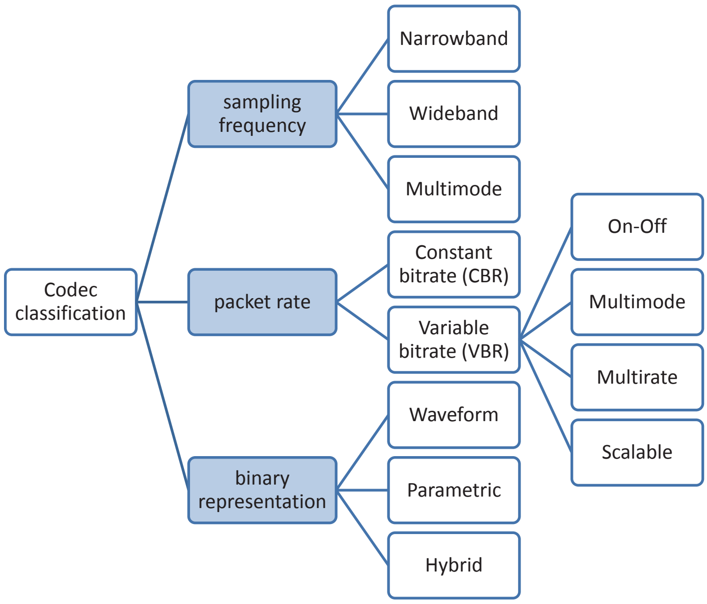


Figure B.2: Classification of codecs according to sampling frequency, packet rate and binary representation of speech signal.

defined set list. This choice can be driven either by the *source*, using the phonetic characteristics of speech content; or by the *network*, using some congestion-related parameter (e.g., delay, packet loss). Beritelli et al. [13] distinguish four different types:

(a) *ON/OFF.* It combines a CBR codec and a VAD. The transmission is discontinuous, featuring talkspurt periods (ON) and periods of silence or background noise (OFF).

(b) *Multimode.* It adapts the coding model to the local signal features, further classifying the ON and OFF classes into relative phonetic subclasses (i.e., voiced/unvoiced (ON), stationary/transient noise (OFF)).

(c) *Multirate.* It is network-driven because it is constituted by a number of CBR coding schemes, each with a different bitrate. Depending on the network conditions, it changes the coding scheme to present an output rate that is never higher than the available bandwidth.

(d) *Scalable.* The data packet obtained by coding each single frame comprises a low bitrate core to which enhancement stages are added to increase the quality of the reconstructed signal. Generally, both multirate and scalable coding techniques also adopt a VAD for silence suppression.

Finally, regarding to the *binary representation* of the speech signal, codecs can be classified into three groups:

1. *Waveform.* They attempt to code the exact shape of the speech signal waveform, without considering in detail the nature of human speech production and speech perception.

2. *Parametric.* Instead of transmitting a direct description of the voice signal, it produces some parameters describing how the signal has been generated, considering the nature of human speech production and perception. The quality will be low but, signals can be transmitted with a very low bitrate.

3. *Hybrid.* They send a number of parameters as well as some waveform-coded information, providing a reasonable compromise between voice quality and coding efficiency. Com-

pared with vocoders, hybrid codecs deliver better quality, and a wideband version of hybrid codecs can exceed waveform-based codec quality. They are largely used in today's VoIP systems.

Generally, VoIP softphones use multiple codecs. However, it is difficult to get to a firm conclusion on which codecs are the most suitable for all kinds of scenarios. In this sense, several aspects should be considered, such as quality, compression, packet loss robustness, frame size and end-to-end delays, processing, memory requirements, backward compatibility, ability to cater to multiple deployments, and acceptance in the market [144]. The most commonly used codecs are listed on Table B.1.

B.3 Speech Quality Measurement

As pointed by Shannon [176], the fundamental problem of communication is reproducing at the receiver either exactly or approximately a message sent by the source. In VoIP-based systems, because of the lossy compression performed by speech codecs, the receiver will always obtain an approximate message of what was spoken at the source. Consequently, evaluation methods are needed for determining the quality of the received message. Particularly, this thesis focuses on *speech quality* alone, that is, on the RTP flow. We will not account for other aspects that make up a satisfactory call, such as delay to get the dial tone, connection success, and service availability.

As depicted in Figure B.3, speech quality evaluation can be performed by *subjective methods*, which involve humans, or *objective methods*, which estimate quality degradation based on suitable algorithms. Objective tests can be further divided into two categories. *Perceptual* methods involve knowledge on human auditory system to evaluate the voice signal quality, and *computational* methods convert speech signal and packet transmission characteristics into numeric parameters, computing them to predict voice quality.

All speech quality evaluation methods make some *comparison* in order to produce their outputs, even though sometimes the comparison is not explicit [116]. In subjective methods, listeners compare the test signal with a reference that they have in their minds, because they are familiar with the natural sound of human voice [59]. Objective perceptual methods are clearly

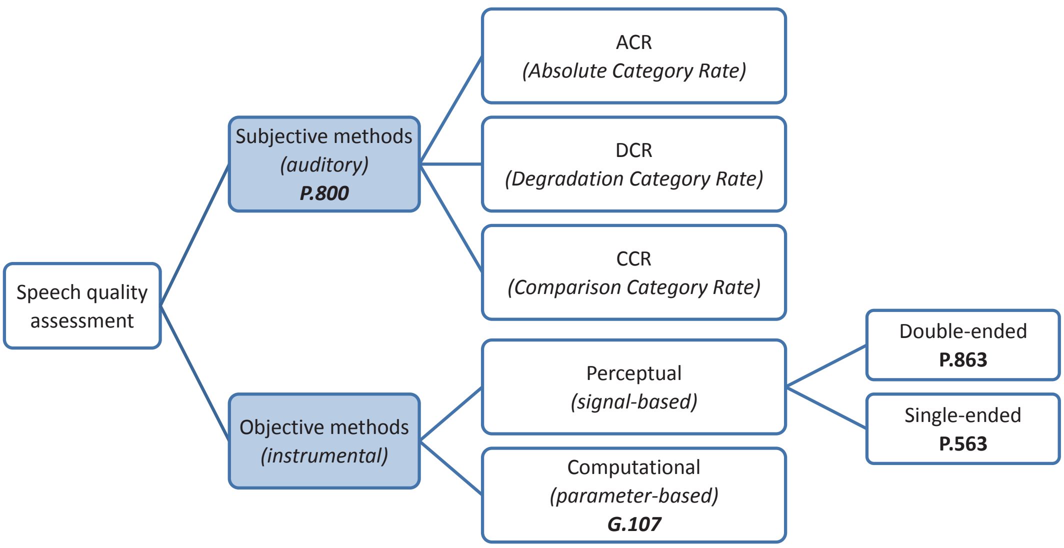


Figure B.3: Classification of speech quality assessment methods.

Table B.2: ITU-T terminology on speech quality.

Methodology	Listening-only test		Conversational test	
	Notation	**Standard**	**Notation**	**Standard**
Subjective	MOS_{LQS}	P.800	MOS_{CQS}	P.800
Objective	MOS_{LQO}	P.863 POLQA	MOS_{CQO}	P.562
Estimated	MOS_{LQE}	Not defined	MOS_{CQE}	G.107 E-model

comparative by definition, as voice signals are contrasted against some knowledge on auditory system characteristics. Finally, in objective computational methods, the comparison is more subtle, as the impairment factor parameters are derived from prior subjective and perceptual results, used for calibrating the model.

Speech quality is usually expressed on the Mean Opinion Score (MOS) scale, which ranges from 1 to 5, where 1 represents the lowest perceived quality, while 5 is the highest perceived quality. MOS values above 3.6 are considered acceptable for toll-quality [94]. The ITU-T Recommendation P.800.1 [93] differentiates MOS scores derived from different methods, adding a subscript identifier, as presented in Table B.2. In the reminder of this section, the main representative standards of each speech quality assessment method are presented.

B.3.1 Subjective Methods

Subjective (or *auditory*) methods attempt to quantify the user perception of speech quality. They rely on the opinion of a panel of listeners, who are asked to rate the quality of sentences read aloud by both male and female speakers over the communication system being tested [107].

The subjective methods can be grouped into two classes: *listening* and *conversational* tests. Listening tests involve listeners that passively rate the quality of short-duration speech signals that they have just heard. Conversational tests, in turn, are interactive, and listeners are asked to rate the quality of a call based on the listening quality and on their ability to talk during the call. Conversational tests account for additional factors such as echo and delay [55].

The ITU-T Recommendation P.800 [85] presents three main methodologies for subjective speech quality evaluation:

1. *Absolute Category Rating (ACR).* It is the most-used subjective method, whose output is the well-known Mean Opinion Score (MOS). Note that some works wrongly takes the MOS – the outcome of the tests – as being the ACR methodology itself.

   In ACR subjective tests, listeners are asked to rate the *absolute* quality of speech samples, without knowing the reference sample. The individual opinion of each listener is expressed as a single number on the MOS scale (Table B.3). The mean of all scores thus obtained represents the MOS of the audio sample under test.

Table B.3: Grades in the MOS scale.

Quality of speech	Grade
Excellent	5
Good	4
Fair	3
Poor	2
Bad	1

Table B.4: Grades in the DMOS scale.

Degradation	Grade
Inaudible	5
Audible, but not annoying	4
Slightly annoying	3
Annoying	2
Very annoying	1

Table B.5: Grades in the CMOS scale.

Category	Grade
Much better	3
Better	2
Slightly better	1
About the same	0
Slightly worse	-1
Worse	-2
Much worse	-3

The involvement of human listeners makes ACR tests expensive and time consuming. Moreover, they are not suitable to real-time monitoring [123]. This has made objective methods very attractive for meeting the demands for voice quality measurement in communications networks.

2. *Degradation Category Rating (DCR).* It is used when good-quality speech samples should be compared. Subjects are presented with a pair of samples for each item. The reference is clearly identified and subjects are asked to rate the test item against the reference on a degradation and annoyance rating scale, the *Degradation Mean Opinion Score (DMOS)*, presented in Table B.4.

   Chapter D is dedicated to explain this method and how we are using it for evaluating user's perception during the moment of codec switching, one of the contributions of this thesis (Figure A.5, p. 31).

3. *Comparison Category Rating (CCR).* It is similar to DCR, but the order of the reference sample and the evaluated coder sample is chosen at random: this method is interesting mostly for speech enhancement systems. The result is a *Comparison Mean Opinion Score (CMOS)*, presented in Table B.5. This method eliminates the ordering restriction of the DCR test at the expense of doubling the total number of trials. For half of the trials (chosen at random), the unprocessed speech is presented first, followed by the processed speech. For the other half, the order is reversed.

B.3.2 Objective Methods

In order to reduce the necessity for time-consuming and costly tests to measure speech quality, much effort has been spent on developing alternative, objective methods, also known as *instrumental methods*. They aim to estimate quality scores correlated with the ones obtained from subjective listening experiments.

Objective methods are classified into two major groups: *perceptual* (or signal-based) and *computational* (or parameter-based). The first group involves knowledge on auditory processing to evaluate the voice signal quality, by pondering the distortions that could affect more the human auditory system. Signal-based approaches can be further classified either as *double-ended* or *single-ended*, as described latter in this section. The computational methods convert the distortion factors affecting the quality of the voice stream into parameters, which are computed in a psychoacoustic scale, providing a quality score.

Double-Ended Perceptual Measurement

Double-ended perceptual methods depend on some form of distance metric between the input (original) and output (degraded) speech signals to estimate subjective quality (see Figure B.4(a)). Research about these methods dates back to the 1980s [55].

The Perceptual Objective Listening Quality Analysis (POLQA) [95] is the current state-of-the-art of double-ended perceptual measurement algorithm. It has withdrawn the well-known Perceptual Evaluation of Speech Quality (PESQ) [87].

Double-ended methods compare the original voice signal at the sender side with the degraded voice signal at the receiver side. The differences are pondered considering the perceptual characteristics of the human auditory system and converted to the MOS scale.

Double-ended methods are suitable for quality benchmarking and intrusive monitoring [189]. However, Hoene [72] points three main drawbacks: (1) they are not able to predict speech

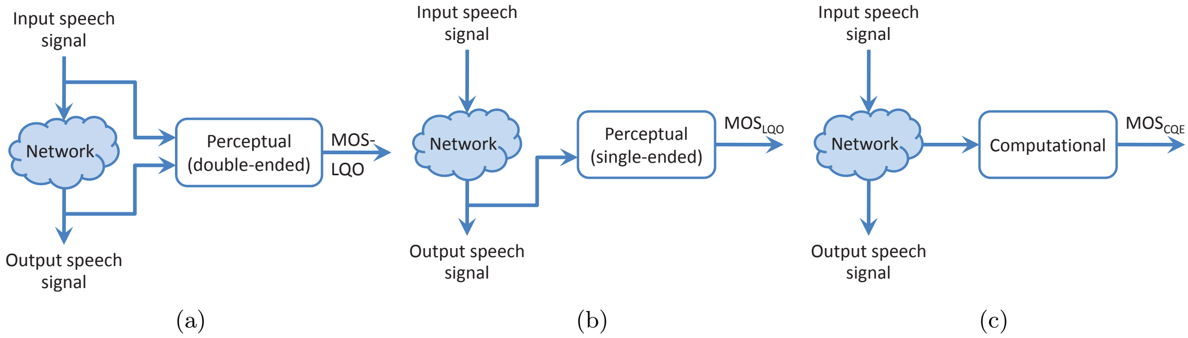


Figure B.4: Block diagram of (a) double-ended (e.g., PESQ) and (b) single-ended (e.g., P.563) perceptual objective measurement, and (c) computational measurement (e.g., E-model).

quality at run-time, owing to the difficulty of having access to both original and degraded signals; (2) they do not take into account end-to-end delay; and (3) they are proprietary.

Single-Ended Perceptual Measurement

Single-ended perceptual methods do not require access to a clean reference signal (see Figure B.4(b)) and commonly rely on models of normative speech behavior. The ITU-T Recommendation P.563 [90] represents the current state-of-the-art of single-ended standard algorithm for traditional telephony applications. A detailed description of the P.563 method is available in [90] and [121]. Recent research, however, has suggested that P.563 performance is compromised for VoIP applications [48, 55]. So, it is not used in this work.

Computational Measurement

The computational methods (or *parameter-based* methods) use network parameters for estimating listening and conversational quality (see Figure B.4(c)). Their basic assumption is that transmission impairments can be transformed into psychological impairment factors, which are additive in the psychoacoustic domain. Their most representative, the E-model [94], describes several parametric models of specific network impairments and their interaction with subjective quality. It is further described in Section B.3.3.

B.3.3 E-model

The ETSI model – known as E-model – was first proposed by the ETSI Technical Report 250 [52]. Nowadays it has been updated by the ITU-T Recommendation G.107 [94]. Although it was originally designed for transmission planning purposes only, it was extended by Clark [40] for speech quality evaluation of live networks. We revised and corrected this extended version in an earlier work [27].

The E-model is not a perfect tool to calculate *absolute* quality level, but it is acceptable for measuring *variations* in quality. Detailed investigations conducted by Takahashi et al. [189] concluded that the E-model can be used for tracking changes in quality, as needed for adaptive management of speech quality.

The output of E-model is the rating factor R, which ranges from 0 to 100, convertible into MOS. It is given by the following expression:

$$R = 93.2 - I_{\mathrm{d}}(codec, delay) - I_{\mathrm{e,eff}}(codec, loss, PLB), \tag{B.2}$$

where the *delay impairment factor* I_{d} reflects all impairments delayed with respect to the transmitted speech signal, and the *effective equipment impairment factor* $I_{\mathrm{e,eff}}$ reflects the impairments associated with codec compression, packet loss rate and packet loss behavior (PLB).

The value of I_{d} is a function of the absolute end-to-end delay, T_{a}, which is given by

$$Ta = T_{\mathrm{net}} + T_{\mathrm{buffer}} + T_{\mathrm{codec}}, \tag{B.3}$$

where T_{net} is the network delay, T_{buffer} is the delay inserted by the dejitter buffer, and T_{codec} is the codec delay given by Equation B.1.

The factor $I_{\mathrm{e,eff}}$, in turn, is determined by the following expression [94]:

$$I_{\mathrm{e,eff}} = I_{\mathrm{e}} + (95 - I_{\mathrm{e}}) \cdot \frac{Ppl}{\frac{Ppl}{BurstR} + Bpl}, \tag{B.4}$$

where Ppl is the packet loss rate as perceived by the receiver application, which includes both network packet loss (Ppl_{net}) and dejitter buffer discard (Ppl_{buffer}); I_{e} is a tabulated value related to the impairment introduced by the codec compression algorithm at 0% packet loss; and Bpl is the codec-specific robustness factor to packet loss. $BurstR$ is the burst ratio, which expresses the PLB.

Note that packet loss is related to network congestion, so it *may extend over several packets*, suggesting a *dependency* between individual loss events [160]. Furthermore, Gros & Chateau [66] have found that subjects react more slowly on improvements than on degradations of quality in the presence of packet loss while evaluating instantaneous quality. Furthermore, quality variations have more impact on subjects' overall judgments when they occur at the middle or at the end of speech sequences. These findings, referred to as *recency effect*, were modeled along with *packet loss behavior* to produce the extended version of the E-model [40], more suitable for an accurate real-time monitoring assessment of speech quality.

Figure B.5 identifies in a general VoIP architecture the measurement points from which the input parameters of the E-model can be taken. Details of how these parameters are computed by the extended E-model can be found in the work of Carvalho et al. [27].

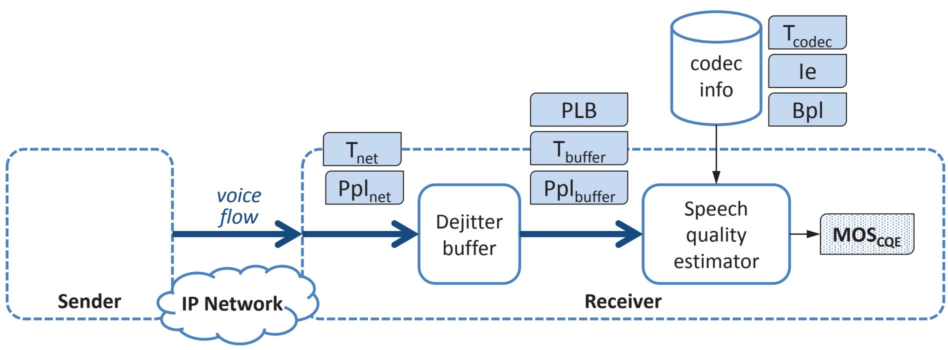


Figure B.5: E-model input parameters at their respective measurement points.

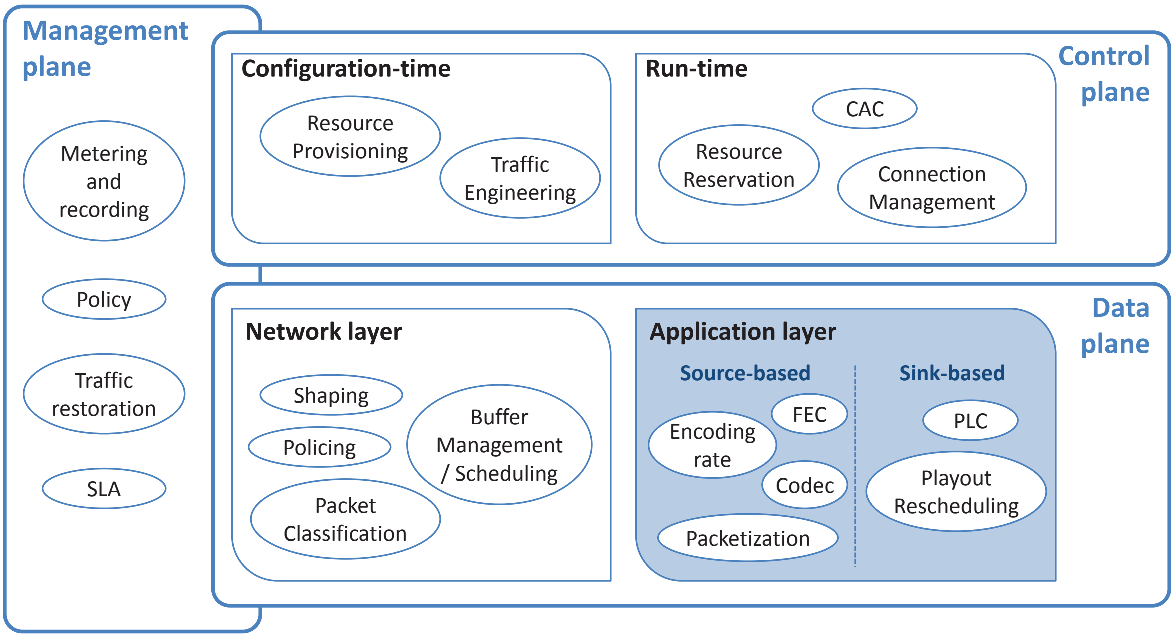


Figure B.6: QoS management mechanisms for VoIP applications.

B.4 Overview of QoS Management of VoIP

The term QoS is used throughout the literature with many meanings, ranging from *user's perception* of the service to a *set of connection parameters* necessary to achieve a particular service quality. Here, we use *QoS* to refer to "a set of service requirements to be met by the network while transporting a flow" [43], and *QoE* to describe resulting service features as perceived by the customer, such as MOS values.

Speech quality evaluation gives a snapshot of QoS and QoE problems in a VoIP system, but it does not offer the solution itself. In this sense, several mechanisms have been developed for managing the QoS and QoE of VoIP calls at run-time. Chen et al. [34] classify these mechanisms into two planes: *control* and *data*. The ITU-T Recommendation Y.1291 [92] adds a third plane: *management*. Bai & Ito [7] subdivide the mechanisms supported by the application layer into *source-based* and *sink-based*. These three views are merged in Figure B.6.

Application-layer mechanisms exploit VoIP-specific characteristics for improving speech quality. Some examples include *codec switching*, *encoding rate control* and *packetization adjustment*, which adapt application's bandwidth demand; *forward error correction (FEC)* and *packet loss concealment (PLC)*, which adapt the robustness against network packet loss; and *playout buffer rescheduling*, which adapts the trade-off between end-to-end delay and packet discard.

Sink-based adaptation mechanisms have a quick response-time, but they only react to network problems. In contrast, *source-based* ones can proactively change the bandwidth demand over the network, but they require a *feedback* message to trigger or stop their operation, which makes their reaction time slower.

In this thesis, we are especially interested in architectures that implemented control mechanisms in the *application layer* and direct handle the *RTP flow*. As already pointed in Section A.1, the scope of this thesis does not include control of signaling and billing information, which also belong to the data plane of the QoS framework. Naturally, most of the surveyed works do not regard themselves as being part of this architectural QoS framework. Anyway, we have selected those works that *place* their solutions at the application layer.

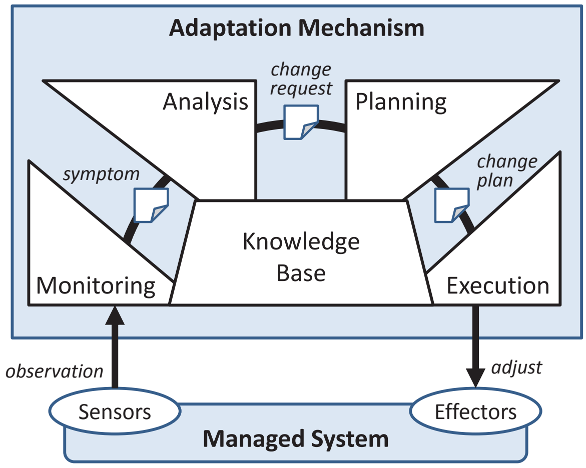


Figure B.7: The feedback loop.

B.5 Self-Adaptive Software

In the previous section, QoS management mechanisms for VoIP where organized into functional planes: management, control, and data. Those mechanisms could be also classified according to their *automation maturity*, that is, how autonomously they may adapt the managed system's behavior in response to changes in the environment.

In this section, we present some basic concepts that will guide our further discussion on Chapter C, where we review the literature about QoS management of VoIP from the Self-Adaptive Software perspective.

B.5.1 The Feedback Control Loop

Adaptivity is not a Boolean property. It may be addressed at diverse points in a system and at different human-interference levels [140]. Furthermore, it uses to come in various guises, yet not explicitly. Anyway, what self-adaptive systems have in common is that design decisions are moved towards run-time to control dynamic behavior, so that they reason about their state (the *self*) and their environment (the *context*) [169]. This implies that a *feedback loop* lies at the heart of self-adaptive systems [140].

A feedback loop, also known as *adaptation* or *autonomic* loop, typically involves four key activities: *monitoring*, *analysis*, *planning* and *execution* (MAPE). These activities are also referred to as collect, analyze, decide, and act, respectively [140].

As depicted in Figure B.7, sensors collect data from the managed system. The feedback cycle starts with the *monitoring* of relevant data that reflect the current state of the system. Next, the system *analyzes* the collected data, structuring and reasoning about the raw data. Then, decisions must be *planned* about how to adapt the system to reach a desirable state. Finally, to implement the decision, the system must *execute* it by means of available effectors. Central to this loop, there is a *knowledge base* that keeps the necessary information about the managed entities and their operations. This reference model is also referred to as MAPE-K by the autonomic computing community [81].

In the context of a VoIP call, the monitoring activity is responsible for collecting relevant data that affects speech quality, such as delay, packet loss, and codec type. Next, the collected data is analyzed to identify unfavorable call conditions and their possible causes. Then, a decision action is planned depending on past actions, network conditions and call configuration. Finally, the planned action is executed, which can entail changes in softphone configuration at sender and receiver endpoints, or cross-layer interactions among components in the voice path. A new control loop restarts, considering new conditions of the call and the results of past executions.

B.5.2 Types of Adaptation

From the works of Shaw [177] and McKinley et al. [131], it can be distinguished two general types of software adaptation:

1. *Parameter adaptation.* It modifies program variables that determine system's behavior. It can be further divided into two subtypes:
   (a) *ON/OFF control.* It simply turns the controllable parameter OFF and ON. Its main advantage is that there is no set of values to choose. However, it can make the controllable parameter to oscillate too often. To alleviate this drawback, it is common to introduce either a dead zone or hysteresis [4].
   (b) *Proportional control.* It tunes the controllable variable in proportion to the degree that the system diverges from the ideal point.
2. *Compositional adaptation.* Also referred to as *reconfigurable* or *structural* adaptation, it exchanges algorithmic or structural components with others in response to changes on system's environment. Hence, an application can adopt new algorithms to address concerns that were unforeseen during system's development. This flexibility supports more than simple tuning of program variables or strategy selection.

For example, Myakotnykh & Thompson [143] proposed an adaptive mechanism for VoIP calls that adjusts, in run-time, the codec used for encoding speech signal, in response to changes in the MOS value. This approach can be considered as being *parameter adaptation*, where the *encoding scheme* is the *parameter* modified by the adaptive solution. On the other hand, if the VoIP terminals had gotten new codec implementations from a trusted repository and reconfigured the switching heuristic, then this would be considered as being a compositional adaptation. To the best of our knowledge, there is no compositional adaptation solution for improving speech quality in VoIP calls implemented at the application layer. Nevertheless, the scope of this thesis does not include demonstrating the feasibility of such a solution.

B.5.3 Adaptation Requirements

The requirements of self-adaptive software can be classified into four logical groups of questions [133]:

1. *Where (object of change).* These questions set out to locate the problem to be solved and the supporting mechanisms for recovery.
2. *When (temporal properties).* These questions address temporal properties such as when a change should be made, in which frequency that change should be taken, or whether change history should be controlled.
3. *What (system properties).* These questions identify which attributes of the system can be changed through adaptive actions.
4. *How (change support).* These questions address how the adjustable attributes can be changed; which adaptive action(s) should be applied; and how the order of changes, their costs and aftereffects are considered for deciding the next action.

These groups of questions are used for eliciting adaptation requirements during *developing* and *operation* phases of the software life cycle. In this thesis, we use these questions for guiding our literature review, identifying the self-adaptive characteristics hidden in the studied architectures. A more detailed discussion about self-adaptive software is provided by Salehie & Tahvildari [169] and Mens et al. [133].

In the next chapter, we shall review the literature about speech quality adaptation at the *application layer*, identifying how fully the four activities of the generic control loop are implemented. Naturally, most of the reviewed works do not regard themselves as implementing a control loop. So, one of the contributions of this thesis resides in reinterpreting those works from the self-adaptive software perspective, classifying the existing approaches and identifying those that remains as open problems.

B.6 Summary

In this chapter, we reviewed some background information about the VoIP components that handles the voice flow during a typical VoIP call (Section B.1), voice codecs and their influence on speech quality perception (Section B.2), and speech quality evaluation methods (Section B.3), such as the Degradation Category Rating – DCR (Section B.3.1), and the E-model, (Section B.3.3). Finally, we provided an overview of QoS management mechanisms for VoIP (Section B.4), and self-adaptive software systems (Section B.5).

Appendix C

Literature Review

It is not the strongest of the species that survives,
nor the most intelligent.
It is the one that is the most adaptable to change.
Charles Galton Darwin (1887–1962)

Contents

In this chapter, we review the works in the literature that used some sort of adaptation for providing QoS control for VoIP applications. As far as adaptation may take various forms, we should elect a criteria for classifying a given work as presenting an adaptive mechanism or not. Thus, our most important guideline is that the system structure should exhibit at least one closed feedback loop.

In Section C.1, we propose a classification system for VoIP architectures that somehow implemented the four activities of the control loop (Figure B.7), based on the considerations of Section B.5 about self-adaptive software. It will serve as a guideline to the literature review about mechanisms of speech quality adaptation at the application layer, presented in sections C.2 and C.3.

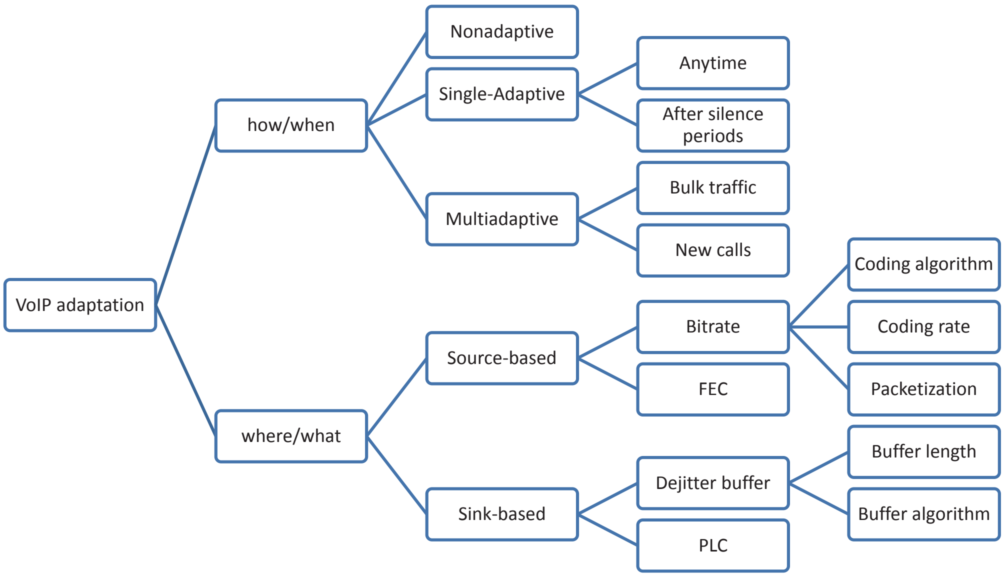


Figure C.1: Classification of approaches to speech quality adaptation at the application layer.

C.1 Classification of Speech Quality Adaptation Mechanisms

Figure C.1 outlines a taxonomy for adaptation mechanisms of speech quality at the application layer. It is based on the where/what/when/how questions for eliciting the requirements for a self-adaptive system, presented in Section B.5.3.

With regard to the requirements about *how* and *when* adaptation should be applied to a managed voice flow, we can expand the three approaches identified by Sfairopoulou et al. [174] to codec-adaptation only as follows:

1. *Nonadaptive.* Performed by an intermediate node, it consists in dropping or blocking new calls that affect the quality of the other ongoing calls. The RTP flow is not adapted. See the work of McGovern et al. [130] as an example.
2. *Single-Adaptive.* Only one call is managed by the VoIP endpoints, in such a manner that adjustable parameters (e.g., codec, packetization, FEC) are tuned during the ongoing call. It can be performed by sender, receiver or intermediate node. It can be applied anytime during the call or silence periods between talkspurts.
3. *Multiadaptive.* Performed by an intermediate node, more than one call is managed at once. It can be applied in two ways:
   (a) *Bulk traffic.* The planning agent can adapt all ongoing calls managed by the intermediate node. See the work of Sabrina & Valin [168] as an example.
   (b) *New calls.* Adaptation is applied only when new calls are accepted by the intermediate node, without modifying current calls. See as examples the works of Chen et al. [32], Gardner et al. [58], and Escobar & Best [51].

Depending on which VoIP component *executes* the adaptation plan, we can classify the approaches to speech quality adaptation into two groups: *source-based*, where the adjustable parameters are available at the sender (Section C.2), and *sink-based*, where the adjustable parameters are available at the receiver (Section C.3). Figure C.2, inspired on the work of Jammeh et al. [99], presents a conceptual diagram of where those adjustable parameters are located in the VoIP components of the RTP flow.

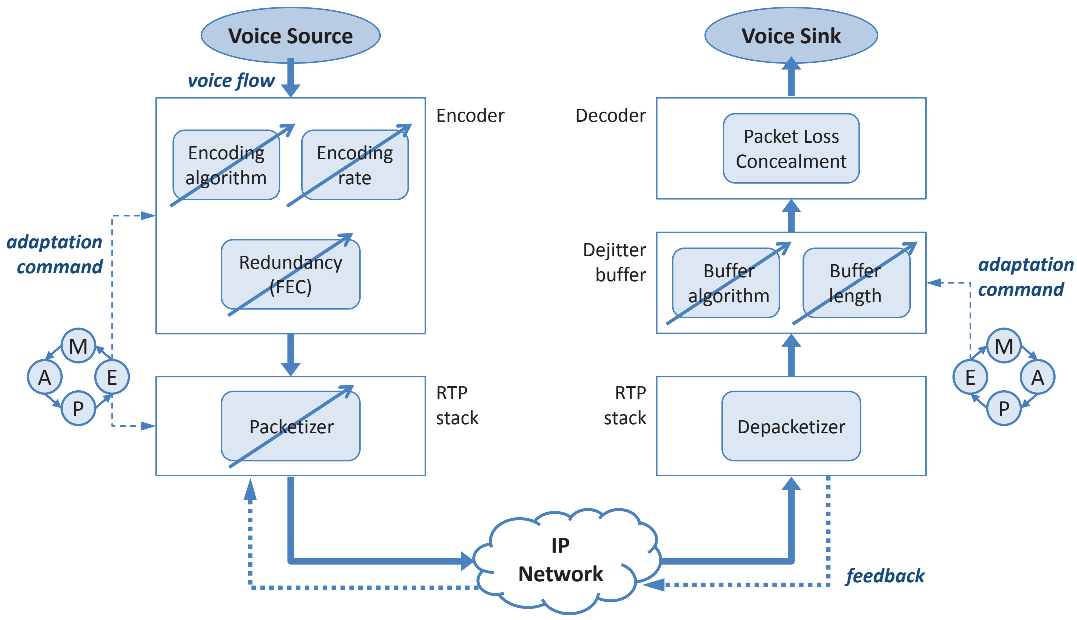


Figure C.2: Placement of adjustable parameters used for executing the adaptation plan in a VoIP architecture.

In the remainder subsections, we explain the attributes used in the headers of tables C.2, C.3, C.5 and C.6 for comparing the diverse VoIP architectures that perform speech quality adaptation at the application layer.

C.1.1 Adaptation-Related Variables

Usually, a system converts input signals into output signals by performing operations on the inputs and intermediate products. The values of measurable properties of system's states are called *variables* [177]. A first step in designing an adaptive mechanism is to identify the key variables of the managed system. An adaptive mechanism usually deals with four kinds of variables, as shown in Figure C.3:

1. *Observation parameters.* They are measurable variables from which the adaptive mechanism can infer the status of the managed system.

2. *Decision metrics.* They characterize the system performance over a sampling period and that the planning agent tries to optimize. They can be equivalent to a single observation parameter, such as delay and packet loss, or a synthesis of a set of observation parameters, such as MOS.

3. *Performance references.* They represent the desired system performance in terms of observation parameters.

4. *Adjustable parameters.* They correspond to the effectors in the feedback loop (Figure B.7), an attribute of the managed system that can be manipulated to apply the necessary adaptations. Essentially, adaptive systems implement a transfer function that takes decision metrics as input and gives the amount of change (if needed) in the adjustable parameters as output.

As an illustration, let us consider an adaptive VoIP mechanism that uses packet loss for determining the MOS. If the MOS value is below 3.6, then some change in codec bitrate is triggered. In this example, *packet loss* is an *observation parameter*, since it cannot be controlled

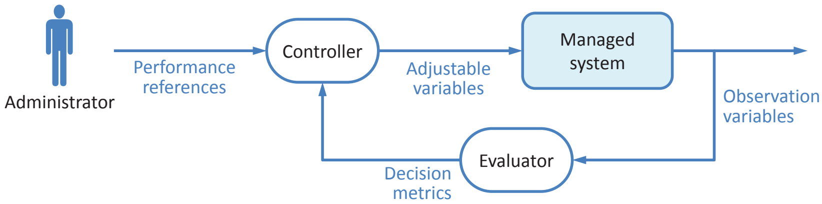


Figure C.3: Variables related to a generic adaptive system.

by sender and receiver, and it is not used for deciding about adjustments in the system. *MOS* is a *decision metric*, because it summaries some observation parameters, and it is used for deliberating about changes. *Codec bitrate* is an *adjustable parameter*, since it can actually be controlled by the sender. Finally, the *MOS threshold* of 3.6 is a *performance reference* to the mechanism of the example.

Observation parameters. The MOS, if taken alone, is not enough to diagnose the cause of problems in speech quality and to support recovery planning over the system [143]. Hence, it is recommended to collect more observation parameters to decide which adjustable parameters should be tuned.

Decision metrics. In the adaptive VoIP architectures here surveyed, the most used decision metrics can be divided into three groups:

1. *QoE metrics.* They characterize the overall acceptability of the service as perceived by the end-user, and its most popular measure is the MOS [184]. In most of the surveyed works, the MOS is determined through the E-model. The only exceptions are the works of Qiao et al. [158], which use a mix of PESQ and E-model, and Mohamed et al. [136], which determine the MOS value using a neural network based on network conditions.

2. *NQoS.* Network QoS parameters comprise all metrics determined from measurements taken at the network layer, such as packet loss, network delay, jitter, bandwidth, throughput, and congestion level.

3. *L2QoS.* Layer-2 QoS parameters comprise all metrics determined at the underlying medium access technology, such as transmission rate and modulation scheme.

Performance references. Performance references should be decoupled from the source code that implements an adaptive VoIP system. A network administrator may update such values, so that the MAPE agents can get them from a knowledge base.

Adjustable parameters. Generally, VoIP applications may tune the parameters listed in the second column of Table C.1. These parameters should not be considered separately, because optimization with respect to a single parameter has harmful effects on the others [143]. This is expressed in the columns “expected benefits” and “side effects”, which lists the factors of the E-model (equations B.2 to B.4) that are affected by changes in the adjustable parameters. These two columns evince the planning agent’s challenge in improving some factor – benefits, constrained by the undesirable associated deterioration in other(s) factor(s) – side effects.

The rightmost column of Table C.1 lists practical limitations in adjusting some VoIP parameters, not covered by the E-model. They are further discussed in the sections correspondent to each adjustable parameter, especially in the paragraphs labeled as *application* and *limitations*.

Table C.1: Adaptive actions that can be performed by either sender or receiver endpoints to improve speech quality, and their expected benefits and side effects.

Endpoint	Adjustable parameter	Expected benefits	Side effects	Practical limitations
Sender	Codec algorithm	Reduction of bandwidth demand	Increase of I_e and T_pkt, change in T_look, and reduction of *Bpl*	Delay for signaling the switching, and audio glitch during transition
		Reduction of I_e	Increase of bandwidth demand	
	Coding bitrate	Reduction of bandwidth demand	Increase of I_e	Audio glitch during rate change
		Reduction of I_e	Increase of bandwidth demand	
	Packetization	Reduction of bandwidth demand	Increase of T_pkt, and reduction of *Bpl*	It requires a larger dejitter buffer
	FEC (redundancy)	Reduction of *Ppl*, and change in PLB	Increase of T_a and bandwidth demand	It requires a larger dejitter buffer
Receiver	Dejitter buffer	Reduction of T_buffer or Ppl_buffer	Increase of T_buffer or Ppl_buffer	Frequent changes may propagate (instead of eliminate) jitter
	PLC mechanism	Increase of *Bpl*	—	Coupled to codec code

In this thesis, we do not cover VoIP architectures that use PLC or header compression as adjustable parameters. Actual implementations of PLC techniques are normally intrinsic to the codec algorithm, such as G.711, G.729 and iLBC [144]. Moreover, header compression techniques are implemented on a hop-to-hop basis, rather than end-to-end [107].

C.1.2 Placement of the MAPE Agents

Basically, the monitoring, analysis, planning and execution (MAPE) agents can be placed in three elements of the VoIP architecture: *sender* endpoint, *receiver* endpoint, or an *intermediate node*. Most of the observation parameters (e.g., end-to-end delay, packet loss and MOS) are determined at the receiver side and sent back via a feedback channel (e.g., RTCP report). However, waiting for RTCP packets may cause further delay in reaction [135]. This is why the most important boundary condition when designing VoIP adaptive systems is to keep *control and feedback information traffic at minimal.*

Here, we focus on the placement of the planning agent, because it determines the dynamics of both *estimation* messages, which gather observation parameters of the managed ongoing call, and *feedback* messages, which determine the adjustment parameters to be tuned accordingly.

Some mechanisms implement the planning agent in an intermediate node, which can be a media gateway [57, 193], a wireless access point performing cross-layer QoS management [32, 175, 108, 195], or a dedicated QoS management node [58, 191].

When the planning agent is implemented in the endpoints, three main strategies are commonly adopted, as depicted in Figure C.4:

1. *Adaptation decision and execution taken by the* sender *of the managed RTP flow.* The planning agent resides in the sender. When it decides to adapt some adjustable parameter, it sends an internal message to the execution agent for applying the changes. This is the most adopted strategy among the surveyed works.

2. *Adaptation decision taken by the* receiver*, and execution taken by the* sender *of the managed RTP* flow*.* The planning agent resides in the receiver. When it decides to adapt,

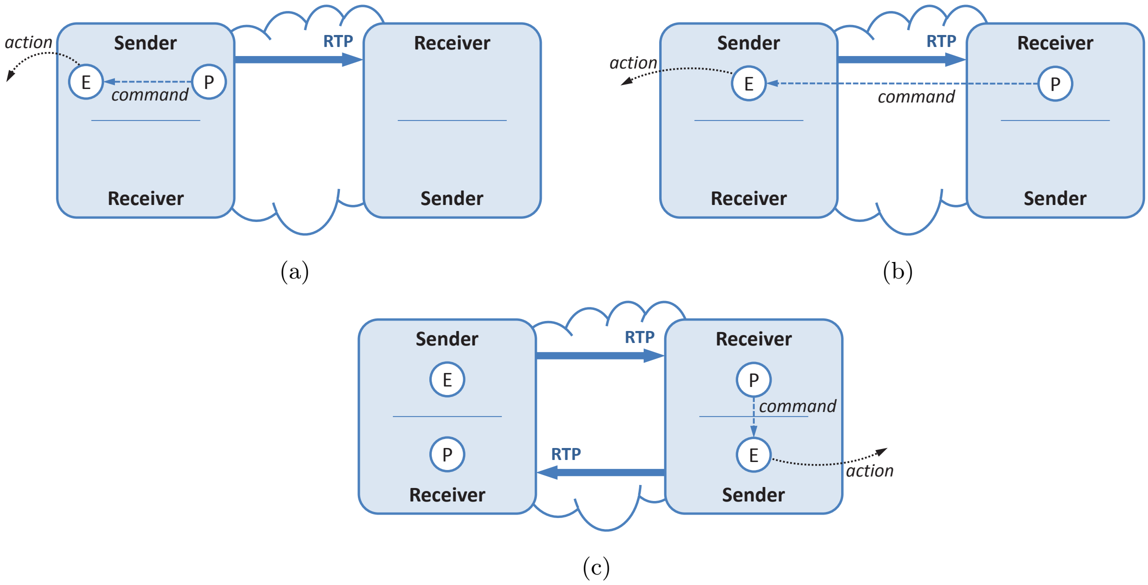


Figure C.4: Three strategies for placing MAPE agents: (a) planning (P) and execution (E) agents at sender, (b) planning agent at the receiver and execution agent at sender, or (c) planning agent at the receiver and execution agent at the sender of the RTP counterflow.

it sends a feedback message to the execution agent implemented in the sender to apply the change. This strategy is adopted by Myakotnykh & Thompson [143].

3. *Adaptation decision taken by the* receiver*, and execution taken by the* sender *of the managed RTP* counterflow. Insofar as a user can act as both speaker and listener, a VoIP call is made up of two RTP flows. If the planning agent in the receiver side of one flow decides to adapt, then it can ask the sender of the opposite flow to apply the change, by means of a reINVITE SIP request, for example. Since they are physically implemented in the same softphone, only an internal message is needed to convey the change plan. This strategy, which is adopted by McGovern et al. [129] and Mkwawa et al. [135], seems to be more efficient in terms of reaction time, as mentioned in Section C.1.5.

Two other placement strategies were proposed by Escobar & Best [51], in which the *caller* analyzes and plans adaptation actions over the RTP flow; and by Jammeh et al. [99], in which the *callee* monitors speech quality and sends an alarm using the SIP instant message (IM) to the *caller*, which runs the adaptation scheme for codec switching. Note that *caller* and *callee* are roles performed by the endpoints during call establishment. After this, the RTP connection is set up, and the endpoints assume the roles of sender and receiver of the RTP flow.

C.1.3 Estimation Procedure

Most of the surveyed works sniff the RTP flow direct or indirect, through RTCP reports, with the purpose of collecting QoS metrics (e.g., T_{a}, *Ppl*) and QoE metrics (e.g., MOS), which help in diagnosing problems of ongoing calls. On the other hand, some adaptive mechanisms monitor parameters at the MAC layer, such as transmission rate, in order to control the sending bitrate [2, 16, 106, 108, 109, 173, 175, 195]. This can be performed by the sender alone, without consulting the receiver, which allows a faster reaction time against quality degradation. Such a cross-layer approach may use also RTCP RR reports for providing a refined adaptation [2, 175, 195]. However, in satellite systems, because of the large end-to-end delay, the use of RTCP RR reports by the adaptation mechanism is not feasible [106].

Finally, observation parameters may also be exchanged by a signaling protocol, such as SIP instant message [99] or eXplicit Control Protocol (XCP) [168].

C.1.4 Feedback Delivery

The planning agent devises the necessary adjustments on the VoIP system to improve speech quality. If the execution agent and the effectors are physically separate from it, then some protocol is needed to convey change plans. Feedback information can be sent through some RTCP report, such as RTCP XR, or by SIP/SDP QoS report [99]. On the other hand, if the planning and execution agents are implemented in the same component, then only internal messages are needed to convey feedback.

C.1.5 Time Dimension

Finally, the following time-related aspects should also be considered in regard to adaptive mechanisms:

1. *Adaptation moment.* If adaptation is executed anytime during a talkspurt, user perception about speech quality may be severely affected [143]. Thus, the execution agent should wait for periods of silence between consecutive talkspurts before applying changes.

2. *Reaction time.* The exchange of feedback messages, such as RTCP reports, is asynchronous. This can retard the reaction time of the adaptation mechanism.

3. *Transients.* The first packets of an RTP flow may be an inadequate amount to draw conclusions from. Additionally, a new adaptation should be devised only after sufficient time has elapsed since the previous one. For example, consider a planning agent implemented in the receiver and an execution agent in the sender. If the sender applies the adjustment immediately, then it will take at least a round-trip time (RTT) for the receiver to perceive the result. Meanwhile, there is no reason for the planner to send new change plans to the execution agent.

4. *Obsoleteness.* As far as data packets can get delayed in the IP network, an adaptation command i may reach its destination after the arrival of a later command $i + j$. If the execution agent takes this information as new, then speech quality may be degraded instead. Because RTCP keeps a counter in its header, many solutions on the literature rely on it to convey feedback.

5. *Overshoot.* Frequent changes in adjustable parameters may cause instability on the VoIP system and degrade speech quality even worse. To avoid this, some works employ the Additive Increase Multiplicative Decrease (AIMD) algorithm, commonly used for congestion control in TCP [13, 139, 158]. However, Ngamwongwattana [147] argues that "the main objective of adaptive-rate control can be different from AIMD, because VoIP performance refers to end-to-end quality rather than throughput."

C.2 Source-Based Adaptation

Source-based adaptation approaches are those *executed* by the sender endpoint to recover or optimize speech quality. The sender can be a VoIP terminal, implemented either in a softphone or hardphone. It can be also a media gateway placed between the IP network and the access network, such as PSTN or mobile telephone network.

Source-based QoS management is designed to improve *long-term* voice flow characteristics, since the sender endpoint has to wait for control messages from the planning agent telling it to change encoding configurations. In contrast, sink-based management is used for improving *short-term* quality. Its fast reaction does not change the encoding characteristics, but manages the delay/loss trade-off.

As shown in Figure C.1, source-based adaptation can be performed by two ways, depending on which adjustable parameters it controls, as described in Table C.1:

1. *Bandwidth control of media information.* Source-based adaptation can regulate the amount of voice information per time unit it delivers to the network. This can be accomplished by switching the current codec, regulating the encoding rate (for VBR codecs), or adjusting the frame per packet ratio (packetization). It is detailed in Section C.2.1.

2. *Redundancy.* The sender can spread redundant information over several packets so that frame information can be recovered at the receiver even if some packets are lost. It is detailed in Section C.2.2.

The remainder of this section groups the sender-based control approaches to VoIP adaptation according to the adjustable parameter used by the planning and execution agents for (1) changing the codec bitrate, or (2) making the voice stream more robust to packet loss. Tables C.2, C.3, C.5 and C.6 summarize our review on the literature about such approaches. Most of their columns correspond to the self-adaptive requirements elicited by the where/what/when/how questions, as described earlier in Section C.1.

C.2.1 Bitrate Control

Bitrate control is also known as source rate control [152], because it can be only performed by the media stream source. As mentioned before, there are three ways for accomplishing bitrate control: coding algorithm switching, encoding rate control, and packetization adjustment, which are detailed in the following subsections.

Coding Algorithm Switching

The first way of realizing bitrate control is by switching the codec currently used to encode the speech input. During call setup, the VoIP endpoints exchange a list of preferred codecs and choose the first match for encoding media. If the planning agent has access to this list, then it can consult other codec matches and determine the most suitable for the current call conditions. A softphone can switch not only between two NB codecs, but also between two WB codecs, or between an NB and a WB codec, or conversely.

The purpose of codec switching is to adapt the source bitrate demand according to the channel bandwidth. If the channel becomes loaded, a low-bitrate codec could be more suitable for maintaining a seamless, though low-quality, voice call. On the other hand, if the channel is not loaded, a high-quality codec can be used instead, allowing a better call experience.

Bolot & Vega-García [20] proposed one of the earliest works to use codec switching as adjustment parameter for VoIP adaptation. Although they claim to perform encoding rate adaptation, they indeed perform codec switching. Multirate codecs such as AMR [53] were only developed two years later, in 1998. Table C.2 summarizes the VoIP architectures that used codec switching as adjustable parameter.

Table C.2: Adaptive architectures based on codec switching.

Reference	Placement of		Decision metric	Estimation source	Feedback delivery	Adaptation moment
	Analyzer	Planner				
Alshakhsi & Hasbullah [2]	sender	sender	QoE, L2QoS	MAC, RTCP	—	anytime
Bolot & Vega-García [20]	receiver	sender	NQoS	RTP	modified RTCP	anytime
Chen et al. [32]	interm. node	interm. node	NQoS	simulation	simulation	end or rate change of another call
Costa & Nunes [41]	receiver	sender	QoE, NQoS	RTCP	custom protocol	anytime
Escobar & Best [51]	caller party	caller party	NQoS	RTCP	new call	beginning of the call
Galiotos et al. [57]	interm. node	interm. node	QoE	RTCP	MGCP MDCX	anytime
Gardner et al. [58]	interm. node	interm. node	QoE	RTP	new call	beginning of the call
Jammeh et al. [99]	callee party	caller party	QoE, NQoS	SIP IM	reINVITE SIP request	anytime
Ksentini [109]	sender	sender	L2QoS	MAC	reINVITE SIP request	anytime
Mazurczyk & Kotulski [127]	receiver	sender	QoE	RTCP	RTP (audio watermark-ing)	anytime
McGovern et al. [129]	receiver	receiver	NQoS	RTP	reINVITE SIP request	anytime
Mohamed et al. [136]	receiver	sender	NQoS, TFRC	RTCP	modified RTCP	anytime
Myakotnykh & Thompson [143]	receiver	receiver	QoE	RTCP	custom protocol	silence period
Ng et al. [146]	sender	sender	NQoS	RTCP RR	reINVITE SIP request	anytime
Roychoudhuri & Al-Shaer [166]	sender	sender	QoE, NQoS	simulation	—	anytime
Sfairopoulou et al. [175]	interm. node	interm. node	QoE, NQoS, L2QoS	MAC, RTCP	reINVITE SIP request	anytime
Tebbani & Haddadou [191]	interm. node	interm. node	QoE	RTP emulation	reINVITE SIP request	anytime
Trad et al. [193]	interm. node	interm. node	NQoS	RTCP RR	simulation	anytime
Tu et al. [194]	receiver	sender	NQoS	RTP	custom protocol	anytime
Tüysüz & Mantar [195]	interm. node	interm. node	QoE, L2QoS	MAC, RTCP	reINVITE SIP request	anytime

Application. According to by Möller et al. [137], switching codecs is advantageous if Ppl is high, and if the changeover helps in reducing the packet loss impact. They have found that "switching the audio bandwidth is roughly equivalent to the quality degradation of 5–10% packet loss, in both NB and WB conditions." This is explained by Equation B.4: an expected reduction in I_e factor should be balanced by a decrease in Ppl, so that the influence of $I_\mathrm{e,eff}$ on R budget can be diminished, and hence quality can be improved by this codec switching.

On the contrary, if Ppl is not expected to be reduced, packetization adjustment may be a better alternative than codec switching, because it does not change the I_e value, at the expense of a higher delay (T_pkt).

Möller et al. [137] also conclude that "switching codecs in order to take profit of a larger audio bandwidth is advantageous only if a sufficiently long period of WB speech transmission remains." This is explained by the recency effect [66], which states that quality variations have more impact on subjects' overall judgments when they occur at the middle or at the end of speech sequences. Unfortunately, it is not possible for the planning agent to know the remaining length of a call before deciding to apply codec switching.

Practical concerns. Two important concerns when applying codec switching without interrupting an ongoing call and the voice stream are the following:

1. *Codec renegotiation.* Codec switching is not implemented only by changing the payload type (PT) field of the RTP header. It requires a codec renegotiation by means of appropriate synchronization between the endpoints in order to avoid data misinterpretation. If the endpoints are SIP clients, this task can be performed by means of a reINVITE request, as detailed by Wältermann et al. [200].
2. *Dejitter buffer management.* After codec renegotiation, the receiver softphone will temporarily handle two voice streams. Considering that audio content is not double-sent to avoid increasing bandwidth consumption, Wältermann et al. [200] identify two basic approaches to deal with two voice streams during the switching from a codec A to B:
   (a) *Hard codec switching (HCS).* It deletes the remaining packets in the dejitter buffer of the codec A, and rejects newly incoming packets yet encoded by the codec A. Thus, only codec B packets are sent and decoded. It obviously introduces a gap in the voice stream, which lead to some speech quality degradation.
   (b) *Soft codec switching (SCS).* Packets remaining in the dejitter buffer are decoded by the codec A. At the same time, the newly created dejitter buffer can be filled with packets from codec B. Once the dejitter buffer of codec A is emptied, the respective media stream instance can be destroyed, and it is switched to the media stream instance including codec B.

Limitations. Although codec switching is very used, it has some drawbacks, like:

- Implementing many codecs in the same platform may be expensive, because some codecs, such as G.729, G.723 and G.728, are proprietary.
- Even using SCS, the transition from one codec to another may not be transparent to users and so cause some distraction [148]. In this sense, Myakotnykh & Thompson [143] point that further investigation is needed to account for what users hear when encoding parameters are changed and how this variation affects user's feelings about a call.

Some other considerations are not disadvantages, but should be further investigated because they affect the overall system:

- Codec complexity and power consumption were not considered as boundary conditions by the surveyed works. In mobile devices, the planning agent should also ponder these aspects while deceiving changes in adjustable parameters.
- Security problems, such as trust, arise when two or more agents have to interact to achieve some goal – speech quality improvement, in our case. For example, a malicious agent can act as being a trusted VoIP planning agent and ask the execution agents of all softphones currently engaged in a conversation to raise their codec bitrate up to the maximum, leading to a general collapse of the underlying IP network.

Encoding Rate Control

A second way of realizing source-based adaptation by means of bandwidth control is by adjusting the encoding rate of variable bitrate (VBR) codecs. Table C.3 summarizes the works that used encoding rate control as an adjustable parameter for VoIP adaptation.

Application. Encoding rate control is recommended when packet loss is persistent (i.e., the bandwidth along the data path is limited). Thus, packet loss can be avoided by switching to a lower coding or sampling rate [73, 79].

Encoding rate control allows a more efficient message exchanging. While codec switching eventually requires a reINVITE SIP request transaction, encoding rate control can use some field of the RTP payload header to carry a change plan of a particular bitrate. For example, the RTP payload header of AMR includes the Codec Mode Request (CMR) field, which was used by Mkwawa et al. [135] and Zhou et al. [207] for changing the current encoding rate.

VBR codecs. Table C.4 lists the most used VBR codecs for bitrate control in VoIP.

The Adaptive MultiRate (AMR) [53] is the most used VBR codec among the surveyed works, because almost all of them used simulation for validating their results, and thus they did not have to worry about royalty issues. The AMR encoder does not select its rate by itself. In fact, another module, the codec mode selection, determines the suitable rate and instructs the AMR encoder (or decoder) which rate to use for encoding (or decoding) [75].

Mobile operators generally use wireless channel information for determining the most suitable AMR rate at a given moment. In the VoIP context, information from other layers can be used for this purpose, such as MAC layer parameters [106, 112, 173]; network throughput [13]; network delay [161, 172]; packet loss [205, 206]; transport delay at UMTS Terrestrial Radio Access Network (UTRAN) [16]; or MOS [80, 99, 125, 158, 207].

Skype is the most widely used VoIP service today. However, it is not publicly known how Skype selects a particular codec for a voice session, because it employs proprietary protocols. Even so, Huang et al. [79] have found that G.729 is always used in SkypeOut calls (PC-to-PSTN service). For PC-to-PC voice calls, different versions of Skype use a distinct VBR codec (see Table C.4). To the best of our knowledge, there is no work in the literature that tries to characterize the proprietary *rate control* algorithm used by Skype. Instead, a few studies have been concentrated on its FEC mechanism, as seen in Section C.2.2.

Barberis et al. [10], Huang et al. [78], and Moura et al. [139] used a hypothetical VBR codec in their simulated mechanisms for adaptive rate control, which consisted of a collection of CBR sources with bitrate values ranging from 8 kbit/s to 64 kbit/s in steps of 8 kbit/s. Finally, Abreu-Sernandez & Garcia-Mateo [1] was, to the best of our knowledge, the only study that

Table C.3: Adaptive architectures based on encoding rate reconfiguration.

Reference	VBR codec	Placement of		Decision metric	Estimation source	Feedback delivery	Adaptation moment
		Analyzer	Planner				
Abreu-Sernandez & Garcia-Mateo [1]	custom	sender	sender	NQoS	RTCP SR	—	anytime
Barberis et al. [10]	simulated	sender	sender	NQoS	RTCP RR	—	anytime
Beritelli et al. [13]	AMR	sender	sender	NQoS	RAP, TRFC	—	anytime
Bilbao et al. [16]	AMR	sender	sender	NQoS	UTRAN	reINVITE SIP request	anytime
Huang et al. [78]	simulated	sender	sender	QoE	simulation	—	anytime
Huang et al. [80]	AMR	sender	sender	QoE	simulation	—	anytime
Jammeh et al. [99]	AMR	receiver	sender	QoE, NQoS	simulation	simulation	anytime
Kalama et al. [106]	AMR	sender	sender	L2QoS	MAC	reINVITE SIP request	anytime
Kawata & Yamada [108]	simulated	interm. node	interm. node	L2QoS	MAC	simulation	anytime
Lee & Pan [112]	AMR-WB	interm. node	interm. node	NQoS, L2QoS	simulation	simulation	anytime
Matta et al. [125]	AMR	receiver	sender	QoE	NA	NA	anytime
Mkwawa et al. [135]	AMR	receiver	receiver	NQoS	RTCP	RTP (CMR field)	anytime
Moura et al. [139]	simulated	receiver	sender	QoE, NQoS	RTCP, dejitter buffer	modified RTCP	silence period
Qiao et al. [158]	AMR	sender	sender	QoE	RTCP	custom protocol	anytime
Rabassa et al. [161]	AMR	sender	sender	NQoS	RTCP	—	anytime
Sabrina & Valin [168]	Speex	receiver	sender	NQoS	modified XCP	modified XCP ACK	anytime
Seo et al. [172]	AMR	receiver	sender	NQoS	RTP	modified RTCP	anytime
Servetti [173]	AMR	sender	sender	NQoS	MAC	—	anytime
Yuhe & Jie [205]	AMR-WB	receiver	sender	NQoS	RTP	simulation	anytime
Zhang et al. [206]	AMR-WB	receiver	sender	NQoS	RTP	custom protocol	anytime
Zhou et al. [207]	AMR	receiver	sender	QoE	RTP	RTP (CMR field)	anytime

Table C.4: Characteristics of most used variable bitrate codecs.

Codec	Bitrate (kbit/s)	Sampling rate (kHz)	Packet length (ms)	Characteristics
AMR [53]	4.75 – 12.2	8	20	Proprietary
AMR-WB (G.722.2) [89]	6.60 – 23.85	16	20	Proprietary
iSAC [64]	10 – 32	32	30, 60	It was Skype's default codec until v.3.1. Proprietary
SVOPC [117]	20	16	20	It was Skype's default codec from v.3.2 to v.3.8. Proprietary
SILK [181]	5 – 20	8	20, 40, 60, 80, 100	Current Skype's default codec since v.4.0. Proprietary, but with royalty free version
	7 – 25	12		
	8 – 30	16		
	20 – 40	24		
Speex [71]	2.15 – 24.6	8	20	Open-source
	3.95 – 44.2	16		
	5.75 – 44.0	32		

used VBR control in an actual VoIP system, implemented on the Motorola DSP96002, with bitrates of 9.4, 7.7 and 5.9 kbit/s.

Limitations. Encoding rate control does not account for sporadic data loss, such as when the background traffic contains short-term bursts that occupy a router's buffer space temporarily. This is better tackled by redundancy control techniques, such as FEC. As already observed by Bolot & Vega-García [20], it should be better to complement rate control techniques with redundancy control to minimize the perceived loss rate at the destinations. Furthermore, as reported by Hoene et al. [73], switching the encoding rate can cause clicking sounds.

Packetization Adjustment

Finally, the third form of bandwidth control as means of source-based adaptation is packetization adjustment. In the VoIP context, packetization is the process of bundling one or more voice frames into an RTP packet. Because VoIP is delay-sensitive, most codecs produce small frame sizes so that they do not cause too much delay.

On the other hand, the 40 bytes of RTP/UDP/IP headers are normally larger than payload size. Thus, precious network bandwidth is wasted to carry overhead information instead of content, and packetization is used for optimizing bandwidth consumption. Moreover, by adjusting packetization, bandwidth consumption can be controlled without affecting voice quality, since it is transparent and glitch-free, unlike codec switching and encoding rate variation.

Some works consider packet length as being only a codec setting, and their planning agent does not explicitly reason about the trade-off among voice packet length, bandwidth availability and end-to-end delay. Hence, we did not classify such works as being packetization adjustment, but rather codec switching. For example, the prototype proposed by McGovern et al. [129] performs codec switching among three settings: G.711 (20 ms), G.729 (10 ms), and G.729 (20 ms). Although the authors have considered two different packet lengths for G.729, the planning agent does not generalize this reasoning for G.711. Thus, in this case, packetization was considered at the *design* phase of the proposed system, instead of at *run-time*.

Table C.5 summarizes the works in literature that used packetization as adjustable parameter of their self-adaptive VoIP architectures. We have omitted the *Adaptation moment* column,

Table C.5: Adaptive architectures based on packetization reconfiguration.

Reference	Placement of		Decision metric	Estimation source	Feedback delivery
	Analyzer	Planner			
Alshakhsi & Hasbullah [2]	sender	sender	L2QoS	MAC	—
Bilbao et al. [16]	sender	sender	NQoS	UTRAN	reINVITE SIP request
Lulu et al. [119]	receiver	sender	QoE, NQoS	NA	simulation
Manousos et al. [123]	interm. node	interm. node	QoE	RTCP SR/RR	proprietary
Myakotnykh & Thompson [143]	receiver	receiver	QoE	RTCP	custom protocol
Ngamwongwattana [148]	receiver	receiver	NQoS	RTP	custom protocol
Oliveira et al. [150]	sender	sender	NQoS	RTCP (DLSR)	—
Tüysüz & Mantar [195]	interm. node	interm. node	QoE, L2QoS	MAC, RTCP	reINVITE SIP request
Yuhe & Jie [205]	receiver	sender	NQoS	RTP	simulation
Zhang et al. [206]	receiver	sender	NQoS	RTP	custom protocol

since almost all works perform adaptation at any moment during a call, except for [143], which waits for periods of silence; and [123], which applies changes only to new calls.

All adaptation mechanisms presented in Table C.5 use a predefined set of packet lengths, except for the work of Oliveira et al. [150], which can indefinitely increase the packet length while the end-to-end delay (T_a) does not reach 300 ms. The choice of this threshold as performance reference, however, is questionable, because speech quality drastically decreases when delay is above 150 ms [88].

Application. Packetization should be increased only when the end-to-end delay is low or the expected benefit of a lower packet loss rate overcomes the burden of a higher delay. As observed by Ngamwongwattana [148], in the limit, a too-large packetization does not give much benefit, because the influence of header size on network bandwidth consumption becomes negligible, and thus packetization is not more needed.

If the bandwidth is limited, one should consider the underlying technology before deciding which adjustable parameter should be adapted [73]. On Ethernet links, both bitrate and packetization shall be adapted. On an IEEE 802.11 wireless, it is sufficient to decrease only packet rate to save a significant share of the bandwidth.

Finally, Myakotnykh & Thompson [143] showed that, if the network is not congested, a change of packet size does not provide any quality improvement. In this case, FEC control or dejitter buffer adjustment can be more effective, because packet loss is not due to network congestion. With higher network load, packet size variation improves quality, despite the multiple negative effects mentioned before. Only if this adjustment does not help, the voice stream bitrate should be changed by switching the codec.

Limitations. Packet size adjustment has some side effects on VoIP quality, as pointed by Myakotnykh & Thompson [143]:

- Increasing packet size leads to an increase of end-to-end delay (T_a). If the current T_a is not too large, then its influence on speech quality is not noticeable. But, if T_a is significant, then an increase of packet duration may be noticeable.

- A loss of one *long* packet impacts more negatively on speech quality than a loss of several *small* packets, because PLC techniques work better for small gaps (4 ms to 40 ms) [152], which is modeled by the *Bpl* factor (Equation B.4).

C.2.2 Forward Error Correction (FEC) Control

Besides bitrate control, source-based adaptive mechanisms can dynamically regulate the amount of redundancy needed for recovering lost packets. Table C.6 summarizes the works in literature that used FEC control as adjustable parameter of their adaptive VoIP architectures. Its column format is different from the preceding tables for simplicity, because all surveyed FEC control mechanisms apply changes at any moment during the call.

Traditional codecs are not robust to transmission errors, because they incorporate recursive filters to remove redundancy without considering error resilience. On the other hand, problems like short-term transient congestion, concurrent traffic, and noisy wireless links introduce errors in the transmission channel. In such cases, reducing the source bitrate alone is not enough for improving speech quality. Hence, FEC can be used for optimally assigning the amount of redundant and information bits in response to varying channel conditions [125].

FEC schemes. Perkins et al. [152] divide FEC techniques into two major groups, as summarized in Figure C.5 (highlighted blocks):

1. *Media-independent FEC.* It uses block codes for providing redundant information. Lost packets are recovered in a *bit-exact* form. This technique has the disadvantage of introducing delay, which may become significantly large. Thus, it should not be combined with packetization, otherwise delay can get so high that redundancy becomes useless. Two widely used media-independent FEC techniques are the following:
   (a) *XOR parity coding.* It performs exclusive-or (XOR) operations between packets to generate parity packets. One simple example is to send, along with every n packet, the result of bitwise XOR on the $n-1$ previous packets. This technique can exactly recover one lost packet if packet losses events are at least separated by n packets. More elaborate schemes can be achieved by different combinations of packets. By increasing the amount of redundancy and delay it is possible to correct a small set of consecutively lost packets.
   (b) *Reed-Solomon coding.* This scheme performs a more powerful error correction than the previous one. From each k data packets produced by the codec, it generates $n-k$ additional check packets and thus transmits n packets over the network. At the receiver, the original k packets can be exactly recovered by receiving any subset of k (or more) packets.
2. *Media-specific FEC.* The source audio is encoded with different quality coders in multiple packets. Redundant audio segments produced by low-bitrate encoding are piggybacked onto a later packet. When a packet is lost, another packet containing the same segment can be used for covering the loss. This technique, also known as low-bitrate redundancy (LBR), has the advantage of low latency (only a single-packet delay is added). However, the choice between primary and redundant encoding is a difficult problem and depends on both bandwidth requirement and computational complexity of the encoding [152].

An intermediate approach between media-independent and media-specific FEC was taken by Johansson et al. [104], which employed XOR parity coding for the sensitive frames only: important frames were transmitted twice, while the remaining frames were transmitted only

Table C.6: Adaptive architectures based on Forward Error Correction (FEC) control.

Reference	Joint bitrate control	Decision metric	FEC scheme	Placement of Analyzer	Placement of Planner	Feedback delivery
Abreu-Sernandez & Garcia-Mateo [1]	multirate	*Ppl* before reconstruction	LBR	sender	sender	RTCP SR
Bolot & Vega-García [20]	multirate	*Ppl* after reconstruction	LBR	receiver	sender	modified RTCP
Boutremans & Boudec [24]	—	MOS	Reed-Solomon and LBR	receiver	sender	modified RTCP
Gong & Kabal [63]	—	dejitter buffer size	LBR	receiver	receiver	modified RTCP
Huang et al. [76]	—	MAC packet loss	Reed-Solomon	receiver	receiver	modified RTCP
Huang et al. [80]	multirate	MOS	XOR	sender	sender	simulation
Johansson et al. [104]	—	*Ppl* after reconstruction	XOR (significant packets)	receiver	sender	modified RTCP
Jung & Ibanez [105]	—	PLB	LBR	receiver	sender	RTCP XR
Li et al. [113]	—	*Ppl* before reconstruction	Reed-Solomon	receiver	sender	custom
Lizhong et al. [118]	—	*Ppl* per FEC block	Reed-Solomon	receiver	sender	simulation
Manousos et al. [123]	packetization	MOS	ON/OFF	interm. node	interm. node	NA
Matta et al. [125]	multirate	MOS	Reed-Solomon	receiver	sender	NA (analytical study)
Mazurczyk & Kotulski [127]	codec switching	MOS	ON/OFF	receiver	sender	RTP (audio watermarking)
Padhye et al. [151]	—	PLB, *Ppl* before *and* after reconstruction	LBR	receiver	sender	modified RTCP
Roychoudhuri & Al-Shaer [167]	multirate	PLB	Reed-Solomon	receiver	sender	NA
Skype [79]	multirate	packet loss	Reed-Solomon (likely)	NA	NA	NA

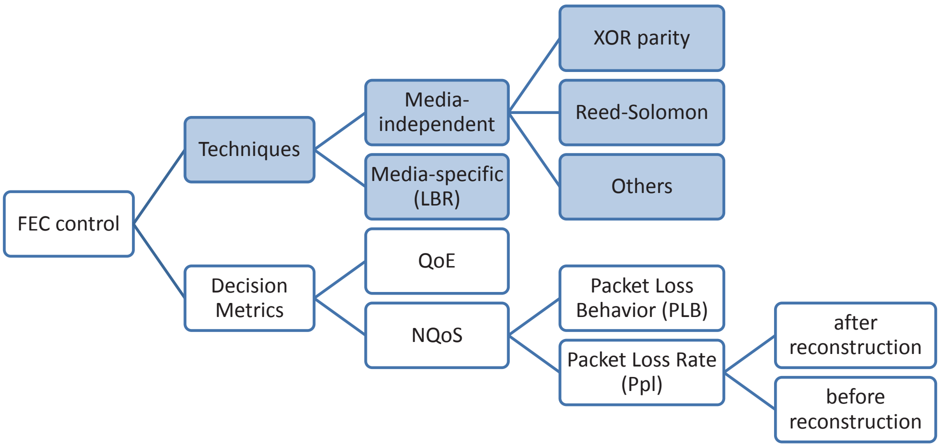


Figure C.5: Classification of techniques and decision metrics used for FEC control.

once. This selective redundancy considers voice characteristics before applying FEC, but it can recover the lost packets in a bit-exact form.

The adaptive mechanisms implemented by Manousos et al. [123] and Mazurczyk & Kotulski [127] do not adjust the redundancy in different levels, but rather activate or deactivate a predefined and static FEC scheme. In Table C.6, we call these partial adaptive approaches as *ON/OFF FEC control*.

Finally, Huang et al. [79] have found that the Skype's FEC mechanism seems to piggyback redundant data to a number of packets based on network loss rate, without considering packet loss burstiness or the employed codec.

Decision metrics. Considering that the primary objective of FEC is to reduce *packet loss* as perceived by application, packet loss rate (Ppl) *after reconstruction* was used for deciding the required redundancy level in seminal works, such as Bolot & Vega-García [20], and in later works, such as Johansson et al. [104] and Lizhong et al. [118]. However, Padhye et al. [151] warn that packet loss "may not be representative under all network conditions," such as during burst losses, when many consecutive packets are lost. In this case, increasing the redundancy may be a waste of bandwidth.

One approach to overcome this problem is to track not only the packet loss after reconstruction, but also *before reconstruction* and *during loss bursts* [104]. Another approach, taken by Jung & Ibanez [105], is to characterize the packet loss behavior within a gap or a burst state and apply the most suitable redundancy level accordingly. One more elaborated packet-loss-related approach was taken by Roychoudhuri & Al-Shaer [167], which not only tracks the PLB, but also tries to predict packet loss due to congestion, based on variations of inter-packet gap and end-to-end delay, and near history of these observations.

Other works also consider the increase of end-to-end delay while awaiting for the arrival of redundant packets. When Ppl_{net} is low, redundancy is needless, but the delay due to FEC can reduce speech quality. Moreover, Ppl *after reconstruction* may include packet discard at the dejitter buffer, so that the planning agent can wrongly judge this metric as being a symptom of lack of redundancy, instead of huge delay. Boutremans & Boudec [24] pointed out this joint problem of FEC and playout delay adjustment, and used the MOS as a decision metric, because it considers both packet loss and end-to-end delay. Similar approach is also taken by Matta et al. [125], Manousos et al. [123], Huang et al. [80], and Mazurczyk & Kotulski [127].

Finally, Gong & Kabal [63] adopt a delay-aware approach, but, instead of MOS, they use the dejitter buffer size and packet loss as decision metrics. Reconstruction is possible only when the buffer size is larger than the time interval between the lost and the redundant packets. Figure C.5 summaries the most used decision metrics used for FEC control among the surveyed works presented on Table C.6.

Joint FEC and Rate Control. Sending redundant packets increases the bandwidth requirements in proportion to the redundancy level [144]. Even so, if piggybacking is used, then this relationship can be attenuated, because of the 40-byte RTP/UDP/IP headers [103]. Anyway, the extra bandwidth required for conveying FEC information may increase the packet loss rate [20]. Thus, the FEC scheme can be coupled to some rate control scheme to provide a more robust adaptive VoIP mechanism.

Indeed, as pointed out in the second column of Table C.6, the three bitrate control techniques can be combined to FEC control for adapting the speech quality of the RTP flow. For example, *codec switching* is also employed by the adaptive mechanism of Mazurczyk & Kotulski [127]; *multirate encoding* adaptation is employed by Abreu-Sernandez & Garcia-Mateo [1], Bolot & Vega-García [20], Huang et al. [80], Matta et al. [125], and Roychoudhuri & Al-Shaer [167]; and *packetization* adjustment is adopted by Manousos et al. [123].

Limitations. FEC mechanisms are limited by the following characteristics:

- *Packet loss behavior.* The redundancy level depends on the time-varying characteristics of network packet loss. On one hand, it does not make sense to send redundant information when packet loss is minimal and uniformly distributed. On the other hand, when the loss of consecutive packets is high, redundant information may require a huge delay to be received, which leads us to the next constraint.
- *Delay.* FEC requires a larger dejitter buffer so that redundant packets can be received in time to recovery the correspondent lost packets. Additionally, destination has to wait longer to decode as more redundancy information is used. Thus, the FEC control mechanism should be aware not to impair speech quality due to delay while trying to minimize the effect of packet loss.

C.3 Sink-Based Adaptation

Sink-based adaptation manages speech quality of a call by *executing* changes in adjustable parameters located on the receiver endpoint. The *decision* about which parameters should be changed may be taken by the receiver itself or by another trusted element of the VoIP architecture, such as the sender or an intermediate node.

The receiver endpoint can be a VoIP terminal or a media gateway placed between the IP network and the access network. Since it can readily sense QoS problems of an RTP flow, it can quickly improve short-term speech quality.

Source- and sink-based adaptation can work simultaneously, implementing two independent feedback loops. In this case, both sender and receiver have to take coordinated actions in order to avoid counteracting each other.

The receiver can improve speech quality in two ways: *playout scheduling adaptation*, which addresses the delay/jitter trade-off; and *PLC adjustment*, which addresses packet loss resilience. Both schemes are briefly reviewed in the following subsections.

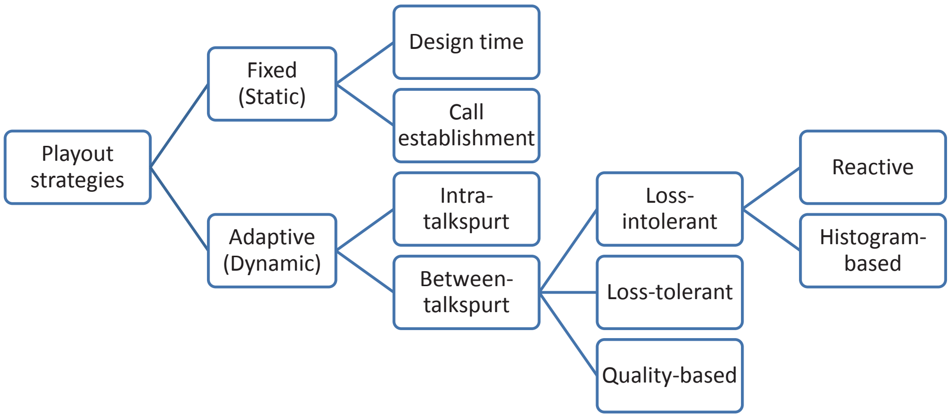


Figure C.6: Classification of playout buffer strategies for IP telephony.

C.3.1 Playout Scheduling Adaptation

Dejitter (or playout) buffers remove jitter by temporarily storing the arriving voice packets and forwarding them to the decoder at regular time intervals. They are responsible also for reordering out-of-sequence packets. Dejitter buffers are broadly categorized as fixed (static) and adaptive (dynamic) size. Atzori & Lobina [5] presented a comprehensive survey about playout scheduling in IP telephony and a taxonomy for dejitter buffer strategies, which is reproduced in Figure C.6 and briefly explained below.

1. *Fixed (static) buffers.* End-to-end delay is kept constant for all voice packets, either at design time or during the call. Such a strategy is inefficient, since it is not resilient against the temporal variability of network behavior.

2. *Adaptive (dynamic) buffers.* They try to find an optimal point in the trade-off between end-to-end delay and packet loss, and dynamically adjust the buffer size accordingly. Depending on *when* they adjust the buffer size, they can be further divided into two groups:

   (a) *Intra-talkspurt.* They adjust the end-to-end delay independently from silence periods, using waveform compression or extension.

   (b) *Between-talkspurt.* They act during periods of silence. They are used more often, because they do not require any signal-processing technique to change the length of the speech. They are further grouped depending on *how* they handle the trade-off between end-to-end delay and packet discard.

      i. *Loss-intolerant.* They estimate network delay and set the playout delay so that only a small fraction of packets are discarded. They do not take PLC into account, resulting in an overestimation of the required playout delay. Narbutt et al. [145] further classify these strategies as *reactive*, which continuously estimate network delay and jitter to calculate playout deadlines; and *histogram-based*, which maintain a histogram of packet delay and choose the optimal playout delay from it.

      ii. *Loss-tolerant.* They monitor the packet-loss ratio or buffer occupancy and adjust playout delay accordingly. An amount of packet loss is then allowed, and playout delay is set to reach this target value.

      iii. *Quality-based.* They seek to maximize some metric linked to the end-user perceived quality, such as MOS.

The quality-based buffers are the most interesting for adaptive VoIP control. Whereas loss-intolerant approaches focus on minimizing *delay*, and loss-tolerant approaches focus on minimizing *packet discard*, quality-based strategies consider *both metrics*. They rely on the quality perceived by the end-user and maximize it through a correct balancing of delay and packet loss [5]. Examples of quality-based adaptive dejitter buffers are [6], [145], and [187]. Basically, the main difference among these solutions poses on the models that they use for describing the packet loss process.

Boutremans & Boudec [24] and Wah & Sat [199] not only presented quality-based playout buffers, but also showed the benefits of jointly approach to playout scheduling adaptation and redundancy control. Their adaptation algorithms control both the dejitter buffer size (at the receiver) and the redundancy level (at the sender), so that the effects of these mechanisms do not neutralize each other.

Hoene et al. [73] studied how dejitter buffers should react to *delay spikes* – i.e., sudden and sharp increases in delay that cannot be predicted in advance. In such cases, the playout scheduling algorithm has to decide between two alternatives: (1) delaying the playout of the speech frame to include such spikes, or (2) dropping those packets. He showed that, if delay spikes occur sporadically, the dejitter buffer should drop the packets within. However, the boundary between *sporadic* and *frequent* states is not clear. Further studies should examine how to draw this difference from events of the ongoing, past, and simultaneous calls among other endpoints.

Finally, Valle et al. [196] proposed a manager of dejitter buffer algorithms, which not only applies changes to the buffer length, but also seeks the most suitable playout policy. Future work may generalize this manager so that endpoints could download, from intermediate trusted nodes (e.g., a signaling proxy), the most recommended dejitter buffer algorithm, based on user's personal preferences, or latest advances in playout buffer research, or payment privilege, or other criteria.

C.3.2 Packet Loss Concealment (PLC) Adjustment

In contrast to FEC, PLC techniques are executed by the receiver endpoint, without any assistance from the sender. Note that PLC is implicitly sender-related, since it is codec-specific, but the concealment itself is performed at the receiver.

PLC techniques try to minimize the effect of the lack of voice packets during playout. From the PLC algorithm perspective, the cause of packet loss is not important. It may have been lost in the network, or may have not arrived in time at destination, or may have not passed in error check, or both original and redundant packet could have been lost. The most important concern here is: what should be played out to replace the gap left by the missing packet(s)?

Perkins et al. [152] presented a comprehensive survey about PLC techniques, which are not reviewed here. Most of the modern low-bitrate codecs have some PLC scheme built into their algorithms. Consequently, in practical VoIP systems, it is not feasible to change the PLC scheme adaptively without changing the codec itself.

C.4 Summary

In this chapter, we reviewed the works in the literature that propose adaptive solutions to tackle the problem of managing speech quality of VoIP call in run-time at the application layer. However, these works do not regard themselves as implementing the feedback control loop, which lies in the core of every self-adaptive system. Thus, we identified the MAPE agents of the control loop and grouped these works into two major groups, depending on the location of the adjustable parameters used to control speech quality. The first group comprises the solutions that adjust parameters located at the *sender* endpoint of the RTP flow, such as codec, encoding rate, packetization, and redundancy (FEC). The second group comprises the solutions that adjust parameters located at the *receiver* endpoint of the RTP flow, such as dejitter buffer and packet loss concealment (PLC).

Receiver-based adaptive solutions have restricted effectiveness, because they act after that the voice flow suffered all kinds of impairments. However, their effect is shortly perceived. On the other hand, sender-based adaptive solutions are more effective in the sense they control the data rate delivered to the network, so it can shape the bit stream in accordance with monitored network conditions. However, they require more time to be applied by the MAPE agents.

Finally, we identified throughout this chapter some open problems that are being tackled in this thesis, as detailed in chapters D, E and F.

Appendix D

Perception of Codec Switching

The history of science knows scores of instances
where an investigator was in possession
of all the important facts for a new theory,
but simply failed to ask the right questions.
Ernst Mayr (1904–2005)

Contents

Codec switching is a technique used by some adaptive VoIP terminals to adjust the sender's bitrate in accordance with monitored network conditions. It consists in changing the codec currently under use for encoding the speech input to another codec, more suitable to deliver the speech content with acceptable speech quality. Because the data load on the network is reduced by using a lower bitrate codec, the probability of high delays and packet loss is expected to decrease too.

As already seen in Section C.2.1, codec switching is a widely used solution for the VoIP adaptation problem (Table C.2, p. 54). However, as pointed by Myakotnykh & Thompson [143], it is still unknown how users perceive speech quality during the exact moment of the switching (short-term perspective). In Section A.2, we have compared this technique to a surgery, in the sense that a procedure used for recovering a system should not ruin the system itself during its application. In other words, the side effects of adaptation should not worsen the problem that it tries to overcome.

In this chapter, we develop a methodology for accounting the degradation perceived at the exact moment of the switching of two codecs belonging to a set of codecs implemented in some softphone. It is based on the Degradation Category Rating (DCR) subjective test for speech quality assessment. In Section D.1, we formalize the problem of selecting a codec switching that minimally impairs user's perception. Next, in Section D.2, we describe the methodology designed for determining the degradation perceived during codec switching. Finally, in Section D.3, we describe and analyze the results obtained from the subjective tests.

D.1 The Problem of Codec Selection

Usually, softphone implementations offer some codecs for compressing the voice flow. As long as technology evolves and newer codecs are made available with more efficient compression/quality ratio, softphone applications has to cope with a large number of codecs. Thus, consider the following definition about the composition of codecs implemented in a softphone.

Definition 1 (Codec Set)**:**
If a given softphone has the codecs $C_1, C_2, \ldots, C_{N_C}, N_C \geqslant 2$, *available in its implementation, then we say that the set* $\mathcal{C} = \{C_1, C_2, \ldots, C_{N_C}\}$ *is the* ***codec set*** *of that softphone.*

Now, consider that the adaptation mechanism implemented in a softphone endowed with a codec set $\mathcal{C}$ employs codec switching for improving speech quality. Thus, there will be a score to express the quality during the switching of every pair of codecs taken from $\mathcal{C}$.

Definition 2 (Degradation Factor)**:**
The degradation in quality perception associated to the switching from codec C_i *to codec* $C_j, i, j \in [1, N_C]$, *is denoted by* $\delta_{i,j}$.

Each $\delta_{i,j}, i, j \in [1, N_C]$, is determined by the Degradation Mean Opinion Score (DMOS) obtained from a DCR test about the perceived quality at the moment of switching from C_i to C_j. Consequently, we have that $1 \leqslant \delta_{i,j} \leqslant 5$. This implies that the perceived quality of a codec switching is inversely proportional to the value of $\delta_{i,j}$. Furthermore, we assume $\delta_{i,i} = 5, \forall i \in [1, N_C]$, because there is no degradation associated when the same encoding scheme is maintained along the call.

Definition 3 (Degradation Matrix)**:**
The degradation matrix $\Delta_{\mathcal{C}}$ *is composed of the* $\delta_{i,j}$ *values related to all permutations of two codecs* $C_i, C_j \in \mathcal{C}$:

$$\Delta_{\mathcal{C}} = \begin{bmatrix} 5 & \delta_{1,2} & \ldots & \delta_{1,N_C} \\ \delta_{2,1} & 5 & \ldots & \delta_{2,N_C} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{N_C,1} & \delta_{N_C,2} & \ldots & 5 \end{bmatrix}$$

Given the preceding definition, we can now state the general problem of codec selection.

Definition 4 (Problem of Codec Selection)**:**

Consider that, during an ongoing VoIP call, speech is encoded by a codec $C_a \in \mathcal{C}$. At a given moment of the call, the analysis agent in the control loop detects some problem and reports it to the planning agent, which decides to switch the encoding from C_a to another $C_i \in \mathcal{C}$. The problem of codec selection consists of determining a target codec $C_i, i \neq a$, such that $\delta_{a,i} = \max\{\{\delta_{a,1}, \delta_{a,2}, \ldots, \delta_{a,n}\} - \{\delta_{a,a}\}\}$.

To solve the Problem of Codec Selection, we have to determine the values of the elements of the degradation matrix $\Delta_{\mathcal{C}}$ by means of a DCR experiment, which is described in a general form in Section D.1.1. In Section D.1.2, we describe how to analyze the outcomes of the DCR experiments by means of statistical hypothesis testing in order to determine the elements of the degradation matrix. Finally, in Section D.1.3, we instantiate the general Problem of Codec Selection to the special case where $\mathcal{C} = \{$G.711, G.726 (32 kbit/s), G.726 (16 kbit/s), G.729$\}$.

D.1.1 DCR Tests

The choice of the DCR method for determining the degradation matrix $\Delta_{\mathcal{C}}$ of a codec set $\mathcal{C}$ was based upon the following considerations:

- Objective tests, such as POLQA, PESQ and E-model, are unsuitable for measuring *variations* in speech quality [189].
- Codec switching is an adaptation technique, as seen in Section C.2.1. Essentially, adaptation requires some kind of *decision* [35], taken by the planning agent between either keeping a current parameter, or choosing a new parameter among some candidates. Such a decision should be toward the candidate that will cause the least possible *degradation* in speech quality during the switching.
- Among the subjective tests – ACR, DCR and CCR –, the only intended for measuring *degradations* in speech quality is the DCR. According to Takahashi [188], in evaluating the fidelity of a system, the ACR method is not necessarily the best one. If we evaluate the coding distortion of speech with background noise by the ACR method, for example, the MOS becomes low even for uncoded conditions because of background noise. In such a case, the DCR method is appropriate.

In DCR tests, subjects are presented to a pair of speech samples: a *reference* and a *target* [188]. The reference is clearly identified, and subjects are asked to rate the target sample against the reference on the DMOS scale. Based on this, we can state the following three definitions.

Definition 5 (Listener Panel)**:**
The set $\mathcal{L} = \{L_1, L_2, \ldots, L_{N_L}\}$ of all N_L subjects that participate as listener of a DCR test run is defined as the ***listener panel*** *of that experiment.*

Definition 6 (Sample-Pair Corpus)**:**
The set $\mathcal{S} = \{S_1, S_2, \ldots, S_{N_S}\}$ of all N_S samples pairs played to the listeners of a DCR test run is defined as the ***sample-pair corpus*** *of that experiment.*

Definition 7 (Talker Panel)**:**
The set $\mathcal{T} = \{T_1, T_2, \ldots, T_{N_T}\}$ of all N_T subjects that utter the sample-pair corpus in a DCR test run, with $\mathcal{L} \cap \mathcal{T} = \varnothing$, is defined as the ***talker panel*** *of that experiment.*

From the three definitions above, we can now define the individual degradation score.

Definition 8 (Individual Degradation Score)**:**
The score individually assigned by a given listener $L_x \in \mathcal{L}$*, to a sample pair* $S_y \in \mathcal{S}$*, uttered by a talker* $T_z \in \mathcal{T}$*, with respect to the degradation perceived during the switching from codec* C_i *to* $C_j, i, j \in [1, N_C]$*, is denoted by* $\widehat{\delta_{i,j}}(L_x, S_y, T_z)$*.*

Thus, the mean of all $\widehat{\delta_{i,j}}$ values assigned by all listeners from the listener panel, to all sample pairs from the sample corpus, uttered by all talkers from the talker panel, constitutes the *raw degradation mean opinion score (raw DMOS)*, which is given by the following expression:

$$\overline{\delta_{i,j}} = \frac{1}{N_L N_S N_T} \cdot \sum_{x=1}^{N_L} \sum_{y=1}^{N_S} \sum_{z=1}^{N_T} \widehat{\delta_{i,j}}(L_x, S_y, T_z) \tag{D.1}$$

Note that the raw score $\overline{\delta_{i,j}}$ is only the average of all assigned values of $\widehat{\delta_{i,j}}(L_x, S_y, T_z)$, but not the $\delta_{i,j}$ value itself in the degradation matrix $\Delta_{\mathcal{C}}$. To determine this latter, we will apply the principles of statistical hypothesis testing, as explained in the next subsection.

D.1.2 Statistical Hypotheses

Statistical hypotheses are statements about theoretical models or about probability or sampling distributions [124]. In this subsection, we specify some statements that will help us in determining the degradation matrix $\Delta_{\mathcal{C}}$ from DCR tests.

There are two hypotheses that must be specified in any statistical testing procedure: the *null* hypothesis and the *alternative* hypothesis. In the context of a statistically designed experiment, the null hypothesis (H_0) defines parameter values or other distributional properties that indicate no experimental effect. The alternative hypothesis (H_1) specifies values that indicate a change or experimental effect for the parameter or distributional property of interest.

A measure of the credibility of the null hypothesis is given by the p-value. For example, suppose that we are comparing two sample means. The null hypothesis H_0 says that the means are the same, and the alternative hypothesis H_1 says that they are different. In this case, a low p-value means that the H_0 is unlikely to be true and the difference is statistically significant. In contrast, a large p-value means that there is no compelling evidence on which to reject the null hypothesis [44].

Note that "the H_0 is rejected" and "the H_0 is false" are different statements. For instance, the null hypothesis can be mistakenly rejected because of a low sample size or a large measurement error [44]. That is, the data set used for hypothesis testing may not be representative, and thus the H_0 may be rejected although it is true, or vice-versa.

Now, we should define the *similarity relation*, which will be used for stating the hypothesis that we want to test regarding to the DCR experiments.

Definition 9 (Similarity Relation)**:**

1. *If the difference between quality degradation factors associated to the switching from codec* C_i *to* C_j *and to the switching from codec* C_i *to* $C_k, i, j, k \in [1, N_C]$ *and* $k \neq j$*, is not significant, then we say that* $\overline{\delta_{i,j}}$ *is* ***similar*** *to* $\overline{\delta_{i,k}}$*, which is denoted by* $\overline{\delta_{i,j}} \sim \overline{\delta_{i,k}}$*.*
2. *If the above does not hold, then we say that* $\overline{\delta_{i,j}}$ *is* ***not similar*** *to* $\overline{\delta_{i,k}}$*, which is denoted by* $\overline{\delta_{i,j}} \nsim \overline{\delta_{i,k}}$*.*

Note that the similarity relation is not an equivalence relation, because it is not transitive. If some $\overline{\delta_{i,j}} \sim \overline{\delta_{i,k}}$ and $\overline{\delta_{i,k}} \sim \overline{\delta_{i,l}}$, with $i, j, k, l \in [1, N_C]$ and j, k, l distinct among each other, we can not claim that $\overline{\delta_{i,j}} \sim \overline{\delta_{i,l}}$, because their respective confidence intervals may not overlap.

Now, to determine $\Delta_{\mathcal{C}}$, we have to find similarity relations between the average results of DCR experiments by means of hypothesis testing. This is accomplished by four tests:

1. *Reliability of a listener's judgment.*

   Among the sample pairs $\mathcal{S}$ presented to the listener panel $\mathcal{L}$, some *anchor* sample pairs should be included, where a codec C_i is switched to the same codec $C_i, i \in [1, N_C]$ [85]. If a listener's judgment is reliable, then it is expected that $\overline{\delta_{i,i}} \sim 5$, because there is no degradation when keeping the same codec during a call.

   Note that each $\widehat{\delta_{i,i}}(L_x, S_y, T_z)$ assigned by a listener L_x is unique, because each S_y and T_z are unique in $\mathcal{S}$ and $\mathcal{L}$, respectively. So, if a listener L_x assigns $\widehat{\delta_{i,i}}(L_x, S_y, T_z) = 5$, then we can regard this outcome as being a *success*; otherwise, as a *failure*. Therefore, a listener's judgement assigned to each anchor sample pair corresponds to a Bernoulli trial.

   Consequently, we can say that the listener's reliability check is a *binomial* experiment, consisting of $N_C \times N_{S,i\to i} \times N_T$ independent trials, where $N_{S,i\to i}$ is the number of elements of the subset $\mathcal{S}_{i\to i} \subset \mathcal{S}$, which groups the sample pairs that contains switchings from codec C_i to C_i.

   Arranging the previous considerations into statistical hypotheses, we have the following:

   $$\mathbf{H_0^R} : \overline{\delta_{i,i}}(L_x) \sim 5, \forall i \in [1, N_C]$$
   $$\mathbf{H_1^R} : \overline{\delta_{i,i}}(L_x) \nsim 5, \forall i \in [1, N_C]$$

   Thus, to check the reliability of a listener $L_x \in \mathcal{L}$, we should compute the difference $(\widehat{\delta_{i,i}}(L_x, S_y, T_z) - 5)$ for each anchor sample pair, and test the null hypothesis (H_0^R) that the average of these differences is not significant using a one-sample t-test on the differences. If L_x is reliable, then we use the grades $\widehat{\delta_{i,j}}(L_x, S_y, T_z), i \neq j$, assigned by him/her to the other non-anchor sample pairs. Otherwise, all grades $\widehat{\delta_{i,j}}(L_x, S_y, T_z)$ assigned by listener L_x are discarded from the experiment and not used in the following tests.

   After discarding the grades of all unreliable listeners, the listener panel $\mathcal{L}$ is reduced to $\mathcal{L}'$, the set of reliable listeners.

2. *Perceptibility of codec switching.*

   Now, given a codec C_a, we should check whether the encoding switchings from codec C_a to $C_i, \forall i \in [1, N_C]$, are perceived in the same way by the panel of reliable listeners $\mathcal{L}'$. In other words, we should check whether any change from codec C_a to every other codec in the set $\mathcal{C}$ (C_a itself included) is significantly perceived by the average reliable listeners.

   Therefore, we have to test the following hypotheses for each codec $C_a \in \mathcal{C}$:

   $$\mathbf{H_0^P} : \overline{\delta_{a,i}} \sim \overline{\delta_{a,a}}, \forall i \in [1, N_C]$$
   $$\mathbf{H_1^P} : \overline{\delta_{a,i}} \nsim \overline{\delta_{a,a}}, \forall i \in [1, N_C]$$

   This check is performed by means of an analysis of variance (ANOVA) over the N_C

independent groups of $\widehat{\delta_{a,i}}, i \in [1, N_C]$. In the special case of $N_C = 2$, this check is reduced to a simple two-sample t-test.

If H_0^P is accepted, then we say that the switching from codec C_a to any other codec in $\mathcal{C}$ is *not perceivable*. Thus we assign $\delta_{a,i} = 5, \forall i \in [1, N_C]$, that is, the entire row in $\Delta_{\mathcal{C}}$ correspondent to the switchings starting from C_a is filled with 5. Otherwise, we say that the switching from codec C_a to some other codec in $\mathcal{C}$ is *perceivable*, and perform the third hypotheses test.

3. *Homogeneity of codec switching perception.*

   In the case that the switching from a given codec C_a to some other codec in $\mathcal{C}$ is checked to be perceivable, we should check whether the encoding switchings from codec C_a to $C_i, i \neq a$, are perceived in the same way by the panel of reliable listeners $\mathcal{L}'$. To accomplish this, we have to test the following hypotheses:

   $$\begin{aligned} \mathbf{H_0^H} &: \overline{\delta_{a,i}} \sim \overline{\delta_{a,j}}, \forall i,j \in [1, N_C], i \neq j, \text{ and } i,j \neq a \\ \mathbf{H_1^H} &: \overline{\delta_{a,i}} \nsim \overline{\delta_{a,j}}, \forall i,j \in [1, N_C], i \neq j \end{aligned}$$

   This check is performed by means of an ANOVA over the $N_C - 1$ independent groups of $\widehat{\delta_{a,i}}, i \neq a$. In the special cases of $N_C = 2$ or $N_C = 3$, this check is reduced to a one- or two-sample t-test, respectively.

   If H_0^H is accepted, then it follows that the switching from codec C_a to any other codec in $\mathcal{C}$ is *homogeneous*. Thus, we set the value of $\delta_{a,i}$ as being the mean of all $\overline{\delta_{a,1}}, \overline{\delta_{a,2}}, \ldots, \overline{\delta_{a,i}}, \ldots, \overline{\delta_{a,N_C}}, \forall i \in [1, N_C], i \neq a$. In other words, we say that the statistical difference among the perceived degradation factors $\delta_{i,j}, i \neq j$, is not significant.

   Conversely, if H_0^H is rejected, then it follows that the switching from codec C_a to any other codec in $\mathcal{C}$ is *not homogeneous*. Thus, we set the value of each $\delta_{a,i}$ as being equal to the correspondent $\overline{\delta_{a,i}}, \forall i \in [1, N_C], i \neq a$. In other words, we say that the statistical difference among the perceived degradation factors $\delta_{i,j}$ is significant.

4. *Commutability of codec switching perception.*

   The degradation matrix can be compared to a distance chart among cities. However, there is no theoretical evidence that it is symmetric, that is, the switching from codec C_i to $C_j, i \neq j$, is perceived in the same way as the opposite switching from C_j to C_i. To check the commutability of a given codec switching, we have to test the following hypotheses:

   $$\begin{aligned} \mathbf{H_0^C} &: \overline{\delta_{i,j}} \sim \overline{\delta_{j,i}}, \forall i,j \in [1, N_C], i \neq j \\ \mathbf{H_1^C} &: \overline{\delta_{i,j}} \nsim \overline{\delta_{j,i}}, \forall i,j \in [1, N_C], i \neq j \end{aligned}$$

   This check is performed by means of a two-sample t-test for each combination of two distinct codecs in the set $\mathcal{C}$.

   If H_0^C is accepted, then it follows that the switching from codec C_i to C_j is *commutable*. Thus, we set $\delta_{i,j} = \delta_{j,i} = (\overline{\delta_{i,j}} + \overline{\delta_{j,i}})/2$. Conversely, if H_0^C is rejected, then it follows that the switching from codec C_i to C_j is *not commutable*. Thus, we set $\delta_{i,j} = \overline{\delta_{i,j}}$ and $\delta_{j,i} = \overline{\delta_{j,i}}$.

Now we are ready to instantiate the Problem of Codec Selection to the codec set used in this thesis, as explained in Section D.2.

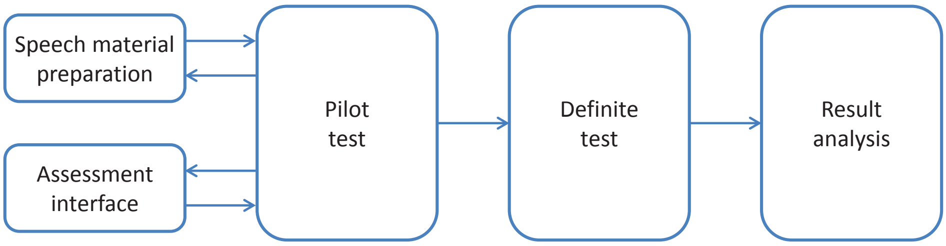


Figure D.1: Methodology for evaluating user's perception during codec switching.

D.1.3 Instantiation

As shown in Table B.1 (p. 35), there is a great variety of codec implementations available in softphones. Consequently, to evaluate all possible combinations of switchings between every possible codec pair is a prohibitive task. Thus, in the present work, we used as a starting point the same codec set studied by Myakotnykh & Thompson [143]: G.711 (64 kbit/s), G.726 (32 kbit/s), and G.729 (8 kbit/s). In this set, we have included the codec G.726 (16 kbit/s) to complete a series of 1:2:4:8 bitrate compression ratios. Furthermore, all these four codecs are available at the PJSIP [155], the VoIP platform that we chose to implement our contributions.

In the remainder of this chapter, we present and describe the methodology that we applied for determining the degradation factors $\delta_{i,j}$ of this four-element set.

D.2 Determination of the Degradation Factors

Degradation factors $\delta_{i,j}$ in the degradation matrix $\Delta_{\mathcal{C}}$ are determined by means of DCR subjective tests. In such experiments, a subject can assume *only one* of the following three roles:

1. *Listeners.* They are responsible for listening to the pairs of speech samples and evaluating the degradation between them.
2. *Talkers.* They are responsible for uttering the sentences that are to be played out to the listeners.
3. *Facilitators.* They are responsible for preparing the speech material, selecting the listeners and the talkers, instructing the listeners about the test procedure, and applying the tests. We had two undergraduate students in Computer Engineering performing this role.

A general view of the methodology for evaluating what users perceive at the moment of codec switching is given in Figure D.1. The first step, described in Section D.2.1, is to prepare the speech material for the subjective tests. We accomplished this by creating speech files containing codec transitions that emulate a codec switching during a VoIP call. The next step is to design and implement a user interface through which the listeners perform the test (Section D.2.2). Then, a pilot test should be run to adjust problems in the interface, to improve the instructions given to the listeners, and to train the listeners in the test facilities and grading process (Section D.2.3). After applying all required adjustments, the definitive DCR tests can be run, using a speech material different from the pilot test (Section D.2.4). The final step is to conduct a statistical analysis of the results, based on the null-hypotheses H_0^R, H_0^P, H_0^H, and H_0^C, for determining the degradation matrix $\Delta_{\mathcal{C}}$ from the raw DMOS values obtained from the DCR tests (Section D.3).

D.2.1 Speech Material Preparation

In a DCR test, subjects are presented to a pair of speech samples: a *reference* and a *target*. For the purposes of this thesis, the reference sample is completely encoded by the same codec (C_A, for example). The target sample is encoded by codec C_A during its first half, and by another codec C_B during the second half, as depicted in the lower part of Figure D.2.

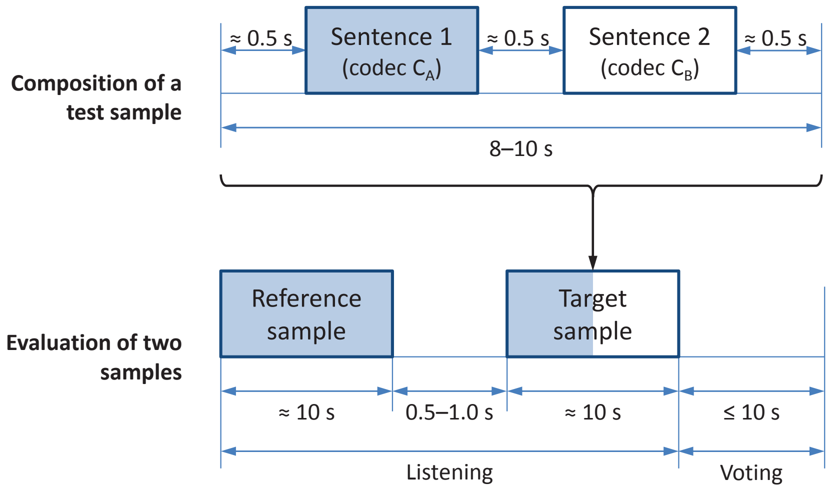


Figure D.2: Composition of two sentences into one sample, and of two samples into a DCR test run.

Furthermore, as pointed by Grancharov & Kleijn [65], a typical sample test contains two short *sentences*, separated by a silence gap of 0.5 s. The resulting two-sentence sample has a duration of about 8 s to 10 s, as shown in the upper part of Figure D.2. In a DCR test run, a subject listens to the reference sample first, then listens to the target sample, which contains the codec switching, and gives a vote in the DMOS scale (Table B.4, p. 38).

Now, remember that we want to evaluate the speech quality at the moment of codec switching. Our codec set Δ_C is composed of four elements ($N_C = 4$): G.711, G.726(32), G.726(16), and G.729. We want to determine the number of 2-combinations from this set. The general formula for determining the number of k-combinations from a set of N_C elements is given by

$$C(N_C, k) = \frac{N_C!}{(N_C - k)!k!}. \tag{D.2}$$

Therefore, we need to perform DCR tests upon $C(4, 2) = 6$ combinations of codec switchings.

Now, for each codec combination $C_A \leftrightarrow C_B$ in the codec set, we need to test three conditions:

1. *Anchor condition.* It is a reference sample composed of a pair of sentences encoded by the same codec ($C_A \to C_A$), included to check the reliability on a listener's judgment.

2. *Switching condition.* It is a sample composed of a pair of sentences encoded by distinct codecs ($C_A \to C_B$). It corresponds to the codec switching that we really intend to evaluate.

3. *Commutation condition.* It is a sample composed of a pair of sentences encoded in the opposite order of the *switching condition* ($C_B \to C_A$), included to check the commutability of a codec switching.

Figure D.3 presents the six combinations that we have to test and their respective conditions. Each codec combination and condition must be evaluated by the same listener from at least four talkers [85, p.22]. This would require the following amount of time for each listener to complete the session test, approximately:

$$6 \text{ combinations} \times 3 \text{ conditions} \times 4 \text{ talkers} \times 30 \text{ seconds} = 2160 \text{ seconds},$$

which corresponds to 36 min. Such amount of time could fatigue the listeners and compromise the quality of the test. Hence, we divided the samples into two groups of listeners, as indicated in Figure D.3, keeping the total duration of evaluation session test inside the recommended range of 20 min to 30 min [9].

To prepare the input speech samples to the DCR test, we designed and followed the flowchart presented in Figure D.4. Each block of the flowchart are described in the following subsections, where we highlight the most important requirements.

Sample Recording

According to ITU-T Recommendation P.800 [85], the speech material should consist of simple, meaningful, short sentences, chosen at random as being easy to understand (from current nontechnical literature or newspapers, for example). Moreover, the recording environment and the talkers should meet the following requirements:

- The talker should be seated in a quiet room with volume from $30\,\mathrm{m}^3$ to $120\,\mathrm{m}^3$. We used a room of $5\,\mathrm{m} \times 5\,\mathrm{m} \times 3\,\mathrm{m} = 75\,\mathrm{m}^3$ for recording all speech samples.
- The talkers must be native speakers of the language in which the tests are being conducted [65]. Hence, in our case, all sentences were recorded in Portuguese.
- The speech material should be recorded by talkers of both genres, and each configuration (i.e., codec switching) should be evaluated by judgments on speech samples generated from at least four talkers. In this thesis, we used four talkers ($N_T = 4$): two male and two female.

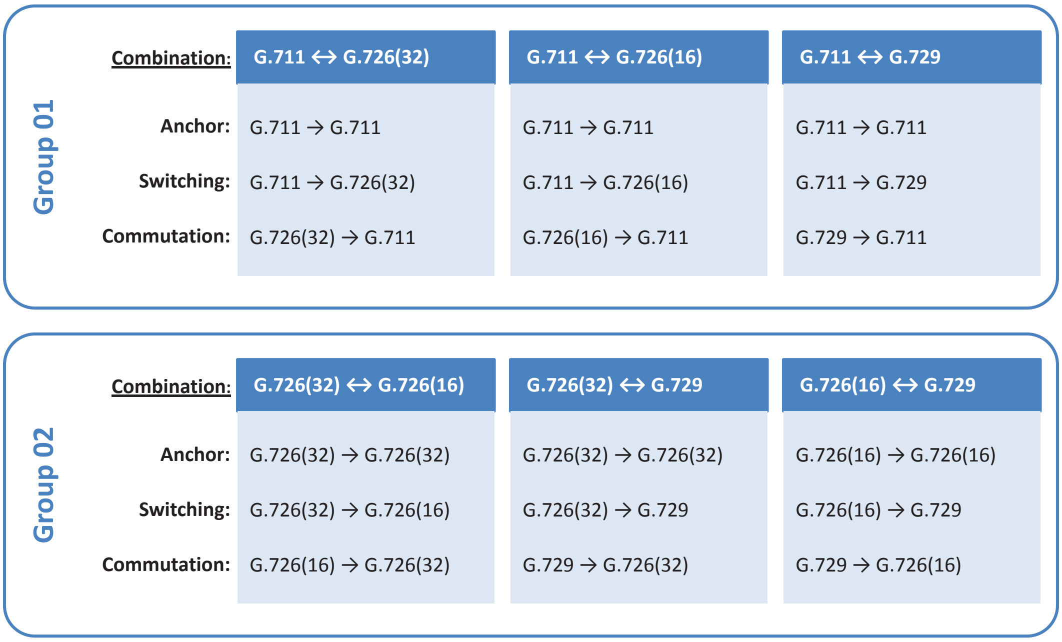


Figure D.3: Test plan of the codec switching combinations.

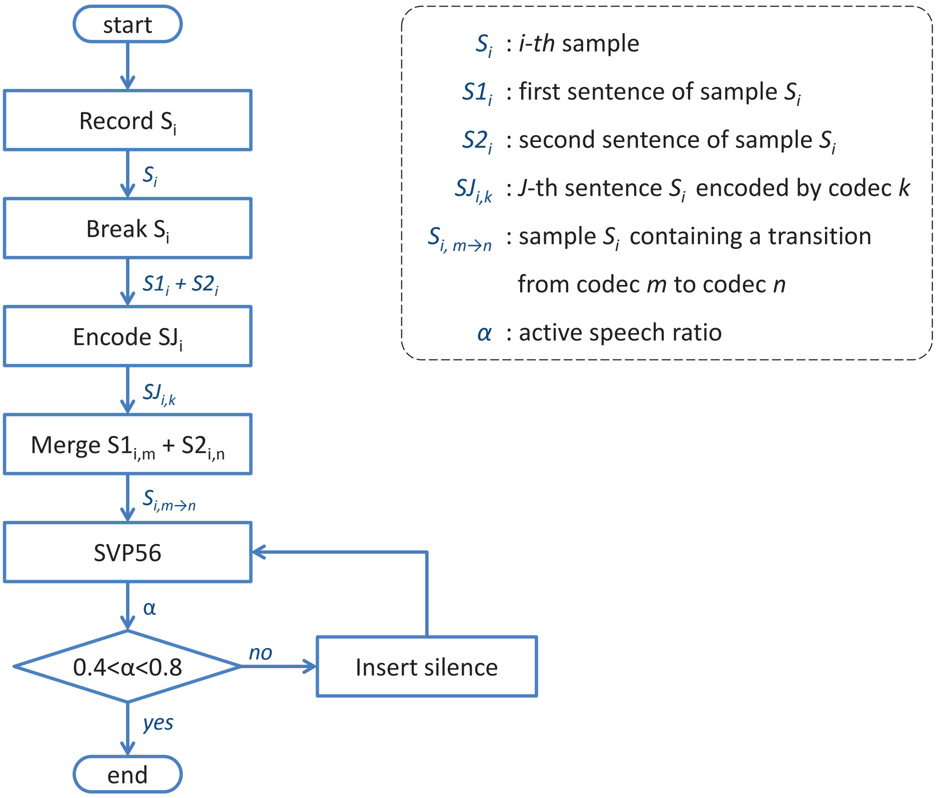


Figure D.4: Flowchart describing the preparation of speech material for DCR tests.

- Talkers should pronounce the sentences fluently but not dramatically, and have no speech deficiencies such as stutter or hoarseness. They should adopt a comfort speaking level that they can maintain fairly constantly.
- The microphone should be positioned between 14 cm and 20 cm from the talker's lips. In some applications, it may be necessary to use a windscreen, which is used if breath puffs from the talker are noticed.
- The activation noise level should be set at $-18\,\mathrm{dB_{FS}}$, and the sampling rate should be 8000 Hz, mono, with 16 bit resolution [9]. In this thesis, we recorded the samples using the software Audacity® [128].

Thus, we recorded 240 phrases, 60 for each talker ($N_S = 60$). The recordings were taken in three different sessions for each talker, to avoid fatigue. The content of this speech material is listed in Appendix I.

After recording the samples following the proper environment, the generated speech samples are ready to be processed by the next steps.

Sample Breaking into Sentences

To understand how phrases were broken into sentences, consider as an example the following phrase (#61 from Table I.2). It is composed of two sentences spoken in Portuguese by the female talker #2.

All he wanted was to be near his family; he has traveled to study abroad for two years.

This phrase was recorded as one single file. Then we used the software Audacity® for manually breaking it into two sentences. The first sentence corresponds to the first line (until "family") and the second sentence to the second line (from "he has").

Sentence Encoding

In this step, we used the softphones `callgen` and `openam` for encoding the sentences with the codecs under study: G.711 (64 kbit/s), G.726 (32 kbit/s), G.726 (16 kbit/s), and G.729 (8 kbit/s), in accordance with the test plan depicted in Figure D.3.

The softphone `callgen` [153] sends prerecorded audio files to another VoIP terminal, using the codec specified in the command line. The softphone `openam` [50] receives and records VoIP calls, acting as an answering machine. Both are built on the top of the OPAL [153].

Sentence Merging into Samples

In this step, we used the software Sox [182] to recombine the two sentence files corresponding to the same phrase into one file again. The resultant files now contain all codec switching combinations and conditions as specified in the test plan of Figure D.3.

Active Speech Ratio

According to Počta et al. [154], the audio samples to be evaluated must have a speech activity ratio (speech/silence ratio) between 40% and 80%. If the speech activity ratio is outside this range, the sample must be edited for inserting or removing silence. This measurement is performed by *svp56*, one of the software tools available at ITU-T Recommendation G.191 [91].

After completing all five steps depicted in the flowchart of Figure D.4, the sample files are ready to be evaluated by the listeners.

D.2.2 Assessment Interface

We designed and implemented a user interface for allowing the subjects to listen to each pair of sample files and to assess the degradation between them. The goal of the interface is twofold: (1) to focus the attention of the listener on the grading process itself, and (2) to facilitate the treatment of data produced at the end of the experiment.

The interface runs in a web browser from a local server. From the perspective of the evaluator (listener), the interface is divided into three stages:

1. *Registration.* Here, the evaluator informs some minimal personal data (full name, gender, age, instruction level), for control purposes.
2. *Training.* Here, the evaluator can get used to the assessment procedure by doing some training tests, which are discarded during the result analysis.
3. *Evaluation.* Here, for each pair of speech samples (reference and target), the evaluator listens to the files, judges the distortion level between them, and assigns a score on the DMOS scale.

Figure D.5 shows the registration screen initially prompted to the listener. Figure D.6 shows the evaluation screen, which is similar to the training screen.

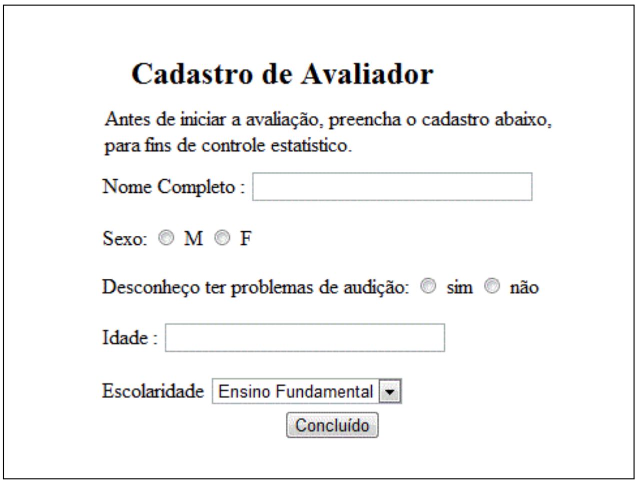


Figure D.5: Registration screen of the assessment interface.

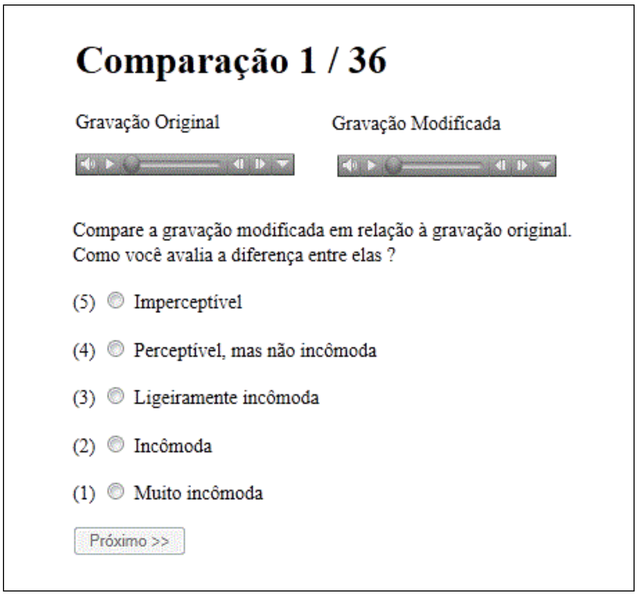


Figure D.6: Evaluation screen of the assessment interface.

D.2.3 Pilot Test

We run a pilot test in order to adjust issues not envisioned during the preparation phase of the DCR tests. The pilot test gave the opportunity to the listeners for familiarizing with the test procedure. Unlike ACR tests, DCR tests require listeners with some experience in auditory tests [65]. Finally, the pilot test offered the opportunity to the facilitators for improving the way they invite listeners to take the tests and how they behave while giving instructions.

The volunteers were taken among students and personnel staff of the Federal University of Amazonas, Brazil, not involved in the experiments. All of them agreed of their own free will to be submitted to the test. Below we describe some comments that they returned:

- The play button (in the user interface) was too small.
- Some kind of token could be rewarded to incentive people to take the tests.
- The color layout was too black-and-white.

D.2.4 Definitive test

After applying the suggestions fed back during the pilot test, we run the definite test with 30 subjects in Group 01 and 30 subjects in Group 02. The age of the 60 participants (19 female and 41 male) covered a range from 18 to 43 years (with an average of 23.9 years). They had, to their own account, normal hearing, and they signed a consent form before taking the tests.

D.3 Result Analysis

The statistical analysis of test results is the final step of the methodology. It aims to identify accurately the average performance of each codec switching combination under test and the statistical significance of any difference among those average performance figures [83]. The latter aspect requires estimation of the variability or variance of the results, such as Student's t-test and ANOVA.

Thus, we checked, by means of hypothesis testing, the four properties presented in Section D.1.2, as shown in the next subsections.

D.3.1 Reliability of a listener's judgment

The reliability check aims to identify which listeners are able to assign a DMOS=5 to the anchor samples ($C_i \to C_i$ transitions). Naturally, we do not expect that listeners assign DMOS=5 for all anchor samples. Hence, we run a statistical test to verify which listeners could do it on the *average*.

Thus, we applied a two-sided t-test for checking the null-hypothesis H_0^R (p. 71). The resultant p-values are reported in Table D.1 for each listener. We rejected the listeners with $p \leqslant 0.01$, so that only 42 out of 60 subjects were considered "reliable". The label "reliable" should be understood in terms of DCR test purposes, rather than in some moral or health sense.

To better understand the meaning of the p-value in the context of the reliability check, take listeners L_3, L_{23} and L_{37} as example. They graded each of the twelve anchor samples presented to them with DMOS=5. Thus, the t-test on their reliability returned a p-value of 1.000. On the other hand, take listener L_1, who assigned DMOS=5 to only two out twelve and DMOS=2

Table D.1: p-values of the reliability check.

Listener	p-value	Listener	p-value	Listener	p-value	Listener	p-value
01	< 1.0e-3	16	0.054	31	0.082	46	0.082
02	0.015	17	< 1.0e-3	32	0.005	47	0.026
03	1.000	18	0.096	33	0.016	48	0.008
04	0.006	19	0.004	34	0.044	49	0.167
05	0.003	20	0.039	35	0.341	50	< 1.0e-3
06	0.166	21	0.337	36	0.008	51	< 1.0e-3
07	0.337	22	0.009	37	1.000	52	0.339
08	0.104	23	1.000	38	0.001	53	0.339
09	< 1.0e-3	24	0.027	39	0.339	54	0.002
10	0.026	25	0.039	40	0.082	55	0.006
11	0.082	26	0.013	41	0.082	56	0.191
12	0.166	27	0.028	42	0.009	57	0.071
13	0.341	28	0.339	43	0.166	58	0.039
14	0.029	29	0.339	44	0.166	59	0.001
15	0.046	30	0.012	45	0.082	60	0.002

Table D.2: Degradation matrix partially filled with initial values.

	To			
From	G.711	G.726(32)	G.726(16)	G.729
G.711	5.0			
G.726(32)		5.0		
G.726(16)			5.0	
G.729				5.0

to four out twelve anchor samples. The resultant p-value was very small (less than 0.001). Consequently, the other grades assigned by listener L_1 were not considered in the other checks.

Now, we can start to build the degradation matrix for the codec set $\mathcal{C} = \{$G.711, G.726(32), G.726(16), G.729$\}$. From Definition 3 (p. 68), we sketch the degradation matrix presented in Table D.2 by filling the main diagonal with 5.0. The other elements are determined by the following checks.

D.3.2 Perceptibility of codec switching

The perceptibility check aims to verify, for each codec C_i in the set, whether the listeners are able to perceive switchings from it to all other codecs, C_i included. In statistical terms, we should check whether, given a codec C_i, there is a significant difference among the grades assigned by the listeners to all switchings $C_i \rightarrow C_j, j \in [1, 4]$. Thus, for each of the four codecs in the set, we performed an ANOVA to verify the null-hypothesis H_0^P (p. 71).

Table D.3 shows the results of the ANOVA tests for the perceptibility check of each evaluated codec. In the second column, the component "switchings" indicates all switchings starting from the codec indicated in the first column. For example, taking G.711 as starting codec, we run an ANOVA test to determine whether the DMOS grades assigned to the switchings G.711→G.711, G.711→G.726(32), G.711→G.726(16) and G.711→G.729 were significantly different.

The results show that the difference among DMOS values assigned to the switchings starting from G.726(16) only is not statistically significant ($p = 0.2155$). A naive reading of this result could claim that listeners did not perceive when the codec was switched from G.726(16) to another one. However, the fact is that G.726 (16 kbit/s) offers the worst absolute speech quality among the four codecs of the set. Although its bitrate is higher than the one of G.729 (8 kbit/s), it has an I_e factor of 40, whereas G.729 has an I_e factor of 10. Furthermore, G.729 is a hybrid codec, so it discards information less sensitive to the human auditory system, in contrast to waveform codecs, such as G.726. G.726 (32 kbit/s) is not a bad codec because it only halves the bitrate of G.711, the best-quality narrowband codec.

Therefore, switchings from G.726(16) to any other codec in the set are perceived by subjects as an improvement. On the other hand, the DMOS scale does not measure improvements in speech quality, but only degradations. Thus, listeners tend to assign the highest value in DMOS scale to the switchings starting from G.726(16).

Consequently, as shown in Table D.4, we fill the row corresponding to these switchings in the degradation matrix with 5.0, because there is no significant difference among these switchings and the anchor condition of G.726(16). Regarding to the switchings starting from G.711, G.726(32) and G.729, there is no evidence to support that each of the three groups is equally perceived ($p < 2.2 \cdot 10^{-16}$). Thus, they should be further checked in regard to homogeneity.

Table D.3: ANOVA table for the perceptibility check.

Starting codec	Component	Degrees of freedom	Sum of squares	Mean square	F-value	p-value
G.711	switchings	3	204.89	68.297	156.11	< 2.2e-16
	residuals	406	177.62	0.437	—	—
G.726(32)	switchings	3	128.87	42.955	89.85	< 2.2e-16
	residuals	311	148.68	0.478	—	—
G.726(16)	switchings	3	4.15	1.384	1.50	0.2155
	residuals	355	328.69	0.926	—	—
G.729	switchings	3	249.73	83.243	168.88	< 2.2e-16
	residuals	424	209.00	0.493	—	—

Table D.4: Degradation matrix partially filled with the results of perceptibility check.

	To			
From	G.711	G.726(32)	G.726(16)	G.729
G.711	5.0			
G.726(32)		5.0		
G.726(16)	5.0	5.0	5.0	5.0
G.729				5.0

D.3.3 Homogeneity of codec switching perception

The homogeneity check aims to verify whether the switchings starting from a given codec to the other codecs in the set, excluding itself, are equally perceived. Note that the anchor samples $C_i \to C_i$ are considered in perceptibility check, but not in the homogeneity check.

Thus, we performed an ANOVA to check the null-hypothesis H_0^H (p. 72). Table D.5 shows the results of the ANOVA tests for homogeneity check of the switchings starting from G.711, G.726(32) and G.729.

The results in the rightmost column of Table D.5 show that the DMOS values assigned to the switchings from G.711 to G.726(32), G.726(16) and G.729 are significantly different from each other ($p < 2.2 \cdot 10^{-16}$). The same conclusion holds for the switchings starting from G.726(32) and G.729 as well. This means that the degradation matrix of this codec set should be filled with the average DMOS values for each codec switching taken individually, as shown in Table D.6.

Table D.5: ANOVA table for the homogeneity check.

Starting codec	Component	Degrees of freedom	Sum of squares	Mean square	F-value	p-value
G.711	switchings	2	151.15	75.574	133.58	< 2.2e-16
	residuals	236	133.51	0.566	—	—
G.726(32)	switchings	2	109.97	54.986	130.30	< 2.2e-16
	residuals	237	100.01	0.422	—	—
G.729	switchings	2	171.08	85.540	124.93	< 2.2e-16
	residuals	264	180.76	0.685	—	—

Table D.6: Degradation matrix partially filled with the results of homogeneity check.

	To			
From	G.711	G.726(32)	G.726(16)	G.729
G.711	5.0	4.7	2.8	4.5
G.726(32)	4.8	5.0	3.3	4.5
G.726(16)	5.0	5.0	5.0	5.0
G.729	4.6	4.5	2.9	5.0

D.3.4 Commutability of codec switching perception

The commutability check aims to verify whether a given codec switching from codec A to B is perceived in the same way as from codec B to A. In statistical terms, we should verify whether there is a significant difference between the DMOS values assigned to the $A \to B$ transition and the ones assigned to the $B \to A$ transition. This can be accomplished by a simple t-test or by an ANOVA on the null-hypothesis H_0^C (p. 72); both lead to the same results. We chose ANOVA for the convenience of reusing the code of our statistical tool (the R) and presenting the results of all three checks in the same table format.

Table D.7 shows the output the ANOVA tests for the commutability check of each of the six combinations in the codec set. We verified that there is no significant difference between the DMOS values assigned to the switchings G.711→G.726(32) and G.726(32)→G.711 ($p = 0.2684$). The same holds for the switchings between G.711→G.729 and G.729→G.711 ($p = 0.3755$), and between G.726(32)→G.729 and G.729→G.726(32) ($p = 0.6846$).

Thus, it means that the switchings between the codecs of each of these three pairs can be considered as commutable. The definitive value of their degradation factors in the degradation matrix is determined by the mean between the direct and the reverse DMOS. For example, in Table D.6, the G.711→G.729 switching (direct) had a DMOS=4.468085, and the G.729→G.711 switching (reverse) had a DMOS=4.568421. Because these two transitions are commutable, their definitive DMOS values are set as the mean of both individual DMOS values (4.518519), as expressed in Table D.8.

On the other hand, Table D.7 shows that there is no evidence to consider all switchings involving G.726(16) as commutable ($p \ll 0.05$). As seen earlier in the perceptibility check, G.726(16) presents the worst absolute speech quality among the tested codecs. So, switchings starting from it are considered as improvements, but switchings terminating in it are considered as deteriorations.

Therefore, the definitive degradation matrix for the codec set is given in Table D.8. Note

Table D.7: ANOVA table for the commutability check.

Codec pair	Component	Degrees of freedom	Sum of squares	Mean square	F-value	p-value
G.711↔G.726(32)	switchings	1	0.336	0.336	1.234	0.2684
	residuals	144	39.151	0.272	—	—
G.711↔G.726(16)	switchings	1	41.174	41.174	60.701	1.276e-12
	residuals	142	96.319	0.678	—	—
G.711↔G.729	switchings	1	0.476	0.476	0.789	0.3755
	residuals	187	112.710	0.603	—	—
G.726(32)↔G.726(16)	switchings	1	13.962	13.962	18.660	2.521e-05
	residuals	189	141.410	0.748	—	—
G.726(32)↔G.729	switchings	1	0.084	0.084	0.166	0.6846
	residuals	141	71.650	0.508	—	—
G.726(16)↔G.729	switchings	1	36.966	36.966	52.543	9.874e-12
	residuals	193	135.783	0.704	—	—

Table D.8: Definitive degradation matrix for the codec set used in this thesis.

	To			
From	G.711	G.726(32)	G.726(16)	G.729
G.711	5.0	4.7	2.8	4.5
G.726(32)	4.7	5.0	3.3	4.5
G.726(16)	5.0	5.0	5.0	5.0
G.729	4.5	4.5	2.9	5.0

that the column where G.726(16) is the destination codec of the switchings contains the worst DMOS values. This means that switchings starting from any other codec of the set to G.726(16) are not a good option (excluding the G.726(16) itself, of course).

Now, remember that the primary objective of codec switching is to reduce bandwidth consumption. Thus, if a high bitrate codec is in use (e.g., G.711) and network conditions are not satisfactory, then it is better to change to a low bitrate codec (e.g., G.729), but with low degradation due switching (i.e., high DMOS) and acceptable speech quality (i.e., low I_e), than to change to an intermediate bitrate codec (e.g., G.726(16)), but with high degradation due switching (i.e., low DMOS) and poor speech quality (i.e., high I_e).

Therefore, an adaptation mechanism should not consider the bitrate alone before deciding on codec switching. The degradation and I_e factors must be also taken into account. The evidence from subjective tests here presented strongly supports that listeners can perceive quality degradation when the codec is switched to a medium-bitrate codec such as G.726(16), but not for a low-bitrate codec with superior speech quality, such as G.729.

Such evidence contradicts the claim that "codecs with higher transmission rates have better quality," explicitly or implicitly found in some works in the literature, such as Chen et al. [32], Costa & Nunes [41], Galiotos et al. [57], Roychoudhuri & Al-Shaer [166], and Trad et al. [193]. At the same time, the DCR tests that we carried out confirm the design choice of Myakotnykh & Thompson [143] to restrict the codec set to only G.711, G.726(32) and G.729, and their suspicion about how codec switching may affect user's feeling about a call.

D.4 Summary

In this chapter, we developed and presented a general methodology for accounting for the speech quality degradation perceived at the exact moment of the switching between two codecs belonging to a softphone's codec set. This methodology is based on the Degradation Category Rating (DCR), a subjective test for speech quality assessment. Next, we detailed how this methodology was applied to a codec set of four elements: G.711 (64 kbit/s), G.726 (32 kbit/s), G.726 (16 kbit/s), and G.729 (8 kbit/s).

The results of the DCR tests evidenced that degradation during codec switching must be considered by an adaptation mechanism, because low-bitrate but high-quality codecs can be better accepted than medium-bitrate but low-quality ones. Furthermore, the perceptibility of codec switching is not commutable, because transitions from a high- to a low-quality codec are regarded as deterioration and the converse is regarded as improvement.

Appendix E

Design of an Adaptive VoIP Solution

Remember that all models are wrong;
the practical question is how wrong
do they have to be to not be useful.
George Edward Pelham Box (1919–)

Contents

An adaptive mechanism can adjust four parameters at the sender endpoint: codec algorithm, bitrate, packetization and redundancy. In this thesis, we chose to tackle the challenges offered by VoIP adaptation based on codec switching, specifically. In the previous chapter, we developed and applied a methodology for determining what users perceive at the exact moment of codec transition.

In this chapter, we design three adaptation strategies for managing the speech quality of VoIP call in run-time. During this task, we tackle the following open problems, as listed in Chapter A:

- Adaptive solutions that use encoding switching as an adjustable parameter underpin their decisions on off-line measurements among a restricted list of codecs. If a new codec should have to be included in this list, then new off-line measurements must be run in order to rearrange the precedence among encoding configurations. Thus, it is necessary to design a flexible framework that explores the trade-offs among encoding, end-to-end delay, packet loss and quality, based on standard parametrization of codec algorithms. This is developed in Section E.1.
- During the design of systems with adaptive properties, layers of abstractions are usually specified to hide complexity, but this practice frequently also hides the control loops [177].

The resulting lack of visibility makes it easy to neglect the control aspects in design, which is so critical for validation and verification. Thus, an *explicit* modeling of the control loop is needed to engineer adaptive VoIP systems properly. This is developed in Section E.2.

- There is no study that evaluates the best placing of the MAPE agents in a decentralized VoIP system. Because of network delay and packet loss probability, sender and receiver endpoints must exchange a minimal, but effective, set of feedback and control messages in order to report and apply adaptations. Section E.3 tackles this open problem.

Finally, in Section E.4, we list and describe the properties that a self-adaptive system should exhibit. Chapter F addresses the experimental design needed for testing such properties on the proposed adaptive VoIP solution. Figure E.1 is a fragment of Figure A.5 (p. 31); It depicts how the issues described in sections E.1 to E.4 are interrelated to one another, as part of an iterative development process.

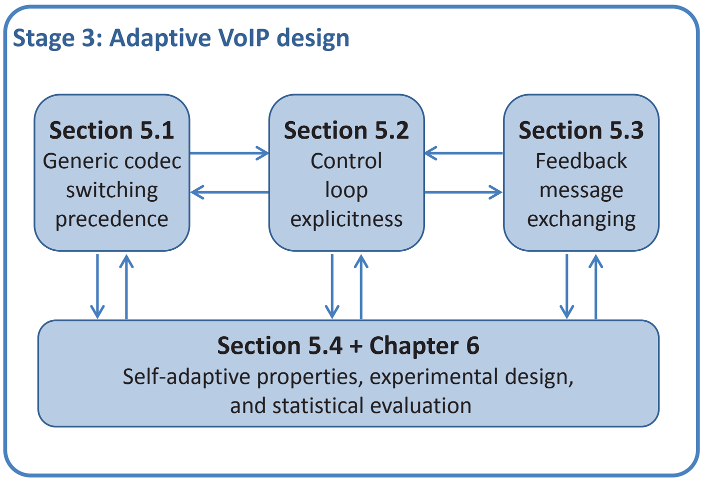


Figure E.1: Overview of interrelationship among the sections of this chapter.

E.1 General Codec Precedence for the Switching Decision

In a sector dominated by standardization, such as the telecommunications, integration among different manufacturers is a major concern. In this sense, proposals of adaptive VoIP terminals may become useless if they are not able to adjust parameters when distinct softphone implementations are used by the talkers.

Thus, a first challenge towards such integration is to modify a softphone so that it can determine dynamically a codec precedence list based on the set of common codecs for each new call. In this case, the decision of switching the codec could be more flexible across multiple softphone implementations, and would not be restricted to the codec set arbitrarily chosen during design. In this section, we tackle this challenge. The starting point is to elect one or more criteria for determining a codec precedence list to be consulted during the codec switching decision.

Most part of the VoIP adaptation mechanisms based on codec switching reviewed in Chapter C take for granted that speech quality is directly proportional to the codec bitrate [32, 41, 57, 166, 193]. This assumption can be valid for waveform codecs that do not implement VAD or PLC techniques, but not for hybrid codecs, specifically designed for packet-based transmission, such as iLBC, AMR and Speex.

Moreover, the influence of codecs on speech quality is not limited to only impairments due to low-bitrate compression. The robustness of a codec to packet loss impacts on speech quality

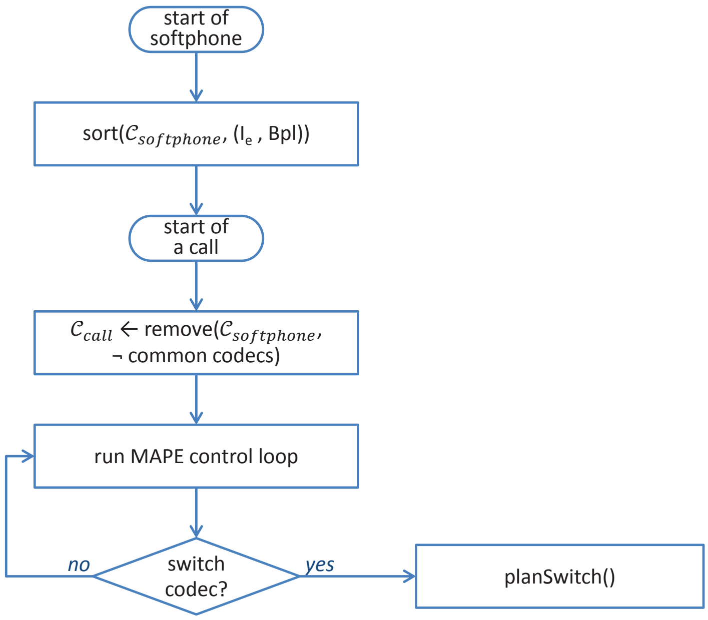


Figure E.2: Flowchart of the codec precedence sorting for a possible codec switching adjustment during the managed call.

also. As already seen in Section B.3.3, the E-model computes the influence of low bitrate compression by means of the I_e factor, and the influence of robustness to packet loss by means of the *Bpl* factor. Therefore, we used these two factors for establishing a precedence order among the codecs implemented in the adaptive VoIP terminal.

As indicated by the flowchart in Figure E.2, after initiated, the softphone sorts its codec set $\mathcal{C}_{\text{softphone}}$ by decreasing order of bitrate, as a first criterion; then by the I_e factor, as a second criterion; and finally by the *Bpl* factor as a third and last criterion. These criteria were determined based on Equation B.4, in which the I_e factor has more impact than the *Bpl* factor.

Continuing the flowchart, if a call is established, then the planning agent creates the codec set of that call ($\mathcal{C}_{\text{call}}$), by removing the codecs that are not common to both VoIP terminals from the softphone's sorted set $\mathcal{C}_{\text{softphone}}$. The first codec in the sorted codec set $\mathcal{C}_{\text{call}}$ is denoted by C_1, and it has the highest bitrate. The last codec is denoted by C_{N_C}, and it has the lowest bitrate amongst the codecs in $\mathcal{C}_{\text{call}}$, where N_C is the number of elements of $\mathcal{C}_{\text{call}}$.

Now, the MAPE control loop can be started for managing the speech quality of the newly established call. This control loop is detailed in the next section. For now, it suffices to know that if the control loop determines that a codec switching is needed, then it must call the function *planSwitch()*, specified in Algorithm 1.

Algorithm 1 basically requires three arguments as input:

1. $\mathcal{C}_{\text{call}}$, the codec set of the ongoing call.
2. C_i, the codec currently under use, which occupies the i-th position in $\mathcal{C}_{\text{call}}$.
3. *step*, which is the number of positions to be increased or decreased in $\mathcal{C}_{\text{call}}$.

To understand the meaning of the argument *step*, consider the following example. If network conditions are favorable, then the planning agent switches the current codec to one with a higher bitrate. Hence, it sets $step = +1$, in the sense that it intends to move one step up in the bitrate ranking, which means to decrease one position in $\mathcal{C}_{\text{call}}$. Otherwise, if network conditions are not favorable and speech quality is below acceptable thresholds, then the planning agent

Algorithm 1 Function *planSwitch()*.

Require: $\mathcal{C}_{\text{call}}$, C_i, *step*

1: $j \leftarrow i + step$
2: $k \leftarrow j - 1 \times \text{sgn}(step)$
3: **if** $(j \geqslant 1$ **and** $j \leqslant N_C)$ **then**
4: **if** $(k \geqslant 1$ **and** $k \leqslant N_C)$ **then**
5: **if** $((\delta_{i,j} \geqslant 3$ **and** $\delta_{i,k} \geqslant 3)$ **or** $(\delta_{i,j} < 3$ **and** $\delta_{i,k} < 3))$ **then**
6: **if** $(I_{\text{e},k} > I_{\text{e},j})$ **then**
7: applySwitch$(C_i \rightarrow C_k)$
8: **else if** $(I_{\text{e},k} = I_{\text{e},j})$ **then**
9: **if** $(Bpl_k > Bpl_j)$ **then**
10: applySwitch$(C_i \rightarrow C_k)$
11: **else**
12: applySwitch$(C_i \rightarrow C_j)$
13: **end if**
14: **else**
15: applySwitch$(C_i \rightarrow C_j)$
16: **end if**
17: **else if** $(\delta_{i,j} \geqslant 3)$ **then**
18: applySwitch$(C_i \rightarrow C_j)$
19: **else if** $(\delta_{i,k} \geqslant 3)$ **then**
20: applySwitch$(C_i \rightarrow C_k)$
21: **end if**
22: **else**
23: applySwitch$(C_i \rightarrow C_j)$
24: **end if**
25: **else**
26: **return** Switching not possible.
27: **end if**

switches the current codec to another with lower bitrate. Hence, it sets $step = -1$ or $step = -2$, depending on the severity of the impairments, in the sense that it intends to move one or two steps down in the bitrate ranking, which means to increase one or two positions in $\mathcal{C}_{\text{call}}$.

Before applying codec switching, function *planSwitch()* verifies whether the codec next to the target codec would be a better alternative. For example, if the planning agent decides to lower the bitrate by switching from codec C_i to C_{i+1}, then the function *planSwitch()* checks if C_{i+2} would not be a better alternative. So, it first compares their switching degradation factors $\delta_{i,i+1}$ and $\delta_{i,i+2}$ (line 5). If both are considered as acceptable (DMOS above 3) or unacceptable (DMOS below 3), then it compares the equipment impairment factors $I_{\text{e},i+1}$ and $I_{\text{e},i+2}$ (line 6). If there is no winner yet, then it compares the *Bpl* values of C_{i+1} and C_{i+2} to decide which one is the best alternative (line 9).

Therefore, function *planSwitch()* does not blindly switch the codec based on the bitrate. Instead, it considers three parameters that affect user's perception about speech quality: the degradation at the exact moment of codec switching (δ_{ij}), the impairment factor due to low-bitrate encoding (I_{e}), and the codec-specific robustness factor to packet loss (*Bpl*).

Function *planSwitch()* also checks whether it really exists a higher or lower bitrate codec to switch to. This is done in lines 3 and 4, where the variables j and k are checked to be in the range of permitted indices of $\mathcal{C}_{\text{call}}$.

Furthermore, it should be emphasized the difference between the right arrow ($\rightarrow$) and the left arrow (←) in Algorithm 1. The → symbol indicates the direction of the codec switching (e.g., from codec C_i to C_j), whereas the ← symbol indicates the attribution of a value to a variable.

The function *applySwitch()*, called inside *planSwitch()*, is performed by the execution agent. It issues a reINVITE SIP request for changing the current codec to the one specified in its argument.

In line 26, function *planSwitch()* returns an error message to the planning agent, instead of calling *planSwitch*($C_i \rightarrow C_i$), which would overload the network with signaling messages for establishing a new session just for keeping the same codec currently in use.

We implemented the function *planSwitch()* in the softphone PJSUA, by modifying its source code. It is written in C and runs on the top of the PJSIP project [155]. To use the codecs G.726 and G.729, it is required to install the Intel IPP library [82].

E.2 Visibility of the Control Loop

According to Müller et al. [35, 141], the control loop is a crucial feature that determines the feedback behavior of a self-adaptive system. Hence, it should be elevated to a first-class entity during modeling, design, and implementation cycles. Also, they point that the idea of increasing the *visibility* (or *explicitness*) of control loops is not new. It can be dated as late as 1995, when Shaw [177] introduced a new software organization paradigm based on control loops, with an architecture dominated by the analysis of a classic feedback loop.

Therefore, to ensure a proper design and evaluation of self-adaptive software systems, we need models for representing important aspects, such as user needs, environment characteristics, and other system properties [35]. In this thesis, we used the models of self-adaptive software characterization proposed by Andersson et al. [3] and Villegas et al. [198] to design an adaptive solution for managing a VoIP call between two terminals.

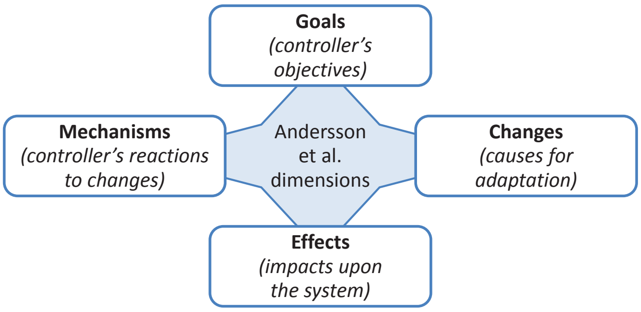
Andersson and colleagues' four-dimension model, summarized in Figure E.3(a), focuses on *specification* attributes. This work has been ratified by the community of Software Engineering for Self-Adaptive Systems in events such as the annually Dagstuhl Seminars [35, 46]. On the other hand, Villegas and colleagues' eight-dimension model, summarized in Figure E.3(b), focuses on *evaluation* attributes. This work is more recent, but it was successfully applied for characterizing sixteen different approaches to self-adaptive systems in the literature. Despite their differences, we combined these two models here.

In Chapter C, we evidenced the control loop that lies hidden in adaptive VoIP solutions found in the literature. In this section, we design an adaptive VoIP solution, with the explicit focus on the control loop, according to the guidelines proposed by Andersson et al. [3] and Villegas et al. [198]. It is organized as follows.

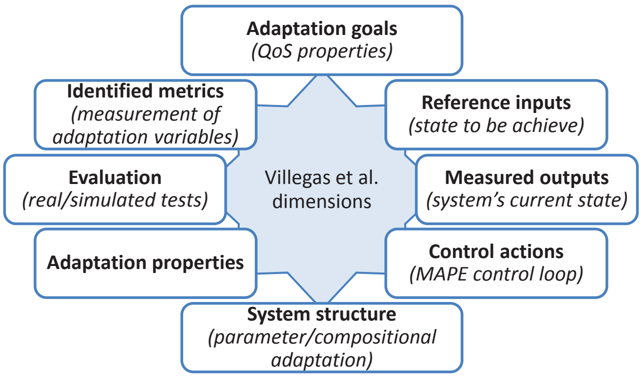
In Subsection E.2.1, we specify the goals an adaptive VoIP call should achieve in the terms of the *goals* dimension of Andersson et al. and the *adaptation goal* dimension of Villegas et al.

In Subsection E.2.2, we list VoIP parameters to adaptation variables in th terms of the *changes* and *mechanisms* dimensions of Andersson et al., and the *reference inputs* and *measured outputs* dimensions of Villegas et al.

Finally, in Subsection E.2.3, we make the control loop explicit on the dynamics of a call between two VoIP terminals in th term of the *mechanisms* dimension of Andersson et al., and the *control actions* and *system structure* dimensions of Villegas et al. Note that the *evaluation* and *identified metrics* dimensions of Villegas et al., and the *adaptation properties* dimensions of Andersson et al. are covered in the next Section E.4.



(a)



(b)

Figure E.3: Overview of the classification dimensions for self-adaptive systems proposed by (a) Andersson et al. [3], focused on *specification*; and (b) Villegas et al. [198], focused on *evaluation*.

E.2.1 Goal Specification

Recalling the definition given by Weyns et al. [201], self-adaptive software systems are "a class of systems that adjust their behavior at run-time to achieve certain functional or quality of service *objectives*." Hence, prior to specifying which adjustments should be taken and in which boundary conditions those adjustments should be applied, we have to define which *objectives* (or *goals*) should be met by those adjustments, because they are the main motivation (or justification) for a system to be adaptive [198].

Thus, the main goal to be pursued by our adaptive VoIP terminal is the following:

Goal 1:
The MOS of an ongoing call might be higher than or equal to 3.6.

According to Villegas et al. [198], one form of adaptation goal definition is by means of preservation of specific QoS properties. Goal 1 fits in this requirement, because it relies on the lower MOS bound for a call to be considered acceptable for toll-quality [94]. Moreover, the MOS value, as determined by the E-model, summarizes not only network-centric parameters, but also user-centric perception about speech quality.

Another goal to be met by the VoIP adaptation mechanism is the following:

Goal 2:
Adaptation of adjustable parameters might not decrease the current MOS value.

Note that Goal 2 above implies that a decrease in some observation parameters is permitted, since the overall call quality does not decrease too.

To close this subsection, Table E.1 presents the attributes of the *goals* dimension in the classification proposed by Andersson et al. [3]. In the leftmost column, we characterize the degree of Goals 1 and 2 according to their criteria.

Table E.1: Characterization of Goals 1 and 2 in terms of the classification proposed by Andersson et al. [3].

Attribute	Definition	Degree
Evolution	whether the goals can change within the lifetime of the system	*static* (they are not expected to change at run-time)
Flexibility	pliability of the goals in the way they are expressed	*unconstrained* (their statements provide flexibility for dealing with uncertainty)
Duration	validity of a goal through the system lifetime	*persistent* (they are valid for all call duration)
Multiplicity	number of goals associated with the self-adaptability aspects of a system	*multiple* (there is more than one goal)
Dependency	how the goals are related to each other	*dependent* and *complementary* (Goal 2 bounds the set of adjustments needed for achieving Goal 1)

E.2.2 Identification of Adaptation-Related Variables

As recommended by Shaw [177] and Villegas et al. [198], after specifying the adaptation goal, we should identify the key variables of a VoIP call related to the adaptation process. In Section C.1.1 (p. 48), we described the variables related to a system managed by a control mechanism, and gave an example in the VoIP context. Now, we have to identify the variables to be handled by our proposal of adaptive VoIP mechanism.

As proof of concept, we used codec switching as adjustment parameter at the sender endpoint. Hence, we identified the variables related to this adjustment only. Note that the contribution of this thesis is not focused on implementing several adjustment parameters of the RTP flow, because the literature is plenty of this, as already seen in Chapter C. Rather, our contribution resides in bringing to surface self-adaptive concepts for formally designing adaptive VoIP systems. Therefore, we picked just one adjustment parameter – codec switching – as demonstration.

Table E.2 shows the variables related to VoIP adaptation by means of codec switching. First, we identified the observation parameters, which are tracked by the monitoring agent when it looks for changes. Observation parameters can be divided into two groups, depending on their origin: (1) those collected at the VoIP terminals (both sender and receiver), and (2) those collected from measurements at the network. Both are processed by the analysis agent and converted into decision metrics.

From Goal 1, we take MOS as a decision metric. In each run of the control loop, the analysis agent should track two measurements of MOS: *instantaneous* MOS and *cumulative* MOS. The instantaneous quality $\mathrm{MOS_{inst}}$ refers to the speech quality between two MOS calculations, that is, to the present measurement time i. The cumulative quality $\mathrm{MOS_{cum}}$ refers to the speech

Table E.2: Mapping of adaptation-related variables to parameters of a VoIP call.

Variable type	VoIP parameter	
Observation parameters	@Terminal	current codec, T_{codec}, T_{buffer}, Ppl_{buffer}, I_{e}, Bpl, Δ_C
	@Network	T_{net}, Ppl_{net}, PLB
Decision metrics	$\mathrm{MOS_{inst}}$, $\mathrm{MOS_{cum}}$	
Performance references	MOS thresholds	
Adjustable parameters	@Sender	codec algorithm
	@Receiver	dejitter buffer length

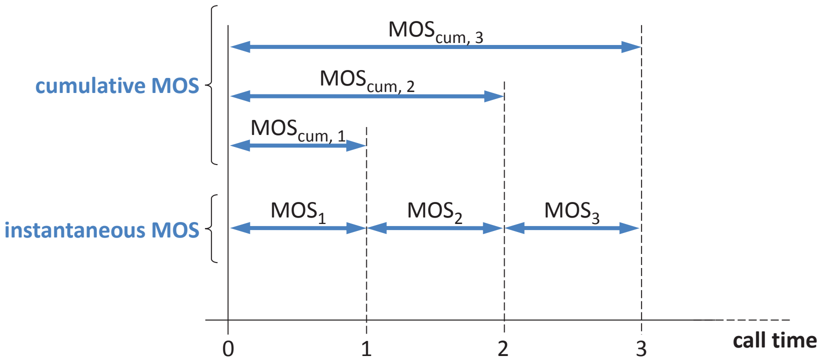


Figure E.4: Instantaneous and cumulative speech quality measurement.

quality since the beginning of the managed call. Figure E.4 depicts the relationship between these two measurements.

The value of MOS_{cum} is determined as a weighed sum of the previous MOS_{inst}. It reflects the recency effect [66], by which recent quality levels have higher weights on the overall perceived quality than quality levels at the beginning of the call. In this thesis, we employed the perceptual model of Rosenbluth [165] for determining the MOS_{cum}, instead of the gap-burst exponential approach described by Clark [40], which we employed in previous works [27, 116]. Rosenbluth's model is more suitable for run-time monitoring, was justified by subjective experiments performed by AT&T [143].

In contrast, Clark's model defines the gap state as being the receiving of at least G_{min} consecutive packets. This demands a prior knowledge of packet loss behavior, which is not available at the measurement time. For example, if the analysis agent is observing the RTP flow during a burst state and receives $n < G_{\text{min}}$ consecutive packets, then it cannot tell whether those packets belong to the current burst state, or they constitute a new gap state.

Rosenbluth's model determines the cumulative MOS by the following expression [143]:

$$\text{MOS}_{\text{cum}} = \frac{\sum_i W_i \cdot \text{MOS}_i}{\sum_i W_i}, \text{ where} \tag{E.1}$$

$$W_i = \max\left[1, 1 + (0.038 + 1.3 \cdot {L_i}^{0.68}) \cdot (4.33 - \text{MOS}_i)^{(0.96+0.61 \cdot {L_i}^{1.2})}\right] \tag{E.2}$$

where MOS_i is the instantaneous MOS during the measurement period, and L_i is a location of a degradation period. This latter is measured on 0-to-1 scale, where 0 indicates the beginning of a conversation, and 1 indicates the measurement time. Hence, this parameter changes proportionally to time from the beginning of a call.

Back to Table E.2, the decision metrics MOS_{inst} and MOS_{cum} should be compared with performance references, that is, thresholds used for triggering a change request from the analysis agent to the planning agent.

Finally, when changes in MOS are detected, this may cause the system to self-adapt. In this case, the planning agent blueprints a change plan over the adjustable parameters. These latter can be located at the sender or at the receiver endpoint.

To close this subsection, we characterize the *observation parameters* related to the adaptation of an ongoing call in terms of the *changes* dimension attributes in Andersson's et al. classification [3] on Figure E.3(a):

1. *Source.* It identifies the origin of the change.

   As already identified in Table E.2, the network delay T_{net}, network packet loss Ppl_{net}, and packet loss behavior (PLB) are *external* to the VoIP terminals. On the other hand, the following parameter are *internal* to a system composed by two VoIP terminals engaged in a call: dejitter buffer delay T_{buffer}, packet discard Ppl_{buffer}, degradation matrix $\Delta_{\mathcal{C}}$, and current codec under use, which determines the codec delay T_{codec} and the I_{e} and *Bpl* factors.

2. *Frequency.* It concerns with how often a particular change occurs.

   The observation parameters T_{net}, Ppl_{net}, T_{buffer}, Ppl_{buffer} and PLB are expected to change *frequently* during the call. On the other hand, the current codec under use and, consequently, T_{codec}, I_{e} and *Bpl*, are expected to change *occasionally*, only if the planning agent switches the encoding scheme. Finally, $\Delta_{\mathcal{C}}$ is expected to change *rarely*, only after some software upgrade of the VoIP terminals.

3. *Anticipation.* It captures whether a change can be predicted ahead of time.

   Although faults are undesirable, changes in T_{net}, Ppl_{net}, T_{buffer}, Ppl_{buffer} and PLB are considered as *foreseen*, because impairments are expected to occur in the underlying network of a VoIP call.

E.2.3 Explicit Identification of the Control Loop

After identifying the variables related to VoIP adaptation, we should specify how the system will react toward changes, that is, we should characterize the adaptation process itself in the MAPE loop context [3, 198].

Figure E.5 shows our proposal for making the control loop explicit in a VoIP call between two terminals, considering only variables available at the application layer. A typical VoIP call comprises two RTP flows, but we restrict our explanation to only one flow for the better understanding of it. Unless indicated, we refer to the sender endpoint of the RTP flow as *sender*, and to the receiver endpoint of the RTP flow as *receiver*.

In Figure E.5, both sender and receiver endpoints run their own instance of the MAPE control loop. At the receiver, the monitoring agent (M_R) may send symptoms to the analysis agent located at the receiver (A_R), as well as to the analysis agent located at the sender (A_S).

The analysis agents (A_S or A_R) convert the symptoms (observation parameters) into decision metrics. If decision metrics ($\mathrm{MOS}_{\mathrm{CQE}}$) do not meet performance references (thresholds), then the receiver's analysis agent (A_R) sends a change request to the receiver's planning agent (P_R) or to the sender's planning agent (P_S). Finally, the receiver's planning agent (P_R) may send a change plan to the receiver's execution agent (E_R) or to the sender's execution agent (E_S).

If the E_R has adjustments to perform, it can be designed in three ways, depending on the localization of the effectors:

1. To adjust receiver's adjustable parameters, such as the dejitter buffer length;
2. To adjust sender's adjustable parameters, such as the codec algorithm; or
3. To adjust the adjustable parameters of the sender endpoint of its own terminal, which originates the RTP counterflow of the VoIP call.

We tackle the problem of selecting the most appropriate strategy in Section E.3. This problem involves the arrangement of planning and execution agents across sender and receiver terminals.

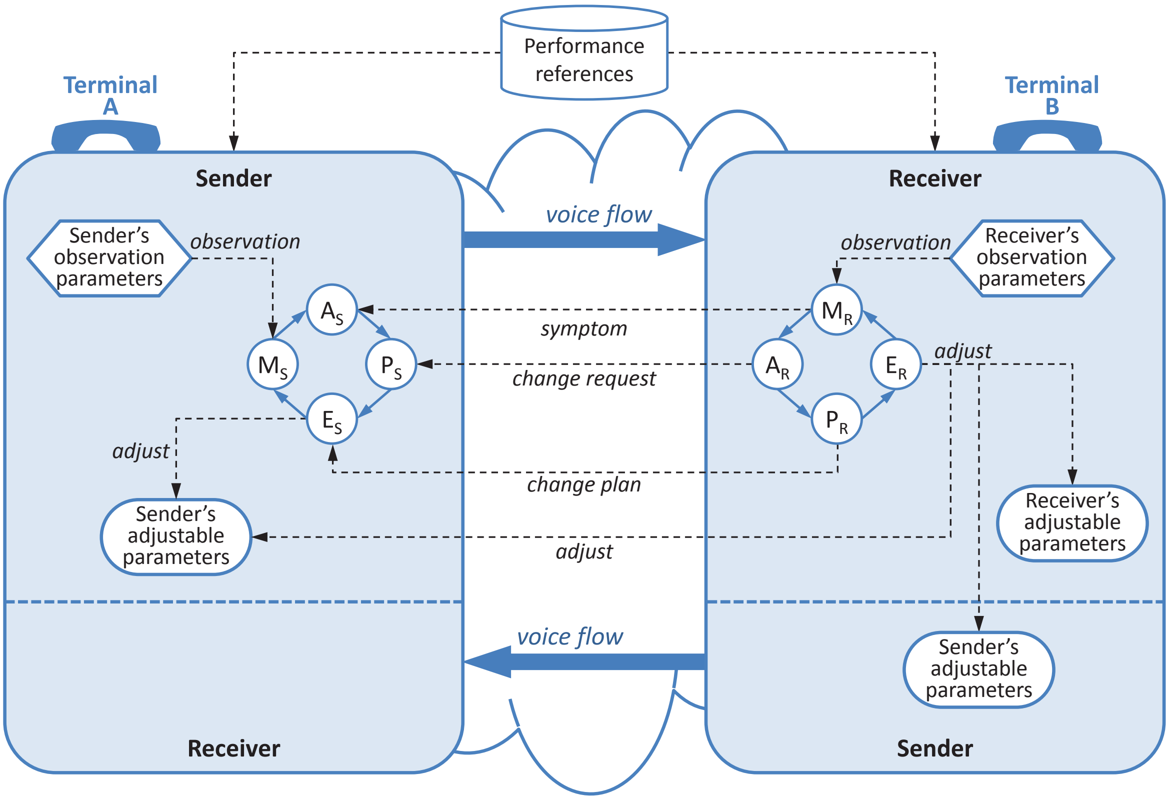


Figure E.5: Explicit identification of the feedback control loop in an adaptive VoIP call between two terminals.

As depicted in Figure E.6, the control loop starts with both sender's and receiver's monitoring agents (M_S and M_R) collecting the observation parameters available at their respective terminals. These symptoms are sent to the analysis agent, which determines the decision metrics MOS_{inst} and MOS_{cum}.

Next, the analysis agents check whether the control mechanism is inside a transient state. This latter is characterized by two situations. The first is the beginning of the call, when the amount of RTP packets is not enough for diagnosing quality problems. This period was set to 10 s. The second situation is determined by the propagation time, that is, the time needed for an adaptation action to be issued and the correspondent effects to be captured by the MAPE agents. This period was set to 5 s. Those transient values where adapted from the work of Myakotnykh & Thompson [143].

If the control loop is outside a transient state, then the analysis agents compare the MOS_{inst} and MOS_{cum} values with the toll-quality threshold of 3.6, as specified by Goal 1. They use the result of such a comparison to fork the flowchart according to a traffic light fashion.

Thus, during each run of the control loop, the flowchart on Figure E.6 may take one of the following branches: *optimization* (green state), *warning* (yellow 1 state), *prudence* (yellow 1 state), and *contingency* (red state). These four branches are described in figures E.7, E.8, E.9, and E.10, respectively.

If both MOS_{cum} and MOS_{inst} values are above 3.6, then the analysis agents call the *optimization* subroutine (green state), described in Figure E.7. If both values are below 3.6, then the analysis agents call the *contingency* subroutine (red state), described in Figure E.10. Now, if MOS_{cum} is above 3.6, but MOS_{inst} is below this threshold, then this means that perhaps speech quality is decreasing. Hence, the analysis agents call the *warning* subroutine (yellow 1 state), described in Figure E.8. Finally, if MOS_{cum} is below 3.6, but MOS_{inst} is above this threshold, then this means that perhaps speech quality is being recovered. Hence, the analysis agents call the *prudence* subroutine (yellow 2 state), described in Figure E.9.

In all diagrams depicted in figures E.6 to E.10, a circled letter in the upper left side of each box indicates which MAPE agent performs the task described inside the box. The location of an agent – either at the receiver or at the sender – can be abstracted in the present discussion. In Section E.3 we present three possible arrangements.

The optimization subroutine (G), depicted in Figure E.7, observes whether the current MOS_{inst} is lower than MOS_{cum}. If so, the speech quality may be decreasing. Thus, the planning agent tells to the execution agent to switch the current codec to another one with lower bitrate, for precaution. On the contrary, network conditions are favorable and hence speech quality can be improved by switching the current codec to a higher-quality one.

Note that the decision diamond in Figure E.7 compares the ΔMOS, the difference between MOS_{cum} and MOS_{inst}, to a constant named $thrs_1$, but not to zero. This avoids unnecessary oscillation, as suggested by [158]. The warning and prudence subroutines employ another constant, named $thrs_2$, which indicate an obvious decrease in perceived speech quality.

The warning subroutine (Y1), depicted in Figure E.8, observes whether the ΔMOS is above $thrs_2$, which indicates a severe decreasing of the instantaneous quality. In this case, a drastic adjustment is issued – to switch to a lower-bitrate codec in two steps – in order to avoid a more severe speech quality degradation. If the ΔMOS is between the two thresholds, then the planning agent tells to the execution agent to lower the codec in one step only. Finally, if the ΔMOS is not meaningful, then the planning agent does nothing, and the control loop gets back to the monitoring activity.

The prudence subroutine (Y2), depicted in Figure E.9, defines ΔMOS oppositely to the two previous subroutines, because MOS_{inst} is necessarily higher than MOS_{cum}. It checks whether the instantaneous quality is getting better (ΔMOS above $thrs_2$), which indicates that the current codec should be switched to one with a lower-bitrate in just one step. On the contrary, the planning agent does nothing, and the control loop gets back to the monitoring activity.

Finally, the contingency subroutine (R), depicted in Figure E.10, deals with the worst-case scenario, when MOS_{cum} and MOS_{inst} diverge from Goal 1. In this case, the only thing to do is to switch the current codec to one with a lower-bitrate in two steps.

To close this subsection, we characterize the proposed MAPE control loop in terms of the

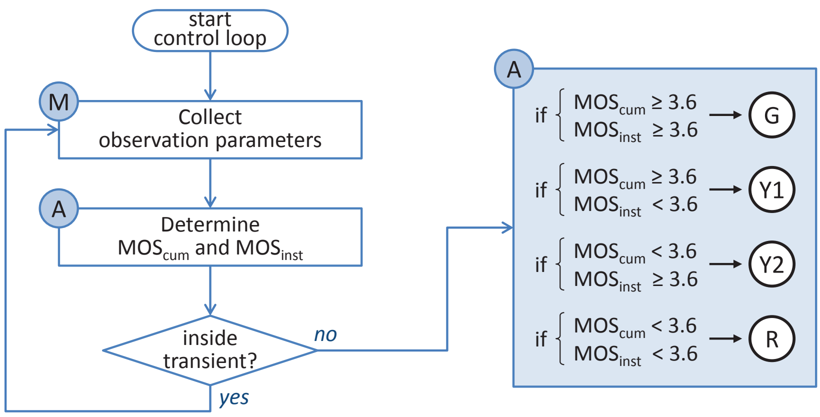


Figure E.6: Flowchart of the monitoring and analysis activities of the control loop in the adaptive VoIP terminal.

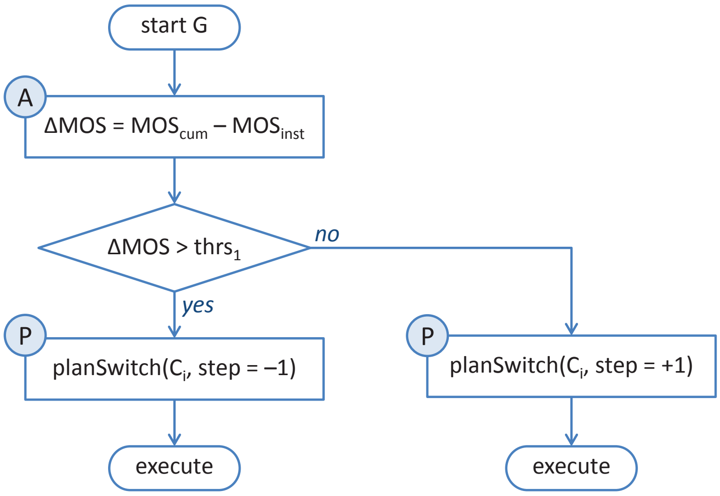


Figure E.7: Flowchart of the optimization subroutine (green).

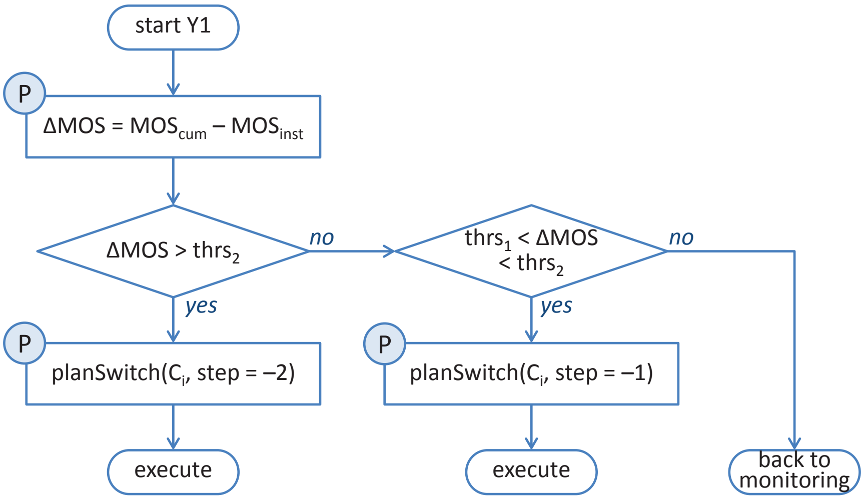


Figure E.8: Flowchart of the warning subroutine (yellow 1).

mechanisms dimension attributes in Andersson's et al. classification [3] on Figure E.3(a).

1. *Type.* It captures whether adaptation is related to the parameters of the system components, or to the composition (structure) of the system, or to a combination of these.

   The adaptive VoIP terminal specified in this section implements a *parametric* adaptation, because it modifies program variables that determine the behavior of the VoIP call toward disturbances in the network.

2. *Autonomy.* It identifies the degree of outside intervention during adaptation.

   The degree of autonomy of the proposed adaptive VoIP terminal is *autonomous*, because at run-time there is no external influence guiding how the agents should adapt.

3. *Organization.* It captures whether the adaptation is done by a single component or distributed amongst several components.

   At the receiver, the execution agent E_R triggers adaptation of parameters in the dejitter buffer. At the same time, codec switching is applied at the sender side, either by the E_R or by E_S, depending on the adopted adaptation strategy. Hence, the organization of the two adaptive VoIP terminals is *decentralized*.

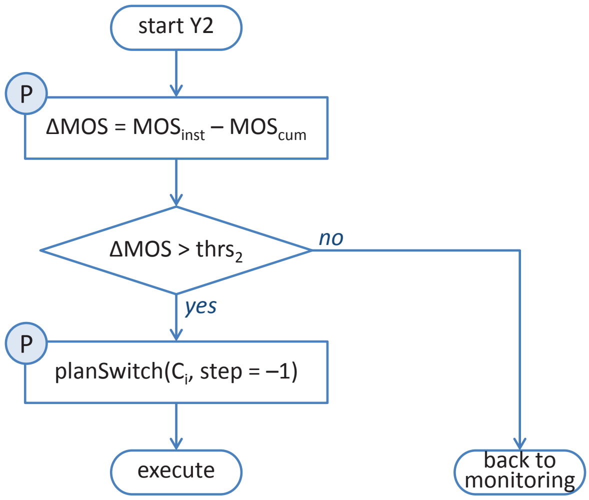


Figure E.9: Flowchart of the prudence subroutine (yellow 2).

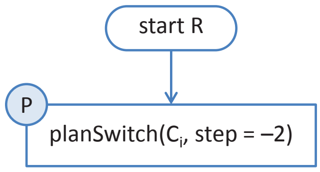


Figure E.10: Flowchart of the contingency subroutine (red).

4. *Scope.* It identifies whether adaptation is localized or involves the entire system.

   Adjustments applied by the E_R at the receiver endpoint itself have a *local* scope, because the sender does not participate on it. Adjustments applied by the E_R over sender's adjustable parameters have a *global* scope, because both endpoints should agree about the codec newly adopted.

5. *Duration.* It refers to how long the adaptation lasts.

   Adjustments applied at the receiver have a short-term impact over user perception (much less than one Round-Trip Time, RTT), but adjustments applied at the sender have a medium-term impact (at least one RTT).

6. *Timeliness.* It captures whether the adaptation period can be guaranteed.

   Adjustments in the dejitter buffer are time-guaranteed, because they are applied by the execution agent E_R at the receiver endpoint itself. On the other hand, codec switching requires some time to complete a handshaking negotiation of a reINVITE SIP request. If any message of this handshake gets lost in the network, then the adaptation will take longer than usually expected.

7. *Triggering.* It identifies whether the change that triggers adaptation is associated with an event or a time slot.

   The specified adaptive VoIP terminal is *event-triggered*, because the planning agent only starts the changing plan when it detects that MOS_{cum} and MOS_{inst} are below the expected thresholds.

E.3 Efficiency of Feedback Message Exchange

From Figure E.5, one can draw several different arrangements of interactions among MAPE agents, sensors and effectors located in both sender and receiver. Each arrangement determines a particular message exchanging dynamics across the IP network for delivering symptoms, change requests and change plans among the MAPE agents to achieve the design goals.

However, three arrangements are of special interest, as depicted in Figure C.4 (p. 51). We call such arrangements as *adaptation strategies* (AS):

1. AS_1 – Adaptation decision and execution taken by the *sender* of the managed RTP flow.
2. AS_2 – Adaptation decision taken by the *receiver*, and execution taken by the *sender* of the managed RTP *flow*.
3. AS_3 – Adaptation decision taken by the *receiver*, and execution taken by the *sender* of the managed RTP *counterflow*.

To the best of our knowledge, these strategies were never compared before. In order to accomplish so, we modified the source code of PJSUA, the SIP client of PJSIP [155], for implementing each one of the adaptation strategies AS_1, AS_2 and AS_3. Thus, we produced three different versions of this softphone.

Next, we designed some experiments to compare the three adaptation strategy AS_1, AS_2 and AS_3. A nonadaptive terminal, named AS_0, was included in this comparison study as a control condition. The purpose is to determine whether the adaptation strategies present, in fact, some performance gain if compared with the nonadaptive terminal. Furthermore, we aim to determine the best strategy that minimizes the amount of feedback, signaling and acknowledge messages. In the next chapter, we detail the experiment setup and statistical methods used for accomplishing this task.

E.4 Self-Adaptation Properties

Besides efficiency of feedback message exchange, the strategies should be evaluated in regard to the properties that characterize them as really *adaptive*. In this section, we describe the desired properties of a self-adaptive system. In the next chapter, we detail the experimental design and the statistical method for demonstrating the compliance of the three strategies to those properties.

Verification and validation of self-adaptive systems are not mature topics yet. It was pointed as one of the challenges on the research roadmap elaborated by the participants of the Dagstuhl Seminar 10431 on Software Engineering for Self-Adaptive Systems [46].

Villegas et al. [198] proposed a framework for evaluating self-adaptive systems. This framework synthesizes the different properties that have been proposed for evaluating a self-adaptation system. Several of these properties are gleaned from control theory and reinterpreted for self-adaptive software [132]. We list those adaptation properties below for the sake of completeness. However, only the first six properties are in the thesis' scope. The three remainders are not tested in this thesis.

1. *Stability.* It is the degree in that the adaptation process converges toward the control objective. A system is said to be stable if its response to a *bounded* input is itself *bounded* by a desirable range, that is, if the decision metrics are within an allowable range to the performance references [132]. In terms of speech quality, the adaptation strategies should

not cause an oscillating MOS if network conditions are stable, yet not favorable (i.e., high loss or delay).

2. *Accuracy.* It ensures that adaptation goals are met, within given tolerances, in terms of how close the managed system approximates to the desired state. Thus, the focus is on the degree of accomplishment of design goals. For example, an adaptive strategy may be stable (i.e., the MOS values should not oscillate along the call), but the average MOS may be under the expected thresholds, hence it is not accurate.

3. *Short Settling Time.* Commonly referred to as recovery time, reaction time, or healing time, it is the time required for the system to reach its steady-state value (e.g., after a unit step input is applied). For classic control systems, the unit step is a signal that is zero for all time $k < 0$, and one for all times $k \geqslant 0$ [70]. The problem in verifying this property in VoIP systems is the definition of the "one level" in the unit step signal. For example, if "one" means a 100% packet loss, then obviously no adaptation strategy will present a short settling time. On the other hand, if "one" means only a 1% packet loss, then even nonadaptive terminals may present a short settling time. In the next chapter, we propose some experiments to determine this "one level" in terms of packet loss and network delay, and in Chapter G, we comment the results.

4. *Small Overshoot.* It expresses how well the adaptation performs under given conditions – the amount of resources used in excess to achieve a required short settling-time before reaching a stable state. For an adaptive VoIP call, this property is related to the number of feedback, signaling and acknowledgment messages that each adaptation strategy AS_1, AS_2 and AS_3 injects in the network.

5. *Robustness.* The adaptation process is robust if the controller is able to operate within desired limits even under unforeseen conditions. For an adaptive VoIP call, this property was checked by comparing the performance of the three adaptation strategies in the presence of non-VoIP background traffic.

6. *Scalability.* It is the capability of a controller to support increasing demands of work with sustained performance using additional computing resources. For an adaptive VoIP call, this property is checked by comparing the performance of the three adaptation strategies when the number of simultaneous VoIP calls in the same underlying network is gradually increased.

7. *Termination.* It guarantees that the list of adaptive operations is finite and its execution will finish, even if the system does not reach the desired state. This property will not be checked, because it requires an expertise on Formal Methods, which is out of the scope of the thesis, as depicted in Figure A.4 (p. 29).

8. *Consistency.* It ensures the structural and behavioral integrity of the managed system after performing an adaptation process. Because this property is related to compositional adaptation, rather than parameter adaptation, it will not be checked.

9. *Security.* It states that both managed system and adaptation mechanism are required to be protected from disclosure (confidentiality), modification (integrity), and destruction (availability). This property will not be checked, because it is also out of the scope depicted in Figure A.4 (p. 29).

E.5 Summary

In this chapter, we described how three open problems about adaptive management of speech quality of a VoIP call between two terminals are being tackled in this thesis project. These problems are interrelated to each other: (1) parametrization of codec switching precedence for supporting the switching decision, (2) explicit design of MAPE agents of the control loop, and (3) efficiency analysis of feedback message exchanging. We developed a prototype using the PJSUA client of the PJSIP project [153]. Three adaptation strategies were implemented: (1) adaptation decision and execution taken by the *sender* of the managed RTP flow; (2) adaptation decision taken by the *receiver*, and execution taken by the *sender* of the managed RTP *flow*; and (3) adaptation decision taken by the *receiver*, and execution taken by the *sender* of the managed RTP *counterflow*.

All three adaptation strategies were evaluated in order to check their compliance degree to six out of nine self-adaptation properties described in this chapter: stability, accuracy, short settling time, small overshoot, robustness, and scalability. In the next chapter, we present the experimental setup and the statistical hypotheses used to check each one of these adaptation properties.

Appendix F

Experiment Design and Methodology

Randomization is something that everybody says they do,
but hardly anybody does properly.
Michael J. Crawley [44]

Contents

During the last fifteen years, several algorithms, mechanisms and systems have been developed and proposed in the literature about VoIP. A few of them, however, have been properly tested, or has their experimental setup been properly described [143, 147, 200].

Reproducibility is one of the main principles of the scientific method, and it refers to the ability of an experiment to be accurately reproduced by someone else working independently. Reproducibility requires the use of a *controlled* experiment environment in a *scalable* way. Consequently, experimental conditions will not be the same as in a real implementation, but they should be modeled as close as possible to reality.

Nevertheless, building realistic models is not all about experimental design. Experiments must be planned so that their outcomes can be meaningful to confirm or reject hypotheses made about the artifact under evaluation. Also, experiments should be properly designed in order to isolate measurement errors and to estimate the contribution of controlled parameters over the outcome.

We begin this chapter by presenting some basic terminology about experimental design in Section F.1. Next, in Section F.2, we review the literature about experimental setup of VoIP applications, introducing some models used for representing voice traffic, background traffic and network topology. Finally, in Section F.7, we present the statistical planning of the experiments conducted to validate our proposal of an adaptive algorithm for speech quality management.

F.1 Basic Terminology of Experimental Design

The terms used in the design of experiments can sometimes be confusing. Here, we define some of the most important terms that we use throughout this chapter [49, 98, 115]:

1. *Response variable.* It is the output value that is measured as the input values are changed. Examples of response variable are the MOS value and the total execution time.
2. *Factors.* They are the input variables of an experiment that can be controlled or changed by the experimenter. For example, the factors of an experiment might include the number of simultaneous calls, background traffic level, network delay, packet loss, and so forth.
3. *Levels.* The levels of a factor are the specific values to which it may be set. These values may be continuous, such as the duration of a call, or they may be categorical, such as the type of adaptive strategy being used.
4. *Treatment.* It is a particular combination of levels of various factors.
5. *Replication.* Replicating an experiment means rerunning it completely, with all the same input levels. Since the measurements of the response variable are subject to random variations, replications of an experiment are used to determine the effect of measurement error on the response variable.
6. *Interaction.* An interaction among factors occurs when the effect of one factor depends on the level of another factor.

The primary goal of the design of experiments is to determine the maximum amount of information about a system with the minimum amount of effort. A key assumption behind the design of experiments is that there is a *nonzero cost* associated with performing an experiment. This cost includes the time and effort required to gather the necessary data, plus the time and effort needed to analyze these data to draw some appropriate conclusions. Consequently, it is important to minimize the number of experiments that must be performed while maximizing the information obtained [98, 115].

To validate our proposal of an adaptive algorithm for speech quality management, we used the *measurement* technique [98]. We chose this technique in order to reuse some tools and expertise developed in previous works. As said in Chapter A, we implemented a QoE measurement tool, based on the E-model, which works on log files generated by softphones [28]. We later improved it to work at run-time [196], and now it is part of the current development version of OPAL VoIP library [153] and has been submitted to the PJSIP platform [155]. In both OPAL and PJSIP, VoIP metrics are conveyed between endpoints through RTCP XR reports [56].

Another reason for choosing the measurement technique is the lack of measurement studies of adaptive VoIP solutions, as seen in Chapter C. Some implementation aspects of VoIP terminals engaged in a VoIP call can be neglected by simulation scenarios, such as how dejitter buffers deal with two RTP flows encoded by different codecs related to the same call session.

F.2 Experimental Setup for Testing VoIP Applications

Many questions arise when designing a testbed for testing VoIP applications. How long should take a call? Should all calls have the same duration? How to model the silence-activity pattern of a typical conversation between two persons? Should non-VoIP background traffic be considered? How can it be modeled? Which is the most appropriate network topology to test an artifact? How to stress a VoIP application against impairments such as delay, jitter and packet loss in a proper and realistic manner? How many times should the experiment be repeated in order to get reliable outcomes? How to randomize the experiment in order to reduce the bias owing to marginal aspects, such as speakers' language, speakers' genre, speech content, time of the day, equipment configuration, and so on?

There are no straight answers to all above questions. However, in this section, we present the most sound models that have been used in the literature to build a testbed as faithful as possible to real implementations. To this purpose, we review the literature about VoIP traffic characterization, and the same literature of Chapter C about VoIP adaptation, but this time focusing on the experimental setup that they have used.

Our objective here is twofold: (1) to present a big picture of best practices for testing VoIP application, otherwise scattered throughout the literature; and (2) to justify the choices that we took to build the testbed used in this thesis.

For didactic reasons, we group the main concerns about the experimental setup for testing VoIP applications into five categories, which are detailed in the next sections:

1. *VoIP traffic characterization.* It deals with models for representing the behavior of users when demanding network resources (e.g., call duration and call arrival), models for characterizing the interaction between two talkers during a conversation, and models for characterizing ON/OFF packet activity during a conversation. This is developed in Section F.3.

2. *Non-VoIP background traffic.* Most VoIP applications share the same medium with other data flows, such as HTTP and P2P applications. Thus, we need models that represent it properly. This is developed in Section F.4.

3. *Topology.* This category concerns to how network and VoIP components can be arranged in an experimental testbed so as to be as realistic as possible to an actual VoIP system implementation. This is developed in Section F.5.

4. *Replication and Randomization.* Replication increases *reliability*, whereas randomization reduces *bias* [44]. Hence, treatment material and combinations should be properly planned to comply with these principles. This is developed in Section F.6.

F.3 VoIP Traffic Characterization

As pointed by Stern et al. [186] and Mattos et al. [126], an accurate model of the ON/OFF characteristics of conversational speech is important for the design and analysis of communication systems. In other words, traffic behavior should be represented as faithful as possible to reality, in order to allow a preview of the impact of changing network characteristics in the service quality.

We can group the models of VoIP traffic characterization proposed in the literature into three basic perspectives, depending on which system's level we observe.

1. *User-demand-level perspective.* It deals with user demand upon an application or system. User behavior must be considered, for example, when testing persistence or capacity of a given IP telephony system. Two variables describe traffic behavior at this perspective: *call holding time* and the *interarrival time between calls.*

2. *Speech-level perspective.* It deals with the dynamics of a conversation between two parties, that is, how they interact with each other during a dialogue, in which periods of talkspurts intercalate with periods of silence. At this perspective, we have two main concerns: transition model between states of talkspurts and silence, and residence time in each state.

3. *Packet-level perspective.* It deals with how packets are generated to form a voice flow. At this perspective, traffic behavior is described in terms of ON and OFF periods.

Figure F.1 provides a more precise picture of how those perspectives are interrelated.

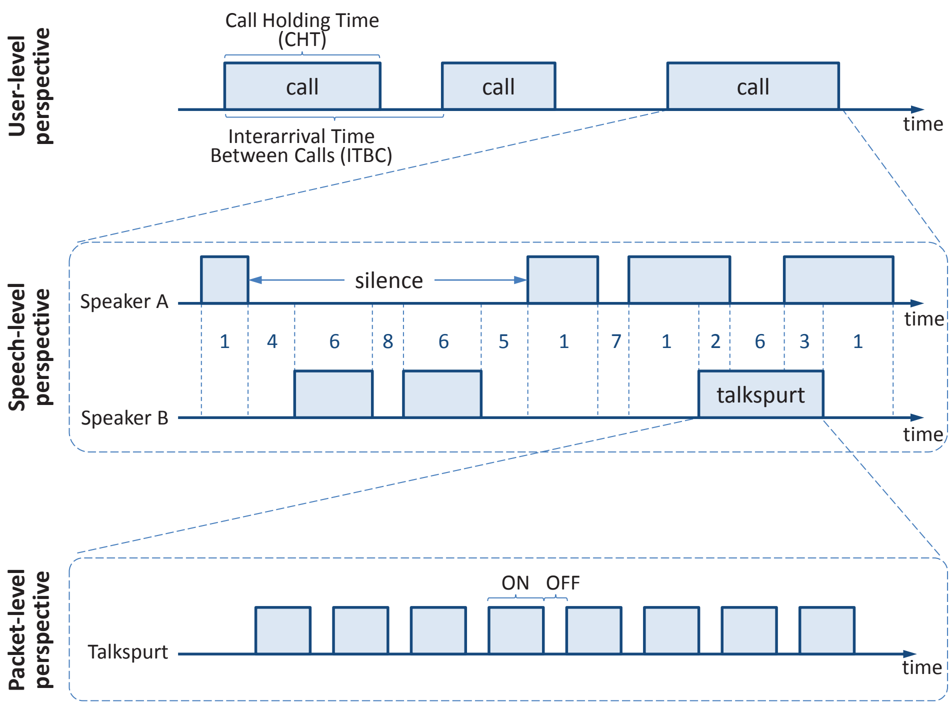


Figure F.1: Relationship between the three perspectives of voice traffic characterization. *User-level* behavior refers to the channel occupation by the bulk group of users. *Speech-level* behavior refers to the interaction between two parties in a specific conversation. *Packet-level* behavior refers to the packet activity after speech encoding, for a specific voice flow.

F.3.1 User-Demand-Level Perspective

At the user-demand-level perspective, we are interested in modeling *channel occupation*, as a result of dynamic creation and tearing down of VoIP calls by several independent users. User demand behavior determines the traffic load offered to a given telephone system (VoIP-based or not). In traditional PSTN, user demand is limited by the number of available physical circuits. In IP telephony, on the other hand, there is no sharp limitation. Nevertheless, the higher the demand is, the higher the network load becomes and, consequently, the call drop probability. In such cases, it is common to use call admission control (CAC) techniques for guaranteeing a minimum speech quality for the ongoing calls [19, 32, 34, 107, 129, 157, 164, 175, 191].

User demand behavior is characterized by two variables:

1. *Call holding time (CHT).* Also known as *call duration* or *conversation time*, it is the time elapsed from the first to the last packet of a voice flow [21, 204]. Note that CHT definition does not account for the *setup time*, which is the period between the opening of the channel and the moment when the call is answered or cleared without success. The total holding time (THT) is the sum of CHT and the setup time. Bodouhi & Hadjinicolaou [18] show that the characteristics of THT and CHT are significantly different from each other, and the misuse of these two notions leads to inaccurate results. In this thesis, we deal with only CHT.

    - *CHT distribution.* In the traditional PSTN, CHT is modeled by the *exponential* distribution, which is directly related to user behavior. PSTN is charged by time, which usually induces the user to make short telephone calls; hence, long calls are rare [21, 68, 126]. On the other hand, many VoIP services are charged by a fixed fare, depending on the provided bandwidth. This change in the billing method is modifying user behavior, and long-time telephonic calls are not a rare event anymore. Consequently, the exponential model fails to model VoIP traffic accurately. Many reports in the literature show that the CHT of VoIP calls has rather a *heavy-tailed* distribution [17, 33, 37, 38, 45, 68, 126, 178]. As shown in Table F.1, there is not a consensus yet, but the majority of works about VoIP traffic characterization have found that Pareto distribution fits the best with live measurements.

    - *CHT mean value.* In PSTN, business calls usually last 3 min [68, 100]. However, end-to-end (E2E) VoIP calls tend to last longer, because they are free. Guha et al [68] measured that the mean CHT of calls placed between two Skype clients is 12 min 53 s. Bonfiglio et al. [21] showed that calls placed between two Skype clients (E2E), last much more than calls placed between a Skype client and a PSTN terminal (E2O – end-to-out). They did not present the mean value of CHT, but from their graphics, one can tell that the median for the E2E calls is 168 s, and the median for the E2O calls is only about 24 s, approximately. The other authors in Table F.1 performed measurements over hybrid architectures (E2O) or paid VoIP services, which explains their lower CHT mean values.

2. *Interarrival time between calls (ITBC).* It represents the time interval between *successive* calls [126].

    - *ITBC distribution.* Most studies [17, 37, 45, 68, 126, 204] claim that the ITBC can be also modeled by an *exponential* distribution for VoIP applications, except for Sheluhin et al. [178], who claim that it rather follows a Pareto distribution.

    - *ITBC mean value.* Authors do not agree on a typical ITBC mean value. On one hand, there is a huge difference in the observation time used for collecting measurements, which ranges from 4 hours to 240 days. On the other hand, the works listed on Table F.1 observed VoIP services of different sizes, but did not informed the number of subscribers, which prevents a comparison between the reported ITBC.

Table F.1 compares the findings of some works about voice traffic characterization. It presents, for each work, the type of VoIP service, the observation period used for collecting data, and CHT mean value and distribution. The figures were taken from passive measurements over four kinds of live VoIP services: (1) PSTN carrier that uses VoIP only in its internal infrastructure, (2) end-to-out public VoIP carrier, (3) university campus VoIP service, or (4) supernodes of the Skype P2P relay.

Regarding to the observation period spent for characterizing voice traffic (third column of Table F.1), we have some comments and reservations to do. Heegaard [69] and Iversen [97] showed that changes in user's context and demand determine voice traffic variation over the day, weeks

Table F.1: Call holding time (CHT) and interarrival time between calls (ITBC) characteristics measured from live VoIP services by some works in the literature.

Reference	**VoIP service**	**Observation period**	**CHT**	
			Mean	**Distribution**
Birke et al. [17]	PSTN carrier	3 weeks	1 min 46 s	inverse Gaussian
Bodouhi & Hadjinicolaou [18]	E2O VoIP carrier	60 days	13 min 7 s	beta
Chen et al. [33]	PSTN carrier	63 days	2 min 28 s	log-normal
Choi et al. [37]	PSTN carrier	1 day	1 min 46 s	log-normal
Dang et al. [45]	Campus	240 days	1 min 54 s	Pareto
Guha et al. [68]	Skype (supernodes)	135 days	12 min 53 s	heavy-tailed
Heegaard et al. [69]	E2O VoIP carrier	30 days	—	not exponential
Mattos et al. [126]	PSTN carrier	4 hours	2 min 23 s	Pareto
	PSTN carrier	4 hours	—	Pareto
Sheluhin et al. [178]	Campus	140 days	2 min 3 s	Pareto
Xi et al. [204]	PSTN carrier	2 days	1 min 54 s	Weibull

and season. As a consequence, the most reliable studies on voice traffic characterization are those with longer observation periods.

Finally, we highlight Heegaard's conclusions to support our choice of setting up the CHT of our experiments:

- CHT will be shorter if the price depends on the duration of the call (time-based tolling), than if the price does not (flat-rate or paid by someone else). For example, Bichler & Clarke [14] observed that fraudulent payphone users placed illegally free calls that lasted 15 min to 20 min on average, whereas legitimate calls lasted 4 min to 5 min. Furthermore, Guha [68] et al. observed that E2E Skype calls, which are free, last more than E2O Skype calls, which are charged.
- CHT will be longer if communicating parties have spare time (at evenings and nights), than if they are in a hurry (at business hours). Iversen observed also such an increase in CHT for private calls during evenings, in measurements taken in 1973 [96] and confirmed in 2001 [97].

In this thesis, we focus on individual voice calls only, during which sender and receiver end-points try to overcome QoE problems by adapting their parameters. Thus, call arrival behavior (ITBC) is not a major concern in our experiments. Instead, our interest is rather focused on the performance and stability of adaptive VoIP mechanisms along the entire call (CHT).

Therefore, we fixed the duration of the calls in 02 minutes, as the same as most of works summarized on Table F.1. This CHT is specially convenient to grasp the whole behavior of adaptive strategies on overcoming disturbances. The analysis of longer calls is compromised because of the huge amount of measurement points collected during experiments.

F.3.2 Speech-Level Perspective

At the speech-level perspective, our interest resides on the *voice activity* of the two participants engaged in a specific VoIP call. Speech can be modeled as short bursts of energy (called *talkspurts*) separated by *silence* gaps. Talkspurt and silence states are commonly referred to as ON and OFF states, respectively [186]. In this thesis, however, we use the terms *ON* and *OFF* to designate whether the codec generates or not voice packets in a *one-way* stream. In contrast, we use the terms *talkspurt* and *silence* to designate the voice activity of the speakers engaged in a *two-way* conversation.

Models of traffic generation at the speech level are necessary when the experimental testbed requires *speech* signals as input to emulate actual conversations between two persons. On the other hand, if we need to generate *packet* traces, as an output of codec processing, we should employ ON/OFF models, as described in Section F.3.3.

During a typical conversation, the two speakers alternate periods of single talk, double talk and mutual silence. Such a combination of talkspurt and silence states of both speakers is described by a *state transition model*, which comprises three key elements: silence gap definition, state transition model, and residence time, as detailed in the next subsections.

Silence Gap Definition

Stern et al. [186] distinguish three types of silence gaps: (1) *listening pauses*, which occur when a party is silent and listening to the other party; (2) *long speaking pauses*, which occur between phrases or sentences while a party is speaking; and (3) *short speaking pauses*, which occur between words or syllables while a party is speaking.

The problem here is to determine the scale resolution used for classifying a speech sample as being a short speaking pause or being part of a talkspurt. This can be regarded as an instance of the coastline paradox [122], which states that the coastline of a landmass does not have a well-defined length.

In practice, the silence gap depends on the sensibility of the codec used for speech processing. For example, if the voice activity detector (VAD) of a given codec cannot detect pauses below 1 ms, there is no point in building a model that considers pauses shorter than this value as being a silence state.

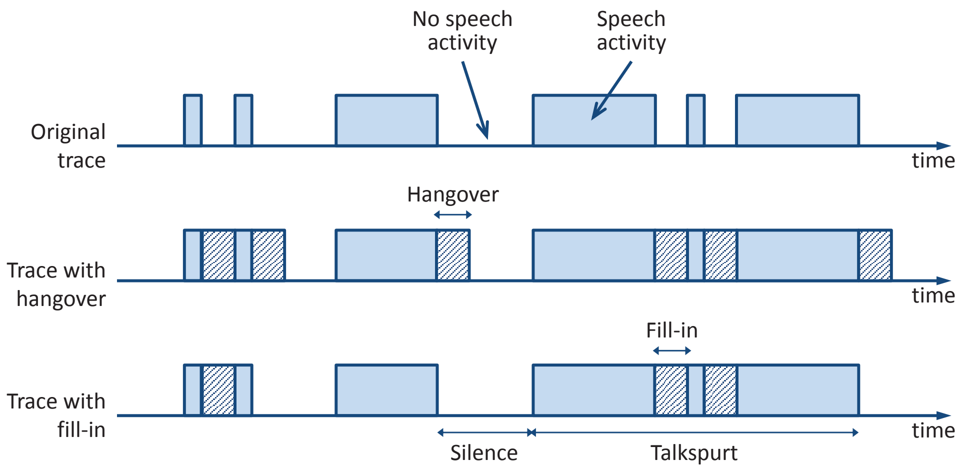


Figure F.2: Illustration of the hangover and fill-in techniques. With hangover, the ending of all talkspurts is extended. With fill-in, all silence periods that have duration of less than the fill-in value are filled in.

Naive implementations of VADs suffer from *end-clipping*, that is, the codec might clip the end of a talkspurt when the speaker's voice goes down. Hence, VADs may implement hangover or fill-in techniques to avoid clipping [67, 102, 134]. When using *hangover*, the codec switches from the ON-state to the OFF-state with a certain delay after the detected end of the talkspurt. The *fill-in* technique bridges a short gap between two intervals of voice activity if the gap is no longer than the fill-in duration, but does not reduce longer silence durations.

Figure F.2, adapted from [134], compares the effect of hangover and fill-in techniques on some speech input. Note that both techniques prolong the duration of the ON-state and are considered by both two-way conversation models (speech-level perspective) and one-way monologue models (packet-level perspective).

State Transition Model

A state transition model describes the dynamics between two talking parties (A and B). It defines the states that represent all devisable combinations of voice activity and inactivity of both speakers, and the allowed transitions among these states. Here, we highlight three state transition models taken from the literature:

1. *Brady 4-state model.* This is the simplest model, proposed by Brady [25] and standardized in the ITU-T Recommendation P.59 [84]. As depicted in F.3, speaker A can be either talking or silent, and views B as either talking or silent. The only restriction is that the two speakers cannot change their states at the same time. Thus, in Figure F.3, diagonal crossings are prohibited.

2. *Brady 6-state model.* Brady [26] expanded the previous 4-state model, which proved to be inadequate in predicting events surrounding double talk. Thus, the double-talk and mutual-silence states in Figure F.3 were dismembered into two states, to consider which party spoke last, as shown in Figure F.4.

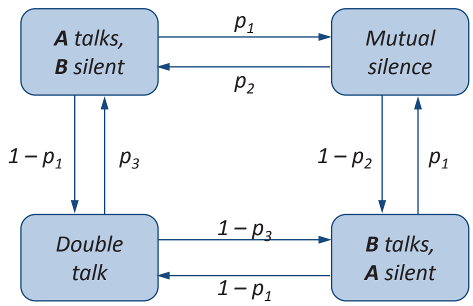


Figure F.3: The Brady 4-state model of two-way conversational speech.

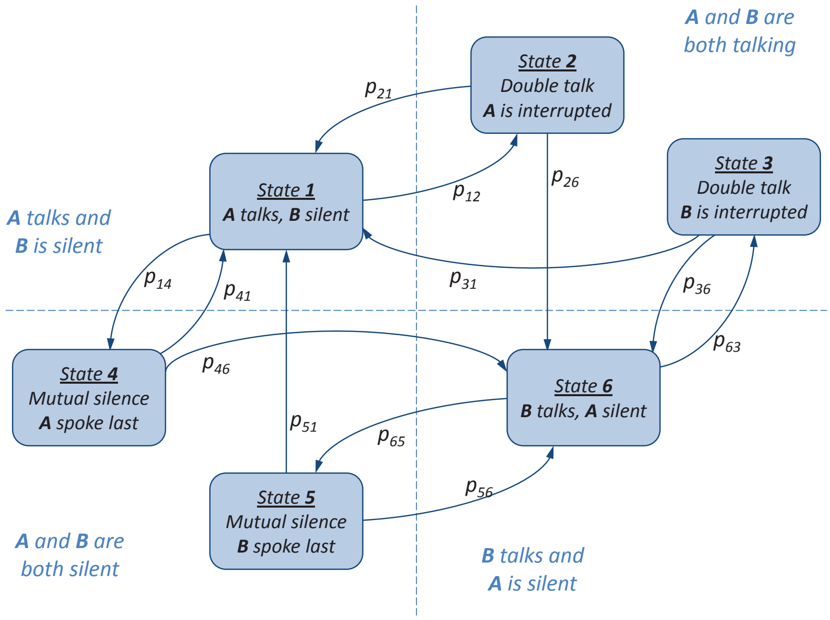


Figure F.4: The Brady 6-state model of two-way conversational speech.

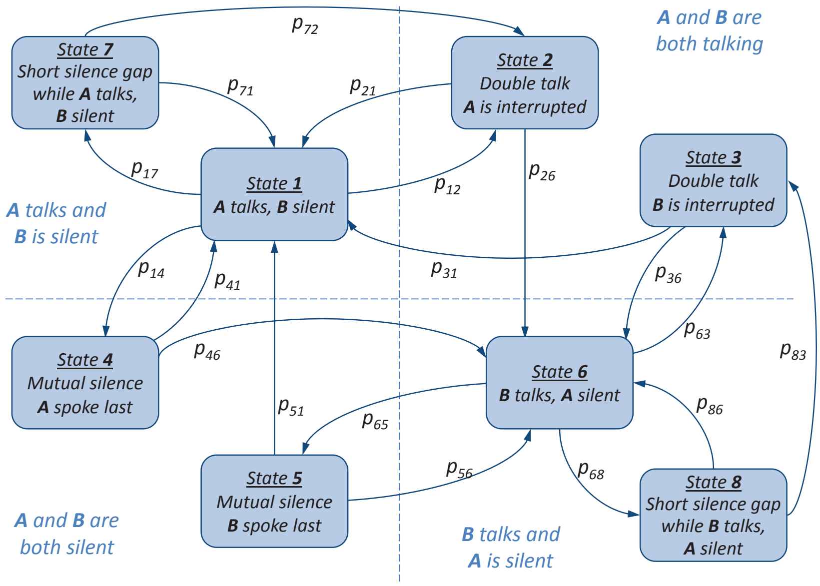


Figure F.5: The Stern-Mahmoud-Wong 8-state model of two-way conversational speech.

3. *Stern-Mahmoud-Wong 8-state model.* In the Brady's 6-state model, all spurts that last less than 15 ms are assumed to be noise and thus considered as being silence. Additionally, all gaps that last less than 200 ms are filled in, as they are assumed to be stop consonants or other minor breaks in continuous speech [26].

   Nevertheless, Stern et al. [186] argue that modern speech processing techniques can track silence periods shorter than 200 ms. Thus, they introduced two new states for representing short breaks during both A and B talkspurts, as depicted in Figure F.5. These short breaks may not be perceptible by the human auditory system, but they are accurately detected by new-generation codecs. Figure F.1 (p. 103) sketches a time diagram of a conversation between A and B, identifying the eight states of the Stern-Mahmoud-Wong model.

   We employed the Stern-Mahmoud-Wong 8-state model for generating the speech files used as input of the experiments carried out in this thesis. In these experiments, we employed softphones to create and tear down VoIP calls, injecting not only RTP traffic, but also signaling traffic (e.g., SIP messages). Hence, we can guarantee that the experimental testbed is very similar to real VoIP implementations.

   In Figure F.5, the terms p_{ij} indicate the state transition probabilities, where the first

Table F.2: State transition parameters for the Stern-Mahmoud-Wong 8-state transition model. Note that they are not transition probabilities.

α_{14}	α_{17}	α_{31}	α_{21}	α_{71}	α_{41}	α_{51}	α_{12} α_{72}
α_{65}	α_{68}	α_{26}	α_{36}	α_{86}	α_{56}	α_{46}	α_{63} α_{83}
0.83305	5.4890	2.1572	2.3245	27.62	2.2222[a]	1.0438[a]	0.27853

[a] During the first 200 ms after transition to states 4 or 5, α_{41}, α_{56}, α_{51}, and α_{46} are set to 0.

Table F.3: State transition probabilities for the Stern-Mahmoud-Wong model.

p_{14}	p_{17}	p_{31}	p_{21}	p_{71}	p_{41}	p_{51}	p_{12}	p_{72}
p_{65}	p_{68}	p_{26}	p_{36}	p_{86}	p_{56}	p_{46}	p_{63}	p_{83}
0.126209	0.831594	0.481335	0.518665	0.990016	0.680404	0.319596	0.042198	0.00998

subscript indicates the original state, and the second subscript indicates the new state. Stern et al. [186] did not explicitly provide the values of each p_{ij} individually. Instead, they give the values of α_{ij} parameters, presented here in Table F.2.

The authors have derived the α_{ij} parameters by averaging the 16 conversations reported by Brady [26], but these values are not express transition probabilities. To calculate the state transition probability p_{ij} correspondent to each α_{ij} parameter of the Stern-Mahmoud-Wong model in Figure F.5, we used the following expression:

$$p_{ij} = \frac{\alpha_{ij}}{\sum_{k=1}^{8} \alpha_{ik}}, \tag{F.1}$$

where the α_{ik} values not listed in Table F.2 are assumed to be zero. The p_{ij} values resultant from Equation F.1 are given in Table F.3.

Note that the state transitions for party B have the same characteristics as the state transitions for party A. Thus, the corresponding state transition probabilities for parties A and B have the same value (e.g., $p_{14} = p_{65}$).

Residence Time

The average time spent in a particular state of the transition model is called *residence time*. The three state models presented before assume that the duration of every state is continuous, hence $p_{ij} = 0, \forall i = j$. All of them also assume that the residence time follows an *exponential* process, with different average values.

On the contrary, Ji et al. [101] present a discrete-time model for two-way conversations, but they have not validated their model against measurements of actual conversations, as performed by Brady [25, 26] and Stern et al. [186]. Gruber [67] models talkspurt durations as being approximately a geometric process and the silent periods as being two suitably weighted geometric process, the discrete counterpart of the exponential distribution. However, he did not present a state transition model with temporal parameters. Therefore, we modeled the residence time as being continuous only, as assumed by Brady [25, 26] and Stern et al. [186].

Many authors, such as Gomes et al. [62], Jiang & Schulzrinne [102], and Menth et al. [134], criticize the exponential modeling of talkspurt and silence periods proposed by Brady [26]. In fact, however, they have confused the speech-level and packet-level perspectives of voice traffic, as discussed earlier.

As said before, at the speech-level perspective, we are interested in modeling the alternation between talkspurt and silence states, which depends only on how two humans talk to each other. Thus, no codec advance can change this. At this level, Brady's and Stern's measurements have proven the suitability of the exponential distribution for modeling two-way conversations.

On the other hand, at the packet-level perspective, we are interested in the output of the encoder. Hence, the human behavior is abstracted. Due to VAD and other speech processing techniques, codecs can indeed alter the exponential behavior when converting human speech into packet stream. Therefore, the results of Gomes et al. [62], Jiang & Schulzrinne [102], and Menth et al. [134] are still valid, but only at the packet-level perspective.

Back to the Stern-Mahmoud-Wong model in Figure F.5, the residence time T_{res} for each state $i \in [1, 8]$ is given by the following expression:

$$T_{\mathrm{res},i} = \frac{1}{\sum_{k=1}^{8} \alpha_{ik}}, \tag{F.2}$$

where the α_{ik} parameters not listed in Table F.2 are assumed to be zero. The $T_{\mathrm{res},i}$ values resultant from Equation F.2 are given in Table F.4.

Table F.4: Average residence times (in ms) for the Stern-Mahmoud-Wong 8-state transition model for voice activity in conversational speech.

State	**Average residence time $T_{\mathbf{res}}$ (ms)**
1	151.501838
2	223.129616
3	223.129616
4	306.184936
5	306.184936
6	151.501838
7	35.844182
8	35.844182

F.3.3 Packet-Level Perspective

At the packet-level perspective, our interest resides on the *packet activity* generated by a *single source*, after user's speech is encoded by some codec. Note that there is no direct mapping between talkspurt and ON phases, or between silence and OFF phases. This mapping depends on the type of encoder (CBR or VBR), on whether VAD techniques are implemented in the encoder, and on the hangover time (see Figure F.2).

Simulation tools are more prone to use traffic models at the packet-level perspective. Generally, they do not handle speech signals directly, but rather generate artificial packet streams that emulate the output of voice codecs.

This thesis did not employ simulation tools in the validation experiments. Hence, we did not use models for voice traffic generation at the packet-level perspective. Anyway, for the sake of completion, we present here a brief overview of the state-of-the-art in this matter.

Menth et al. [134] identify three classes of source models for speech traffic at the packet-level perspective, which we explain as follows:

1. *Voice codecs with constant bitrate.* Because the output of CBR codecs is independent of the speech input, CBR traffic is simply generated by sending packets of fixed size at regular time intervals. Actual values of packet size and transmission rate must match some of those CBR codecs listed in Table B.1 (p. 35).

2. *CBR codecs with silence detection.* CBR codecs with VAD/DTX mechanisms can detect voice activity and transmit packets of fixed size only while the user is talking. Thus, the output on the network layer consists of *ON* and *OFF* phases. Average duration and distribution of such ON and OFF periods are a matter of much discussion in the literature. Some works are rather vague and claim only that the distributions of ON and OFF periods are not exponential [47] or they are heavy-tailed [102], without pointing out a specific probability density function. On the other hand, most of the works disagree about which distribution these ON and OFF periods follow: Pareto [45, 126], log-normal [30], hyperexponential [15], gamma for ON periods and Weibull for OFF periods [11], and negative binomial or geometric [134].

3. *Voice codecs with variable bitrate.* VBR codecs send data at regular intervals, but use

variable packet sizes [134]. To the best of our knowledge, only Menth et al. [134] have proposed a source model for generating VBR traffic (AMR and iSAC specifically). They have used a memory Markov chain to model time series of successive packet sizes.

F.4 Non-VoIP Background Traffic

Myakotnykh & Thompson [143] showed that the behavior of voice traffic significantly depends on the presence of – and the load of – data traffic in the network. In other words, the speech quality is affected by not only high *link utilization*, but also by the *proportion of voice and data traffic* in the network.

Unfortunately, most of the approaches reviewed in Chapter C did not consider the influence of the background traffic on the performance of their adaptive VoIP solutions [1, 13, 20, 32, 58, 63, 80, 105, 108, 112, 113, 118, 125, 129, 136, 139, 151, 158, 161, 168, 175, 191, 205, 207].

Two other works claim that they investigated the influence of background traffic on their adaptive VoIP mechanisms, but did not present any detail about which protocols, bitrates or packet sizes were used [16, 206].

On the other hand, a few number of research works generated only fixed CBR background traffic for testing their adaptive VoIP mechanisms [24, 41, 76, 193, 195]. Nevertheless, the bitrate values chosen for the CBR background traffic are rather arbitrary and do not reflect any realistic scenario.

Another approach to generate background traffic is by starting file transfers, via FTP, during the call [10, 99, 123, 150, 173]. Once again, this is a rather arbitrary experimenters' choice and does not necessarily reflect an average statistical behavior of any known data network used in real life.

Finally, we highlight the simulation studies carried out by Myakotnykh & Thompson [143] and Ngamwongwattana [148]. They used ten and nine sources, respectively, to generate Pareto-distributed ON/OFF background traffic with parameter $\alpha = 1.5$. This setting produces Long-Range Dependent traffic with behavior similar to the Internet. These studies benefit from earlier ones about Internet traffic characterization [39, 190], which found that "typical" packet sizes are concentrated around three values: 60% of the packets are 40 bytes, 25% are 550 bytes, and 15% are 1500 bytes. We used this mix in our experiments.

Barberis et al. [10] modeled interfering traffic as ten Pareto ON/OFF sources over UDP connections in one of their experiment scenarios. However, they used different parameters (Pareto's $\alpha = 0.5$, and packet size of strictly only 1 kB), which were not justified to be taken from any measurement study.

F.5 Topology

In a communication network, the term *topology* refers to the way in which the end points attached to the network are interconnected [183]. Modern data networks follow a many-level hierarchical structure, for economical reasons. At one level, we have end users sharing the resources of a local access network. At the top level, we have inter-domain routers carrying out bulk data traffic. As a consequence, end users dispute shared local resources, especially during busy hours. This bottleneck behavior must not be ignored when designing the topology of the experiments.

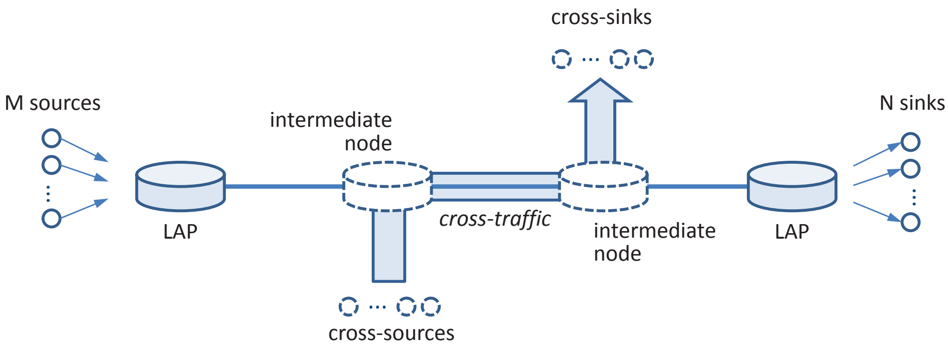


Figure F.6: Generalization of the bottleneck topologies usually adopted by most adaptive VoIP studies in their experiment setup.

Most of the works about adaptive VoIP mechanisms reviewed in Chapter C usually adopt some variation of the generic bottleneck topology depicted in Figure F.6. More complex topologies, while more realistic, have the drawback of providing results that do not lend themselves to reliable interpretation [10].

Basically, the bottleneck topology consists of $M \geqslant 1$ sources disputing a local access point (LAP), and $N \geqslant 1$ sinks attached to another LAP. The LAP can be a wireless AP, a wired router, or a switch, for example. The two LAPs in the topology may be directly interconnected by a link of restricted bandwidth, or may have some intermediate nodes or networks lying in between them, processing cross-traffic (data or voice).

In this thesis, we built the topology depicted in Figure F.7 for testing the three adaptive strategies described in Section E.4. We used two notebooks and a PC. Each notebook was located in a different network (192.168.0.0 and 172.16.0.0), and run four virtual machines, which implemented the SIP clients. The PC was configured to work as a gateway between these networks.

All virtual machines in the topology run the Linux Ubuntu 10.04 64-bits, supported by the Intel Integrated Performance Primitives (IPP) library [82], needed for adding the G.729 and G.726 codecs to the PJSIP. The gateway run the Linux Ubuntu 11.04 32-bits.

The four virtual machines of each notebook implemented the adaptive strategies AS_1, AS_2 and

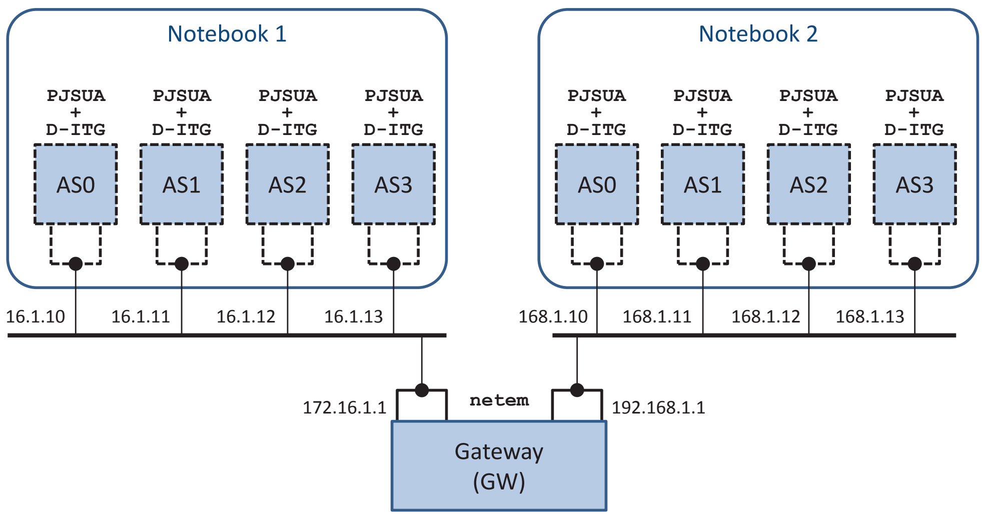


Figure F.7: Network topology adopted in the experiments.

AS_3, plus the nonadaptive reference AS_0. Additionally, each one of the eight virtual machines run a traffic generator program that sent and received background VoIP and TCP traffics among the machines. The evaluated VoIP traffic was sent from a machine AS_i of one notebook to the correspondent AS_i machine of the other notebook. Concurrent and background traffic was sent as depicted in Figure F.8.

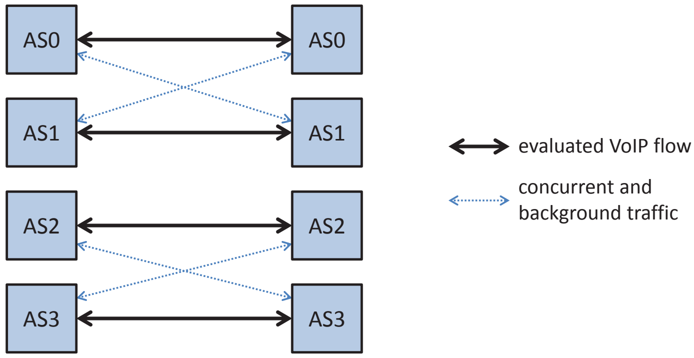


Figure F.8: Overview of traffic exchange between the virtual machines of the experiment.

To inject background traffic, we employed the D-ITG tool [22]. TCP traffic was modeled with inter-packet time following a Pareto distribution of shape parameter $\alpha = 1.5$ and scale parameter $\beta = 0.5$. We employed the the packet size same characteristics used by Myakotnykh & Thompson [143] and Ngamwongwattana [148], as discussed in the previous section.

The gateway was configured to restrict the bandwidth of all IP addresses to only 256 kbit/s, by means of the traffic control settings of the Linux kernel (`tc` command). In some experiments, such as those intended to check the terminals' response to a bounded input (stability property), we employed the emulation tool Netem [192] for controlling the injection of packet loss into the network.

F.6 Randomization and Replication

Randomization is a schedule for allocating treatment material and for conducting treatment combinations in a design of experiment such that the conditions in one run neither depend on the conditions of the previous run nor predict the conditions in the subsequent runs [149]. *Replication* is the repetition of an experimental condition so that the variability associated with the phenomenon can be estimated. According to Crawley [44], *replication increases reliability*, whereas *randomization reduces bias.*

To randomize the experiments, each replication used a different speech file as voice input of the generated VoIP calls. The speech content was taken from the speech database available at Supplement 23 of ITU-T P-series Recommendations [86], and the speech files used in the DCR tests described in Chapter D. Then we applied the Stern-Mahmoud-Wong 8-state model [186] to produce speech files with the talkspurt-silence characteristics of two-way conversational speech.

Next, a *sufficient* number of replications was run for each treatment to obtain the measurements needed for statistically checking the self-adaptive properties elicited in Section E.4. Usually, the higher the sample size is, the higher is precision. However, the cost of the inquiry constitutes another important constraint. Thus, one should find the minimal number of observations to take into account in order to detect a significant difference (if there is one) between the different treatments [49].

According to Dodge [49] and Crawley [44], in cases where the sampling distribution is not normally distributed, such a detection is approximately valid when the sample size n is larger than 30. Hence, in all experiments carried out in this thesis, we run 30 replications of each treatment under test, using a different pair of speech files on each replication.

F.7 Performance Evaluation of Adaptive VoIP Applications

The performance evaluation of the adaptation strategies AS_1, AS_2 and AS_3 against the reference nonadaptive strategy AS_0 consists in verifying the compliance of those three strategies to six of the self-adaptive properties listed in Section E.4: stability, accuracy, short settling time, small overshoot, robustness and scalability. From the results of comparison tests, one can tell which strategies are more efficient or not than the reference VoIP terminal.

To test the six properties, we performed two kinds of experiments, described in next subsection:

1. Injection of packet-loss peaks; and
2. Injection of concurrent calls and background traffic.

F.7.1 Injection of packet-loss peaks

In this group of experiments, we set the AS_i terminals located on the 192.168.0.0 network to originate 30-second calls to the correspondent AS_i terminals located on the 172.16.0.0 network. At 10 s after the beginning of the call, we inject a peak of packet loss in the network during another 10 s. The final 10 s of the call, there was no impairment affecting the call.

The peak levels assumed one of the following values: 0%, 1.5%, 3%, 5%, 10%, and 15%. Those levels were the same used by Raake [160] for deriving I_e and *Bpl* values for some codecs. The granularity is higher for low values of loss because of the relative error. For example, if packet loss presents an oscillation of 1% in the experiments with 1.5%-level, this will be much more significant than in the experiments with 15%-level.

Note that each level gave origin to a different experiment run. From these experiments, we measured two outputs for each adaptive strategy AS_0, AS_1, AS_2, AS_3:

1. Instantaneous MOS (MOS_{inst}) across time.
2. Number of reINVITE requests generated across several levels of peak loss.

This set of experiments was used for evaluating five of six adaptive properties: stability, accuracy, short settling time, small overshoot, and robustness. Results and analysis are discussed in the next chapter.

F.7.2 Injection of concurrent and background traffics

In this group of experiments, we set the AS_i terminals located on the 192.168.0.0 network to originate 2-minute calls to the correspondent AS_i terminals located on the 172.16.0.0 network. The gateway was configured to limit the download and upload bandwidths up to 256 kbit/s, for the IP addresses of all VoIP terminals.

Each virtual machine in Figure F.7 was configured to originate and receive concurrent and background traffics. Because the `tc` tool in Linux guarantees the reservation of 256 kbit/s, the only way we find to disturb the calls was by injecting the traffics in the same machines used for evaluating the adaptation strategies.

The concurrent traffic consisted of ten VoIP flows, carrying one G.711 frame per packet. In the beginning of the call, all ten voice flows were active, and two of them were deactivated at every 20 s. So, at the last 20 s of the call, there were no concurrent VoIP call

The background traffic consisted of twenty TCP flows. Inter packet time was modeled by a Pareto distribution with shape parameter $\alpha = 1.5$ and scale parameter $\beta = 0.5$. Packet sizes were fixed in three values [39, 190]: 60% of the packets were 40 bytes, 25% were 550 bytes, and 15% were 1500 bytes. This gave an average consumption of

$$\frac{20 \times (0.6 \times 64\mathrm{B} + 0.25 \times 550\mathrm{B} + 0.15 \times 1500\mathrm{B}) \times 8^{\mathrm{bit}}/_{\mathrm{B}}}{\frac{1.5 \times 0.5}{1.5 - 1}\mathrm{ms}} \approx 41\,\mathrm{kbit/s}.$$

From these experiments, we measured three outputs for each adaptive strategy AS_0, AS_1, AS_2, AS_3:

1. Instantaneous MOS ($\mathrm{MOS}_{\mathrm{inst}}$) across time.
2. Instantaneous end-to-end delay (T_{a}) across time.
3. Instantaneous packet loss (Ppl) across time.

This set of experiments was used for evaluating the scalability property and for reinforcing the analysis of the peak loss experiments. Table F.5 summarizes the experimental parameters to be adopted to evaluate the adaptation strategies. Results and analysis are discussed in the next chapter.

Table F.5: Summary of experimental parameters.

Parameter	**Experiment**	
	Injection of packet-loss peaks	**Injection of concurrent calls and background traffic**
Call Duration	30 s	120 s
Concurrent calls	—	10, 8, . . . , 0 at every 20 s
Background traffic	—	20 TCP flows, with Pareto-distributed inter-packet time ($\alpha = 1.5$, $\beta = 0.5$), with packet sizes of 64 B (60%), 550 B (25%), and 1500 B (15%)
Network packet loss	levels of 0%, 1.5%, 3%, 5%, 10%, and 15%, from 10 s to 20 s	—
Replications	30	
Speech model	Stern-Mahmoud-Wong 8-state model [186]	
Bandwidth	256 kbit/s	
MAC protocol	Gigabit Ethernet 1000 Base-T	

F.8 Summary

In this chapter, we reviewed some models of voice traffic characterization. They were divided according to the observation perspective: *user-demand level*, suitable for studying the capacity of telephony systems; *speech level*, suitable for investigation studies involving injection of speech material, such as the present thesis; and *packet level*, suitable for simulation studies, in which it is not feasible to encode and decode speech material.

Next, we reviewed some models for representing the alternating behavior of talkspurt and silence during a conversation between two parties, justifying our choice on the Stern-Mahmoud-Wong 8-state model [186]. Some considerations about background traffic, topology, randomization and replications were also covered here.

Finally, we presented the experiment environment that we used for testing the three adaptation strategies AS_1, AS_2 and AS_3 plus the nonadaptive reference AS_0. In the next chapter, we present and analyze the results of these experiments.

Appendix G

Result Analysis

. . . durch planmässiges Tattonieren.
(. . . through systematic, palpable experimentation.)
Carl Friedrich Gauss (1777–1855)
Response, when asked how he came upon his theorems

Contents

This chapter presents and discusses the results of the experiments described in the previous one. It is divided into three sections, correspondent to the output of the two groups of tests conduced in this thesis: injection of packet-loss peaks, and injection of concurrent and background traffics. The degree of compliance to the adaptive properties is also discussed along with the text.

G.1 Response to Packet-Loss Peak – Instantaneous MOS

Figures G.1 to G.6 show the response of the adaptive strategies to a variable packet-loss peak, injected during the intermediate 10 s of a 30-second call. Peak levels assumed the following values: 0%, 1.5%, 3%, 5%, 10%, and 15%. RTCP XR packets were exchanged between the PJSUA terminals at every one second, AS_0 included. Each point in the graphs indicate an instantaneous MOS measurement.

The horizontal dashed line represents the minimum threshold of MOS=3.6 for toll-quality, which corresponds to the Goal 1 of the adaptive mechanisms. The shaded area around the plotted lines represents the 95%-level confidence interval, taken from the 30 replications of each experiment. Each experiment used a different pair of speech files, so as to randomize the effect of such factor on the experimental environment.

Note that a signaling is also part of the call. So, the graphs represent the duration of media flow only, and do not take into account the time required for signaling exchanging.

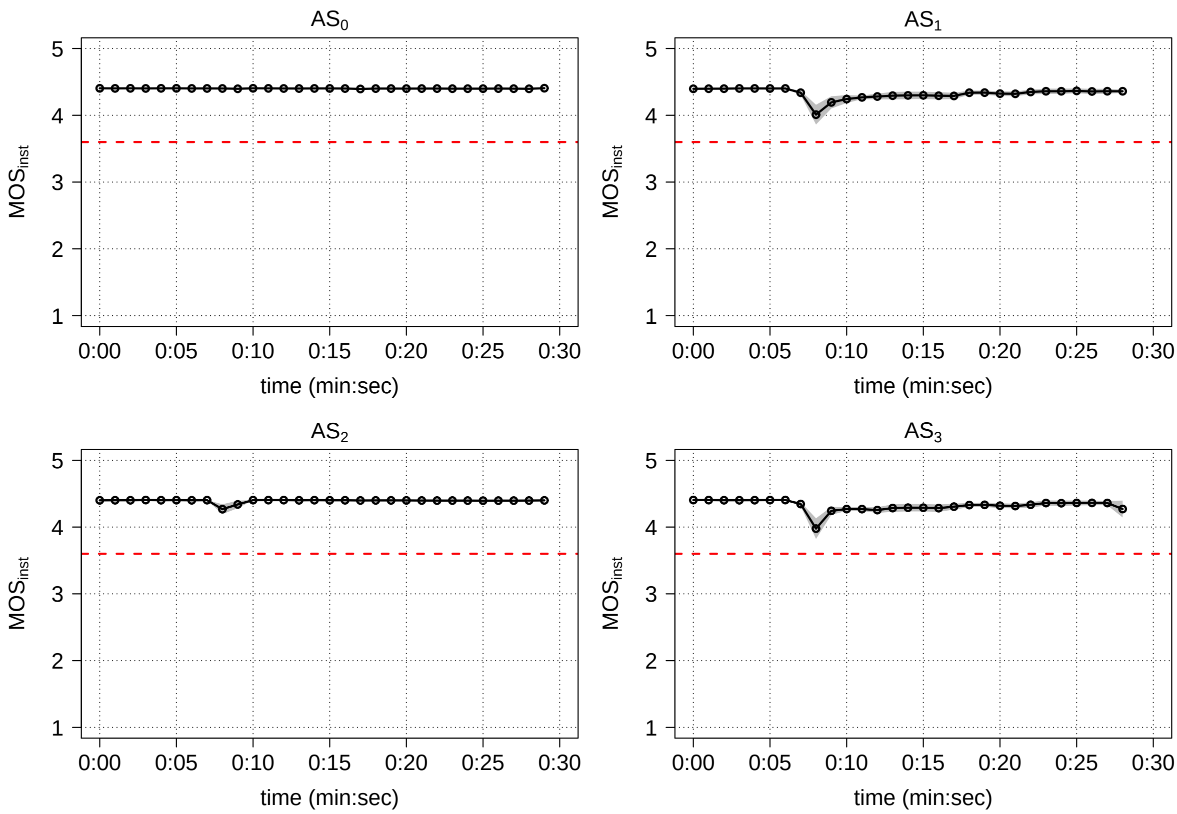


Figure G.1: Quality response to a packet-loss peak of 0%.

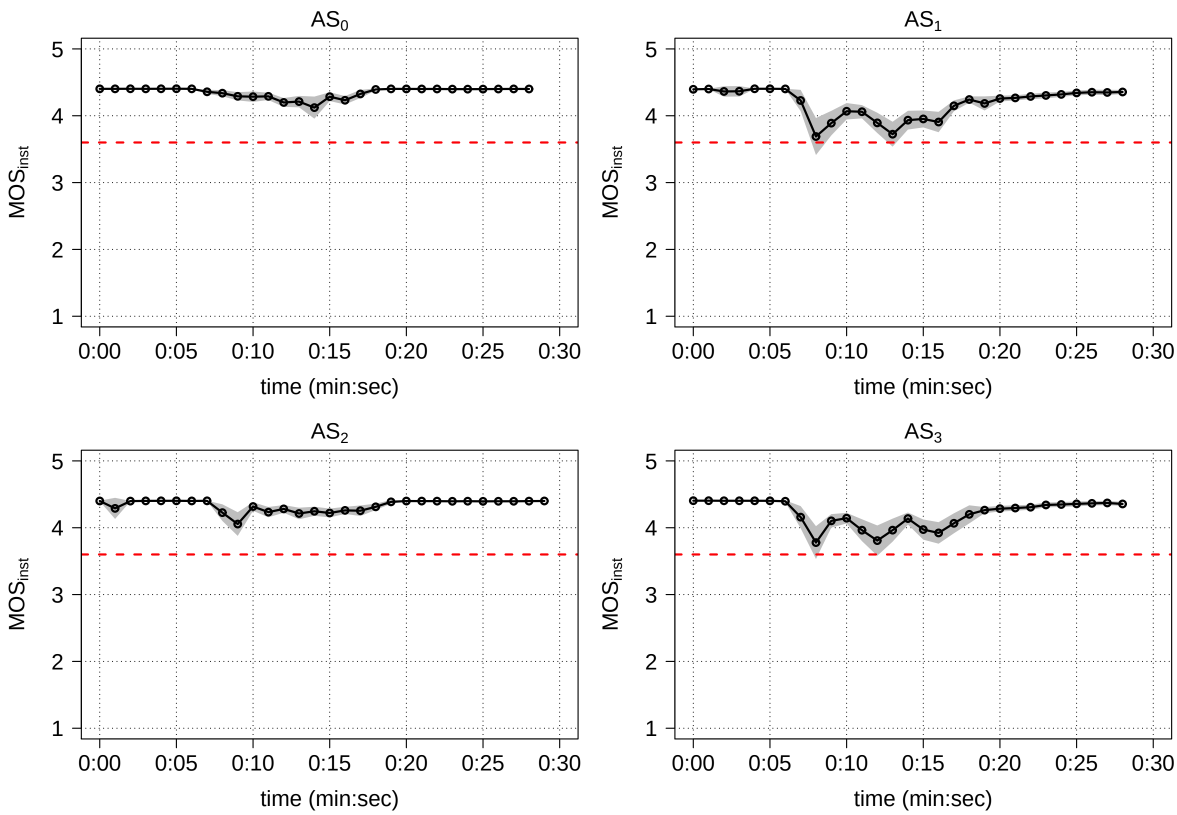


Figure G.2: Quality response to a packet-loss peak of 1.5%.

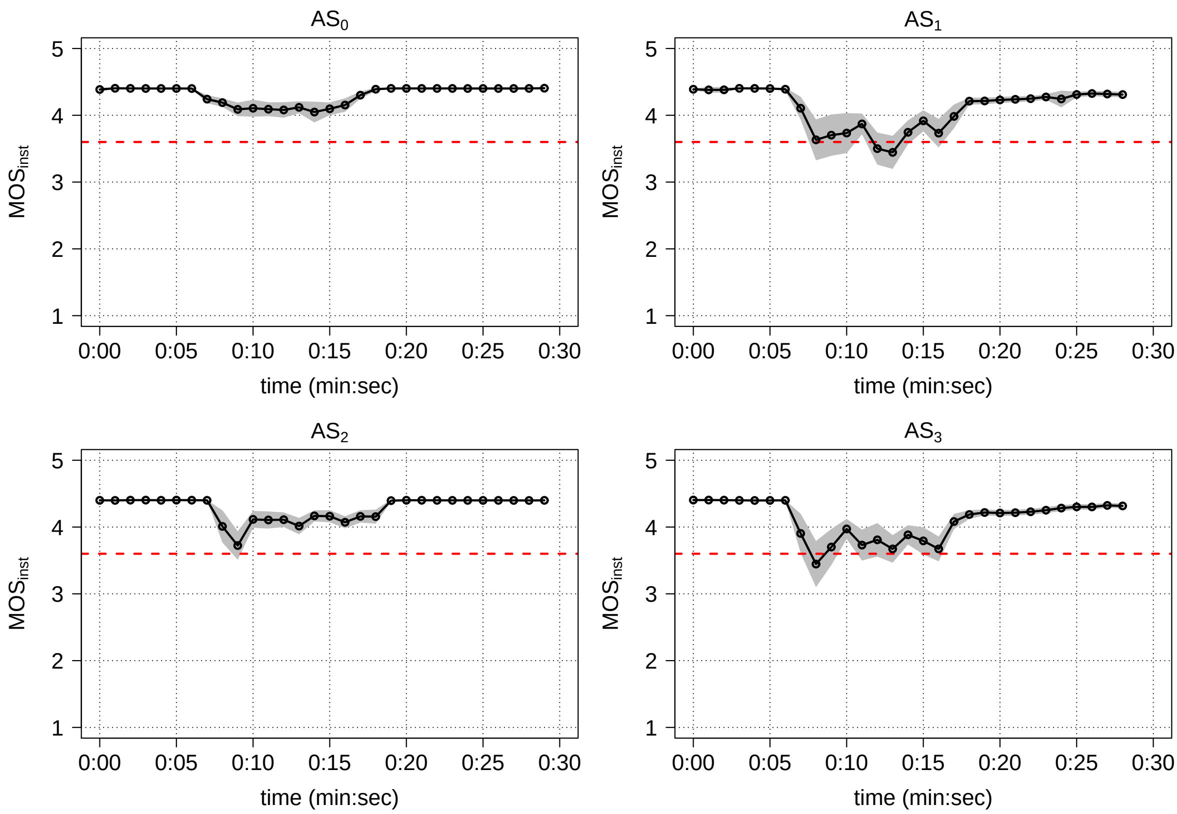


Figure G.3: Quality response to a packet-loss peak of 3%.

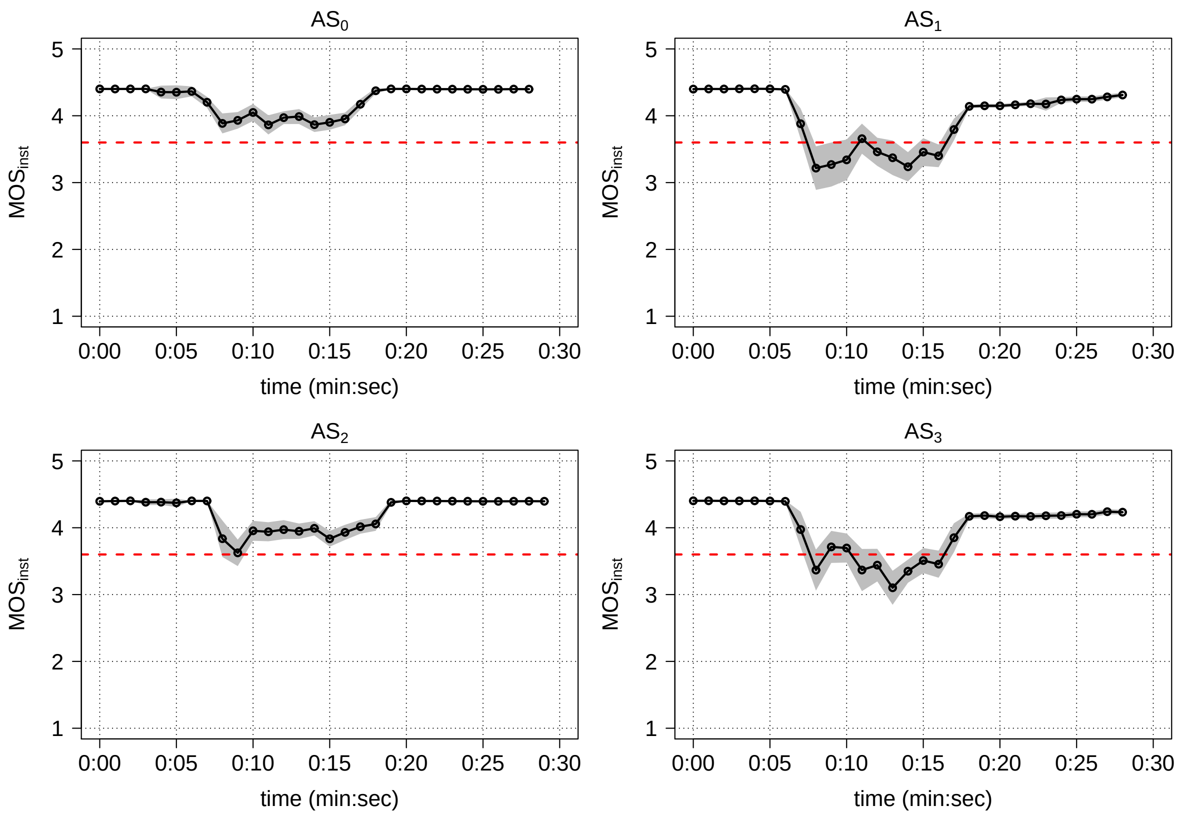


Figure G.4: Quality response to a packet-loss peak of 5%.

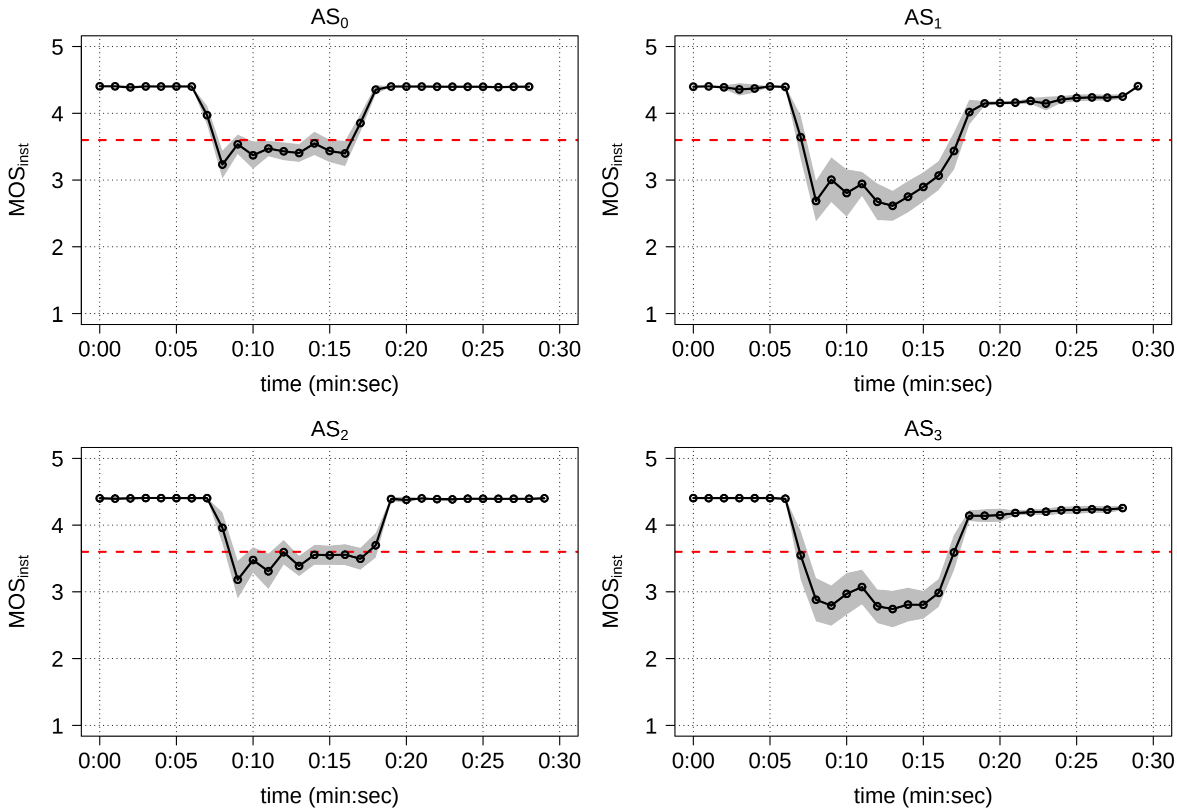


Figure G.5: Quality response to a packet-loss peak of 10%.

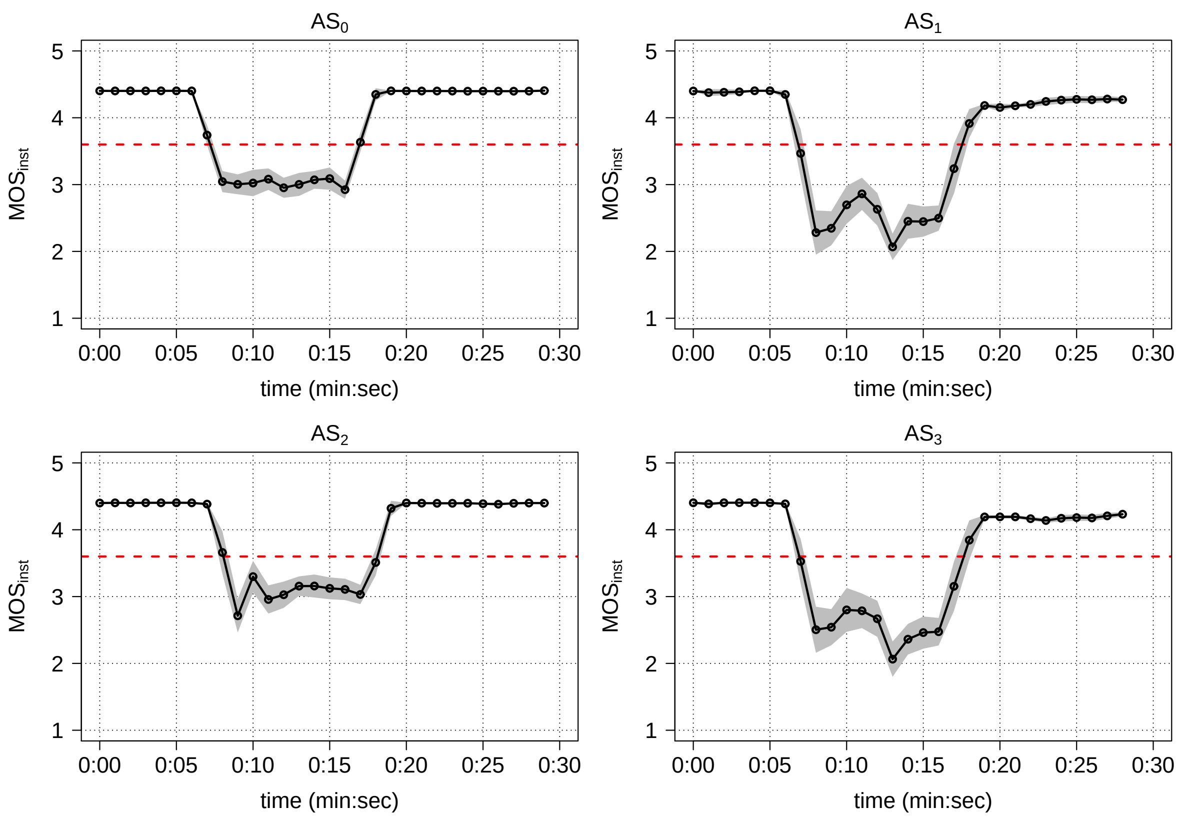


Figure G.6: Quality response to a packet-loss peak of 15%.

Stability property analysis. Injecting a 0% peak of packet loss should be regarded as different from not injecting packet loss at all. So, when we tell to the traffic controller of the Linux kernel to inject a packet loss of 0%, some disturbance – yet small – is added to the voice flow.

Now, take Figure G.1 as first example. One can see that the non-adaptive strategy AS_0 is not affected by the 0% packet loss injection. The AS_2 terminal (adaptation decided by the receiver and taken by the remote sender) is a little affected at the time point of 8 s. However, the terminals AS_1 (adaptation decided and taken by the sender) and AS_3 (adaptation decided by the receiver and taken by the local sender) were noticeably affected by such disturbance, not only at the moment of peak injection but during all the period of the peak.

This observation is aggravated as long the packet loss level of the peak is increased. For example, in Figure G.1, the injection of a peak of 1.5% packet loss causes a noticeable oscillation of the MOS_{inst} values when using the terminals AS_1 and AS_3. For AS_0 and AS_2, MOS_{inst} has a small and smooth decrease.

This can be explained by two reasons:

1. AS_2 is the adaptive strategy that injects the least number of reINVITE requests (see Figure G.7). This not only avoids oscillations in codec switching, but also do not overload the already disturbed network with more control messages.
2. The high granularity of instantaneous MOS measurements (one per second) may lead to a hasty assessment of the actual voice quality, which leads, in turn, to hasty and maybe unnecessary adjustments.

Accuracy property analysis. Accuracy is related to how close the managed system approximates to the desired state defined by the adaptation goals. According to this criteria, one can see from figures G.3 and G.4 that the adaptive strategies AS_1 and AS_3, in the way they were implemented, are not accurate, because they eventually worsen the speech quality while not dealing properly with disturbances in the network. Under the same levels of packet loss peak (3% and 5%), the nonadaptive strategy do not let speech quality to go down the minimum threshold of MOS=3.6, whereas AS_1 and AS_3 noticeably deteriorate speech quality.

The MOS_{inst} given by the terminal AS_2 touches the threshold line, but if consider the confidence interval, it also fails on the accuracy check. The reason for this is that the new codec switched to presents a lower I_e value, affecting the overall value of the instantaneous MOS.

Short Settling Time property analysis. From figures G.1 to G.6, one can see that the settling times of the adaptive strategies was the period of the disturbance. The injected packet loss peak lasted 10 s, and the observed perturbation on speech quality lasted about the same quantity in these graphs.

Statistical analysis. When visually comparing figures G.1 to G.6, AS_1 and AS_3 clearly perform worse than the reference condition AS_0, but the same cannot be said about AS_2. Hence, we performed a statistical test to verify it so.

Due to the repeated MOS measurements (MOS_{inst}) over time, points that are close together in time are in some sense more closely connected than points that are widely separated in time [120]. Therefore, we cannot apply an ANOVA test, because it does not reflect this sequential structure, in the fixed effects (adaptation strategy and packet loss), in the random effects (replication), and in the correlation structure (time dimension).

Table G.1: Fixed effects for the interaction among adaptation strategy, packet loss peak and time on the MOS response for the packet loss injection experiments.

	Value	Standard Error	Degrees of freedom	t-value	p-value
Intercept	4.263	0.143	22057	29.757	0.00
AS_1	-0.074	0.007	22057	-10.136	0.00
AS_2	-0.001	0.007	22057	-0.110	0.91
AS_3	-0.073	0.007	22057	-9.934	0.00
Packet loss peak	-0.102	0.003	22057	-32.430	0.00
Time	0.009	0.005	22057	1.823	0.07

So, we applied the analysis of repeated measures data [54, 120] for interpreting the bunch of data collected during the packet-loss peak experiments. This modeling approach often allows insights that are hard to gain from approaches that ignore or do not take advantage of the sequential structure [120].

Particularly, we used the *linear mixed-effects* models for repeated measures data, as described by Everitt & Hothorn [54] and Maindonald & Braun [120]. The essential feature of this model is that correlation amongst the repeated measurements on the same unit arises from shared, unobserved variables [54].

Additionally, we used dropout analysis in longitudinal data to deal with missing instantaneous MOS values. This is likely to happen specially at the end of the call, because sometimes the duration of media flow session can vary in one or two seconds across the multiple replications and among the four VoIP terminals.

Table G.1 shows the output of the linear mixed-effects analysis computed by the R tool [159]. We modeled the instantaneous MOS as the response variable of interest, given by the interaction of the following explanatory variables: adaptation strategy, packet loss peak and time. The call replications were modeled as random effect.

The low p-value for AS_1 and AS_3 tells us that these factors have a significant effect on the model. This means that they contribute to the variability of MOS values in comparison with the AS_0 terminal, that is, AS_1 and AS_3 are significantly different from AS_0. In contrast, there is no evidence to reject that AS_0 and AS_2 have different effect on the collected data ($p = 0.91 \gg 0.05$).

G.2 Response to Packet-Loss Peak – reINVITE Requests

Figure G.7 shows the number of reINVITE SIP requests exchanged by the AS_1, AS_2, and AS_3 during the call period. Note that AS_0 is not adaptive, hence no requests were sent by it.

AS_2 sent just one reINVITE request in all replications for all packet loss levels. So, its confidence interval is zero. In turn, AS_1 and AS_3 presented a very similar behavior of sending an increasing number of reINVITE requests as long as the level of packet loss injected in the network was increased.

It is not possible to distinguish, from Figure G.7, which terminal, whether AS_1 or AS_3, sends the higher amount of reINVITE messages. So, for this purpose, we run a Wilcoxon test, a non-parametric alternative to the Student t-test.

Since the p-value was 0.29 (> 0.05), we can claim that there is no significant difference between the terminals AS_1 or AS_3 in regard with the number of reINVITE requests sent during a short call, across different levels of injected network packet loss.

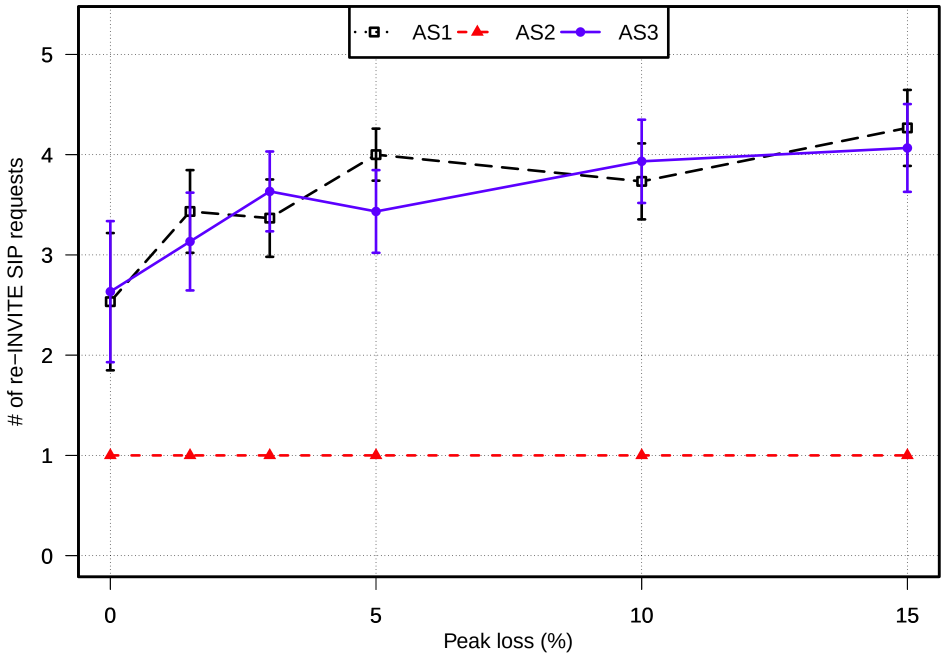


Figure G.7: Number of reINVITE SIP requests exchanged between the terminals during a 30-second-call, at different levels of packet loss peak.

Figure G.8 illustrates the transaction of signaling and media flow between two terminals AS_3, during a 30-second call submitted to a packet loss peak of 15%. It was taken by means of screen shots from the Wireshark tool.

Small Overshoot property analysis. From Figure G.7, one can see that AS_1 and AS_3 inject a high number of reINVITE requests. Also, the high rate of RTCP XR packet sending showed to be inefficient for the terminals to recover from disturbances.

For example, for the AS_3 terminal, which adapts the local sender, there is no need in reporting VoIP metrics to the remote sender so frequently. Thus, it should be modified to not overload the network, which may improve its performance.

G.3 Response to Concurrent and Background Traffics

Figures G.9 to G.12 exhibits a triplet of graphs, which shows the evolution of instantaneous MOS, end-to-end delay (T_a) and packet loss rate (Ppl) during the 2-minute calls. Each figure corresponds to the output of one out the four strategies AS_0, AS_1, AS_2, and AS_3. Along with the VoIP session carried out by each evaluated terminal, the correspondent virtual machine generated concurrent and background traffics by means of the D-ITG tool [22].

Concurrent traffic was gradually decreased from ten to zero voice flows, in steps of two, at every 20 s. Background TCP traffic was modeled with inter-packet time following a Pareto distribution of shape parameter $\alpha = 1.5$ and scale parameter $\beta = 0.5$. Amongst the twenty TCP flows injected in the network by each virtual machine, the size of 60% of the data packets were 40 B, 25% were 550 B, and 15% were 1500 B.

The shaded areas correspond to the 95% level confidence interval of the three measurements. The dashed lines on each plot corresponds to the following thresholds: MOS=3.6, $T_a = 150$ ms, and $Ppl = 5\%$. MOS, delay and packet loss ranges are all the same across figures G.9 to G.12.

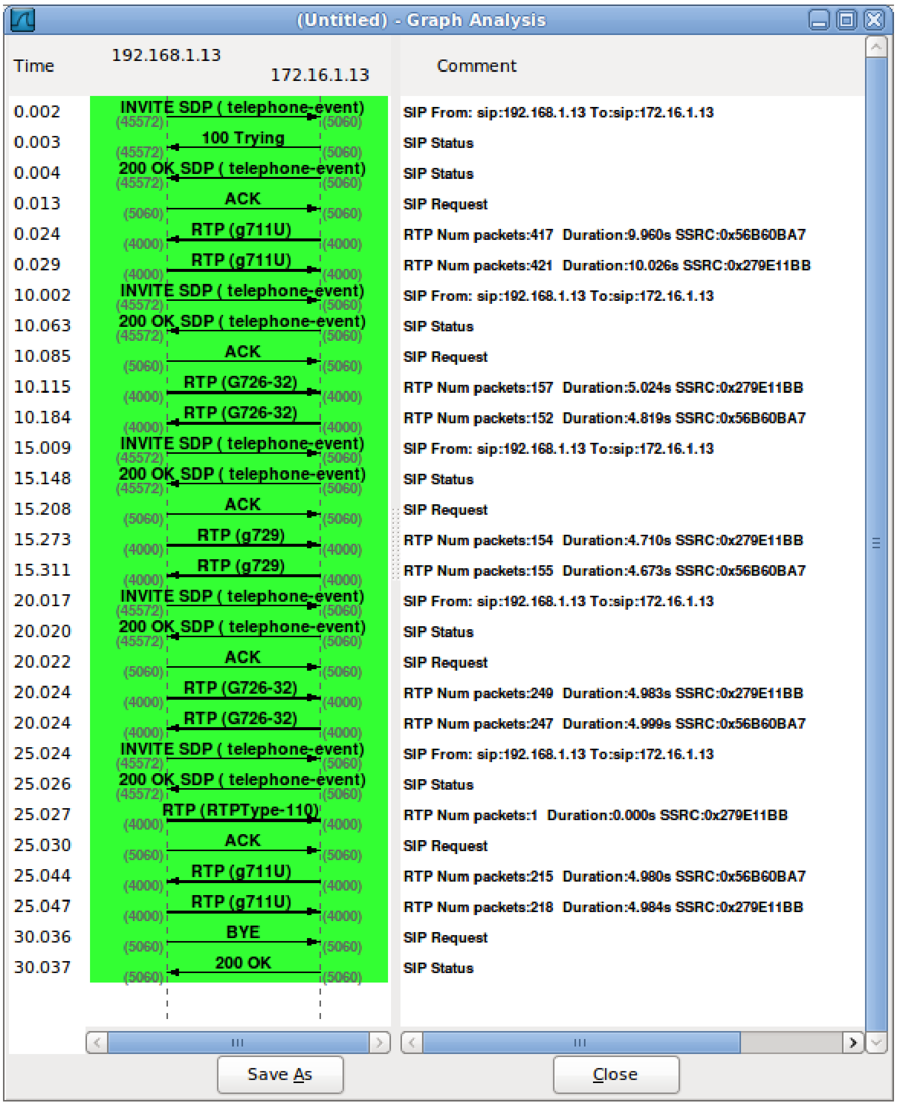


Figure G.8: Signaling and voice flow during a 30-second-call between two AS_3 terminals, with a peak-loss of 15% from 10 s to 20 s.

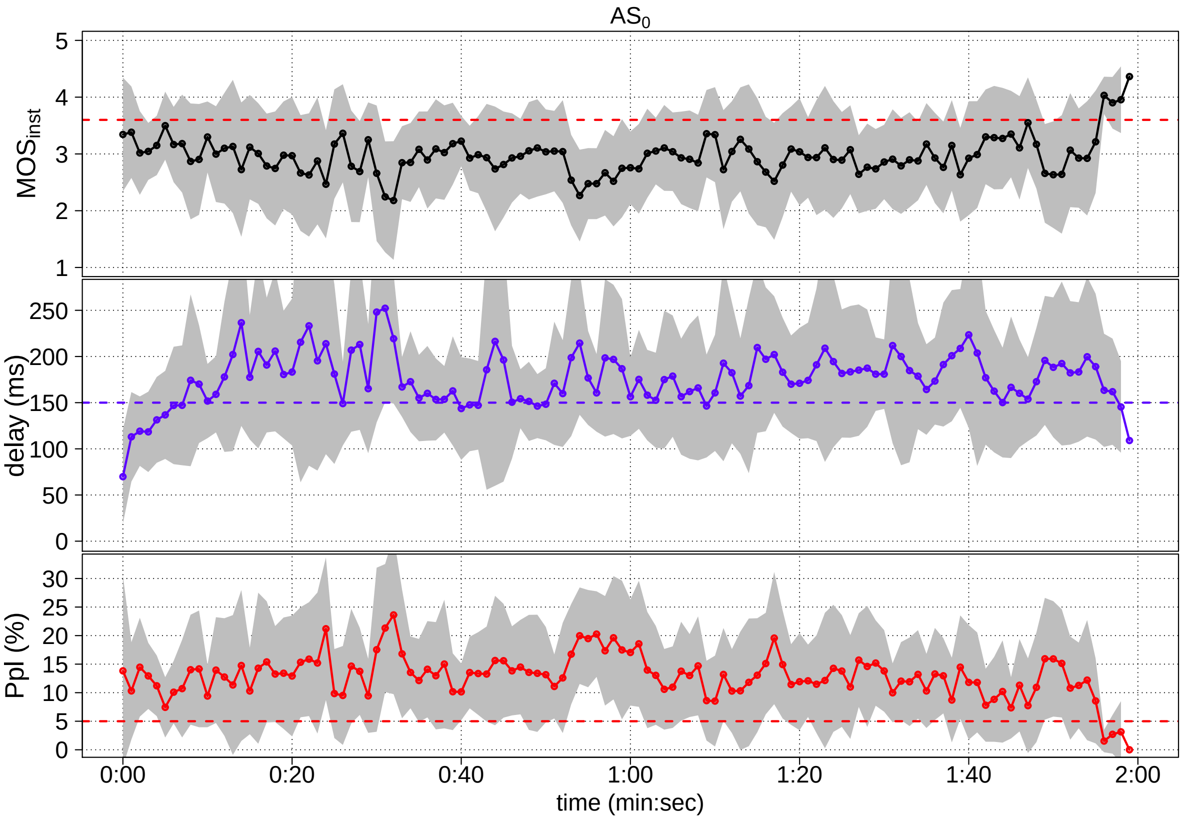


Figure G.9: MOS and delay variation during the call for AS_0.

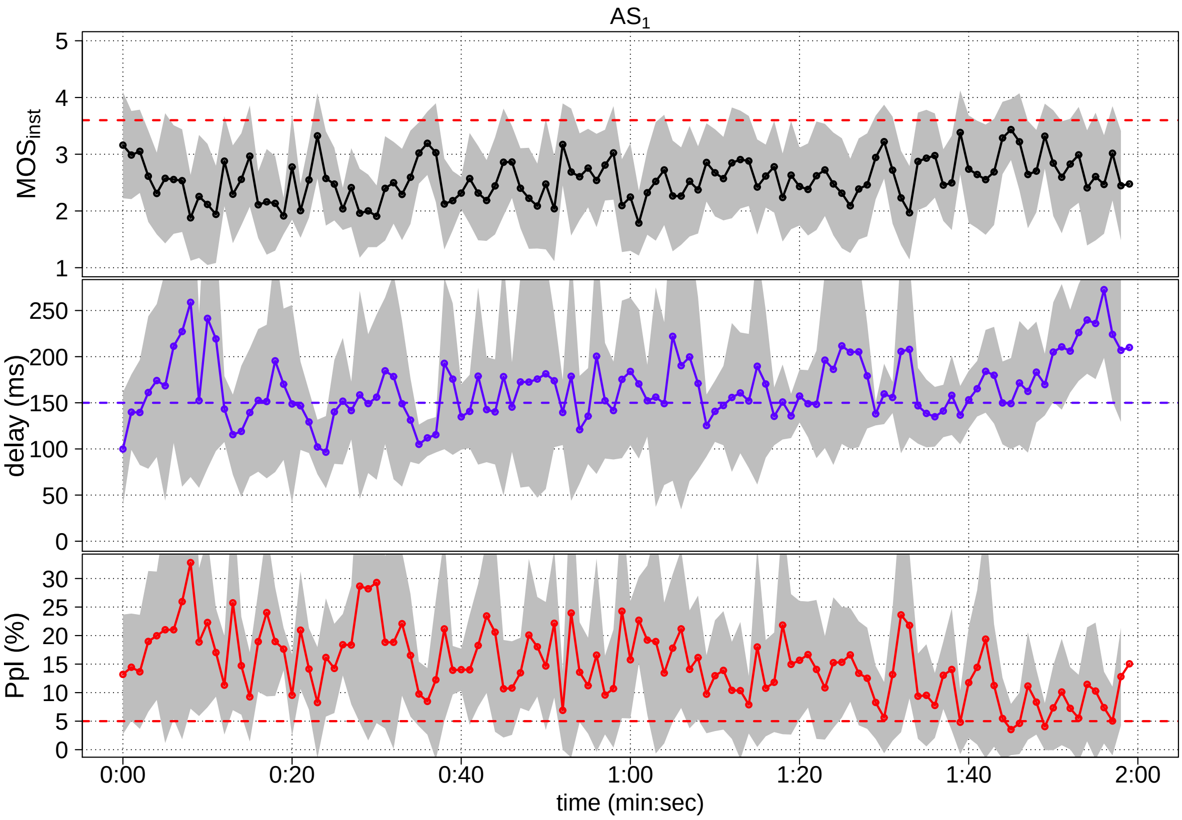


Figure G.10: MOS and delay variation during the call for AS_1.

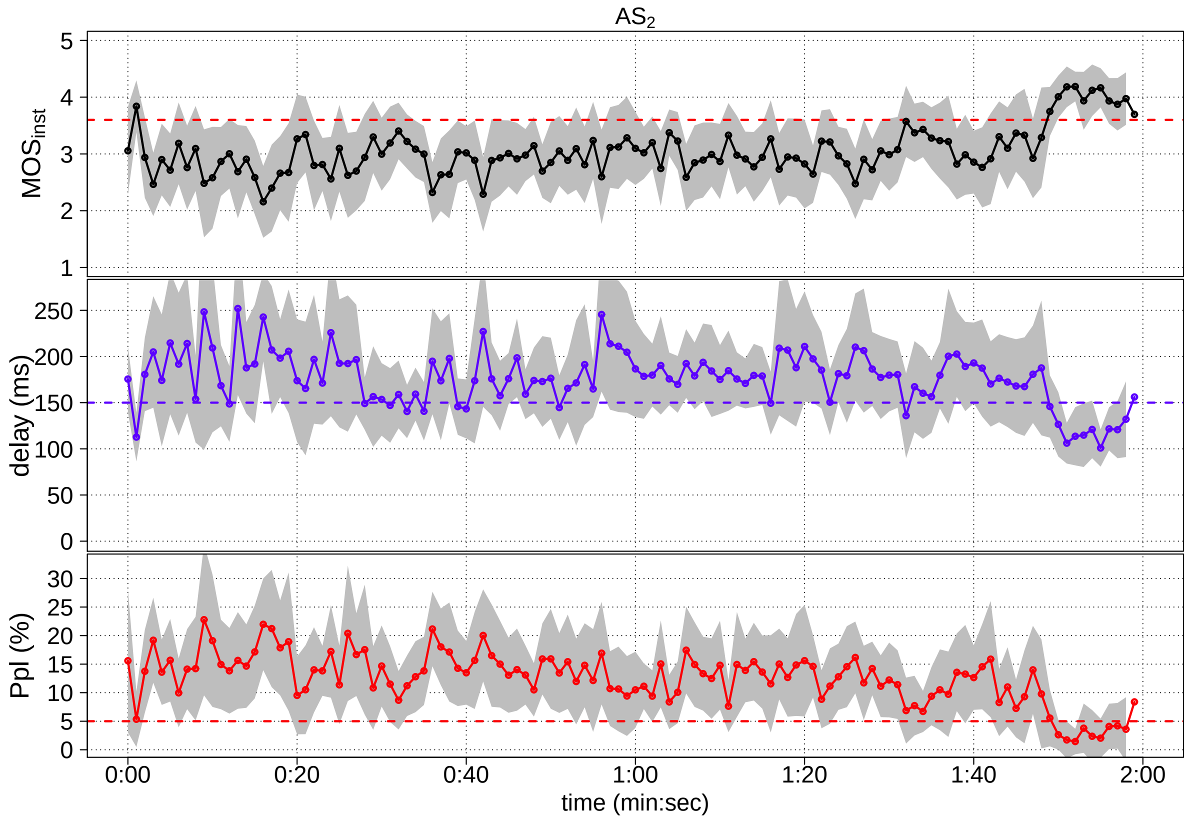


Figure G.11: MOS and delay variation during the call for AS_2.

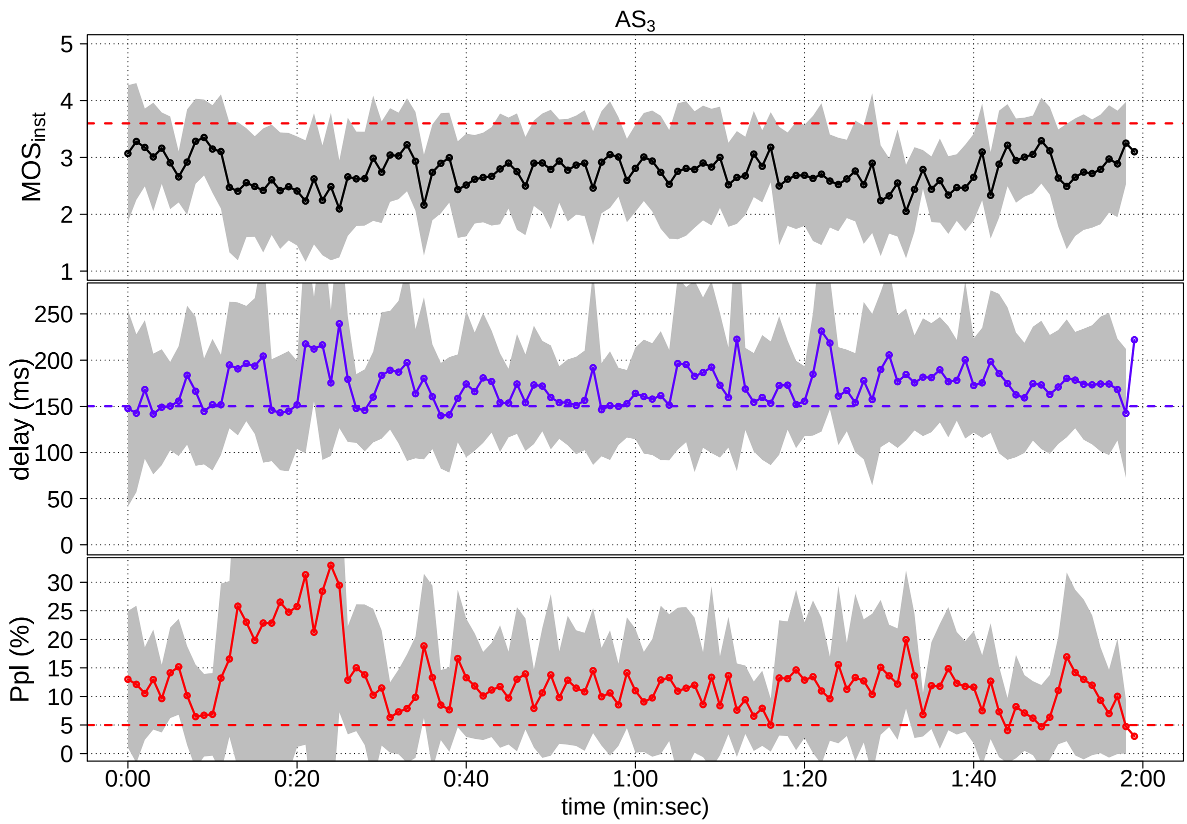


Figure G.12: MOS and delay variation during the call for AS_3.

Table G.2: Fixed effects for the interaction among adaptation strategy, packet loss peak and time on the MOS response for the traffic injection experiments.

	Value	Standard Error	Degrees of freedom	t-value	p-value
Intercept	2.806	0.201	3782	13.941	0.00
adpt.strgAS1	-0.407	0.040	3782	-10.053	0.00
adpt.strgAS2	0.077	0.040	3782	1.914	0.06
adpt.strgAS3	-0.216	0.041	3782	-5.328	0.00
Time	0.003	0.002	3782	1.432	0.15

Robustness and Scalability properties analysis. None adaptation strategy AS_1, AS_2 and AS_3, in the way they were implemented, showed to be robust to concurrent traffic during the first 100 s of the call, where there was at least two concurrent VoIP flows generated by another application, the D-ITG.

In regard with scalability, AS_1 and AS_3 showed poor performance also, giving MOS_{inst} values below the 3.6 threshold. AS_2, however, showed to be scalable to background TCP traffic alone during the final 10 s of the call, the only period in which it accomplished its design goal.

Table G.2 shows the output of the linear mixed-effects analysis computed by the R tool [159] on the data output of this group of experiments. This time, we modeled the instantaneous MOS as the response variable of interest, given by the interaction of the following explanatory variables: adaptation strategy and time. Again, the call replications were modeled as random effect.

The low p-value for AS_1 and AS_3 tells us that AS_1 and AS_3 are significantly different from AS_0. In turn, the p-value of AS_2 is approximately 0.06, around the hypothesis rejection threshold of 0.05. Thus, during longer calls, there is no strong evidence that it performs equal to or different from the nonadaptive terminal AS_0.

G.4 Summary

Oppositely to expectations, the adaptive strategies did not showed a better performance than the nonadaptive terminal, except for AS_2, which showed a better performance in the presence of background TCP data only traffic (the final 10 s of the call in Figure G.11). AS_1 and AS_3 performed worse than AS_0 in all experiments. AS_2 performed as the same as the nonadaptive terminal in the experiments of injection of packet-loss peaks.

Nevertheless, these results are not conclusive for rejecting adaptation as a solution to the voice quality optimization problem. Note that we used *adaptation strategy* as explanatory variable of the experiments, but it hinders other aspects, such as the sample size needed for determining the instantaneous MOS (i.e., granularity); the RTCP XR sending rate; and the minimum transient time between consecutive adaptation actions.

Moreover, as future work, more appropriate assessment methods should be developed for adaptive VoIP applications. For example, artificially injecting a constant level of packet loss or delay during all call prevents any attempt of the adaptation mechanism to overcome the disturbance, because no matter which adjustment action is taken, the experienced disturbance will not change. Furthermore, a constant level of some impairment is not normally observed in real computer networks.

Appendix H

Conclusion

I do not fear computers. I fear lack of them.
Isaac Asimov (1920–1992)

Contents

VoIP calls are susceptible to a variety of impairments, such as lossy compression, network delay, packet loss rate, oscillating packet loss behavior, among others. Using the principles of self-adaptive software for overcoming such problems seems to be a reasonable solution, as glimpsed by many approaches in the literature.

Although some research effort has been carried out to develop adaptive solutions for managing the speech quality of ongoing VoIP calls at the application layer, there is a lack of works tackling this problem in actual VoIP terminals running across actual computer networks. The present work tried to take the first steps in this sense.

In this final chapter of the thesis, we recall the contributions and suggest some works and challenges for future investigation.

H.1 Contributions

The contributions brought by this thesis are listed below:

1. *Perception of Codec Switching.* The results of the DCR tests evidenced that degradation during codec switching must be considered by an adaptation mechanism, because low-bitrate but high-quality codecs can be better accepted than medium-bitrate but low-quality ones. Also, the perceptibility of codec switching is not commutable, because transitions from a high- to a low-quality codec are regarded as deterioration and the converse is regarded as improvement.

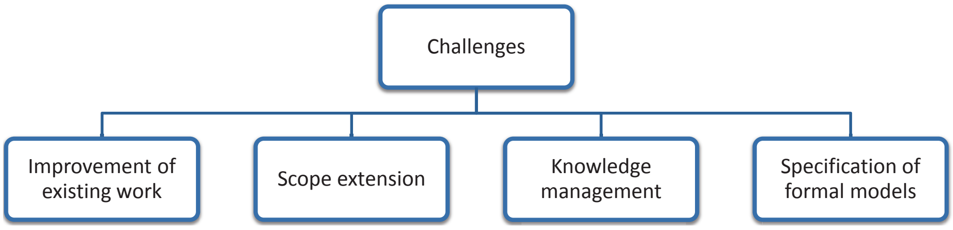


Figure H.1: Challenges in designing VoIP adaptation systems.

2. *General Codec Precedence for the Switching Decision.* Instead of running offline tests to determine the best combinations of codec switchings, we used an algorithm that decides it in run-time, based on the equipment impairment factor (I_e).

3. *Visibility of the Control Loop.* We formalized the project of an adaptive VoIP terminal according to the principles of self-adaptive software. We not only put the control loop in evidence, but also focused on goals to be pursuits and properties to be verified. Although the evaluation tests did not confirmed the efficiency of such an approach, they did not also reject it. Instead, they showed a variety of concerns not envisioned in previous simulation studies.

4. *Survey of Best Practices on VoIP Measurement Experiments.* In Chapter F, we presented a comprehensive survey on the main concerns needed for testing VoIP applications by means of measurements.

5. *Software.* We extended the code of the Open Phone Abstraction Library (OPAL) [153] for calculating the E-model at run-time and sending it into RTCP XR packets. Both contributions are incorporated by the official distribution of this software. Furthermore, we extended the code of PJSIP [155] for calculating the E-model, which will be submitted after some needed refinements.

6. *Research papers.* During the thesis project, we produced the following works:

   - *Synchronizing Web Browsing Data with Browserver* [29], a conference paper that tackles the self-adaptation problem over a synchronization service of web browsing data across multiple devices.

   - *Dynamical Management of Dejitter Buffers Based on Speech Quality* [196], a conference paper that presents an adaptive solution for managing speech quality by switching the current dejitter buffer algorithm according to measured network conditions

   - *Pursuing Credibility in Performance Evaluation of VoIP over Wireless Mesh Networks* [138], a book chapter that.

   - *Survey on Application-Layer Mechanisms for Speech Quality Adaptation in VoIP* a survey paper to the ACM Computing Surveys. We received an acceptance e-mail with major revisions, applied the changes, and we are expecting the final approval.

H.2 Future Work

New opportunities arise when we regard the problem of QoS control of VoIP under the perspective of self-adaptive software. This vision poses new challenges for developing and validating VoIP systems with adaptation characteristics. This section identifies the challenges that lie in the intersection between VoIP and self-adaptive software, which should be tackled in future work. As outlined in Figure H.1, they are grouped into four basic key-points, which are detailed in the remainder of this section.

H.2.1 Improvement of Existing VoIP Adaptation Mechanisms

Here, we highlight the main challenges related to the enhancement of existing VoIP adaptation mechanisms, sparsely mentioned throughout this thesis.

- *Decision metrics.* From tables C.2 to C.6, we saw that MOS is the decision metric mostly used for determining whether adaptation is needed. Because of the widespread use of mobile devices, memory and power constraints should also be considered as decision metrics during the planning activity. The challenge here is how to model such parameters and their interrelationship with speech quality metrics.

- *Adaptation robustness.* Most of the works surveyed in Chapter C consider the influence of network impairments over the RTP packets only, but not over the control packets. Hence, they wrongly take for granted that control information will always be delivered. Since these works have not considered failures in the feedback mechanism itself, they could not tackle the problem of adaptation robustness against the same impairments by which the voice flow is affected. The challenge here is to guarantee, with some confidence level, that adaptive mechanisms will be resilient enough to face disturbances in the network.

- *Compositional adaptation.* As mentioned in Section B.5, software adaptation can be implemented by two general approaches: *parameter* adaptation and *compositional* adaptation. The surveyed works in Chapter C perform only parameter adaptation, as well as the three adaptation strategies implemented in this thesis. One further step is to design and develop compositional strategies for QoS control of VoIP, by which adaptive mechanisms can adopt new strategies beyond those endowed during design. Therefore, compositional adaptation requires research advances in knowledge management and interaction among agents in charge of the MAPE activities, as we shall see in sections H.2.3 and H.2.4.

H.2.2 Extension of VoIP Adaptation Scope

This thesis has concentrated on self-adaptive mechanisms targeted for QoS control of the RTP flow in VoIP systems. Here, we present the challenges in integrating QoS management with other VoIP aspects, such as security, signaling and calling experience.

- *Security.* QoS control assumes that impairments in VoIP have no malicious nature. However, a huge packet loss may be a result of a security attack, and so reducing bitrate or increasing redundancy will not improve speech quality. Thus, an integrate approach between QoS control and security poses as a challenge in this matter.

- *Trust.* An adaptive VoIP system can be regarded as a collection of agents that interact with one another in an open distributed environment. Consequently, agents are likely to be faced with a number of possible *interaction* partners with varying properties, as for example malicious intentions, incompetence to overcome some QoS problem, high/low availability. Thus, to minimize the uncertainty associated with interactions among networked peers, agents should be designed to assign a trust judgment to their counterparts and to reason about this.

  Trust is a general research topic in computer science, and should not be regarded as a security concern only. As claimed by Ramchurn et al. [162], "if an agent has performed poorly because of changes in its environment, it should not be taken to be dishonest or a liar." Thus, the MAPE agents should ponder whether the adaptation commands that they receive are plausible to be applied or will not lead to resource starvation or network overload, bounded by real-time constraints.

- *Signaling.* VoIP calls are established by means of signaling protocols, which support supplementary services, such as NAT traversal, number identification, remote call pickup, music and recording on hold or transfer, blacklist, and multiconference support. The challenge here is not only to endow such signaling services with adaptive behavior individually, but also to guarantee that the composition among them will not degrade the performance of the overall system.

- *Call experience.* As mentioned in Section B.3, a VoIP call begins when the user picks up the phone and ends when one of the parties disconnects. Thus, current VoIP adaptation mechanisms should account for other aspects that make up a satisfactory call. The challenge here is how to model and quantify those aspects, and how they can be interrelated algorithmically to reflect user's expectations.

H.2.3 Knowledge Management

Central to the four activities of the feedback loop (Figure B.7), there is a *knowledge-base*, whose role is to maintain information about the managed entities and their operations. As seen throughout this thesis, so far research has addressed adaptive properties in VoIP systems, but not much attention has been paid to knowledge management in this context. In general sense, knowledge management refers to the activities of collecting, storing, delivering, and reasoning about the knowledge of the networks to benefit network operations and management [23]. The challenges related to this topic, when applied to VoIP systems, are briefly described as follows.

- *Knowledge modeling and representation.* According to Hoek & Wooldridge [197], knowledge representation "focuses on how to *represent* and *reason* about environments with various different properties." However, most of software artifacts *implicitly* assume a certain model of their domain, with the aim of recording observations and solving problems. Thus, information corresponding to this model is scattered within the artifacts in a rather nontransparent manner. Consequently, when two agents are required to interact, this implicitness can cause misinterpretation of exchanged information, with undesirable consequences. A more scalable solution would be to define a shared model of reality interpretable by all agents. This concept corresponds to the term *ontology.*

  According to Wooldridge [203], "an ontology is a formal specification of a set of terms, intended to provide a common basis of understanding about some domain." Therefore, an ontology should be adopted to specify agents formally and to automate interpretation.

  However, there are a few ontology proposals in the literature to represent the domain of computer networks and, in particular, VoIP communications. Two examples are the work of Geneiatakis et al. [60], which emphasizes the representation of security concepts for SIP signaling, covering only a few QoS concepts; and the NetQoSOnt [156], which is oriented to network services and provides a formal, extendable and machine-processable specification to state QoS requirements.

  Two relevant challenges in this topic are (1) how to integrate QoS and security ontologies for VoIP aiming a holistic approach of both malicious and nonmalicious faults, and (2) how those ontologies can be reused for providing a self-adaptive capability to current VoIP systems across multiple equipment vendors.

- *Knowledge storage and dissemination.* Traditional network management systems often store information in centralized databases, which raises scalability and performance concerns. However, insofar as self-adaptive agents heavily rely on knowledge management infrastructure to acquire their operating context, and achieve their objectives, there is a need for scalable knowledge storage and delivery.

  Some key challenges in this topic include (1) how to aggregate monitored data over time and space efficiently? (2) How to determine the minimal amount of data that can be shared among endpoints without overloading the network, at the same time it permits a correct diagnosis of call quality? (3) How to determine which type of data and level of detail are valuable for VoIP endpoints or signaling proxies to have access? (4) How to protect users' and companies' privacy while gathering information about their calls for QoS-control only purposes, instead of spying or advertising?

- *Reasoning and cognition.* In data network context, cognition refers to the ability of the system to interpret its objectives, reasoning about its current state, and planning for future actions based on its current knowledge. This reasoning process needs a solid knowledge-base, containing information about the network, the VoIP service and adaptation actions that can be undertaken. These topics have been studied extensively in the AI area, but so far have little application in VoIP technology.

  Some challenges in this topic include (1) how to detect correlation among events, such as recurrent decreasing in MOS after some set of rules is applied? (2) How to detect that a sequence of adaptation actions is being taken cyclically, but no improvement in speech quality is observed? (3) How to determine that some event is occurring *frequently* or *sporadically*?

H.2.4 Specification of Formal Models

In order to assure that adaptive VoIP systems will behave properly to changes in the environment (network condition, users' preferences, service demand), we need techniques for designing and verifying properties of components and the system as a whole. During the design phase, which comprises specification and implementation tasks, the main interest is to describe and model both *individual* and *social* properties of the components of the system. Individual properties are those observed in the *micro* scale, that is, related to each single component such as softphones, proxies, and gateways. Social properties, in turn, are those observed in the *macro* scale, that is, emerged from the interactions among the components, such as cooperation between softphones, or negotiation between a softphone and a signaling proxy. During the verification phase, the interest is to determine whether the system under consideration possesses the properties specified during the design phase [8].

One way to represent and understand properties of agent systems is by means of *formal models.* Formal models use logical formalisms for capturing three basic features: (1) the *information* that the agent possesses of itself, the environment and other agents (knowledge and beliefs); (2) the *mental states* of individual agents (goals and desires); and (3) the possible *interactions* in the system. Formal models are needed not only to assure the correctness of programs, but also to allow computer systems to reason about its internal states and environment, in a really *self*-adaptive way.

Here, we describe some challenges associated to the design and verification of formal models for adaptive VoIP systems. The challenges were divided into two major groups: those related to the *properties* of the adaptive system, and those related to the *scale* of observation in which those properties are situated.

Basic System Properties

Software and hardware systems in general are expected to exhibit some basic properties, such as safety, liveness, and fairness. Particularly, adaptive VoIP systems should be designed and verified according to those properties, which are briefly presented in the following.

- *Functional correctness.* A system is considered to be *correct* whenever it satisfies all properties obtained from its specification [8]. Therefore, the main concern during specification is to set an appropriate list of desired properties. For an adaptive VoIP system, two essential properties are (1) that it should never tear down an ongoing call while adjusting softphones' parameters, and (2) that the outcome of an adaptation action should always increase speech quality.

- *Reachability.* This property states that some particular state *can be reached* [12]. In the VoIP context, some examples are the following: "we can obtain $Ppl < 1\%$," "we cannot have MOS < 3.6," or "we can change the packetization rate without reducing MOS."

  One classic example is to check the existence deadlock and livelock states. We say that a system is in *deadlock* when no progress is possible, that is, no further event can be executed. Since a VoIP adaptation mechanism must always be running the feedback loop, it should not reach a deadlock state before the end of the call.

  On the other hand, we say that a system is in *livelock* when there is a set of nonterminal states in the system that are reachable from one another, but with no transition going out of this set. For example, a VoIP adaptation mechanism may indefinitely and cyclically apply some sequence of parameter adjustments without substantially improving speech quality. In this case, it should try another adaptation strategy, or stop changing parameters.

- *Safety.* This property expresses that, under certain conditions, an event *never occurs* [12]. In other words, it states that something *bad* will not happen during a system execution [110].

  A typical safety violation is when two or more agents have nonmutually exclusive read and write access to a shared resource. In IP telephony, this problem is likely to arise in SIP proxy servers, for example. Usually, software implementations of such entities consist of a *core* component, which implements the basic call functions, and *module* components, which add some functionality and run on the top of the core component. If the media module generates a reINVITE SIP request for switching the current codec, but another modules imposes some restriction or delay to such requests, undesired side effects arise from this interaction. Therefore, formal techniques are needed to guarantee safety properties.

- *Liveness.* This property expresses that, under certain conditions, some event *will ultimately occur* [12]. Liveness properties require some progress in a system execution, stating that something *good* will happen in the future [110]. Mere reachability is not the issue: liveness properties state that some event will happen regardless of the system behavior [12].

  For QoS control of real-time media, the guarantee that something good will happen *in the future* is not enough. If a voice packet arrives at the receiver after its successor has been played out, then speech quality will be affected, and the VoIP system will not fulfill its design objectives. So, to specify and verify adaptive VoIP systems, liveness properties should be bounded by real-time properties.

  There are a few approaches that tackle safety and liveness guarantee in VoIP, such as [31] and [36]. However, they concentrate on SIP signaling only and do not use formal models for proving correctness, so that their results cannot be extended for agent reasoning. Anyway, these works have the merit of serving as good starting points to tackle this challenge.

- *Fairness.* This property expresses that, under certain conditions, an event will occur (or will fail to occur) *infinitely often* [12].

  In computer networks, fairness deals with the *distribution* of network *resources* among applications. Consequently, fairness is achieved when network resources are distributed in such a way that satisfies the justified expectations of agents that participate in the system [202]. Some network protocols consider fairness explicitly. For example, TCP considers the distribution of throughput, and some MAC protocols consider the distribution of link bandwidth.

  As pointed by Wierzbicki [202], the problem of achieving fairness in a packet switched environment with varying loads from many sources is understood to be hard, because most of the proposed solutions cannot use centralized control. Particularly, some VoIP architectures are centralized by definition, such as those maintained by public operators or enterprises, in which the RTP flow can pass through a signaling *proxy* or a media *gateway.* Thus, these nodes can be endowed with fair algorithms that allocate bandwidth and configure codec, packetization, FEC and other QoS parameters.

  In contrast, VoIP calls set up among ad hoc softphones, without intermediation of a central node, are more prone to have problems in distributing resources. The softphone's interest is to use a codec with the best quality, but this behavior can prevent other softphones from experiencing the same quality, because of network-resource limitations. For instance, it would be unfair to have, at the same local network and under the same QoS fee, one VoIP call using the best codec available and consuming a substantial share of the available bandwidth, and ten others VoIP flows using the worst codec and consuming a small portion of the available bandwidth.

- *Timeliness.* According to Le Lann [111], a real-time system is "a computing system where initiation and termination of activities must meet specified *timing constraints.*" Thus, correctness in time-critical systems depends not only on the logical result of the computation, but also on the time at which the results are produced [185]. A system designed in accordance with this definition is said to be entrusted with *timeliness properties* [111].

  Users of a VoIP system can tolerate some amount of delay, and even loss caused by too late events. In this case, real-time properties need to be satisfied only in the average case or to a certain percentage, because a late answer is still a valid answer. However, the mathematical description of such properties is more complex, because it requires some kind of logic that would take probability into account. Additionally, there is the dilemma of building a predictable system situated in an environment that exhibits a no-fully predictable behavior, such as data networks. Therefore, the assurance of reachability, safety, liveness and fairness properties becomes more challenging.

Observation Scale

The basic properties mentioned in the preceding section can be observed in different scales of the system. For example, at the component scale, a fairness property of a softphone may assure that all its MAPE agents should have fair access to memory space. On the other hand, at the system scale, when multiple softphones try to establish calls by means of a proxy, this latter should provide fair resource sharing to all softphones attached to it, given that all softphones have same privilege classes (e.g., premium, basic). Here, we present some challenges related to the observation scale of the adaptive VoIP system.

- *Individual scale.* Components are the primitive elements from which the system is built up. Dividing the problem into smaller chunks helps in tackling complexity because it limits the designer's scope. Thus, these components have to be conceived in terms of *goals* they have to achieve and *actions* that they can perform to pursuit these goals. The main concern is not in the interaction among components, but in guaranteeing that each component exhibits the desired properties.

- *Social scale.* A system can be regarded as a collection of agents that are arranged in various relationships to one another. Assuming that each component will attain to its design goals, the main concern here is to model the relationships that arise from the interaction among them, such as cooperation, negotiation, competition, and trust.

  Since multiagent architectures allow problems to be solved in a distributed fashion, they can be used for overcoming intermittent and unreliable network connections. Skomeršić & Parat [180] have shown how to apply the multiagent paradigm to control a group of Asterisk IP PBX, at the signaling level. As future work – not envisioned by the authors – this approach could be extended for QoS control of the calls managed by such a system.

  Whilst traditional temporal logic can be efficiently used for designing and verifying properties of single components, they can be inappropriate for systems composed of complex arrangements of agents and exposed to such an indeterminism as found in computer networks [203]. The challenge here – not limited to VoIP systems only – is how to couple different logic representations from different levels (individual, social, knowledge) to achieve the system's goals.

Appendix I

Recorded Sentences

Tables I.1 to I.4 below lists the sentences uttered by the four speakers (two females and two males) involved in the DCR experiments described in Chapter D. These sentences are written in Portuguese.

Table I.1: Sentences recorded by female speaker 1.

Number	Sentence
1	Governo pretende concluir até maio acordo para banda larga no país. A meta é disponibilizar internet rápida por preços populares.
2	Setor gastronômico criou seiscentos empregos em Manaus. O desempenho é superior ao apresentado no ano passado.
3	O banco abre às nove horas, no entanto chegarei às oito.
4	As crianças estavam muito felizes, já que o Natal havia chegado.
5	Todos saíram da reunião preocupados após o anúncio de cortes nos gastos da empresa.
6	A torcida permaneceu otimista: Faltava um gol para o time ser campeão.
7	Ele estudava com grande interesse. Os resultados eram vistos em suas notas.
8	Tubarões foram levados pelas chuvas para terra firme. Os animais foram encontrados após inundações na Austrália.
9	Os jornais divulgam cenas dos estragos na região serrana do Rio. Analistas apontam tragédia como primeiro desafio do governo Dilma.
10	Novo satélite permite medir energia do sol com mais precisão, pois captura mais dados sobre irradiação solar.
11	Dificuldade de acesso às notas do Enem irrita candidatos. Alguns estudantes levaram horas para acessar o site.
12	Os torcedores saíram felizes do estádio, pois há dez anos seu time não era campeão.
13	Dois países não assinaram as medidas. Brasil e Turquia eram contra as sanções econômicas.
14	Novas obras foram aprovadas pelo governo. A construção da estrada trará melhorias ao interior.
15	O gás acabou lá em casa, mas o distribuidor estava aberto.
16	A ciência não pode prever futuros acontecimentos com exatidão, mas pode calcular a probabilidade do fato ocorrer.
17	Teu coração sabe compreender quando preciso de uma amiga. Teus olhos sensíveis se endurecem quando preciso de uma lição.
18	Ponte desaba e interrompe trânsito em rodovia de Goiás. Segundo o Corpo de Bombeiros, não houve vítimas.

Continue on next page

Number	Sentence
19	Hoje, a metade dos brasileiros vive sob sensação de insegurança. O problema está entre os principais desafios do próximo governo.
20	O presente não era o que ele queria ganhar, porém era notório seu sorriso de felicidade.
21	Deixar as roupas no varal foi um erro. A chuva as molhou enquanto estávamos fora.
22	O esforço é o primeiro passo para o progresso: as vitórias surgem com o passar do tempo.
23	Os jogadores ganharam confiança após venceram a primeira partida.
24	O navio chegou a Fortaleza com atraso, mas todos ficaram contentes em pisar em terra firme.
25	Carlos quer construir sua casa. Ele deseja deixar de pagar aluguel.
26	O operário se feriu durante o trabalho, mas a equipe médica prestou os primeiros socorros.
27	O Amazonas foi o maior produtor de gás natural em janeiro. Houve uma alta de seis por cento em relação ao mês anterior.
28	Foi feita uma limpeza na praia do Tupé. Os moradores contribuíram com a ação.
29	Eduardo passou no vestibular, por isso sua mãe estava muito feliz.
30	Os turistas acordaram cedo para ir à praia, e foram surpreendidos com calor daquela cidade.
31	O casal abriu a porta ansioso, a fim de conhecer o novo lar.
32	O trânsito estava intenso, por isso estacionei o carro e esperei.
33	O filme que eu assisti agradou o público. Todos queriam ver novamente.
34	As flores são umedecidas pelo orvalho. O vento espalha o pólen pelo chão.
35	O casal se mudou para Curitiba. Ambos conseguiram bons empregos por lá.
36	Olga vai casar no mês que vem, no entanto ainda não distribuiu os convites.
37	É perigoso tomar banho neste rio, porque existem piranhas na água.
38	A loja fechava mais tarde do que pensei. Não precisávamos ter saído tão cedo.
39	Caso você não estude, ficará muito ansioso para a prova.
40	Espero encontrar o caminho de volta, pois o tempo está passando e estou atrasado.
41	O povo era contra a instalação da hidrelétrica, mas um acordo foi aceito pelos moradores.
42	Tereza tinha medo de raios e trovões. Durante tempestades ela se trancava em seu quarto.
43	Tens muito talento para música, Por isso deve continuar com seus sonhos.
44	Estávamos ansiosos pelo resultado. Quando a aprovação saiu, foi só alegria.
45	A boiada foi contida pelos vaqueiros. Agora, o rebanho seguia pelo caminho correto.
46	Os jovens sonham com uma profissão melhor, mas o estudo é a única forma de alcançar esse objetivo.
47	Embora estivesse confiante na vitória, a derrota humilhante foi inevitável.
48	O dia amanheceu chuvoso, fazendo com que Rômulo ficasse em casa.

Continue on next page

Number	Sentence
49	Os terrenos deixam lugar para os edifícios, e os apartamentos vão substituindo as casas.
50	Embora a prova estivesse fácil, demorei bastante para terminar.
51	É interessante que você compareça, caso contrário, ela ficará muito triste.
52	O fiscal verificou todos os documentos, mas tudo estava em ordem.
53	A expectativa é de que a safra aumente, já que o clima tem ajudado a colheita.
54	O porteiro impediu a entrada de Alfredo, mas um telefonema resolveu o problema.
55	É comum aparecer tubarões nesta praia, Não é recomendado que você tome banho aqui.
56	No portão da casa estava escrito: Nosso cão não é seu amigo.
57	Precisa-se de estradas bem construídas. pois as rodovias atuais não suportam a demanda.
58	O vulcão parecia extinto, mas ele voltou a dar sinais de atividade.
59	A mesa não era muito grande, mas era perfeita para apoiar o som.
60	O lago era grande e limpo. Costumávamos brincar em suas margens.

Table I.2: Sentences recorded by female speaker 2.

Number	Sentence
61	Tudo o que ele queria era estar perto de sua família. Ele viajou para estudar no exterior há dois anos.
62	As obras ainda não foram definidas, mas os custos já passam de cem mil.
63	Feira de tecnologia expõe carros inteligentes. Os novos automóveis estacionam sozinhos.
64	Ela tem vergonha de si mesma, sem saber o quanto é linda.
65	Remoção de navio com ácido sulfúrico deve levar dias. Autoridades afirmam que a carga não vazou para o rio.
66	Quanto mais você escutar a sua voz interior, melhor você ouvirá o que está tocando do lado de fora.
67	A semana começa com pancadas de chuva e baixa temperatura. O centro meteorológico confirmou a possibilidade de chover granizo.
68	Rede hoteleira comemora recorde de ocupação. Opções de férias na cidade atraíram os turistas.
69	Quinhentos voos estavam previstos para esta manhã. Duzentos e noventa sofreram atrasos e cancelamentos.
70	O número de afetados pelo clima no Brasil triplicou. O aumento de inundações é o item mais impressionante.
71	Ando à procura de espaço, por isso arrumei o quarto.
72	A ideia do presente é simulada para a criação do futuro. Tudo aquilo que hoje é realidade, um dia foi ficção nos filmes.
73	Prejudiquei você naquele momento, agora eu quero recompensá-lo.
74	Os jornais nada publicaram, mas o fato repercutiu em todo o bairro.
75	A tentativa não deu certo, mas ganhamos experiência.
76	A comida está sem sal, mas fora isso está boa.

Continue on next page

Number	Sentence
77	Mamãe ficou muito apavorada, quando o ladrão entrou em casa.
78	Eu até esqueceria o passado, se você não ficasse me lembrando.
79	Paulo provocou a briga, mas saiu antes que se machucasse.
80	Estava tarde para sair, mesmo assim Joana foi sozinha.
81	O cego se deixa levar pelo guia. O cão treinado são seus olhos.
82	O IBAMA fiscalizou extração ilegal de madeira, Mesmo assim havia novas áreas sendo desmatadas.
83	Os operários trabalham dia e noite, mesmo assim a obra só ficará pronta em dois anos.
84	Leia este livro aqui, por favor, pois ele tem as respostas que você procura.
85	Os livros custavam dez reais cada um. Contudo, a coleção completa era mais em conta.
86	É preciso gastar menos água, pois a conta do mês passado veio alta.
87	Só vou ao Teatro Amazonas, se não chover de tarde.
88	Fui a Copacabana para assistir a festa, e os fogos transformaram o céu do Rio de Janeiro.
89	Desde que chegamos aqui, não cessaram as visitas.
90	Pedi que saíssem da sala, mas todos estavam ocupados.
91	Preciso estudar todas as provas, já que será todo assunto na avaliação final.
92	A prova foi comentada pelo professor, e está disponível na secretaria.
93	O ônibus chegará a qualquer momento, por isso fiquem atentos para não perdê-lo.
94	O documento está dentro da escrivaninha, mas eu não tenho a chave para abri-la.
95	Conversamos sobre você ontem, mas ela não quer lhe conhecer.
96	Muitas festas estão acontecendo naquela cidade. O povo parece mais contente.
97	Foi impecável a performance do piloto. Sua vitória consequência natural.
98	Amanhã começam as aulas nas escolas, mas algumas salas continuam sem cadeiras.
99	O bando sobrevoou a cidade, mas não foi possível pousar.
100	A multidão de torcedores invadiu o campo, pois o time havia sido campeão.
101	Algumas folhas estavam em branco, pois o escritor não havia terminado sua obra.
102	Não tenho nenhum livro em papel, pois estudo somente pelo computador.
103	Um milhão de pessoas assistiram ao show, apesar do ingresso estar um pouco caro.
104	Mais de uma criança se machucou no brinquedo, por isso os peritos interditaram o parque.
105	O hotel e a cidade são maravilhosos, embora sejam muito distantes do aeroporto.
106	A empresa não pretendia aumentar os salários, mas se comprometeu em oferecer benefícios.

Continue on next page

Number	Sentence
107	Quanto mais o povo sabe, mais quer saber sobre a história.
108	A conferência está marcada para as quinze horas, mas deve-se chegar vinte minutos antes.
109	A rua que eu moro é arborizada, o que torna o ambiente mais agradável.
110	Escrevendo sem prestar atenção, você não vai aprender nada.
111	Fabiana continua eufórica, já que ganhou a promoção.
112	Os candidatos permanecem na sala de provas, mesmo já tendo terminado o exame.
113	Ela falava tão alto no auditório, que quase ficou rouca.
114	Meus primos partiram esta manhã e voltarão na semana que vem.
115	Após analisar todos os relatórios, foram comprovadas as irregularidades.
116	Assistimos a todos os jogos deste campeonato, até os dos times que já estavam eliminados.
117	Sem que ele percebesse, Renato chegou perto das abelhas.
118	Ao acabar a prova, iremos finalmente para casa.
119	Durante a noite faltou luz em casa, mas havia algumas velas na gaveta.
120	A lei entrou em vigor ontem, mas poucos ficaram sabendo.

Table I.3: Sentences recorded by male 1.

Number	Sentence
121	Os motoristas que atravessam de Manaus para Manacapuru enfrentam fila. Um acidente ocorreu com uma das balsas que fazem a travessia.
122	A Argentina é o terceiro maior exportador de soja do planeta. Mas este ano sua produção será menor devido ao clima seco.
123	A situação parecia perigosa, por isso resolvi esperar por uma ajuda.
124	Quinze voluntários irão atuar como guias turísticos nas balsas do porto. O grupo irá mostrar aos passageiros informações sobre a geografia amazônica.
125	O engenheiro elaborou o projeto e os pedreiros que executaram a obra.
126	Polo Industrial fecha o ano de dois mil e dez com faturamento recorde. Esse é o melhor desempenho das empresas da Zona Franca de Manaus.
127	O sinal digital está disponível para noventa milhões de pessoas. A digitalização completa está prevista para o ano de dois mil e dezesseis.
128	Ministério da Saúde aponta queda em casos de malária no Amazonas. A redução foi de aproximadamente trinta por cento em dois mil e dez.
129	A capital amazonense tem sofrido intensas variações de temperatura, causando aumento pela procura por atendimento médico para crianças.
130	Temos confiança em você, mas você não vai a esse passeio.
131	Tropa de Elite dois' está na seleção de Panorama do Festival de Berlim. A primeira versão do filme ganhou o Urso de Ouro em dois mil e nove.
132	Acredita-se que aqueles parlamentares estão envolvidos no esquema. A imprensa procura confirmar as supostas ilegalidades.
133	Decreto estabelece novas regras para uso de reservas e parques. Regulamento favorecerá crescimento do ecoturismo em todo o Estado.
134	Especialistas afirmam que novo governo não mudará situação do Haiti. Segundo turno das eleições no país deverá ser disputado em fevereiro.

Continue on next page

Number	Sentence
135	Já passam das quatro horas, e Felipe ainda não voltou.
136	O sol clareou o horizonte, e assim pude ver as montanhas.
137	Exército distribui água aos afetados por seca no Nordeste. Cidades de Minas Gerais também são atendidas pela Operação Pipa.
138	As abelhas não só produzem mel, mas também polinizam as flores.
139	Grupo de setenta coalas perde habitat na Austrália após enchentes. Pesquisadores correm em socorro à espécie com a baixa das águas no local.
140	Uma iguana apareceu na Avenida causando espanto às pessoas. O bicho saiu de uma área verde que fica próxima à avenida.
141	O fardo era muito pesado, por isso usei o carrinho para levá-lo.
142	Custa corrigir os velhos hábitos. Assumi-los é o primeiro passo.
143	O garoto pulou o muro alto. Sua intenção era resgatar a bola.
144	A chuva havia parado de cair, mas as ruas continuavam alagadas.
145	O relatório não foi terminado no prazo, pois os dados estavam incompletos.
146	Como eu havia falado várias vezes antes, a prova não estava muito fácil.
147	A maratona era a mais importante do ano, por isso participaram atletas de varias nacionalidades.
148	O vento soprou tão forte durante a tempestade que placas e árvores foram derrubadas nas ruas.
149	A história do filme era triste. Lágrimas foram arrancadas da plateia.
150	O exército brasileiro estava alerta caso os narcotraficantes cruzassem a fronteira.
151	Apesar do bombeiro estivesse ferido, ele continuou carregando a moça.
152	Os jogadores tinham sido convocados, e novos nomes foram chamados para a seleção.
153	Fiquei com os cabelos sujos de areia, mas a brincadeira na praia foi divertida.
154	Como a porta estava entreaberta, foi inevitável não olhar por ela.
155	As meninas ficaram meio nervosas, mas conseguiram apresentar o trabalho.
156	As cordas eram bastante fortes, o que facilitou o trabalho de resgate.
157	O frio e a chuva prejudicaram o jogo, por isso a partida foi adiada pra semana que vem.
158	O trânsito foi interrompido na rua principal, o que causou um longo congestionamento.
159	As árvores foram queimadas durante o incêndio, o que causou o interdição do parque.
160	A sala tinha uma lousa pequena, mas as cadeiras eram confortáveis.
161	Foi rápido ir de voadeira a Parintins, e ainda pude conhecer outras cidades.
162	O carro estava com um dos faróis queimados. Mesmo assim, o guarda liberou o condutor.
163	Quanto mais você fumar, mais grave ficará sua doença.
164	Faltou energia no início do primeiro tempo, mas ele acompanhou o jogo pelo rádio de pilha.

Continue on next page

Number	Sentence
165	Todos os presentes já estão aqui, porém ainda falta comprar os cartões.
166	O homem parecia embriagado. E ainda ele dirigia em alta velocidade.
167	À proporção que o tempo passa, tudo vai voltando ao estado normal.
168	Só havia alguns alunos na sala. Já que a maioria estava nos jogos universitários.
169	Algumas lâmpadas estavam queimadas, mas o zelador as trocou a pedido da direção.
170	Sobraram muitos pães na bandeja, mas em compensação o café acabou rápido.
171	A tenista não atendeu o repórter em particular. Ela só falou durante a coletiva.
172	Enquanto as crianças brincavam no balanço, os pais as reparavam constantemente.
173	João esqueceu que era feriado, e quando ele chegou na escola não havia ninguém.
174	O professor ressaltou a importância do voto, e os alunos concordaram com a opinião.
175	Após investir em treinamento de pessoal, a produção da empresa elevou.
176	Ana respondeu todas as questões, embora não tivesse estudado muito.
177	Não me simpatizei com ela, mas as suas opiniões são interessantes.
178	Depois que avisei ao rapaz sobre o perigo, ele foi atrás do equipamento de segurança.
179	Embora estivesse doente, Joaquim trabalhava muito.
180	Ainda me lembro da casa em que morávamos, pois ela ficava muito longe do trabalho.

Table I.4: Sentences recorded by male speaker 2.

Number	Sentence
181	O menino correu muito atrás do ônibus, pois imaginou que o próximo iria demorar.
182	As chamas foram controladas pela brigada de incêndio do edifício. Segundo o corpo de bombeiros não houve feridos graves.
183	Turistas que visitaram o Amazonas no ano passado fizeram queixas. As principais reclamações foram os serviços de transporte e limpeza.
184	Houve um crescimento do fluxo de passageiros que vão para Manaus. Isto só foi possível com a ampliação dos voos diretos e diários.
185	Admiro o modo como Renato trabalha, pois ele desempenha muito bem sua atividade.
186	Duzentas mil pessoas sofrem com inundações na Austrália. Enchentes atingem área do tamanho da França e Alemanha juntas.
187	Como era sua primeira viagem, ela ficou nervosa no avião.
188	Já que você está em casa, cuide bem das crianças.
189	O ladrão arrombou a janela e limpou tudo o que havia na casa.
190	A criança correu para a rua quando viu a pipa no céu, por sorte o carro freou a tempo de evitar um grave acidente.
191	O pescador esperou a tarde toda, mas nenhum peixe mordeu a isca.
192	Aquele abraço era a sua forma de sentir a vida, era a maneira que ela havia escolhido se sentir protegida.

Continue on next page

Number	Sentence
193	O preço dos conversores da TV digital acumula queda expressiva. Atualmente eles custam próximo de duzentos reais.
194	Depois de treinar bastante, Joaquim conseguiu a classificação.
195	Sete bairros terão serviço de abastecimento de água interrompido. Uma manutenção no reservatório vai ser realizada.
196	O time conquistou a classificação com muito esforço. Os resultados vieram após a contratação do novo técnico.
197	Lei municipal dita normas ambientais para construção. Os projetos terão de criar meios de uso racional de energia e água.
198	O total de brasileiros que desembarcaram em Buenos Aires cresceu. A procura pela cidade aumentou com a valorização do real.
199	Você nunca sabe que resultados virão da sua ação, mas se você nada fizer não existirão resultados.
200	Governo do Rio de Janeiro testa asfalto ecológico em estradas. A ideia é ter uma produção de asfalto de qualidade a baixo custo.
201	Marcos chegou cansado da praia, por isso foi dormir mais cedo hoje.
202	O jogo será transmitido pela televisão, portanto ficarei em casa para assisti-lo.
203	A maioria dos vereadores votou contra o projeto. A proposta prejudicava os estudantes.
204	Os meninos quebraram a vidraça, mas o jogo na rua continuou assim mesmo.
205	Os cientistas elaboraram um novo medicamento, sua composição permite uma recuperação mais rápida.
206	Eu me lembrei de seu aniversário, mas esqueci de trazer o presente.
207	Houve um acidente na Avenida, mas a trânsito fluía bem.
208	O garçom encheu a taça de vinho. O cliente pediu que deixasse a garrafa.
209	Suelem foi ao baile de formatura. Seu vestido novo chamava a atenção.
210	O calouro ainda não sabia onde era sua sala, mas as placas indicavam o caminho.
211	A luz do dia sumia no horizonte, logo foi preciso ascender os faróis.
212	Mesmo estando muito atarefado, João foi ao passeio.
213	O farol iluminava a noite escura, sinalizando a chegada dos barcos.
214	O chão ficou coberto de flores, enquanto a noiva desfilava até o altar.
215	O suco estava com um gosto diferente, foi então que percebi que faltava açúcar.
216	Mesmo que vença todos os jogos, o time de João será rebaixado.
217	Encontrei o livro que procurava. Ele estava com um amigo meu.
218	Os peixes não se deixavam fisgar. Parece que a isca não esta muito boa.
219	Entreguei o dinheiro ao padeiro, E amanhã pegarei o bolo pronto.
220	Todos entraram calados na sala, mas durante a aula ninguém fazia silêncio.
221	O médico pedia paciência, já que os familiares estavam nervosos.
222	Eles brigavam todas as manhãs, mas todas as noites eles estavam juntos.

Continue on next page

Number	Sentence
223	Quando ele foi finalmente ao médico, a doença já tinha se espalhado.
224	Marcos pagou todas as contas, mesmo sabendo que ficaria sem dinheiro.
225	Aloísio me emprestou o carro, mesmo contra a sua vontade.
226	Dois reais bastam para mim, pois o almoço custa um e vinte.
227	Ela leu apenas a primeira página, porque o livro não era interessante.
228	Ganhar pouco não interessa, já que o importante é fazer o que gosta.
229	Este documento não vale mais, pois a data de vencimento já passou.
230	A pequena cidade possui apenas três ruas, por isso todos os moradores se conheciam.
231	As aulas agradaram muito os alunos: todos estão confiantes para o Enem.
232	A crise política instalou-se no país, por isso a economia teve grandes perdas.
233	Ele está pedindo muito dinheiro pela casa, embora sua localização não seja boa.
234	O jogo a que assisti estava horrível. Nenhum dos dois times era profissional.
235	O ataque não fez nenhum gol, pois os zagueiros adversários estavam alertas.
236	Elas ficaram meio aborrecidas, porque Marcos chegou muito atrasado.
237	A nova fábrica prosperou bastante, trazendo desenvolvimento pra cidade.
238	Joana ficou triste depois que Anderson partiu, mesmo sabendo que ele iria voltar.
239	Hoje o clima amanheceu chuvoso, e a temperatura estava muito baixa.
240	O menino queria saber as horas, pois ele estava atrasado para aula.

Bibliography

[1] Abreu-Sernandez, V., and Garcia-Mateo, C. Adaptive multi-rate speech coder for VoIP transmission. *IEE Electronics Letters Online 36*, 23 (Nov. 2000), 1978–1980.

[2] Alshakhsi, S. A. A., and Hasbullah, H. Improving QoS of VoWLAN via Cross-Layer-Based Adaptive Approach. In *International Conference on Information Science and Applications (ICISA)* (Jeju Island, South Korea, Apr. 2011), pp. 1–8.

[3] Andersson, J., de Lemos, R., Malek, S., and Weyns, D. Modeling Dimensions of Self-Adaptive Software Systems. In *Software Engineering for Self-Adaptive Systems*, B. Cheng, R. de Lemos, H. Giese, P. Inverardi, and J. Magee, Eds., vol. 5525 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2009, pp. 27–47.

[4] Åström, K. J., and Murray, R. M. *Feedback Systems: an Introduction for Scientists and Engineers*. Princeton University Press, Princeton, NJ, USA, 2008.

[5] Atzori, L., and Lobina, M. L. Playout Buffering in IP Telephony: a Survey Discussing Problems and Approaches. *IEEE Communications Surveys & Tutorials 8*, 3 (3rd Quarter 2006), 36–46.

[6] Atzori, L., and Lobina, M. L. Playout Buffering of Speech Packets Based on a Quality Maximization Approach. *IEEE Transactions on Multimedia 8*, 2 (Apr. 2006), 420–426.

[7] Bai, Y., and Ito, M. R. QoS control for video and audio communication in conventional and active networks: Approaches and comparison. *IEEE Communications Surveys Tutorials 6*, 1 (First Quarter 2004), 42–49.

[8] Baier, C., and Katoen, J.-P. *Principles of Model Checking*. The MIT Press, Cambridge, MA, USA, 2008.

[9] Barbedo, J. G. A. *Avaliação objetiva de qualidade de sinal de áudio e voz*. PhD thesis, Universidade Estadual de Campinas (UNICAMP), Campinas, Brazil, 2004. in Portuguese.

[10] Barberis, A., Casetti, C., Martin, J. C. D., and Meo, M. A simulation study of adaptive voice communications on IP networks. *Computer Communications 24*, 9 (May 2001), 757–767.

[11] Bellalta, B., Oliver, M., and Rincón, D. Performance of the GPRS RLC/MAC protocols with VoIP traffic. In *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications* (Sept. 2002), vol. 1, pp. 305–309.

[12] Bérard, B., Bidoit, M., Finkel, A., Laroussinie, F., Petit, A., Petrucci, L., Schnoebelen, P., and McKenzie, P. *Systems and Software Verification*. Springer, New York, NY, USA, 2001.

[13] Beritelli, F. B., Ruggeri, G., and Schembra, G. TCP-Friendly Transmission of Voice over IP. *European Transactions on Telecommunications 14*, 3 (2003), 193–203.

[14] Bichler, G., and Clarke, R. V. Eliminating Pay Phone Toll Fraud at the Port Authority Bus Terminal in Manhattan. *Preventing Mass Transit Crime* (1996), 93–115.

[15] Biernacki, A. VoIP Source Model based on the Hyperexponential Distribution. In *Proceedings of World Academy of Science, Engineering and Technology* (Feb. 2006), vol. 11, pp. 202–206.

[16] Bilbao, N., Fajardo, J. O., and Liberal, F. PQoS-driven VoIP service adaptation in UMTS networks. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques (Simutools'09)* (Rome, Italy, Mar. 2009), ICST, pp. 91:1–91:8.

[17] Birke, R., Mellia, M., Petracca, M., and Rossi, D. Experiences of VoIP traffic monitoring in a commercial ISP. *International Journal of Network Management 20*, 5 (September/October 2010), 339–359.

[18] Bodouhi, K., and Hadjinicolaou, M. G. Holding, Conversation and Setup Times Probability Distributions of Bulk Internet Telephony Traffic for VoIP/PSTN Gateways. In *Proceedings of the 19th International Teletraffic Congress (ITC'19)* (2005), pp. 461–472.

[19] Bohnert, T. M., and Monteiro, E. On Autonomic QoS Provisioning for Voice over IP Services. In *Proceedings of the 4th IEEE Consumer Communications and Networking Conference (CCNC'07)* (Jan. 2007), pp. 582–586.

[20] Bolot, J.-C., and Vega-García, A. Control mechanisms for packet audio in the internet. In *Proceedings of the 15th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'96)* (San Francisco, CA, USA., Mar. 1996), IEEE, pp. 232–239.

[21] Bonfiglio, D., Mellia, M., Meo, M., and Rossi, D. Detailed Analysis of Skype Traffic. *IEEE Transactions on Multimedia 11* (Jan. 2009), 117–127.

[22] Botta, A., Dainotti, A., and Pescapè, A. Multi-protocol and multi-platform traffic generation and measurement. In *Proceeding of the 26th Annual Joint Conference of the IEEE Computer and Communications (INFOCOM'07), DEMO Session* (Anchorage, USA, May 2007), IEEE.

[23] Boutaba, R., Xiao, J., and Zhang, Q. *Toward Autonomic Networks: Knowledge Management and Self-Stabilization.* Autonomic Computing and Networking. Springer, New York, NY, USA, 2009, pp. 239–260.

[24] Boutremans, C., and Boudec, J.-Y. L. Adaptive Joint Playout Buffer and FEC Adjustment for Internet Telephony. In *Proceeding of the 22nd Annual Joint Conference of the IEEE Computer and Communications (INFOCOM'03)* (San-Francisco, CA, USA, Mar. 2003), vol. 1, IEEE, pp. 652–662.

[25] Brady, P. T. A Statistical Analysis of On-Off Patterns in 16 Conversations. *Bell System Technical Journal* (Jan. 1968), 73–91.

[26] Brady, P. T. A Model for Generating On-Off Speech Patterns in Two-Way Conversation. *Bell System Technical Journal* (Sept. 1969), 2445–2472.

[27] Carvalho, L., Lima, A., Aguiar, R., Mota, E., and Souza, J. An E-model implementation for speech quality evaluation in VoIP systems. In *Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC'05)* (La Manga del

Mar Menor, Spain, June 2005), vol. 1, IEEE, pp. 933–938.

[28] Carvalho, L. S. G. An E-model Implementation for Objective Speech Quality Evaluation of VoIP Communication. Master's thesis, Computer Science Departament, Federal University of Amazonas (UFAM), Manaus, Brazil, Sept. 2004. in Portuguese.

[29] Carvalho, L. S. G., Valle, R., Passito, A., Mota, E., Mota, E., Novellino, R., and Penaranda, A. Synchronizing Web Browsing Data with Browserver. In *Proceedings of the 15th IEEE symposium on Computers and Communications (ISCC'10)* (Riccione, Italy, June 2010), IEEE, pp. 738–743.

[30] Casilari, E., Montes, H., and Sandoval, F. Modelling of Voice Traffic Over IP Networks. In *Proceedings of the 3rd Internacional Symposium on Communication Systems, Networks and Digital Signal Processing* (July 2002), pp. 411–414.

[31] Chan, K. Y., and v. Bochmann, G. *Methods for Designing SIP Services in SDL with Fewer Feature Interactions.* Feature Interactions in Telecommunications and Software Systems VII. IOS Press, Amsterdam, The Netherlands, 2003, pp. 59–76.

[32] Chen, J.-J., Lee, L., and Tseng, Y.-C. Integrating SIP and IEEE 802.11e to support handoff and multi-grade QoS for VoIP-over-WLAN applications. *Comput. Netw. 55* (June 2011), 1719–1734.

[33] Chen, W.-E., Hung, H.-N., and Lin, Y.-B. Modeling VoIP Call Holding Times for Telecommunications. *IEEE Network 21*, 6 (2007), 22–28.

[34] Chen, X., Wang, C., Xuan, D., Li, Z., Min, Y., and Zhao, W. Survey on QoS Management of VoIP. In *Proceedings of the 2nd International Conference on Computer Networks and Mobile Computing (ICCNMC'03)* (Washington, DC, USA, Oct. 2003), IEEE Computer Society, pp. 69–77.

[35] Cheng, B. H. C., de Lemos, R., Giese, H., Inverardi, P., Magee, J., Andersson, J., Becker, B., Bencomo, N., Brun, Y., Cukic, B., Serugendo, G. D. M., Dustdar, S., Finkelstein, A., Gacek, C., Geihs, K., Grassi, V., Karsai, G., Kienle, H. M., Kramer, J., Litoiu, M., Malek, S., Mirandola, R., Müller, H. A., Park, S., Shaw, M., Tichy, M., Tivoli, M., Weyns, D., and Whittle, J. Software engineering for self-adaptive systems: A research roadmap. In *Software Engineering for Self-Adaptive Systems* (2009), B. H. C. Cheng, R. de Lemos, H. Giese, P. Inverardi, and J. Magee, Eds., vol. 5525 of *Lecture Notes in Computer Science*, Springer, pp. 1–26.

[36] Cheung, E., and Zave, P. Generalized Third-Party Call Control in SIP Networks. In *Principles, Systems and Applications of IP Telecommunications. Services and Security for Next Generation Networks*, H. Schulzrinne, R. State, and S. Niccolini, Eds. Springer, New York, NY, USA, 2008, pp. 45–68.

[37] Choi, Y., Silvester, J. A., and hul Kim, H. Analyzing and Modeling Workload Characteristics in a Multiservice IP Network. *IEEE Internet Computing 15*, 2 (March/April 2011), 35–42.

[38] Ciullo, D., Mellia, M., and Meo, M. Traditional IP measurements: What changes in a today multimedia IP network. In *Proceedings of the 4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks (IT-NEWS'08)* (Feb. 2008), pp. 262–267.

[39] Claffy, K., Miller, G., and Thompson, K. The nature of the beast: recent traffic measurements from an Internet backbone. In *Proceedings of INET Internet Global*

Summit (Geneva, Switzerland, July 1998).

[40] Clark, A. D. Modeling the Effects of Burst Packet Loss and Recency on Subjective Voice Quality. In *Proceedings of the 2nd IP-Telephony Workshop* (Columbia University, NY, USA, Apr. 2001), pp. 123–127.

[41] Costa, N., and Nunes, M. S. Dynamic adaptation of quality of service for VoIP communications. *International Journal On Advances in Networks and Services 2*, 2-3 (2009), 155–166.

[42] Cover, T. M., and Thomas, J. A. *Elements of Information Theory.* Wiley-Interscience, New York, NY, USA, July 2006.

[43] Crawley, E. S., Rajagopalan, B., Nair, R., and Sandick, H. A Framework for QoS-based Routing in the Internet. Request for Comments (RFC) 2386, Aug. 1998.

[44] Crawley, M. J. *Statistics – An Introduction using R.* John Wiley & Sons, Inc., New York, NY, USA, 2005.

[45] Dang, T. D., Sonkoly, B., and Molnár, S. Fractal Analysis and Modeling of VoIP Traffic. In *Proceedings of NETWORKS 2004* (2004), pp. 217–222.

[46] de Lemos, R., Müller, H. A., Giese, H., Shaw, M., Andersson, J., Baresi, L., Becker, B., Bencomo, N., Brun, Y., Cukic, B., Desmarais, R., Dustdar, S., Geihs, K., Engels, G., Goeschka, K. M., Gorla, A., Grassi, V., Inverardi, P., Karsai, G., Kramer, J., Litoiu, M., Lopes, A., Magee, J., Malek, S., Mankovskii, S., Mirandola, R., Mylopoulos, J., Nierstrasz, O., Pezzè, M., Prehofer, C., Schï>½fer, W., Schlichting, R., Schmerl, B., Smith, D. B., Sousa, J. P., Tamura, G., Tahvildari, L., Villegas, N. M., Vogel, T., Weyns, D., Wong, K., and Wuttke, J. Software engineering for self-adaptive systems: A second research roadmap. In *Software Engineering for Self-Adaptive Systems* (Dagstuhl, Germany, May 2011), R. de Lemos, H. Giese, H. Müller, and M. Shaw, Eds., no. 10431 in Dagstuhl Seminar Proceedings, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany. Draft Version of May 20, 2011.

[47] Deng, S. Traffic Characteristics of Packet Voice. In *Proceedings of the 30th IEEE International Conference on Communications (ICC'95)* (Seattle, WA, USA, June 1995), vol. 3, pp. 1369–1374.

[48] Ding, L., Radwan, A., El-Hennawey, M. S., and Goubran, R. A. Performance Study of Objective Voice Quality Measures in VoIP. In *Proceedings of the 12th IEEE Symposium on Computers and Communications (ISCC'07)* (Aveiro, Portugal, July 2007), IEEE, pp. 197–202.

[49] Dodge, Y. *The Concise Encyclopedia of Statistics.* Springer Reference. Springer, 2008.

[50] Equivalence Pty. Ltd. openam: H.323 answering machine, Aug. 2011. `http://packages.debian.org/stable/comm/openam`.

[51] Escobar, M. A., and Best, M. L. Convivo communicator: An interface-adaptive VoIP system for poor quality networks. *Journal of Information, Communication & Ethics in Society 1*, 3 (2003), 167–180.

[52] ETSI. Speech communication quality from mouth to ear for 3,1 kHz handset telephony across networks. Technical Report ETR 250, July 1996.

[53] ETSI. GSM 06.90: Digital cellular telecommunications system (Phase 2+). Adaptive Multi-Rate (AMR) speech transcoding, version 7.0.0, 1998.

[54] Everitt, B., and Hothorn, T. *An Introduction to Applied Multivariate Analysis with R.* Springer, 2011.

[55] Falk, T., and Chan, W. Hybrid Signal-and-Link-Parametric Speech Quality Measurement for VoIP Communications. *IEEE Transactions on Audio, Speech, and Language Processing 16*, 8 (Nov. 2008), 1579–1589.

[56] Friedman, T., Caceres, R., and Clark, A. D. RTP Control Protocol Extended Reports (RTCP XR). Request for Comments (RFC) 3611, Nov. 2003.

[57] Galiotos, P., Dagiuklas, T., and Arkadianos, D. QoS Management for an enhanced VoIP platform using R-factor and network load estimation functionality. In *Proceedings of the 5th IEEE International Conference on High Speed Networks and Multimedia Communications (HSNMC'02)* (Jeju Island, Korea, July 2002), IEEE, pp. 305–314.

[58] Gardner, M. T., Frost, V. S., and Petr, D. W. Using Optimization to Achieve Efficient Quality of Service in Voice over IP Networks. In *Proceedings of the 22nd IEEE International Performance, Computing, and Communications Conference (IPCCC'03)* (Phoenix, AZ, USA, Apr. 2003), IEEE, pp. 475–480.

[59] Gaspar, Z., and Gocza, I. Voice Quality Degradation Recognition Using the Call Lengths. In *Proceedings of the 12th International Conference on Optimization of Electrical and Electronic Equipment (OPTIM'10)* (Brasov, Romania, May 2010), IEEE, pp. 1034–1039.

[60] Geneiatakis, D., Lambrinoudakis, C., and Kambourakis, G. An ontology-based policy for deploying secure SIP-based VoIP services. *Computers & Security 27*, 7-8 (2008), 285–297.

[61] Gokhale, S. S., and Lu, J. *QoS Monitoring of Voice-over-IP Services.* VoIP Handbook: Applications, Technologies, Reliability, and Security. CRC Press, Boca Raton, FL, USA, 2009, ch. 6, pp. 101–119.

[62] Gomes, J. V. P., Inácio, P. R. M., Lakic, B., Freire, M. M., Silva, H. J. A., and Monteiro, P. P. Source Traffic Analysis. *ACM Transactions on Multimedia Computing, Communications and Applications 6* (Aug. 2010), 21:1–21:23.

[63] Gong, Q., and Kabal, P. Quality-based playout buffering with FEC for conversational VoIP. In *11th Annual Conference of the International Speech Communication Association (INTERSPEECH)* (Makuhari, Chiba, Japan, Sept. 2010), ISCA, pp. 2402–2405.

[64] Google Inc. WebRTC, Aug. 2011. `http://sites.google.com/site/webrtc/`.

[65] Grancharov, V., and Kleijn, W. B. *Speech Quality Assessment.* Springer Handbook of Speech Processing. Springer-Verlag Berlin Heidelberg, Berlin, Germany, 2008, ch. 5, pp. 83–99.

[66] Gros, L., and Chateau, N. Instantaneous and Overall Judgements for Time-Varying Speech Quality: Assessments and Relationships. *Acta Acustica united with Acustica 87*, 3 (2001), 367–377.

[67] Gruber, J. G. A Comparison of Measured and Calculated Speech Temporal Parameters Relevant to Speech Activity Detection. *IEEE Transactions on Communications 30*, 4 (Apr. 1982), 728–738.

[68] Guha, S., Daswani, N., and Jain, R. An experimental study of the Skype peer-

to-peer VoIP system. In *Proceedings of the International workshop on Peer-To-Peer Systems (IPTPS'06)* (Santa Barbara, CA, USA, Feb. 2006).

[69] Heegaard, P. E. Evolution of Traffic Patterns in Telecommunication Systems. In *Proceedings of the 2nd International Conference on Communications and Networking in China (CHINACOM'07)* (Aug. 2007), pp. 28–32.

[70] Hellerstein, J. L., Diao, Y., Parekh, S., and Tilbury, D. M. *Feedback Control of Computing Systems*. John Wiley & Sons, 2004.

[71] Herlein, G., Valin, J.-M., Heggestad, A. E., and Moizard, A. RTP Payload Format for the Speex Codec. Request for Comments (RFC) 5574, June 2009.

[72] Hoene, C. *Internet Telephony over Wireless Links*. PhD thesis, Technical University of Berlin, TKN, Dec. 2005.

[73] Hoene, C., Karl, H., and Wolisz, A. A Perceptual Quality Model Intended for Adaptive VoIP Applications. *International Journal of Communication Systems 19*, 3 (Apr. 2006), 299–316.

[74] Hoene, C., Marsh, I., Schäfer, G., and Wolisz, A. Error Propagation After Concealing a Lost Speech Frame. In *Proceedings of the 1st Multimedia Communications Workshop: State of the Art and Future Directions* (Istanbul, Turkey, June 2006).

[75] Homayounfar, K. Rate Adaptive Speech Coding for Universal Multimedia Access. *IEEE Signal Processing Magazine 20*, 2 (Mar. 2003), 30–39.

[76] Huang, C.-W., Sukittanon, S., Ritcey, J. A., Chindapol, A., and Hwang, J.-N. An Embedded Packet Train and Adaptive FEC Scheme for VoIP Over Wired/Wireless IP Networks. In *Proceeding of the 31st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'06)* (Toulouse, France, May 2006), vol. 5, IEEE, pp. 429–432.

[77] Huang, L.-Y., Chen, Y.-M., Chung, T.-Y., and Hsu, C.-H. Adaptive VoIP Service QoS Control based on Perceptual Speech Quality. In *Proceedings of the 9th International Conference on Advanced Communication Technology* (Feb. 2007), vol. 2, pp. 885–890.

[78] Huang, T.-Y., Chen, C.-M., Chen, K.-T., and Huang, P. Towards User-Centric Rate Adaptations for VoIP Traffic. In *Proceedings of the Conference of the Special Interest Group on Data Communication (SIGCOMM)* (Kyoto, Japan, Aug. 2007), ACM. Poster Session.

[79] Huang, T.-Y., Huang, P., Chen, K.-T., and Wang, P.-J. Could Skype Be More Satisfying? A QoE-Centric Study of the FEC Mechanism in an Internet-Scale VoIP System. *IEEE Network 24*, 2 (March/April 2010), 42–48.

[80] Huang, Y., Korhonen, J., and Wang, Y. Optimization of Source and Channel Coding for Voice Over IP. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'05)* (Amsterdam, Netherlands, July 2005), IEEE.

[81] Huebscher, M. C., and McCann, J. A. A survey of autonomic computing — degrees, models, and applications. *ACM Comput. Surv. 40*, 3 (2008), 1–28.

[82] Intel. Intel Integrated Performance Primitives (IPP), Aug. 2011. `http://www.intel.com/software/products/ipp/`.

[83] ITU-R. Methods for the Subjective Assessment of Small Impairments in Audio Systems Including Multichannel Sound Systems. ITU-R Recommendation BS.1116-1, Oct. 1997.

[84] ITU-T. Artificial conversational speech. ITU-T Recommendation P.59, Mar. 1993.

[85] ITU-T. Methods for Subjective Determination of Transmission Quality. ITU-T Recommendation P.800, Aug. 1996.

[86] ITU-T. ITU-T coded-speech database. ITU-T P-series Recommendations – Supplement 23, Feb. 1998.

[87] ITU-T. Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. ITU-T Recommendation P.862, Feb. 2001.

[88] ITU-T. One-way transmission time. ITU-T Recommendation G.114, May 2003.

[89] ITU-T. Wideband coding of speech at around 16 kbit/s using Adaptive Multi-Rate Wideband (AMR-WB). ITU-T Recommendation G.722.2, July 2003.

[90] ITU-T. Single ended method for objective speech quality assessment in narrow-band telephony applications. ITU-T Recommendation P.563, May 2004.

[91] ITU-T. Software tools for speech and audio coding standardization. ITU-T Recommendation G.191, Sept. 2005.

[92] ITU-T. An architectural framework for support of quality of service in packet networks. ITU-T Recommendation Y.1291, May 2006.

[93] ITU-T. Methods For Subjective Determination of Transmission Quality. ITU-T Recommendation P.800.1, July 2006.

[94] ITU-T. The E-Model, A Computational Model For Use in Transmission Planning. ITU-T Recommendation G.107, Apr. 2009.

[95] ITU-T. Perceptual objective listening quality assessment. ITU-T Recommendation P.863, Jan. 2011.

[96] Iversen, V. B. Analysis of real teletraffic processes based on computerized measurements. *Ericsson Technics 29*, 1 (1973), 13–64.

[97] Iversen, V. B. Teletraffic engineering handbook, June 2001.

[98] Jain, R. K. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling.* John Wiley & Sons, 1991.

[99] Jammeh, E., Mkwawa, I.-H., Khan, A., Goudarzi, M., Sun, L., and Ifeachor, E. Quality of experience (QoE) driven adaptation scheme for voice/video over IP. *Telecommunication Systems* (2010), 1–13.

[100] Jeszensky, P. J. E. *Sistemas Telefônicos.* Manole, 2004. in Portuguese.

[101] Ji, L., Yin, X., Shi, X., and Wang, Z. Conversational Model Based VoIP Traffic Generation. In *Proceedings of the 3rd International Conference on Networking and Services* (Washington, DC, USA, June 2007), IEEE Computer Society, pp. 14–19.

[102] Jiang, W., and Schulzrinne, H. Analysis of On-Off Patterns in VoIP and Their Effect on Voice Traffic Aggregation. In *Proceedings of the 9th International Conference on Computer Communications and Networks* (Oct. 2000), pp. 82–87.

[103] Jiang, W., and Schulzrinne, H. Comparison and Optimization of Packet Loss Repair Methods on VoIP Perceived Quality under Bursty Loss. In *Proceedings of the 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'02)* (Miami Beach, FL, USA, May 2002), ACM, pp. 73–81.

[104] Johansson, I., Frankkila, T., and Synnergren, P. Bandwidth efficient AMR operation for VoIP. In *Proceedings of IEEE Workshop on Speech Coding* (Tsukuba, Ibaraki, Japan, Oct. 2002), IEEE, pp. 150–152.

[105] Jung, Y., and Ibanez, A. A. Improving wireless VoIP quality by using adaptive packet coding. *Electronics Letters 46*, 6 (2010), 459–460.

[106] Kalama, M., Acar, G., Evans, B., and Isoard, A. VoIP over DVB-RCS satellite systems: Trial results and the impact of adaptive speech coding using cross-layer design. *Computer Networks 52*, 13 (Sept. 2008), 2461–2472.

[107] Karapantazis, S., and Pavlidou, F.-N. VoIP: A comprehensive survey on a promising technology. *Computer Networks 53*, 12 (Aug. 2009), 2050–2090.

[108] Kawata, T., and Yamada, H. Adaptive Multi-Rate VoIP for IEEE 802.11 Wireless Networks with Link Adaptation Function. In *Proceedings of the 49th IEEE Global Telecommunications Conference (GLOBECOM'06)* (San Francisco, CA, USA, Nov. 2006), IEEE, pp. 1–5.

[109] Ksentini, A. Enhancing VoWLAN Service Through Adaptive Voice Coder. In *Proceedings of the 14th IEEE Symposium on Computers and Communications (ISCC'09)* (Sousse, Tunisia, July 2009), IEEE, pp. 673–678.

[110] Lamport, L. Proving the Correctness of Multiprocess Programs. *IEEE Transactions on Software Engineering SE-3*, 2 (Mar. 1977), 125–143.

[111] Lann, G. L. Critical Issues for the Development of Distributed Real-Time Computing Systems. In *Proceedings of the 2nd IEEE Workshop on Future Trends of Distributed Computing Systems* (Cairo, Egypt, 1990), IEEE, pp. 96–105.

[112] Lee, T.-Y., and Pan, J.-Y. Improving R-score of Adaptive VoIP Codec in IEEE 802.16 networks. In *Proceedings of the 14th Asia-Pacific Conference on Communications (APCC'08)* (Akihabara, Tokyo, Japan, Oct. 2008), IEEE, pp. 1–5.

[113] Li, L., Xing, G., Han, Q., and Sun, L. Adaptive Voice Stream Multicast Over Low-Power Wireless Networks. In *Proceedings of the 31st Real-Time Systems Symposium (RTSS)* (San Diego, FL, USA, Dec. 2010), IEEE, pp. 292–301.

[114] Liang, Y. J., Färber, N., and Girod, B. Adaptive Playout Scheduling and Loss Concealment for Voice Communication Over IP Networks. *IEEE Transactions on Multimedia 5*, 4 (Dec. 2003), 532–543.

[115] Lilja, D. J. *Measuring Computer Performance: A Practitioner's Guide.* Cambridge University Press, 2000.

[116] Lima, A. F. M., Carvalho, L. S. G., Souza, J. N., and Mota, E. S. A framework for network quality monitoring in the VoIP environment. *International Journal of Network Management 17*, 4 (July/August 2007), 263–274.

[117] Lindblom, J. A Sinusoidal Voice Over Packet Coder Tailored for the Frame-Erasure Channel. *IEEE Transactions on Speech and Audio Processing 13*, 5 (Sept. 2005), 787–798.

[118] Lizhong, W., Muqing, W., Lulu, W., and Mojia, L. An Adaptive Forward Error Control Method for Voice Communication. In *Proceeding of the 2nd International Conference on Networking and Digital Society (ICNDS'10)* (Sidney, Australia, May 2010), vol. 2, IEEE, pp. 186–189.

[119] Lulu, W., Muqing, W., and Dapeng, W. Wireless VoIP adaptive source rate control

algorithm. In *Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom'09)* (Beijing, China, Sept. 2009), IEEE, pp. 4788–4791.

[120] Maindonald, J., and Braun, W. J. *Data Analysis and Graphics Using R – An Example Based Approach*, 3rd ed. Cambridge University Press, 2010.

[121] Malfait, L., Berger, J., and Kastner, M. P.563 — The ITU-T Standard for Single-Ended Speech Quality Assessment. *Audio, Speech, and Language Processing, IEEE Transactions on 14*, 6 (Nov. 2006), 1924–1934.

[122] Mandelbrot, B. How Long Is the Coast of Britain? Statistical Self-Similarity and Fractional Dimension. *Science 156*, 3775 (May 1967), 636–638.

[123] Manousos, M., Apostolacos, S., Grammatikakis, I., Mexis, D., Kagklis, D., and Sykas, E. Voice-Quality Monitoring and Control for VoIP. *IEEE Internet Computing 9*, 4 (July 2005), 35–42.

[124] Mason, R. L., Gunst, R. F., and Hess, J. L. *Statistical Design and Analysis of Experiments: With Applications to Engineering and Science*, 2nd ed. Wiley Series in Probability and Statistics. Wiley & Sons, 2003.

[125] Matta, J., Pépin, C., Lashkari, K., and Jain, R. A source and channel rate adaptation algorithm for AMR in VoIP using the E-model. In *Proceedings of the 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'03)* (Monterey, USA, June 2003), ACM, pp. 92–99.

[126] Mattos, C. I., Ribeiro, E. P., and Pedroso, C. M. A New Model for VoIP Traffic Generation. In *Proceedings of the 7th International Telecommunications Symposium (ITS'10)* (Manaus, Brazil, Sept. 2010).

[127] Mazurczyk, W., and Kotulski, Z. Adaptive VoIP with audio watermarking for improved call quality and security. *Journal of Information Assurance and Security (JIAS) 2*, 3 (Sept. 2007), 226–234.

[128] Mazzoni, D. Audacity: Free Audio Editor and Recorder, Aug. 2011. `http://audacity.sourceforge.net/`.

[129] McGovern, P., Murphy, S., and Murphy, L. Addressing the link adaptation problem for VoWLAN using codec adaptation. In *Proceedings of the 49th IEEE Global Telecommunications Conference (GLOBECOM'06)* (San Francisco, CA, USA, Nov. 2006), IEEE, pp. 1–6.

[130] McGovern, P., Murphy, S., and Murphy, L. Protection Against Link Adaptation for VoWLAN. In *Proceedings of the 15th IST Mobile and Wireless Communications Summit* (Myconos, Greece, June 2006).

[131] McKinley, P. K., Sadjadi, S. M., Kasten, E. P., and Cheng, B. H. C. Composing Adaptive Software. *Computer 37* (July 2004), 56–64.

[132] Meng, A. On Evaluating Self-Adaptive Software. In *Self-Adaptive Software*, P. Robertson, H. Shrobe, and R. Laddaga, Eds., vol. 1936 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2001, ch. 6, pp. 65–74.

[133] Mens, T., Buckley, J., Zenger, M., and Rashid, A. Towards a Taxonomy of Software Evolution. In *Proceedings of the 2nd International Workshop on Unanticipated Software Evolution* (Warsaw, Poland, Apr. 2003), ETAPS.

[134] Menth, M., Binzenhöfer, A., and Mühleck, S. Source Models for Speech Traffic

Revisited. *IEEE/ACM Transactions on Networking 17*, 4 (Aug. 2009), 1042–1051.

[135] Mkwawa, I.-H., Jammeh, E., Sun, L., and Ifeachor, E. C. Feedback-Free Early VoIP Quality Adaptation Scheme in Next Generation Networks. In *Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM'10)* (Miami, FL, USA, Dec. 2010), IEEE, pp. 1–5.

[136] Mohamed, S., Cervantes-Pérez, F., and Afifi, H. Integrating networks measurements and speech quality subjective scores for control purposes. In *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'01)* (Anchorage, AK, USA, Apr. 2001), vol. 2, IEEE, pp. 641–649.

[137] Möller, S., Wältermann, M., Lewcio, B., Kirschnick, N., and Vidales, P. Speech Quality While Roaming in Next Generation Networks. In *Proceedings of the 44th IEEE International Conference on Communications (ICC'09)* (Dresden, Germany, June 2009), IEEE, pp. 1–5.

[138] Mota, E., Mota, E., Carvalho, L., Nascimento, A., and Hoene, C. *Pursuing Credibility in Performance Evaluation of VoIP over Wireless Mesh Networks.* Wireless Mesh Networks. INTECH, Jan. 2011, pp. 185–208.

[139] Moura, N. T., Vianna, B. A., Albuquerque, C. V. N., Rebello, V. E. F., and Boeres, C. MOS-Based Rate Adaption for VoIP Sources. In *Proceedings of the 42nd IEEE International Conference on Communications (ICC'07)* (Glasgow, Scotland, UK, June 2007), IEEE, pp. 628–633.

[140] Muller, H., Kienle, H. M., and Stege, U. *Autonomic Computing Now You See It, Now You Don't.* Software Engineering. Springer, New York, NY, USA, 2009, pp. 32–54.

[141] Müller, H., Pezzè, M., and Shaw, M. Visibility of Control in Adaptive Systems. In *Proceedings of the 2nd International Workshop on Ultra-Large-Scale Software-Intensive Systems (ULSSIS'08)* (New York, NY, USA, 2008), ACM, pp. 23–26.

[142] Myakotnykh, E. S. *Adaptive Speech Quality in VoIP Communications.* PhD thesis, University of Pittsburgh, 2009.

[143] Myakotnykh, E. S., and Thompson, R. A. Adaptive rate voice over IP quality management algorithm. *International Journal on Advances in Telecommunications 2*, 2 (2009), 98–110.

[144] Nagireddi, S. *VoIP Voice and Fax Signal Processing.* John Wiley & Sons, Hoboken, New Jersey, Sept. 2008.

[145] Narbutt, M., Kelly, A., Murphy, L., and Perry, P. Adaptive VoIP Playout Scheduling: Assessing User Satisfaction. *IEEE Internet Computing 9* (July/August 2005), 18–24.

[146] Ng, S. L., Hoh, S., and Singh, D. Effectiveness of adaptive codec switching VoIP application over heterogeneous networks. In *Proceedings of the 2nd International Conference on Mobile Technology, Applications and Systems* (Guangzhou, China, Nov. 2005), IEEE, pp. 1–7.

[147] Ngamwongwattana, B. *Sync & Sense Enabled Adaptive Packetization VoIP.* PhD thesis, University of Pittsburgh, Apr. 2007.

[148] Ngamwongwattana, B. Effect of Packetization on VoIP Performance. In *Proceedings of the 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON'08)* (Krabi, Thai-

land, May 2008), vol. 1, IEEE, pp. 373–376.

[149] NIST/SEMATECH. NIST/SEMATECH e-Handbook of Statistical Methods, Aug. 2011. `http://www.itl.nist.gov/div898/handbook/`.

[150] Oliveira, K. K. C., Kelner, J., Sadok, D., and Nóbrega, O. Avaliação de um Algoritmo Adaptativo de Transmissão de Voz em Redes IP com QoS. In *Proceedings of the 21th Brazilian Symposium on Computer Networks (XXI SBRC)* (Natal, Brazil, May 2003), SBC, pp. 425–440. in Portuguese.

[151] Padhye, C., Christensen, K. J., and Moreno, W. A New Adaptive FEC Loss Control Algorithm for Voice Over IP Applications. In *Proceeding of the 19th IEEE International Performance, Computing, and Communications Conference (IPCCC'00)* (Phoenix, AZ, USA, Feb. 2000), IEEE, pp. 307–313.

[152] Perkins, C., Hodson, O., and Hardman, V. A Survey of Packet Loss Recovery Techniques for Streaming Audio. *IEEE Networks 12*, 5 (Sept. 1998), 40–48.

[153] Post Increment. Opal – Open Source Voice, Video and Fax, Aug. 2011. `http://www.opalvoip.org/`.

[154] Počta, P., and Terpák, T. Packet Loss and Coding Impact on Quality of Synthesized Speech Predicted by PESQ and P.563 Models. In *Proceedings of the 9th Measurement of Speech, Audio and Video Quality in Networks (MESAQIN)* (Prague, Republic Czech, June 2010).

[155] Prijono, B. PJSIP, Aug. 2011. `http://www.pjsip.org/`.

[156] Prudêncio, A. C., Sheibel, M. L., Willrich, R., Tazi, S., and Sancho, G. Application and Network QoS Mapping Using an Ontology-Based Approach. In *Proceedings of the 3rd International Conference on Communication Theory, Reliability, and Quality of Service* (June 2010), IEEE, pp. 214–219.

[157] Qian, Y., Hu, R. Q., and Chen, H.-H. A call Admission Control Framework for Voice over WLANs. *Wireless Communications, IEEE 13*, 1 (Feb. 2006), 44–50.

[158] Qiao, Z., Sun, L., Heilemann, N., and Ifeachor, E. C. A new method for VoIP Quality of Service control use combined adaptive sender rate and priority marking. In *Proceedings of the 39th IEEE International Conference on Communications (ICC'04)* (Paris, France, June 2004), IEEE, pp. 1473–1477.

[159] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

[160] Raake, A. *Speech quality of VoIP: Assessment and Prediction.* John Wiley & Sons, Chichester, UK, 2006.

[161] Rabassa, A., St-Hilaire, M., Lung, C.-H., Lambadaris, I., Goel, N., and Zaman, M. New speech traffic background simulation models for realistic VoIP network planning. In *Proceedings of the 13th International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)* (Ottawa, Canada, July 2010), IEEE, pp. 364–371.

[162] Ramchurn, S. D., Huynh, D., and Jennings, N. R. Trust in multi-agent systems. *The Knowledge Engineering Review 19*, 01 (2004), 1–25.

[163] Ramjee, R., Kurose, J., Towsley, D., and Schulzrinne, H. Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-area Networks. In *Proceedings*

of the 13th IEEE Networking for Global Communications (INFOCOM'94) (June 1994), vol. 2, pp. 680–688.

[164] Ríos, A., Alcober, J., Oller, A., and Sempere, V. Design and Implementation of a Measurement and Alert System of QoS parameters in SIP based Internet Telephony. In *Proceedings of the 2nd Conference on Next Generation Internet Design and Engineering (NGI'06)* (València, Spain, 2006), pp. 117–122.

[165] Rosenbluth, J. H. Testing the Quality of Connections Having Time Varying Impairments. Tech. rep., Committee Contribution T1A1.7/98-031, 1998.

[166] Roychoudhuri, L., and Al-Shaer, E. S. Adaptive Rate Control for Real-Time Packet Audio Based on Loss Prediction. In *Proceedings of the 47th IEEE Global Telecommunications Conference (GLOBECOM'04)* (Dallas, TX, USA, November/December 2004), vol. 2, IEEE, pp. 634–638.

[167] Roychoudhuri, L., and Al-Shaer, E. S. Autonomic QoS Optimization of Real-time Internet Audio using Loss Prediction and Stochastic Control. In *Proceedings of the 11th International IFIP/IEEE Network Operations and Management Symposium (NOMS'08)* (Salvador, Brazil, Apr. 2008), IEEE.

[168] Sabrina, F., and Valin, J.-M. Adaptive Rate Control for Aggregated VoIP Traffic. In *Proceedings of the 51st IEEE Global Communications Conference (GLOBECOM'08)* (New Orleans, USA, Nov. 2008), IEEE, pp. 1–6.

[169] Salehie, M., and Tahvildari, L. Self-Adaptive Software: Landscape and Research Challenges. *ACM Transactions on Autonomous and Adaptive Systems (TAAS) 4*, 2 (May 2009), 1–42.

[170] Sanneck, H. Concealment of Lost Speech Packets Using Adaptive Packetization. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems* (June/July 1998), IEEE, pp. 140–149.

[171] Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V. RTP: A Transport Protocol for Real-Time Applications. Request for Comments (RFC) 3050, July 2003.

[172] Seo, J. W., Woo, S. J., and Bae, K. S. A Study on the Application of an AMR Speech Codec to VoIP. In *Proceedings of the 26th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'01)* (Salt Lake City, UT, USA, May 2001), vol. 3, IEEE, pp. 1373–1376.

[173] Servetti, A., and Martin, J. C. D. Adaptive Interactive Speech Transmission Over 802.11 Wireless LANs. In *Proceedings of the 1st IEEE International Workshop on DSP in Mobile and Vehicular Systems* (Nagoya, Japan, Apr. 2003), IEEE.

[174] Sfairopoulou, A., Bellalta, B., Macián, C., and Oliver, M. A Comparative Survey of Adaptive Codec Solutions for VoIP over Multirate WLANs: A Capacity versus Quality Performance Trade-Off. *EURASIP Journal on Wireless Communications and Networking 2011* (2011), 1–13.

[175] Sfairopoulou, A., Macián, C., and Bellalta, B. Dynamic measurement-based codec selection for VoIP in multirate IEEE 802.11 WLANs. In *Proceedings of the 8th COST 290 Management Committee Meeting* (Malaga, Spain, Feb. 2007), pp. 1–21.

[176] Shannon, C. E. *The Mathematical Theory of Communication.* University of Illinois Press, Urbana, IL, USA, Sept. 1949.

[177] Shaw, M. Beyond Objects: A Software Design Paradigm Based on Process Control.

ACM SIGSOFT Software Engineering Notes 20, 1 (Jan. 1995), 27–38.

[178] Sheluhin, O., Smolskiy, S., and Osin, A. *Self-Similar Processes in Telecommunications*. John Wiley & Sons, Inc., New York, NY, USA, 2007.

[179] Skoglund, J., Kozica, E., Linden, J., Hagen, R., and Kleijn, W. B. *Voice over IP: Speech Transmission over Packet Networks*. Springer Handbook of Speech Processing. Springer-Verlag Berlin Heidelberg, 2008, ch. 15, pp. 307–330.

[180] Skomeršić, M., and Parat, N. Interoperability of open-source VoIP and multi-agent systems. In *Proceedings of the 31st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (Opatija, Croatia, May 2008).

[181] Skype. SILK: Super Wideband Audio Codec, Aug. 2011. `https://developer.skype.com/silk`.

[182] SoX – Sound eXchange. Swiss Army knife of sound processing programs, Aug. 2011. `http://sox.sourceforge.net/`.

[183] Stallings, W. *Data and Computer Communications*, 8 ed. Prentice Hall, 2007.

[184] Stankiewicz, R., Cholda, P., and Jajszczyk, A. QoX: What is It Really? *Communications Magazine, IEEE 49*, 4 (Apr. 2011), 148–158.

[185] Stankovic, J. A. Misconceptions about real-time computing: A serious problem for next-generation systems. *Computer 21*, 10 (Oct. 1988), 10–19.

[186] Stern, H. P., Mahmoud, S. A., and Wong, K.-K. A Model for Generating On-Off Patterns in Conversational Speech, Including Short Silence Gaps and the Effects of Interaction Between Parties. *IEEE Transactions on Vehicular Technology 43*, 4 (Nov. 1994), 1094–1100.

[187] Sun, L., and Ifeachor, E. New models for Perceived Voice Quality Prediction and their Applications in Playout Buffer Optimization for VoIP Networks. In *Proceedings of the 39th IEEE International Conference on Communications (ICC'04)* (Paris, France, June 2004), IEEE, pp. 1478–1483.

[188] Takahashi, A. *QoE Assessment and Management of VoIP Services*. VoIP Handbook: Applications, Technologies, Reliability, and Security. CRC Press, Boca Raton, FL, USA, 2009, ch. 9, pp. 153–167.

[189] Takahashi, A., Yoshino, H., and Kitawaki, N. Perceptual QoS Assessment Technologies for VoIP. *IEEE Communications Magazine 42*, 7 (July 2004), 28–34.

[190] Taqqu, M. S., Willinger, W., and Sherman, R. Proof of a Fundamental Result in Self-Similar Traffic Modeling. *ACM Computer Communications Review 27* (Apr. 1997), 5–23.

[191] Tebbani, B., and Haddadou, K. Codec-based Adaptive QoS Control for VoWLAN with Differentiated Services. In *Proceedings of the 1st IFIP Wireless Days (WD'08)* (Dubai, Nov. 2008), IEEE, pp. 1–5.

[192] The Linux Foundation. netem, Aug. 2011. `http://www.linuxfoundation.org/collaborate/workgroups/networking/netem`.

[193] Trad, A., Ni, Q., and Afifi, H. Adaptive VoIP Transmission over Heterogeneous Wired / Wireless Networks. In *Proceedings of the 2nd International Workshop on Multimedia Interactive Protocols and Systems (MIPS 2004)* (Grenoble, France, Nov. 2004),

ACM, pp. 25–36.

[194] Tu, C., Liu, X., and Wu, Z. A research on adaptive QoS technology. In *Proceedings of the 1st International Conference on Future Computer and Communication (ICFCC'09)* (Kuala Lumpur, Malaysia, Apr. 2009), IEEE, pp. 164–166.

[195] Tüysüz, M. F., and Mantar, H. A. A Cross Layer QoS Algorithm to Improve Wireless Link Throughput and Voice Quality over Multi-rate WLANs. In *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference (IWCMC'10)* (Caen, France, June 2010), ACM, pp. 209–213.

[196] Valle, R., Carvalho, L. S. G., Aguiar, R. B., Mota, E., and Freitas, D. Dynamical Management of Dejitter Buffers Based on Speech Quality. In *Proceedings of the 15th IEEE Symposium on Computers and Communications (ISCC'10)* (Riccione, Italy, June 2010), IEEE, pp. 56–61.

[197] van der Hoek, W., and Wooldridge, M. *Multi-Agent Systems.* Handbook of Knowledge Representation. Elsevier Science Ltd, Amsterdam, The Netherlands, Jan. 2008, ch. 24, pp. 887–928.

[198] Villegas, N. M., Müller, H. A., Tamura, G., Duchien, L., and Casallas, R. A Framework for Evaluating Quality-Driven Self-Adaptive Software Systems. In *Proceeding of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS'11)* (Waikiki, Honolulu, HI, USA, May 2011), ACM, pp. 80–89.

[199] Wah, B. W., and Sat, B. The Design of VoIP Systems With High Perceptual Conversational Quality. *Journal of Multimedia 4*, 2 (Apr. 2009), 49–62.

[200] Wältermann, M., Lewcio, B., Vidales, P., and Möller, S. A Technique for Seamless VoIP-Codec Switching in Next Generation Networks. In *Proceedings of the 43rd IEEE International Conference on Communications (ICC'08)* (Beijing, China, May 2008), IEEE, pp. 1772–1776.

[201] Weyns, D., Malek, S., and Andersson, J. FORMS: a FOrmal Reference Model for Self-adaptation. In *Proceeding of the 7th International Conference on Autonomic Computing (ICAC'10)* (Washington, DC, USA, June 2010), ACM, pp. 205–214.

[202] Wierzbicki, A. *Trust and Fairness in Open, Distributed Systems.* Springer-Verlag, New York, NY, USA, 2010.

[203] Wooldridge, M. *An Introduction To Multiagent Systems*, 2nd ed. John Wiley & Sons, Chichester, England, UK, 2009.

[204] Xi, B., Chen, H., Cleveland, W. S., and Telkamp, T. Statistical analysis and modeling of Internet VoIP traffic for Network Engineering. *Electronic Journal of Statistics 4* (2010), 58–116.

[205] Yuhe, S., and Jie, X. New Solutions of VoIP on Multi-hop Wireless Network. In *Proceedings of the IITA International Conference on Control, Automation and Systems Engineering (CASE)* (Zhangjiajie, China, July 2009), IEEE, pp. 199–202.

[206] Zhang, H., Zhao, J., and Yang, O. Adaptive Rate Control for VoIP in Wireless Ad Hoc Networks. In *Proceedings of the 43rd IEEE International Conference on Communications (ICC'08)* (Beijing, China, May 2008), IEEE, pp. 3166–3170.

[207] Zhou, J., She, X., and Chen, L. Source and Channel Coding Adaptation for Optimizing VoIP Quality of Experience in Cellular Systems. In *Proceeding of the 11th IEEE*

Wireless Communications and Networking Conference (WCNC'10) (Sidney, Australia, Apr. 2010), IEEE, pp. 1–6.