

CAIO CESAR DE SOUZA BISPO

IMS PARA COMPUTAÇÃO EM NUVEM

São Carlos

2012

CAIO CESAR DE SOUZA BISPO

IMS PARA COMPUTAÇÃO EM NUVEM

Trabalho de Conclusão de Curso
apresentado à Escola de Engenharia
de São Carlos, da Universidade de
São Paulo

Curso de Engenharia de Computação
com ênfase em Robótica

ORIENTADORA: Mônica de Lacerda Rocha

São Carlos

2012

FOLHA DE APROVAÇÃO

Nome: Caio Cesar de Souza Bispo

Título: “IMS para Computação em Nuvem”

Trabalho de Conclusão de Curso defendido em 26 / 11 / 2012.

Comissão Julgadora:

Resultado:

Prof. Dr. Ricardo Augusto Souza Fernandes
SEL/EESC/USP

Aprovado

M.Sc. Mariana Massimino Feres
SEL/EESC/USP

Aprovado

Orientador:

Profa. Dra. Mônica de Lacerda Rocha - SEL/EESC/USP

Coordenador pela EESC/USP do Curso de Engenharia de Computação:

Prof. Associado Evandro Luís Linhari Rodrigues

Sumário

Lista de Figuras	7
Lista de Siglas	8
Resumo	11
Abstract	12
1. Introdução.....	13
1.1. Motivação	13
1.2. Objetivo	13
1.3. Organização dos Capítulos.....	14
2. Redes Heterogêneas.....	15
2.1. Conceito e Histórico da telefonia celular [1]	15
2.2. Gerações de redes [1]	16
2.3. Primeira Geração de Redes – 1G [1]	16
2.3.1. NMT	16
2.3.2. AMPS	17
2.4. Segunda Geração de Redes – 2G [1]	17
2.4.1. TDMA	18
2.4.2. CDMA.....	18
2.4.3. GSM	19
2.5. Terceira Geração de Redes – 3G [1]	19
2.5.1. UMTS	20
2.6. WiMAX [12].....	21
2.7. Nova Geração de Redes [4]	22
3. Computação em Nuvem [5]	23
3.1. Introdução	23
3.2. Paradigma emergente	24
3.3. Características da Computação em Nuvem [16].....	24

3.4.	SaaS.....	25
3.4.1.	Implementação do Modelo SaaS	26
3.4.2.	Características do SaaS	27
3.5.	PaaS.....	27
3.5.1.	O modelo tradicional <i>On-Premises</i>	28
3.5.2.	O novo modelo de <i>Cloud (Off-Premises)</i>	28
3.5.3.	Características principais do PaaS	29
3.6.	IaaS.....	29
3.7.	<i>Hadoop</i>	31
4.	<i>IP MultimediaSubsystem (IMS)</i> [6].....	32
4.1.	Introdução	32
4.2.	Motivações para a utilização do IMS	32
4.3.	Arquitetura IMS TISpan	33
4.3.1.	Visão geral	33
4.3.2.	NASS	34
4.3.3.	RACS	35
4.3.4.	O Núcleo IMS	35
4.3.5.	Componentes adicionais	36
4.4.	Protocolos utilizados em IMS.....	36
4.4.1.	Assinatura e descrição do fluxo de mídia.....	37
4.4.2.	Autenticação, Autorização e Auditoria (AAA) [14]	37
4.4.3.	Protocolos adicionais.....	37
4.5.	Suporte QoS em IMS.....	38
4.5.1.	Princípio do gerenciamento QoS	38
4.5.2.	Arquitetura QoS.....	38
5.	Plataforma Android	40
5.1.	Conceito	40
5.2.	Histórico	40
5.3.	Arquitetura [7]	41

5.3.1.	<i>Kernel Linux</i>	41
5.3.2.	Bibliotecas	42
5.3.3.	Ambiente de Execução.....	42
5.3.4.	<i>Framework</i>	43
5.3.5.	Aplicativos	43
5.4.	Ambientes de Desenvolvimento [7].....	44
5.5.	Estado de uma aplicação [7].....	44
6.	Aplicações com a sobreposição de tecnologias [4]	45
6.1.	Introdução	45
6.2.	Nuvem IMS com Mecanismo QoS	46
6.2.1.	Camada IaaS no sistema.....	48
6.2.2.	Camada PaaS no sistema	49
6.2.3.	Camada SaaS no sistema	49
6.3.	Design do Sistema e Análise de Desempenho	49
6.3.1.	Implementação do Sistema	50
6.4.	Análise do Desempenho.....	54
7.	Conclusões.....	56
	Referências Bibliográficas	57

Lista de Figuras

Figura 1 - <i>DynaTAC</i> 8000x [2]	15
Figura 2 - Arquitetura UMTS [3]	21
Figura 3 - Sistema em Computação em Nuvem para acesso à Internet [17].....	23
Figura 4 - Arquitetura e Interfaces do IMS TISPAN [6].....	33
Figura 5 - Núcleo IMS [6]	35
Figura 6 - Interação do RACS com funções de transferência [6].....	39
Figura 7 - IMS em computação em nuvem [4].....	46
Figura 8 - Modelo do sistema, com SaaS, PaaS e IaaS inclusos [4]	47
Figura 9 - Cenário de teste para análise de desempenho do sistema proposto [4]	50
Figura 10 - Interface web do sistema <i>Hadoop</i> [4].....	51
Figura 11 - Ambiente operacional do IMS [4]	52
Figura 12 - Módulos CSCF [4]	53
Figura 13 - Telefone com <i>Android</i> acessando serviços da nuvem [4]	53
Figura 14 - Gráfico de capacidade do sistema (%) pelo número de clientes [4]	54
Figura 15 - Gráfico do rendimento (Mb/s) por tempo (s) [4]	55

Lista de Siglas

3GPP: *Third Generation Partnership Project*

AMPS: *Advanced Mobile Phone System*

API: *Application Programming Interface*

APK: *Android Package*

A-RACF: *Access Resource and Admission Control Facility*

ARP: *Address Resolution Protocol*

ASP: *Application Service Provider*

BGCF: *Breakout Gateway Control Function*

BGF: *Border Gateway Function*

BSC: *Base Station Control*

BTS: *Base Transceiver Station*

CDMA: *Code Division Multiple Access*

COPS: *Common Open Policy Service*

FDMA: *Frequency Division Multiple Access*

GGSN: *Gateway GPRS Support Node*

GPRS: *General Packet Radio Service*

GPS: *Global Positioning System*

GSM: *Global Systems for Mobile Communications*

HDFS: *Hadoop Distributed File System*

HLR: *Home Location Register*

HSS: *Home Subscribe Server*

HTML: *HyperText Markup Language*

HTTP: *Hypertext Transfer Protocol*

I-CSCF: *Interrogating Call Session Control Function*

IMS: *IP Multimedia Subsystem*

IP: *Internet Protocol*

L2TP: *Layer 2 Termination Point*

MCS: *Monitoring Control and Surveillance*

MeGaCo: *Media Gateway Control*

MGCF: *Media Gateway Control Function*

MRFC: *Multimedia Resource Function Control*

MRFP: *Multimedia Resource Function Processor*

NASS: *Network Attachment Subsystem*

NGN: *Next Generation Network*

NMT: *Nordic Mobile Telephone*

PaaS: *Platform as a Service*

PCEF: *Policy Charging and Enforcement Function*

PCRF: *Policy Charging and Rules Function*

P-CSCF: *Proxy Call Session Control Function*

PDN: *Public Data Network*

QoS: *Quality of Service*

RCEF: *Resource Control Enforcement Function*

REST: *Representational State Transfer*

RNC: *Radio Network Controller*

RTP: *Real Time Protocol*

SaaS: *Software as a Service*

S-CSCF: *Serving Call Session Control Function*

SDK: *Software Development Kit*

SGSN: *Serving GPRS Support Node*

SIP: *Session Initiation Protocol*

SLF: *Subscriber Location Function*

SOA: *Service-Oriented Architecture*

SOAP: *Simple Object Access Protocol*

TDMA: *Time Division Multiple Access*

TISPAN: *Telecommunications and Internet converged Services and Protocols for Advanced Networking*

UE: *User Equipment*

UPSF: *User Profile Server Function*

UTRAN: *Universal Terrestrial Radio Access Network*

VLR: *Visitor Location Register*

WiMAX: *Worldwide Interoperability for Microwave Access*

WLAN: *Wireless Local Area Network*

WWW: *World Wide Web*

Resumo

Com o desenvolvimento veloz da Internet nos últimos anos, são necessárias novas tecnologias, as quais suportem a alta demanda de usuários pelos recursos da web. Neste trabalho, foram estudadas quatro tecnologias: redes heterogêneas, computação em nuvem, *IP Multimedia Subsystem* (IMS) e plataforma Android. Após a revisão bibliográfica dos quatro conceitos listados acima, foi realizado um estudo de aplicações multimídia em computação em nuvem, utilizando a política de Qualidade de Serviço (QoS) do IMS com a finalidade de resolver questões de heterogeneidade e mobilidade, e assim satisfazer as necessidades do usuário. O sistema proposto foi um mecanismo IMS em computação em nuvem, o qual possui três camadas: IaaS (Infraestrutura como serviço), PaaS (Plataforma como serviço) e SaaS (Software como serviço), e o resultado obtido desse sistema implementado foi uma capacidade de rendimento maior com a utilização da computação em nuvem.

Palavras-chave: redes heterogêneas, computação em nuvem, IMS, plataforma Android, QoS

Abstract

With the Internet fast development in the last years, new technologies are required, which support the high demand from users for web resources. In this work, four technologies were studied: heterogeneous networks, cloud computing, *IP Multimedia Subsystem* (IMS) and Android platform. After bibliographic review of the four concepts listed above, a study about multimedia applications in cloud computing was performed, using the Quality of Service (QoS) policy of IMS, in order to resolve heterogeneity and mobility issues, and thus satisfy the user needs. The system proposed was a IMS cloud computing mechanism, which has three layers: IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service), and the result of this implemented system was a higher throughput capacity with cloud computing usage.

Key-words: heterogeneous networks, cloud computing, IMS, Android platform, QoS

1. Introdução

1.1. Motivação

Nos dias atuais, a sociedade presencia um rápido desenvolvimento das aplicações da Internet, gerando uma constante modificação das tecnologias para comunicação sem fio e computação móvel. Entre essas tecnologias, estão as redes heterogêneas, a computação em nuvem, o IMS e a Plataforma Android.

As redes heterogêneas consistem na nova geração de redes, a qual provê enlaces com a utilização das tecnologias 3G e WiMAX, e dependendo das prioridades pré-definidas, o usuário pode conectar-se à rede mais adequada. Uma rede heterogênea é otimizada pela mudança dinâmica e automática do acesso à rede [4].

O conceito de computação em nuvem refere-se à utilização da memória e das capacidades de armazenamento e cálculo de computadores e servidores compartilhados e interligados por meio da Internet, seguindo o princípio da computação em grade. O armazenamento de dados é realizado em serviços os quais poderão ser acessados de qualquer lugar do mundo, a qualquer hora, sem a necessidade de instalação de programas ou de armazenar dados, e o acesso a programas, serviços e arquivos é remoto, através da Internet [5].

O IMS (*IP Multimedia Subsystem*) utiliza padronização técnica proposta pelo 3GPP (*3rd Generation Partnership Project*, uma tecnologia de padronização de arquivos multimídia para telefones celulares). Esta plataforma se baseia no protocolo SIP (*Session Initiation Protocol*), uma arquitetura aberta a qual fornece voz, dados e vídeo, e pode atuar como uma rede fixa, com ou sem fio. A proposta do IMS é prover serviços numa infraestrutura de rede 3G e permitir QoS em diferentes redes [6].

A plataforma Android é um conjunto de softwares para dispositivos móveis. Consiste num sistema operacional, *middleware* e aplicações com mobilidade. Esta plataforma permite a introdução de novas aplicações devido ao ambiente de desenvolvimento em código aberto do *Kernel* do Linux [7].

1.2. Objetivo

O objetivo deste trabalho é o estudo dos princípios das quatro tecnologias citadas acima, e de aplicações multimídia de alta qualidade, utilizando os conceitos delas, de modo a satisfazer necessidades do usuário, como recursos sempre disponíveis, e em tempo real.

1.3. Organização dos Capítulos

O conteúdo da monografia está dividido em seis capítulos, sendo o primeiro capítulo a introdução, explicando a motivação, objetivo e divisão dos capítulos do trabalho.

O segundo capítulo descreve o conceito das redes heterogêneas, passando pelo histórico da telefonia celular, as gerações de rede 1G, 2G, 3G, e a sobreposição das tecnologias 3G e WiMax.

O terceiro capítulo apresenta o conceito de computação em nuvem, além das camadas PaaS, SaaS e IaaS.

O quarto capítulo trata dos conceitos de IMS, incluindo motivação, arquiteturas, protocolos utilizados, e a política de QoS.

O quinto capítulo apresenta a Plataforma Android, incluindo o conceito, histórico e a descrição das camadas deste sistema operacional para dispositivos móveis.

O sexto e último capítulo apresenta um estudo sobre aplicações multimídia de alta qualidade em computação em nuvem, utilizando política de QoS do IMS, com a finalidade de satisfazer os requisitos do usuário, e resolver aspectos de homogeneidade e mobilidade.

2. Redes Heterogêneas

2.1. Conceito e Histórico da telefonia celular [1]

O telefone celular é um dispositivo no qual sua comunicação é feita por meio de ondas eletromagnéticas, permitindo a transmissão bidirecional de voz e dados em uma área geográfica, que é dividida em células. Cada uma dessas células é servida por um transmissor/receptor.

Em 1947, foi desenvolvido nos Laboratórios Bell, da AT&T, o conceito do celular. Em 1970, a mesma AT&T promoveu a construção do AMPS (*Advanced Mobile Phone Service*), o primeiro sistema telefônico celular de alta capacidade. No final de 1983, entrou em operação comercial o primeiro sistema celular nos Estados Unidos, o *DynaTAC 8000x*, da Motorola, ilustrado na figura 1.



Figura 1 - *DynaTAC 8000x* [2]

No Brasil, o primeiro celular lançado foi pela TELERJ, no Rio de Janeiro, em 1990. Começou a operar o Sistema Móvel Celular, com capacidade para dez mil terminais.

Antigamente, ainda na tecnologia analógica, os telefones celulares possuíam apenas a realização de chamadas como função, mas hoje eles já são usados para

enviar mensagens, tirar fotos, filmar, ouvir músicas, gravar lembretes, navegar na Internet, baixar aplicativos, etc.

2.2. Gerações de redes [1]

Existem diversas tecnologias para a difusão de ondas eletromagnéticas nos telefones celulares, baseadas na distribuição ou compressão de informações:

- Primeira Geração (1G): sistemas NMT e AMPS;
- Segunda Geração (2G): GSM, CDMA e TDMA;
- Terceira Geração (3G): UMTS.

2.3. Primeira Geração de Redes – 1G [1]

As redes móveis da primeira geração são analógicas, ou seja, a difusão das ondas eletromagnéticas ocorre de forma contínua. A seguir serão descritos os sistemas NMT e AMPS.

Uma desvantagem das redes 1G é que são mais sensíveis a interferências, devido aos sinais serem analógicos. E com isso elas são utilizadas somente para voz.

2.3.1. NMT

O primeiro sistema de telefonia móvel automático é o NMT (*Nordic Mobile Telephone*). As operadoras de países nórdicos especificaram entre os anos de 1973 e 1977, e começaram a operar em 1981, devido ao alto congestionamento e requisitos das redes móveis manuais ARP (*Address Resolution Protocol*) na Finlândia e MTD (*Mobile Telephone system D*) na Suécia, Dinamarca e Noruega.

O NMT baseia-se em tecnologia analógica e existe em duas variações: NMT-450 e NMT-900. Os números indicam a banda de frequência utilizada em MHz.

As especificações de NMT eram abertas, possibilitando que várias empresas desenvolvessem *hardware* e conseqüentemente houvesse a redução dos preços, e assim a Nokia e a Ericsson foram muito beneficiadas.

Os primeiros aparelhos NMT foram projetados para ficar no carro com um teclado, e as versões portáteis podiam ser carregadas por uma pessoa, porém eram muito pesadas.

2.3.2. AMPS

O AMPS (*Advanced Mobile Phone System*) foi desenvolvido em 1979 nos Laboratórios Bell, e entrou em operação no mundo a partir de 1983, tornando-se o principal sistema analógico.

Esse sistema foi padronizado pela EIA-553 (padrão desenvolvido pela Associação da Indústria Eletrônica), e serviu como base para outros sistemas analógicos. Nos Estados Unidos, o AMPS foi padronizado para a frequência de 800 MHz. Além disso, ele utiliza FDMA (*Frequency Division Multiple Access*), no qual o usuário possui um canal de frequência alocada em um instante, e é trocado para outro canal se o original sofre interferência.

2.4. Segunda Geração de Redes – 2G [1]

A segunda geração de redes surgiu no início dos anos 90, e utiliza tecnologia digital para transmissão de dados, a qual possui diversas vantagens sobre a tecnologia analógica (1G), entre elas:

- Melhor uso do espectro;
- A qualidade da transmissão da voz não é deformada conforme aumenta-se a distância;
- Mais segura, é mais difícil de ser decodificada;
- Exige menos potência transmitida;
- Utiliza receptores e transmissores mais baratos.

As redes 2G utilizam três tecnologias de acesso múltiplo: TDMA, CDMA e GSM. A primeira (*Time Division Multiple Access*) possibilita o triplo de chamadas do que um único canal FDMA. A segunda (*Code Division Multiple Access*) aloca o espectro inteiro em qualquer instante. E a última (*Global Systems for Mobile Communications*) divide um canal com frequência de 25 MHz em 124 frequências de

200KHz cada, e utiliza oito intervalos de tempo para transmitir até 9,6 Kbits por segundo.

2.4.1. TDMA

O TDMA divide um canal de frequência em até seis intervalos de tempo diferentes. Cada usuário ocupa um desses intervalos de tempo específicos na transmissão, impedindo problemas de transferência. Ele é utilizado pelos sistemas celulares da segunda geração, como o IS 54, IS 136 e o GSM, em suas interfaces com a estação móvel.

Um dos padrões de comunicação de voz através de ondas de rádio é feito por operadoras nos serviços de telefonia celular digital, baseando-se em TDMA. Cada canal celular é dividido em três intervalos de tempo, de forma a aumentar a quantidade de dados a ser transmitida, e cada canal TDMA americano possui a mesma largura de banda dos canais AMPS (1G), e é utilizado por três assinantes. O sinal digitalizado de cada assinante (64 Kb/s) é comprimido para 8 Kb/s através de vocoders (padrão IS). Esse sinal comprimido é transmitido pelo mesmo canal, um de cada vez. Com isso os padrões TDMA IS-54 e IS-136 triplicam a capacidade do padrão AMPS. O IS-136 diferencia-se do IS-54 por possuir introdução de um canal de controle digital.

2.4.2. CDMA

O CDMA é um método utilizado tanto para telefonia celular quanto para rastreamento via satélite (GPS), e usa os prefixos tecnológicos como o IS-95 da primeira geração e o IS-2000 da terceira geração.

Essa tecnologia codifica dados utilizando um código especial associado a cada canal, e usa propriedades de interferência dos códigos especiais para realizar a multiplexagem. Além disso, o CDMA faz uma busca pelos sistemas celulares digitais do telefone que empregam esse esquema de acesso múltiplo.

Além do CDMA ser utilizado para o GPS, ele é utilizado em outros sistemas de comunicações, como o sistema satélite OmniTRACS para a logística do transporte. E, além disso, permite um melhor desempenho em aplicativos multimídia, além de transmissão de voz pelo celular.

2.4.3. GSM

A tecnologia GSM é a mais popular para telefones celulares no mundo. A grande presença desse sistema faz com que o *roaming* (conexão em redes em áreas fora da localidade de onde o usuário está registrado) internacional seja comum através de acordos entre operadoras de telefonia celular.

O GSM é um padrão aberto desenvolvido pela 3GPP, e diferencia-se de seus antecessores sendo que os canais de voz e o sinal são digitais. Com isso, a comunicação de dados foi acoplada ao sistema logo no início.

A codificação da voz é feita de forma complexa, na qual erros na transmissão possam ser encontrados e corrigidos. Depois a codificação digital é enviada aos intervalos de tempo, cada um com duração de 577 milissegundos, e uma capacidade de 116 bits codificados. Cada terminal deve ter uma agilidade de frequência, podendo locomover-se entre os intervalos de tempo usados para envio, recepção e controle dentro de um *frame* completo. Simultaneamente, um celular verifica outros canais para identificar se o sinal é mais forte e encaminhar a transmissão para eles, em caso de uma resposta positiva.

Essa tecnologia possui como vantagens novos serviços a relativo baixo custo para os usuários, e baixo custo de infraestrutura devido à competição aberta entre as operadoras. Porém, o sistema GSM é baseado na rede TDMA, menos avançada que a CDMA.

2.5. Terceira Geração de Redes – 3G [1]

A terceira geração de tecnologias de telefonia celular baseia-se na família de normas do ITU-T (*International Telegraph Union*) no âmbito do Programa Internacional de Telecomunicações Móveis (IMT-2000). As tecnologias dessa geração possibilitam às operadoras da rede o fornecimento de serviços avançados, porque elas têm maior capacidade de rede em função de uma melhoria na eficiência espectral. Entre os serviços existe a telefonia por voz e a transmissão de dados a longas distâncias, e geralmente as taxas dos serviços são de 5 a 10 Mb/s.

As redes 3G permitem telefonia móvel de longo alcance e evoluíram com a finalidade de incorporar redes de acesso à Internet em velocidades mais altas e vídeo-telefonia, em comparação às redes definidas pelo padrão IEEE 802.11 (Wi-Fi, WLAN),

as quais são de curto alcance, ampla largura de banda e originalmente desenvolvidas para redes de dados, além de não terem preocupação quanto ao consumo de energia.

A tecnologia 3G surgiu no começo de 2003 na Europa, inicialmente no Reino Unido e na Itália. Em alguns países a implantação das redes 3G foi tardia, em virtude de altos custos adicionais para licenciamento do espectro. Em vários países, as redes 3G não utilizam as mesmas frequências de rádio que as redes 2G usam, com isso as operadoras tinham que construir redes completamente novas e licenciar novas frequências.

A principal característica da tecnologia 3G é suportar um grande número de clientes de dados e voz, especialmente em regiões urbanas, e também maiores taxas de dados a um custo incremental menor que na tecnologia 2G. Além disso, ela usa o espectro de radiofrequência em bandas identificadas, oferecidas pela ITU-T para a terceira geração de serviços móveis IMT-2000, e depois licenciadas para as operadoras.

O IMT-2000 consiste de seis interfaces de rádio:

- W-CDMA;
- CDMA2000;
- TD-CDMA/TD-SCDMA (*Time Division Code Division Multiple Access / Time Division Synchronous Code Division Multiple Access*);
- UWC (*Universal Wireless Communications*, normalmente é implementado como EDGE - *Enhanced Data Rates for GSM Evolution*);
- DECT (*Digital Enhanced Cordless Telecommunications*);
- Mobile WiMAX.

2.5.1. UMTS

O UMTS (*Universal Mobile Telecommunication System*) é uma tecnologia 3G, estabelecido como uma evolução para operadoras de GSM, o WCDMA.

O desenvolvimento de padrões para o GSM foi conduzido pelo ETSI (*European Telecommunications Standards Institute*) até o ano 2000. Após isso o 3GPP tornou-se responsável. O 3GPP é um projeto de várias organizações de padronizações de diversos lugares do mundo com a finalidade de definir um sistema celular global de terceira geração UMTS.

A principal missão do UMTS é prover um padrão universal para as comunicações pessoais, com o auxílio do mercado de massa e com a qualidade de serviços equivalente à rede fixa. E em relação ao UMTS, três quesitos são relevantes: rádio acesso de banda larga, *roaming* inteligente e alta capacidade.

A arquitetura UMTS é ilustrada na figura 2.

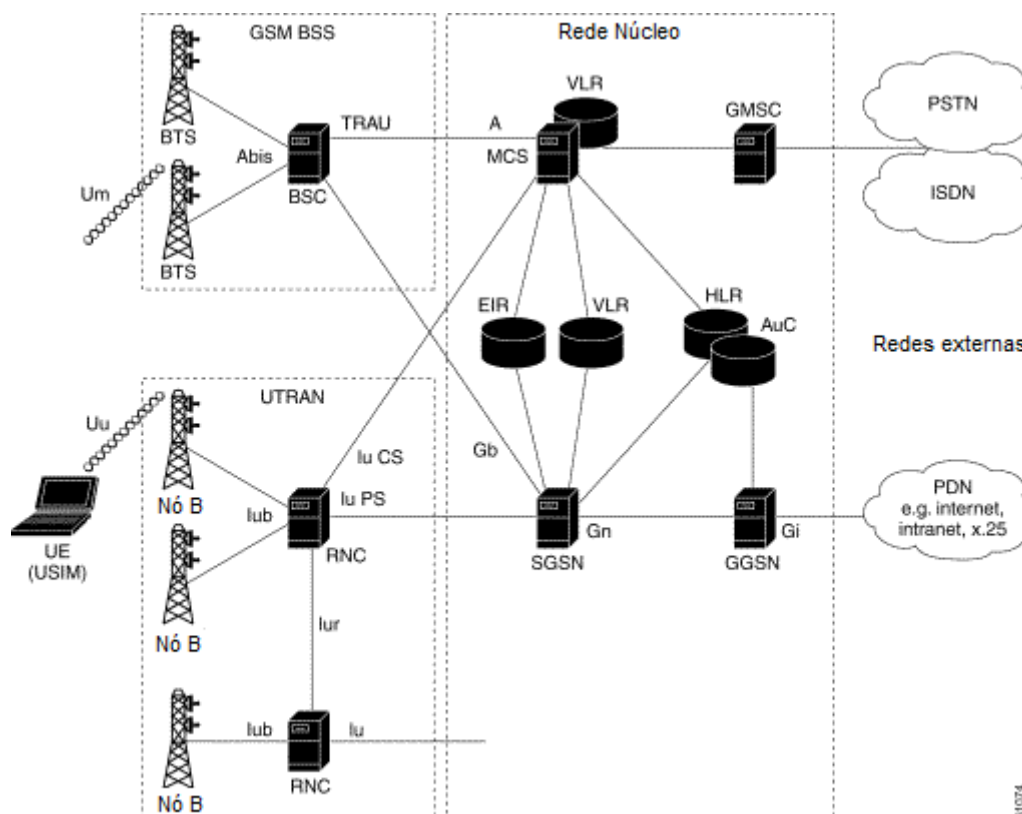


Figura 2 - Arquitetura UMTS [3]

(UTRAN = Universal Terrestrial Radio Access Network; RNC = Radio Network Controller; UE = User Equipment; PDN = Public Data Network; BTS = Base Transceiver Station; SGSN = Serving GPRS Support Node; GGSN = Gateway GPRS Support Node; EIR = Equipment Identity Register, VLR = Visitor Location Register; HLR = Home Location Register, MCS = Monitoring Control and Surveillance; BSC = Base Station Control; GSM = Global System for Mobile Communications)

2.6. WiMAX [12]

WiMAX (Worldwide Interoperability for Microwave Access) é uma tecnologia sem fio com a finalidade de oferecer acesso banda larga a distâncias de 6 a 9 quilômetros. Essa tecnologia é implantada em células, e da estação base é possível a

transmissão para uma estação terminal, a qual provê acesso a uma rede local ou de forma direta até os aparelhos dos usuários.

O WiMAX possui como uma das principais aplicações a oferta de acessos banda larga à Internet, e ele foi desenvolvido tendo em vista as seguintes aplicações.

- WiMAX Fixo: as estações terminais podem possuir mobilidade restrita. O local onde localiza-se a estação terminal pode variar dentro da célula, porém ela está parada quando em operação;
- WiMAX Móvel: a rede WiMAX é composta por um conjunto de células, e os terminais são portáteis e móveis, assim é possível trocar de célula durante a comunicação.

2.7. Nova Geração de Redes [4]

As redes heterogêneas são a próxima geração de redes de comunicação, as quais fornecem enlaces sem fio através de 3G, WLAN (*Wireless Local Area Network*) ou WiMAX em redes sobrepostas. Os usuários podem conectar à rede mais apropriada, dependendo da prioridade pré-definida.

A terceira geração de redes suporta comunicação multimídia, e para realizar uma rede QoS, um serviço portador com características definidas é configurado a partir da fonte de um serviço para seu destino. Os serviços das redes 3G possuem classes QoS diferentes para quatro tipos diferentes de tráfego: classe de conversação, classe de streaming, classe interativa e classe background.

A tecnologia de banda de base sem fio WiMAX fornece acesso de longa distância para redes de faixa larga, e os protocolos IEEE 802.11x para WLAN utilizam o mesmo protocolo básico para tecnologias de modulação sobre o ar.

Nas redes heterogêneas, as tecnologias 3G podem suportar uma taxa máxima de 144 Kbps em um ambiente de alta mobilidade, 384 Kbps em um ambiente de baixa ou média mobilidade, e 2 Mbps em um ambiente estático. As redes WiMAX suportam terminais móveis locomovendo-se em velocidades de até 120 Km/h. Porém quando a velocidade do terminal aumenta, o desempenho do sistema piora.

3. Computação em Nuvem [5]

3.1. Introdução

O termo “nuvem” é uma metáfora utilizada para a Internet, onde a palavra foi utilizada para representar um esboço da infraestrutura e dos recursos computacionais, os quais aparecem ocultos ao nível do usuário, que só enxerga o local de origem e o local de destino.

A figura 3 ilustra um exemplo de sistema em computação em nuvem, com a utilização de um modelo cliente/servidor para acesso à Internet.

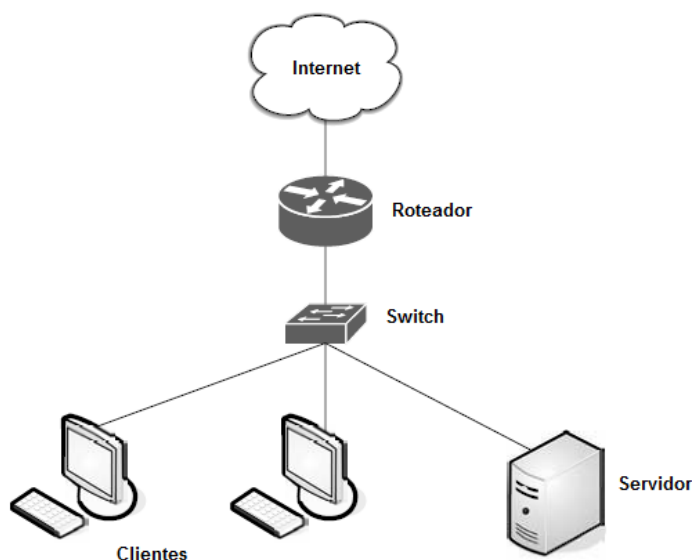


Figura 3 - Sistema em Computação em Nuvem para acesso à Internet [17]

O primeiro conceito de computação em nuvem veio em 1961, quando John McCarthy sugeriu que a tecnologia de *time-sharing* (compartilhamento de recursos computacionais entre vários usuários) dos computadores poderia levar a um futuro promissor. Porém, nos anos 70 a ideia perdeu força quando as tecnologias relacionadas à área de TI um dia foram incapazes de sustentar tal modelo computacional. Entretanto, na virada do milênio, o conceito foi revitalizado, e o termo “computação em nuvem” começou a emergir em ciclos tecnológicos.

3.2. Paradigma emergente

A computação em nuvem é um paradigma emergente, que está sendo utilizado pelas empresas, fornecendo servidores virtuais, os quais departamentos de TI e usuários podem realizar o acesso. Antigamente, as empresas adotavam o serviço de computação utilitária, um serviço no qual os clientes recebem recursos de TI através de terceiros, e pagam pelo que usa. Porém, esse tipo de serviço era usado para necessidades que não eram críticas, e assim foi motivada a mudança de paradigma.

3.3. Características da Computação em Nuvem [16]

Existem várias características de um ambiente de computação em nuvem. Nesta seção serão descritas as principais características.

Uma característica é a de elasticidade e escalonamento, na qual a computação em nuvem propicia a ilusão de recursos computacionais infinitos disponíveis. Assim, os usuários tem a expectativa de que os recursos podem ser fornecidos em qualquer quantidade e a qualquer momento, e com isso espera-se que os recursos adicionais possam ser fornecidos de forma autônoma, de acordo com o aumento da demanda, e retidos em caso de diminuição da mesma.

Outra característica é a de autoatendimento, no qual o consumidor de serviços da computação em nuvem tem a expectativa de adquirir recursos computacionais, de acordo com a sua necessidade e em tempo real. Para isso as nuvens devem permitir o acesso em autoatendimento, com a finalidade de que os usuários possam solicitar, personalizar, pagar e utilizar os serviços desejados sem intervenção humana.

O faturamento e medição por uso é outra característica importante. Uma vez que o usuário possui a opção de requisitar e utilizar somente a quantidade de recursos necessários, esses recursos devem ser precificados com base no uso de baixa duração, como por exemplo, medição por horas de uso. Com isso as nuvens implementam recursos que garantem um eficiente comércio de serviços.

A customização também é uma característica importante. No atendimento a vários usuários, verifica-se a grande diferença de necessidades dos mesmos, e assim é fundamental a capacidade de personalização dos recursos da nuvem.

3.4. SaaS

O “Software como um Produto” é o modelo tradicional de distribuição de *software*, no qual ele é instalado diretamente no *hardware* do computador. Diferente dele, o SaaS (*Software como um Serviço*) possui uma distribuição de aplicativos por um serviço conectado à web, armazenados em um servidor, disponibilizados aos clientes através da Internet.

O SaaS está se tornando um modelo dominante na distribuição de tecnologias suportadas pela web, orientadas por uma arquitetura SOA (Arquitetura Orientada a Serviço), cujo princípio é de que as funcionalidades implementadas pelas aplicações necessitam estar disponíveis na forma de serviços. Consequentemente, o SaaS está se tornando um modelo cada vez mais popular, e está relacionado à condição de assinatura de uma licença, como no modelo pré-pago. Além disso, ele possui algumas funcionalidades aos seus clientes pelo seu caráter de baixo custo, permitindo que estes possuam benefícios sem haver custos e complicações internas na instalação, gerenciamento, suporte e compra de licenças.

Uma grande evolução feita pelos ISP's (*Internet Service Providers*, responsáveis por oferecer acesso à Internet) é o fornecimento de plataformas maiores para design, desenvolvimento e uso de *softwares* em todas as áreas de negócios e computação pessoal, com a finalidade de fornecer uma banda maior, e uma constante introdução de microprocessadores mais potentes acoplados aos dispositivos de armazenamento de dados com baixo custo. As aplicações desse tipo de computação em nuvem também devem interagir com outros dados e aplicações igualmente em uma grande diversidade de ambientes e plataformas.

O modelo de gerenciamento de hospedagem das aplicações é parecido com o modelo ASP (*Application Service Provider*). No ASP, os aplicativos dos *softwares* são comercializados e distribuídos através da Internet. Já o outro modelo faz a distribuição por demanda, na qual o fornecedor permite acesso ao cliente em sua rede para uma cópia do *software* feita especialmente para essa finalidade.

Hoje em dia, diversos tipos de *software* são disponibilizados para o modelo SaaS, com a finalidade de gerenciamento de contas e relacionamento junto aos clientes, *software* de e-mails, recursos humanos, segurança em TI, gerenciamento de serviços de TI, vídeo-conferência, análise de estatísticas da rede e acesso à web, gerenciamento de conteúdo web, etc.

As soluções do modelo SaaS foram implementadas especificamente para trabalharem com um navegador, e a forma de implementação está descrita a seguir.

3.4.1. Implementação do Modelo SaaS

Vários tipos de componentes de *softwares* e de *frameworks* de aplicação podem ser aplicados na implementação de aplicações SaaS. A aparição de novas tecnologias, componentes e aplicações modernos podem diminuir drasticamente o tempo de comercialização e o custo para converter um produto do modelo tradicional para uma solução SaaS.

A arquitetura SaaS pode ser classificada em quatro níveis de maturidade, e os atributos chaves são configuração, eficiência multidisciplinar e escalabilidade. Cada nível diferencia-se do nível anterior pela adição de um desses atributos. Os níveis são:

3.4.1.1. Nível 1 – Ad-hoc / Custom

Esse nível possui a menor maturidade dos quatro níveis. Cada cliente tem uma única e customizada versão do aplicativo de hospedagem, e o aplicativo é executado no servidor de hospedagem.

Na migração de um aplicativo tradicional ou cliente-servidor para este nível de SaaS, é necessária maturidade para mais um esforço de desenvolvimento, e que reduz custos operacionais por consolidar o servidor de *hardware*.

3.4.1.2. Nível 2 – Configurabilidade

O nível de configurabilidade fornece maior flexibilidade do programa através da configuração de metadados (são dados que possuem informações sobre outros dados). Nesse nível, vários clientes podem utilizar instâncias separadas para a mesma aplicação, possibilitando ao vendedor diversificar as configurações de acordo com as necessidades do cliente e com diversas opções. Além disso, esse nível permite que o vendedor reduza a carga de manutenção por ser capaz de realizar atualizações baseando-se num código comum.

3.4.1.3. Nível 3 – Eficiência para vários usuários

Esse nível adiciona o caráter de eficiência em locação múltipla ao segundo nível, ou seja, um simples programa nesse nível possuirá a capacidade de servir todos os clientes. Essa abordagem possibilita maior eficiência na utilização dos recursos, sem haver diferença aparente para o usuário final, porém é limitado quanto à questão de escalabilidade.

3.4.1.4. Nível 4 – Escalabilidade

A escalabilidade é adicionada ao nível anterior com a utilização de uma arquitetura em multicamadas. Esta arquitetura é capaz de suportar uma carga de exploração de aplicações idênticas sendo executadas em centenas ou milhares de servidores. A capacidade do sistema pode ser acrescida ou reduzida de forma dinâmica de forma a equacionar a demanda de carga, adicionando ou removendo servidores, sem a necessidade de modificar a arquitetura dos aplicativos e *softwares*.

3.4.2. Características do SaaS

Desenvolver aplicações em uma arquitetura orientada a serviço é usualmente um problema de maior complexidade, em comparação com os modelos tradicionais de desenvolvimento de *software*. Com isso, as aplicações SaaS são em maioria precificadas de acordo com o número de usuários os quais possuem acesso ao serviço. As principais características dos *softwares* SaaS são:

- Gerenciamento de rede e acesso para comercialização de *software* disponível em locais centrais ao invés de um site para cada cliente, permitindo o acesso remoto às aplicações via Internet;
- Distribuição de aplicações de um modelo “um para vários” (instância simples, arquitetura de multi-locações), em oposição ao modelo “um para um”;
- Atualizações que remediam qualquer necessidade de download e instalação feita pelo usuário.

3.5. PaaS

A computação em nuvem tem evoluído de forma a incluir plataformas para o desenvolvimento e execução de aplicações web. Este conceito é conhecido como

PaaS (Plataforma como um Serviço). É considerada uma evolução do SaaS, e cria todas as facilidades necessárias para o suporte de um ciclo de vida completo da implementação e entrega de aplicações web e serviços inteiramente disponíveis na Internet, sem a necessidade de downloads ou instalações feitas por desenvolvedores, gerentes de TI ou usuários finais.

Diferente do modelo IaaS, no qual desenvolvedores podem criar um sistema operacional específico para os aplicativos, os desenvolvedores do modelo PaaS preocupam-se apenas com a implementação baseando-se na web, e normalmente não se importam com o sistema operacional utilizado. Com isso, os serviços PaaS possibilitam que os usuários esforcem-se na inovação ao invés da infraestrutura complexa, inovação na qual usuários com conexão à Internet podem desenvolver um aplicativo e facilmente difundi-los para outros usuários em escala global.

3.5.1. O modelo tradicional *On-Premises*

A abordagem tradicional para a implementação e execução de aplicativos *on-premises* (softwares locais) sempre foi complexa, cara e de altos riscos. Desenvolver a própria solução nunca trouxe qualquer garantia de sucesso, e cada solução necessita de um hardware específico, um sistema operacional, um banco de dados, servidores web, etc.

Uma vez criados os ambientes de hardware e software, uma equipe de programadores necessitava utilizar complexas plataformas de desenvolvimento para implementar as suas próprias aplicações. Além disso, eram necessários um banco de dados, uma equipe de redes e pessoas experientes em gerenciamento de sistemas para manter tudo em perfeitas condições de execução. Inevitavelmente, os desenvolvedores eram obrigados a modificar a aplicação por causa de um detalhe, gerando novos ciclos de testes antes da distribuição. Não bastassem apenas esses problemas, grandes empresas consumiam grandes quantidades de energia para manter suas centrais de dados na temperatura ideal. Com isso surgiu um novo modelo (*Off-Premises*).

3.5.2. O novo modelo de *Cloud (Off-Premises)*

PaaS oferece um modelo mais veloz e com melhor custo-benefício para o desenvolvimento e distribuição de aplicações, e fornece toda a infraestrutura

necessária para a execução de aplicações sobre a Internet. Assim como a *Amazon.com*, *eBay*, *Google*, *iTunes* e o *YouTube*, o novo modelo de computação em nuvem permite que novas funcionalidades sejam entregues aos novos mercados através de navegadores web.

PaaS baseia-se em um modelo de assinatura ou mediação, no qual os usuários pagam apenas pelo que utilizam. Ele ainda oferece algumas facilidades para o design e desenvolvimento da aplicação, testes, hospedagem, integração, segurança, escalabilidade, armazenamento, etc.

3.5.3. Características principais do PaaS

Entre as principais características do modelo PaaS, encontram-se serviços para implementar, testar, armazenar e gerenciar aplicações com a função de suportar o ciclo de vida do desenvolvimento de aplicações. O PaaS, baseado na interface web para a criação de ferramentas, fornece um nível de suporte de forma a simplificar a criação de interfaces com o usuário, seguindo padrões como as linguagens HTML (*HyperText Markup Language*), JavaScript ou alguma outra. Suportar uma arquitetura de inúmeros locatários auxilia na remoção das preocupações de desenvolvimento com relação ao uso das aplicações por muitos usuários concorrentes.

Outra característica é a integração com servidores web e bancos de dados. O suporte para o protocolo SOAP (*Simple Object Access Protocol*) e outras interfaces possibilita ao PaaS criar combinações de serviços web (chamados de *mashups*), bem como ter a capacidade de acessar bancos de dados e reutilizar serviços mantidos em redes privadas.

3.6. IaaS

Infraestrutura como um Serviço (IaaS) é a distribuição da infraestrutura de computação como um serviço em um ambiente de plataforma virtual. IaaS alavanca significativa tecnologia, serviços e investimentos em centrais de dados para entregar TI como um serviço aos clientes.

Diferente da tradicional terceirização, a qual requer negociações complexas e contratos longos para a construção da infraestrutura, o IaaS é centrado em torno de um modelo pré-definido, padronizado e otimizado para as aplicações dos clientes, simplificando o trabalho criando um serviço de escolha.

Os fornecedores de IaaS gerenciam a hospedagem das aplicações em suas infraestruturas, e os clientes mantêm a propriedade e o gerenciamento de suas aplicações, sem preocupar-se com a hospedagem das operações e a manutenção da infraestrutura. Na implementação desses fornecedores, são necessárias as seguintes camadas de componentes:

- Hardware do computador, o qual é ajustado para garantir escalabilidade;
- Rede do computador, incluindo roteadores, *firewalls*, etc.;
- Conectividade da Internet, geralmente em OC 192 *backbones* (a linha de transmissão *Optical Carrier* 192 tem a capacidade de transferir 9,95 gigabits de dados por segundo);
- Ambiente de virtualização de plataforma para executar máquinas virtuais específicas de clientes;
- Acordos para níveis de serviços.

Em vez de adquirir espaço em datacenter, servidores, software, equipamento de rede, etc., os clientes do IaaS alugam esses recursos especialmente como um serviço terceirizado. E os benefícios primordiais para a utilização desse tipo de serviço terceirizado são:

- Pronto acesso a um ambiente pré-configurado, que normalmente é baseado no ITIL (*The Information Technology Infrastructure Library*, um *framework* customizado para as melhores práticas projetadas com a finalidade de promover qualidade nos serviços computacionais no setor de TI);
- Utilização da tecnologia mais recente para os equipamentos de infraestrutura;
- Plataformas altamente seguras, protegidas e isoladas de violações;
- Baixos riscos, em razão da manutenção dos recursos ser realizada por empresas terceirizadas;
- Habilidade de gerenciar as oscilações de demanda;
- Baixos custos que permitem gastos com a contratação de serviços, ao invés de investir capital em infraestrutura;
- Redução de tempo, custos e complexidade em adição a novos recursos ou capacidades.

3.7. Hadoop

Hadoop é uma plataforma de software para um ambiente de desenvolvimento em Java, que possui alta escalabilidade, eficiência e segurança. Os dois componentes do *Hadoop* são o HDFS (*Hadoop Distributed File System*) e o *MapReduce*. O primeiro é um sistema de arquivos distribuídos cuja execução é feita em hardware, e o segundo é um modelo de programação com a finalidade de processamento de dados em computação distribuída [13].

No *framework hadoop*, o HDFS divide uma grande quantidade de informações em vários blocos, os quais são armazenados em diferentes nós, e o *MapReduce* é utilizado para obter o valor dos arquivos.

4. IP MultimediaSubsystem (IMS) [6]

4.1. Introdução

O IMS (*IP Multimedia Subsystem*) é arquitetura de padronização das redes de nova geração (NGN) voltada para a distribuição de serviços multimídia, como voz, vídeo e dados, baseada no protocolo de Internet (IP). Seu objetivo é a sobreposição de dois dos paradigmas em comunicações mais bem sucedidos: a Internet e as redes de celulares [14].

O início do IMS foi no começo de 2003, em uma versão cujo foco era de facilitar o desenvolvimento de novos serviços em redes móveis. Em 2005 o IMS evoluiu, passando a ser padronizado pela TISPAN (*Telecommunications and Internet converged Services and Protocols for Advanced Networking*), cujos detalhes serão descritos durante o capítulo.

4.2. Motivações para a utilização do IMS

Conforme visto na Introdução, o rápido desenvolvimento da Internet gerou como consequência a necessidade de novas tecnologias visando atender as necessidades do usuário. Alguns serviços da Internet, como o e-mail e o WWW (*World Wide Web*) foram possíveis por causa de protocolos disponíveis para qualquer desenvolvedor poder implementar o seu próprio aplicativo [14].

Enquanto o rendimento médio por usuário está diminuindo para várias operadoras de rede, o IMS é visto por muitos como uma solução que deve suportar a criação e desenvolvimento de serviços inovadores por operadoras ou terceiros e, portanto, criar novas perspectivas de negócio, pois as operadoras possuem o conhecimento dos atuais serviços utilizados pelo consumidor, em IMS. Além disso, seu rápido desenvolvimento reduz o tempo de mercado e estimula a inovação.

Outra motivação é a do IMS ser uma arquitetura horizontal, a qual oferece um conjunto de funções para chamadas de habilitadores de serviço, que podem ser utilizado por diversos serviços, tornando a implementação destes mais simples e rápida.

Outra questão importante que o IMS possui é prover qualidade de serviço (QoS). Com a evolução da Internet, aumentou a necessidade do fornecimento de serviços multimídia em tempo real. Não há garantia que a rede fornece largura de

banda suficiente para um usuário, nem que não haverá atrasos na entrega dos pacotes. O IMS é responsável pela sincronização das conexões, fazendo com que os serviços sejam fornecidos em tempo real, provendo QoS [14].

4.3. Arquitetura IMS TISPAN

4.3.1. Visão geral

A arquitetura IMS TISPAN é um serviço o qual controla a infraestrutura conforme a figura 4.

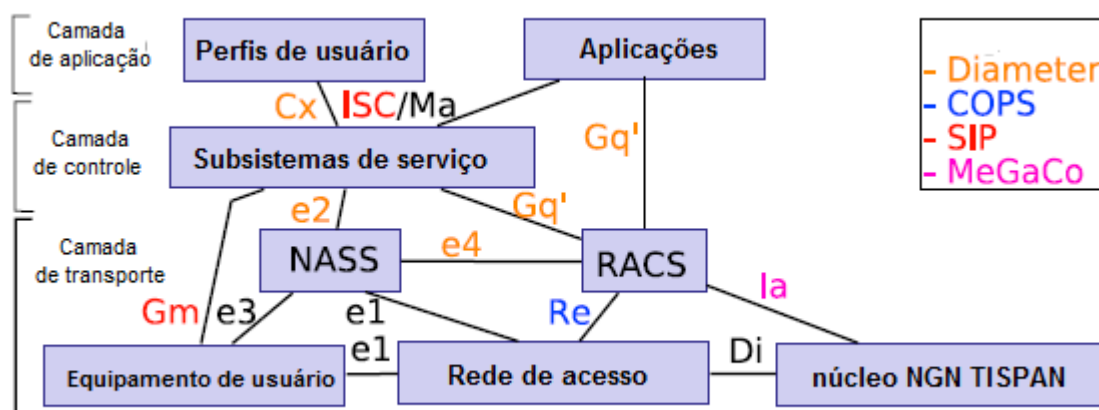


Figura 4 - Arquitetura e Interfaces do IMS TISPAN [6]

(NASS = Network Attachment Subsystem; TISPAN = Telecommunications and Internet converged; RACS = Resource and Admission Control Subsystem; COPS = Common Open Policy Service; SIP = Session Initiation Protocol; MeGaCo = Media Gateway Control)

Ela é dividida em:

- Uma camada de aplicação, que consiste em um servidor de aplicações (AS), o qual hospeda os serviços do IMS e o HSS (*Home Subscribe Server*), um sistema, o qual inclui um banco de dados de usuário mestre, que oferece suporte a entidades da rede IMS, para manipular chamadas;
- Uma camada de controle composta por vários subsistemas de serviços, entre eles o núcleo IMS;
- Uma camada de transporte, que consiste no equipamento de usuário.

Muitos subsistemas de serviços podem coexistir em uma arquitetura IMS, como por exemplo o núcleo IMS e a função de emulação PSTN (rede pública de telefonia comutada), responsável por identificar a rede telefônica mundial comutada por circuitos.

4.3.2. NASS

O NASS (*Network Attachment Subsystem*) é um subsistema que fornece endereço IP e outros parâmetros de configuração dos equipamentos do usuário dinamicamente. Suas funcionalidades podem ser resumidas da seguinte maneira: o NASS desempenha o papel de um servidor DHCP (protocolo de configuração dinâmica de host), um cliente RADIUS (*Remote Authentication Dial In User Service*, um protocolo de rede que provê autenticação, autorização e auditoria, que serão vistos nesse capítulo), e fornece funcionalidades de gerenciamento de localização. Mais precisamente ele provê:

- Endereço IP e parâmetros de configuração;
- Autenticação do usuário;
- Autorização de acesso à rede, baseando-se no perfil do usuário;
- Configuração de acesso à rede, também se baseando no perfil do usuário;
- Gerenciamento de localização.

O NASS pode ser dividido em várias entidades funcionais:

- NACF (função de configuração de acesso à rede), responsável pela alocação do endereço IP para o equipamento do usuário, e pode fornecer parâmetros adicionais. Pode ser implementado por um servidor DHCP;
- AMF (função de gerenciamento de acesso), que é na maior parte das vezes um equipamento de interface entre o acesso à rede e o NACF;
- CLF (função de localização da sessão de conectividade e repositório), o qual é utilizado para associar o endereço IP do usuário à sua informação sobre a localização. Além disso, ele pode armazenar informações sobre o usuário;
- UAAF (função de autorização de acesso do usuário), responsável pela autenticação para acesso à rede, baseado no perfil do usuário armazenado no PDBF;

- PDBF (função de base de dados do usuário), que armazena os perfis dos usuários e dados de autenticação;
- CNGCF (função de configuração do CNG), o qual é utilizado para configurar o CNG (*Customer Network Gateway*), quando necessário.

4.3.3. RACS

O RACS (*Resource and Admission Control Function*), outro subsistema que não é IMS, é utilizado para controle de admissão. Ele pode ser dividido em dois blocos funcionais: S-PDF (*Service Policy Decision Function*) e A-RACF (*Access Resource and Admission Control Facility*). O S-PDF toma as decisões políticas e é capaz de enviar recursos para A-RACF, e o A-RACF é responsável pelo controle de admissão, ou seja, ele verifica se os recursos requeridos podem ser alocados para o acesso envolvido, e retorna o resultado do controle de admissão para o S-PDF.

4.3.4. O Núcleo IMS

O núcleo IMS é utilizado para a sessão e controle de mídia. Sua estrutura é ilustrada na figura 5.

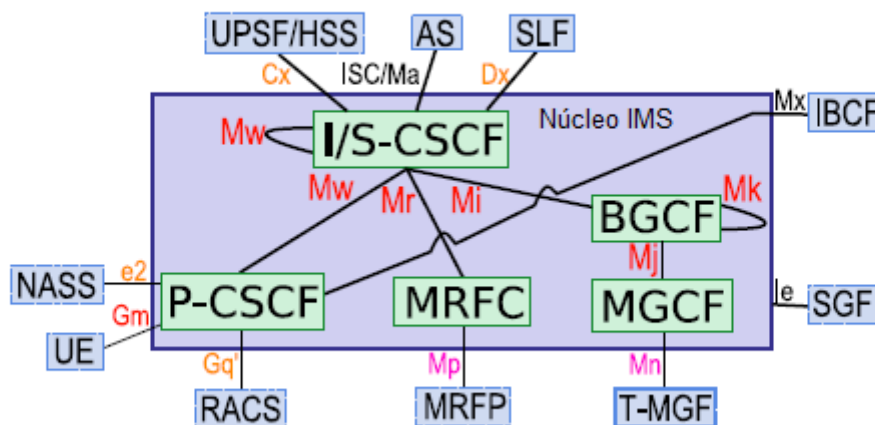


Figura 5 - Núcleo IMS [6]

(UPSF = *User Profile Server Function*; HSS = *Home Subscriber Server*; SLF = *Subscriber Location Function*; P-CSCF = *Proxy - Call Session Control Function*; I/S-CSCF = *Interrogating/Serving - Call Session Control Function*; BGCF = *Breakout Gateway Control*

Function; MRFC = *Media Resource Function Control*; MGCF = *Media Gateway Control Function*; NASS = *Network Attachment Subsystem*)

Os componentes do núcleo IMS são os seguintes:

- CSCF (*Call Session Control Function*), responsável pelo estabelecimento, monitoramento, suporte e gerenciamento das interações de serviço do usuário. Ele pode desempenhar três funções diferentes: S-CSCF (*Serving-CSCF*), P-CSCF (*Proxy-CSCF*) e I-CSCF (*Interrogating-CSCF*). O primeiro é o servidor proxy, o qual controla a sessão de comunicação, chamando os servidores de aplicações relacionados aos serviços requisitados, o segundo é o ponto de contato do IMS para os agentes de usuário SIP, e o terceiro fornece um *gateway* para os outros domínios;
- MRFC (*Multimedia Resource Function Control*), o qual é utilizado para controlar o MRFP (*Multimedia Resource Function Processor*), o qual essencialmente fornece funcionalidades de adaptação de conteúdo e transcodificação;
- BGCF (*Breakout Gateway Control Function*), responsável pelo interfuncionamento com o domínio de comutação de circuito;
- MGCF (*Media Gateway Controller Function*), o qual é usado para controlar um *gateway* de mídia.

4.3.5. Componentes adicionais

Os servidores de aplicações são entidades SIP, as quais executam os serviços. Eles podem ter uma interface para o *framework* SOA ou para a função de controle de serviços GSM.

O HSS ou UPSF (*User Profile Server Function*) é definido em um banco de dados, armazenando informações do perfil do usuário. Ele pode ser visto como uma evolução do HLR (*Home Location Register*), um banco de dados de informações de assinantes para redes móveis, como CDMA, TDMA e GSM.

4.4. Protocolos utilizados em IMS

Muitos dos protocolos utilizados em IMS são padronizados pelo IETF (*Internet Engineering Task Force*), uma comunidade internacional responsável por propor

melhorias ao uso da Internet, e também a padronização de tecnologias e protocolos relacionados. Esses protocolos estão descritos nas seções abaixo [15].

4.4.1. Assinatura e descrição do fluxo de mídia

O principal protocolo de assinatura utilizado no IMS é chamado de SIP (Protocolo de Iniciação de Sessão). Este utiliza alguns princípios do HTTP (*Hypertext Transfer Protocol*) e SMTP (*Simple Mail Transfer Protocol*), os dois protocolos mais bem sucedidos da Internet.

SIP está sendo selecionado em IMS especialmente porque ele cumpre os requerimentos do IMS, e é considerado flexível e seguro. Sua principal missão é o estabelecimento, modificação e encerramento de sessões multimídia entre dois terminais. E o corpo de suas mensagens é composto usando o SDP (Protocolo de Descrição de Sessão), uma sintaxe para descrever os fluxos de mídia (endereço, porta, tipo de mídia, codificação, etc.).

4.4.2. Autenticação, Autorização e Auditoria (AAA) [14]

Diameter é um protocolo AAA, o qual substitui o protocolo RADIUS. Ele consiste de um protocolo base e de aplicações *Diameter*, customizações feitas para se encaixarem em aplicações particulares em determinados ambientes.

Esse protocolo é utilizado em várias interfaces do IMS, mesmo sem as interfaces utilizarem as mesmas aplicações. Por exemplo, o IMS define uma aplicação *Diameter* para interagir com o protocolo SIP, e outra aplicação para contabilizar o controle de crédito do usuário.

4.4.3. Protocolos adicionais

O protocolo MeGaCo (*Media Gateway Control*), também chamado de H248, é o sucessor do MGCP (*Media Gateway Control Protocol*), usado para controlar as funções de serviços de mídia em um ambiente IMS.

Outro protocolo é o RTP (*Real Time Protocol*), cuja função é fornecer funções de transporte para transmitir dados em tempo real. É usado em conjunto com o RTCP (*Real Time Control Protocol*) a fim de permitir o monitoramento da entrega de dados, e fornecer a funcionalidade de identificação e controle.

Além dos protocolos citados anteriormente, existe o protocolo COPS (*Common Open Policy Service*), responsável pela especificação cliente/servidor com a finalidade de apoio na política de controle sobre QoS e protocolos de sinalização.

4.5. Suporte QoS em IMS

A ideia de transformar o melhor esforço do IP da rede introduzindo QoS fim-a-fim é uma garantia importante para o desenvolvimento do IMS. Essa é uma consideração chave, pois o nível de QoS o qual a arquitetura IMS é capaz de fornecer determina os serviços que podem ser implantados.

4.5.1. Princípio do gerenciamento QoS

Duas estratégias são usualmente associadas para fornecer um bom nível de QoS das redes. A primeira consiste em evitar congestionamento, e pode ser feita implementando o CAC (*Connection Admission Control*, um conjunto de ações e permissões nas redes de comunicação que identificam qual conexão é permitida, baseando-se na capacidade de rede), reserva de recursos ou simplesmente redimensionamento da rede. A segunda foca na gerência de congestionamento, na qual se baseia geralmente na diferenciação de tráfego, de forma a providenciar melhor QoS, e vários padrões estão relacionados à essa ideia, sendo o *DiffServ* o mais conhecido.

4.5.2. Arquitetura QoS

As redes IMS suportam tanto controle de admissão como diferenciação QoS. As funções principais da rede principal envolvidas na rede IMS estão descritas na figura 6.

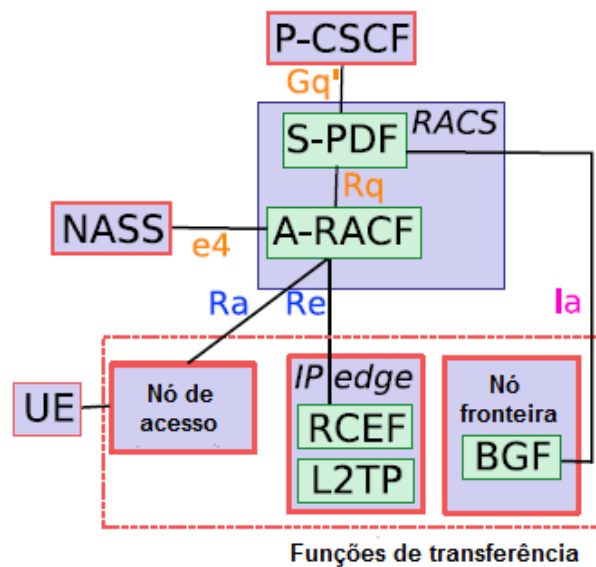


Figura 6 - Interação do RACS com funções de transferência [6]

(P-CSCF = Proxy - Call Session Control Function; NASS = Network Attachment Subsystem; A-RACF = Acronym – Resource Access Control Facility; BGF = Border Gateway Function; RCEF = Resource Control Enforcement Function; L2TP = Layer 2 Termination Point)

Nessa figura são encontrados os seguintes elementos.

- P-CSCF, o qual serve como porta de entrada para o domínio IMS, e também como servidor proxy de saída para o UE;
- RACS, responsável por tomar as decisões políticas e interfaces com funções de transferência;
- Componentes da camada de transporte que são aplicadas nas decisões políticas.

As funções da camada principal de transporte estão descritas abaixo:

- RCEF (*Resource Control Enforcement Function*), responsável por impor políticas sob o controle do A-RACF;
- BGF (*Border Gateway Function*), o qual impõe políticas e funções NAT (*Network Address Translation*) sob o controle do S-PDF;
- L2TP (*Layer 2 Termination Point*), responsável pelos procedimentos da camada 2 de acesso à rede.

5. Plataforma Android

5.1. Conceito

A plataforma Android consiste em uma pilha de componentes de *software*, desenvolvida para celulares e tablets. Entre esses componentes de *software* estão sistema operacional, bibliotecas, *frameworks* de *middleware* e aplicações chave. Essa plataforma baseia-se no *kernel* do Linux 2.6, com diversas alterações feitas de forma a adaptar o sistema operacional aos dispositivos móveis. Ele é responsável por alguns processos fundamentais do sistema, como gerenciamento de memória, segurança, etc [7].

Algumas das características do Android são o *framework* de aplicações, o qual permite o reuso e a substituição de componentes; a máquina virtual *Dalvik*, a qual executa o *software* na linguagem de programação Java, e compilado num formato especial de *bytecodes*; o banco SQLite para armazenar dados estruturados, um navegador baseado no *WebkitEngine*, o qual é usado também pelo *Safari*; telefonia GSM, 3G, Wifi, e outras características [7].

5.2. Histórico

Em 2003 foi fundada a empresa *Android Inc.* em Palo Alto, Califórnia. Ela desenvolvia sistemas operacionais para telefones móveis, mas os projetos eram feitos em sigilo, e em meados de 2005 a *Google* comprou essa empresa. Nessa época foi implementada uma plataforma para telefones celulares baseada no sistema operacional *Linux*, com a finalidade de ser flexível, e de código aberto [8].

Em 2008 foi lançado o primeiro aparelho com Android, o *HTC Dream G1*, o qual possuía integração com o Gmail e o *Android Market*, a loja de aplicativos da *Google* [8].

No primeiro semestre de 2009 surgiu a versão 1.5 do Android, conhecida também como *Cupcake*, com novidades como integração com o *Youtube* e o teclado virtual [8].

Em Setembro de 2009 foi lançada a versão 1.6, apelidada de *Donut*, tendo indicador de nível de bateria, gravador de voz e galeria de fotos como principais novidades [8].

Um mês depois surgiu a versão 2.0, ou *Eclair*. Nessa versão o usuário tem acesso à navegação pelo *Google Maps*, múltiplas contas de e-mail, além de suporte a *Bluetooth 2.1* e *HTML5* [8].

No primeiro semestre de 2010 a *Google* anunciou a versão 2.2 (*Froyo*). Além da melhoria na velocidade na realização de tarefas, uma novidade foi a opção de salvar aplicativos em cartões de memória [8].

No final de 2010 foi lançado o *Gingerbread* (Android 2.3), o qual possui uma interface para celulares e tablets. Algumas das melhorias dessa versão foram o suporte para chamadas pela internet, além do teclado e opções de edição [8].

O Android 3.0 (*Honeycomb*) foi lançado no primeiro semestre de 2011 para os tablets. As principais melhorias foram a customização da tela principal e o compartilhamento via *bluetooth* [8].

No segundo semestre de 2011 a *Google* liberou a versão 4.0 do Android (*Ice Cream Sandwich*). Entre as melhorias estão maior facilidade no compartilhamento de arquivos e o reconhecimento facial para desbloqueio de tela [8].

Em 2012 surgiu a versão 4.1 do Android, também conhecida como *JellyBean*. Além da melhoria na velocidade da interface e na tela principal, essa versão possui como recursos um aplicativo novo para a câmera, um novo motor de busca e o *Google Now*, um aplicativo inteligente, que se baseia nas buscas recentes do usuário para resultados de pesquisa mais refinados [9].

5.3. Arquitetura [7]

A arquitetura da plataforma Android é dividida em: *kernel Linux*, bibliotecas, ambiente de execução, *framework* e aplicativo.

5.3.1. Kernel Linux

A camada de *kernel Linux* baseia-se no *kernel* do *Linux 2.6*, e é responsável pelas tarefas fundamentais do sistema, como segurança, gerenciamento de memória e de processos, pilha de protocolos de rede e modelo de *drivers*. Além disso, ela atua

como uma camada de abstração entre o *hardware* e os demais componentes da plataforma, estes executados a nível de usuário.

Muitas alterações foram feitas no *Linux* para adaptar o sistema às características dos celulares e tablets, como adições ao sistema de gerenciamento de energia, conhecido como *wakelocks*, e um sistema que permite finalizar processos, quando a memória disponível está baixa, chamado de *lowmem killer*.

A camada do *Kernel Linux* consiste de drivers de display, câmera, bluetooth, memória flash, USB, teclado, Wifi.

5.3.2. Bibliotecas

A camada de bibliotecas da plataforma Android possui um conjunto de bibliotecas C/C++ utilizadas pelo sistema. Entre essas bibliotecas, encontram-se a biblioteca C padrão (Libc) do BSD (*Berkeley Software Distribution*, um sistema operacional derivado do Linux), adaptada para os dispositivos executando *Linux*. Essas bibliotecas suportam os principais formatos de áudio e vídeo, imagens estáticas, visualização de câmeras 2D e 3D, funções para *browsers web*, gráficos, aceleração de *hardware*, renderização 3D, fontes *bitmap* e vetorizadas e acesso ao banco *SQLite* (uma biblioteca da linguagem C na qual é implementado um banco de dados SQL). Todas essas funcionalidades encontram-se no *framework* para o desenvolvimento de aplicações.

Algumas das bibliotecas encontradas são: *Surface Manager*, *Media Framework*, *SQLite*, *WebKit*, etc.

5.3.3. Ambiente de Execução

A plataforma Android possui a máquina virtual *Dalvik* ao invés da máquina virtual Java (JVM), mesmo usando a linguagem de programação Java, pelo motivo de *Dalvik* ser otimizada para execuções em dispositivos móveis.

Cada aplicativo da plataforma Android executa o seu próprio processo em uma instância da máquina virtual *Dalvik*. Essa máquina foi escrita para ser executada

eficientemente em outras instâncias e rodando arquivos com a extensão *.dex* (arquivos executáveis, que contém o código escrito para Android) em um formato de *bytecode* projetado para ocupar menos espaço e carregar com maior velocidade. Os arquivos *.dex* e outros recursos como imagem são compactados em um único arquivo, cuja extensão é *.apk* (*Android Package File*), representando a aplicação final, a qual é executada pelo usuário.

5.3.4. Framework

Na camada de *Framework* da plataforma Android, os desenvolvedores têm acesso ao mesmo *Framework* das API's (*Application Programming Interface*, um conjunto de padrões definidos por um *software* para que as funcionalidades sejam construídas e utilizadas de maneira não tão evidente para os usuários) utilizadas para as aplicações do Android, facilitando o reuso de componentes, e abstraindo a maioria dos procedimentos necessários para fazer uma aplicação funcionar.

Uma grande vantagem da camada de *Framework* de aplicação do Android é que cada aplicação dessa plataforma pode publicar suas capacidades para serem usadas por outras aplicações que estiverem sendo executadas.

Entre algumas das API's, temos:

- *Location Manager*;
- *Telephony Manager*;
- *Content Providers*;
- *Window Manager*;
- *Resource Manage*;
- *Notification Manage*;
- *Activity Manager*.

5.3.5. Aplicativos

A camada mais alta da arquitetura da plataforma Android é a de aplicativos em nível de usuário, como e-mail, *browser*, mapas, calendários, etc. Esses são programados na linguagem de programação Java, e convertidos para arquivos *.dex*.

5.4. Ambientes de Desenvolvimento [7]

O SDK (*Software Development Kit*) é um dispositivo móvel virtual, que possibilita ao desenvolvedor executar e testar as aplicações desenvolvidas, e é composto pela API das classes Java e pelo emulador de Android. O SDK foi disponibilizado pela *Google* antes mesmo da utilização da plataforma Android.

Além do SDK, outros ambientes de desenvolvimento que podem ser utilizados são o *Eclipse*, *Netbeans*, *Intelli J IDEA*, *JDK 5*, *JDK 6*, *Apache Ant 1.6.5* ou superiores para *Linux* e *Mac*, *1.7* ou superiores para *Windows*, sendo o *Eclipse* o ambiente de desenvolvimento principal, e usado pela *Google*. Além disso, há o ADT (*Android Development Tools*), um plugin que integra o emulador ao *Eclipse*, permitindo que novos projetos Android sejam criados de forma mais rápida.

Uma aplicação Android é dividida em quatro blocos: *Activity*, *Intent Receiver*, *Service* e *Content Provider*. O primeiro é o bloco mais utilizado, no qual cada *Activity* representa uma tela da aplicação apresentada ao usuário, o *Intent Receiver* é responsável pela reposta da aplicação a eventos externos. O *Service* é utilizado para códigos de vida longa que rodam sem interface para o usuário, e o *Content Provider* é o provedor de conteúdo.

5.5. Estado de uma aplicação [7]

O sistema operacional Android é responsável pelo gerenciamento de processos, podendo ser necessária até mesmo a finalização deles, para liberar memória e recursos, ou ainda reiniciá-los quando a situação estiver controlada, levando em conta os estados de prioridade da aplicação que podem ser de:

- Primeiro plano, quando há uma atividade sendo utilizada;
- Visível, porém há uma outra janela sobre ela;
- Serviço, o qual roda um código de serviço, sem UI (*User Interface*);
- Fundo, quando possui uma atividade que não está visível;
- Vazio, quando não há nenhum componente ativo, servindo como cachê para a próxima vez que a aplicação for iniciada.

6. Aplicações com a sobreposição de tecnologias [4]

6.1. Introdução

Foram vistos os conceitos de redes heterogêneas, computação em nuvem, IMS e plataforma Android, as quais atendem as necessidades impostas pelo rápido desenvolvimento da Internet. Após a revisão bibliográfica desses quatro conceitos separadamente, este capítulo será sobre tecnologias que sobrepõem as metodologias vistas nos capítulos anteriores.

A computação em nuvem possui uma grande expectativa não só porque ela possibilita um maior armazenamento de dados, mas também uma maior capacidade de desempenho computacional. O IMS possui como uma de suas características o suporte a redes heterogêneas e provê política de Qualidade de Serviço (QoS). Além disso, para computação móvel, é necessária uma plataforma aberta para o acesso dos serviços multimídia, e o Android atende esse requisito devido ao seu desenvolvimento em *open source* no *kernel* do Linux. Todas essas questões devem ser levadas em conta no desenvolvimento de tecnologias de comunicação *wireless* e aplicações da nova geração.

A figura 7 ilustra uma arquitetura IMS em computação em nuvem, utilizando redes heterogêneas. Nesse contexto, o IMS possui como funções fornecer QoS para os serviços, prover mecanismos de extensão de carga para os serviços multimídia e integrar os serviços *All-IP*.

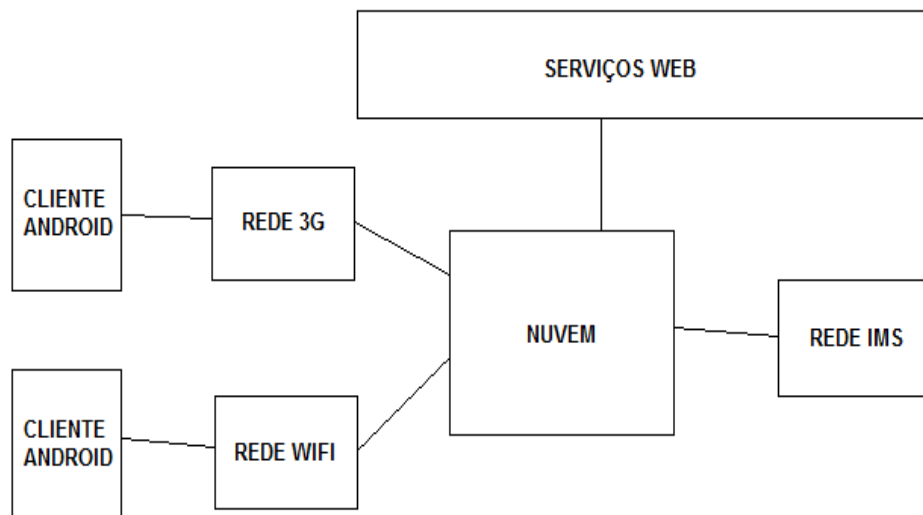


Figura 7 - IMS em computação em nuvem [4]

A arquitetura IMS, a qual é baseada no sistema 3GPP, é definida pelo módulo de política QoS, no qual consiste do PCRF (*Policy Charging and Rules Function*), responsável pela definição de regras da política QoS, PCEF (*Policy Charging and Enforcement Function*), o qual recebe as essas regras para transmitir informações específicas de configuração, e o repositório de política e interface de gerenciamento web.

A função do sistema em nuvem é de fornecer escalabilidade e recursos virtuais, e sua arquitetura é dividida em três camadas: SaaS, PaaS e IaaS, de forma a prover serviços multimídia, como VoIP e streaming de vídeo.

6.2. Nuvem IMS com Mecanismo QoS

Para suportar aplicações multimídia de alta qualidade, este estudo propõe uma nuvem IMS com política de QoS para resolver as questões de heterogeneidade e mobilidade para satisfazer os requisitos do usuário.

Nesta nuvem IMS, a camada IaaS (Infraestrutura como um serviço) é a plataforma de desenvolvimento usada para projetar o sistema IMS, a camada PaaS (Plataforma como um serviço) fornece uma plataforma para os administradores do sistema desenvolverem o *framework Hadoop* necessário para o sistema de arquivos distribuídos e tecnologia *MapReduce*, e a camada SaaS (Software como um serviço)

possibilita usuários a obterem informações e serviços através de sistemas de computação em nuvem.

A figura 8 ilustra o sistema, incluindo as camadas IaaS, PaaS e SaaS.

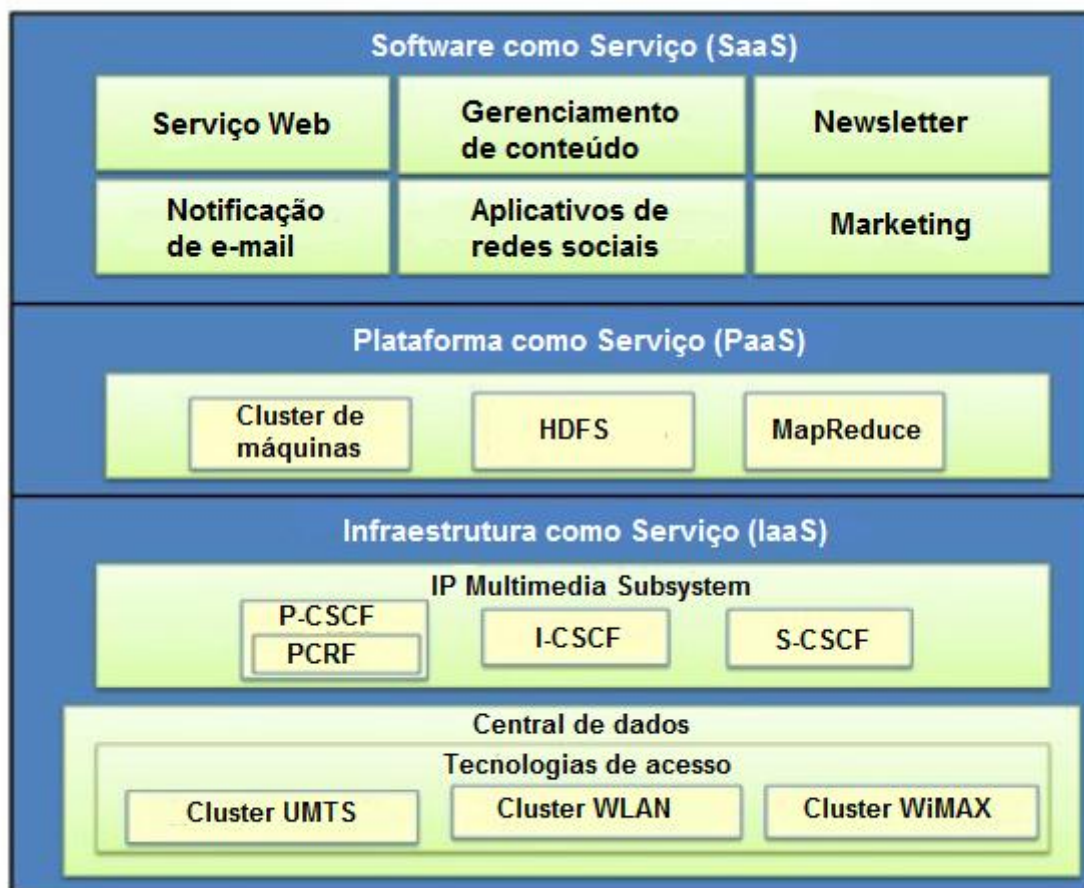


Figura 8 - Modelo do sistema, com SaaS, PaaS e IaaS inclusos [4]

(UMTS = *Universal Mobile Telecommunications System*; WLAN = *Wireless Local Area Network*; WiMAX = *Worldwide Interoperability for Microwave Access*; HDFS = *Hadoop Distributed File System*; PCRF = *Policy Charging and Rules Function*; P-CSCF = *Proxy Call Session Control Function*; I-CSCF = *Interrogating Call Session Control Function*; S-CSCF = *Serving Call Session Control Function*)

O IaaS geralmente acessa redes heterogêneas como UMTS, WLAN e WiMAX, com a finalidade de atingir um serviço de alta qualidade. O PaaS gerencia a nuvem no HDFS (*Hadoop Distributed File System*) e o MapReduce para analisar as preferências

do usuário. O SaaS define os serviços os quais habilitam usuários para recuperar suas aplicações desejadas.

A seguir as especificações das três camadas nesse sistema.

6.2.1. Camada IaaS no sistema

Na camada IaaS (Infraestrutura como um Serviço), o serviço é acessado através de diferentes redes em um ambiente IMS, o qual é implementado em uma plataforma aberta e possui uma arquitetura padrão para fornecer as aplicações multimídia. Além disso, outra função da camada IMS é a de controle do tráfego, analisando o resultado de diferentes acessos dos gateways de rede, e baseando-se nesses resultados, é escolhido o melhor acesso à rede de forma a satisfazer os requerimentos QoS do usuário.

O padrão IMS define explicitamente os parâmetros QoS para negociação entre dois UE's (*User Equipments*) antes de estabelecer uma sessão. A finalidade disso é verificar as conexões entre dois UE's para garantir que os recursos suficientes estão disponíveis, assim quando os parâmetros QoS estiverem confirmados, a rede IMS faz uma requisição à rede do núcleo e obtém o acesso de forma a reservar recursos para a sessão. A negociação entre as duas UE's funciona da seguinte maneira: quando a UE1 envia uma mensagem para a UE2, esta mensagem contém uma requisição de QoS, a qual verifica a UE1 e o nível de subscrição S-CSCF (*Serving – Call Session Control Function*) da UE2. Após essa verificação, a UE2 retorna a proposta QoS, e esta verifica a UE1 e o nível de subscrição S-CSCF da UE2 para identificar os parâmetros QoS baseados nesse nível. Depois disso, o UE1 recebe a proposta QoS da UE2, e estabelece uma sessão ou uma mensagem de atualização do SIP (*Session Initiation Protocol*) necessária para uma nova negociação.

O mecanismo QoS do IMS modifica o estado de eficiência de forma a manter a qualidade da transmissão, e além disso garante a qualidade da transmissão do pacote através de parâmetros de status relacionados à rede, como taxa de transmissão, atraso e taxa de erro.

6.2.2. Camada PaaS no sistema

A camada PaaS fornece uma plataforma para os administradores do sistema desenvolverem o *framework Hadoop* necessário para o sistema de arquivos distribuídos e tecnologia *MapReduce*.

O sistema de computação em nuvem do cluster *Hadoop* é utilizado para fornecer armazenamento de dados em um sistema HDFS, e para a construção desse cluster, são utilizados mecanismos computacionais como o *MapReduce*, um modelo de programação paralela para processamento de dados, e que é utilizado pela *Google* em larga escala nos ambientes de computação distribuída. Uma vantagem da utilização do sistema de arquivos distribuídos do *framework Hadoop* é que sua tolerância a altas falhas possibilita a implantação de dispositivos de hardware por preços baixos.

6.2.3. Camada SaaS no sistema

Essa camada é implementada usando o sistema do *Hadoop* para extrair o que o usuário deseja para ter acesso a informações e aplicações. Os usuários podem, por exemplo, acessar aplicações através de serviços web, como vídeo streaming, gerenciamento de processos e redes sociais.

O SaaS fornece um website o qual permite que administradores façam upload dos dados de suas organizações, executem gerenciamento de serviços, e acessem a interface de usuário de registro. Além disso, o website fornece consoles para monitorar o gerenciamento das imagens de máquina na nuvem.

6.3. Design do Sistema e Análise de Desempenho

O desempenho do mecanismo QoS da nuvem IMS proposta foi avaliado em um sistema de nuvem IMS. A figura 9 ilustra o cenário de teste, no qual foram incluídas tecnologias de acesso 3G, WiMAX e WiFi.

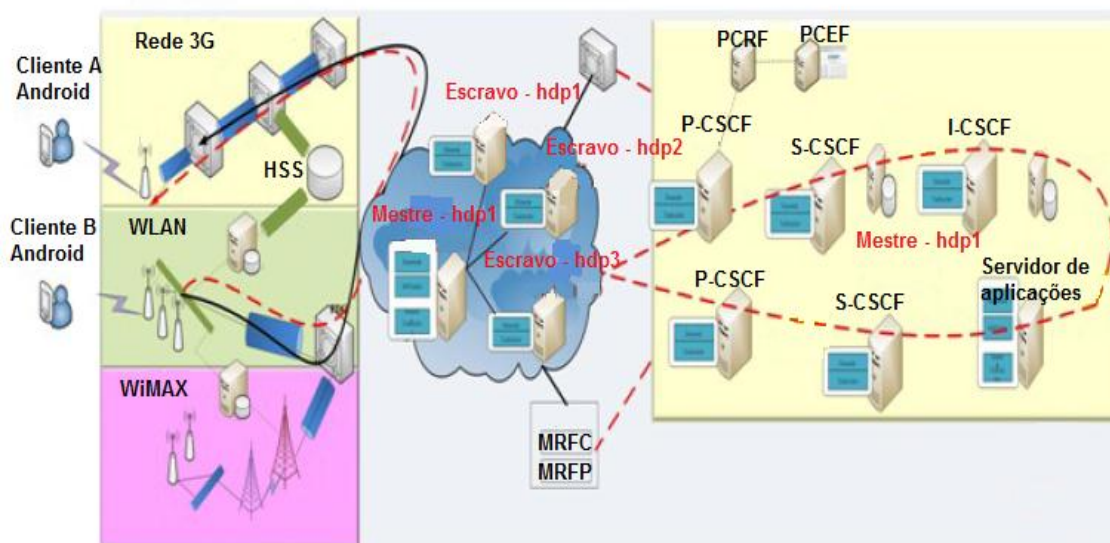


Figura 9 - Cenário de teste para análise de desempenho do sistema proposto [4]

(WLAN = *Wireless Local Area Network*; WiMAX = *Worldwide Interoperability for Microwave Access*; PCRF = *Policy Charging and Rules Function*; PCEF = *Policy Charging and Enforcement Function*; P-CSCF = *Proxy Call Session Control Function*; I-CSCF = *Interrogating Call Session Control Function*; S-CSCF = *Serving Call Session Control Function*; HSS = *Home Subscriber Server*; MRFC = *Multimedia Resource Function Control*; MRFP = *Multimedia Resource Function Processor*)

Neste cenário, o cliente Android acessa a rede núcleo da nuvem para obter o serviço de um fornecedor com VoIP ou tráfego de streaming de vídeo, e o sistema não possui um limite mínimo ou máximo de clientes.

A simulação testada inclui três servidores descritos abaixo:

- hdp-1: possui capacidades da máquina mestre do cluster *Hadoop* e IMS;
- hdp-2: possui capacidade da máquina escravo do cluster *Hadoop*;
- hdp-3: possui capacidade da máquina escravo do cluster *Hadoop*.

6.3.1. Implementação do Sistema

A instalação do sistema *Hadoop* inclui os três servidores citados acima, e a interface do usuário é mostrada na figura 10.

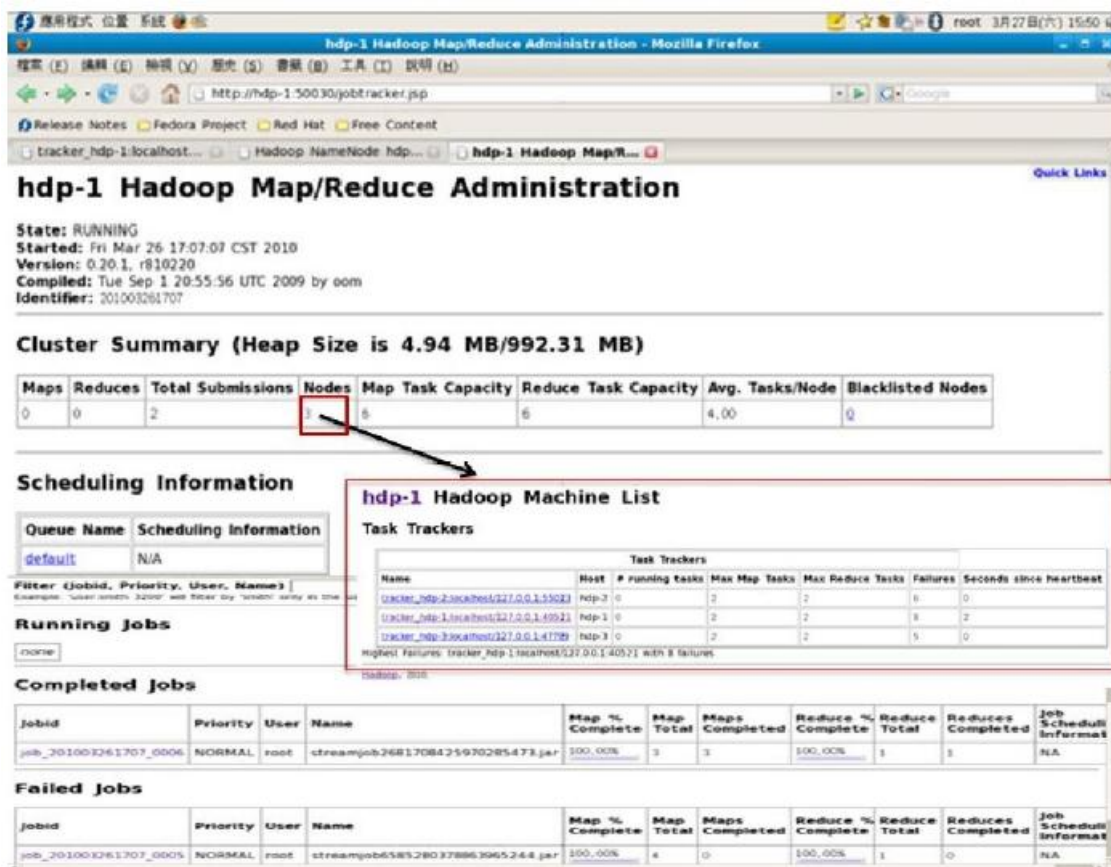


Figura 10 - Interface web do sistema Hadoop [4]

Para obter os serviços do cluster Hadoop, foram utilizados serviços web REST, e as configurações de usuário podem ser alteradas por métodos HTTP “get”, “put”, “delete”.

Quanto a operação do IMS núcleo, a figura 11 ilustra o ambiente IMS estabelecido. Neste sistema proposto, são definidos os componentes IMS na máquina mestre (hdp-1), e os clientes podem comunicar-se utilizando o protocolo SIP para acessar a rede IMS.

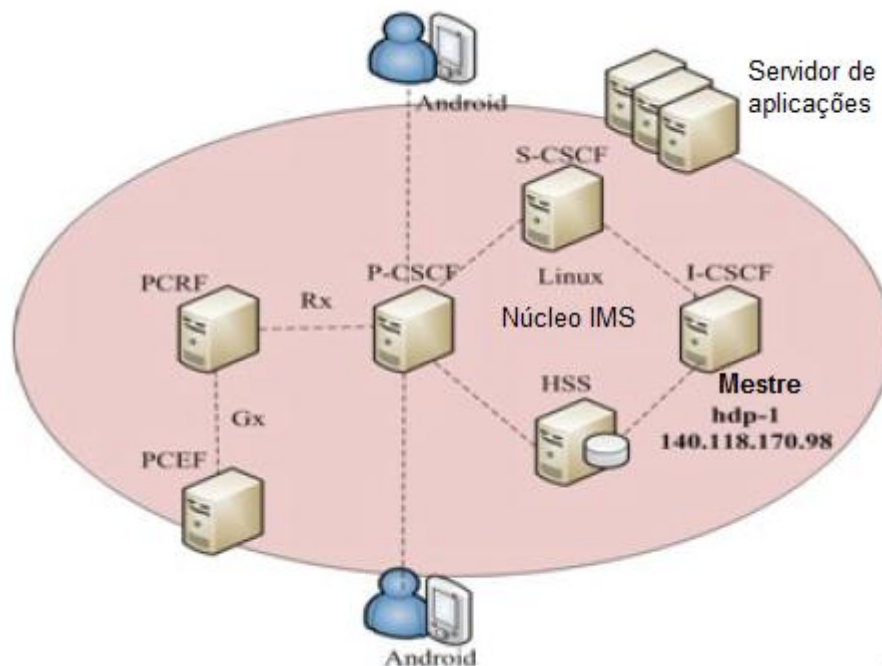


Figura 11 - Ambiente operacional do IMS [4]

(P-CSCF = Proxy Call Session Control Function; I-CSCF = Interrogating Call Session Control Function; S-CSCF = Serving Call Session Control Function; HSS = Home Subscriber Server)

A figura 12 ilustra os quatro módulos CSCF iniciados. O sistema define todas as regras de política do domínio, incluindo autorização de codec, domínio, classe QoS e classe de aplicação.

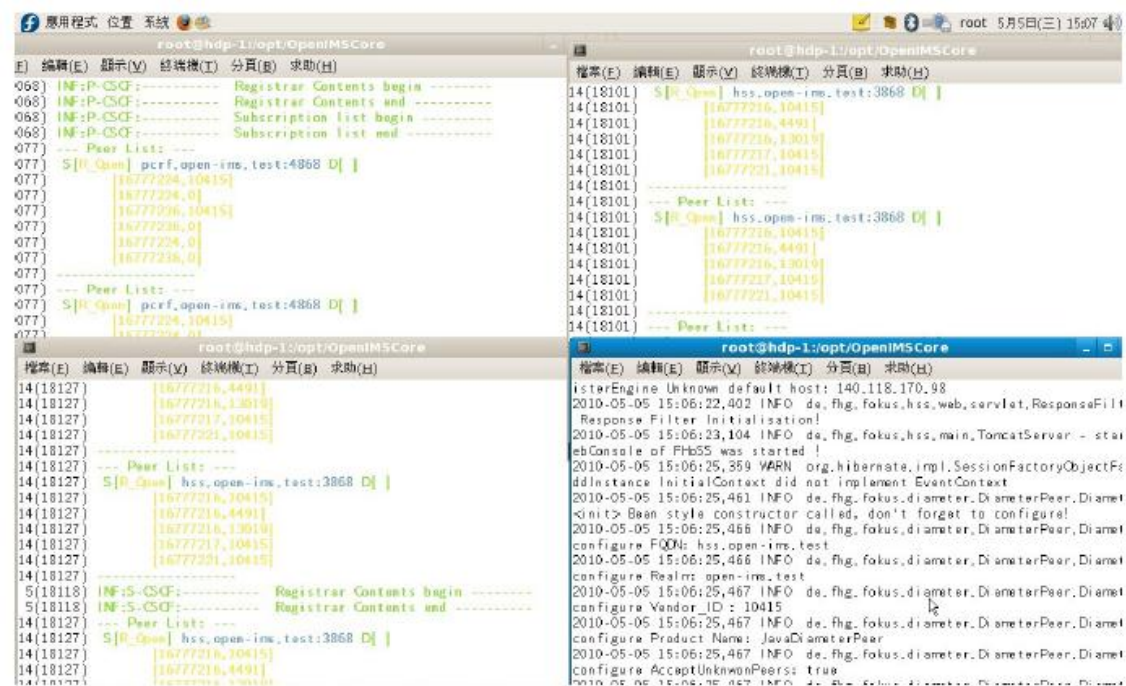


Figura 12 - Módulos CSCF [4]

Usuários podem aproveitar serviços multimídia, como músicas, filmes, serviços meteorológicos, notícias, etc., através de um telefone com a plataforma Android, a qual é utilizada para acessar a nuvem via QR code, e serviços podem ser acessados através da interface da nuvem. Isso é ilustrado na figura 13.



Figura 13 - Telefone com Android acessando serviços da nuvem [4]

6.4. Análise do Desempenho

Os resultados da análise estão descritos nos dois gráficos a seguir. O gráfico da figura 14 compara os resultados com (curva em vermelho) e sem o sistema de computação em nuvem (curva em azul) quando o requisito do cliente está definido para recuperar 128MB. Analisando o gráfico é possível concluir que mais de vinte clientes podem causar sobrecarga do sistema e consequentemente reduzir a capacidade deste, quando não é utilizado o paradigma de computação em nuvem, e setenta clientes quando o mecanismo é usado.

O gráfico da figura 15 ilustra o rendimento do sistema para VoIP, streaming de vídeo e navegador web com o passar do tempo. Para garantir a prioridade do VoIP sobre os demais serviços, o tráfego do navegador web deve ser sacrificado na política proposta. Três aplicações ativam o mecanismo de política QoS, e em torno de 30 segundos, o tráfego do navegador web (curva em vermelho) é reduzido drasticamente, e os rendimentos de VoIP (curva em azul) e streaming de vídeo (curva em verde) aumentam.

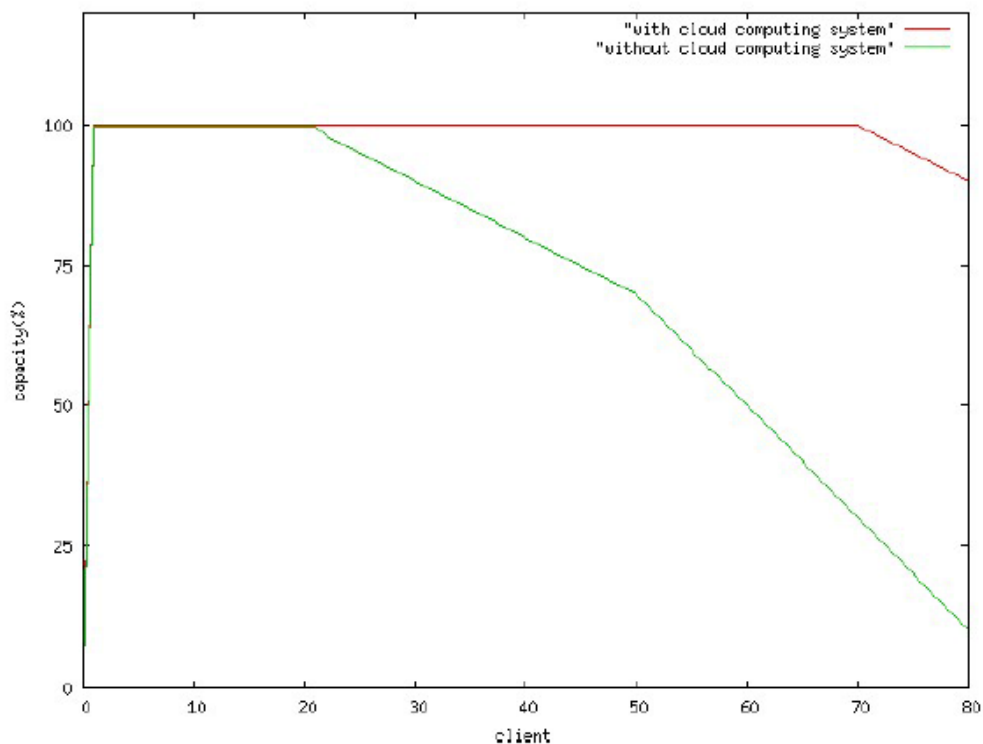


Figura 14 - Gráfico de capacidade do sistema (%) pelo número de clientes [4]

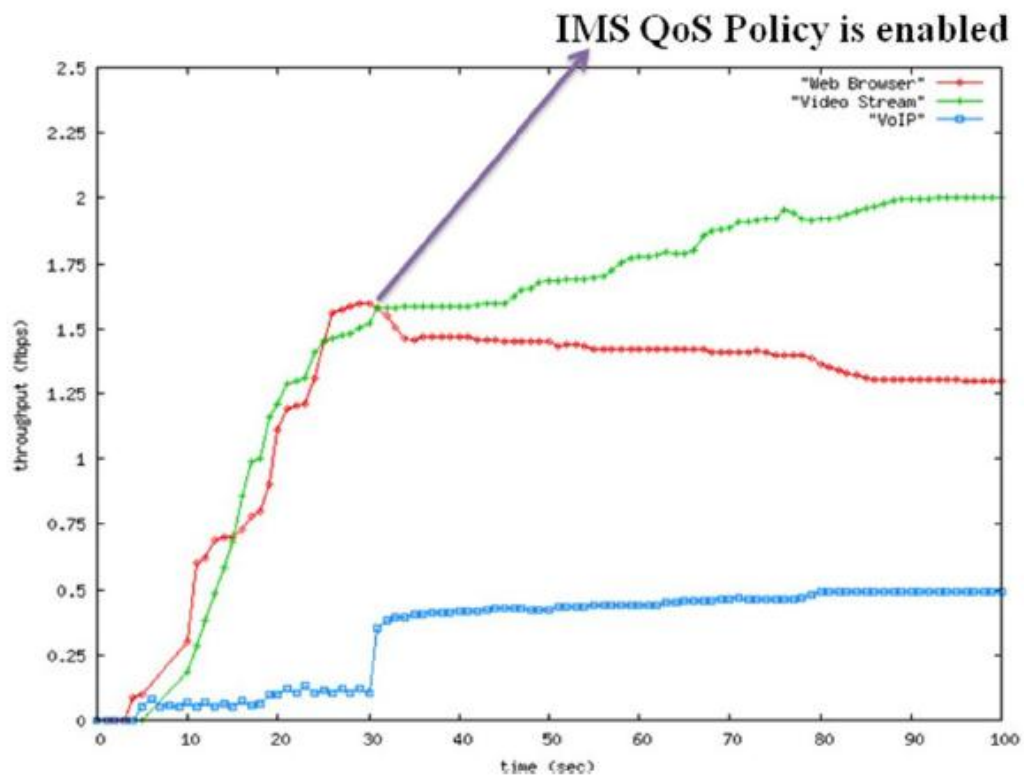


Figura 15 - Gráfico do rendimento (Mb/s) por tempo (s) [4]

7. Conclusões

Esse trabalho propôs o estudo dos princípios de quatro tecnologias: redes heterogêneas, computação em nuvem, *IP Multimedia Subsystem* e Plataforma Android. Foi feita uma revisão bibliográfica desses conceitos, alguns deles já vistos em disciplinas do curso de Engenharia de Computação da USP, nas disciplinas de Serviços de Telecomunicações e Redes de Faixa Larga, e Sistemas Distribuídos.

Após o estudo das tecnologias separadamente, foi realizado um estudo de um mecanismo IMS em nuvem, o qual possui três camadas: IaaS (Infraestrutura como serviço), PaaS (Plataforma como serviço) e SaaS (Software como serviço).

A camada IaaS acessa principalmente redes heterogêneas como UMTS, WLAN e WiMAX através do IMS, a qual utiliza módulo de política QoS para atingir um serviço de alta qualidade, e a rede núcleo como a central de dados da nuvem. A camada PaaS gerencia a nuvem no sistema de arquivos distribuídos *Hadoop* e o mecanismo *MapReduce* para analisar as preferências do usuário. A camada SaaS possibilita aos usuários acessarem aplicações desejadas, como a web, redes sociais e serviços.

Como esperado, a implementação do sistema comprova que sua capacidade de rendimento com computação em nuvem é maior do que sem computação em nuvem, quando vários usuários requisitam serviços.

Referências Bibliográficas

- [1] Tanenbaum, A. S.; “Redes de Computadores”; Editora Campus (Elsevier), 4ª Edição, 2003
- [2] <http://www.jack.eti.br/www/?p=842>; acesso em Outubro de 2012
- [3] <http://shanmg.wordpress.com/3g-4g/>; acesso em Outubro de 2012
- [4] J.-L. Chen et al., “IMS Cloud Computing Architectures for High-Quality Multimedia Applications”, Proceedings of 2011 7th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 1463-1468; Julho de 2011
- [5] James F. Ransome, John W. Rittinghouse, “Cloud computing Implementation, Management, and Security”; CRC Press; 2010
- [6] Bertrand, Gilles, “The IP Multimedia Subsystem in Next Generation Networks”; Maio de 2007
- [7] Santos, Vanessa; Sousa, Aislan; Neto Otílio Paulo, Introdução a Plataforma Android. Disponível em < <http://pt.scribd.com/doc/15981338/Introducao-a-Plataforma-Android>>, acesso em Setembro de 2012
- [8] <http://super.abril.com.br/galerias-fotos/conheca-historia-android-sistema-operacional-mobile-google-688822.shtml#0>, acesso em Setembro de 2012
- [9] <http://www.android.com/whatsnew/>, acesso em Setembro de 2012
- [10] <http://developer.android.com/tools/sdk/eclipse-adt.html>, acesso em Novembro de 2012
- [11] http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm, acesso em Novembro de 2012
- [12] http://www.teleco.com.br/wimax_tecnologia.asp, acesso em Novembro de 2012
- [13] http://hadoop.apache.org/docs/r0.17.1/hdfs_design.html, acesso em Novembro de 2012
- [14] Camarillo, G; Garcia-Martin, M. A; “The 3G IP Multimedia Subsystem”; 2ª Ed; John Wiley & Sons Ltd; 2006
- [15] <http://www.ietf.org/>, acesso em Novembro de 2012

[16] Pedrosa, Paulo H. C.; Nogueira, Tiago; "Computação em Nuvem"; artigo disponível em <http://www.ic.unicamp.br/~ducatte/mo401/1s2011/T2/Artigos/G04-095352-120531-t2.pdf>, acesso em Novembro de 2012

[17] Velte, Toby; Velte, Anthony; Elsenpeter, Robert; "Cloud Computing: A Practical Approach"; McGraw-Hill Osborne Media; 1ª Edição, 2009