

**Universidades Lusíada**

Cardoso, Manuel Maria Pinto do Souto, 1982-

**Web voice communication by phone (IPv6)**

<http://hdl.handle.net/11067/3641>

**Metadados**

<b>Data de Publicação</b>	2013
<b>Resumo</b>	“Web Voice Communication by Phone Over IPv6” é a evolução inevitável das telecomunicações. Esta dissertação de mestrado tem como objectivo fundamental a investigação e demonstração deste tipo de comunicação. Num futuro próximo, as telecomunicações deixarão de ter custo. É exigido pelos consumidores, uma comunicação ou troca de informação com qualidade, rapidez e sem custos. Grande parte das pessoas que já utiliza internet no telemóvel, e já acedem a aplicações principalmente para substituir o S...
<b>Palavras Chave</b>	Telefonia pela Internet, TCP/IP (Protocolo de rede de computador), Sistemas de comunicação sem fios
<b>Tipo</b>	masterThesis
<b>Revisão de Pares</b>	Não
<b>Coleções</b>	[ULL-FCHS] Dissertações

Esta página foi gerada automaticamente em 2024-04-25T05:38:27Z com informação proveniente do Repositório



**UNIVERSIDADE LUSÍADA DE LISBOA**  
Faculdade de Ciências da Economia e da Empresa  
Mestrado em Ciências da Computação

**Web voice communication by phone (IPv6)**

**Realizado por:**  
Manuel Maria Pinto do Souto Cardoso  
**Orientado por:**  
Prof. Doutor Paulo Jorge Gonçalves Pinto

**Constituição do Júri:**

Presidente: Prof. Doutor Mário Caldeira Dias  
Orientador: Prof. Doutor Paulo Jorge Gonçalves Pinto  
Arguente: Prof. Doutor Eng. Joaquim Mesquita da Cunha Viana

Tese aprovada em: 21 Julho de 2014

Lisboa  
2014



**UNIVERSIDADE LUSÍADA DE LISBOA**  
Faculdade de Ciências da Economia e da Empresa  
Mestrado em Ciências da Computação

Web voice communication by phone (IPv6)

Manuel Maria Pinto do Souto Cardoso

Lisboa

Maio 2013



**UNIVERSIDADE LUSÍADA DE LISBOA**  
Faculdade de Ciências da Economia e da Empresa  
Mestrado em Ciências da Computação

Web voice communication by phone (IPv6)

Manuel Maria Pinto do Souto Cardoso

Lisboa

Maio 2013

Manuel Maria Pinto do Souto Cardoso

## Web voice communication by phone (IPv6)

Dissertação apresentada à Faculdade de Ciências da Economia e da Empresa da Universidade Lusíada de Lisboa para a obtenção do grau de Mestre em Ciências da Computação.

Orientador: Prof. Doutor Paulo Jorge Gonçalves Pinto

Lisboa

Maio 2013

## Ficha Técnica

**Autor** Manuel Maria Pinto do Souto Cardoso  
**Orientador** Prof. Doutor Paulo Jorge Gonçalves Pinto  
**Título** Web voice communication by phone (IPv6)  
**Local** Lisboa  
**Ano** 2013

### Mediateca da Universidade Lusíada de Lisboa - Catalogação na Publicação

CARDOSO, Manuel Maria Pinto do Souto, 1982-

Web voice communication by phone (IPv6) / Manuel Maria Pinto do Souto Cardoso ; orientado por Paulo Jorge Gonçalves Pinto. - Lisboa : [s.n.], 2013. - Dissertação de Mestrado em Ciências da Computação, Faculdade de Ciências da Economia e da Empresa da Universidade Lusíada de Lisboa.

I - PINTO, Paulo Jorge Gonçalves, 1956-

#### LCSH

1. Telefonia pela Internet
2. TCP/IP (Protocolo de rede de computador)
3. Sistemas de comunicação sem fios
4. Universidade Lusíada de Lisboa. Faculdade de Ciências da Economia e da Empresa - Teses
5. Teses - Portugal - Lisboa

#### Internet telephony

1. Internet telephony
2. TCP/IP (Computer network protocol)
3. Wireless communication systems
4. Universidade Lusíada de Lisboa. Faculdade de Ciências da Economia e da Empresa - Dissertations
5. Dissertations, Academic - Portugal - Lisbon

#### LCC

1. TK5105.8865.T46 2013

## **APRESENTAÇÃO**

### **Web Voice Communication by Phone (IPv6)**

Manuel Maria Pinto do Souto Cardoso

“Web Voice Communication by Phone Over IPv6” é a evolução inevitável das telecomunicações. Esta dissertação de mestrado tem como objectivo fundamental a investigação e demonstração deste tipo de comunicação.

Num futuro próximo, as telecomunicações deixarão de ter custo. É exigido pelos consumidores, uma comunicação ou troca de informação com qualidade, rapidez e sem custos. Grande parte das pessoas que já utiliza internet no telemóvel, e já acedem a aplicações principalmente para substituir o *Short Message Service (SMS)* também para a utilização do Messenger, Facebook etc.

O presente trabalho sugere a todos os utilizadores de internet principalmente no telemóvel, passem a utilizar a uma aplicação Web Voice Communication by Phone, tendo já acesso à internet pelo telemóvel quer seja por Wi-fi ou outra.

As chamadas demasiado caras impostas pelas operadoras, passavam a não existir. Num futuro em que as comunicações fossem livres para quem tivesse um simples dispositivo móvel e ligação à internet.

**Palavras-chave:** WCbP, IPv6, Wi-fi.

# PRESENTATION

## Web Voice Communication by Phone (IPv6)

Manuel Maria Pinto do Souto Cardoso

“Web Communication by Phone Voice over IPv6”, is an inevitable evolution of telecommunications. The main objective of this thesis is making a research and a demonstration of this type of communication.

In the near future, telecommunications will no longer be a cost. Communication or information exchanges, with quality speed and without cost, are required by consumers. The vast majority of people who already use mobile internet are currently accessing applications primarily to replace the *Short Message Service (SMS)* and for the use of Messenger, Facebook as well.

The present thesis suggests that all Internet users, largely in the mobile phone market are bound to switch to a Web application by Phone Voice Communication and Internet access via mobile phone either by *Wi-Fi* or other

Calls too expensive imposed by suppliers, shouldn't exist anymore, in my personal and professional opinion, in a quite near future communications will become would be free to everyone just by using a simple mobile device and internet connection.

**Keywords:** WCbP, IPv6, Wi-fi.



## LISTA DE ILUSTRAÇÕES

Ilustração 1 - Utilizadores de internet por 100 habitantes.....	12
Ilustração 2 - “ <i>WVCbP</i> ” esquema base.....	17
Ilustração 3 - Modelo genérico “ <i>WVCbP</i> ” .....	19
Ilustração 4 - Exemplo de uma <i>dht</i> .....	25
Ilustração 5 - Esquema <i>wman-wimax</i> .....	28
Ilustração 6 - Aplicações <i>wimax</i> .....	29
Ilustração 7 - <i>Bandwidth vs. Range of wireless systems</i> .....	30
Ilustração 8 - Redes <i>wbl</i> .....	31
Ilustração 9 - Datagram <i>Ipv6</i> com cabeçalhos de extensão.....	38
Ilustração 10 - Estrutura do pacote <i>Ipv6</i> . .....	38
Ilustração 11 - Formato geral do endereço <i>Ipv6</i> . .....	40
Ilustração 12 - Processo de autoconfiguração <i>Ipv6</i> numa máquina. ....	43
Ilustração 13 - Exemplo de uma arquitectura do sistema.....	46
Ilustração 14 - Modelo <i>tcp/ip</i> simplificado.....	49
Ilustração 15 - Esquema da aplicação <i>client1-server-client2</i> .....	51
Ilustração 16 - Mensagem <i>register</i> .....	52
Ilustração 17 - Formato da mensagem de resposta ao <i>login</i> .....	52
Ilustração 18 - Mensagem <i>login</i> .....	53
Ilustração 19 - Mensagem <i>logout client-server</i> .....	53
Ilustração 20 - Diagrama de blocos dos eventos ocorridos na validação de clientes...	54
Ilustração 21 - Mensagem <i>call client1-server</i> .....	54

Ilustração 22 - Mensagem <i>available server-client2</i> .....	55
Ilustração 23 - Mensagem <i>timeout server-client2</i> .....	55
Ilustração 24 - Mensagem <i>accepted server-cliente2</i> .....	56
Ilustração 25 - Mensagem <i>answer client1-server</i> .....	56
Ilustração 26 - Mensagem <i>established client1-client2</i> .....	57
Ilustração 27 - Diagrama de blocos dos eventos ocorridos na ligação entre clientes...	58
Ilustração 28 - Arquitectura de rede <i>skype</i> .....	62
Ilustração 29 - Diagrama de estados do algoritmo de <i>login</i> .....	64
Ilustração 30 - Alguns países onde a comunicação é gratuita.....	69
Ilustração 31 – Sistemas Operativos de smartphones a utilizar <i>voicecheap</i> .....	71
Ilustração 32 - Página de autenticação.....	75
Ilustração 33 - Página inicial.....	75
Ilustração 34 - Página de perfil.....	76
Ilustração 35 - Página de contactos.....	77

## LISTA DE TABELAS

Tabela 1 - Classificação de Sistemas <i>p2p</i> .....	23
Tabela 2 – <i>Wimax Versus LTE</i> .....	32
Tabela 3 - Protocolos Internet ( <i>Tcp/Ip</i> ).....	48

## LISTA DE ABREVIATURAS, SIGLAS E ACRÓNIMOS

- API - Application Programming Interface
- ARP - Address Resolution Protocol
- CAN - Content Addressable Network
- DBMS - Database Management System
- DHCP - Dynamic Host Configuration Protocol
- DHT - Distributed Hash Table
- DNS - Domain Name Server
- DSL - Digital Subscriber Line
- FTP - File Transfer Protocol
- HTTP - Hyper Text Transfer Protocol
- ICMPv4 - Internet Control Message Protocol for IPv4
- ICMPv6 - Internet Control Message Protocol for IPv6
- IGMP - Internet Group Management Protocol
- IP - Internet Protocol
- IPsec - Internet Protocol Security
- IPv4 - Internet Protocol version 4
- IPv6 - Internet Protocol version 6
- ISP - Internet Service Provider
- NAT - Network Address Translation
- NAT - Network Address Translation
- P2P - Peer-to-Peer
- QoS - Quality of Service
- REST - Representational State Transfer
- SHA-1 - Secure Hash Algorithm 1
- SOAP - Simple Object Access Protocol
- SQL - Structured Query Language
- TCP - Transmission Control Protocol
- UID - Unique ID
- URL - Uniform Resource Locator
- VIP - Very Important Person
- WiMAX - Worldwide Interoperability for Microwave Access
- WLAN - Wireless Local Area Network
- WSDL - Web Service Definition Language

XML - Extensible Markup Language  
API - Application Programming Interface  
CAN - Content Addressable Network  
DBMS - Database Management System  
DHT - Distributed Hash Table  
FTP - File Transfer Protocol  
HTTP - Hyper Text Transfer Protocol  
NAT - Network Address Translation  
P2P - Peer-to-Peer  
REST - Representational State Transfer  
SHA-1 - Secure Hash Algorithm 1  
SOAP - Simple Object Access Protocol  
SQL - Structured Query Language  
TCP - Transmission Control Protocol  
UID - Unique ID  
URL - Uniform Resource Locator  
VIP - Very Important Person  
WiMAX - Worldwide Interoperability for Microwave Access  
WSDL - Web Service Definition Language  
LTE - Long Term Evolution  
TDD - Long Term Evolution  
FDD - Long Term Evolution

## SUMÁRIO

1.	INTRODUÇÃO .....	11
1.1.	ENQUADRAMENTO .....	11
1.2.	MOTIVAÇÃO.....	11
1.3.	HISTÓRIA .....	12
1.4.	OBJECTIVOS.....	13
1.5.	TECNOLOGIAS E FERRAMENTAS UTILIZADAS .....	13
2.	METODOLOGIA DA INVESTIGAÇÃO .....	15
3.	ESTADO DA ARTE .....	17
3.1.	TECNOLOGIA WEB VOICE COMMUNICATION BY PHONE.....	17
3.2.	CONCEITOS WVCBP .....	18
3.3	INTRODUÇÃO À TECNOLOGIA VOIP .....	19
3.4	ARQUITECTURA PEER-TO-PEER .....	21
3.4.1	CLASSIFICAÇÃO DE APLICAÇÕES PEER-TO-PEER .....	21
3.4.2	. ESTRUTURA DAS REDES PEER-TO-PEER .....	22
3.4.3	. DHT - DISTRIBUTED HASH TABLE.....	24
4.	WIMAX - WORLDWIDE INTEROPERABILITY FOR MICROWAVE ACCESS.....	27
4.1.	INTRODUÇÃO WIMAX.....	27
4.2.	APLICAÇÕES WIMAX .....	29
4.3.	REDES WBL.....	31
4.4	WIMAX VERSUS LTE .....	32
4.5	SÍNTESE .....	34
5.	IPV6 - PROTOCOLO DA PRÓXIMA GERAÇÃO .....	35
5.1.	INTRODUÇÃO AO IPV6 .....	35
5.2.	CARACTERÍSTICAS DE IPV6.....	36
5.2.1.	CABEÇALHO IPV6.....	38
5.2.2.	ESTRUTURA DO PACOTE IPV6 .....	38
5.3	ENDEREÇAMENTO IPV6.....	39

5.3.1	SINTAXE DO ENDEREÇO IPV6 .....	39
5.4	TIPOS DE ENDEREÇOS IPV6 .....	39
5.5	ESTADO ACTUAL DE DESENVOLVIMENTO IPV6 NO MUNDO .....	41
5.6	IMPLEMENTAÇÕES DE IPV6 .....	42
5.7	PROJECTOS DE IMPLEMENTAÇÃO DE IPV6 .....	43
5.8	SÍNTESE IPV6 .....	44
6.	MODELO - WEB VOICE COMMUNICATION BY PHONE .....	45
6.1.	ARQUITECTURA DO CLIENTE .....	47
6.1.1.	FUNCIONAMENTO GERAL DO CLIENTE .....	47
6.2.	PROTOCOLO DE COMUNICAÇÃO .....	47
7.	TECNOLOGIAS ACTUAIS .....	59
7.1.	SKYPE .....	59
7.1.1.	LOGIN .....	63
7.1.2.	PRIMEIRA LIGAÇÃO .....	65
7.1.3.	SEGURANÇA .....	65
7.1.4.	OUTROS SERVIÇOS .....	66
7.1.5.	SÍNTESE DO SKYPE .....	67
7.2.	MOBILEVOIP .....	68
7.3.	VOIPCHEAP .....	71
8.	DESENVOLVIMENTO DE PÁGINA WEB .....	73
8.1.	IMPLEMENTAÇÃO DA PÁGINA WEB .....	74
9.	CONCLUSÃO .....	79
10.	TRABALHO FUTURO .....	81
	REFERÊNCIAS .....	83
	BIBLIOGRAFIA .....	87
	APÊNDICES .....	89
	APÊNDICE A .....	91









## **1. INTRODUÇÃO**

### **1.1. ENQUADRAMENTO**

Com a rápida evolução das comunicações, sistemas e tecnologias de informação, ao longo dos últimos anos, somos confrontados com diversas mudanças a nível social político e económico. Estimulada por essa constante adaptação às novas tecnologias resulta a globalização, uma realidade que nos coloca perante um acesso quase ilimitado a troca de informação, comunicação e partilha. Verifica-se um sucessivo aumento do poder computacional, em contraste com a diminuição do seu custo, assim como a optimização, aumento da largura de banda nas redes de computadores e capacidade de armazenamento. Essencialmente este crescimento deve-se ao facto das tecnologias de transporte de comutação por pacotes, possibilitarem em simultâneo, o transporte integrado de dados, áudio e vídeo.

A tecnologia apresentada *WVCBP (Web Voice Communication by Phone)* é uma resposta às grandes mudanças que, principalmente as empresas que estão constantemente sujeitas a actualizações na área das comunicações. Tendo como base *VoIP*, Voz sobre IP, não implica a cobrança por impulsos, gerando assim grande interesse em termos económicos, ao contrário das operadoras dos telefones convencionais.

A utilização global da internet como comunicação de dados de voz e imagem poderá caracterizar-se como uma evolução face à situação actual que separa utilizadores de telefone fixo e móvel, onde os utilizadores do *WVCBP* poderão comunicar com outros utilizadores, ligados à Internet.

### **1.2. MOTIVAÇÃO**

A demonstração e investigação do tema apresentado é o resultado de tecnologias já existentes hoje em dia, inovando no sentido de causar uma mudança profunda no mundo das telecomunicações, através desta tecnologia *WVCBP* o utilizador terá acesso a troca de informação e dados de forma muito mais económica e prática.

A tecnologia *VoIP* ou o *Skype*, sendo estas *Freenets* (redes livres), poderão ser utilizadas em telefones móveis, através do desenvolvimento de uma estrutura sólida e de livre troca de informação, revolucionando um tipo de produto e serviço novo sem a

necessidade de recorrer a uma grande operadora móvel para a comunicação ou troca de dados, apenas simples ligação à internet.

Outra motivação da dissertação é demonstrar as vantagens, o baixo custo e fácil utilização do Sistema desenvolvido (WVCBP) e ainda a adaptação a *IPv6* indispensável para o futuro das comunicações.

### 1.3. HISTÓRIA

As comunicações ao longo do tempo têm sofrido grandes alterações. Desde os primórdios do telefone inventado por Alexander Graham Bell, que o mundo tem-se adaptado a novas tecnologias, com uma evolução exponencial. O lançamento da Internet em 1969, ainda chamada na altura ARPANET, foi um marco importante para a história das comunicações da humanidade.

Em 1980 surge a primeira geração de telemóvel analógico (1G) actualmente já atingimos a quarta geração de telemóveis (4G) já baseada em IP colocando os dispositivos em rede, utilizando banda larga, que permite acesso mais rápido à rede. Este tipo de tecnologia associa-se a muitas outras, as mais importantes WiMAX e LTE, por serem tecnologias que disponibilizam acesso à internet a mais de 30 Km com grande velocidade.

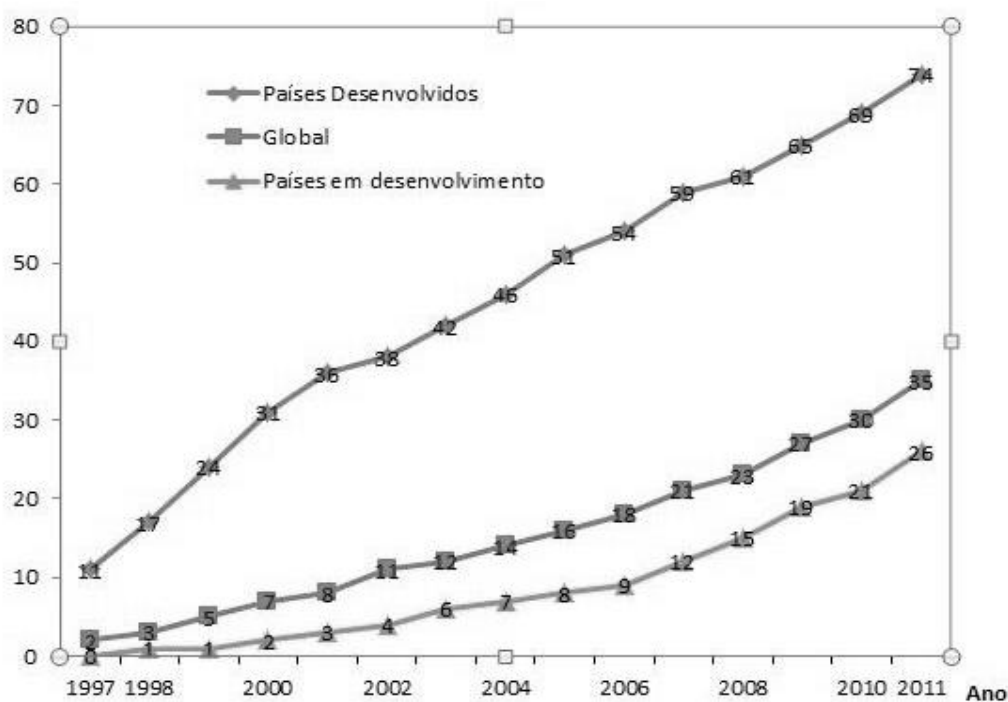


Ilustração 1 – “Utilizadores de Internet por 100 habitantes”, ITU: Committed to connecting the world (Junho de 2011)

## 1.4. OBJECTIVOS

O objectivo fundamental desta dissertação é apresentar e investigar a estrutura de um modelo de comunicação através de dispositivos ligados entre si utilizando a técnica P2P (peer-to-peer) e através da rede de internet global captada por meios e tecnologias que se adequam às necessidades propostas, como a captação a longa distância, rapidez de transmissão de dados e rápida configuração para acesso.

Permitindo assim implementar uma nova aplicação, aproveitando as funcionalidades da tecnologia já existente, como base tecnologia VoIP adaptando-a ao protocolo da nova geração, o IPv6 (*Internet Protocol version 6*).

## 1.5. TECNOLOGIAS E FERRAMENTAS UTILIZADAS

Para implementar o modelo *Web Voice Communication by Phone* apresentado e dissertado, foi desenvolvido um Software programável baseado em C#. Foram utilizadas várias tecnologias e ferramentas, fazem parte do modelo de investigação científico, descrito com mais detalhe no capítulo 2:

- Tecnologias utilizadas:
  - Modelo de redes *Wireless* de Banda Larga (Wi-max) – tecnologia utilizada para acesso internet, dando uma projecção global ao modelo apresentado;
  - Tecnologia *Voip* (*Voice Over Internet Protocol*) – tecnologia que abre portas à vertente comercial do projecto, com a transmissão de dados seja por voz vídeo ou texto, sobre o protocolo *Internet Protocol* a tecnologia VoIP foi desenvolvida para reduzir custos na transmissão de dados, reduzindo assim o custo de chamadas taxadas ao minuto.
  - Tecnologia *P2P* (*Peer to Peer*) - desenvolvido com o intuito de aprofundar o conhecimento das características e técnicas inerentes ao *P2P*, aplicando-as no âmbito do sistema a desenvolver. É igualmente realizado um estudo sobre alguns sistemas e tipos de comunicação, cujo conhecimento poderá ser útil à aplicação WVCbP. Além de dar a conhecer a própria tecnologia a ser usada, este estudo permite ter uma noção sobre o impacto que o sistema poderá causar na comunidade *P2P*.
  - IPv6 - Protocolo da Próxima Geração - É apresentada nesta dissertação a próxima versão do protocolo da Internet, chamado *Internet Protocol Version*

6 (IPv6). Útil para o projecto apresentado, dando uma projecção no futuro das telecomunicações, explicado no capítulo 5

- Ferramentas técnicas utilizadas:
  - Linguagem *C#* - Foi adoptada esta linguagem por se tratar de uma ferramenta útil neste tipo de projectos assim como já conhecida no âmbito do mestrado.
  - *Visual Studio 2010* – Ambiente de desenvolvimento para aplicações *C#*
  - Biblioteca *System.Net.Sockets* – Biblioteca utilizada no desenvolvimento de Sockets UDP (*User Datagram Protocol*) descrito em detalhe no capítulo 7
  - Ambiente Windows – Desenvolvido neste ambiente também com intuito de realização de testes em *Windows Phone*

## 2. METODOLOGIA DA INVESTIGAÇÃO

O método científico de investigação escolhido para a elaboração da dissertação, foi o método construtivista, sustentado pelo método crítico, baseando-se na observação crítica dos acontecimentos, o método experimental, fundamentado na experimentação directa da ideia e por fim o método demonstrativo destacando as ferramentas e técnicas utilizadas para desenvolvimento da tese.

A área de pesquisa para esta tese abrange a vasta área de engenharia informática destacando-se telecomunicações, redes e programação. A principal área de investigação cai no campo da pesquisa construtiva desde que começou com a observação de estudos de caso e a partir daí uma nova teoria foi construída.

A abordagem para o método que permite que a teoria a ser demonstrada é uma técnica de investigação científica típica baseada em várias etapas:

- Base na observação de casos de estudo, foi feita a definição de objectivos e a questão a investigar;
- A revisão e estudo de literatura geral especializada foi feita, para descobrir se a investigação era realmente inovadora ou se já havia muitas investigações sobre o assunto, e também para fornecer alguma base de fundo sobre o assunto para o investigador;
- O modelo que apoia a teoria foi abordado e investigado para explicar toda a teoria/ideia do estudo.
- Por último, uma etapa de utilização prática da teoria, provando que, não só é útil, mas também tecnicamente possível.

Após o modelo definido, foi necessário para a sua aplicação prática a implementação do protótipo/modelo.

O desenvolvimento deste sistema baseia-se na implementação de um conjunto de ferramentas já existentes de forma a cumprir os objectivos propostos. Neste sentido é apresentada a metodologia de estudo de soluções e ferramentas que poderão ser adequadas ao sistema.

A escolha e estudo dessas ferramentas passam pelas diferentes partes que compõem o sistema. Após definição dos objectivos, é feita a estruturação das diferentes partes da entidade central que seguem a seguinte ordem:

- Definição da estrutura geral do sistema;
- Definição e estudo do modelo;
- Definição e implementação do protocolo de comunicação;
- Implementação do modelo proposto;
- Implementação da página *web*;
- Simulações de teste às funcionalidades implementadas em todo o sistema.

O desenvolvimento de cada uma das partes acima descritas foi acompanhado por testes separados antes de avançar para as fases seguintes.

Finalmente, as conclusões foram tomadas e, com base nessas conclusões metas de trabalho foram definidas.



### 3. ESTADO DA ARTE

Neste capítulo é estudada a tecnologia *WVCbP*, assim como conceitos de sistemas *WVCbP* que fazem parte do sistema desenvolvido, com o intuito de aprofundar o conhecimento das características e técnicas inerentes ao *WVCbP*, aplicando-as no âmbito do sistema a desenvolver. É igualmente realizado um estudo sobre alguns sistemas e tipos de comunicações, cujo conhecimento poderá ser útil. Além de dar a conhecer a tecnologia a ser utilizada, este estudo permite ter uma noção sobre o impacto que o sistema poderá causar nas comunidades *Voip* e *P2P*.

#### 3.1. TECNOLOGIA WEB VOICE COMMUNICATION BY PHONE

Uma Comunicação não é mais do que a troca de informação/dados entre pessoas ou máquinas, este processo exige uma intermediação técnica adaptável ao meio e às circunstâncias em que se encontram os interlocutores.

Através de voz podemos interagir com outra pessoa, estando à nossa frente ou no outro lado do mundo, basta ter o mecanismo certo, uma tecnologia de qualidade, rápida e a baixo custo. *Web Voice Communication by Phone* poderá ser o futuro das comunicações por voz. Continuando o desenvolvimento de outras aplicações associadas à internet e telemóveis, como por exemplo o “Skype”.



Ilustração 2 – “WVCbP” Esquema base. (Ilustração nossa, 2012)

A tecnologia *Web Voice Communication by Phone* conjuga o facto de através do telemóvel, ter acesso à internet e a várias aplicações dedicadas a diversos tipos de

actividades e serviços. A tecnologia apresentada tem a capacidade de realizar chamadas directamente para outro dispositivo com o mesmo *software*.

As ligações feitas utilizam apenas a troca de dados, não tendo de utilizar linha telefónica ou qualquer operadora móvel, apenas largura de banda captada por uma de várias tecnologias de emissão de sinal *Wireless*.

### 3.2. CONCEITOS *WVCbP*

*WVCbP* é baseada na utilização da tecnologia *VoIP* que tem crescido com grande expressão na última década. A justificação reside no baixo custo para o utilizador final.

Para conceptualizar *WVCbP* são resumidos os seguintes requisitos:

- **Aplicação *WVCbP*** – Aplicação proposta como base de todo o sistema a implementar.
- **Dispositivos** – Normalmente dispositivos móveis, com a aplicação *WVCbP* instalada previamente, com capacidade para suportar tecnologias como 3G *Voip*.
- **Ligação à internet** – É apresentada a ligação WiMax para acesso à internet a longa distancia.

É apresentado um modelo genérico:

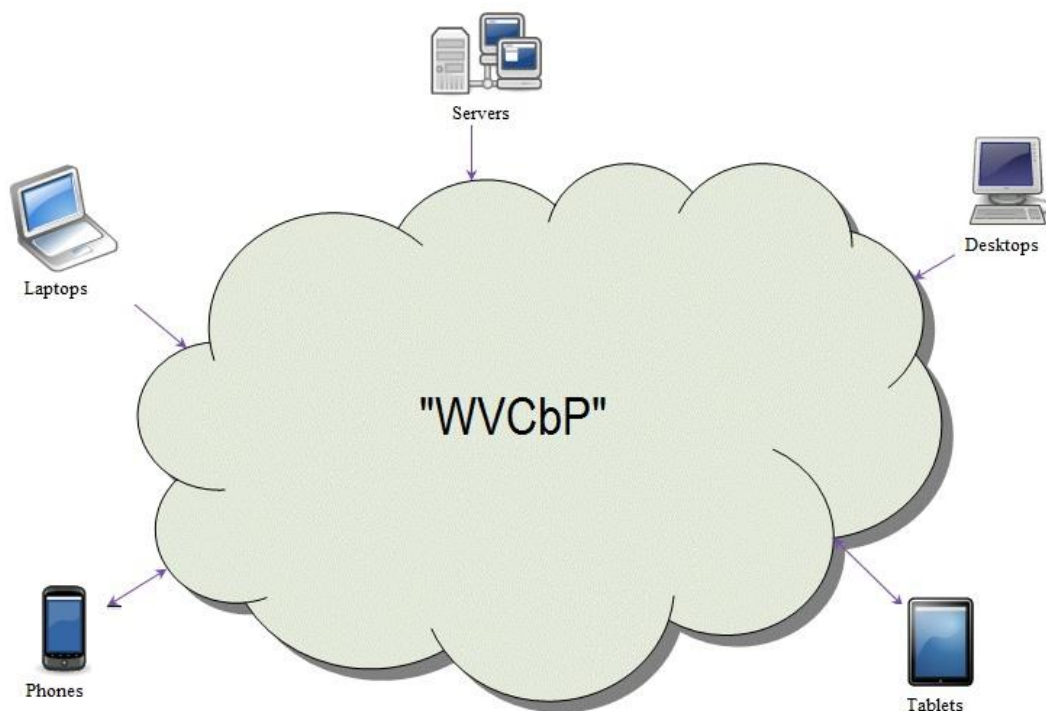


Ilustração 3 - Modelo genérico "WVCbP". (Ilustração nossa, 2012)

### 3.3 INTRODUÇÃO À TECNOLOGIA VOIP

A tecnologia VoIP consiste na transmissão de voz sobre o protocolo IP, Internet Protocol. Esta tecnologia foi desenvolvida a fim de reduzir os custos das chamadas telefônicas.

As chamadas VoIP não são cobradas por impulsos, por esta razão os custos das chamadas diminuem, sendo apenas necessário o pagamento mensal da ligação à rede. Com isso não existe qualquer cobrança, para as chamadas a longas distâncias, o que acontece com os telefones convencionais. [Freitas, 2009, online]

Outro factor que faz com que os custos associados às chamadas VoIP sejam reduzidos, em comparação com o telefone convencional, é a partilha dos canais de comunicação. Pois numa rede IP, o mesmo meio físico é partilhado por várias ligações. Por outro lado, nos telefones convencionais dois canais são reservados para cada conversação, independente da existência ou não de tráfego de voz nos circuitos, ou seja, mesmo nos momentos de silêncio. [Ganguly, Bhatnagar 2008, online]

Ao cliente VoIP, é atribuído um endereço URL, que poderá ser utilizado por este em qualquer país do mundo, facilitando assim aos utilizadores que o pretendem contactar, utilizando o mesmo endereço URL, para esse efeito, à semelhança do que acontece com os endereços de e-mail.

Pode-se ligar a um cliente VoIP, através de um computador ligado à rede, de um telefone VoIP ou ainda um telefone convencional com um adaptador VoIP.

Actualmente com a existência de vírus na Internet e de “*hackers*”, a segurança na Internet envolve algum risco, comprometendo assim todas as tecnologias, como é o caso do *VoIP*, que utiliza a rede de internet para transporte de dados.

A tecnologia VoIP utiliza alguns sistemas de sinalização de chamada, como são o exemplo dos protocolos *SIP*, *H.323*, *IAX* e o *MGCP*.

O protocolo de sinalização, o *SIP* que inclui voz, vídeo e dados, durante a comunicação nas redes de comutação por pacotes.

Perante os protocolos de sinalização, *SIP* e *H.323*, considera-se que a utilização do protocolo *SIP* possui algumas vantagens face à utilização do protocolo *H.323*.

A sinalização do protocolo *SIP* é mais simples, sendo esta textual utilizando caracteres ASCII, ao passo que a sinalização do *H.323*, é em binário, provocando assim alguns atrasos no envio dos dados.

O que o protocolo *SIP* faz em apenas uma transacção, a primeira versão do *H.323* fazia em quatro ou cinco trocas de mensagens. [Freitas, 2009, online]

A tecnologia VoIP utiliza a rede de comutação por pacotes, não havendo a necessidade de circuitos, o que não acontece com as redes de comutação por circuitos, que é utilizada pelos telefones convencionais. A utilização das redes de

comutação por pacotes, possibilita a tecnologia VoIP, de “empacotar”, os dados, áudio e vídeo, enviando-os em simultâneo.

Perante as vantagens citadas anteriormente, da utilização da tecnologia VoIP, esta permite ainda a partilha de uma mesma rede, como rede de transporte de dados e de sinalização, economizando assim a infra-estrutura e manutenção da rede.

A comutação da rede IP é feita através de *software*, por dispositivos como routers e *switches*, e não através de hardware como na rede pública actual. Por este motivo, a tecnologia VoIP, está sujeita a alguns atrasos ou perdas de pacotes, sendo esta uma das desvantagens [Zhao, Kubiawicz, And Joseph, 2001].

### **3.4 ARQUITECTURA PEER-TO-PEER**

No subcapítulo seguinte é estudada a arquitectura P2P, assim como sistemas P2P semelhantes ao desenvolvido, com o intuito de aprofundar o conhecimento das características e técnicas inerentes ao P2P, aplicando-as no âmbito da presente dissertação. É igualmente realizado um estudo sobre alguns sistemas e tipos de comunicação, cujo conhecimento poderá ser útil. Além de dar a conhecer a tecnologia a ser utilizada, este estudo permite ter uma noção sobre o impacto que o presente sistema poderá causar na comunidade P2P.

Arquitecturas *peer-to-peer* são sistemas distribuídos que consistem em nós interligados capazes de se auto-organizar dentro da topologia da rede com o propósito de partilhar recursos [Carlson And R. Gustafson, 2001, p.347] tais como conteúdo, ciclos de *cpu*, armazenamento e largura de banda. Devem ser capazes de se adaptar a falhas, mantendo uma conectividade e desempenho aceitáveis sem necessitar de um servidor ou autoridade globalmente centralizada.

#### **3.4.1 CLASSIFICAÇÃO DE APLICAÇÕES PEER-TO-PEER**

As arquitecturas P2P devido às suas características referidas acima e extensas funcionalidades permitem classificar diferentes categorias de aplicabilidade [Theotokis and Spinellis, 2004, p. 335]:

- Comunicação e colaboração: Nesta categoria as aplicações têm como principal objectivo estabelecer funcionalidades de comunicação e colaboração entre os

nós, geralmente comunicações em tempo real. Aplicações de *chat* como o *MSN* ou o *Skype* inserem-se nesta categoria;

- Computação Distribuída: Estas aplicações têm como finalidade aproveitar o poder computacional (ciclos de *cpu*) dos vários nós. Tarefas que exigem elevado poder de processamento são distribuídas pela rede obtendo-se assim um maior grau de rapidez de processos e conseqüentemente uma resposta mais rápida.
- Serviços de suporte: Infra-estruturas que disponibilizam serviços de segurança a nível da internet, como protecção à negação de serviço ou vírus [Theotokis and Spinellis, 2004, p. 205];;
- Base de dados: Sistemas distribuídos de base de dados baseados em Infra-estruturas P2P que se comprometem com a procura distribuída das informações de forma eficiente;
- Distribuição de conteúdos: Esta é a categoria em que se insere a maior parte dos sistemas P2P desde a simples partilha de ficheiros até sistemas mais complexos de publicação, armazenamento, e organização segura e eficiente de todo o tipo de conteúdos. Alguns exemplos são o *Napster*, *Kazaa*, *Gnutella*, *Freenet*, *FreeHaven*, entre outros. Esta é a categoria mais relevante e que irá ser discutida em pormenor, uma vez que o sistema desenvolvido foca-se na distribuição e replicação de conteúdos [wikipedia, 2001, online].

### 3.4.2 . ESTRUTURA DAS REDES PEER-TO-PEER

A estrutura das redes peer-to-peer é definida consoante o modo como a rede *overlay* é criada. Através de um modo não determinístico (ad hoc) em que os nós e conteúdos vão sendo adicionados aleatoriamente ou através de regras específicas. Perante estas características é possível dividir a estrutura das redes P2P:

- Desestruturada: A localização do conteúdo é completamente independente da topologia da rede *overlay*. O mecanismo para a pesquisa de conteúdos mais usual neste tipo de redes é o método por inundação. A pesquisa é propagada por todos os nós vizinhos até ser atingindo um valor máximo de *hops-to-live*. Outros tipos de mecanismos mais sofisticados como a utilização de percursos padrão ou através de índices de localização de conteúdos. Estes sistemas são

apropriados para nós passageiros, ou seja, que estão frequentemente a juntar-se e a sair da rede. Exemplos deste tipo de sistemas são o Napster, Gnutella, Kazaa, FreeHaven [Plonka, 2000, online];

- Estruturada: Este tipo de redes surgiu no sentido de colmatar as deficiências de escalabilidade que as redes não estruturadas apresentam. A topologia da rede overlay é estritamente controlada e os conteúdos estão especificamente localizados, para que as pesquisas sejam realizadas mais rapidamente. Existe um mapeamento do conteúdo e a sua localização na forma de uma tabela de encaminhamento para que as pesquisas sejam eficientemente encaminhadas [Theotokis and Spinellis, 2004, p. 335]. Uma das desvantagens das redes estruturadas é que é difícil manter a eficiência do encaminhamento das pesquisas, uma vez que os nós estão constantemente a juntar-se e a sair da rede. Exemplos de sistemas estruturados são o *Chord*, *CAN*, *PAST*, *Kademlia*, *Tapestry* [Stoica, Morris, Karger and Kaashoek, 2001, p. 160].

Tabela 1 - Classificação de alguns sistemas p2p

	Centralização		
	Híbrida	Parcial	Nenhuma
<b>Desestruturada</b>	Napster, Publius	Kazaa, Morpheus, Gnutella, Edutella	Gnutella, FreeHaven
<b>Infra-estruturas estruturadas</b>			Chord, CAN, Pastry, Tapestry
<b>Sistemas estruturados</b>			OceanStore, Scan, PAST, Kademlia

Fonte: The International Journal of Computer and Telecommunications Networking, New York (Vlachos, S. Androutsellis-Theotokis, and D. Spinellis, 2004, pp. 195 - 205)

### 3.4.3 . DHT - DISTRIBUTED HASH TABLE

Uma DHT (*Distributed Hash Table*) é um sistema descentralizado e estruturado de distribuição de dados cujo objectivo é providenciar um serviço de localização baseado em tabelas *hash* [Balakrishnan, Kaashoek, Karger, Morris, and Stoica, 2003 p. 43]. Nesta tabela são armazenados pares (chave, valor) e qualquer nó que participe numa rede DHT pode retornar um determinado valor com recurso à chave associada a esse valor. Uma rede DHT é composta por nós distribuídos que têm a responsabilidade do mapeamento das chaves em valores. Estes nós contêm igualmente informações sobre os outros nós na rede para efeitos de encaminhamento, formando uma rede *overlay*. Estas características permitem que estes sistemas se destaquem pelas seguintes propriedades:

- **Descentralização:** Sistema descentralizado, ou seja, não existe nenhuma entidade central que controle a rede. A rede tem a capacidade de autogestão;
- **Escalabilidade:** Tem a capacidade de funcionamento eficiente mesmo com a entrada de inúmeros nós;
- **Tolerância a falhas:** Sistema tolerante a falhas, por não possuir um servidor central que possa corromper a estrutura da rede, bem como suportar a entrada, saída e falha dos vários nós, sem alterar o funcionamento da rede;
- **Balanceamento de carga:** A distribuição das chaves pelos vários nós permite um melhor desempenho na pesquisa dos dados, assim como uma melhor tolerância às falhas dos nós.

As DHTs usam chaves, tanto para identificar nós como para dados. Essas chaves são compostas por números únicos, gerados pelo algoritmo SHA-1. O SHA-1 (*Secure Hash Algorithm 1*) é uma função *hash* criptográfica, actualmente muito usada, que gera códigos de 160 *bits* a partir de ficheiros com comprimento máximo igual a 1 *bit* [Cannièrre And C. Rechberger, 2006, p. 145]. Para guardar um ficheiro com um determinado nome e conteúdo, é aplicada a função *hash* sobre o nome gerando a chave que fica assim associada ao conteúdo criando assim o par (chave, valor). Este par é então encaminhado ao longo da rede *overlay* até ao nó responsável por essa chave. Para a recuperação do conteúdo, é enviada uma mensagem com o *hash* do nome do ficheiro para a rede, até ser encontrado o nó que o contém. É retornado assim o conteúdo associado a essa chave.



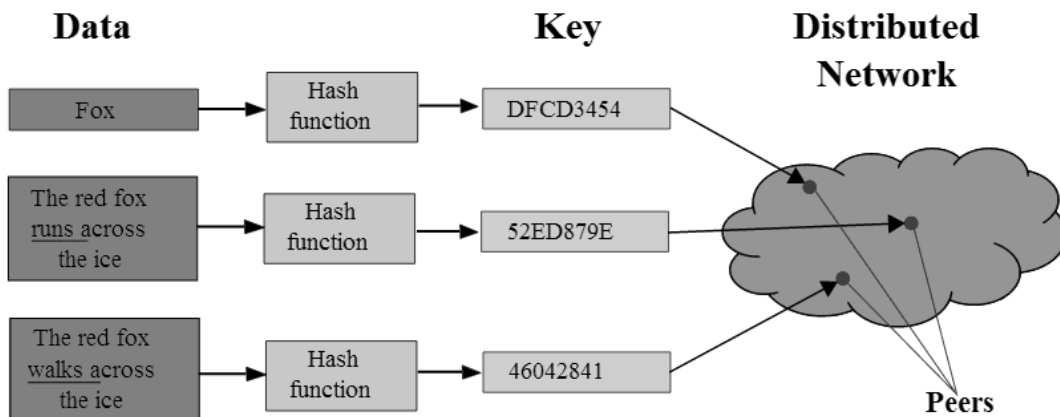


Ilustração 4 - Exemplo de uma DHT, (2007, [http://en.wikipedia.org/wiki/File:DHT\\_en.svg](http://en.wikipedia.org/wiki/File:DHT_en.svg))

A primeira implementação *DHT* foi denominada por *The Beyond Browsers System* e desde aí foram desenvolvidos outros sistemas:

- CAN;
- Chord;
- Pastry;
- Tapestry;
- Kademlia.

Estes sistemas *DHT*, cuja arquitetura é estudada nos capítulos seguintes, tornaram-se importantes infra-estruturas de suporte a conhecidos sistemas *P2P* como o *Bit Torrent*, *eMule*, *Freenet*, *Limewire*, *Vuze* entre outros. [wikipedia, 2012, online].



## 4. WIMAX - WORLDWIDE INTEROPERABILITY FOR MICROWAVE ACCESS

(Redes Wireless de banda Larga)

Neste capítulo é apresentada uma solução *WIMAX* destinada aos utilizadores *WVCbP*, que para acederem a toda a rede de internet ficam condicionados a ligações impostas pelas operadoras móveis com elevado custo de tráfego ou limitadas a ligações *Wireless* com pouco alcance que para a aplicação estudada não são adequadas.

Para ultrapassar este problema, neste capítulo é dada a conhecer uma solução *WIMAX* viável e interessante de analisar. O estudo das características deste sistema, permite o conhecimento de algumas normas e tentará definir e clarificar este conceito fazendo referência a tecnologias que têm enquadramento nas redes *WVCbP*. [Deepak Pareek, 2006]

### 4.1. INTRODUÇÃO WIMAX

A rede *Wireless* de banda larga *WIMAX* está enquadrada numa das áreas de grande progresso em telecomunicações, com maior interesse em crescimento e desenvolvimento, é nesta perspectiva que faz sentido dizer-se que é a próxima geração de sistemas de comunicação entre dispositivos, utilizando internet como rede de comunicação. Estes sistemas acumulam várias tecnologias sem fio, conseguem suportar maior largura de banda comparando com os actuais, capacitando assim um nível superior de comunicação sem interrupções.

A presente necessidade dos utilizadores estarem constantemente ligados à internet, é cada vez maior, a qualquer hora ou lugar. É imperativo que se integre a internet com as comunicações móveis, muito embora se saiba a dificuldade na cobertura a todas as áreas onde haja fraca propagação.

WMAN (Wireless Metropolitan Area Networks) é um termo associado à norma *IEEE 802.16* (geralmente chamada *WIMAX*). Esta tecnologia de banda larga sem fios, é destinada a vários tipos de enquadramentos, como a redução de custos, pois não implica a estrutura de cablagem, ou seja, maior mobilidade como pode ser visto na Ilustração 5

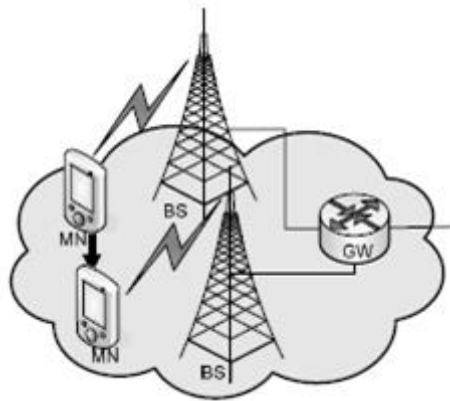


Ilustração 5 – “Esquema WMAN-WiMAX”, (Pareek,2006, p. 52)

O “*IEEE 802.16 WirelessMAN® Standard for Wireless Metropolitan Area Networks*” padrão que define a norma *IEEE 802.16*, foi-lhe atribuído o nome WiMax.

Este padrão é a base da tecnologia *WiMax (Worldwide Interoperability for Microwave Access)*, com grande performance em troca de informação, quando comparado com *WLAN (Wireless Local Area Network)*. A interoperabilidade do WiMax funciona como um sistema *GSM (Global System for Mobile)*, onde existem *BS (Base Stations)* que fazem a transmissão de sinais captados por estruturas próprias via satélite, localizadas perto do cliente. Depois a ligação é feita através de uma simples rede *ethernet LAN (local area network)*.

São apresentadas abaixo algumas características do *WiMAX*:

- Área abrangida até 50 Km;
- Qualidade nas taxas de transmissão a partir de 70 Mbps;
- Modulação Digital com espectro de 5 bits/segundo/Hz;
- Rapidez e qualidade nas operações;
- Plataforma para voz e vídeo;
- Frequências entre 2 GHz e 11GHz (802.16a);

Além destas características é importante frisar o facto de ser também uma tecnologia *non-line-of-sight*, ou seja sem uma linha contínua entre receptor e emissor, podendo assim haver obstáculos superáveis pela transmissão.

## 4.2. APLICAÇÕES WiMAX

Este tipo de tecnologia torna-se muito apelativa sobretudo em ambientes com grande troca de informação, abrindo portas para as mais variadas tecnologias. Nestes ambientes de redes da próxima geração, os utilizadores querem acesso a todos os serviços possíveis, como vídeo e voz. É exigido pelo cliente cada vez maior apoio novos serviços em tempo real com alta qualidade, independentemente da mobilidade dos utilizadores.

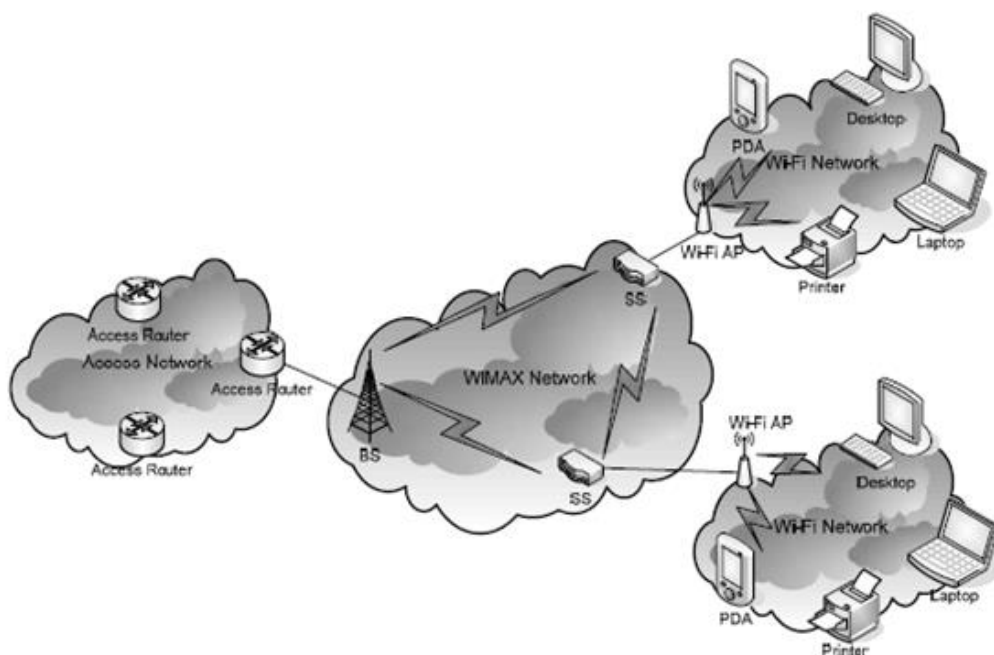


Ilustração 6 – “Aplicações WiMAX”, (Pareek, 2006, p.98)

A tecnologia apresentada apesar de trazer grandes vantagens, comparando com o *wireless* existente, tecnologia como *Wi-Fi* e *UMTS*, não é com o objectivo de competir com estes tipos de tecnologia ou para erradicá-los, mas cooperar com eles. Cada um com características diferentes como se pode ver na Ilustração 7, o *WiMAX* está no meio das tecnologias já existentes, uma vez que suporta uma grande largura de banda assim como um excelente alcance.

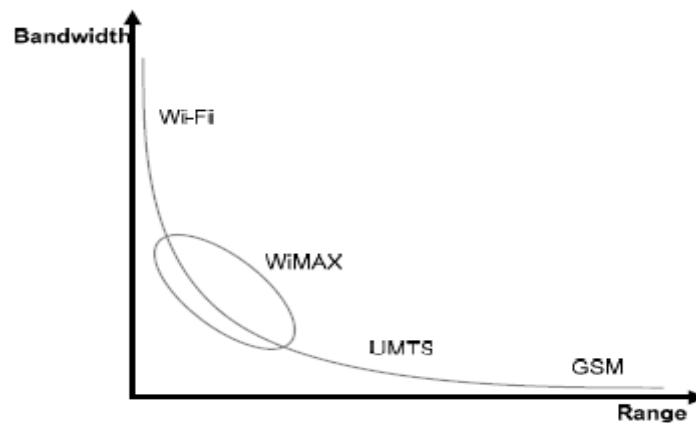


Ilustração 7 – “Bandwidth vs. Range of wireless systems”, (ITU,2011)

Actualmente é já possível, ter mais de uma interface sem fio, para acesso à internet, através de computadores portáteis, *Smartphones*, ou *tablets*, no entanto, no futuro, será muito mais comum ter mais de uma interface, o que permite aos utilizadores estarem ligados à internet permanentemente, aumentando assim a gama de conectividade.

Para evitar interrupção de ligação à rede, é necessário maior largura de banda prejudicando a mobilidade, o WiMax consegue moderar todos estes requisitos e tirar vantagens das interfaces dos equipamentos, tendo sempre em conta, a cada vez maior exigência e necessidade do utilizador final.

O padrão *IEEE 802.21* desde Março de 2004 pretende resolver esse problema. Foram mais de 30 empresas a juntarem-se ao projecto. O primeiro protocolo-padrão, foi definido em Maio de 2005.

O padrão *IEEE 802.21* emergente vai apoiar mecanismos que permitem transferência de dados sem descontinuidades entre as redes do mesmo tipo (*handovers horizontais*), bem como *handovers* entre diferentes tipos de redes (*handovers verticais*). O padrão *IEEE 802.21* disponibiliza informação para permitir a mobilidade essencialmente para telefones móveis, *GSM*, *GPRS*, *Wi-Fi*, *Bluetooth*, *802.11* e *802.16*.

### 4.3. REDES WBL

*Wireless Banda Larga (WBL)* abrange todas as tecnologias *Wireless* que podem contribuir para o objectivo fundamental deste capítulo do trabalho que é, fazer com que haja cobertura a todos os utilizadores *WVCbP*, a cobertura sem fios em questão. É portanto o termo encontrado, para melhor caracterizar o conjunto de tecnologias possíveis para fornecer acesso *Wireless* às mais críticas áreas com difícil acesso à internet.

Pode-se observar na Ilustração 8 o universo das tecnologias *Wireless*: *WPAN* (*Wireless Personal Area Network*) onde se incluem tecnologias como o Bluetooth; depois as normas 802.11 na figura designadas por *WLAN*; *WMAN* normas 802.16; o *WWAN* (*Wireless Wide Area Network*) onde se incluem tecnologias como o *GSM*, *UMTS* e *CDMA* (*Code Division Multiple Access*); e por último a *WRAN*, que é uma tecnologia ainda em investigação, que terá como objectivo fornecer internet a zonas rurais, utilizando o mesmo mecanismo que a distribuição de sinal de TV.

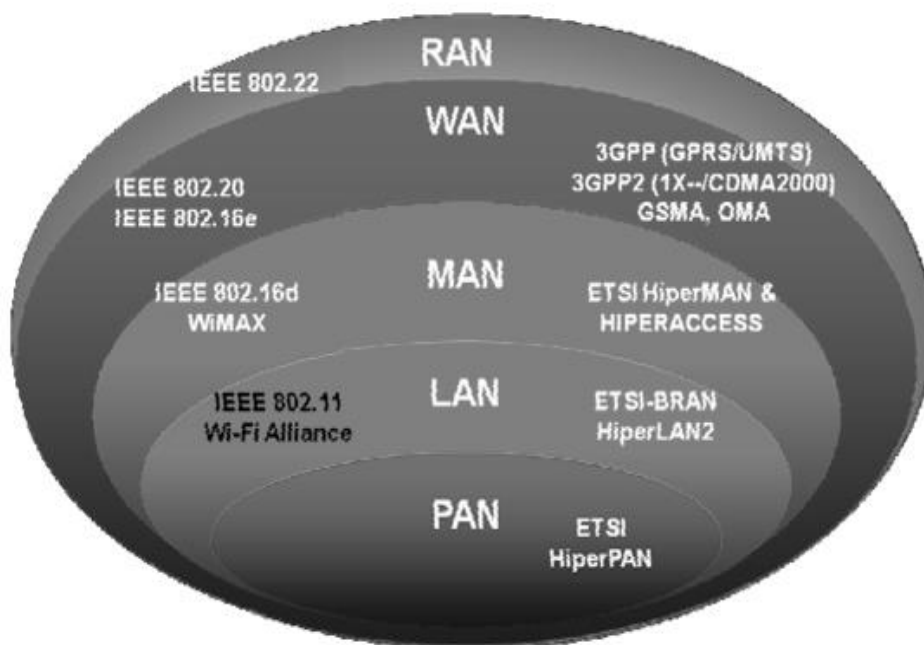


Ilustração 8 –“ REDES WBL”, (Rhoton,2002,p. 33).

## 4.4 WiMAX VERSUS LTE

Os sistemas de telecomunicações *WiMAX* e *LTE (Long - Term Evolution)* são tecnologias muito semelhantes enquadradas num dos objectivos desta dissertação, é feito neste subcapítulo uma comparação pertinente entre as duas redes com expressão a nível mundial. Sendo muito parecidas *WiMAX* e *LTE* existem no entanto algumas particularidades que diferem entre estes sistemas. A Tabela 4.1 apresenta uma comparação entre as principais características do *WiMAX* e do *LTE*.

Tabela 2 – WiMAX versus LTE

	<i>WiMAX</i>	<i>LTE</i>
Arquitectura Rede	Baseado em <i>IP</i>	Baseado em <i>IP</i>
Tecnologia de Acesso	Ligação descendente: <i>OFDM</i>	Ligação descendente: <i>OFDM</i>
	Ligação ascendente: <i>OFDM</i>	Ligação ascendente: <i>SC-DMA (Single Carrier)</i>
Largura Banda Canal [MHz]	1,25; 3,5; 5; 10; 20	1,25; 2,5; 5; 10; 15; 20
Técnicas Duplexagem	<i>TDD</i> e <i>FDD</i> ; Preferencialmente <i>TDD</i>	<i>TDD</i> e <i>FDD</i> ; Preferencialmente <i>FDD</i>
Espaçamento entre sub-portadoras [kHz]	Varia entre 7 a 20 kHz. Valor típico de 10 kHz para <i>WiMAX</i>	15 kHz (Fixo)
	Móvel.	
Tamanho Prefixo Cíclico	Variável: 1/32; 1/16, 1/8, e 1/4	Curto: 5,21  Longo: 16,67
Bandas de Frequência [GHz]	2,3; 2,5; 3,5 GHz	Existentes: 800, 900, 1800, 1900 MHz Novas frequências: Entre 800 MHz a 2,62 GHz
Modulação	<i>QPSK</i> , <i>16-QAM</i> , e <i>64-QAM</i>	<i>QPSK</i> , <i>16-QAM</i> e <i>64-QAM</i>



Débito Binário	Ligação descendente: 75 Mbps	Ligação descendente: 100 a 325 Mbps
	Ligação ascendente: 25 Mbps	Ligação ascendente: 50 a 86,4 Mbps
Raio da Célula	2 a 7 km	5 km
Capacidade da Célula	100 a 200 utilizadores	> 200 utilizadores (até 5 MHz) 400 utilizadores ( superior a > 5 MHz)
MIMO	Ligação descendente: 2x2; 2x4; 4x2; 4x4	Ligação descendente: 2x2; 2x4; 4x2; 4x4
	Ligação ascendente: 1x2; 1x4	Ligação ascendente: 1x2; 1x4; 2x2; 2x4
Mobilidade	Até 120 km/h	Até 350 km/h
Herança de Rede	<i>IEEE 802.16a para IEEE 802.16d</i>	GSM, GPRS, UMTS, HSPA

**Fonte:** ITU: Committed to connecting the world, <http://www.itu.int/ITU-D/ict/statistics/> (2012).

## 4.5 SÍNTESE

Em suma neste capítulo são abrangidas várias soluções para colmatar falhas e necessidades inerentes à utilização da rede, tecnologias de acesso à internet, dando aos utilizadores *WVCbP* e aos diversos tipos de locais, particularmente às zonas com menos capacidades de acesso, tal como por exemplo as zonas rurais. Neste momento a administração Pública está cada vez mais sensível para a importância da utilização da Internet, como forma de desenvolvimento das suas populações, pelo que começam a interessar-se por implementar cada vez mais redes *Wireless*. Supõe-se que as tecnologias emergentes a curto prazo serão o *IEEE 802.11n40* e o *IEEE 802.1641* (*WiMax*), a longo prazo prevê-se o aparecimento das redes *IEEE 802.2242* (*WRAN-Wireless Regional Area Networks*).

## 5. IPV6 - PROTOCOLO DA PRÓXIMA GERAÇÃO

Neste capítulo é apresentada a próxima versão do protocolo de Internet, chamado *Internet Protocol Version 6 (IPv6)*, no enquadramento dos objectivos propostos para a dissertação, em que se apontaria como inovação tecnológica a aplicação *WVCbP* possa ser executada no protocolo IPv6. Com base no protocolo *IPv4*, os investigadores tentaram manter os aspectos fortes e reformular as suas limitações. Consequentemente, IPv6 traz novas funcionalidades, tais como suporte a segurança e mobilidade, suporte melhorado para cabeçalhos de extensão, simplificação do cabeçalho base entre outras, o que é uma evolução da versão anterior.

### 5.1. INTRODUÇÃO AO IPV6

O protocolo IPv6 (*Internet Protocol version 6*) aparece no seguimento do protocolo utilizado actualmente pelos utilizadores com acesso à internet, o IPv4 (*Internet Protocol version 4*). Em 1981 a norma *RFC791* entrou em vigor, e até hoje não sofreu grandes alterações. É de enaltecer o projecto inicial do IPv4 ultrapassando testes de escalabilidade, inicialmente uma rede local, até ao salto exponencial de utilizadores para os dias de hoje.

O aparecimento do IPv6 deve-se essencialmente aos seguintes factores :

- A grande adesão de utilizadores de internet, compromete a distribuição dos *IP's* pelos equipamentos de ligação, devido ao facto da versão ser limitada a gerar  $2 \times 10^{32}$  (4 294 967 296 endereços), que neste momento se aproxima da exaustão levando à utilização do *Network Address Translator (NAT)* para distribuir *IP's* associando apenas um único *IP* público;
- Complexidade na configuração do *IPv4*, é exigido pelo utilizador uma simplificação na distribuição dos *IP's* muitas vezes recorrendo a configuração de endereço *stateful* como o *Dynamic Host Configuration Protocol (DHCP)* ou então feita a distribuição manualmente. Mais à frente iremos ver uma autoconfiguração numa máquina pelo IPv6;
- A segurança é outro factor responsável, pela passagem ao novo protocolo. A maior complexidade do *IPv6* aumenta os níveis de segurança. O protocolo de segurança *IPSec*, não é actualmente suficiente por ser opcional;

- A qualidade de serviço do IPv4 é questionável, sendo muitas vezes posta em causa, a Qualidade de Serviço (QoS): o campo ToS (Type of Service) do IPv4 tem funcionalidade limitada;

[Odson, 2012, online]:

## 5.2. CARACTERÍSTICAS DE IPv6

- **Grande espaço de endereços** - IPv6 tem 128 bits (16 bytes)  $3,4 \times 10^{38}$  endereços IP de origem e destino. O espaço de endereço do IPv6 foi configurado para permitir vários níveis de sub-rede e de endereço da *Internet backbone* para as sub-redes individuais dentro de uma organização. Mesmo com a estimativa mais pessimista feita por especialistas de *IP*, o *IPv6* daria para 1.564 endereços para cada metro quadrado da superfície do planeta.
- **Novo formato de cabeçalho** - O cabeçalho IPv6 tem um novo formato, que é projectado para manter a sobrecarga de cabeçalho a um mínimo. Ou seja, conseguindo mover-se ambos os campos não essenciais e campos opção para cabeçalhos de extensão que são colocados após o cabeçalho IPv6. O cabeçalho IPv6 simplificado fornece processamento mais eficiente em *routers* intermediários. Cabeçalhos IPv4 e IPv6 não são interoperáveis. Um *host* ou *router* deve usar uma implementação do IPv4 e IPv6, a fim de reconhecer e processar os formatos de cabeçalho. O novo cabeçalho IPv6 é apenas duas vezes maior que o cabeçalho IPv4, embora os endereços IPv6 sejam quatro vezes maiores que os endereços IPv4.
- **Maior eficiência hierárquica no endereçamento e routing** - Os endereços *IPv6* globais usados por parte do *IPv6* da Internet são projectados para criar de uma forma eficiente, uma infra-estrutura de encaminhamento hierarquizado, que se baseia na ocorrência de múltiplos níveis de serviços de Internet. Os *routers de backbone*, têm tabelas de routing muito menores, o que corresponde a uma infra-estrutura de routing de nível superior.
- **Configuração Endereço *Stateful* e *Stateless*** – Grande vantagem na configuração de host, o IPv6 suporta a configuração de endereços *stateful* (na presença de um servidor *DHCP*) e ainda a configuração de endereço *stateless* (na ausência de um servidor *DHCP*). Com a configuração de endereços *stateless*, os

dispositivos podem configurar-se automaticamente com os endereços IPv6 (chamados de endereços *link-local*) e com endereços fornecidos por *routers* locais. Mesmo na ausência de um *router*, os dispositivos na mesma ligação podem automaticamente configurar-se com endereços *link-local* e comunicar sem configuração manual.

- **Built-in de segurança** – O suporte para *IPsec* para o protocolo IPv6 é uma exigência. Esta exigência é uma solução baseada em padrões de segurança da rede e promove a interoperabilidade entre diferentes implementações IPv6.
- **Built-in Mobility** – Uma característica do IPv6 é a sua grande mobilidade. O IPv6 tem mecanismos otimizar autoconfiguração IPv6 e a flexibilidade. A é uma grande vantagem face ao IPv4.
- **Suporte Multicast** - Desde o início do projecto, o suporte multicast foi também uma prioridade para a implementação do *IPv6*. Todas as implementações e dispositivos *IPv6* são dimensionados para multicast.
- **Melhor suporte para QoS** – A interacção de QoS com *IPsec* é tratado pela implementação do IPv6, mesmo quando a carga do pacote é encriptado. O protocolo *IPv6*, com os recentes campos do cabeçalho fazem a gestão da transmissão, onde é identificada e gerida. Aumentando assim os níveis de qualidade de tráfego entre ligações e troca de dados.
- **O protocolo Neighbor Discovery** - O IPv6 baseado no *ICMPv6 (Internet Control Message Protocol IPv6)* e a gestão a interacção de nós vizinhos da mesma ligação, a grande vantagem é a possibilidade de fazer detecção dos *hosts* mais próximos e configura-los automaticamente. *Neighbor Discovery* substitui o *Broadcast Address Resolution Protocol (ARP)* pertencente ao IPv4.
- **Extensibilidade** - IPv6 pode ser facilmente estendido para novas funcionalidades, adicionando cabeçalhos de extensão após o cabeçalho IPv6. Ao contrário das opções no cabeçalho IPv4, que só pode suportar 40 bytes de opções, o tamanho dos cabeçalhos de extensão do IPv6 são apenas limitados pelo tamanho do pacote IPv6. [Zhao, Kubiawicz, And Joseph, 2001].

### 5.2.1. CABEÇALHO IPv6

Na ilustração seguinte é apresentado o exemplo de um *datagram* do protocolo *Ipv6*. Este é muito semelhante a um *datagram* IPv4, mas melhorado, mantém as melhores funções de IPv4, afasta ou minimiza as más, ainda acrescenta novas funções quando são necessárias.

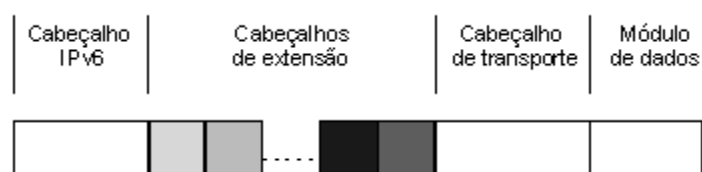


Ilustração 9 – “Datagram IPv6 com cabeçalhos de extensão”,(Calado, 2012)

### 5.2.2. ESTRUTURA DO PACOTE IPv6

A Ilustração 10 apresenta a estrutura do pacote *IPv6* com as seguintes características:

- O cabeçalho IPv6 está sempre presente (40 *bytes*);
- 2 campos de endereço origem e destino utilizam 16 *bytes* cada um;
- 8 *bytes* para informação geral no cabeçalho.

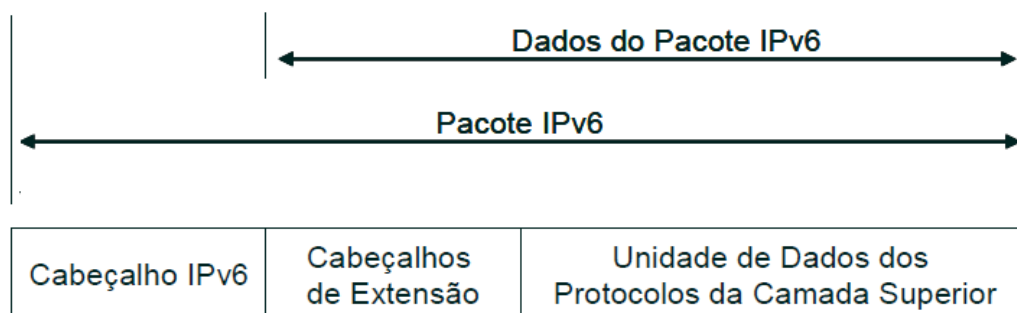


Ilustração 10 – “Estrutura do Pacote IPv6”,(Calado, 2012)

## 5.3 ENDEREÇAMENTO IPV6

O protocolo IPv6 sucessor do IPv4 tem endereços de 128 bits de comprimento, ou seja,  $2^{96}$  vezes o espaço de endereços IPv4. Os  $2^{128}$  ou  $3.4 \times 10^{38}$  endereços a serem atribuídos a máquinas com acesso à internet.

Mudanças muito significativas a nível de tamanho, os novos *datagrams* do protocolo IPv6 dando origem aos *jumbograms* isto é um *datagram* maior do que o limite de tamanho de uma determinada tecnologia, como *udp*. *Jumbograms IPv6* são destinados ao uso futuro e existem apenas em redes privadas para fins de investigação.

### 5.3.1 SINTAXE DO ENDEREÇO IPV6

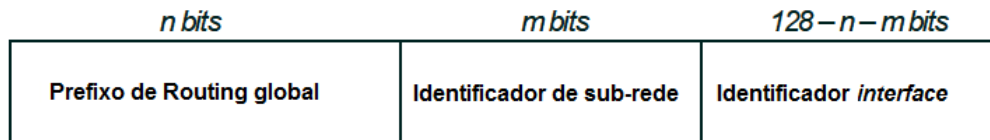
Os endereços IPv4 são representados em notação decimal. O endereço de 128 bits do IPv6 é dividido em limites de 16 bits. Cada um desses limites é convertido num hexadecimal de 4 dígitos (base 16), o número, que é separado por dois pontos. Um exemplo de representação: 2001:1ac8:23b6:05d1:2323:6d6e:0722:5423

## 5.4 TIPOS DE ENDEREÇOS IPV6

A arquitectura de endereçamento IPv6 define 3 tipos diferentes de endereços:

- *Unicast (ponto-a-ponto)*: é um tipo de transmissão de pacotes enviados a partir de uma fonte para outro destino. Em unicast, existe apenas um emissor e um receptor. A maioria das aplicações hoje em dia são Unicast. Exemplo: *HTTP*, *SMTP*, *FTP*, que utilizam o protocolo de transporte *TCP*.
- *Anycast (one-to-one-of-many)*: tipo de transmissão, onde os pacotes são transmitidos por *anycast* para os destinos através do destino mais curto, topologia de routing é ainda uma *policy-based routing*. Exemplo: *BGP (Border Gateway Protocol)*.
- *Multicast (one-to-many)*: é uma tecnologia de transmissão de pacote de dados que é enviada para um grupo específico de receptores do remetente. O protocolo utilizado nesta transmissão é o *UDP* (protocolo de nível baixo)

Na Ilustração 11, um endereço *IPv6* é caracterizado por: prefixo de routing global, o identificador de sub-rede e o identificador *interface*.



**Ilustração 11** – “Formato Geral do Endereço *IPv6*”, (Calado, 2012)



## 5.5 ESTADO ACTUAL DE DESENVOLVIMENTO IPV6 NO MUNDO

Estados Unidos da América: o governo dos EUA criou uma agenda *IPv6*, que identificou várias actividades para a implementação do *IPv6*. Em Dezembro de 2009 o Regulamento de Aquisição Federal dos EUA foi alterado para exigir o uso do perfil *USGv6* e o programa de testes *USGv6* para definir e verificar requisitos *IPv6* em todas as novas aquisições de serviços e estruturas de Tecnologias de informação.

Na União Europeia: Em 2010, a Comissão Europeia estabeleceu uma agenda Digital para a Europa, que dá prioridade ao protocolo *IPv6* em toda a União Europeia e globalmente. A CE tem organizado vários eventos para explorar as questões profundas da implementação do *IPv6*, foi realizada uma conferência em Dezembro de 2010 dedicada ao *IPv6*. Em Junho de 2011, representantes de governos reuniram com membros do corpo técnico e comunidades de negócios a propósito do protocolo *IPv6*. Na Alemanha o governo alemão tem planos para uma rede *IPv6*, uma comunicação entre todos os municípios alemães que já estão a servir como um modelo para outras redes estratégicas do governo.

No Brasil: O *Brasil Internet Registry Nacional (NIC BR)* começou a fazer implementações *IPv6* em 2003. *NIC BR* e o Comité Gestor da Internet no Brasil, (*CGI BR*) liderou um projecto nacional para promover a implementação *IPv6*, trabalhando em colaboração com engenheiros de telecomunicações, num programa de estudo e investigação. Desde que o projecto foi lançado em 2008, o número de utilizadores do sistema no Brasil aumentou quase 15 vezes.

São apresentados alguns dos grupos mais importantes de investigação e implementação no mundo, com o objectivo comum da utilização global do protocolo *IPv6*:

- *AFRINIC - African Network Information Center;*
- *APNIC - Asia-Pacific Network Information Centre;*
- *ARIN - American Registry for Internet Numbers;*
- *LACNIC - Latin American and Caribbean Internet Addresses Registry;*
- *RIPE NCC - Regional Internet Registry (RIR) for Europe, the Middle East Network Coordination Centre*

## 5.6. IMPLEMENTAÇÕES DE IPV6

São apresentadas, algumas implementações importantes no mundo, do protocolo IPv6 com o objectivo de dar a conhecer e acompanhar a evolução do desenvolvimento do protocolo IPv6. Alguns destes exemplos são:

- 6REN: actividade de empreendedorismo norte americana com o objectivo de incentivar a adesão por parte de utilizadores a serviços de rede IPv6, exemplifica redes IPv6 de elevada qualidade;
- 6NET: actividade de empreendedorismo europeia que tem o objectivo de mostrar a evolução e crescimento da Internet, acompanhado pela utilização do protocolo IPv6;
- 6BONE: A missão original do 6bone foi estabelecer uma rede para fomentar o desenvolvimento, teste e implementação de IPv6.

A Ilustração seguinte caracteriza uma implementação de um processo de autoconfiguração numa máquina do protocolo IPv6:

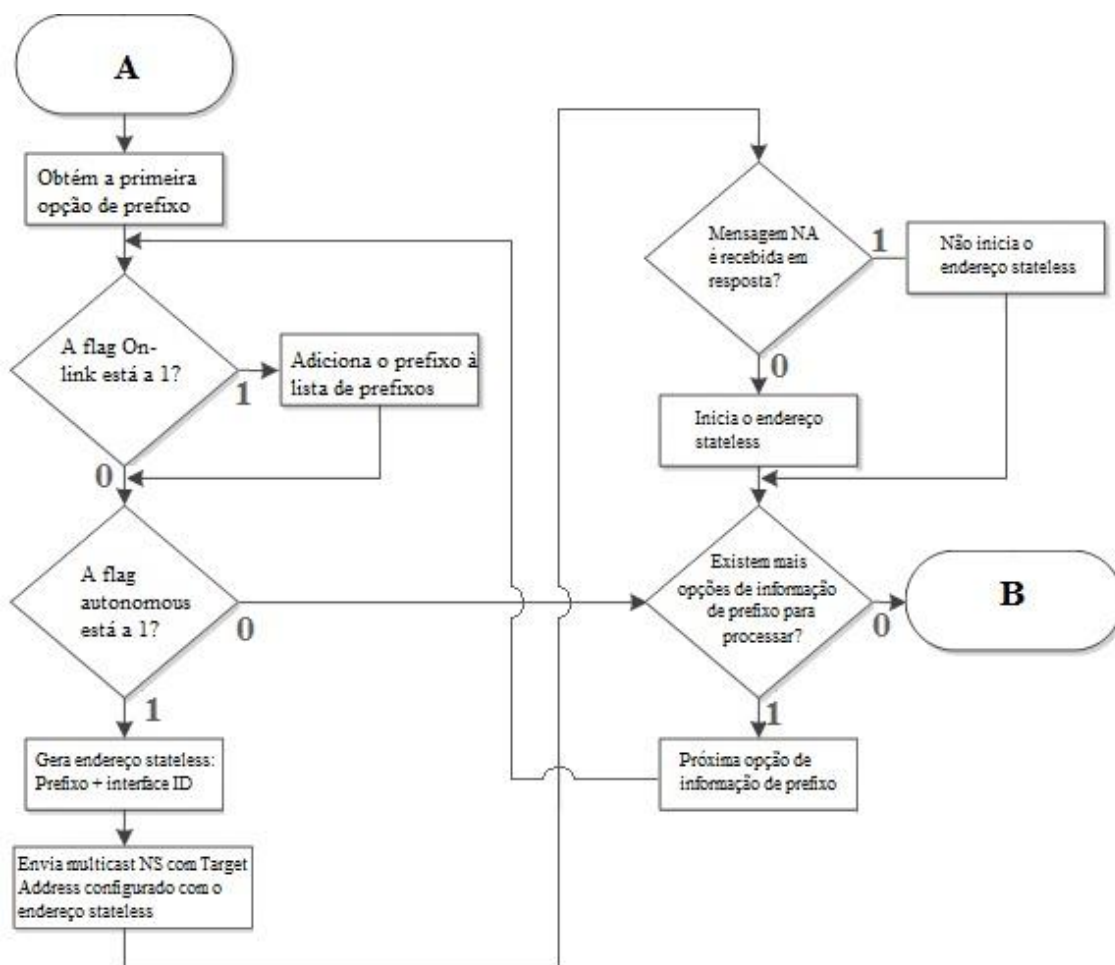


Ilustração 12 – “Processo de Autoconfiguração IPv6 numa Máquina” (Odson, 2012, online)

## 5.7. PROJECTOS DE IMPLEMENTAÇÃO DE IPv6

Estão em curso vários tipos de projectos, por todo o mundo, com o objectivo de implementar o protocolo IPv6 no mercado. São apresentados alguns exemplos deste tipo de projectos:

- *World IPv6 Day* foi um teste técnico e um evento de publicidade em 2011, patrocinado e organizado pela *Internet Society*, vários grandes fornecedores testaram e promoveram ao público a implementação prática do protocolo IPv6;

- O projecto *KAME* foi um esforço conjunto de seis empresas no Japão que visam proporcionar um livre *IPv6* e *IPSec* (para *IPv4* e *IPv6*) pilha de protocolos de implementação de variantes do *BSD Unix*;
- Projecto *USAGI* é voltado para sistemas Linux;
- O Microsoft Windows XP/Vista/7 suportam *IPv6* para fins de desenvolvimento. No Windows XP SP1 (2002) e Windows Server 2003, o *IPv6* está incluído e configurado para a implementação comercial;
- A Apple também já disponibilizou um kit de desenvolvimento com *IPv6* e *IPSec* para Mac OS X.

[Wikipedia.com, 2012, online]

## 5.8. SÍNTESE IPv6

No capítulo 5 são introduzidas algumas das características do protocolo estudado o *IPv6*, comparando-o com o actual protocolo em utilização.

Com a finalidade de fazer face aos objectivos inicialmente propostos na dissertação, foi feita a apresentação de várias implementações e projectos, adequados à tecnologia em estudo, o *WVCbP*. A tecnologia *Web voice communication by phone*, visa ser adaptada a este tipo de protocolo, sendo na área de telecomunicações uma inovação a implementar futuramente.

Comparando com o *IPv4*: a segurança *IPSec* está disponível no *IPv4* e no *IPv6* é obrigatória. A autoconfiguração em DHCP para Hosts e para *IPv6 Serverless*. Endereçamento 32 bits (NAT) enquanto o *IPv6* tem 128 bits de endereçamento

## 6. MODELO - WEB VOICE COMMUNICATION BY PHONE

Neste capítulo é explicada a estrutura geral do sistema desenvolvido assim como a definição das partes envolvidas. O sistema é constituído por duas entidades, cliente e servidor. É composto por um ou mais clientes que formam entre si uma rede de comunicação e uma entidade central, o servidor.

O cliente apresenta a interface ao utilizador, possibilitando o registo e autenticação no sistema, além de permitir explorar grande parte das funcionalidades do mesmo. Tem como principais funcionalidades a comunicação por voz, pesquisa e armazenamento de contactos da rede. Para a informação de contactos estar disponível é imposta uma rede *DHT* com objectivo de pesquisar contactos entre os vários peers. Para partilha e visualização da informação em detalhe dos contactos.

O servidor é a entidade central que cria a rede e oferece aos clientes o serviço de registo e autenticação no sistema permitindo-lhes juntarem-se à rede. Tem como principais funcionalidades estabelecer ligação entre clientes, registo de novos clientes, aprovação de comunicação e a gestão de informação de clientes, através de um modelo dinâmico, com o intuito de garantir disponibilidade, maior acessibilidade e persistência dos mesmos [Bhagwan, Savage And Voelker, 2003]. Além da gestão de todo o sistema, pode funcionar como um cliente normal, realizando funções de comunicação com os próprios clientes.

As funcionalidades da entidade cliente são explicadas resumidamente no próximo capítulo. No capítulo 6.1 é explicada toda a implementação da entidade central assim como todas as suas funcionalidades.

A figura 13 mostra um simples exemplo para melhor compreensão da arquitectura do sistema.

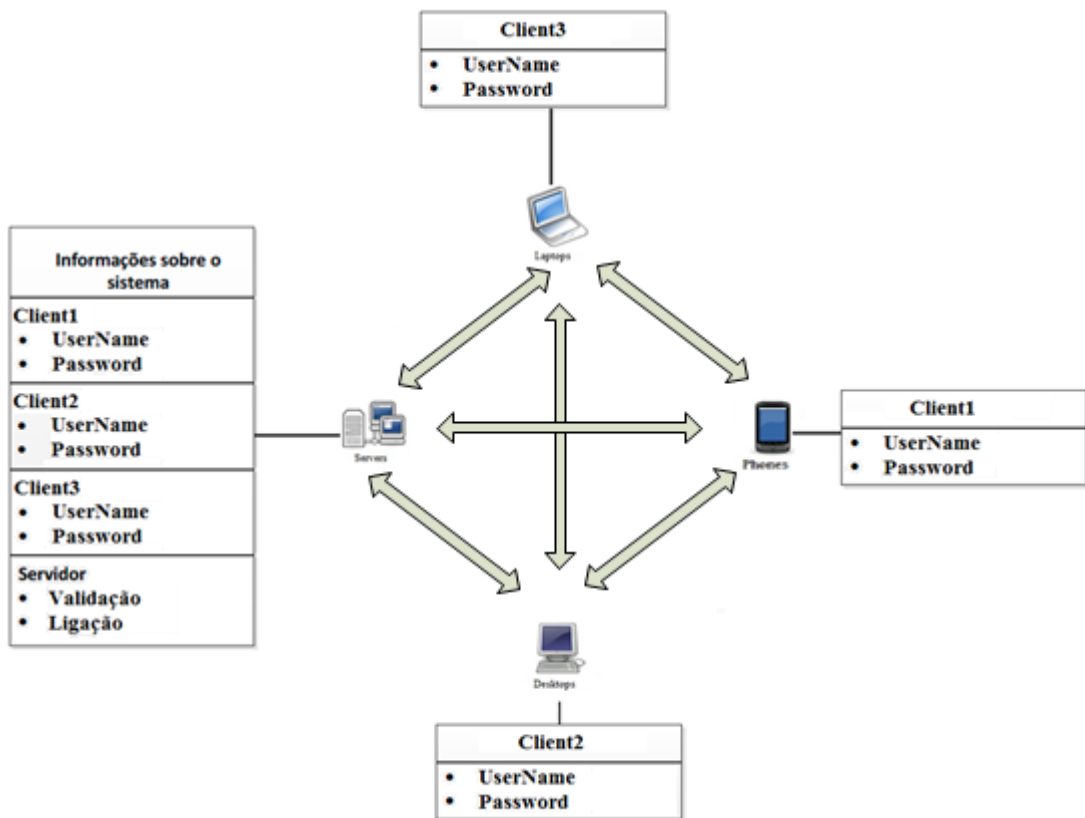


Ilustração 13 – “Exemplo de uma arquitectura do sistema – WVCbP”, (Ilustração nossa, 2012)

Tanto os clientes representados como peers, como o servidor, estão interligados através da rede. Pode existir comunicação entre todos eles desde que depois de feito o registo no servidor, disponham de um “*username*” e uma “*password*”. É possível localizar clientes entre os vários peers. Cada peer comunica com o servidor disponibilizando informações sobre o seu perfil armazenados para efeitos de registo. Ficam disponíveis através de *Web Services*, que utilizam, o protocolo de comunicação HTTP (*Hyper Text Transfer Protocol*). O servidor guarda todas estas informações numa base de dados. Tal como os *peers*, o servidor pode funcionar como um *peer*, podendo comunicar com qualquer cliente.

## **6.1. ARQUITECTURA DO CLIENTE**

O *software* cliente, como referido anteriormente, permite essencialmente a comunicação e pesquisa de clientes por parte dos utilizadores, bem como a comunicação com o servidor, que inclui o serviço de registo/autenticação no sistema e a disponibilização de toda a informação sobre os perfis dos utilizadores. Este software foi desenvolvido recorrendo à linguagem C# e disponibiliza uma interface gráfica para uma fácil interacção do utilizador, no acesso a todas as funcionalidades do sistema. Este capítulo apresenta as funcionalidades e características do cliente. Não são aprofundadas todas as soluções da arquitectura, pois, apesar de ser uma componente essencial no sistema, não é o objectivo deste trabalho o desenvolvimento e estudo.

### **6.1.1. FUNCIONAMENTO GERAL DO CLIENTE**

O cliente é a entidade responsável por disponibilizar uma interface que permite a interacção do utilizador com o sistema. Essa interface deverá permitir o acesso a todas as funcionalidades do sistema, de acordo com os objectivos propostos. Neste sentido, foram implementadas as seguintes funcionalidades:

- Integração do cliente numa rede;
- Registo e autenticação no sistema;
- Comunicação com o servidor;
- Comunicação entre utilizadores;
- Partilha de perfis com base de dados no servidor;
- Pesquisa de perfis de clientes no sistema;
- Interface gráfica;

## **6.2. PROTOCOLO DE COMUNICAÇÃO**

O projecto desenvolvido baseia-se num sistema P2P, gerido de forma centralizada, em que os clientes comunicam com uma entidade central para que esta faça a gestão de todo o sistema. O servidor central oferece serviços de registo, autenticação, gestão de dados de perfis, entre outros, pelo que a comunicação entre cliente e servidor baseia-

se em pedidos/respostas. Seguindo este padrão de troca de mensagens, as ligações entre cliente/servidor podem ser *stateless* ou *stateful*.

Uma ligação *stateless* é uma ligação que é iniciada e terminada entre chamadas. Não é necessário guardar informação da ligação entre o cliente e o servidor. Uma ligação *stateful* é, por sua vez, uma ligação constante que guarda informação da chamada entre um cliente e um servidor [Sheldon, 2001, online].

Os *Web Services* são *web APIs* criadas do lado do servidor cujo objectivo é fornecer serviços ao utilizador. É possível identificar duas grandes classes de *Web Services*: “*Big*” *Web Services* e *RESTful Web Services*. Ambas utilizam como protocolo de comunicação, o HTTP [Pautasso, Zimmermann, And Leymann, 2008, p.805].

Os “*Big*” *Web Services* consistem na troca de mensagens XML (*Extensible Markup Language*) baseadas no protocolo SOAP (*Simple Object Access Protocol*). Nestes sistemas, os serviços são descritos através do protocolo WSDL (*Web Service Definition Language*) e é o tipo de *Web Services* mais utilizado na indústria.

Os *RESTful Web Services*, são *Web Services* implementados sobre HTTP segundo os princípios de REST (*Representational State Transfer*). Não necessitam do protocolo WSDL ou mensagens XML para definição dos serviços.

Tabela 3 - Protocolos Internet (TCP/IP)

Camada	Protocolo
5.Aplicação	HTTP, SMTP, FTP, SSH,Telnet, SIP, RDP, IRC,SNMP, NNTP, POP3, IMAP,BitTorrent, DNS, Ping ...
4.Transporte	TCP, UDP, RTP, SCTP,DCCP ...
3.Redes	IP (IPv4, IPv6) , ARP, RARP,ICMP, IPsec ...
2.Enlace	Ethernet, 802.11 WiFi,IEEE 802.1Q, 802.11g,HDLC, Token ring, FDDI,PPP,Switch ,Frame relay,
1.Física	Modem, RDIS, RS-232,EIA-422, RS-449,Bluetooth, USB, ...

Fonte: ITU: Committed to connecting the world, <http://www.itu.int/ITU-D/ict/statistics> (2012)



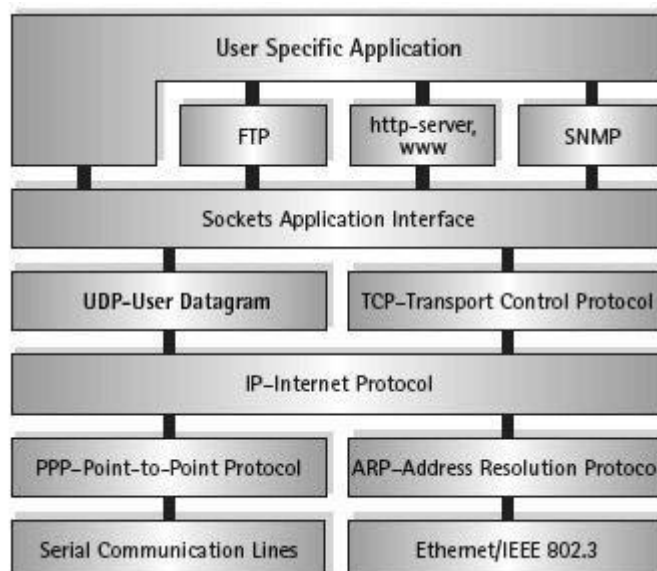


Ilustração 14 – “Modelo TCP/IP simplificado”, (ITU 2012, online)

Outra forma de comunicação entre cliente/servidor é através de *sockets* TCP (*Transmission Control Protocol*). O protocolo TCP baseia-se numa ligação *stateful* através de um *socket* entre o cliente e o servidor [Comer, 2006, p.321].

A escolha para interface de comunicação do sistema poderia recair entre *sockets* TCP e UDP (*User Datagram Protocol*), pois tanto uma como outra possuem vantagens e desvantagens. Os *sockets* TCP, com o estabelecimento de uma ligação contínua, facilitariam a detecção e tratamento de erros ocorridos durante a comunicação, por isso mais fiável garantindo a comunicação entre clientes e servidor.

Algumas características técnicas do TCP:

- Orientado à ligação;
- Ponto a ponto;
- Confiabilidade;
- Entrega ordenada;
- Controle de fluxo;

Por se basearem em ligações *stateful* originariam uma sobrecarga no servidor central se existissem demasiadas ligações activas.

O protocolo utilizado para permitir a ligação entre clientes é o protocolo UDP (*User Datagram Protocol*). Este protocolo, após estudo e testes realizados, revelou-se ser o mais indicado para integração no sistema.

O protocolo UDP faz parte da camada de transporte. A sua simplicidade permite o envio de um *datagram* encriptado em pacotes IPv4 ou IPv6. Ao contrário do protocolo TCP este tipo de protocolo não garante a entrega do pacote enviado, característica típica do protocolo UDP.

Este protocolo UDP por não garantir a sua entrega não é um protocolo fiável. Neste projecto, foram implementados alguns mecanismos de controlo para por exemplo saber se um cliente que se pretende contactar está *online*. Estes mecanismos de controlo utilizados na aplicação do tipo *timeouts* e *acknowledgments* controlam, através de um espaço curto de tempo, se existe resposta de socket a ser contactado. Existe esta necessidade de controlo do protocolo UDP por se tratarem de *datagrams* com registos indivisíveis.

Outra característica do protocolo *UDP*: como não existe obrigação de uma ligação contínua entre clientes e servidor, pode-se dizer que é uma actividade de protocolo sem ligação longa.

Ou seja, na aplicação proposta o cliente *UDP* vai enviar, através de um só *socket* criado, vários *datagram* para diferentes clientes e servidor. Também através da utilização do mesmo *socket* o servidor pode ler vários *datagram* com origem em clientes diferentes.

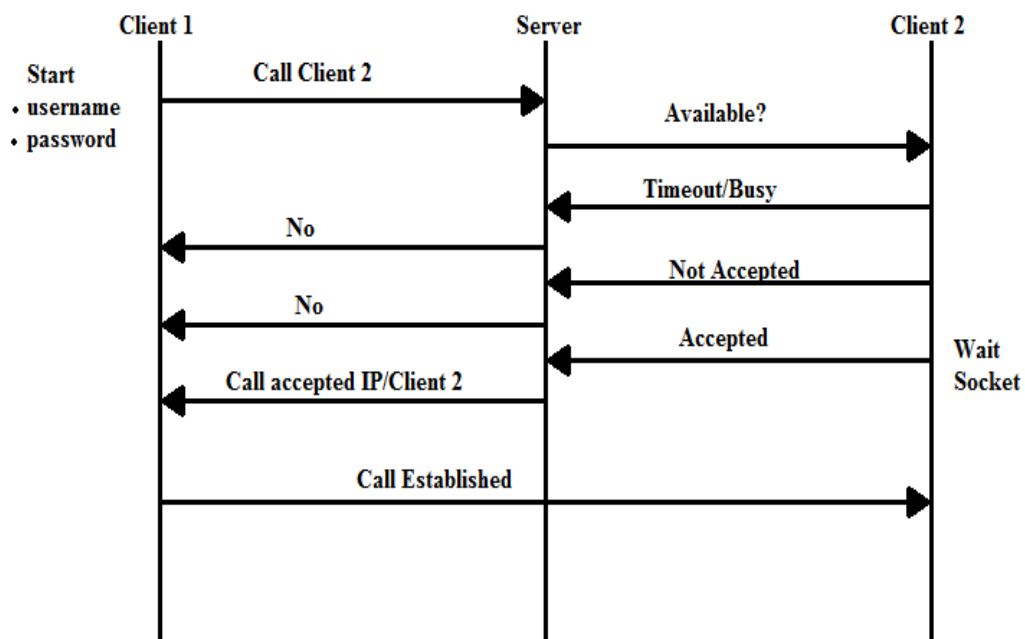


Ilustração 15 – “Esquema da aplicação Client1-Server-Client2”, (Ilustração nossa, 2012)

A implementação do protocolo *UDP* é realizada no servidor permitindo aos clientes a interacção e acesso ao serviço de registo/autenticação. Para o efeito foi definido o seguinte conjunto de mensagens:

- **Register:** Esta mensagem, tal como o nome indica, serve para o utilizador efectuar o registo no sistema. Este envia o *username*, que tem de ser único, a palavra passe, o tipo de utilizador – normal ou VIP, e o código que certifica que o utilizador está regularizado para se registar como utilizador VIP. O servidor confirma se o registo foi bem-sucedido (figura 16).

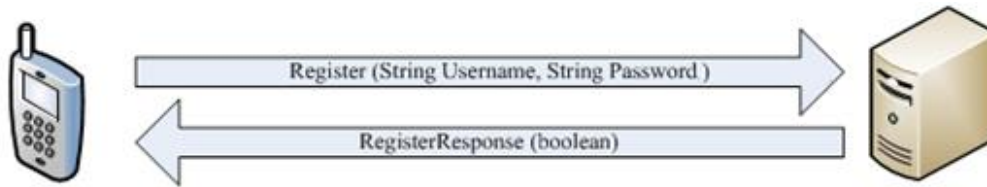


Ilustração 16 - Mensagem *Register*. (Ilustração nossa, 2012)

- **Login:** O cliente envia esta mensagem para se autenticar no servidor. É enviado o username, password, endereço *IP* da máquina que se está a tentar autenticar-se, a porta e tempo da ligação. Após o envio desta mensagem, caso se confirme a autenticação do cliente, o servidor envia uma *String* com os parâmetros que permitem o cliente interagir correctamente com o sistema podendo contactar outros clientes, (figura 18) os parâmetros são:
  - O tipo de *socket* a utilizar;
  - O número de porta a que o cliente deverá ligar-se;
  - O tempo de *ping* em segundos, que corresponde ao intervalo de tempo em que o cliente deverá comunicar, para verificar o estado do cliente, ou seja, se está *online*;
  - O tipo de conta correspondente ao utilizador a tentar autenticar-se.
  - O intervalo de tempo em segundos, a indicar a resposta à validação de cliente;

SocketUDP	IP/Porta	TempoPing	Autenticação	RespostaValidação
-----------	----------	-----------	--------------	-------------------

Ilustração 17 - Formato da mensagem de resposta ao *login*. (Ilustração nossa, 2012)

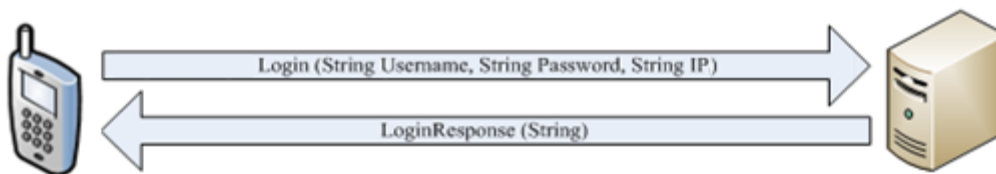


Ilustração 18 - Mensagem *Login*. (Ilustração nossa, 2012)

- **Logout:** Mensagem enviada quando um utilizador pretende terminar a sessão na máquina em que foi iniciada a mesma. É enviado o respectivo *username* e endereço IP da máquina em questão (figura 19).

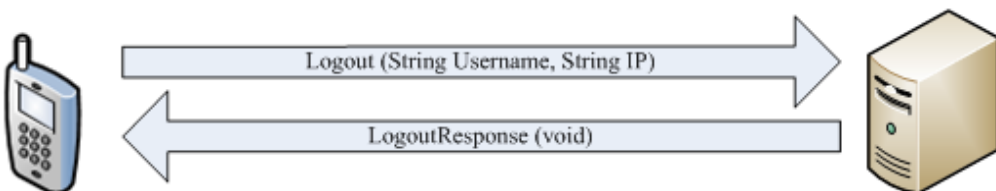
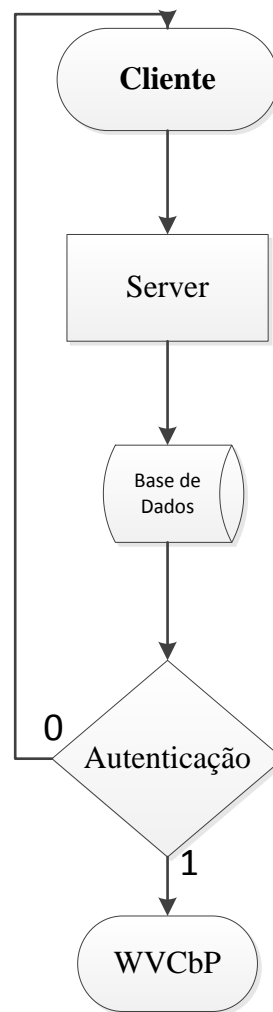
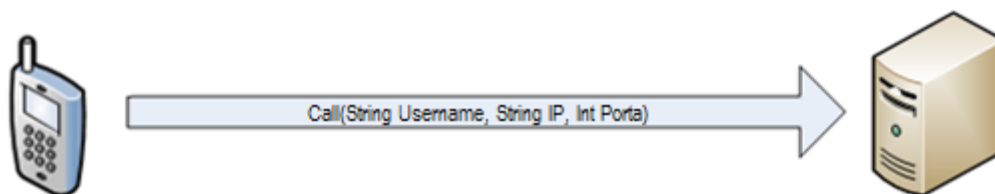


Ilustração 19 - Mensagem *Logout Client-Server*. (Ilustração nossa, 2012)



**Ilustração 20** - Diagrama de blocos dos eventos ocorridos na validação de clientes. (Ilustração nossa, 2012)

- **Call:** Mensagem para ser enviada quando um cliente pretende contactar outro cliente depois do início da sessão na máquina em que foi iniciada. É enviado para o servidor o respectivo *username*, endereço IP e porta da máquina a contactar (figura 21).



**Ilustração 21** - Mensagem *Call* Client1-Server. (Ilustração nossa, 2012)

- **Available:** Mensagem enviada com o intuito de verificar se o cliente a ser contactado está disponível a ser contactado. A verificação ocorre quando o cliente confirma a disponibilidade. O servidor irá interpretar a resposta à sua disponibilidade. Existem três respostas possíveis por parte do cliente: A positiva, a negativa e a de *timeout* quando não existe resposta (figura 22).

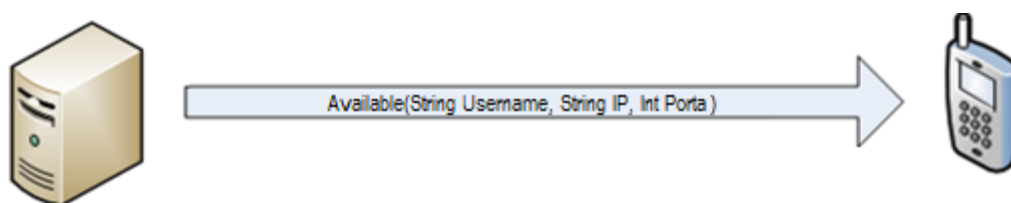


Ilustração 22 - Mensagem *Available* Server-Client2. (Ilustração nossa, 2012)

- **Timeout:** Mensagem de resposta enviada periodicamente, segundo um intervalo de tempo definido. Tem como objectivo o servidor verificar se existe resposta da parte do cliente para validar a chamada (figura 23).

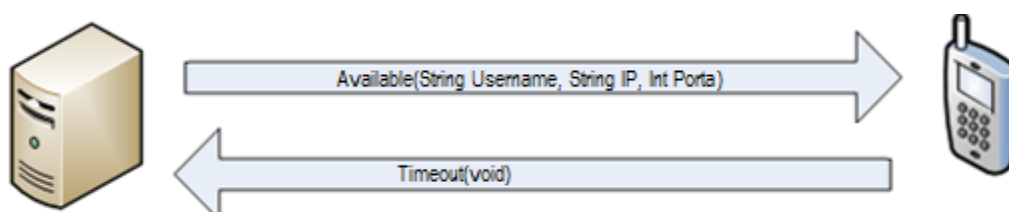


Ilustração 23 - Mensagem *timeout* Server-Client2. (Ilustração nossa, 2012)

- **Accepted:** Mensagem enviada em resposta à mensagem *Available*, quando um cliente pretende contactar outro, o segundo poderá aceitar ou rejeitar o contacto, para isso terá que responder durante um certo período de tempo respeitando a mensagem *Timeout* (figura 24).

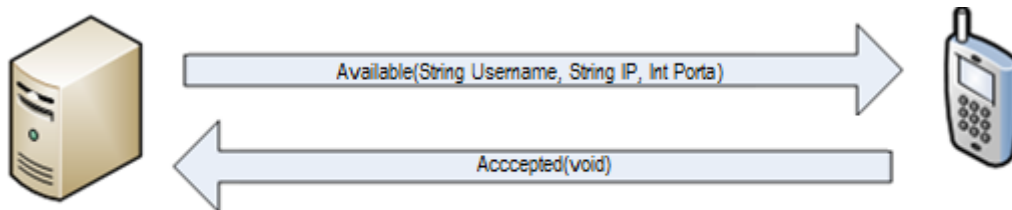


Ilustração 24 - Mensagem *Accepted* Server-Cliente2. (Ilustração nossa, 2012)

- **Answer:** Mensagem enviada pelo servidor em resposta à mensagem *Call*, depois do servidor verificar se o cliente a ser contactado, deseja receber o contacto do cliente1 a mensagem *Answer* é enviada (figura 25).

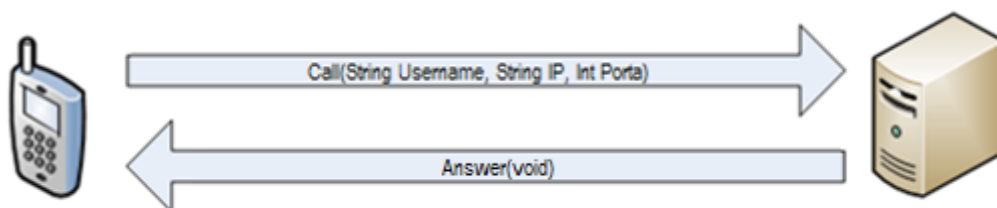


Ilustração 25 – Mensagem *Answer* Client1-Server. (Ilustração nossa, 2012)



- **Established:** Mensagem enviada entre os dois clientes quando é feita a ligação e tem como objectivo a troca de dados entre clientes assim como a identificação dos mesmos. Nesta mensagem é indicado o nome de utilizador e o endereço *IP* da máquina do utilizador (figura 30). O servidor responde com uma *string* em que indica o *hash* dos conteúdos a remover da partilha e respectivas dimensões em *bytes* (figura 26)

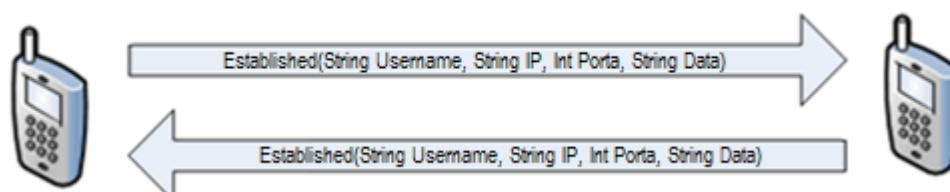


Ilustração 26 - Mensagem *Established Client1-Client2*. (Ilustração nossa, 2012)

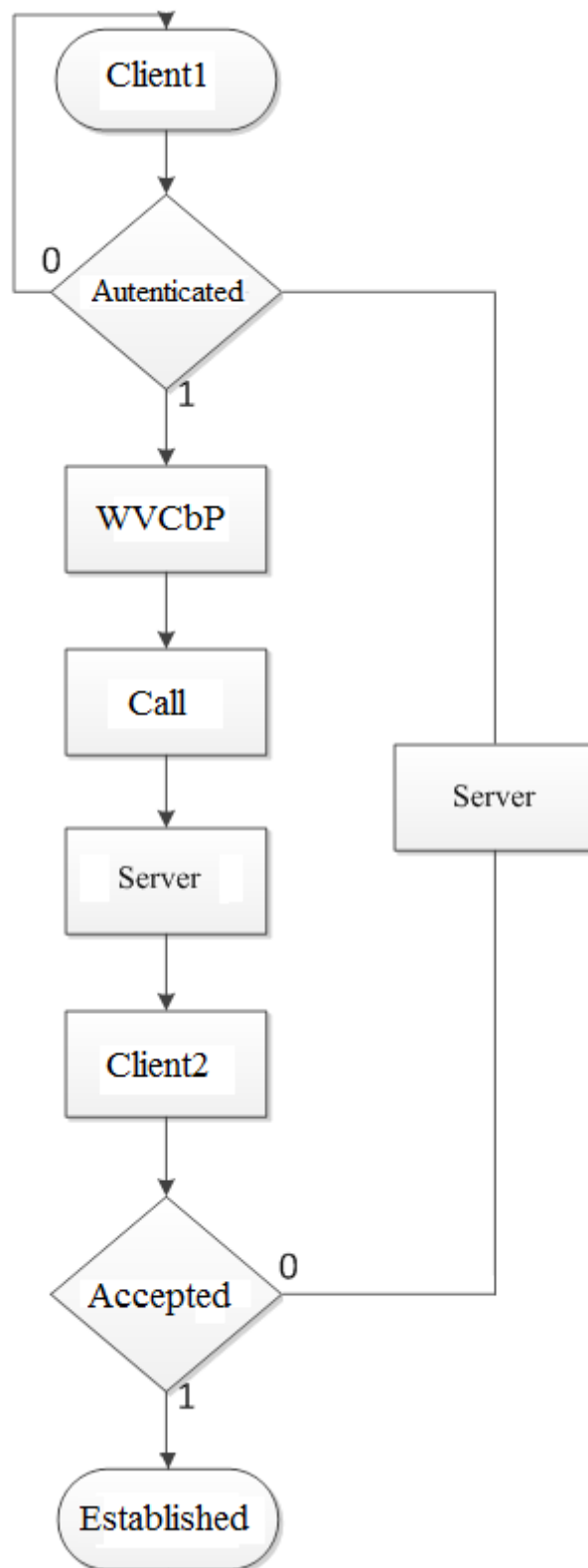


Ilustração 27 - Diagrama de blocos dos eventos ocorridos na ligação entre clientes. (Ilustração nossa, 2012)

## 7. TECNOLOGIAS ACTUAIS

Neste capítulo são estudados dois sistemas *P2P* que definem alguns dos conceitos explicados nos capítulos anteriores e relevantes para o desenvolvimento do projecto *WVCbP*. Este estudo permite um melhor conhecimento de sistemas, cujas arquitecturas poderão ser úteis para o desenvolvimento do projecto.

### 7.1. SKYPE

A tecnologia Skype, muito conhecida no mundo Web é uma aplicação em que, os utilizadores podem fazer chamadas de voz e de vídeo, do dispositivo que estão a utilizar, seja móvel ou não, para outro dispositivo com o Skype instalado, ou para um telefone fixo ou móvel.

O Skype tem mais de 600 milhões de utilizadores registados em todo o mundo. É um serviço baseado na tecnologia *VoIP* (*Voice over Internet Protocol*). A principal diferença entre o Skype e outros clientes padrão *VoIP*, é que o Skype é um *peer-to-peer* ao invés de um sistema de cliente servidor, faz processamento em segundo plano em máquinas com o software *Skype* instalado, o nome original proposto – *Sky peer-to-peer* reflecte isso.

A tecnologia *Web Voice Communication by Phone* poderá utilizar grande parte da estrutura já em funcionamento do Skype. Actualmente é possível fazer o *download* para vários tipos de telemóveis, como *IPhone*, *Android*, *Symbian*, mas este acesso é muito limitado e pouco desenvolvido, só é possível para alguns países e ainda a operadora de telefone poderá cobrar uma chamada local por cada chamada feita pelo *Skype*. Este tarifário pode ser incluído como parte do plano de chamadas.

*“A Microsoft anunciou hoje que comprou o Skype, a empresa de comunicações através da Internet...” in Publico, 10-05-2011*

Será o futuro das comunicações?

## Skype: Um Caso de Sucesso

Skype é uma aplicação, como já referido, baseada em *VoIP* e *Peer-to-Peer (P2P)*, desenvolvida pelo Skype Technologies S.A. A empresa foi fundada por Janus Friis e Niklas Zennstrom, os mesmos empresários que desenvolveram o software de troca de informação *KaZaa*, em 2003. Ao contrário do antecessor, que gerava receita através de anúncios e publicidade, o Skype não tem a intenção de actuar da mesma forma, é completamente livre de publicidade, e hoje vende serviços de chamadas telefónicas através da integração da rede de dados com a operadora convencional analógica (*PSTN*) e operadora móvel, por contrato estabelecido com as operadoras locais, de longa distância e telemóveis. Utiliza um protocolo de *IP* como *gateway* para integração entre as operadoras. Estes serviços são conhecidos como *SkypeOut*, e permitem através da compra de créditos, realizar comunicações para qualquer telefone do mundo, fixo ou móvel, com tarifas extremamente reduzidas, e *SkypeIn*, o que fornece um número de telefone fixo, através do pagamento de uma pequena anuidade, para que os utilizadores possam receber no Skype chamadas originadas de telefones comuns. Ainda existe o serviço pago de voice-mail. As ligações *Skype-Skype* são gratuitas.

O modelo de computação *Peer-to-Peer* bem característico da rede *Skype* é amplamente implementado na Internet, possui várias diferenças em relação ao esquema *client-server*. No padrão de comunicação *client-server* o fluxo de transmissão é assimétrico, a maioria do tráfego ocorre no sentido *client-server* e o cliente na maioria das vezes é um participante passivo na transmissão. Nos sistemas *P2P* todos os integrantes da rede possuem o mesmo programa (código-fonte) e cada membro da rede pode actuar como *client-server* de recursos.

As redes *P2P* são sistemas distribuídos sem controlo centralizado, nos quais o programa que é executado em cada integrante da rede é equivalente em funcionalidade. Os participantes do sistema *Peer-to-Peer* podem agir como *client-server* de recurso. Numa rede virtual sobre a rede física (normalmente a Internet) e colaboram uns com os outros em tarefas específicas, como partilha de informação, responsável pela ampla popularização do *software Napster e Kazaa*. Assim a rede *P2P* tornou-se um próximo passo natural da evolução e teve um impacto significativo com a empresa *Skype*. Este tipo de computação representa uma alternativa ao modelo *client-server* muito conhecido e possui um potencial ainda não explorado na totalidade. [Ganguly, Bhatnagar 2008, online]

Cada tipo de aplicação P2P possui próprios requisitos de segurança. Entre os requisitos fundamentais de segurança, está o controlo de acesso, permite que cada grupo ponha em prática a própria política de autorização. Assim, cada utilizador pode estabelecer quem são as entidades que podem ter acesso a recursos e quais as condições a cumprir, para terem esse direito. Além dos requisitos de segurança, o sucesso de uma rede *Peer-to-Peer* depende de factores como o protocolo de comunicação utilizado, a possibilidade de operar em ambientes protegidos como, por exemplo, aqueles que têm *firewalls*, a arquitectura de distribuição utilizada que pode ser totalmente descentralizada ou não, o tempo decorrido desde uma solicitação até ao tempo de resposta, o número de utilizadores do sistema e o suporte a ambientes que utilizam a tradução de endereços de rede (*NAT*) para possibilitar a comunicação entre elementos com endereços *IP* privados e endereços *IP* válidos na Internet [Ratnasamy, Francis, Handley, Karp and Shenker, 2001, p.161].

### **Arquitectura de rede Skype**

A maior parte das características de uma rede P2P, são adequadas para definir a arquitectura de uma rede *Skype*, no entanto há alguns aspectos importantes que diferem o *Skype* de um *software P2P* totalmente puro, ou seja, o sistema *Skype* utiliza um servidor central de autenticação para validação dos utilizadores, para acesso à rede e à própria utilização da aplicação. Esta autenticação é um tipo de chave privada com assinatura digital embebida em *RSA-Executable* (*tipo de assinatura digital portátil*)

Outra característica importante da arquitectura *Skype*, consiste em ser composta por dois tipos de nós, os próprios clientes instalam localmente a aplicação, chamado de cliente *Skype* (*CS*), onde certos clientes podem actuar como super-utilizadores ou super-nós (*SN*), que podem ser entendidos como ponto de extremidade da nuvem *Skype*, actua como *proxy* para os nós/utilizadores comuns da rede, sendo assim possível conseguir ter várias sessões abertas e em comunicação, como pode ser observado na Ilustração 28

Para actuar com estas características, os nós devem possuir alguns pré-requisitos, como, por exemplo, possuir um endereço *IP* público, não possuir restrições quanto a *firewall*, *CPU*, *RAM* insuficientes e uma determinada banda mínima de acesso à Internet, entre outras.

Outra importante entidade da rede é o servidor de login Skype, o único elemento de rede centralizado e localizado em Amsterdão, Holanda. Quando recursos como *SkypeIn* ou *SkypeOut* são utilizados, a comunicação passa necessariamente através de servidores localizados em vários países e áreas de comunicação.

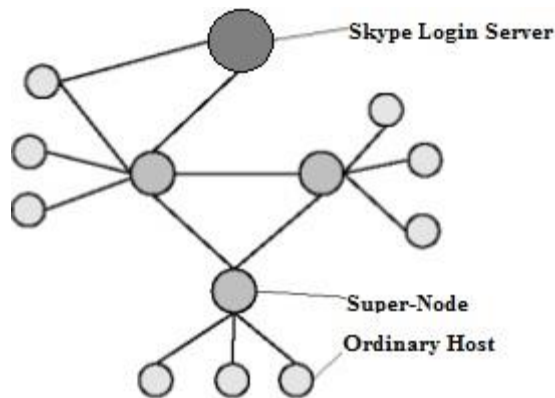


Ilustração 28 - Arquitectura de rede Skype. (Skype, 2012, online)

### ***RSA executables***

A criptografia *RSA* é um sistema criptográfico onde a chave de codificação é pública, permite a qualquer pessoa codifique mensagens, a chave de descodificação é privada. A impossibilidade de se quebrar o sistema de criptografia *RSA* ocorre devido à não existência de algoritmos eficientes para o processo de divisão de inteiros. Actualmente são utilizados números com 150 algarismos, para os quais, com a capacidade de computação actual, o processo de factoração levaria milhares de anos.

Por se tratar de uma rede descentralizada, depois do login cada CS deve construir e manter actualizada uma tabela para identificar os *super-nós* conhecidos, a esta tabela damos o nome de *host cache (HC)*, que contém o endereço *IP* e o respectivo número da porta dos *super-nós* que estiverem *on-line* e está localizada no registo do sistema operativo. Pelo menos uma entrada válida deve existir nos CS e um máximo de 200 entradas, caso contrário o CS não poderá ter acesso à rede e irá dar um erro do tipo *login failure*.

O *Skype* também mantém outra listagem local criptografada armazenada no registo do sistema operativo, a lista de contactos. Caso o utilizador se ligue através da mesma conta noutro computador, ele terá que reconstruir manualmente a lista de contactos, esta lista não se encontra armazenada em servidores na rede. A estratégia de utilizar

o máximo possível do *hardware* do utilizador também pode ser entendida como um grande factor de sucesso. Podemos perceber que a rede só é responsável pelas informações de *login*, e cada nó fica responsável por todo o processamento e armazenamento das restantes informações.

O suporte de redes locais com *IP's* não válidos e com *firewall* nos *routers* são pontos importantes do próprio funcionamento, onde as referências nos sugerem a utilização dos protocolos ou variações instaladas localmente em cada nó e não ao uso de um servidor *NAT*. Isto acontece, porque o excesso de tráfego relacionado às informações de controlo trocadas entre o cliente e o servidor *NAT* seria decisivo para a qualidade final das chamadas. [Ganguly, Bhatnagar 2008, online]

### 7.1.1. LOGIN

O processo de *login* talvez seja a função mais crítica de todas as operações e a mais difícil de ser mapeada. A autenticação ocorre, como sabemos, no servidor de *login*, é notificada a presença do nó na rede. Além disso, ocorre o processo de determinação do tipo de *NAT* e *firewall* que se encontra no lado do cliente, ainda é verificado todo o processo de *search* dos nós da rede com endereços *IP* públicos que estão on-line, isto é, os super-nós, com objectivo de dar continuidade ao acesso à rede *Skype*.

A figura 29 representa um esquema do processo de *login*. Em primeiro lugar, a tabela *HC* encontra-se vazia, logo a seguir à primeira instalação o *CS* recebe uma actualização de endereços *IP* e pares de portas conhecidos dos super-nós, para que o *CS* se possa ligar à rede. O *CS* tenta estabelecer uma comunicação através de pacotes *UDP* e caso não tenha resposta satisfatória, tenta ligar-se através do protocolo *TCP* utiliza a mesma tabela inicial *HC*, o próprio endereço *IP* e ainda a porta 80 (*HTTP*). Em caso de nova falha, o *CS* tenta ligar-se através do *IP* contido em *HC* e porta 443 (*HTTPS*). O *CS* aguarda por mais 6 segundos e caso a ligação não seja estabelecida, todo o processo entra em *loop* por mais quatro vezes até que ao final a falha no *login* é reportada caso não tenha sucesso na ligação, esta falha pode dever-se ao facto de não haver uma tabela *HC* válida. No caso da utilização de uma *firewall*, o *CS* pode estar bloqueado para tráfego *UDP* e habilitado para *TCP*. Mesmo assim, o programa irá funcionar normalmente, com a vantagem de não trocar pacotes *UDP*, reduz, portanto, o número de passos a efectuar na ligação e, conseqüentemente o fluxo de bits.

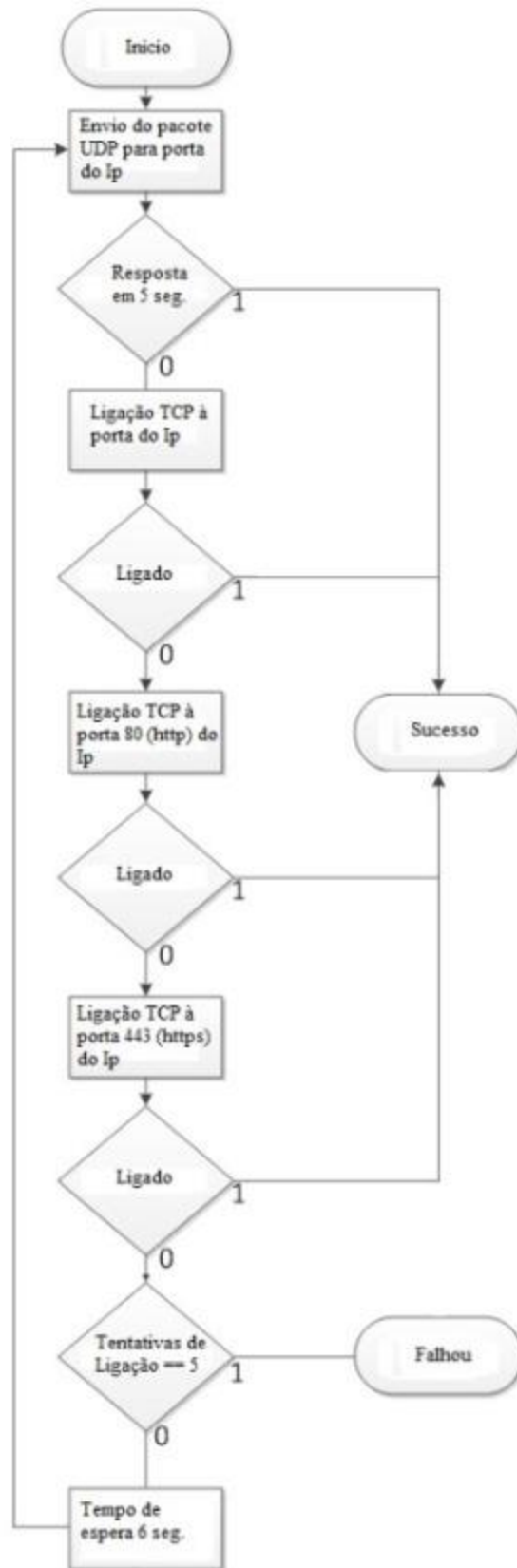


Ilustração 29 – “Diagrama de estados do algoritmo de login”, (Bhatnagar, 2008,p.91)



### 7.1.2. PRIMEIRA LIGAÇÃO

O próximo passo que se segue após inserir o login e *password* no programa, o CS estabelece a comunicação com o (s) super-nó (s), a partir da tabela *HC*. O CS envia pacotes *UDP* para pelo menos quatro endereços de *SN* contidos na tabela *HC*, e os *SN* que responderem passam a trocar mensagens *TCP* com o CS, embora necessite de apenas um *SN* para manter a ligação. Depois da troca de alguns pacotes *TCP* com o *SN*, o CS adquire o endereço do servidor de login, que também é bem conhecido 80.160.91.11, e cujo nome é *ui.skype.com*, localizado em Amsterdão. Depois do processo de validação de nome e *password*, o CS mantém a ligação ao *SN* até que um dos dois termine a sessão. Caso o *SN* seja *desligado*, CS estabelece outra ligação *TCP* com outro *SN* baseado na tabela *HC*.

Para clientes Skype que estejam inseridos em ambiente com *NAT*, todo o processo é transparente para o utilizador e todo o processo de ligação segue as mesmas regras que clientes com *IP* válidos na Internet.

Como a rede Skype é por definição muito dinâmica, CS deve manter actualizada a tabela de *routing* [Zhao, Kubiawicz, And Joseph, 2001].

Para isso, depois do processo de login, o CS envia pacotes *UDP* para um dos 22 nós distintos e actualiza a tabela *HC* com aqueles nós que reponderam a solicitação, desde que a ligação não esteja a impedir tráfego *UDP* devido a uma *firewall*.

### 7.1.3. SEGURANÇA

Outro factor importante é a questão da segurança em relação às contas de utilizador, quando realizam chamadas destinadas à rede telefónica convencional ou mesmo entre a rede de pacotes e também no que se refere à transferência de arquivos. A segurança das chamadas tem muitos aspectos envolvidos e é difícil de ser analisada como um todo, o Skype não publica o algoritmo de troca de chaves, o protocolo e o básico do desenho dos certificados. Outros factores que também influenciam são, a segurança da máquina onde está instalado o *software* em relação ao acesso à Internet, a existência de programas do tipo *spyware*, o facto de estarmos a utilizar redes não centralizadas, etc. Não podemos saber, portanto, a precisão, o grau de segurança e que tipos de ataques a rede e a aplicação podem suportar.

A segurança dos dados enviados sobre uma ligação criptografada depende de muitos factores, inclui um tipo de algoritmo criptográfico, as chaves são alteradas, como o algoritmo é implementado, o protocolo que transporta o algoritmo e a implementação.

Para o sistema de segurança implementado pela Empresa pode ser pobre ou altamente complexo, principalmente porque não temos muitos dados sobre o protocolo de comunicação, diferentemente de outras aplicações VoIP que são bem conhecidas. Porém, ainda não há registos sobre clonagens de contas de utilizadores, e sabe-se que a segurança é um aspecto mais poderoso quando comparado com redes *PSTN*. Estes sistemas são mais vulneráveis quando comparado com sistemas de *VoIP* sobre *VPN's*.

- Outras preocupações relacionadas com as questões de segurança na implementação do *software* são:
- o *default* do programa guardar um histórico do áudio das conversas em arquivo;
- Podemos saber quando é que outros utilizadores estão ligados num dado instante por *default*;
- Não se sabe se os *SN's* que encaminham as mensagens podem interceptá-las, o tráfego é processado localmente e por fim, os serviços *SkypeIn* e *SkypeOut* podem ser vulneráveis em pontos onde a chamada é transmitida pelas redes convencionais, nestes pontos as chamadas são descriptadas.

[Ganguly, Bhatnagar 2008, online]

#### **7.1.4. OUTROS SERVIÇOS**

- **Vários computadores a utilizar uma só conta** - O *Skype* permite que um utilizador esteja ligado à mesma conta simultaneamente em mais de uma máquina, assim as chamadas para este utilizador podem ser atendidas em locais diferentes. Uma chamada ao ser atendida num determinado terminal, implica que outras chamadas feitas para outros *login's* sejam canceladas. No caso do *chat*, o processo é o mesmo, ao contrário do *MSN Messenger*, que possui um sistema de registo que permite o login em apenas um computador.

- **Chamada em Espera** - A chamada em espera é mais um recurso do *software Skype*, tecnicamente CS envia mensagens periódicas sobre *TCP* para *SN* para manter a ligação.
- **Audiokonferência** - A audiokonferência suporta até quatro pessoas em conferência em qualquer lugar do mundo com acesso Internet.

### 7.1.5. SÍNTESE DO SKYPE

O *Skype* é o primeiro *software* baseado em redes *P2P* a utilizar redes sobre *IP*, não utiliza protocolo de padrão aberto e as mensagens são criptografadas, de modo que a análise e descrição torna-se mais difícil. Algumas das principais funções e componentes foram documentadas, através da análise do próprio funcionamento do *Skype*.

O *software Skype* em comparação com outros operadores de *IP* foi capaz na primeira semana atingir 60 mil *downloads* (Agosto de 2003). Pela primeira vez na história da internet, protocolos que trabalham em complexos ambientes de rede sobreviveram e tornaram-se amplamente difundidos. Não comparamos directamente, mas o protocolo de padrão aberto *Session Initiation Protocol (SIP)* ainda tem as suas limitações, como é o facto de ter muita dificuldade em trabalhar com *NAT*.

A popularidade do *Skype* deve-se ao facto do *software* e da rede serem gratuitos, da excelente qualidade da voz na maior parte do tempo de ligação quando comparado com a rede analógica tradicional e aos concorrentes directos *MSN* e *Yahoo IM*, demonstrado pelo uso de *codecs*. Além disso, também não é necessário ter um endereço *IP* público na Internet e a instalação, que por ser tão simples (não precisa ser configurado) também se destaca como factor de sucesso. Sem contar com a facilidade de instalação e difusão nos *players* de voz que acabaram por absorver o *software*, como:

- *Applications/Soft phones: Skype, Nortel, Webex, Hotsip, Marratech, Gatelinx, KPhone, XTen;*
- *IP Phones: WorldGate, Grandstream, Pingtel;*
- *Chip: Audiocodes, TI Telogy, LeadTek, Mindspeed.*
- *Sistemas Operativos: Windows 9x ou superior, Pocket PC, Linux, Mac OS X.*

Como já foi referido o Skype também tem outros recursos como o *chat*, procura, e transferência de arquivos, logo não existe a necessidade de se utilizar mais do que um *software* para comunicação pela Internet.

## **7.2. MOBILEVOIP**

O *MobileVoip* é uma aplicação VoipFree, ainda pouco conhecida mas começa a ganhar expressão no mundo Web. Através das suas funcionalidades e do seu baixo custo, conseguiu já cativar cerca de 200.000 subscritores. Esta tecnologia permite aos utilizadores registados usufruir de chamadas sem custos, desde a aplicação instalada nos *Smartphones* ou no PC, entre estes e também para linhas telefónicas tradicionais ou móveis em diversos países espalhados pelo mundo.

Este *software* é de *download* gratuito, pode ser instalado em várias marcas de *Smartphones*. A aplicação *MobileVoip* é de muito fácil utilização, e é muito intuitiva a utilização de todas as funcionalidades associadas.

Apesar do *MobileVoip* permitir chamadas gratuitas desde o Computador Pessoal, para linhas telefónicas tradicionais ou móveis, existe um limite semanal de 300 minutos por endereço IP. A partir desse limite imposto, são activadas tarifas específicas para este tipo de chamadas, mas sempre tarifas muito reduzidas.

Andorra	free
Argentina	free
Australia	free
Austria	free
Belgium	free
Canada*	free
Czech Rep.	free
Denmark	free
Estonia	free
France	free
Germany	free
Hong kong*	free
Hungary	free
Iceland	free
Ireland	free
Italy	free
Luxembourg	free
Malaysia	free
Netherlands	free
New zealand	free
Norway	free
Poland	free
Portugal	free
Russian	free
South korea	free
Spain	free
Sweden	free
Switzerland	free
U. kingdom	free
U.S.A.*	free

**Ilustração 30** - Alguns Países onde a comunicação é gratuita, (MobileVoip, 2012, online)

### ▪ Tipos de Chamadas - *MobileVoip*

Através da tecnologia *MobileVoip* é possível realizar chamadas sem custo a partir do telemóvel pessoal, desde que tenha uma ligação à internet e seja um *Smartphone*, com a instalação do software (freeware) instalado. É compatível com as plataformas iPhone, Android e Symbian, pode-se dizer que é *user friendly*, com as seguintes funcionalidades:

- **Chamada Voip** – Para realizar esta opção de chamada que é uma ligação directa e imediata, é necessário uma ligação à internet 3G/WiFi, é necessário também ter subscrito um operador Voip e as taxas aplicadas serão de acordo com este operador do serviço;
- **Chamar de volta** – A funcionalidade "Chamar de volta" a ligação à internet que está a utilizar não precisa de ser particularmente forte. Com uma ligação 3G/WiFi mais fraca, com tráfego relativamente reduzido quanto possível, é possível configurar uma chamada "Chamar de volta".
- **Acesso Local** – A funcionalidade "acesso local" está disponível para um conjunto de países, é possível utilizar o *MobileVOIP* configurando a aplicação para uma chamada de "Acesso Local". No caso de uma fraca ligação à internet, será taxado pela operadora uma chamada local.

### 7.3. VOIPCHEAP

A tecnologia *VoipCheap* é um *software* baseado em *IP*, com a possibilidade de fazer chamadas sem custo para o utilizador, do dispositivo onde está instalado o *software*, entre dispositivos com o *voiccheap* instalado e para telefones comuns. Este tipo de chamada já abrange 36 países em todo o mundo, 23 países europeus incluindo Portugal.

Com a funcionalidade de se realizar uma chamada sem necessidade de instalação do *software*, apenas aceder ao *site* do *VoiceCheap* e registar o próprio número de telefone e o de destino da chamada. Basta ter uma ligação de banda larga (cabo ou *DSL*) e um computador de gama média. Nesta tecnologia as tarifas pagas são mais baixas ainda do que as do Skype, é possível que venha a ter grande projecção futura por ser um *software* de elevada qualidade.

A figura seguinte representa os Sistemas Operativos de Smartphones onde poderá ser instalado o *software* e fazer uso de todas as funcionalidades da tecnologia *VoiceCheap*:



Ilustração 31 – Sistemas Operativos de Smartphones a utilizar *VoiceCheap* (*VoiceCheap*, 2012, online)





## 8. DESENVOLVIMENTO DE PÁGINA WEB

O desenvolvimento de uma página web permite aos utilizadores a visualização de algumas informações referentes ao estado do sistema, para poderem fazer comunicações entre clientes da aplicação *WVCbP* (esta última funcionalidade pertence já a trabalho futuro). A possibilidade de poder aceder remotamente às informações do sistema é vantajosa a nível da monitorização do comportamento de utilizadores online.

Ainda de salientar que este serviço *Web*, não é solução para os nossos objectivos, mas ajuda a perceber melhor um sistema *WVCbP*, dando a conhecer algumas tecnologias *Web*. Algumas funcionalidades descritas podem pertencer ao capítulo de trabalho futuro, abrindo portas para um maior desenvolvimento deste projecto.

Outro dos objectivos pretendidos, com a construção de uma página é a capacidade desta poder oferecer ao administrador total controlo na definição dos vários parâmetros do sistema e disponibilizar um serviço de autenticação aos utilizadores.

Deste modo podem também os utilizadores monitorizar o seu perfil e gestão de contactos. O controlo dos parâmetros, aliado à informação sobre o estado de todo o sistema, permite obter uma maior informação global do sistema.

As funcionalidades alcançadas no desenvolvimento da página, na óptica do utilizador são:

- Registo e adesão ao serviço *WVCbP*;
- Gestão do próprio perfil;
- Visualização de contactos *online*;
- Possibilidade de contacto para dispositivos com a aplicação instalada (trabalho futuro);
- Na óptica do administrador, há a possibilidade da definição e modificação de todos os parâmetros do sistema.

O desenvolvimento da página web foi realizado em PHP. O PHP é uma das linguagens script mais utilizadas pela indústria na produção de páginas web dinâmicas. As principais vantagens da utilização desta plataforma são:

- Desempenho;
- Estabilidade;
- Segurança;
- Simplicidade;
- Código aberto.

O código PHP é embebido numa página HTML e executado do lado do servidor, aliviando os recursos do lado do cliente. A simplicidade na implementação e compreensão da linguagem PHP, juntamente com a capacidade de ser executado em diferentes plataformas, tais como, UNIX, Mac e Windows, são outras das vantagens desta tecnologia. No contexto do sistema desenvolvido, o PHP permite a interacção com a base de dados *MySQL* e é carregado no servidor Apache, possibilitando a obtenção fácil e rápida dos dados que irão ser disponibilizados na página web.

Devido à simplicidade e capacidade de ser executado em múltiplas plataformas, foi escolhido o PHP. Constituiu igualmente uma forma de aprendizagem de uma linguagem tão popular, com o desenvolvimento da página web [Matthews, J. Cole, And J. D. Gradecki, 2003, p.23].

## **8.1. IMPLEMENTAÇÃO DA PÁGINA WEB**

Na construção da página web, foi utilizada a ferramenta Adobe Dreamweaver. Esta ferramenta de desenvolvimento é intuitiva, pois fornece modelos e assistência na construção da página, permite a integração de várias tecnologias como o CSS, *JavaScript*, *PHP*, entre outras.

A aplicação web é composta essencialmente por uma página que disponibiliza o serviço de autenticação e uma página que contém o menu principal para acesso às várias funcionalidades.

O serviço de autenticação permite que apenas os utilizadores registados no sistema possam aceder à página principal. Recorrendo ao PHP, para cada utilizador autenticado, é criada uma sessão que guarda informações temporárias do mesmo. Essas informações estão disponíveis em todas as páginas da aplicação. Numa sessão, é criada uma identificação única (do inglês, *Unique ID – UID*) para cada

visitante e todas as informações são guardadas com base nesta identificação. O UID é guardado num *cookie* ou propagado num URL. O acesso é interdito a qualquer das páginas caso o utilizador não tenha realizado a autenticação com sucesso. A figura 40 mostra o aspecto da página web de autenticação.

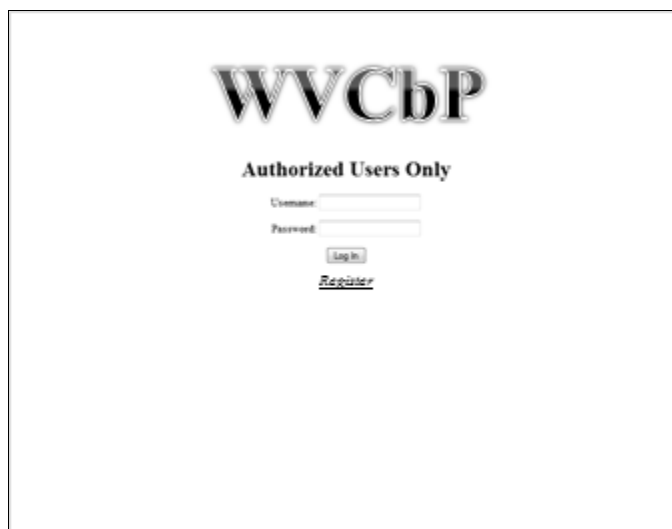


Ilustração 32 - Página de autenticação. (Ilustração nossa, 2012)

Após realização da autenticação é apresentada a página inicial como se pode visualizar na figura seguinte. Em todas as páginas, no canto superior direito, é mostrada a identificação do utilizador e a opção para efectuar *Logout*, regressando à página de autenticação.

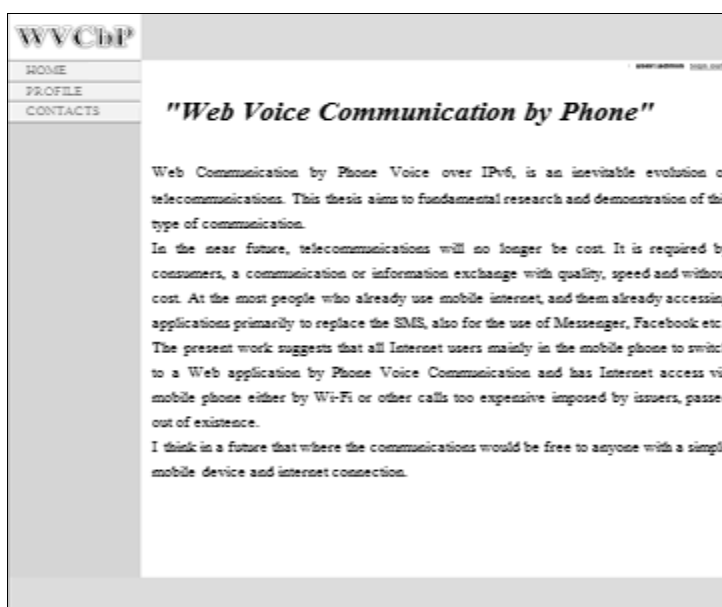
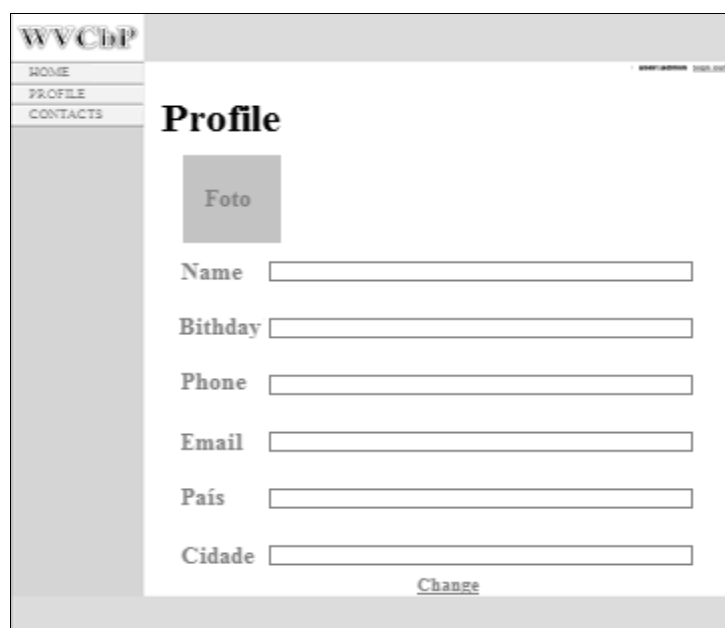


Ilustração 33 - Página inicial. (Ilustração nossa, 2012)

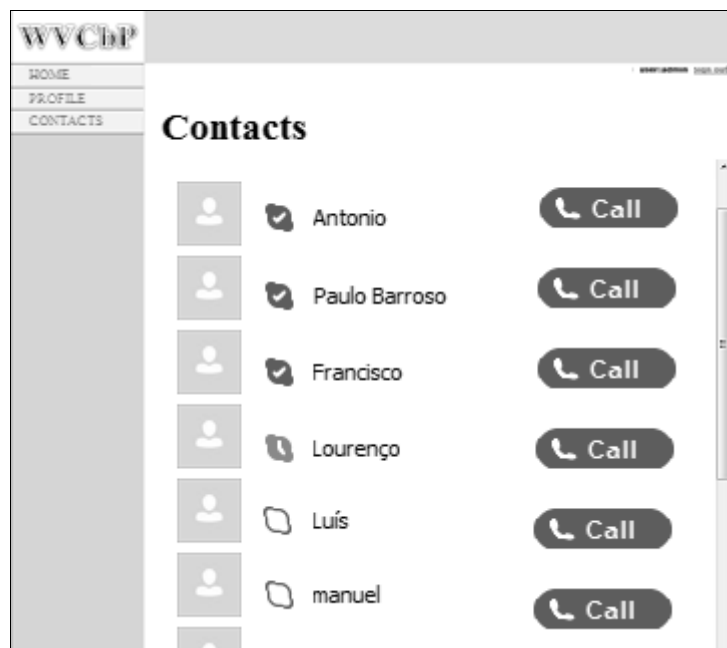
A página profile.php permite visualizar o perfil do utilizador, o cliente pode alterar todas as informações associadas.



The screenshot shows a web interface for WVCbP. On the left, there is a navigation menu with links for HOME, PROFILE, and CONTACTS. The main content area is titled "Profile" and contains a form for user information. The form includes a placeholder for a photo labeled "Foto", and input fields for Name, Birthday, Phone, Email, País, and Cidade. A "Change" button is located at the bottom right of the form.

Ilustração 34 - Página de Perfil. (Ilustração nossa, 2012)

A página contacts.php permite visualizar os contactos do utilizador, a funcionalidade *call* é apenas um desenvolvimento futuro da aplicação onde, o cliente poderá fazer uma chamada para os seus contactos.



**Ilustração 35** – Página de Contactos. (Ilustração nossa, 2012)



## 9. CONCLUSÃO

O desafio fundamental proposto nesta tese passava pelo desenvolvimento e criação de um modelo, que resultasse numa solução/produto para utilizadores de dispositivos que através de uma ligação à internet pudessem comunicar por voz e sem recurso a uma operadora móvel.

O sistema apresentado deverá ser relevante e suficiente para enriquecer o conhecimento sobre diferentes tecnologias. São o exemplo das tecnologias estudadas, redes wireless de banda larga, como o WiMax, implementado sobre o *Internet Protocol* do futuro (IPv6), para o objectivo de implementação do WVCbP tendo sempre como base a tecnologia P2P.

Esse desafio foi cumprido, com a criação de um protótipo capaz de comunicar entre equipamentos onde esteja instalada a aplicação, independentemente do tipo de equipamento ou fabricante.

Esse protótipo serve de base à criação de um novo produto *WVCbP*, que permitirá à mesma gerir melhor a sua infra-estrutura tecnológica, assim como fornecer aos seus clientes, serviços de telecomunicações com custos muito baixos e controlados, apenas dependentes de uma ligação à internet.

No desenvolvimento deste projecto pretendeu-se agregar múltiplas funcionalidades com base na tecnologia *P2P*. O sistema desenvolvido, além de possibilitar a comunicação entre utilizadores, teve também como objectivo assegurar uma maior coerência em relação ao futuro próximo quando este tipo de sistemas estiver implementado no protocolo IPv6 aproveitando os recursos dos computadores pessoais dos vários utilizadores.

Inicialmente e após definição dos objectivos a alcançar para o projecto, foi realizado um estudo sobre sistemas IPv6. Esse estudo abrangeu as características dos vários tipos de sistemas existentes, com o objectivo de conhecer sistemas semelhantes ao desenvolvido e ao mesmo tempo aprender técnicas que pudessem ser úteis no âmbito do projecto.

Finalizado o estudo, foi feito o desenvolvimento da aplicação apenas com um servidor. Para cumprir os objectivos foi definido o protocolo de comunicação e um modelo *Web Voice Communication by Phone*.





## 10. TRABALHO FUTURO

O sistema foi desenvolvido sempre com o objectivo de outras soluções poderem vir a ser implementadas. Essas soluções deverão aumentar a diversidade das funcionalidades e melhorar a eficiência de todo o sistema. Ao longo do projecto foram surgindo outras soluções que poderão servir de trabalho futuro.

### • Chamadas em conferência

A solução a nível de gestão dos contactos no futuro implicaria atribuir a disponibilidade de várias chamadas em grupo de clientes, esta funcionalidade teria as opções a adaptar:

- Permitir que o utilizador defina o número e os contactos que pretende contactar em conferência;
- Implementar o serviço de registo deste tipo de chamadas no sistema pela página web;
- Disponibilizar aos utilizadores através da página web, informações específicas sobre o estado dos contactos online;
- Desligar selectivamente um dos participantes;
- Terminar a conferência, desligando todos os participantes;
- Partilha de perfis e todos os conteúdos associados sobre informação do utilizador;

### • Chamadas de Vídeo

A aplicação implementada poderá evoluir para a opção de chamadas em vídeo, esta adaptação implicaria a utilização do protocolo RTP (*Real-time Transport Protocol*), protocolo mais adequado para a transmissão em *streaming*. A partir desta funcionalidade poderão surgir inúmeras ideias, do ponto de vista profissional como videoconferências.

### • Bases de dados de perfis

A aplicação desenvolvida poderá ter uma vertente de partilha de ficheiros mais aprofundada associada à informação de utilizadores. Uma ampla e estruturada Base de dado e um sistema de gestão de bases de dados (DBMS) será interessante associar a este *software*.

Ainda outra área importante é o desenvolvimento de técnicas voltadas à segurança no desempenho. Segurança e desempenho são sem dúvida áreas de importância na utilização da arquitectura Web nas mais variadas aplicações.

## REFERÊNCIAS

ADOBE. (2012) Adobe Dreamweaver. [Online]. <http://www.adobe.com> [Consult. em 2012]

ANDROUTSELLIS-THEOTOKIS, Stephanos ; SPINELLIS, Diomidis (2004) - A Survey of Peer-to-Peer Content Distribution Technologies in ACM Computing Surveys (CSUR), New York, 2004, pp. 335 – 371.

BALAKRISHNAN, Hari KAASHOEK, M. Frans, KARGER, David, MORRIS, Robert ; STOICA, Ion - Looking up data in P2P systems," in Communications of the ACM, New York, 2003, pp. 43 – 48.

BHAGWAN, Ranjita SAVAGE, Stefan ; G. M. VOELKER, Geoffrey (2003) - Understanding Availability. San Diego.

CANNIÈRE Christophe De, ; RECHBERGER Christian, "Finding SHA-1 Characteristics: General Results and Applications, Institute for Applied Information Processing and Communications (IAIK) 2006 p.145.

CARLSON, Violet Hoffman ; GUSTAFSON, Ralph (2001) - The Rise and Fall of Napster - An Evolutionary Approach, in Proceedings of the 6th International Computer Science Conference on Active Media Technology, London, 2001, pp. 347 – 354.

COMER, Doug (2006) - Internetworking with TCP/IP: Principles, protocols, and architecture, 5th ed., P. Hall, Ed., 2006.

DEEPAK PAREEK - 2006 – WiMAX: Taking wireless to the MAX [consultado em 2012]

FREITAS, Fábio (2009) - Estudos tecnologias VoIP, Paper pdf - [www.filipefreitas.net](http://www.filipefreitas.net) [em 2012]

GARBER L.,(2008) - Mobile Wimax: The next wireless batle ground.Computer v. 41,2008 [Consult. em 2012]

ITU: Committed to connecting the world, <http://www.itu.int/ITU-D/ict/statistics/> [Consult. em 2012]

MATTHEWS, Mark, COLE, Jim ; GRADECKI, Joseph D. (2003) - MySQL and Developer's Guide. Indiana: Wiley Publishing.

MOBILEVOIP (2012). [Online]. <http://mobilevoip.com> [Consult. em 2012]

ODSON, Doreen Introdução e Histórico Do Ipv6. <http://www.lcmi.ufsc.br> [Consult. em 2012]

PAUTASSO, Cesare, ZIMMERMANN, Olaf ; LEYMANN, Frank "Restful web services vs. "big" web services: making the right architectural decision," in Proceeding of the 17th international conference on World Wide Web, New York, 2008, pp. 805-814.

PLONKA. Dave (2000) UW-Madison Napster Traffic Measurement. [Online]. <http://net.doit.wisc.edu>

RATNASAMY, Sylvia FRANCIS, Paul, HANDLEY, Mark, Mark, KARP, Richard ; SHENKER, Scott (2001) - A Scalable Content-Addressable Network," in Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, San Diego, 2001, pp. 161 - 172.

RHOTON, John (2002) - The wireless Internet explained [Consult. em 2012]

SAMRAT Ganguly, SUDEEPT Bhatnagar (2008) - Voip wireless, P2P and New Enterprise Voip.

STOICA, Ion, MORRIS, Robert, KARGER, David, KAASHOEK, M. Frans ; BALAKRISHNAN, Hari "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," in Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, San Diego, 2001, pp. 149-160.

SHELDON, Tom - Encyclopedia of Networking and Telecommunication.: McGraw-Hill Professional , 2001. [Online]. <http://www.linktionary.com/s/state.html> [Consult. em 2012]

TELES-CALADO, Rodrigo - IPv4 para IPv6 Migração ou co-existência? – <http://www.ircmania.com.br/artigos>. [Consult. em 2012]

VOIPCHEAP (2012). [Online]. <http://VoipCheap.com> [Consult. em 2012]

W3C. (2007, Junho) - Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. [Online]. <http://www.w3.org>

XcepticZP. (2007) Freenet Request Sequence. [Online].<http://en.wikipedia.org>  
[Consult. em 2012]

ZHAO, Ben, KUBIATOWICZ, John ; JOSEPH, Anthony D. - Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing, 2001.



## BIBLIOGRAFIA

BHAGWAN, Ranjita, SAVAGE, Stefan ; VOELKER, Geoffrey (2003) - Replication Strategies for Highly Available Peer-to-peer Storage Systems in Future directions in distributed computing: research and position papers. Berlin: Springer-Verlag, 2003, pp. 153- 158

GALLAGHER ; R. WILKERSON. (2012) - - Network Performance Statistics for the past 24 Hours.. [Consult. 2012]. Disponível em WWW: < URL: <http://eddie.csd.sc.edu> >.

GROUP. (2012) PHP. [Consult. em 2012] [Consult. 2012]. Disponível em WWW: < URL: <http://pt2.php.net> >.

INFOPÉDIA: Dicionário da Língua Portuguesa, Porto Editora, <http://infopedia.com> [Consult. em 2012]

JÚNIOR, Wilson (2006) - A Nova Geração de Comunicação, Monografia apresentada como pré-requisito de conclusão do Curso de Pós-Graduação de Administração em Redes Linux do Centro Universitário Federal de Lavras em Lavras-MG. Minas Gerais – Brasil

LOURENÇO, Rogério (2005) - Codificação e Transmissão de Voz na Internet: um Estudo de Caso sobre o Software SKYPE– Universidade Federal Fluminense (UFF) Niterói – RJ – Brasil Laboratório de Pesquisas em Comunicação de Dados Multimédia

PALET, Jordi (2007) - IPv6 Overall Status, IPv6 Task Force, IPv6 Steering Committee, Report, 14 February 2003. [Consult. em 2012]

RIBEIRO SANTOS, Cleymone (2004) – Integração IPv6 num ambiente cooperativo seguro Instituto de Computação Universidade Estadual de Campinas

SANTOS, Nuno (2008) - Architectures and Algorithms for IPv4/IPv6-Compliant Optical Burst Switching Networks, Thesis submitted to the University of Beira Interior in candidature for the Degree of Doctor of Philosophy in Computer Science and Engineering

SELADA, Rodrigo, SALVADOR, Nuno (2008) - Redes Wireless de Banda Larga, Universidade de Trás-os-Montes e Alto Douro, Departamento de Engenharias - Vila Real 2008

STADLER, John (2009) - An Investigation Into Skype Technologies S.A. [Consult. em 2012]

TAVARES DE MELO, Márcio (2008) - Desempenho de Redes Wimax com Mobilidade Universidade de Aveiro - in Proceeding of the 7th international conference, Aveiro, 2008, pp. 55-64.



## APÊNDICES

---



## **APÊNDICE A**

---

### **EXEMPLO DE SOCKET UDP PROGRAMÁVEL**



```
//Server UDP

using System;
using System.Net;
using System.Net.Sockets;
using System.Text;
public class UDPListener
{
    private const int listenPort = 1001;
    public static int Main()
    {
        bool done = false;
        UdpClient listener = new UdpClient(listenPort);
        IPEndPoint groupEP = new IPEndPoint(IPAddress.Any, listenPort);

        Boolean exception_thrown = false;
        Socket sending_socket = new Socket(AddressFamily.InterNetwork, SocketType.Dgram,
        ProtocolType.Udp);
        IPAddress send_to_address = IPAddress.Parse("127.0.0.1");

        string received_data;
        byte[] receive_byte_array;
        try
        {
            while (!done)
            {
                Console.WriteLine("Waiting for broadcast");
                receive_byte_array = listener.Receive(ref groupEP);
                Console.WriteLine("Received a broadcast from {0}", groupEP.ToString());
                received_data = Encoding.ASCII.GetString(receive_byte_array, 0,
                receive_byte_array.Length);
                Console.WriteLine("data follows \n{0}\n\n", received_data);

                if (received_data == "Client2")
                {
                    IPEndPoint sending_end_point = new IPEndPoint(send_to_address, 1002);
                    string text_to_send = "\nA Chamar... Client1\n\nAceitar(s) Rejeitar(n)";

                    // the socket object must have an array of bytes to send.
                    // this loads the string entered by the user into an array of bytes.
                    byte[] send_buffer = Encoding.ASCII.GetBytes(text_to_send);

                    // Remind the user of where this is going.
                    Console.WriteLine("sending to address: {0} port: {1}",
                    sending_end_point.Address,
                    sending_end_point.Port);
                    try
                    {
                        sending_socket.SendTo(send_buffer, sending_end_point);
                    }
                    catch (Exception send_exception)
                    {
                        exception_thrown = true;
                        Console.WriteLine(" Exception {0}", send_exception.Message);
                    }
                    if (exception_thrown == false)
                    {
                        Console.WriteLine("Message has been sent to the broadcast address");
                    }
                }
            }
        }
    }
}
```

```
        else
        {
            exception_thrown = false;
            Console.WriteLine("The exception indicates the message was not sent.");
        }
    }
}
}
catch (Exception e)
{
    Console.WriteLine(e.ToString());
}
listener.Close();
return 0;
}
}
```

```
//Client1
using System;
using System.Net;
using System.Net.Sockets;
using System.Text;
class Program
{
    private const int listenPort = 1003;
    static void Main(string[] args)
    {
        Boolean done = false;
        Boolean exception_thrown = false;
        UdpClient listener = new UdpClient(listenPort);
        IPEndPoint groupEP = new IPEndPoint(IPAddress.Any, listenPort);

        Socket sending_socket = new Socket(AddressFamily.InterNetwork, SocketType.Dgram,
        ProtocolType.Udp);

        IPAddress send_to_address = IPAddress.Parse("127.0.0.1");

        IPEndPoint sending_end_point = new IPEndPoint(send_to_address, 1001);

        Console.WriteLine("Enter text to broadcast via UDP.");
        Console.WriteLine("Enter a blank line to exit the program.");

        string received_data;
        byte[] receive_byte_array;
        while (!done)
        {
            Console.WriteLine("Enter text to send, blank line to quit");
            string text_to_send = Console.ReadLine();
            if (text_to_send.Length == 0)
            {
                done = true;
            }
            else
            {
                byte[] send_buffer = Encoding.ASCII.GetBytes(text_to_send);

                Console.WriteLine("sending to address: {0} port: {1}",
                sending_end_point.Address,
                sending_end_point.Port);
                try
                {
                    sending_socket.SendTo(send_buffer, sending_end_point);
                }
                catch (Exception send_exception)
                {
                    exception_thrown = true;
                    Console.WriteLine(" Exception {0}", send_exception.Message);
                }
                if (exception_thrown == false)
                {
                    Console.WriteLine("Message has been sent to the broadcast address");
                }
                else
                {
                    exception_thrown = false;
                    Console.WriteLine("The exception indicates the message was not sent.");
                }
            }
        }
    }
}
```

```
Console.WriteLine("Waiting for broadcast");
receive_byte_array = listener.Receive(ref groupEP);
Console.WriteLine("Received a broadcast from {0}", groupEP.ToString());
received_data = Encoding.ASCII.GetString(receive_byte_array, 0,
receive_byte_array.Length);
Console.WriteLine("data follows \n{0}\n\n", received_data);

IPEndPoint sending_endPoint = new IPEndPoint(send_to_address, 1002);
string text_toSend = Console.ReadLine();

// the socket object must have an array of bytes to send.
// this loads the string entered by the user into an array of bytes.
byte[] sendBuffer = Encoding.ASCII.GetBytes(text_toSend);

// Remind the user of where this is going.
Console.WriteLine("sending to address: {0} port: {1}",
sending_end_point.Address,
sending_end_point.Port);
try
{
    sending_socket.SendTo(sendBuffer, sending_endPoint);
}
catch (Exception send_exception)
{
    exception_thrown = true;
    Console.WriteLine(" Exception {0}", send_exception.Message);
}
if (exception_thrown == false)
{
    Console.WriteLine("Message has been sent to the broadcast address");
}
else
{
    exception_thrown = false;
    Console.WriteLine("The exception indicates the message was not sent.");
}

}
} // end of while (!done)
} // end of main()
} // end of class Program
```



```

//Client2
using System;
using System.Net;
using System.Net.Sockets;
using System.Text;
public class UDPListener
{
    private const int listenPort = 1002;
    public static int Main()
    {
        bool done = false;
        UdpClient listener = new UdpClient(listenPort);
        IPEndPoint groupEP = new IPEndPoint(IPAddress.Any, listenPort);

        Boolean exception_thrown = false;
        Socket sending_socket = new Socket(AddressFamily.InterNetwork, SocketType.Dgram,
        ProtocolType.Udp);
        IPAddress send_to_address = IPAddress.Parse("127.0.0.1");

        string received_data;
        byte[] receive_byte_array;
        try
        {
            while (!done)
            {

                Console.WriteLine("Waiting for broadcast");
                receive_byte_array = listener.Receive(ref groupEP);
                Console.WriteLine("Received a broadcast from {0}", groupEP.ToString());
                received_data = Encoding.ASCII.GetString(receive_byte_array, 0,
                receive_byte_array.Length);
                Console.WriteLine("data follows \n{0}\n\n", received_data);

                string text_to_send = Console.ReadLine();
                if (text_to_send == "n")
                {
                    done = true;
                }
                if (text_to_send == "s")
                {
                    Console.WriteLine("Ligado... Cliente1");
                    IPEndPoint sending_end_point = new IPEndPoint(send_to_address, 1003);
                    string text_toSend = Console.ReadLine();

                    // the socket object must have an array of bytes to send.
                    // this loads the string entered by the user into an array of bytes.
                    byte[] send_buffer = Encoding.ASCII.GetBytes(text_toSend);

                    // Remind the user of where this is going.
                    Console.WriteLine("sending to address: {0} port: {1}",
                    sending_end_point.Address,
                    sending_end_point.Port);
                    try
                    {
                        sending_socket.SendTo(send_buffer, sending_end_point);
                    }
                    catch (Exception send_exception)
                    {
                        exception_thrown = true;
                        Console.WriteLine(" Exception {0}", send_exception.Message);
                    }
                }
            }
        }
    }
}

```

```
    }  
    if (exception_thrown == false)  
    {  
        Console.WriteLine("Message has been sent to the broadcast address");  
    }  
    else  
    {  
        exception_thrown = false;  
        Console.WriteLine("The exception indicates the message was not sent.");  
    }  
    }  
    }  
    }  
} catch (Exception e)  
{  
    Console.WriteLine(e.ToString());  
}  
listener.Close();  
return 0;  
}  
}
```

