

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Transmissão de Vídeo Sobre Redes Emalhadas

Hugo Melchior Silva De Gouveia E Silva

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Manuel Alberto Pereira Ricardo

Co-orientador: José Manuel Soares Oliveira

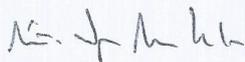
Junho de 2014

A Dissertação intitulada

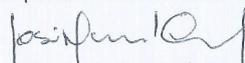
“Transmissão de Vídeo Sobre Redes Emalhadas”

foi aprovada em provas realizadas em 22-07-2014

o júri



Presidente Professor Doutor Mário Jorge Moreira Leitão
Professor Associado do Departamento de Engenharia Eletrotécnica e de
Computadores da Faculdade de Engenharia da Universidade do Porto



Professor Doutor José Manuel Tavares Vieira Cabral
Professor Auxiliar do Departamento de Eletrónica Industrial da Escola de Engenharia
da Universidade do Minho



Professor Doutor Manuel Alberto Pereira Ricardo
Professor Associado do Departamento de Engenharia Eletrotécnica e de
Computadores da Faculdade de Engenharia da Universidade do Porto

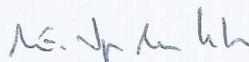
O autor declara que a presente dissertação (ou relatório de projeto) é da sua
exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente
autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou
inspirados em trabalhos de outros autores, e demais referências bibliográficas
usadas, são corretamente citados.



Autor - Hugo Melchior Silva de Gouveia e Silva

Faculdade de Engenharia da Universidade do Porto

Provar



Resumo

O desenvolvimento tecnológico nas telecomunicações tem sofrido uma evolução acentuada nos últimos anos, centrando-se atualmente o foco da investigação na área das comunicações móveis. Nesta área, a aposta tem sido na definição e no desenvolvimento de novos tipos de rede, essencialmente apropriadas para ambientes distintos dos tradicionais ambientes cobertos pelos operadores de telecomunicações.

As redes emalhadas, habitualmente designadas por redes mesh, surgem neste contexto. Desenvolvidas a partir das conhecidas redes ad-hoc, as redes mesh prometem essencialmente inovação na organização e na partilha de internet numa rede.

Desde o aparecimento do vídeo, como formato de armazenamento e transmissão de informação, que a sua transmissão em tempo real constitui um aspeto alvo de estudo, uma vez que essa transmissão impõe requisitos elevados ao nível dos diferentes parâmetros de qualidade de serviço associados a uma dada rede, tais como a taxa de transmissão, o atraso, a variação do atraso ou a taxa de perda de pacotes.

Esta dissertação pretende efetuar o estudo sobre a transmissão de vídeo em tempo real sobre redes emalhadas, seguindo uma abordagem baseada no desenvolvimento de testbeds cuja complexidade foi aumentando com o decorrer do trabalho. Os testbeds desenvolvidos tiveram como elemento central pequenos computadores designados de Raspberry Pi model B, que desempenharam os papéis quer de nós de rede, quer de terminais de aquisição de vídeo. Estes testbeds percorreram vários tipos de redes e configurações, nomeadamente redes com um ponto de acesso, redes ad-hoc com diferentes números de nós, culminando o estudo com as pretendidas redes mesh.

Abstract

The technological development in telecommunications has undergone a marked evolution in recent years, focusing its research currently in the area of mobile communications. In this area, the bet has been in the identification and development of new types of networks essentially appropriate for environments that are not usually covered by traditional telecom operators.

Mesh networks arise in this context. Developed from the known ad-hoc networks, mesh networks promise innovation essentially in organization and internet sharing on a network.

Since the appearance of video, as an information storage and transmission format, that its real-time transmission is a target aspect of study. This is mainly because video real-time transmission imposes high requirements on different quality of service parameters associated with a given network, such as transmission rate, delay, jitter or packet loss rate.

This dissertation aims to make the study on the transmission of real-time video over mesh networks, following an approach based on the development of testbeds whose complexity has been increasing over the course of the work. The testbeds have as central elements small computers named Raspberry Pi model B, who played the roles of both the network nodes and the video capture terminals. These testbeds toured various types of networks and configurations, namely networks with an Access Point, ad-hoc networks with different number of nodes, finishing the study with the desired mesh networks.

Agradecimentos

Esta dissertação é um culminar de uma grande caminhada, onde foram enfrentados medos, desafios e obstáculos. No entanto, para ultrapassar todas estas situações, existiu um enorme esforço e dedicação, mas sem dúvida que o apoio de todos foi essencial.

Agradeço desta maneira a todos aqueles que me apoiaram de uma forma direta ou indireta.

Em primeiro lugar, gostaria de agradecer aos meus orientadores o Professor Manuel Alberto Pereira Ricardo e o Professor José Manuel Soares Oliveira pela partilha de saberes, valores e por acreditarem em mim.

Em segundo lugar gostaria de agradecer aos meu pais por terem sido os meus pilares para a concretização de todos os objetivos, sempre com muito carinho, apoio, amor, auxílio e dedicação.

Em terceiro lugar à minha namorada Marta pelas suas palavras de incentivo e força, pela sua compreensão nos momentos difíceis, pelo pensamento positivo e por acreditar em mim na concretização da dissertação.

Em quarto ao Paulo Jorge Lopes Soares Vaz que esteve sempre presente nos momentos mais difíceis ajudando a atravessar os diferentes obstáculos.

Hugo Melchior Silva de Gouveia e Silva

*‘Qualquer que seja o Futuro,
continuará a haver noites de Luar,
Serra de Sintra e o Tejo a correr para mar.’*

Sr. Professor José Hermano Saraiva

Conteúdo

1	Introdução	1
1.1	Descrição do projecto	2
1.2	Objetivos da dissertação	3
1.3	Estrutura da dissertação	3
2	Estado da arte	5
2.1	Redes emalhadas	5
2.1.1	Qualidade de serviço nas redes emalhadas	6
2.1.2	Projetos universitários	7
2.1.3	Soluções comerciais	8
2.2	Norma IEEE 802.11	8
2.2.1	IEEE 802.11a	8
2.2.2	IEEE 802.11b	9
2.2.3	IEEE 802.11g	9
2.2.4	IEEE 802.11n	9
2.2.5	Comparação das normas IEEE 802.11a, IEEE 802.11b, IEEE 802.11g e IEEE 802.11n	9
2.2.6	IEEE 802.11aa	10
2.3	Transmissão de vídeo	10
2.3.1	<i>Codecs</i> de vídeo	10
2.3.2	Protocolos de transmissão de vídeo	11
3	Tecnologias	15
3.1	Nós de rede e terminais de acesso	15
3.1.1	<i>Raspberry Pi</i>	15
3.1.2	Sistemas operativos	17
3.2	Comunicação sem fios	18
3.2.1	<i>TP-LINK</i> modelo <i>TL-WN722N</i>	18
3.3	Aquisição de vídeo	19
4	Etapas do Projeto	21
4.1	1ª Etapa - Instalação do terminal de aquisição de vídeo (Terminal de acesso)	21
4.1.1	Instalação do <i>Raspberry Pi</i>	21
4.1.2	Instalação do sistema operativo Raspbian	22
4.1.3	Instalação da câmara	23
4.2	2ª Etapa - <i>Streaming</i> de vídeo entre o terminal de acesso e o recetor (Ligação direta via cabo <i>ethernet</i>)	23
4.3	3ª Etapa - Ligação sem fios entre o terminal de acesso e o recetor	24

4.4	4ª Etapa - Instalação de um nó de rede	24
4.5	5ª Etapa - Rede <i>ad-hoc</i> com um nó	25
4.6	6ª Etapa - Rede <i>ad-hoc</i> com dois nós	25
4.7	7ª Etapa - Rede emalhada	26
5	Análise de resultados	29
5.1	<i>Testbeds</i>	29
5.1.1	<i>Testbed 1</i> - Comunicação sem fios através de <i>router</i> pessoal	29
5.1.2	<i>Testbed 2</i> - Comunicação sem fios através de um <i>Access Point</i>	30
5.1.3	<i>Testbed 3</i> - Rede <i>ad-hoc</i> com ligação direta entre os terminais (1 nó)	30
5.1.4	<i>Testbed 4</i> - Rede <i>ad-hoc</i> (1 nó) com 2 <i>hops</i>	31
5.1.5	<i>Testbed 5</i> - Rede <i>ad-hoc</i> com ligação direta entre os terminais (2 nós)	32
5.1.6	<i>Testbed 6</i> - Rede <i>ad-hoc</i> (2 nós) com 3 <i>hops</i>	32
5.1.7	<i>Testbed 7</i> - Rede emalhada (2 nós)	33
5.1.8	<i>Testbed 8</i> - Rede emalhada (2 nós), mas com distância acentuada entre os nós da rede	35
5.2	Cenários de teste	39
5.2.1	Cenário (a) - Cenário de referência	39
5.2.2	Cenário (b) - Cenário calmo	39
5.2.3	Cenário (c) - Cenário movimentado	40
5.3	Resultados	40
5.3.1	<i>Testbed 1</i> - Comunicação sem fios através de <i>router</i> pessoal	40
5.3.2	<i>Testbed 2</i> - Comunicação sem fios através de <i>Access Point</i>	44
5.3.3	<i>Testbed 3</i> - Rede <i>ad-hoc</i> com ligação direta entre os terminais (1 nó)	47
5.3.4	<i>Testbed 4</i> - Rede <i>ad-hoc</i> (1 nó) com 2 <i>hops</i>	49
5.3.5	<i>Testbed 5</i> - Rede <i>ad-hoc</i> com ligação direta entre os terminais (2 nós)	51
5.3.6	<i>Testbed 6</i> - Rede <i>ad-hoc</i> (2 nós) com 3 <i>hops</i>	54
5.3.7	<i>Testbed 7</i> - Rede emalhada (2 nós)	56
5.3.8	<i>Testbed 8</i> - Rede emalhada (2 nós), mas com distância acentuada entre os nós da rede	59
6	Conclusões	63
6.1	Revisão do trabalho desenvolvido	63
6.2	Contribuições	64
6.3	Trabalho futuro	64
	Referências	66

Lista de Figuras

2.1	Rede emalhada [1]	6
2.2	Codificação de um transmissão de vídeo	10
2.3	Exemplo de transmissão de vídeo	12
2.4	Comunicação com uso dos protocolos RTP e RTSP	12
2.5	Objetivos do uso dos protocolos RTP e RTSP	13
3.1	<i>Raspberry Pi Model B</i>	16
3.2	Especificações do <i>Raspberry Pi Model B</i>	16
3.3	<i>Raspberry Pi Model A</i>	17
3.4	Interface do sistema operativo <i>Raspbian</i>	18
3.5	Linha de comandos do sistema operativo <i>OpenWRT</i>	18
3.6	Antena <i>TP-LINK</i> modelo <i>TL-WN722N</i>	19
3.7	Módulo câmara <i>Raspberry Pi</i>	19
3.8	Modelo B com câmara	20
3.9	Câmara <i>Pi NoIR</i>	20
4.1	Equipamentos que compõem o <i>Raspberry Pi</i>	22
4.2	Escolha do sistema operativo <i>Raspbian</i>	22
4.3	Ligação via cabo <i>Ethernet</i> entre o computador e o <i>Raspberry Pi</i>	23
4.4	Primeiro teste de <i>streaming</i> de vídeo	24
4.5	Ligação sem fios entre o computador e o <i>Raspberry Pi</i>	24
4.6	Nó de rede - <i>Raspberry Pi</i> contendo o sistema operativo <i>OpenWrt</i>	25
4.7	Comunicação através de um <i>Access Point</i>	25
4.8	Comunicação <i>ad-hoc</i>	26
4.9	Comunicação <i>ad-hoc</i> com dois nós	26
4.10	Rede <i>Mesh</i> com dois nós	27
5.1	Caraterização do <i>Testbed 1</i> - Comunicação sem fios através de <i>router</i> pessoal	30
5.2	Caraterização do <i>Testbed 2</i> - Comunicação sem fios através de <i>Access Point</i>	30
5.3	Caraterização do <i>Testbed 3</i> - Rede <i>ad-hoc</i> com ligação direta entre os terminais (1 nó)	31
5.4	Comando <i>traceroute</i> que comprova a comunicação através de 2 <i>hops</i>	31
5.5	Caraterização do <i>Testbed 4</i> - Rede <i>ad-hoc</i> (1 nó) com 2 <i>hops</i>	32
5.6	Caraterização do <i>Testbed 5</i> - Rede <i>ad-hoc</i> com ligação direta entre os terminais (2 nós)	32
5.7	Comando <i>traceroute</i> que comprova a comunicação através de 3 <i>hops</i>	33
5.8	Caraterização do <i>Testbed 6</i> - Rede <i>ad-hoc</i> (2 nós) sem ligação direta dos terminais	33
5.9	Página fornecida pela aplicação <i>luci</i> para a configuração do protocolo <i>OLSR</i>	34
5.10	A interface <i>wireless</i> com o modo rede emalhada	34

5.11	Visualização num nó dos outros elementos presentes na rede	35
5.12	Caraterização do <i>Testbed 7</i> - Rede emalhada com dois nós e com ligação direta dos terminais	35
5.13	Os elementos da rede que utilizam o protocolo <i>OLSR</i>	36
5.14	Detalhes da ligação do recetor	37
5.15	Comando <i>traceroute</i> no emissor	37
5.16	Detalhes da ligação do recetor com <i>IP</i> estático	38
5.17	Caraterização do <i>Testbed 8</i>	38
5.18	Comando <i>traceroute</i> presente no emissor em que se verifica uma redução drástica no tempo de resposta	39
5.19	Detalhes dos pacotes <i>OLSR</i> através da plataforma <i>Wireshark</i>	39
5.20	<i>Testbed 1</i> - Cenário (a) - Sinal do emissor	41
5.21	<i>Testbed 1</i> - Cenário (a) - Sinal do recetor	41
5.22	<i>Testbed 1</i> - Cenário (b) - Sinal do emissor	41
5.23	<i>Testbed 1</i> - Cenário (b) - Sinal do Recetor	42
5.24	<i>Testbed 1</i> - Cenário (c) - Sinal do emissor	42
5.25	<i>Testbed 1</i> - Cenário (c) - Sinal do recetor	42
5.26	Valores do Cenário sem nada para o <i>testbed 1</i>	43
5.27	Valores do Cenário calmo para o <i>testbed 1</i>	44
5.28	Valores do Cenário movimentado para o <i>testbed 1</i>	44
5.29	<i>Testbed 2</i> - Cenário (a) - Sinal do emissor	44
5.30	<i>Testbed 2</i> - Cenário (a) - Sinal do recetor	45
5.31	<i>Testbed 2</i> - Cenário (b) - Sinal do emissor	45
5.32	<i>Testbed 2</i> - Cenário (b) - Sinal do recetor	45
5.33	<i>Testbed 2</i> - Cenário (c) - Sinal do emissor	46
5.34	<i>Testbed 2</i> - Cenário (c) - Sinal do recetor	46
5.35	<i>Testbed 3</i> - Cenário (a) - Sinal do emissor	47
5.36	<i>Testbed 3</i> - Cenário (a) - Sinal do recetor	47
5.37	<i>Testbed 3</i> - Cenário (b) - Sinal do emissor	47
5.38	<i>Testbed 3</i> - Cenário (b) - Sinal do recetor	48
5.39	<i>Testbed 3</i> - Cenário (c) - Sinal do emissor	48
5.40	<i>Testbed 3</i> - Cenário (c) - Sinal do recetor	48
5.41	<i>Testbed 4</i> - Cenário (a) - Sinal do emissor	49
5.42	<i>Testbed 4</i> - Cenário (a) - Sinal do recetor	49
5.43	<i>Testbed 4</i> - Cenário (b) - Sinal do emissor	50
5.44	<i>Testbed 4</i> - Cenário (b) - Sinal do recetor	50
5.45	<i>Testbed 4</i> - Cenário (c) - Sinal do emissor	50
5.46	<i>Testbed 4</i> - Cenário (c) - Sinal do recetor	51
5.47	<i>Testbed 5</i> - Cenário (a) - Sinal do emissor	51
5.48	<i>Testbed 5</i> - Cenário (a) - Sinal do recetor	52
5.49	<i>Testbed 5</i> - Cenário (b) - Sinal do emissor	52
5.50	<i>Testbed 5</i> - Cenário (b) - Sinal do recetor	52
5.51	<i>Testbed 5</i> - Cenário (c) - Sinal do emissor	53
5.52	<i>Testbed 5</i> - Cenário (c) - Sinal do recetor	53
5.53	<i>Testbed 6</i> - Cenário (a) - Sinal do emissor	54
5.54	<i>Testbed 6</i> - Cenário (a) - Sinal do recetor	54
5.55	<i>Testbed 6</i> - Cenário (b) - Sinal do emissor	54
5.56	<i>Testbed 6</i> - Cenário (b) - Sinal do recetor	55

5.57 <i>Testbed 6</i> - Cenário (c) - Sinal do emissor	55
5.58 <i>Testbed 6</i> - Cenário (c) - Sinal do recetor	55
5.59 <i>Testbed 7</i> - Cenário (a) - Sinal do emissor	56
5.60 <i>Testbed 7</i> - Cenário (a) - Sinal do recetor	57
5.61 <i>Testbed 7</i> - Cenário (b) - Sinal do emissor	57
5.62 <i>Testbed 7</i> - Cenário (b) - Sinal do recetor	57
5.63 <i>Testbed 7</i> - Cenário (c) - Sinal do emissor	58
5.64 <i>Testbed 7</i> - Cenário (c) - Sinal do recetor	58
5.65 <i>Testbed 8</i> - Cenário (a) - Sinal do emissor	59
5.66 <i>Testbed 8</i> - Cenário (a) - Sinal do recetor	59
5.67 <i>Testbed 8</i> - Cenário (b) - Sinal do emissor	60
5.68 <i>Testbed 8</i> - Cenário (b) - Sinal do recetor	60
5.69 <i>Testbed 8</i> - Cenário (c) - Sinal do emissor	60
5.70 <i>Testbed 8</i> - Cenário (c) - Sinal do recetor	61

Lista de Tabelas

2.1	Valores comparativos entre as normas 802.11a, 802.11b, 802.11g e 802.11n [2].	9
3.1	Especificações da antena <i>TP-LINK</i> modelo <i>TL-WN722N</i>	19
5.1	<i>Testbed 1</i> - Cenário (a) - Valores dos parâmetros de rede	41
5.2	<i>Testbed 1</i> - Cenário (b) - Valores dos parâmetros de rede	42
5.3	<i>Testbed 1</i> - Cenário (c) - Valores dos parâmetros de rede	42
5.4	<i>Testbed 2</i> - Cenário (a) - Valores dos parâmetros de rede	45
5.5	<i>Testbed 2</i> - Cenário (b) - Valores dos parâmetros de rede	46
5.6	<i>Testbed 2</i> - Cenário (c) - Valores dos parâmetros de rede	46
5.7	<i>Testbed 3</i> - Cenário (a) - Valores dos parâmetros de rede	47
5.8	<i>Testbed 3</i> - Cenário (b) - Valores dos parâmetros de rede	48
5.9	<i>Testbed 3</i> - Cenário (c) - Valores dos parâmetros de rede	49
5.10	<i>Testbed 4</i> - Cenário (a) - Valores dos parâmetros de rede	49
5.11	<i>Testbed 4</i> - Cenário (b) - Valores dos parâmetros de rede	50
5.12	<i>Testbed 4</i> - Cenário (c) - Valores dos parâmetros de rede	51
5.13	<i>Testbed 5</i> - Cenário (a) - Valores dos parâmetros de rede	52
5.14	<i>Testbed 5</i> - Cenário (b) - Valores dos parâmetros de rede	53
5.15	<i>Testbed 5</i> - Cenário (c) - Valores dos parâmetros de rede	53
5.16	<i>Testbed 6</i> - Cenário (a) - Valores dos parâmetros de rede	54
5.17	<i>Testbed 6</i> - Cenário (b) - Valores dos parâmetros de rede	55
5.18	<i>Testbed 6</i> - Cenário (c) - Valores dos parâmetros de rede	56
5.19	<i>Testbed 7</i> - Cenário (a) - Valores dos parâmetros de rede	57
5.20	<i>Testbed 7</i> - Cenário (b) - Valores dos parâmetros de rede	58
5.21	<i>Testbed 7</i> - Cenário (c) - Valores dos parâmetros de rede	58
5.22	<i>Testbed 8</i> - Cenário (a) - Valores dos parâmetros de rede	59
5.23	<i>Testbed 8</i> - Cenário (b) - Valores dos parâmetros de rede	60
5.24	<i>Testbed 8</i> - Cenário (c) - Valores dos parâmetros de rede	61

Abreviaturas e Símbolos

APs	Access Points
BSSID	Basic Service Set Identification
CD	Compact Disk
CPE	Customer Premises Equipment
DSSS	(Direct Sequence Spread Spectrum)
DVD	Digital Versatile Disk
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISO	International Organization for Standardization
ITU-T	International Telecommunication Union
MAC	Medium Access Control
MIT	Massachusetts Institute of Technology
MMC	Multi Media Card
MPEG	Moving Picture Experts Group
NOOBS	New Out Of Box System
NTSC	National Television System Committee
OLSR	Optimized Link State Routing
PAL	Phase Alternating Line
PHY	Physical
QoS	Quality of Service
RTCP	Real-Time Transport Control Protocol
RTP	Real-Time Transport Protocol
RTSP	Real-Time Streaming Protocol
SD	Secure Digital
SDHC	Secure Digital High Capacity
SDP	Session Description Protocol
SDXC	Secure Digital Extended Capacity
SECAM	Séquential Couleur à Mémoire
SSH	Secure Shell
TCP	Transmission Control Protocol
UCSB	University of California at Santa Barbara
UDP	User Datagram Protocol
VCEG	Video Coding Experts Group
VHS	Video Home System
VoIP	Voice over Internet Protocol

Capítulo 1

Introdução

Nos últimos anos temos assistido a diferentes desenvolvimentos tecnológicos motivados principalmente pela crescente vontade de satisfazer as mais variadas necessidades da humanidade. Uma necessidade básica do homem sempre foi a de comunicar entre si. A comunicação entre os homens tem evoluído de forma acentuada no último século. Inicialmente com a comunicação de voz sobre a rede telefónica fixa. Mais recentemente com a *Internet*, as redes móveis e o aumento crescente da capacidade dessas redes. Atualmente é cada vez mais habitual pretender comunicar para além da voz também dados de vídeo, não só sobre redes previamente estruturadas mas também sobre redes circunstanciais ou oportunistas. Neste sentido, esta dissertação insere-se no âmbito da transmissão de dados de vídeo em tempo real sobre redes *Wireless Mesh*.

As redes *Wireless Mesh* (*Wireless Mesh Networks*) são um exemplo específico das *redes ad-hoc*. Quando um grupo de dispositivos de rede (nós) pretende partilhar pacotes de informação, sem ter que aceder a uma estrutura fixa e pré-determinada, forma aquilo que habitualmente se designa por uma rede *ad-hoc* (possivelmente móvel). Estas redes são utilizadas em situações em que uma estrutura de rede não se encontra disponível ou não se tem a certeza da segurança dessa mesma estrutura ou ainda em situações de emergência. As suas aplicações cobrem áreas tão diversas como a área militar (busca e salvamento em situações de desastre), a área empresarial (redes de tempo limitado presentes em salas de reuniões), a área da aeronáutica (aeroportos), entre outras áreas. As redes *Wireless Mesh* face às redes *ad-hoc* apresentam uma estrutura fixa (não sendo pré-determinada), em que os nós se encontram numa topologia em malha e que se organizam e se configuram automaticamente permitindo a ligação à rede de dispositivos móveis, utilizando interfaces sem fios, reduzindo desta forma a sua complexidade [3]. Para além disso, são redes *Wireless Multi-Hop*, isto é, para a comunicação dos dados é necessário que estes atravessem diferentes nós intermédios, antes de alcançar o destino desejado. Neste sentido as redes *Wireless Mesh* vão ter um lugar de destaque num futuro próximo, pois são caracterizadas principalmente pela sua simplicidade e baixo custo [4].

Como as redes *Wireless Mesh* são antes demais redes *wireless*, elas operam usando o *standard*

IEEE 802.11. Este *standard*, criado pelo IEEE, é conhecido no senso comum como *Wi-Fi*, sendo nos dias de hoje a forma de acesso à *Internet* de maior sucesso [2]. As redes baseadas no *standard* IEEE 802.11 caracterizam-se por fornecer acesso à *Internet* em “qualquer lugar” e a “qualquer hora”, permitindo aos dispositivos ligados a essas redes a característica habitualmente designada por *always on*. Os pontos de acesso (APs) IEEE 802.11 têm sido instalados exaustivamente no universo tecnológico mundial, seja por organizações, fundações, particulares e até mesmo cidades, para oferecer à sociedade *Internet* sem fios, podendo ter acesso através dos computadores, *smartphones*, equipamento industrial, entre outros dispositivos que tenham interfaces sem fios. O IEEE 802.11 apresenta um alcance rádio limitado, ou seja, a cobertura de grandes áreas geográficas apenas é atingida através da instalação de um número elevado de pontos de acesso. Para redes de enorme dimensão deve-se pôr de lado a hipótese de ter ligações por fio entre pontos de acesso, pois a estrutura que lhe vai estar associada será muito complexa [4].

Recentemente, diversos sistemas de telecomunicações suportam cada vez mais transmissões em tempo real, sendo que a transmissão de vídeo tem um papel fulcral. Devido a esta situação, a qualidade de serviço no suporte de vídeo tornou-se numa questão essencial. No mundo da investigação encontramos diversos artigos sobre mecanismos presentes nas diferentes redes que suportam a qualidade de serviço (QoS). No entanto, muitos investigadores restringem-se apenas a provar que o mecanismo que propõem tem a possibilidade de reduzir a taxa de perda de pacotes (*packet loss rate*), o atraso do pacote (*packet delay*) ou o *jitter* do pacote (*packet jitter*), considerando apenas estes parâmetros para a caracterização da qualidade da transmissão de vídeo. Na verdade, os parâmetros mencionados não podem ser facilmente e unicamente transformados em qualidade de transmissão de vídeo, mas sim esta transformação pode ser diferente para cada esquema de código, método de perda por ocultação (*loss concealment scheme*) e manipulação *delay/jitter* (*delay/jitter handling*)[5].

1.1 Descrição do projecto

Nas últimas décadas, as câmaras de vídeo-vigilância e as redes de câmaras de vídeo têm-se tornado ubíquas e pervasivas, cobrindo lugares tão distintos como centros comerciais, estações de transportes ou vias de comunicação. Os sistemas de monitorização atuais não permitem atingir o grau de autonomia desejado, continuando a ser operados por pessoas. Em consequência, estes sistemas consistem tipicamente num número reduzido de câmaras instaladas de forma a que o espaço coberto seja maximizado ou de forma a cobrir preferencialmente áreas críticas.

A diminuição de preço e os avanços tecnológicos ocorridos recentemente nas câmaras de vídeo permitem um novo paradigma: a cobertura de grandes áreas por câmaras de alta qualidade, normalmente caras, pode ser substituída por múltiplas câmaras mais baratas, cada uma cobrindo uma parte da área total com maior detalhe. Para além de aumentar o detalhe de cobertura, esta abordagem também contribui para minimizar o número de zonas invisíveis induzidas por obstáculos. Este novo paradigma é também apoiado pela forte tendência, observada atualmente, na

investigação de soluções para o processamento automático de dados de vídeo, minimizando a necessidade dos operadores visualizarem toda a informação proveniente das câmaras.

Neste novo paradigma, a utilização de tecnologia de redes sem fios para interligação das várias câmaras surge como a mais adequada, permitindo uma maior facilidade e flexibilidade de instalação de novos nós. Um sistema formado por estas pequenas câmaras interligadas entre si, pode ser visto como uma rede sem fios de sensores de vídeo. Um exemplo de contexto de utilização natural deste tipo de rede de sensores de vídeo será o da monitorização de parques de estacionamento de grandes dimensões.

O aumento do número de câmaras leva a que a rede tenha que ser flexível e que suporte o envio de um grande número de fluxos de vídeo para um centro de monitorização, onde o vídeo é armazenado, processado e analisado. As redes sem fios *Mesh* IEEE 802.11, pela sua flexibilidade, constituem uma solução atrativa que permite a instalação/remoção fácil e económica de câmaras, a instalação em lugares menos acessíveis, cobrindo áreas geográficas maiores, e a auto-organização das câmaras de forma a manterem o contacto com o centro de monitorização.

1.2 **Objetivos da dissertação**

Na descrição efetuada anteriormente foi definido como principal objetivo o estudo da viabilidade técnica da transmissão de vídeo em tempo real produzido por câmaras de vigilância através de uma rede *Wireless Mesh* baseada no *standard* IEEE 802.11, onde cada nó possui duas interfaces de rede. Mais especificamente, pretende-se efetuar o estudo, a definição, a implementação e a avaliação de uma solução de comunicações para uma rede de sensores de vídeo que alcance um bom desempenho nos diferentes parâmetros de rede, sendo os mais críticos na transmissão de vídeo em tempo real, a capacidade da rede, o atraso, a variação do atraso (jitter) e a taxa de perda de pacotes.

1.3 **Estrutura da dissertação**

Esta dissertação encontra-se organizada em 6 capítulos.

O Capítulo 2 apresenta o estado da arte das áreas do projeto proposto, baseado na revisão de artigos e dissertações. Neste capítulo, é inicialmente descrita a topologia e o tipo de rede, de seguida a análise do *standard* IEEE 802.11 e por fim a análise e descrição dos *codecs* e protocolos para transmissão de vídeo.

O Capítulo 3 identifica e descreve as tecnologias que foram importantes no desenvolvimento da dissertação.

O Capítulo 4 identifica e descreve as etapas que permitiram a conclusão da dissertação.

O Capítulo 5 apresenta os resultados obtidos nas diferentes etapas da dissertação.

Por fim, o Capítulo 6 apresenta as conclusões obtidas.

Capítulo 2

Estado da arte

Neste capítulo é feita uma descrição do estado da arte nas áreas relacionadas com o problema proposto nesta dissertação. Inicialmente são descritas as principais características das redes *Mesh* e são dados alguns exemplos de soluções comerciais e soluções propostas por grupos de investigação para este tipo de redes, o IEEE 802.11. De seguida, será analisada a norma de rede que suporta as redes emalhadas. Por fim, será feita uma breve descrição dos *codecs* e protocolos para a implementação do serviço de *streaming* de vídeo.

Com este estudo iremos identificar os pontos fortes e os pontos fracos de todos os recursos que irão ser utilizados durante o desenvolvimento do projeto e ter uma ideia definida e concreta acerca dos requisitos e desafios na realização das diferentes vertentes deste trabalho.

2.1 Redes emalhadas

O aparecimento das redes emalhadas inverteu o paradigma do uso de fios para construir a espinha dorsal (*backbone*) da rede. Neste sentido o *backbone* da rede *Mesh* é construída sem qualquer fio, como é ilustrado na Figura 2.1, em que os terminais da rede podem ser acedidos por cabo ou por *Wi-Fi*.

Uma das vantagens é que os nós presentes no *backbone*, como têm uma localização fixa, não possuem limitações de energia, permitindo desta forma suprimir grande parte das restrições das redes *ad-hoc*.

Diversos projetos sobre redes *Mesh* têm surgido nos últimos anos, sendo desenvolvidos por universidades e centros de investigação e desta maneira são implementadas principalmente em *campus* para serem acedidos por utilizadores nas proximidades. Encontramos a nível mundial diversos projetos de redes do tipo *mesh* como o *RoofNet* no MIT, o *Google Mesh* na Califórnia, o *VMesh* na Grécia, o *MeshNet* na UCSB, o *Microsoft Mesh*, entre outros.

O ideal presente no desenvolvimento das redes *Mesh* é o da construção das cidades do futuro, fornecendo o acesso livre às pessoas que não têm condições para ter uma ligação de banda larga

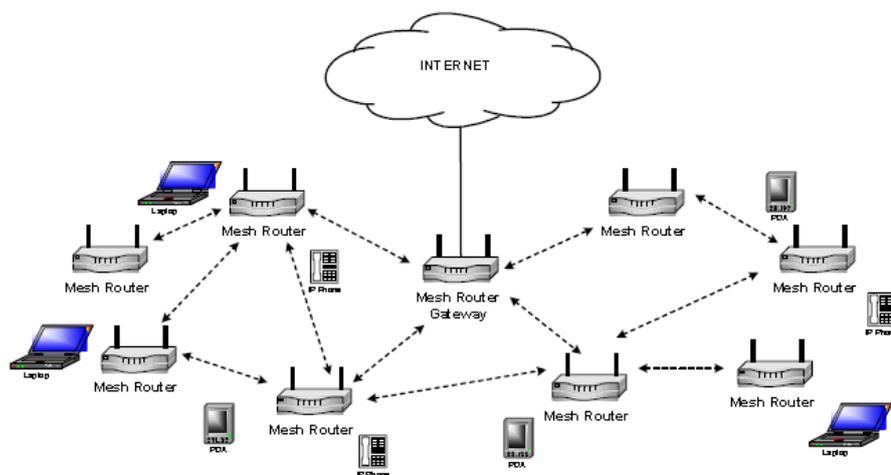


Figura 2.1: Rede emalhada [1]

tradicional do tipo xDSL ou cabo, que têm um custo tradicionalmente elevado.

As aplicações multimídia criaram um interesse acentuado na investigação, pois estas apresentam novos desafios, bastante característicos, como o fornecimento da qualidade de serviço (QoS) e a comunicação em grupo (*multicasting*). Estes desafios tornam a investigação mais complexa devido a fatores como a própria mobilidade das estações, que implica uma gestão cuidada das localizações das estações, como as limitações do meio ambiente e dos aparelhos envolvidos e por fim a qualidade da transmissão num ambiente sem fios [1].

2.1.1 Qualidade de serviço nas redes emalhadas

A Qualidade de Serviço (QoS) é o resultado da união de exigências dos serviços ao qual a rede, que é responsável pelo transporte do fluxo de dados entre o emissor e o recetor, deve responder. Isto é, o que as redes oferecem para garantir determinados serviços com certos tipos de tráfegos, através de tecnologias específicas.

Dos diversos parâmetros que a qualidade de serviço controla, a largura de banda, o atraso, a variação do atraso (*jitter*), a taxa de perdas, a segurança, a prioridade, a resistência, são os que têm maior influência nas aplicações multimídia em tempo real. Os parâmetros revelados anteriormente são negociados entre o utilizador do serviço e a entidade que o presta, surgindo desta maneira contratos de nível de serviço (SLAs - *Service Level Agreements*).

O papel cada vez mais crítico assumido pela qualidade de serviço tornou-se um requisito necessário no desenvolvimento das redes sem fios, isto porque diversas aplicações multimídia usam *Internet* de banda larga, sendo estas bastante afetadas por atrasos na rede, para além de precisarem da maior banda possível [1].

2.1.1.1 Parâmetros de qualidade na transmissão de vídeo em tempo real

Para o sucesso deste projeto, em particular a transmissão de vídeo em tempo real, é necessário garantir o cumprimento de alguns parâmetros que a seguir vão ser apresentados. Estes parâmetros permitem a emissão de vídeo sem falhas, não esquecendo que esses parâmetros estão aglomerados no QoS da rede.

- **Débito Binário** — produção de um fluxo de dados a um ritmo constante ou a ritmo variável. A redução do ritmo na produção de fluxo pode levar a resultados que não se pretende ao nível da qualidade.
- **Degradação do sinal** — um sinal que sofra mudanças constantes de qualidade, durante a comunicação, leva à insatisfação dos utilizadores. Este facto é devido a erros na transmissão, sendo necessário ter preparadas estratégias para a contenção destes erros.
- **Atraso** — devido á codificação do vídeo este parâmetro assume um papel fundamental. Neste sentido é necessário limitá-lo e garantir que não existem variações numa qualidade mais elevada.
- **Jitter** — é a garantia de que a informação vai ser processada num determinado intervalo de tempo. Isto é, variação do tempo de atraso consoante os atrasos não constantes introduzidos, devido aos diferentes mecanismos que compõem o sistema.
- **Perdas** — garantia de que o sistema não perca pacotes, pois irá afetar negativamente a transmissão e a qualidade.

Todos estes parâmetros variam com o tipo de rede e com aparelhos utilizados para a sua construção. Ignorá-los seria um erro tremendo, pois tais parâmetros asseguram a satisfação dos utilizadores [2].

2.1.2 Projetos universitários

Como já foi mencionado anteriormente, diversos projetos universitários têm surgido e até mesmo com apoios de grandes empresas na área do desenvolvimento tecnológico, para promover o uso da *Internet* gratuito e na procura do desenvolvimento de cidades inteligentes no futuro. Em seguida irão ser mencionados três projetos que têm tido uma forte divulgação a nível internacional [6].

2.1.2.1 RoofNet

O projeto *RoofNet* consiste numa arquitetura com mais do que quarenta nós, com localização fixa, dispersados por alguns quilómetros quadrados numa cidade, em que cada nó contém um computador com um determinado *software*, uma placa de rádio com o *standard* 802.11b e uma antena instalada no telhado. Este projeto foi criado com o objetivo de compreender a construção

de uma melhor rede sem fios comunitária e cooperativa, não esquecendo o princípio das redes *Mesh* que é sua fácil implementação [7].

2.1.2.2 MesNet

O projeto *MeshNet* é construído com uma mistura de *routers Wireless IEEE 802.11 b/g* e de computadores equipados com interfaces de rede sem fios *IEEE 802.11 a/b/g*. Os trinta nós estão espalhados pelos escritórios e laboratórios do *campus* da UCSB [8].

2.1.2.3 Microsoft Mesh

Microsoft Mesh é constituído por vinte e três nós, primariamente estes são constituídos por computadores portáteis e que correm o *software Microsoft Windows XP* e por uma placa de rádio com *standard 802.11a* [9].

2.1.3 Soluções comerciais

A grande desvantagem das soluções comerciais em relação às soluções propostas pelos projetos universitários é o facto dos *routers mesh* terem um custo elevado. Em contrapartida diversos governos procuram essas soluções para a construção das cidades digitais, como por exemplo *Dublin* ou *Taipei* onde encontramos os produtos da *Nortel*. Mais recentemente, os produtos da *Cisco* estão a ser usados na cidade *Tiradentes* no Brasil [6].

2.1.3.1 Nortel

Diversos produtos da *Nortel* foram lançados para o mercado, incluindo o *wireless access point* (WAP) que é um sistema *dual-radio* que suporta um *link* de acesso de 2,4 GHz e um *link* de fluxo de 5 GHz, equipado com antenas *smart*. Este WAP utiliza uma camada física com *standard 802.11a* e sobre a camada *Medium Access Control* (MAC) um protocolo *mesh* entre APs para descoberta, criação, monitorização, manutenção e restabelecimento do fluxo de informação [10].

2.1.3.2 Cisco

Os produtos da *Cisco* para as redes *mesh* foram colocados no mercado em Março de 2005. Os seus produtos incluem uma rica variedade de controlos de segurança, principalmente, nos pontos de acesso que executam o sistema operativo da *Cisco* e são compatíveis com o *standard IEEE 802.11i* [11].

2.2 Norma IEEE 802.11

2.2.1 IEEE 802.11a

O *standard IEEE 802.11a*, surgiu em 1999, utiliza uma frequência de 5 GHz que face aos outros *standards 802.11* é superior (utilizam uma gama de frequência no valor de 2,4 GHz. O seu

propósito é fornecer débitos superiores (na ordem dos 54 Mbit/s) aos 11 Mbit/s apresentados pelo *standards* 802.11b, sendo desta forma mais imune a interferências, por exemplo de microondas ou aparelhos Bluetooth. Em contrapartida, a utilização de frequências mais altas reduz o alcance e torna mais difícil o papel de manter a qualidade do sinal na presença de obstáculos, como por exemplo paredes [2].

2.2.2 IEEE 802.11b

O facto do *standard* original IEEE 802.11 ser demasiado lento (débitos de 2 Mbit/s) e estar a atrasar a evolução das redes sem fios, foi criado o *standard* IEEE 802.11b. Neste sentido, o *standard* IEEE 802.11b obtém débitos até 11 Mbit/s, tendo diferenças na camada física face ao *standard* original IEEE 802.11 o que proporciona o aumento dos débitos e melhor conectividade. Oferece na camada física duas opções com novas velocidades, 5,5 Mbit/s e 11 Mbit/s [2].

2.2.3 IEEE 802.11g

A convergência entre o *standard* IEEE 802.11a, na área dos débitos, e o *standard* IEEE 802.11b, na área da frequência com que trabalha, os 2,4 GHz, permitiu o surgimento do *standard* IEEE 802.11g. O seu principal objetivo é promover a interoperabilidade com os equipamentos do *standard* IEEE 802.11b que se encontram instalados [2].

2.2.4 IEEE 802.11n

Com a necessidade de aumentar o débito e a gama de potência surge o *standard* IEEE 802.11n. Foi criado com o intuito de suportar aplicações que necessitam de débitos elevados. Este *standard* alterou as camadas MAC e PHY para se obter um débito mínimo de 100 Mbit/s, onde se observa que os valores apresentados pelos *standard* b e g são quadruplicados. A configuração de múltiplas antenas e o aumento da largura de banda os débitos binários podem chegar aos 600 Mbit/s [2].

2.2.5 Comparação das normas IEEE 802.11a, IEEE 802.11b, IEEE 802.11g e IEEE 802.11n

Tabela 2.1: Valores comparativos entre as normas 802.11a, 802.11b, 802.11g e 802.11n [2].

Atributos	Normas			
	IEEE 802.11a	IEEE 802.11b	IEEE 802.11g	IEEE 802.11n
Débito máximo over-the-air	54 Mbit/s	11 Mbit/s	54 Mbit/s	600 Mbit/s
Banda de frequência	5 GHz	2,4 GHz	2,4 GHz	2,4 GHz e 5 GHz
Cobertura <i>indoor</i>	50 m	100 m	100 m	150 m
Canais sem sobreposição	13	3	3	3(2,4 GHz) e 23(5 GHz)

Na Tabela 2.1 encontram-se os principais aspetos de cada tecnologia. Conclui-se que o *standard* IEEE 802.11n é o que tem maior débito e o que apresenta maior cobertura. É o mais indicado

para serviços de voz e imagem de alta definição. Mas os outros *standards* como são mais antigos, são ao mesmo tempo mais baratos [2].

2.2.6 IEEE 802.11aa

Presente na camada MAC, este *standard* criado em Março de 2008 é direcionado para *streaming* vídeo/áudio, permite que se diminua a qualidade da *stream* no caso em que a capacidade do canal seja insuficiente. Através deste novo *standard* o bloqueio de material multimédia por insuficiência da capacidade do canal deixou de ser problema, a solução passou por reduzir drasticamente a qualidade de transmissão para continuar a reprodução do material multimédia [12].

2.3 Transmissão de vídeo

2.3.1 Codecs de vídeo

A representação digital de vídeo quando não está comprimida tem um elevado débito, por essa razão existe uma necessidade de a comprimir (Figura 2.2). Entende-se por codificação a representação da informação na saída do emissor por um reduzido número de símbolos que se pretende transmitir, reduzindo por sua vez a largura de banda, não esquecendo que a definição da decodificação é o processo inverso. Ao analisar as características do codificador/decodificador é importante salientar a taxa de compressão, a complexidade e o atraso presente no sinal. Para retirar as informações supérfluas ou de pouca importância, a codificação usa uma análise temporal e/ou espacial das imagens iniciais [2].

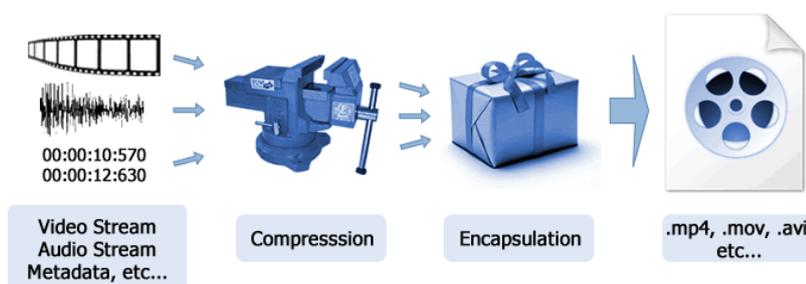


Figura 2.2: Codificação de um transmissão de vídeo

Seguidamente irão ser apresentados os diferentes *codecs*, sendo que os mais recentes são o MPEG-4 e o H.264/AVC.

- **H.261** — A norma mais utilizada para a realização de videoconferências. Como necessita de uma elevada taxa de compressão, os algoritmos de vídeo onde o nível de qualidade inferior funcionam perfeitamente em terminais com visores mais reduzidos. Quando atinge a capacidade máxima a qualidade é semelhante à de um *Video Home System* (VHS) [2].

- **H.263** — A ITU-T desenvolveu o H.261 de forma a aumentar a sua eficiência, atingindo uma nova norma, o H.263, expandindo a aplicação destas técnicas a velocidades de transmissão mais baixas para que os primeiros modems que executavam a velocidades inferiores a pudessem seguir. Surgindo em 1995, o H.263, em conjunto com H.261 apresenta um codificador híbrido, bem como um conjunto adicional de funcionalidades, quase todas opcionais [2].
- **MPEG-1** — O desenvolvimento do MPEG-1 decorreu entre 1988 e 1993, com a necessidade de codificar vídeo e áudio para armazenamento digital em *Compact Disk* (CD), com qualidade VHS [2].
- **MPEG-2** — Com o aparecimento da televisão e da televisão em alta definição, desenvolveu-se a partir do MPEG-1, o MPEG-2. Possibilitou o suporte de novas tecnologias como *Digital Versatile Disk* (DVD), transmissão digital de televisão nos formatos PAL, SECAM ou NTSC, com diferentes resoluções espaço-temporais e mais recentemente a tecnologia *Blue-Ray* que também pode utilizar esta norma. Sendo uma das normas com mais sucesso cooperou para entrada na designada “era digital” [2].
- **MPEG-4** — Surgiu em 1999 com a transmissão de televisão em redes sem fios. Devido aos novos algoritmos suporta codificações a débitos baixos, atingido uma qualidade superior em MPEG-2. Tem como principais objetivos a interatividade ao nível dos conteúdos, a eficiência na compressão e o acesso universal. Entende-se por interatividade ao nível dos conteúdos, o acesso, a gestão, a manipulação, a edição, a codificação e a composição eficiente de conteúdos, sempre com possibilidade de acesso aleatório ao conteúdo da informação audiovisual [2].
- **H.264/AVC** — Surgiu em 2003, numa parceria dos grupos MPEG da ISO/IEC e *Vídeo Coding Experts Group* (VCEG). A aplicação principal desta norma era a transmissão de vídeo em qualidade DVD com menos recursos, tendo em especial atenção o desempenho de canais com erros, nomeadamente canais móveis e canais presentes na *Internet*. O objetivo do MPEG-4 é permitir o uso por utilizadores com baixos débitos enquanto a norma H.264 é para utilizadores que têm não só recursos mais baixos como também para os que têm recursos elevados [2].

2.3.2 Protocolos de transmissão de vídeo

Para além dos *codecs* de vídeo, é também importante referir os protocolos de transmissão de vídeo (Figura 2.3). Com estes protocolos temos a possibilidade de controlar a qualidade na transmissão de vídeo. Em seguida vão ser especificados os protocolos com maior relevância.

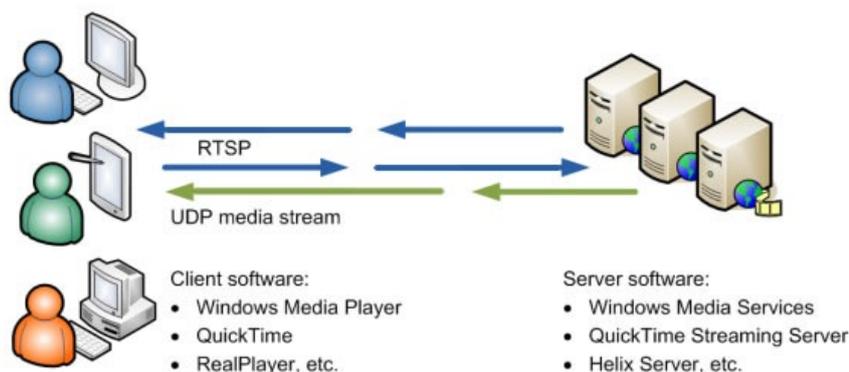


Figura 2.3: Exemplo de transmissão de vídeo

2.3.2.1 RTP e RTCP

Para a transmissão de vídeo interativo ou VoIP utiliza-se um protocolo com características de tempo real, foi definido pelo IETF e tem como designação *Real-Time Transport Protocol* (RTP) (Figura 2.4). Este protocolo adiciona aos dados um número de sequência e uma etiqueta temporal. Como não implementa nenhum mecanismo de controlo, não garante que os dados sejam entregues, permite sim, que os fluxos diferentes de informação de texto, vídeo e som sejam transferidos para o mesmo cliente. Apesar desses fluxos poderem ser transmitidos sobre diversos protocolos, o protocolo que em regra geral é o mais utilizado é o UDP. Como foi referido anteriormente o protocolo RTP não implementa um mecanismo de controlo, a solução encontrada para este problema foi a criação do protocolo *Real-Time Transport Control Protocol* (RTCP), que não transporte dados de informação, mas que usado em conjunto com o RTP assegura a implementação de um mecanismo de controlo e o transporte dos dados. Outra função é o uso periódico do protocolo RTCP para a partilha de informação sobre as características da qualidade de transmissão presente entre os equipamentos [13].

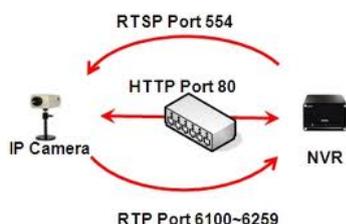


Figura 2.4: Comunicação com uso dos protocolos RTP e RTSP

2.3.2.2 RTSP

O protocolo que permite que os clientes de multimédia operem como um controlo remoto na transmissão de conteúdo que pretende visualizar, esse protocolo tem a designação de *Real-*

Time Streaming Protocol (RTSP) (Figura 2.5). Permite ao reproduzidor multimédia interagir com o servidor, modificando o modelo como o fluxo é apresentado. Fornece ao utilizador comandos, tais como, avançar, retroceder, parar ou voltar a reproduzir [13].

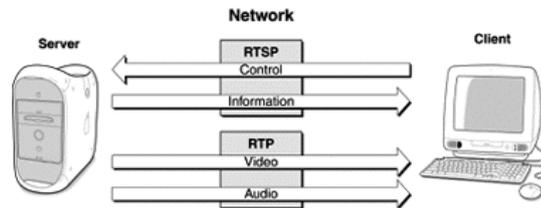


Figura 2.5: Objetivos do uso dos protocolos RTP e RTSP

2.3.2.3 SDP

Na camada de aplicação é implementado o protocolo *Session Description Protocol* (SDP), tendo como função, antes do início da transmissão, a descrição das características da sessão multimédia. Essa descrição inclui o título e a finalidade da sessão, que tipo de multimédia da transmissão, se é áudio ou vídeo, em que protocolo vai ser transportado o fluxo de dados. Ressalvando que o protocolo SDP transporta dados multimédia [13].

Capítulo 3

Tecnologias

Neste capítulo vão ser descritas as tecnologias que tiveram maior impacto no desenvolvimento do trabalho efetuado nesta dissertação. A apresentação está dividida em três secções, cada uma cobrindo respetivamente as tecnologias relacionadas com os nós de rede e terminais de acesso, as tecnologias de comunicação sem fios e as tecnologias de aquisição de vídeo.

A identificação destas tecnologias foi determinada pelo objetivo principal do trabalho, que neste caso foi estudar a transmissão de vídeo sobre redes *mesh*. Sendo uma rede *mesh* por natureza uma rede sem fios, os nós de rede deverão ser elementos com capacidade de comunicação sem fios e preferencialmente elementos móveis.

3.1 Nós de rede e terminais de acesso

3.1.1 *Raspberry Pi*

O *Raspberry Pi* é um pequeno computador com um tamanho semelhante a um cartão de crédito, mais especificamente 85.6 mm x 56 mm x 21 mm, com o peso de 45 g, ao qual se podem ligar diversos periféricos, tais como monitores, teclados ou ratos. Este equipamento é habitualmente utilizado em projetos de investigação, desempenhando o mesmo papel de um computador normal, sendo no entanto atualmente adquiridos principalmente por jovens que pretendem dar os primeiros passos na programação. Para além destas funções, o *Raspberry Pi* permite também desempenhar atividades habitualmente executadas num computador pessoal, tais como editar folhas de cálculo, processar texto ou até mesmo jogar.



Figura 3.1: *Raspberry Pi Model B*

3.1.1.1 Model B

O modelo B é uma variante do *Raspberry Pi* com especificações mais avançadas, tais como memória *RAM* de 512 MB, duas portas *USB* e uma porta *Ethernet* de 100 Mbps. É o modelo mais popular no mercado, permitindo a realização de projetos reais tão variados como máquinas de *Arcade*, tanques robôs, aprendizagem na área da computação, ser configurado como servidor *web* ou até mesmo jogar *Minecraft*. Na figura 3.2 apresentam-se os diferentes módulos que constituem a placa do modelo B.

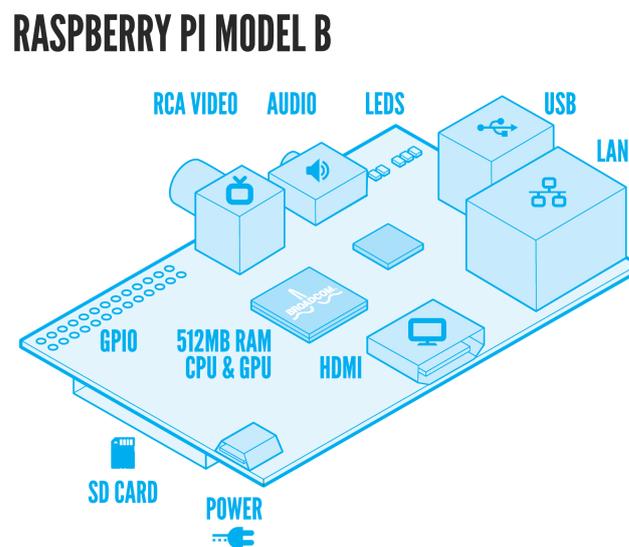


Figura 3.2: Especificações do *Raspberry Pi Model B*

3.1.1.2 Model A

O modelo A é uma variante do *Raspberry Pi* com especificações de baixo nível, mais especificamente, memória *RAM* de 256 MB e uma única porta *USB*. Uma diferença relevante para o modelo B é a ausência de porta *Ethernet*. Este modelo é utilizado para pequenos projetos devido

à falta de algumas interfaces e de um *chip* Ethernet, sendo no entanto mais leve e menos exigente em termos de energia que o modelo B. Geralmente é usado na área da robótica e em projetos que tenham requisitos de baixo peso e baixo consumo de energia. O modelo A em combinação com uma placa Wi-Fi é ideal para que o *Raspberry Pi* se comporte como um *media centre*, em conjunto com uma televisão.

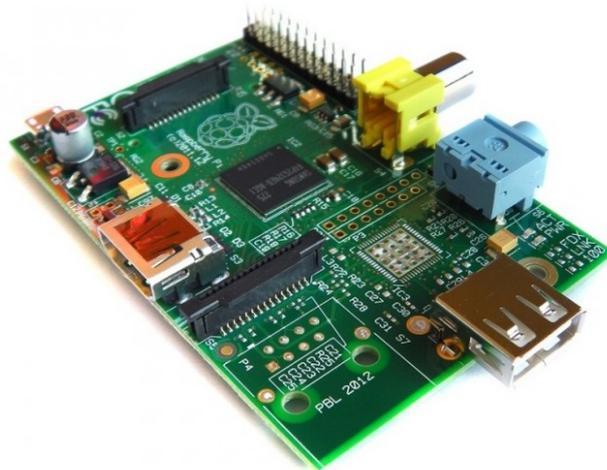


Figura 3.3: *Raspberry Pi Model A*

3.1.2 Sistemas operativos

3.1.2.1 *Raspbian*

O sistema operativo *Raspbian*, que faz parte do *installer NOOBS (New Out Of Box System)*, é baseado no sistema operativo *Debian* e foi criado especificamente para o *hardware* do *Raspberry Pi*. Este sistema operativo é constituído por um conjunto de programas e funcionalidades que permitem o funcionamento do *Raspberry Pi* na sua plenitude. Finalizado em Julho de 2012, este sistema operativo continua no entanto em desenvolvimento para se tornar cada vez mais um sistema estável e que obtenha a melhor desempenho possível.

3.1.2.2 *OpenWRT*

O *OpenWRT* é um sistema operativo baseado no *kernel* do *Linux*, sendo principalmente usado em equipamentos que gerem o tráfego de redes. Este sistema foi criado com o intuito de ocupar as memórias limitadas disponibilizadas pelos *routers* pessoais. Pode ser configurado através de uma linha de comandos (*bash shell*) ou por uma interface *web (LuCI)*. O *OpenWRT* pode ser utilizado

Tabela 3.1: Especificações da antena *TP-LINK* modelo *TL-WN722N*

Interface	USB 2.0
Normas <i>Wireless</i>	IEEE 802.11n, IEEE 802.11g, IEEE 802.11b
Frequência	2.400-2.4835GHz
Taxa de Transmissão	11n: Até 150Mbps(dinâmico), 11g: Até 54Mbps(dinâmico), 11b: Até 11Mbps(dinâmico)
Modos <i>Wireless</i>	Ad-Hoc / modo Infraestrutura
Segurança <i>Wireless</i>	64/128 bit WEP, WPA-PSK/WPA2-PSK

Figura 3.6: Antena *TP-LINK* modelo *TL-WN722N*

3.3 Aquisição de vídeo

3.3.0.1 Câmera

O módulo câmera *Raspberry Pi* permite a obtenção de fotografias e vídeo em alta-definição. A câmera tem cinco megapíxeis de *fixed-focus*, e suporta as resoluções 1080p, 720p e o modo vídeo VGA90. A ligação da câmera ao *Raspberry Pi* é feita através de um cabo de fita de 15 cm. As utilizações mais populares são a videovigilância e a captura de imagens na vida selvagem.

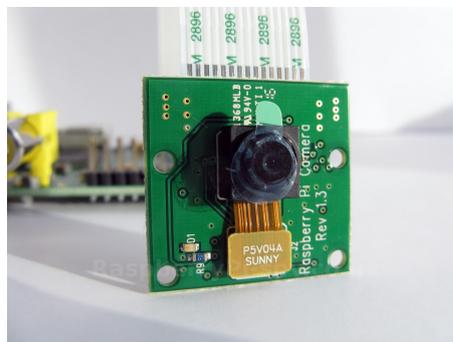
Figura 3.7: Módulo câmera *Raspberry Pi*



Figura 3.8: Modelo B com câmara

3.3.0.2 Câmera *Pi NoIR*

O *Pi NoIR* é um novo modelo do módulo câmara *Raspberry Pi*, que faz tudo o que uma câmara normal oferece, mas não aplica um filtro de luz infravermelha. A palavra *NoIR* significa neste caso *No Infrared*. As imagens retiradas durante o dia poderão não ter a qualidade mais desejável. No entanto, esta câmara tem a capacidade de ver no escuro com luz infravermelha. Este género de câmara é muito utilizada em aplicações de estudo da vida selvagem, uma vez que os LEDs infravermelhos permitem monitorizar os animais noturnos sem os perturbar.



Figura 3.9: Câmera *Pi NoIR*

Capítulo 4

Etapas do Projeto

O objetivo principal da dissertação foi estudar os aspetos mais relevantes associados à transmissão de vídeo sobre redes emalhadas, habitualmente designadas por redes *mesh*. Neste capítulo é feita a descrição das etapas que foram executadas para atingir esse objetivo.

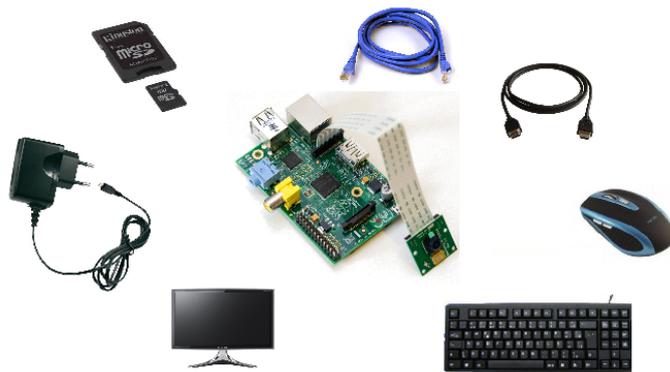
Os desenvolvimentos efetuados seguiram uma abordagem passo a passo, partindo do desenvolvimento de cenários de comunicação mais simples para cenários mais complexos, quer em termos de tecnologias de comunicação utilizadas quer em termos do número de nós envolvidos. Cada uma das próximas secções descreve os desenvolvimentos ocorridos em cada uma dessas etapas.

4.1 1ª Etapa - Instalação do terminal de aquisição de vídeo (Terminal de acesso)

4.1.1 Instalação do *Raspberry Pi*

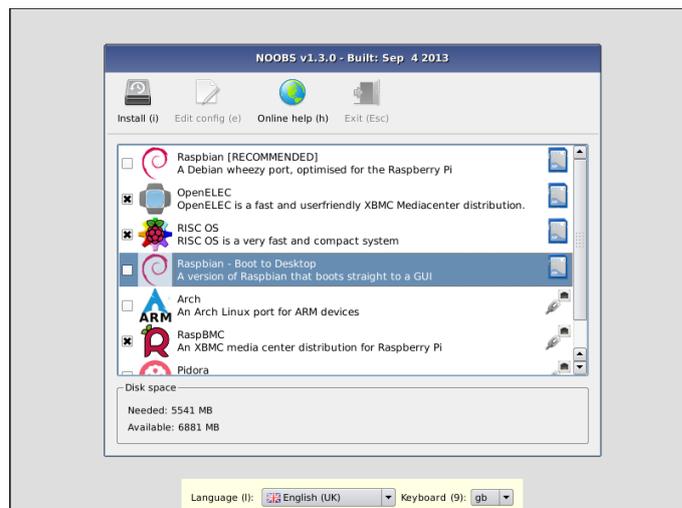
Esta etapa teve como primeiro passo a reunião de todo o equipamento necessário para o funcionamento do *Raspberry Pi*, mais especificamente, um cartão de memória, um carregador, um cabo *HDMI*, um cabo de rede *Ethernet*, um teclado (com porta *USB*), um rato (com porta *USB*), uma capa protetora e por fim um monitor para visualização das interações com o sistema operativo.

É importante referir que, durante esta etapa, quando se fez a ligação do equipamento ao *Raspberry Pi*, foi preciso ter em atenção a possibilidade de a eletricidade estática danificar tanto o equipamento como o seu desempenho. Para evitar essa possibilidade é necessário fazermos uma descarga elétrica da nossa eletricidade estática, entrando em contacto como um objeto metálico. De seguida, introduzimos a capa protetora no *Raspberry Pi* para podermos manusear as ligações sem problemas. Por fim, devemos ligar todo o equipamento à exceção do carregador, que só deverá ser ligado no final de todas as outras ligações estarem efetuadas.

Figura 4.1: Equipamentos que compõem o *Raspberry Pi*

4.1.2 Instalação do sistema operativo Raspbian

Os *Raspberry Pis* utilizados neste projeto trouxeram de base cartões de memória com um sistema operativo pré-instalado designado por *NOOBS*. Este sistema operativo limita o cartão de memória a utilizar apenas 1.1GB da sua capacidade, sendo neste caso necessário formatá-lo para podermos utilizar os 8GB, correspondentes à capacidade total do cartão, o que é recomendado pela comunidade que criou o *Raspberry Pi*. Para esse efeito utiliza-se o programa *SD Formatter* para termos uma formatação segundo os requisitos do cartão de memória. Em seguida usa-se o programa *Win 32 Disk Imager* para introduzirmos no cartão de memória a imagem do novo sistema operativo. Finalizando, introduz-se o cartão de memória no *Raspberry Pi* e liga-se o carregador à eletricidade, o que permite ao *Raspberry Pi* fazer o seu *booting*. Ao iniciarmos os *Raspberry Pi* devemos escolher a imagem do sistema operativo chamado de *Raspbian*, cuja imagem é semelhante ao sistema operativo criado pela comunidade *Linux* designado por *Debian*.

Figura 4.2: Escolha do sistema operativo *Raspbian*

Como primeira ação no *Raspberry Pi* introduziram-se os comandos necessários para obtermos acesso à *internet*. A ligação à *internet* permite que se faça a atualização do dispositivo e a instalação de programas necessários para o desenvolvimento do projeto de dissertação.

A segunda ação foi a instalação do programa *PuTTY* no computador pessoal para se fazer uma ligação, por um servidor *ssh*, direta entre o computador e o *Raspberry Pi*. Este programa permite-nos trabalhar no ambiente de trabalho do computador pessoal com a linha de comandos do *Raspberry Pi*.

4.1.3 Instalação da câmara

A verificação do funcionamento correto da câmara foi efetuada testando nomeadamente a capacidade da câmara tirar fotografias e capturar vídeo. A aquisição de fotografias foi efetuada no formato *JPEG*, tendo sido o vídeo capturado no formato *H.264*, que é um dos formatos mais utilizados principalmente quando o vídeo se destina a ser transmitido sobre uma rede de computadores. No contexto do *Raspberry Pi*, a visualização de vídeo é feita através do programa *VLC media player*, permitindo este programa efetuar a leitura de diferentes tipos de *codecs* tais como *MPEG-2*, *DivX*, *H.264*, entre outros.

4.2 2ª Etapa - Streaming de vídeo entre o terminal de acesso e o recetor (Ligação direta via cabo ethernet)

O objetivo desta segunda etapa do desenvolvimento do trabalho foi testar a possibilidade de efetuar *streaming* de vídeo entre o *Raspberry Pi* e o computador, isto é, transmitir em tempo real o sinal de vídeo adquirido pela câmara do *Raspberry Pi*. Para tal, a forma mais controlada de o fazer foi através de ligação direta, entre o computador e o *Raspberry Pi*, através do cabo de rede *ethernet*. Para abrir a porta no *Raspberry Pi* por onde foi transmitido o *streaming* de vídeo foi utilizado o programa *Netcat*. O programa *Mplayer* foi utilizado para visualizar esse *streaming* no computador.

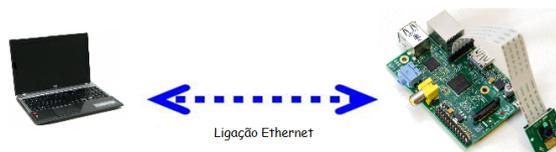


Figura 4.3: Ligação via cabo *Ethernet* entre o computador e o *Raspberry Pi*

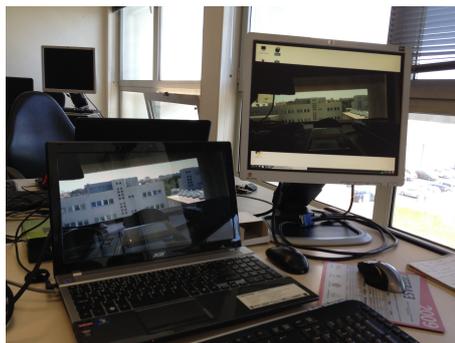


Figura 4.4: Primeiro teste de *streaming* de vídeo

4.3 3ª Etapa - Ligação sem fios entre o terminal de acesso e o recetor

Nesta etapa, o objetivo foi desenvolver comunicação sem fios entre o terminal de acesso e o recetor

Inicialmente instalou-se o *driver* da antena *TP-LINK* modelo *TL-WN722N*, para iniciar as primeiras comunicações. Numa primeira fase foram feitos testes de conectividade entre o computador e o *Raspberry Pi* com o sistema operativo *Raspbian*. Estes testes foram realizados com o auxílio de um *router* pessoal, tal como representado na Figura 4.5. Após esses testes de conectividade, concretizou-se a transmissão do primeiro *streaming* de vídeo entre o computador e o *Raspberry Pi* usando comunicação sem fios.

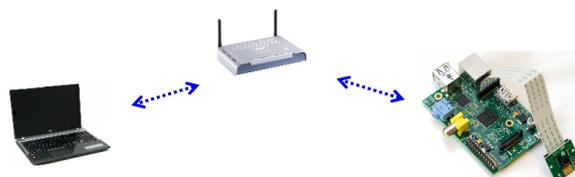


Figura 4.5: Ligação sem fios entre o computador e o *Raspberry Pi*

4.4 4ª Etapa - Instalação de um nó de rede

Com o objetivo principal de criar uma rede emalhada, sobre a qual se pudesse testar a transmissão de vídeo, optou-se por desenvolver os nós de rede usando equipamentos *Raspberry Pi*, uma vez que permitiam com alguma facilidade a mobilidade desses mesmos nós.

A instalação deste primeiro nó de rede seguiu os passos da primeira etapa, com a diferença principal de se utilizar o sistema operativo designado por *Open WRT* (sendo a versão escolhida aquela que se adequa às especificações do *Raspberry Pi*). Comparando com o sistema operativo

NOOBS, o *Open WRT* tem características mais apropriadas para o desenvolvimento de aplicações de rede, sendo habitualmente utilizado nestes contextos.



Figura 4.6: Nó de rede - *Raspberry Pi* contendo o sistema operativo *OpenWrt*

4.5 5ª Etapa - Rede *ad-hoc* com um nó

Nesta etapa intervêm três equipamentos, nomeadamente, o computador (recetor), o *Raspberry Pi* com a câmara (emissor) e o nó que vai providenciar a rede, constituído pelo *Raspberry Pi* com o sistema operativo *OpenWrt*. Este nó, numa primeira fase, irá comportar-se como um *Access Point* (AP), proporcionando numa segunda fase uma rede *ad-hoc*.



Figura 4.7: Comunicação através de um *Access Point*

Como se pode ver nas Figuras 4.7 e 4.8, as duas fases diferem essencialmente nas rotas criadas na rede. A Figura 4.7 representa uma rede em modo infra-estrutura, com a existência de um AP, enquanto a Figura 4.8 representa uma típica rede *ad-hoc*, onde a infra-estrutura desaparece e todos os elementos criam rotas entre si automaticamente.

4.6 6ª Etapa - Rede *ad-hoc* com dois nós

O passo seguinte, natural para o cumprimento dos objetivos da dissertação, foi incluir mais nós na estrutura criada anteriormente. Nesse sentido, adicionou-se um novo nó à rede para aumentar

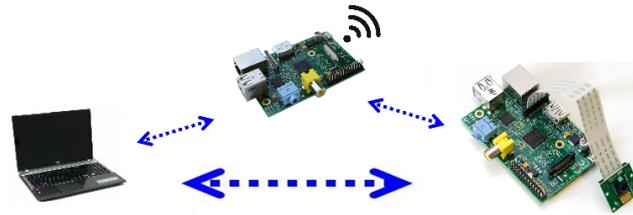


Figura 4.8: Comunicação *ad-hoc*

a complexidade desta. Este nó é configurado com o sistema operativo *OpenWrt* executando os mesmos passos da 4ª etapa. Tendo dois nós de rede, podemos fazer diferentes tipos de testes, sendo o mais importante aquele que obriga a informação a passar pelos dois nós antes de atingir os pontos periféricos da rede. Para tal é necessário introduzir manualmente as rotas para obrigar os pacotes transmitidos a passarem pelos dois nós. Na Figura 4.9 observamos este esquema, introduzindo manualmente as rotas.

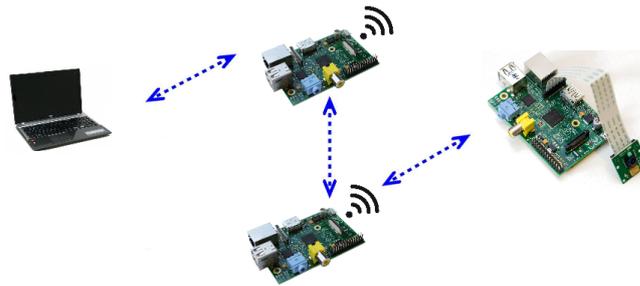


Figura 4.9: Comunicação *ad-hoc* com dois nós

4.7 7ª Etapa - Rede emalhada

Como passo final para a concretização do objetivo principal da dissertação, foi necessário transformar a rede *ad-hoc* numa rede *mesh*. Para atingir este tipo de rede introduziu-se nos nós anteriormente instalados o protocolo *OLSR*. Este protocolo permite à rede auto-configurar-se e auto-organizar-se. As rotas são criadas automaticamente e com o objetivo de ter rotas que permitam o maior fluxo de pacotes possível. Caso a rota criada permita um fluxo limitado, o protocolo *OLSR* cria novas rotas para aumentar o fluxo de pacotes. A Figura 4.10 representa as diferentes rotas criadas pelo protocolo.

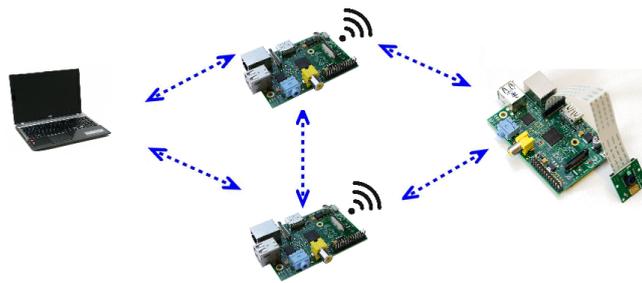


Figura 4.10: Rede *Mesh* com dois nós

Capítulo 5

Análise de resultados

Neste capítulo são apresentados os resultados obtidos nas diferentes experiências efetuadas ao longo do projeto de dissertação.

A apresentação dos resultados está organizada por *testbed*, correspondendo cada *testbed* sensivelmente a uma etapa do desenvolvimento apresentada no capítulo anterior.

A próxima secção descreve os *testbeds* utilizados para obter os resultados, nomeadamente os seus objetivos, configurações e particularidades relevantes para o desenvolvimento do projeto. A secção seguinte descreve os três cenários de tráfego utilizados para testar cada *testbed*. Por fim, a última secção deste capítulo apresenta os resultados obtidos em cada *testbed*, testado com cada um dos cenários de tráfego.

5.1 *Testbeds*

5.1.1 *Testbed 1 - Comunicação sem fios através de router pessoal*

O primeiro *testbed* é constituído por três equipamentos, mais especificamente um computador, um *router* (que contém duas antenas) e um *Raspberry Pi*, com o sistema operativo *Raspbian* e uma interface *wireless* (Figura 5.1). Com este *testbed* pretendeu-se testar a ligação *wireless* entre os dois terminais. O *router* desempenha o papel de fornecer a rede, comportando-se como uma *gateway*, encaminhando o tráfego entre os dois terminais, caso exista um pedido de comunicação. Qualquer tipo de comunicação passa sempre pelo *router*, sendo este o elemento que diferencia este *testbed* dos restantes. Este *testbed* surge como uma primeira experiência de ligação *wireless*, e como evolução natural após se ter testado com sucesso a ligação por cabo.

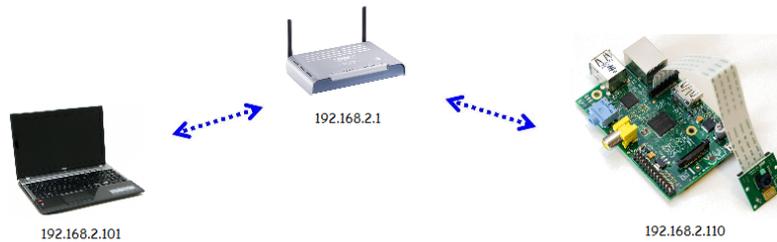


Figura 5.1: Caracterização do *Testbed 1* - Comunicação sem fios através de *router* pessoal

5.1.2 *Testbed 2* - Comunicação sem fios através de um *Access Point*

Comparando com o *Testbed 1*, este *testbed* pretendeu ser o primeiro passo nas experiências de transmissão de sinal de vídeo em tempo real sobre nós de rede desenvolvidos especificamente durante a elaboração do projeto. O nó de rede neste caso desempenha o papel de um *Access Point* que cria uma rede sem fios infra-estruturada, providenciando comunicação entre os terminais.

Este *testbed* é constituído por três equipamentos, mais especificamente por dois *Raspberry Pi* e pelo computador que recebe todo o tráfego de vídeo enviado pelo terminal que contém a câmara (Figura 5.2). Os dois *Raspberry Pi* distinguem-se pelo sistema operativo instalado. O primeiro, além de conter a câmara, pois é um terminal, contém o sistema operativo *Raspbian* que permite uma maior facilidade no manuseamento das especificações da câmara, enquanto o segundo contém o sistema operativo *OpenWrt* que facilita a criação e estabilização da rede que pretendemos. Neste *testbed* conseguimos de certa forma simular um *router*, já que o *Access Point* vai encaminhar o tráfego que recebe para diferentes direções dependendo do endereço de destino.



Figura 5.2: Caracterização do *Testbed 2* - Comunicação sem fios através de *Access Point*

5.1.3 *Testbed 3* - Rede *ad-hoc* com ligação direta entre os terminais (1 nó)

Para ir ao encontro do objetivo principal da dissertação era necessário criar uma rede *ad-hoc*, já que a rede *Mesh* é um caso específico desta. Como a rede *ad-hoc* não possui uma estrutura, este

testbed permite obter resultados de uma comunicação direta entre os dois terminais dando origem numa fase mais adiantada a comparações com rotas estáticas. Face ao *tesbed* anterior apenas se alterou o tipo de rede, mantendo na mesma três dispositivos, mas todos posicionados ao mesmo nível. Este novo nó tem a particularidade de criar rotas tal como indica a Figura 5.3.

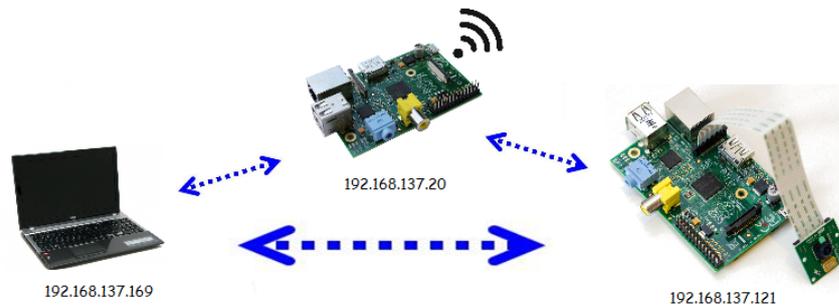


Figura 5.3: Caraterização do *Testbed 3* - Rede *ad-hoc* com ligação direta entre os terminais (1 nó)

5.1.4 *Testbed 4* - Rede *ad-hoc* (1 nó) com 2 hops

A necessidade de testar a ocorrência de variações de qualidade na transmissão de vídeo conduziu à ideia de criação de um *testbed* que obrigue o tráfego a passar primeiro pelo nó e depois ser retransmitido para o recetor, que neste caso é o computador pessoal. A criação de uma rota seguindo o comando

```
route add -host 192.168.137.169 gw 192.168.137.20
```

permite acabar com a ligação direta entre os terminais, aproximando desta maneira a rede a uma pequena estrutura. Pretende-se desta maneira simular um caso em que, devido a limitações do ambiente ao qual se acede à rede, a comunicação entre os terminais é feita através de nó de rede, como meio de comunicação entre ambos.

```
pi@raspberrypi ~$ traceroute 192.168.137.169
traceroute to 192.168.137.169 (192.168.137.169), 30 hops max, 60 byte packets
 1 192.168.137.20 (192.168.137.20) 5.606 ms 6.707 ms 8.111 ms
 2 192.168.137.169 (192.168.137.169) 29.043 ms 31.894 ms 16.603 ms
pi@raspberrypi ~$
```

Figura 5.4: Comando *traceroute* que comprova a comunicação através de 2 hops

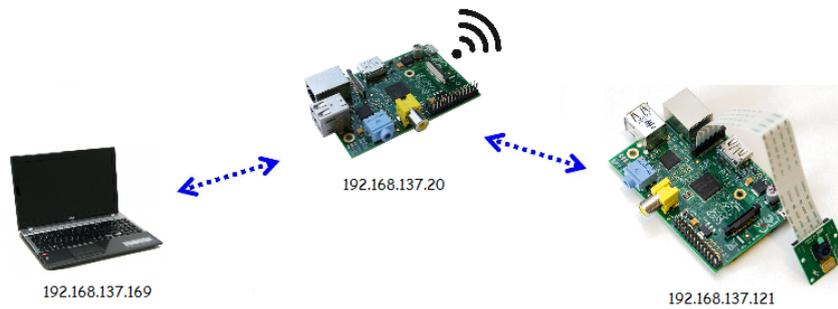


Figura 5.5: Caraterização do *Testbed 4* - Rede ad-hoc (1 nó) com 2 hops

5.1.5 *Testbed 5* - Rede ad-hoc com ligação direta entre os terminais (2 nós)

O objetivo deste *testbed* foi aumentar a complexidade da rede, introduzindo um novo nó. Este novo equipamento é outro *Raspberry Pi* com o sistema operativo *OpenWrt*, desempenhando o papel de um nó de rede semelhante ao existente nos *testbeds* anteriores. Como se pode observar na Figura 5.6, todos os equipamentos contêm rotas que permitem a comunicação entre todos, que é o comportamento típico de uma rede *ad-hoc*. Havendo comunicação direta entre os dois terminais (emissor e recetor), a comunicação entre um terminal e o outro é feita diretamente sem passar pelos nós.

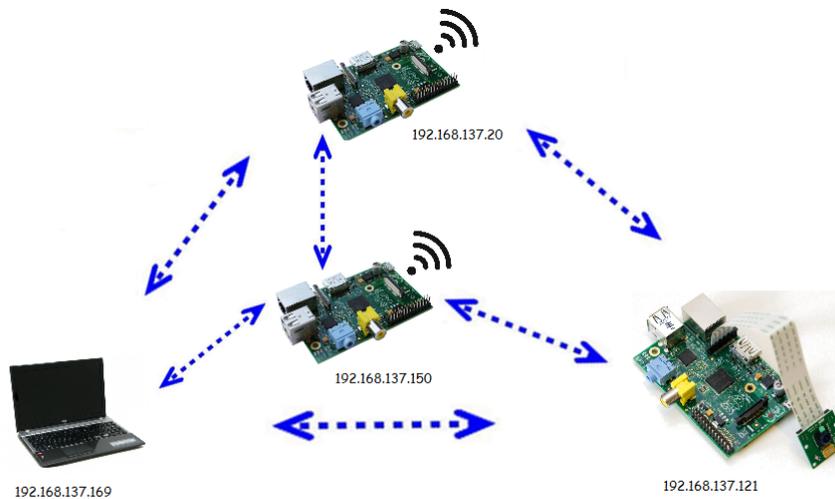


Figura 5.6: Caraterização do *Testbed 5* - Rede ad-hoc com ligação direta entre os terminais (2 nós)

5.1.6 *Testbed 6* - Rede ad-hoc (2 nós) com 3 hops

Constituído pelos mesmos tipos de equipamento que o *testbed* anterior, este *testbed* impede a comunicação direta entre os terminais, definindo rotas estáticas. Foram consideradas duas soluções possíveis para impedir a comunicação direta entre os dois terminais. A primeira solução

passaria por dividir a rede em duas sub-redes. Ao termos duas redes teríamos uma máscara que limitava a gama de números *IP* para os quais o terminal poderia comunicar. A outra solução seria a definição de rotas estáticas do tipo *host*. A importância deste *testbed* está na observação da degradação da qualidade de serviço já que a transmissão de vídeo terá de passar por diferentes elementos de rede até chegar ao destino, executando desta maneira três *hops*.

```

pi@raspberrypi ~$ traceroute 192.168.137.169
traceroute to 192.168.137.169 (192.168.137.169), 30 hops max, 60 byte packets
 1 192.168.137.150 (192.168.137.150)  3.599 ms  4.667 ms  6.007 ms
 2 192.168.137.20 (192.168.137.20) 10.075 ms 12.790 ms 15.433 ms
 3 192.168.137.169 (192.168.137.169) 50.033 ms 61.225 ms 61.013 ms
pi@raspberrypi ~$

```

Figura 5.7: Comando *traceroute* que comprova a comunicação através de 3 *hops*

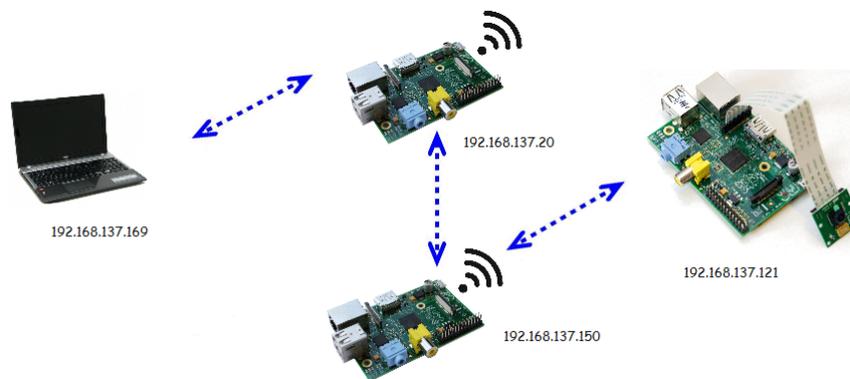


Figura 5.8: Caracterização do *Testbed 6* - Rede ad-hoc (2 nós) sem ligação direta dos terminais

5.1.7 *Testbed 7* - Rede emalhada (2 nós)

Sendo o objetivo da dissertação a construção de uma rede emalhada, utilizaram-se os conhecimentos adquiridos na área das redes *ad-hoc* para construir este novo tipo de rede. As redes emalhadas são criadas sobre redes *ad-hoc*, instalando o protocolo *OLSR* nos nós da rede.

Foram criados então dois novos nós (o 192.168.137.40 e o 192.168.137.155) em modo *ad-hoc* e introduziu-se de seguida o protocolo *OLSR*, tal como pode ser observado na Figura 5.9. Para a introdução deste protocolo foi necessário a introdução de uma nova interface de rede, já que com as redes *ad-hoc* podia-se difundir o sinal através de uma *bridge* entre a interface *ethernet* e a interface *wireless*. Assim, criando esta nova interface, que neste caso é *wireless*, permite aos nós ter comunicação com os clientes da rede através de cabo ou por *wi-fi*. Desta forma, podemos por exemplo ligar um *router* à rede tanto por cabo ou por *wi-fi*, difundido a *internet* para todos os elementos da rede.

Face aos *testbeds* anteriores, aquele que permite ter uma comparação mais direta é o *Testbed 5*, ou seja o *testbed* com rede *ad-hoc* com ligação direta entre os terminais (2 nós), pois tal

como a rede emalhada contém dois nós e é difundida uma rota que permite a ligação direta entre os terminais (neste caso, o computador e o *Raspberry pi* com câmara).

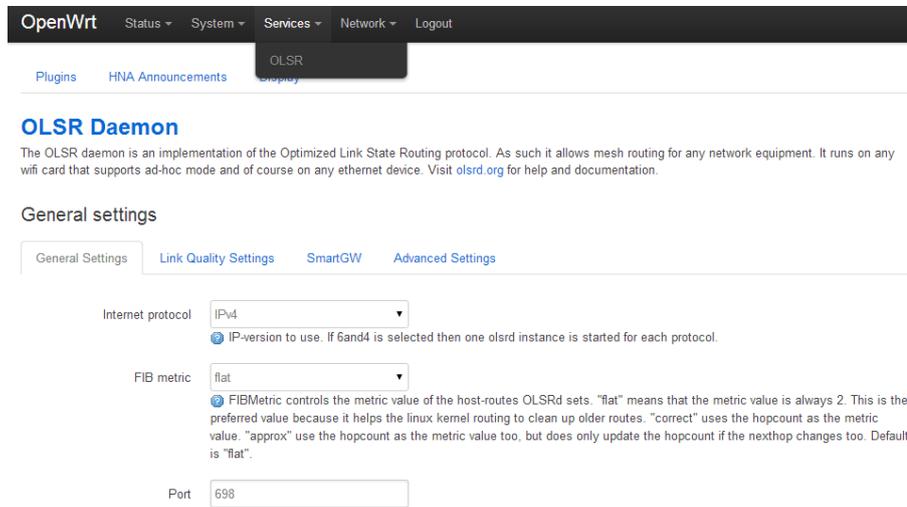


Figura 5.9: Página fornecida pela aplicação *luci* para a configuração do protocolo *OLSR*

A Figura 5.10 comprova que a interface associada, neste caso a *wireless*, opera no modo *mesh*, verificando-se assim a utilização de uma rede emalhada. A Figura 5.11 apresenta as estações que estão associadas ao nó que está a ser visualizado. Podemos observar não só os *BSSIDs* (Basic Service Set Identification) dos elementos da rede como também os seus números *IP*.

Interfaces

Enable	Network	Mode	Hello	TC	MID	HNA	
<input checked="" type="checkbox"/>		mesh	5.0s / 40.0s	2.0s / 256.0s	18.0s / 324.0s	18.0s / 108.0s	Edit Delete
<input checked="" type="checkbox"/>	wlan0:	mesh	5.0s / 40.0s	2.0s / 256.0s	18.0s / 324.0s	18.0s / 108.0s	Edit Delete

[Add](#)

Figura 5.10: A interface *wireless* com o modo rede emalhada

Wireless

Generic 802.11bgn Wireless Controller (radio0) SSID: OpenWrt
Mode: Ad-Hoc
Channel: 11 (2.462 GHz)
Bitrate: 27.5 Mbit/s
BSSID: 96:36:62:3A:25:0F
Encryption: -

100%

Associated Stations

MAC-Address	Network	Signal	Noise	RX Rate	TX Rate
▬ C0:4A:00:1B:CB:A3	Ad-Hoc "OpenWrt"	-14 dBm	0 dBm	1.0 Mbit/s, MCS 0, 20MHz	1.0 Mbit/s, MCS 0, 20MHz
▬ C0:4A:00:1B:B9:84	Ad-Hoc "OpenWrt"	-16 dBm	0 dBm	1.0 Mbit/s, MCS 0, 20MHz	1.0 Mbit/s, MCS 0, 20MHz
▬ 44:6D:57:1F:EB:C1	Ad-Hoc "OpenWrt"	-34 dBm	0 dBm	54.0 Mbit/s, MCS 0, 20MHz	54.0 Mbit/s, MCS 0, 20MHz

Figura 5.11: Visualização num nó dos outros elementos presentes na rede

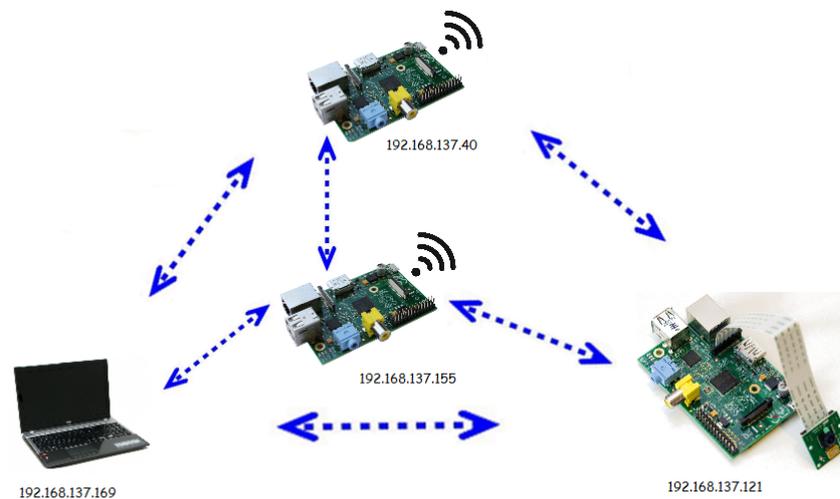


Figura 5.12: Caracterização do *Testbed 7* - Rede em malha com dois nós e com ligação direta dos terminais

5.1.8 *Testbed 8* - Rede em malha (2 nós), mas com distância acentuada entre os nós da rede

O presente *testbed* surgiu com o objetivo de testar a eficácia do protocolo *OLSR* (Optimized Link State Routing). O que se pretendeu foi verificar se com o afastamento dos elementos da rede, tais como os nós e os terminais de acesso, a qualidade de transmissão continuava a ser garantida e ao mesmo tempo se havia a capacidade da rede de se reconfigurar, utilizando rotas alternativas para a comunicação entre os terminais. Neste sentido, utilizou-se a mesma configuração de rede descrita no *testbed* anterior, mas com a alteração da distância entre os equipamentos.

É necessário referir que o protocolo *OLSR* apenas é instalado nos nós de rede, como se pode observar na Figura 5.13. Os dois terminais, o emissor e o recetor, não utilizam este protocolo, pois apenas se comportam como clientes da rede.



Figura 5.13: Os elementos da rede que utilizam o protocolo *OLSR*

O primeiro nó que foi criado (192.168.137.40), comporta-se como nó principal, pois quando acedemos à rede com o recetor, num caso em que o *IP* é atribuído automaticamente, este assume este nó como *gateway*, como se pode observar na Figura 5.14. Seguidamente, ligou-se o emissor ao nó (192.168.137.40) e afastou-se o recetor para um piso superior, fora do ambiente do *testbed*. O comando *traceroute*, executado no emissor, permite verificar que existe uma enorme dificuldade em aceder ao recetor, o que se observa na Figura 5.15. Ou seja, aqui encontra-se presente uma ligação direta entre o nó e o recetor.

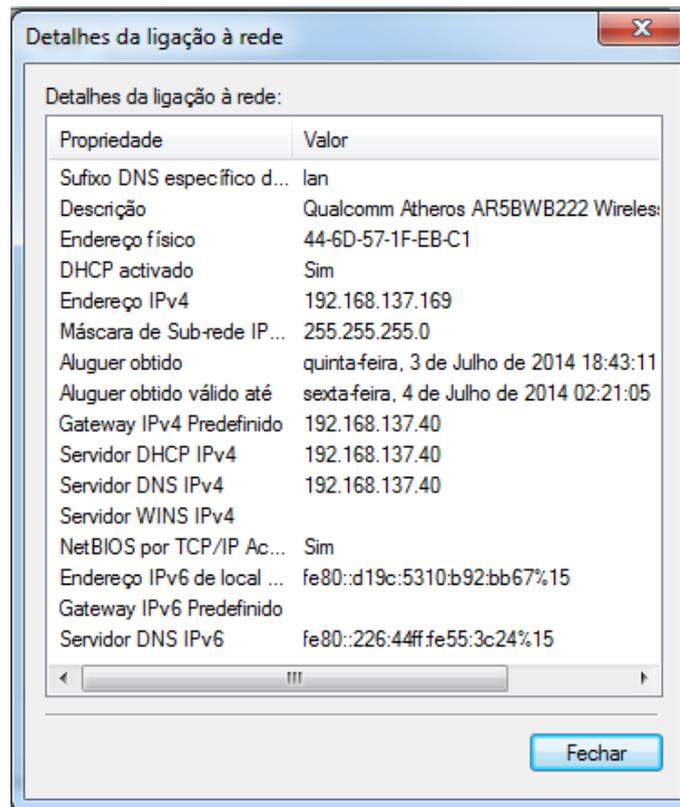


Figura 5.14: Detalhes da ligação do recetor

```

pi@raspberrypi ~ $ traceroute 192.168.137.169
traceroute to 192.168.137.169 (192.168.137.169), 30 hops max, 60 byte packets
 1 192.168.137.40 (192.168.137.40) 1.761 ms 2.137 ms 2.507 ms
 2 192.168.137.169 (192.168.137.169) 1625.200 ms 22.352 ms 23.891 ms

```

Figura 5.15: Comando *traceroute* no emissor

Numa segunda fase, em que se pretende testar o protocolo *OLSR*, introduziu-se um *IP* estático no recetor, como podemos ver na Figura 5.16, que contém a gateway 192.168.137.155. Isto, porque o recetor tem enorme dificuldade em aceder diretamente ao nó principal, e em contrapartida o nó (192.168.137.155) encontra-se no corredor, tal como descreve a Figura 5.17.

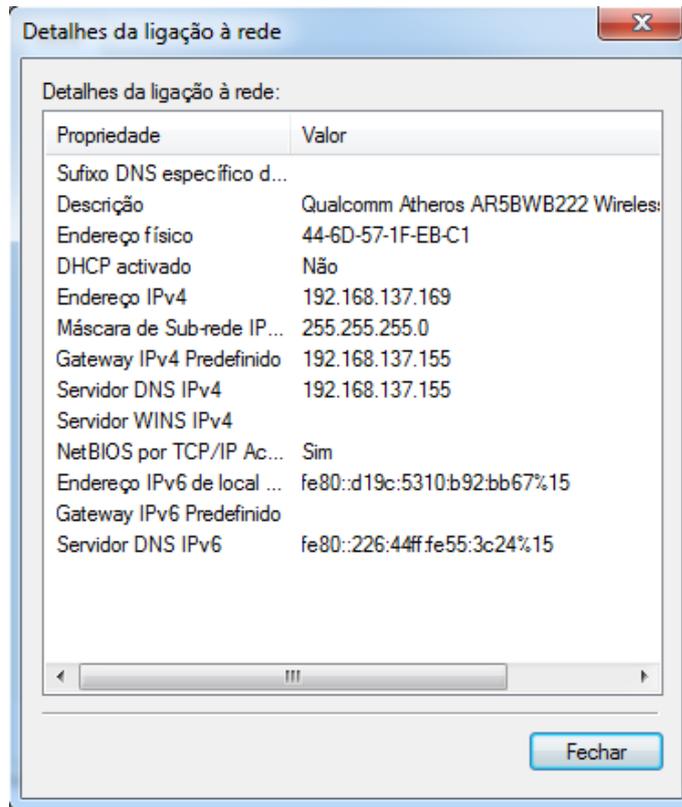


Figura 5.16: Detalhes da ligação do recetor com *IP* estático

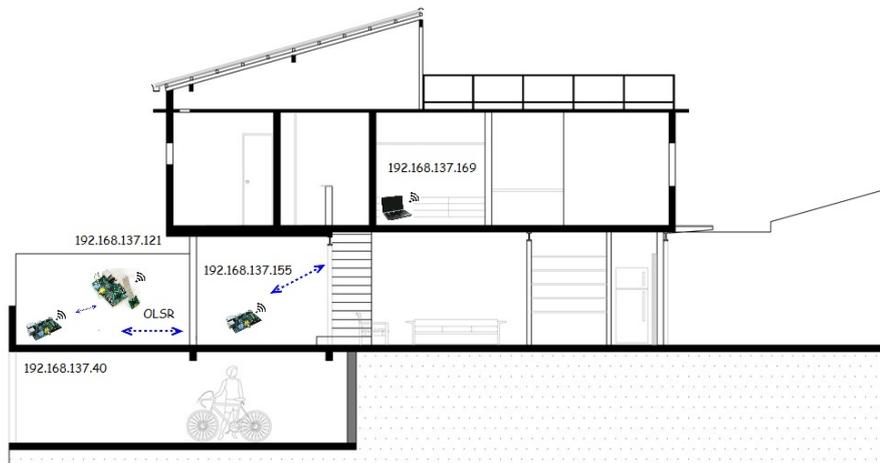


Figura 5.17: Caraterização do *Testbed* 8

Face a esta nova caracterização da rede, verificamos através do comando *traceroute* no emissor que o tempo de resposta diminui drasticamente comparando com a primeira fase mencionada. A Figura 5.18 permite concluir que o protocolo *OLSR* tem efeito na ligação, pois escolhe uma rota que tem menores custos. A razão pela qual não aparece nesta figura uma instância com

192.168.137.155, antes de atingir o *IP* do recetor, é o protocolo não conter rotas estáticas entre os dois nós, apenas se verifica melhoria nos tempos de emissão e receção.

```

pi@raspberrypi ~$ traceroute 192.168.137.169
traceroute to 192.168.137.169 (192.168.137.169), 30 hops max, 60 byte packets
 1 192.168.137.40 (192.168.137.40) 1.748 ms 1.698 ms 1.670 ms
 2 192.168.137.169 (192.168.137.169) 2.097 ms 3.542 ms 2.975 ms

```

Figura 5.18: Comando *traceroute* presente no emissor em que se verifica uma redução drástica no tempo de resposta

A Figura 5.19 demonstra a utilização do protocolo *OLSR*. Esta figura foi retirada dos ficheiros de análise do *Wireshark*, neste caso no recetor, e contém os pacotes *OLSR* e as solicitações de vizinhos.

66282	121.962916	fe80::d19c:5310:b92ff02::1:ff55:3c24	192.168.137.169	ICMPv6	86 Neighbor solicitation for fe80::226:44ff:fe55:3c24 from 44:6d:57:1f:eb:c1
66283	122.024905	192.168.137.121	192.168.137.169	TCP	66 [TCP Keep-Alive] 34876 > complexlink [ACK] Seq=60625168 Ack=1 Win=249 Len=0
66284	122.024909	192.168.137.169	192.168.137.121	TCP	66 [TCP ZeroWindow] (comp) Len=24876 [ACK] Seq=1 Ack=60625169 Win=0 Len=0
66285	122.077668	192.168.137.155	192.168.137.255	OLSR v1	110 OLSR (IPv4) Packet, Length: 68 Bytes
66286	122.138117	192.168.137.40	192.168.137.255	OLSR v1	134 OLSR (IPv4) Packet, Length: 92 Bytes

Figura 5.19: Detalhes dos pacotes *OLSR* através da plataforma *Wireshark*

Tal como o proposto, este protocolo teve bastante sucesso na manutenção da qualidade de serviço global da transmissão de vídeo, tendo-se obtido resultados muito semelhantes a ligações diretas e com distâncias reduzidas entre os nós da rede e os terminais.

5.2 Cenários de teste

Para podermos abordar diferentes aspetos no estudo da transmissão de vídeo sobre redes emalhadas, decidiu-se por à prova os vários *testbeds* sob três cenários distintos de tráfego. Pretendia-se que estes cenários de tráfego possibilitassem uma fácil comparação dos resultados obtidos nos vários *testbeds*.

Os três cenários de tráfego foram designados por “cenário de referência”, “cenário calmo” e “cenário movimentado”.

5.2.1 Cenário (a) - Cenário de referência

O cenário de referência é caracterizado por uma imagem fixa, isto é, não existe qualquer índice de movimento aquando da transmissão de vídeo. Em termos concretos, optou-se por filmar uma parede branca durante dois minutos. Espera-se que os gráficos do recetor correspondentes a este cenário contenham sinais de vídeo uniformes, com variações muito reduzidas.

5.2.2 Cenário (b) - Cenário calmo

Este cenário distingue-se do cenário de referência pelo facto de conter imagens com ligeiras movimentações durante intervalos de tempo curtos e intercalados. Exemplos destes tipos de movimentos são o movimento de uma mão, uma pessoa a atravessar o raio de visão da câmara

a um velocidade normal, ou uma pessoa nesse mesmo raio de visão a fazer movimentos lentos. Este cenário permite avaliar a alteração no sinal de vídeo, que no cenário anterior era constante. Durante os dois minutos de transmissão espera-se notar nos gráficos do recetor ligeiros picos que representam as movimentações suaves, ou seja, uma alteração no sinal de vídeo, embora ligeira.

5.2.3 Cenário (c) - Cenário movimentado

O cenário movimentado tenta ser o oposto do cenário anterior. Neste cenário são incluídas movimentações mais acentuadas, quer seja em termos de um aumento do período de existência de movimento, quer seja pela velocidade com que o movimento é efetuado. Este tipo de movimentações podem ser provocadas por uma ou várias pessoas no raio de visão da câmara a fazer movimentos mais repentinos, com o objetivo de que a câmara tenha dificuldade na captação da imagem que pretende transmitir. A ideia foi gerar uma quantidade maior de tráfego, mais heterogéneo ao longo do período de transmissão. Espera-se que este cenário provoque instabilidade no sinal de vídeo observado no recetor, isto é, este sinal irá ser constituído por diversos picos não sendo desta maneira um sinal contínuo, criando um distanciamento acentuado para o cenário de referência.

5.3 Resultados

Nesta secção são apresentados os resultados das experiências efetuadas nos diferentes *testbeds*. Em cada *testbed* foram testados os três cenários de tráfego apresentados na secção anterior. Assim, para cada *testbed* e para cada cenário são apresentados dois gráficos e uma tabela. Os dois gráficos caracterizam o sinal de vídeo que sai do emissor e que chega ao recetor, respetivamente. Todos estes gráficos representam no eixo dos *xx* o tempo de transmissão, medido em segundos. Cada transmissão tem sensivelmente dois minutos de duração, sendo o início e o final da transmissão comandados por quem está a executar o teste. No eixo dos *yy* está representado o número de pacotes transmitidos. A tabela contém os valores dos parâmetros de rede mais importantes neste contexto, nomeadamente o débito médio, número de pacotes enviados pelo emissor, número de pacotes recebidos pelo recetor, número de pacotes perdidos e taxa de perda de pacotes. O valor do débito médio nestas tabelas correspondem aos débitos medidos no emissor.

A monitorização e aquisição destes resultados foi efetuada pelo programa de análise de redes *Wireshark*, monitorizando a porta 5001 (*TCP* ou *UDP*), que foi a porta utilizada pelo terminal de acesso de aquisição de vídeo para transmitir o sinal de vídeo para o recetor.

Salienta-se ainda o facto de o estudo incidir sobre a transmissão de vídeo adquirido em tempo real. Por isso, embora os sinais de vídeo testados em cada cenário serem semelhantes, estes são sempre únicos, uma vez que eram obtidos em tempo real.

5.3.1 *Testbed* 1 - Comunicação sem fios através de *router* pessoal

5.3.1.1 Cenário (a) - Cenário de referência

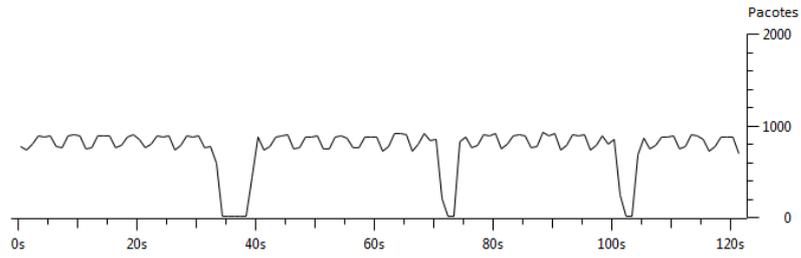


Figura 5.20: *Testbed 1* - Cenário (a) - Sinal do emissor

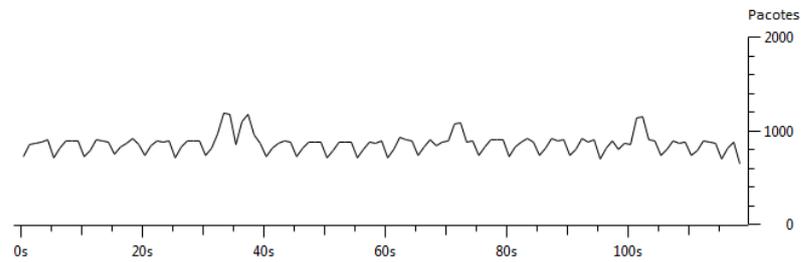


Figura 5.21: *Testbed 1* - Cenário (a) - Sinal do recetor

Tabela 5.1: *Testbed 1* - Cenário (a) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
752,914	64747	63050	1697	2,62%

5.3.1.2 Cenário (b) - Cenário calmo

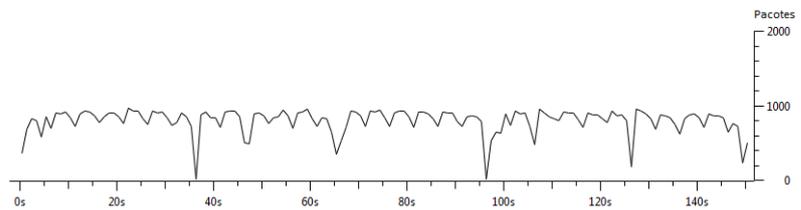


Figura 5.22: *Testbed 1* - Cenário (b) - Sinal do emissor

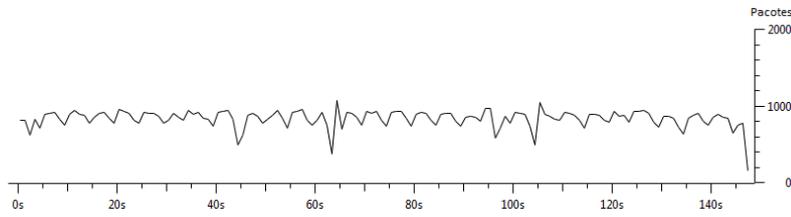


Figura 5.23: *Testbed 1* - Cenário (b) - Sinal do Recetor

Tabela 5.2: *Testbed 1* - Cenário (b) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
796,513	85130	83065	2065	2,43%

5.3.1.3 Cenário (c) - Cenário movimentado

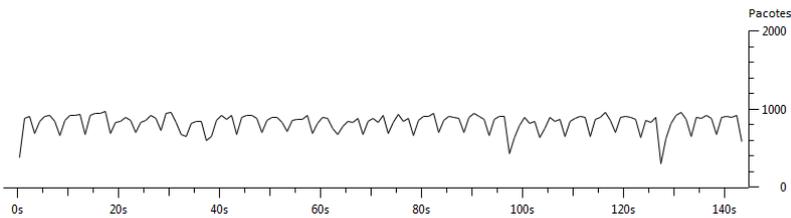


Figura 5.24: *Testbed 1* - Cenário (c) - Sinal do emissor

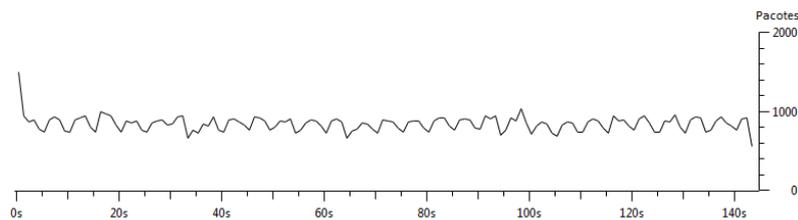


Figura 5.25: *Testbed 1* - Cenário (c) - Sinal do recetor

Tabela 5.3: *Testbed 1* - Cenário (c) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
816,57	82757	82676	81	0.10%

Fazendo uma análise pormenorizada aos resultados verificámos que existem quebras no sinal de vídeo no emissor. Isto é devido ao facto de o sistema de transmissão de vídeo do emissor ser feito através de uma *cache*. No recetor utiliza-se um programa de visualização que recebe o *streaming* de vídeo preenchendo também uma *cache*. Esta *cache* tem como função principal permitir ordenar os pacotes que possam ter sido recebidos de forma desordenada. A visualização no recetor só é efetuada após ter havido uma ordenação dos pacotes recebidos. Sendo assim, o atraso visual entre o momento da aquisição de vídeo no emissor e a visualização no recetor é provocado principalmente pelo protocolo de transmissão em tempo real que tem de ordenar os pacotes para iniciar a visualização final, e não pelo atraso provocado pela rede. Em termos subjetivos, neste primeiro *testbed* assistiu-se nos dois primeiros cenários a um atraso observado visualmente de cerca de três segundos, enquanto no último cenário esse atraso foi sensivelmente de quatro segundos.

Ao longo dos três cenários, o sinal do recetor vai sofrendo alterações, iniciando-se com um sinal estável no cenário de referência, depois com um sinal com ligeiros picos no cenário calmo e por fim um sinal instável no cenário movimentado.

No capítulo 2 foi explicitado que a norma *IEEE 802.11g* atinge um débito máximo de 54 Mbit/s numa banda de frequência de 2.4 GHz. Neste sentido, esta norma utiliza uma técnica de modulação denominada de *DSSS (Direct Sequence Spread Spectrum)* que divide a gama de frequência em 14 canais de 22 MHz cada um. Isto leva a concluir que quanto mais complexo for o cenário, maiores serão os valores de débito médio, porque os terminais em ambientes mais complexos necessitam de transmitir mais pacotes requisitando uma ligação com maior débito. Para além disso, as redes *wireless* com a norma *IEEE 802.11g* podem sofrer interferências de diversos tipos de equipamentos como por exemplo micro-ondas e telefones sem fio.

Observando as Figuras 5.26 a 5.28, verificamos que existe um aumento na capacidade do canal (em Mbits/s) o que origina um aumento no débito médio (em pacotes por segundo).

Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	91813	91813	100.000%	0	0,000%
Between first and last packet	121,944 sec				
Avg. packets/sec	752,914				
Avg. packet size	959,967 bytes				
Bytes	88137466	88137466	100.000%	0	0.000%
Avg. bytes/sec	722772,599				
Avg. MBit/sec	5,782				

Figura 5.26: Valores do Cenário sem nada para o *testbed 1*

Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	120132	120132	100.000%	0	0,000%
Between first and last packet	150,822 sec				
Avg. packets/sec	796,513				
Avg. packet size	971,164 bytes				
Bytes	116667879	116667879	100.000%	0	0,000%
Avg. bytes/sec	773544,567				
Avg. MBit/sec	6,188				

Figura 5.27: Valores do Cenário calmo para o *testbed 1*

Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	117437	117437	100.000%	0	0,000%
Between first and last packet	143,818 sec				
Avg. packets/sec	816,570				
Avg. packet size	962,369 bytes				
Bytes	113017686	113017686	100.000%	0	0,000%
Avg. bytes/sec	785840,953				
Avg. MBit/sec	6,287				

Figura 5.28: Valores do Cenário movimentado para o *testbed 1*

5.3.2 *Testbed 2 - Comunicação sem fios através de Access Point*

5.3.2.1 Cenário (a) - Cenário de referência

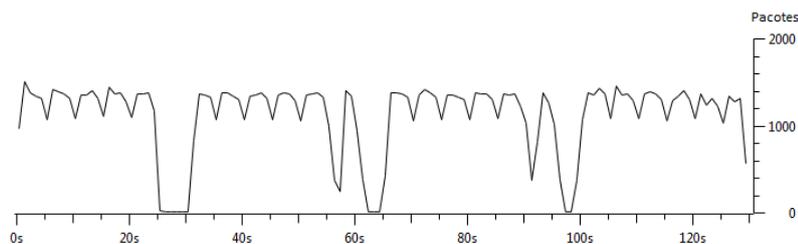


Figura 5.29: *Testbed 2 - Cenário (a) - Sinal do emissor*

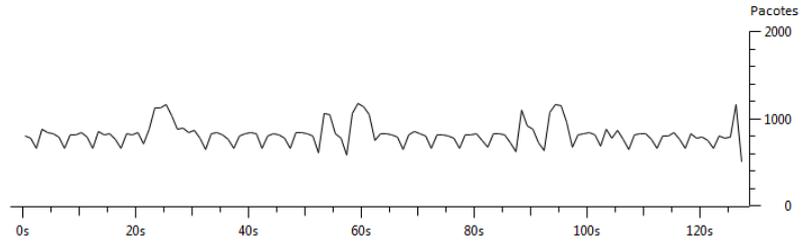


Figura 5.30: *Testbed 2* - Cenário (a) - Sinal do recetor

Tabela 5.4: *Testbed 2* - Cenário (a) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
1119,746	119016	116673	2343	1,97%

5.3.2.2 Cenário (b) - Cenário calmo

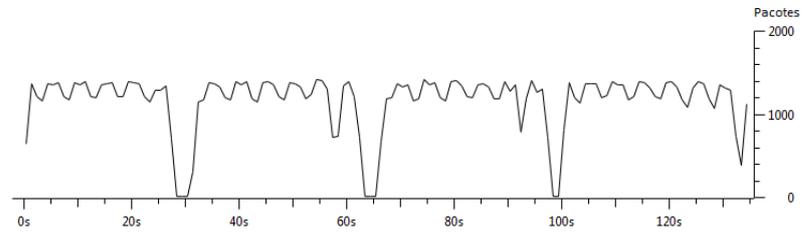


Figura 5.31: *Testbed 2* - Cenário (b) - Sinal do emissor

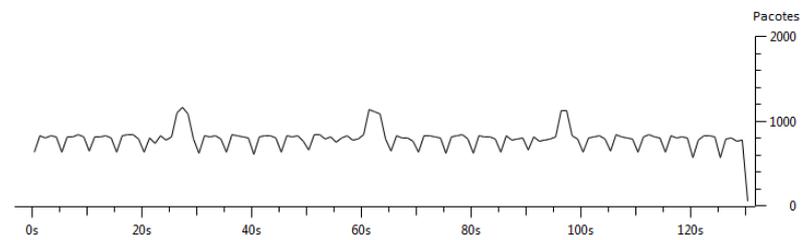
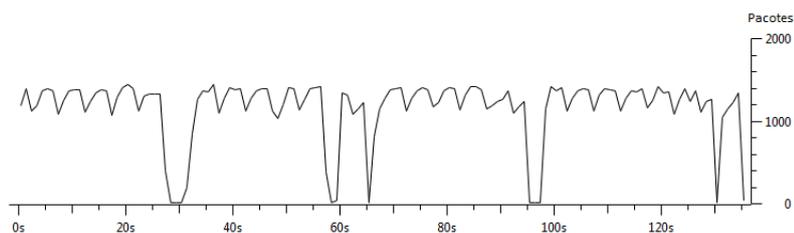
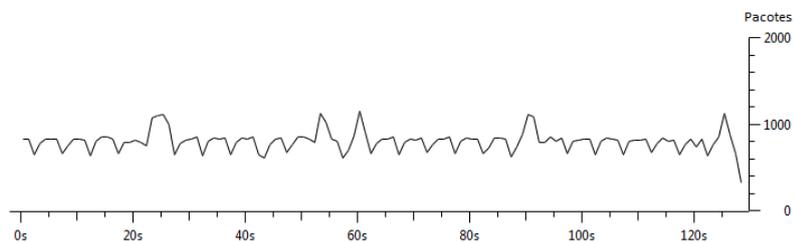


Figura 5.32: *Testbed 2* - Cenário (b) - Sinal do recetor

Tabela 5.5: *Testbed 2* - Cenário (b) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
1158,144	127895	122743	5152	4,03%

5.3.2.3 Cenário (c) - Cenário movimentado

Figura 5.33: *Testbed 2* - Cenário (c) - Sinal do emissorFigura 5.34: *Testbed 2* - Cenário (c) - Sinal do recetorTabela 5.6: *Testbed 2* - Cenário (c) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
1160,055	128115	121284	6831	5,33%

Analisando os resultados obtidos nos três cenários do *Testbed 2*, e comparando com os resultados do *testbed* anterior, verificamos um aumento na taxa de pacotes perdidos. Isto verifica-se pelo facto de o *router* pessoal ter maior capacidade, enquanto que a antena que utilizamos no *Raspberry Pi* é mais limitada em termos de capacidade. Como era expectável, o atraso visual aumentou para valores, no primeiro e segundo cenário de cerca de quatro segundos e no último cenário sensivelmente para cinco segundos.

5.3.3 Testbed 3 - Rede ad-hoc com ligação direta entre os terminais (1 nó)

5.3.3.1 Cenário (a) - Cenário de referência

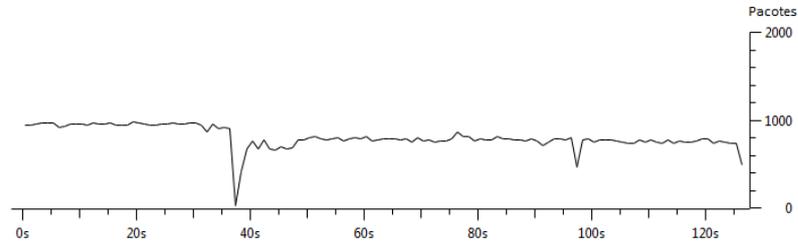


Figura 5.35: Testbed 3 - Cenário (a) - Sinal do emissor

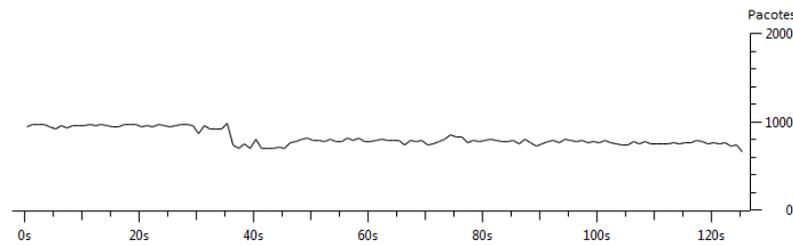


Figura 5.36: Testbed 3 - Cenário (a) - Sinal do receptor

Tabela 5.7: Testbed 3 - Cenário (a) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
804,547	72249	71862	387	0,54%

5.3.3.2 Cenário (b) - Cenário calmo

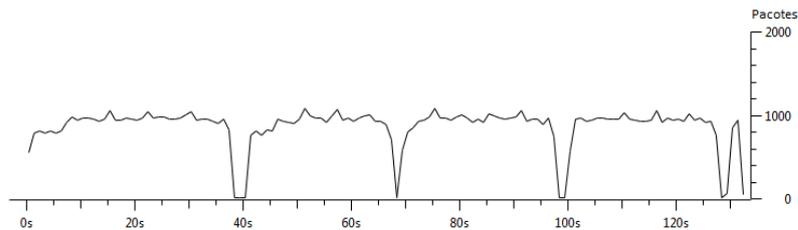


Figura 5.37: Testbed 3 - Cenário (b) - Sinal do emissor

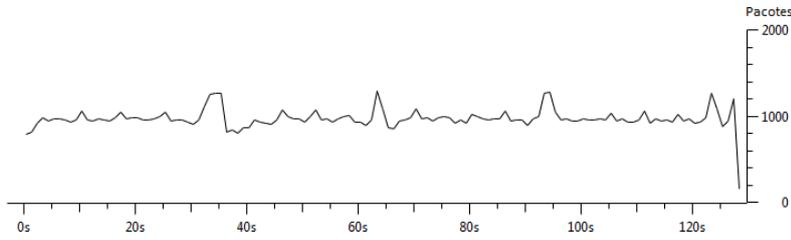


Figura 5.38: *Testbed 3* - Cenário (b) - Sinal do recetor

Tabela 5.8: *Testbed 3* - Cenário (b) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
869,948	80658	78303	2355	2,92%

5.3.3.3 Cenário (c) - Cenário movimentado

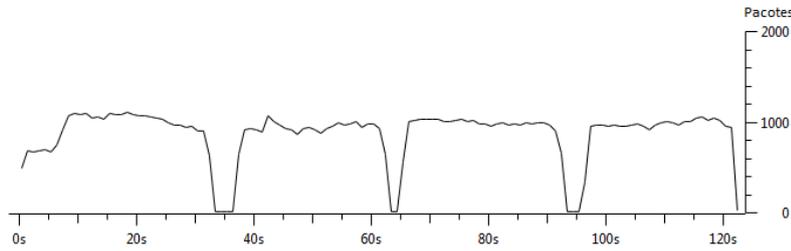


Figura 5.39: *Testbed 3* - Cenário (c) - Sinal do emissor

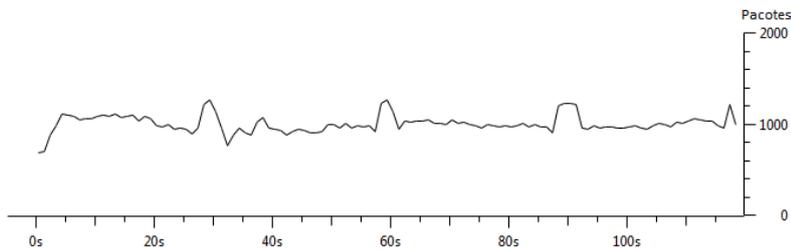


Figura 5.40: *Testbed 3* - Cenário (c) - Sinal do recetor

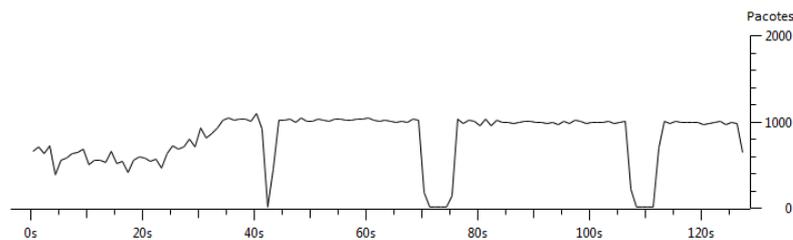
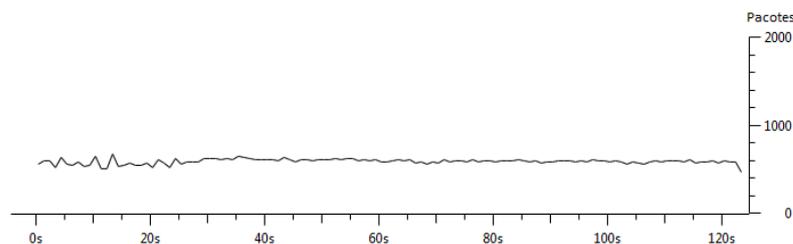
Tabela 5.9: *Testbed 3* - Cenário (c) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
874,193	74982	72932	2050	2,73%

Como se pode observar, neste *testbed* a taxa de perda de pacotes diminuiu face ao *Testbed 2*, com *Access Point*, pois com uma rede *ad-hoc* vai existir comunicação direta entre os terminais, isto é, o tráfego não tem de passar pelo nó de rede. Até mesmo no atraso visual, chegamos a valores no primeiro e segundo cenários de cerca de dois segundos e no terceiro cenário a valores na ordem dos três segundos.

5.3.4 *Testbed 4* - Rede ad-hoc (1 nó) com 2 hops

5.3.4.1 Cenário (a) - Cenário de referência

Figura 5.41: *Testbed 4* - Cenário (a) - Sinal do emissorFigura 5.42: *Testbed 4* - Cenário (a) - Sinal do recetorTabela 5.10: *Testbed 4* - Cenário (a) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
823,152	83731	79995	3736	4,46%

5.3.4.2 Cenário (b) - Cenário calmo

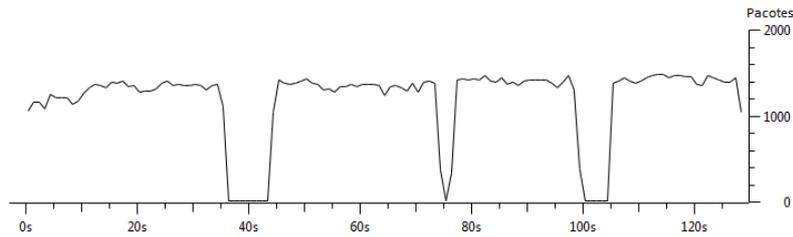


Figura 5.43: *Testbed 4* - Cenário (b) - Sinal do emissor

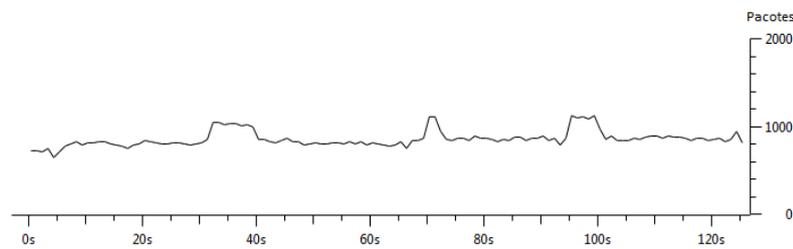


Figura 5.44: *Testbed 4* - Cenário (b) - Sinal do receptor

Tabela 5.11: *Testbed 4* - Cenário (b) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
1200,715	124893	120563	4330	3,47%

5.3.4.3 Cenário (c) - Cenário movimentado

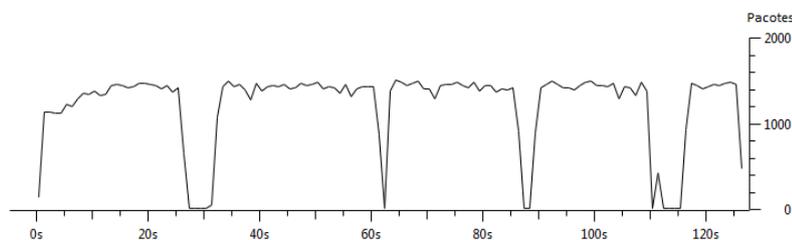


Figura 5.45: *Testbed 4* - Cenário (c) - Sinal do emissor

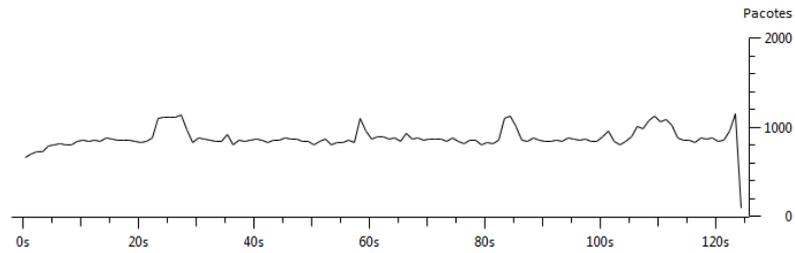


Figura 5.46: *Testbed 4* - Cenário (c) - Sinal do recetor

Tabela 5.12: *Testbed 4* - Cenário (c) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
1238,323	126769	123214	3555	2,80%

Comparando os resultados obtido no *Testbed 3* com os resultados obtidos neste *testbed*, chegamos à conclusão que a taxa de perda de pacotes aumenta. Isto deve-se essencialmente ao facto de o número de *hops* na transmissão ter aumentado. No *testbed* anterior apenas era efetuado um *hop* enquanto que neste *testbed* são efetuados dois *hops*. Ao nível do atraso visual observa-se também um aumento. O primeiro e segundo cenário passam a ter um valor de atraso visual de cerca de quatro segundos enquanto o terceiro cenário tem um valor de cerca de cinco segundos.

Apesar de se ter obtido valores diferentes para os débitos entre este *testbed* e o anterior, é visível que a taxa de perda de pacotes com dois *hops* é muito superior à de um *hop*. A tendência destes resultados é que se aumentarmos os débitos presentes nos valores com um *hop*, a taxa de perda diminui. O que permite concluir que quanto maior for o número de *hops* na transmissão, maior é a taxa de perda de pacotes, pois o débito médio tem influência direta no cálculo da taxa de perda de pacotes.

5.3.5 *Testbed 5* - Rede *ad-hoc* com ligação direta entre os terminais (2 nós)

5.3.5.1 Cenário (a) - Cenário de referência

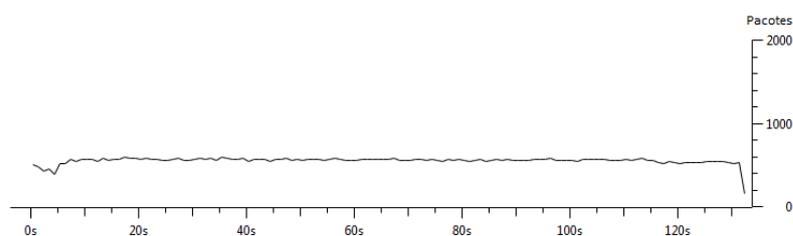


Figura 5.47: *Testbed 5* - Cenário (a) - Sinal do emissor

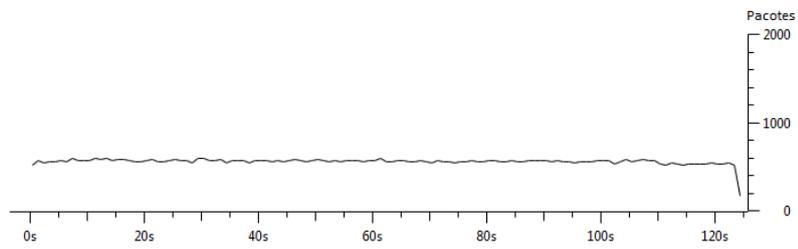


Figura 5.48: *Testbed 5* - Cenário (a) - Sinal do recetor

Tabela 5.13: *Testbed 5* - Cenário (a) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
572,075	51323	48216	3107	6,05%

5.3.5.2 Cenário (b) - Cenário calmo

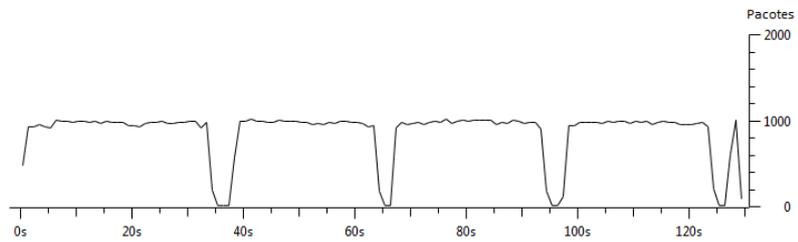


Figura 5.49: *Testbed 5* - Cenário (b) - Sinal do emissor

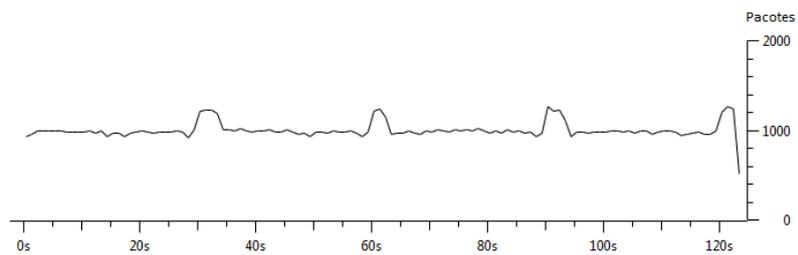
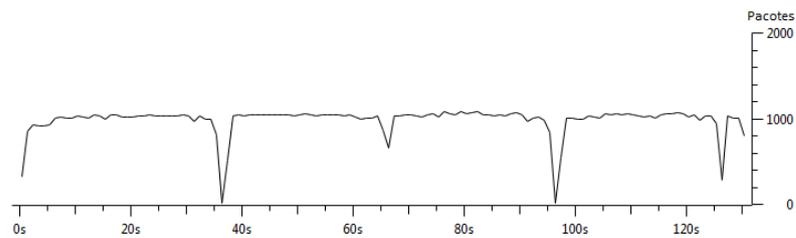
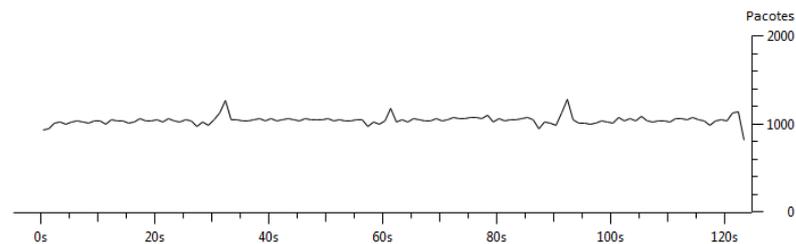


Figura 5.50: *Testbed 5* - Cenário (b) - Sinal do recetor

Tabela 5.14: *Testbed 5* - Cenário (b) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
881,818	77911	74578	3333	4,28%

5.3.5.3 Cenário (c) - Cenário movimentado

Figura 5.51: *Testbed 5* - Cenário (c) - Sinal do emissorFigura 5.52: *Testbed 5* - Cenário (c) - Sinal do recetorTabela 5.15: *Testbed 5* - Cenário (c) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
1002,092	90239	85431	4808	5,53%

O facto de se incluir um novo nó na rede não significa forçosamente uma melhoria significativa em termos do desempenho da rede. Neste caso concreto, verificou-se precisamente uma situação contrária. Comparando as taxas de perda de pacotes nos *Testbeds* 3 e 4, correspondentes a redes *ad-hoc* com um nó, com a mesma taxa no presente *testbed*, observamos um aumento significativo. Também a nível do atraso visual atingimos valores na ordem dos quatro segundos para os dois primeiros cenários e na ordem dos cinco segundos para o terceiro cenário, valores superiores e da mesma ordem de grandeza relativamente aos *Testbeds* 3 e 4, respetivamente.

5.3.6 Testbed 6 - Rede ad-hoc (2 nós) com 3 hops

5.3.6.1 Cenário (a) - Cenário de referência

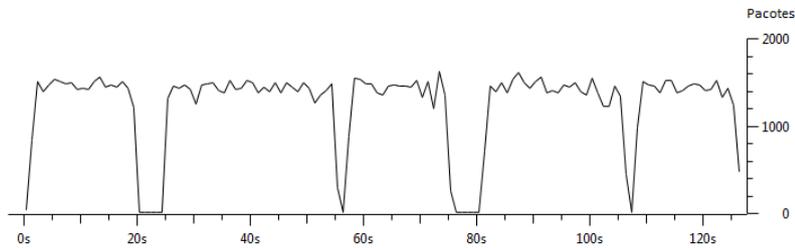


Figura 5.53: Testbed 6 - Cenário (a) - Sinal do emissor

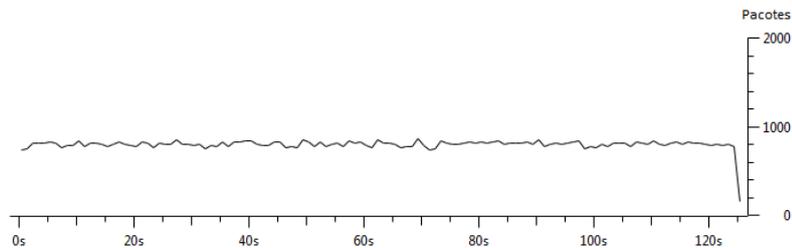


Figura 5.54: Testbed 6 - Cenário (a) - Sinal do receptor

Tabela 5.16: Testbed 6 - Cenário (a) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
1240,497	133488	130473	3015	2,26%

5.3.6.2 Cenário (b) - Cenário calmo

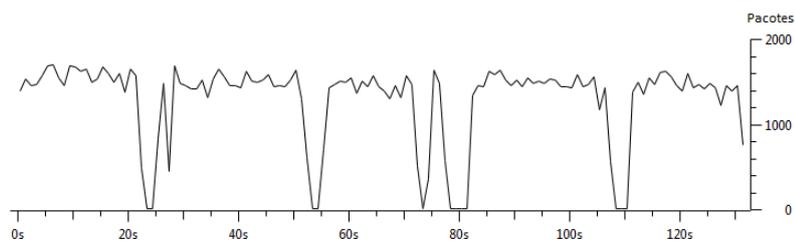


Figura 5.55: Testbed 6 - Cenário (b) - Sinal do emissor

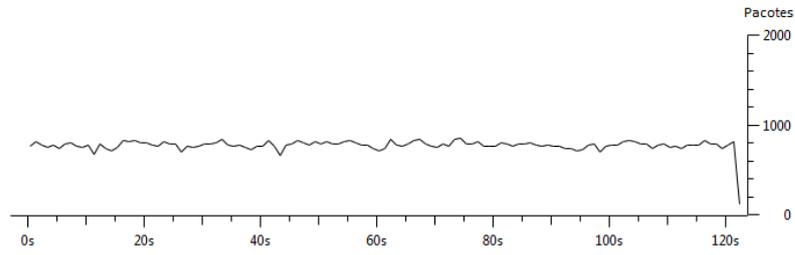


Figura 5.56: Testbed 6 - Cenário (b) - Sinal do recetor

Tabela 5.17: Testbed 6 - Cenário (b) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
1294,041	147963	136955	11.008	7,44%

5.3.6.3 Cenário (c) - Cenário movimentado

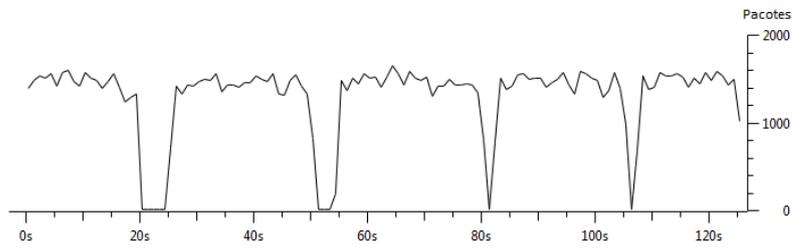


Figura 5.57: Testbed 6 - Cenário (c) - Sinal do emissor

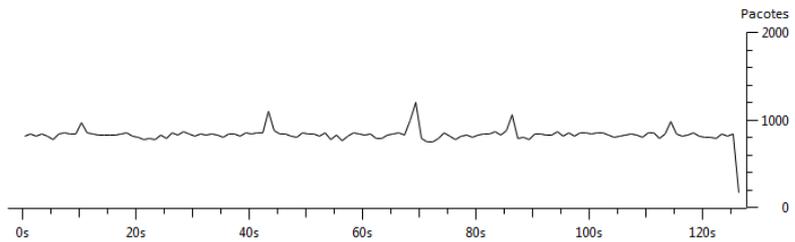


Figura 5.58: Testbed 6 - Cenário (c) - Sinal do recetor

Tabela 5.18: *Testbed* 6 - Cenário (c) - Valores dos parâmetros de rede

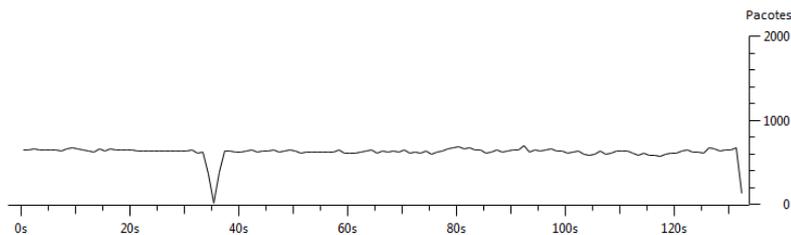
Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
1305,795	142426	142025	401	0,28%

Neste *testbed*, apesar de a rede ser mais complexa, uma vez que a transmissão é obrigatoriamente feita através de três *hops*, a taxa de perda de pacotes não é muito elevada, comparando com os *testbeds* anteriores. Isto é devido ao facto de a distância entre o terminal e o nó ser menor do que a distância entre os terminais. Esta solução mostra-se adequada para casos de comunicações a largas distâncias, em que uma comunicação direta entre os terminais, devido à sua distância, ter um custo elevado na perda de pacotes, sendo preferível a rede efetuar mais *hops* na comunicação entre os terminais. O atraso visual que se observa na transmissão da imagem é no primeiro e no segundo cenário na ordem dos seis segundos e no terceiro cenário na ordem dos sete segundos.

Neste *testbed*, com 3 *hops*, verifica-se que o débito médio aumentou face ao *testbed* de ligação direta com dois nós (*Testbed* 5). Neste caso, impuseram-se rotas estáticas entre os elementos da rede que permitem ao terminal emissor concentrar-se apenas numa só ligação (ligação entre o terminal e o nó). No *Testbed* 5, o terminal emissor tem que gerir três ligações de rede, ou seja, existe uma ligação entre o terminal emissor com cada um dos nós e com o terminal recetor, o que provoca a diminuição do débito para cada ligação.

5.3.7 *Testbed* 7 - Rede emalhada (2 nós)

5.3.7.1 Cenário (a) - Cenário de referência

Figura 5.59: *Testbed* 7 - Cenário (a) - Sinal do emissor

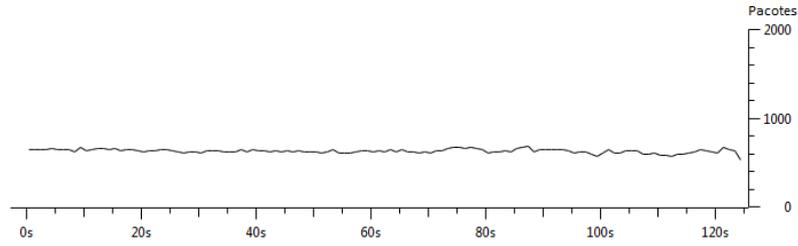


Figura 5.60: *Testbed 7* - Cenário (a) - Sinal do recetor

Tabela 5.19: *Testbed 7* - Cenário (a) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
620,62	58221	54976	3245	5,57%

5.3.7.2 Cenário (b) - Cenário calmo

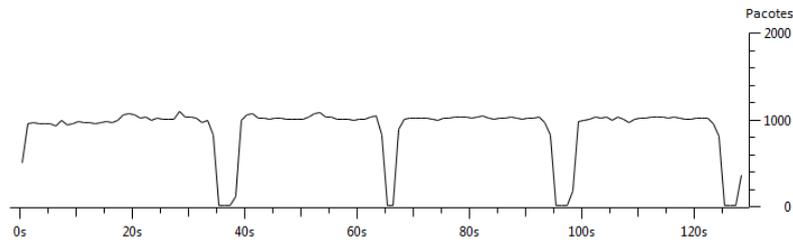


Figura 5.61: *Testbed 7* - Cenário (b) - Sinal do emissor

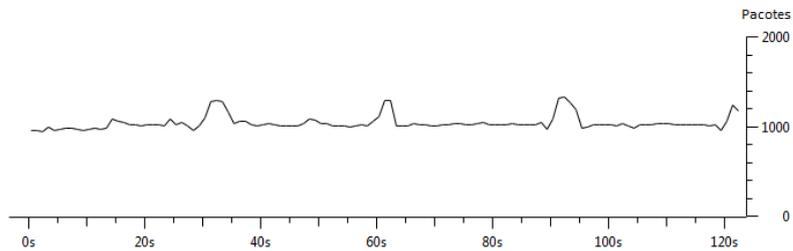
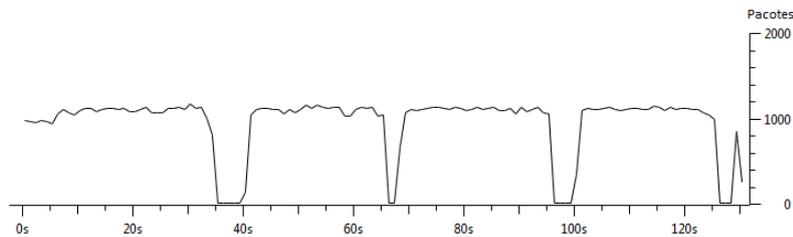
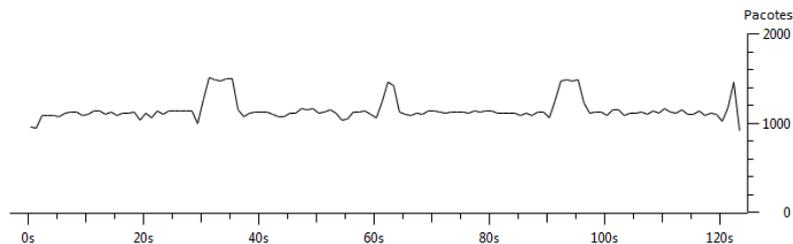


Figura 5.62: *Testbed 7* - Cenário (b) - Sinal do recetor

Tabela 5.20: *Testbed 7* - Cenário (b) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
911,048	79988	76549	3439	4,30%

5.3.7.3 Cenário (c) - Cenário movimentado

Figura 5.63: *Testbed 7* - Cenário (c) - Sinal do emissorFigura 5.64: *Testbed 7* - Cenário (c) - Sinal do recetorTabela 5.21: *Testbed 7* - Cenário (c) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
970,997	87520	83119	4401	5,03%

Face aos resultados anteriores nota-se que a rede emalhada obtém melhor resposta tanto a nível da taxa de pacotes perdidos como também a nível do atraso visual na transmissão do vídeo. Este *testbed* apresenta resultados a nível de atraso visual que mostram um melhoramento significativo, na ordem de um segundo. A rede emalhada tem um atraso visual para o cenário de referência e para o cenário calmo na ordem dos três segundos e para o cenário movimentado na ordem dos quatro segundos.

5.3.8 Testbed 8 - Rede emalhada (2 nós), mas com distância acentuada entre os nós da rede

5.3.8.1 Cenário (a) - Cenário de referência

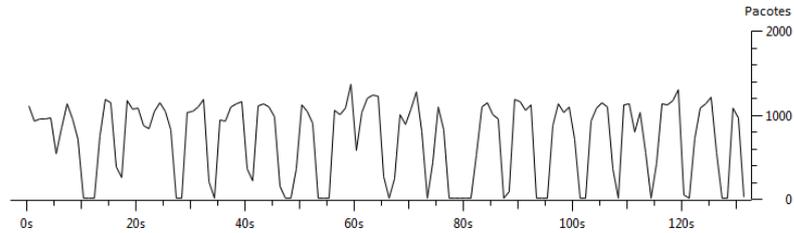


Figura 5.65: Testbed 8 - Cenário (a) - Sinal do emissor

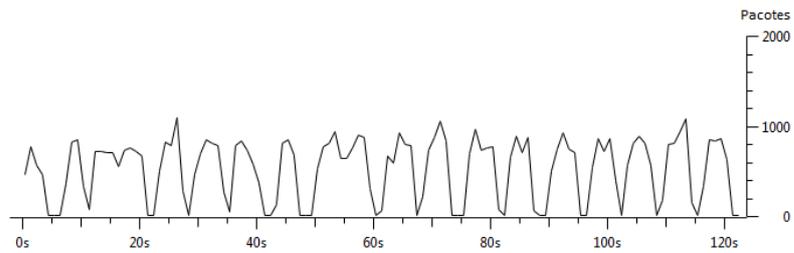


Figura 5.66: Testbed 8 - Cenário (a) - Sinal do receptor

Tabela 5.22: Testbed 8 - Cenário (a) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
725,084	75708	69741	5967	7,88%

5.3.8.2 Cenário (b) - Cenário calmo

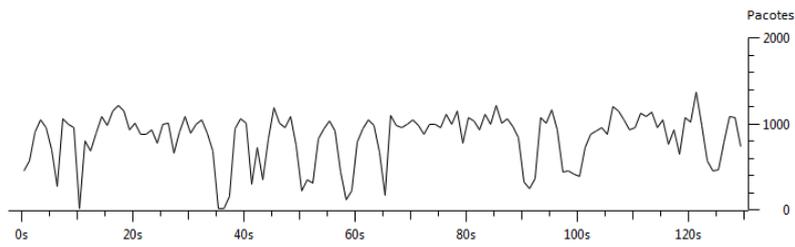


Figura 5.67: *Testbed 8* - Cenário (b) - Sinal do emissor

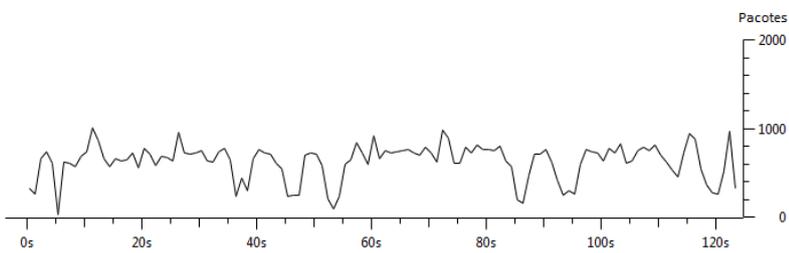


Figura 5.68: *Testbed 8* - Cenário (b) - Sinal do receptor

Tabela 5.23: *Testbed 8* - Cenário (b) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Receptor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
841,695	87612	82026	5586	6,38%

5.3.8.3 Cenário (c) - Cenário movimentado

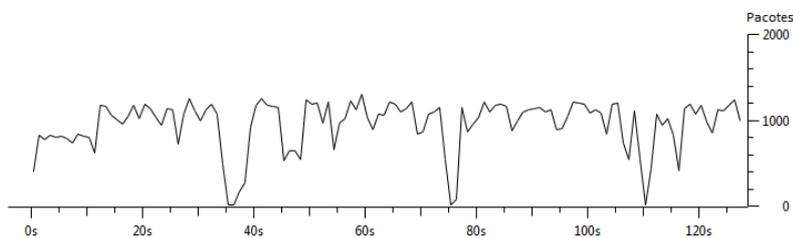


Figura 5.69: *Testbed 8* - Cenário (c) - Sinal do emissor

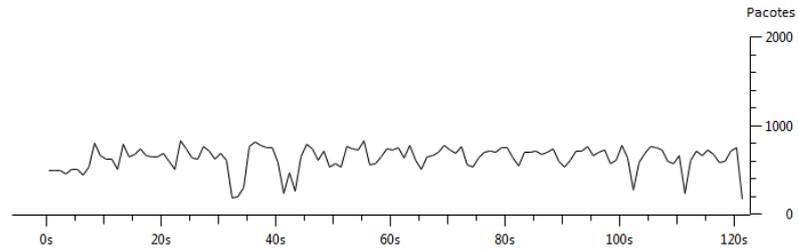


Figura 5.70: *Testbed 8* - Cenário (c) - Sinal do recetor

Tabela 5.24: *Testbed 8* - Cenário (c) - Valores dos parâmetros de rede

Débito Médio (pacotes/segundo)	Pacotes Emissor	Pacotes Recetor	Número de Pacotes Perdidos	Taxa de Perda De Pacotes
958,173	96792	90232	6560	6,78%

Os presentes resultados permitem verificar que apesar de existirem obstáculos no meio em que a rede se encontra, o protocolo *OLSR* permite que a qualidade visual se mantenha. Ao nível do aspeto visual, a imagem mantém uma qualidade interessante e nos cenários verificou-se um atraso na transmissão da ordem dos quatro segundos, apesar de a taxa de perda de pacotes ter aumentado ligeiramente.

Capítulo 6

Conclusões

6.1 Revisão do trabalho desenvolvido

A presente dissertação teve como meta principal desenvolver uma solução para transmissão de vídeo sobre redes emalhadadas. Durante o desenvolvimento do projeto foram definidas diversas etapas para atingir o objetivo ao qual a dissertação se propunha. Estas etapas permitiram compreender progressivamente os diferentes aspetos envolvidos neste estudo.

A primeira etapa permitiu o estudo do *streaming* de vídeo sobre cabo *ethernet*, usando uma rede entre o computador e o *Raspberry Pi*. O objetivo nesta fase foi adquirir o primeiro contacto com os protocolos de transmissão de vídeo no contexto do *Raspberry Pi*.

O próximo passo foi passar para a comunicação sem fios. Neste sentido, a primeira solução foi criar uma rede acedida através de um *Access Point*. Em muitos casos este *Access Point* é um *router* que fornece *internet* aos diversos equipamentos ao qual se encontra ligado. A segunda solução foi criar uma rede *ad-hoc*, com o objetivo de atingir a plataforma base para a construção de uma rede emalhadada. O fundamento das redes *ad-hoc* é a igualdade de direitos entre todos os equipamentos ligados à rede, ou seja, não existe nenhuma estrutura hierárquica para a partilha de recursos.

Foram feitos diversos testes na rede *ad-hoc* para se obterem valores que permitam comparar estas redes com as redes emalhadadas. Os primeiros testes foram efetuados com dois terminais e um nó, testando a comunicação direta entre os terminais e posteriormente a introdução de rotas que obrigassem o tráfego de vídeo a passar pelos três elementos da rede. De seguida, introduziu-se mais um nó de rede, de forma a aumentar a complexidade da rede. Os testes com dois nós comprovaram que se o tráfego tiver de atravessar os quatro elementos da rede então obtêm-se piores resultados a nível visual. No entanto, caso a rede venha a sofrer congestionamentos, poderá esta solução ser eficaz para que um cliente que pretenda aceder à rede não seja rejeitado.

Os diferentes *testbeds* apresentados anteriormente permitiram definir uma solução para a transmissão de vídeo sobre redes emalhadadas, verificando-se melhorias significativas quanto ao aspeto

visual. Além de se obter um melhoramento a nível de imagem, comprovou-se que, face às redes *ad-hoc*, o tempo de resposta na transmissão de vídeo melhorou, ou seja, o atraso visual é menor.

6.2 Contribuições

Devido à enorme massificação das comunicações, cada vez mais comunidades surgem para o desenvolvimento de novas tecnologias e soluções. A comunidade *Raspberry Pi* é uma das mais ativas nos últimos tempos, onde investigadores, e não só, partilham os seus conhecimentos de forma global. Muitos projetos foram e estão a ser desenvolvidos através desta comunidade tendo por base o aparelho *Raspberry Pi*. O tema desta dissertação surge neste contexto, pretendendo usar o *Raspberry Pi* como solução flexível e de baixo custo como elemento central num sistema de transmissão de vídeo sobre redes emalhadas.

Tendo as redes emalhadas um potencial significativo para se imporem no campo das futuras redes *wireless*, esta dissertação pretende ser um contributo para a introdução destas mesmas redes na construção de novas soluções para a transmissão de vídeo em contextos onde não existem redes estruturadas. Da mesma forma, as redes emalhadas podem ser um excelente contributo na comunidade *Raspberry Pi*, permitindo assim a partilha de conhecimentos nesta mesma área usando o aparelho *Raspberry Pi model B* como tecnologia central na construção de futuras redes.

Com as soluções apresentadas na dissertação é possível construir uma rede com os *Raspberry Pis* a desempenharem o papel de nós de rede, ter terminais de captação de vídeo que são também *Raspberry Pis* e ter terminais de acesso para aceder aos conteúdos de *streaming* de vídeo. Pretendeu-se que as soluções apresentadas possibilitem uma implementação a larga escala, nomeadamente à escala de edifícios de grande dimensão como centros comerciais e sedes empresariais.

Um contributo relevante deste trabalho foi o facto de os testes apresentados terem sido feitos em sistemas reais, constituindo desta forma uma mais valia para esta área.

6.3 Trabalho futuro

Na análise efetuada nesta dissertação surgem diversas propostas para o melhoramento e procura de alternativas para a solução desenvolvida. Sendo a investigação em redes emalhadas um dos tópicos mais atual nas telecomunicações, existe um enorme interesse no desenvolvimento desses melhoramentos, tanto em termos de investigação, como também em termos comerciais.

A primeira proposta seria estudar o mesmo problema adicionando mobilidade aos terminais de acesso (quer ao emissor quer ao recetor). O objetivo neste caso seria estudar a influência da mobilidade dos terminais na qualidade de transmissão de vídeo sobre redes emalhadas.

Como segunda proposta seria observar os resultados em situações de acesso concorrido, ou seja, aumentando o número de terminais com capacidade de aquisição de vídeo. Esta situação iria levar a situações de congestionamento da rede, quer no acesso à rede quer entre os nós de rede. O estudo aqui iria incidir sobre as formas de a rede reagir a essas situações de congestionamento.

Outra aspeto cujo estudo seria bastante interessante neste contexto seria o aumento do número

de nós presentes na rede. Este aumento do número de nós iria possibilitar à rede um maior dinamismo, aumentar a sua área de propagação, gerir melhor o tráfego, aceitar o acesso a um número maior de terminais com câmaras e ter um maior número de recetores de sinal. Este estudo iria certamente comprovar a importância do conceito *mesh* neste tipo de comunicações, face às redes *ad hoc* tradicionais.

Referências

- [1] António Jorge Gomes Abelém, Célio Vinicius Neves Albuquerque, Débora Christina Muchaluat Saade, Elisangela Santana Aguiar, Jairo Lino Duarte, José Eduardo Mendonça Da Fonseca, and Luiz Claudio Schara Magalhães. *Redes mesh: Mobilidade, qualidade de serviço e comunicação em grupo*. 2007.
- [2] Nuno Filipe da Costa Santos. *Análise e comparação das tecnologias Wi-Fi e homeplug para transmissão de streams de vídeo*. Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2010.
- [3] Olavo José Luiz Jr and Anfré Luiz Przybysz. *Infra-estrutura e roteamento em redes wireless mesh*. *Synergismus scyentifica UTFPR*, 2(1), 2007.
- [4] Tânia Cláudia dos Santos Pinto Caçada. *Topology Aware Channel Assignment in Single-radio Stub Wireless Mesh Networks*. PhD thesis, Faculdade De Engenharia Da Universidade Do Porto, 2012.
- [5] Jirka Klaue, Berthold Rathke, and Adam Wolisz. *Evalvid—a framework for video transmission and quality evaluation*. In *Computer Performance Evaluation. Modelling Techniques and Tools*, pages 255–272. Springer, 2003.
- [6] D. Saade, C. Albuquerque, L. Magalhaes, D. Passos, J. Duarte, and R. Valle. *Redes em malha: Solução de baixo custo para popularização do acesso à internet no brasil*. *XXV Simpósio Brasileiro de Telecomunicações-SBrT*, pages 1–6, 2007.
- [7] John Bicket, Sanjit Biswas, Daniel Aguayo, and Robert Morris. *Architecture and evaluation of the MIT Roofnet mesh network*. 2005.
- [8] Henrik Lundgren, Krishna Ramachandran, Elizabeth Belding-Royer, Kevin Almeroth, Michael Benny, Andrew Hewatt, Alexander Touma, and Amit Jardosh. *Experiences from the design, deployment, and usage of the UCSB MeshNet testbed*. *Wireless Communications, IEEE*, 13(2):18–29, 2006.
- [9] Richard Draves, Jitendra Padhye, and Brian Zill. *Comparison of routing metrics for static multi-hop wireless networks*. *ACM SIGCOMM Computer Communication Review*, 34(4):133–144, 2004.
- [10] Ian F Akyildiz, Xudong Wang, and Weilin Wang. *Wireless mesh networks: a survey*. *Computer networks*, 47(4):445–487, 2005.
- [11] A Gerkis and J Purcell. *A survey of wireless mesh networking security technology and threats*. *SANS Institute*, 2006.

- [12] G. R. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. P. Costa, and B. Walke. The IEEE 802.11 universe. *IEEE Communications Magazine*, 48(1):62–70, 2010.
- [13] Nuno Filipe Dos Santos Novo. Agregação de acessos em redes multi-rádio. Master's thesis, Faculdade De Engenharia Da Universidade Do Porto, 2012.