

**Faculdade de Engenharia da Universidade do Porto**



**FEUP**

**Multicast em Redes Emalhadas 802.11:  
Novas métricas de encaminhamento e suporte  
eficiente de mobilidade de terminais**

Luís Miguel Pinto Martins

Dissertação realizada no âmbito do  
Mestrado Integrado em Engenharia Electrotécnica e de Computadores  
Major Telecomunicações

Orientador: Prof. Dr. José Ruela  
Co-orientador: Eng.º Rui Campos

Julho de 2011



A Dissertação intitulada

“Multicast em Redes Emalhadas 802.11: Novas métricas de encaminhamento e suporte eficiente de mobilidade de terminais”

foi aprovada em provas realizadas em 18-07-2011

o júri



Presidente Professor Doutor Hélio Mendes de Sousa Mendonça  
Professor Auxiliar do Departamento de Engenharia Electrotécnica e de  
Computadores da Faculdade de Engenharia da Universidade do Porto



Professora Doutora Susana Isabel Barreto de Miranda Sargento  
Professora Auxiliar do Departamento de Electrónica, Telecomunicações e  
Informática da Universidade de Aveiro



Professor Doutor José António Ruela Simões Fernandes  
Professor Associado do Departamento de Engenharia Electrotécnica e de  
Computadores da Faculdade de Engenharia da Universidade do Porto



Mestre Rui Lopes Campos  
Investigador do INESC-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 extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.



Autor - Luís Miguel Pinto Martins

Faculdade de Engenharia da Universidade do Porto



# Resumo

O acesso ubíquo e sem fios à Internet é actualmente um requisito fundamental para os utilizadores. Neste contexto, a tecnologia 802.11 tem-se assumido como uma das principais tecnologias de acesso sem fios. As redes emalhadas 802.11 são encaradas como solução para aumentar a cobertura 802.11 de forma flexível e eficiente economicamente, de forma a responder à crescente procura de acessos 802.11. Recentemente têm sido propostas soluções para a criação automática de redes emalhadas 802.11, com especial destaque para a solução WiFIX que introduz várias melhorias relativamente ao IEEE 802.11s, ainda em fase de pré-ratificação. No entanto, apesar de mais eficiente no que diz respeito à difusão de tráfego *multicast*, o suporte de mobilidade de terminais conduz, em certos casos, a tempos de reacquirição de fluxo *multicast* elevados e a métrica de encaminhamento usada é comum ao tráfego *unicast* e *multicast*. No âmbito desta dissertação pretende-se implementar uma nova solução para a difusão de tráfego *multicast* em redes emalhadas 802.11 com suporte eficiente de mobilidade e considerando novas métricas de encaminhamento *multicast*.



# Abstract

Nowadays the ubiquitous access to the Internet using wireless networks is a fundamental requirement to end-users. In this context, the 802.11 technology has assumed an important role in wireless access network environments. The 802.11 mesh networks have been the major solution to extend the 802.11 coverage in a flexible and economically efficient way, to respond to the increasing demand for 802.11 accesses. Recently there have been several proposed solutions to the automatic creation of mesh networks 802.11, with special emphasis on the WiFIX method, which introduces several key improvements compared to the IEEE 802.11s draft.

But although it is more efficient in what concerns multicast routing, the mobility support still produces, in some cases, a high multicast flux reacquisition time, and the metric used is common to both the unicast and the multicast traffic.

We propose a new solution to forward multicast traffic in 802.11 wireless mesh networks, with efficient mobility support and considering new multicast routing metrics.





# Agradecimentos

Gostaria de apresentar os meus sinceros agradecimentos aos meus orientadores, Prof. Dr. José Ruela e Eng. Rui Campos por toda a ajuda disponibilizada durante a revisão do documento. Gostaria também de agradecer ao Eng. Rui Campos por toda a disponibilidade e ajuda demonstrada durante a realização do trabalho.

O Autor



# Conteúdo

Conteúdo.....	vii
Lista de Figuras .....	ix
Lista de Tabelas.....	xi
Abreviaturas e Símbolos .....	xiii
Capítulo 1 .....	1
Introdução.....	1
1.1 Contextualização .....	1
1.2 Problema.....	2
1.3 Motivação .....	2
1.4 Objectivos .....	3
1.5 Contribuições.....	3
1.6 Estrutura da Dissertação .....	4
Capítulo 2 .....	5
Estado da Arte.....	5
2.1 802.11s .....	5
2.2 WiFIX .....	5
2.3 WiFIX 2.0.....	6
2.4 Métricas de encaminhamento .....	6
2.5 Protocolos de encaminhamento <i>multicast</i> .....	8
2.6 Gestão de mobilidade.....	10
2.7 Conclusão .....	13

Capítulo 3.....	15
Especificação da solução WiFIX 3.0.....	15
3.1 Nova métrica.....	16
3.2 Mecanismo de histerese.....	17
3.3 <i>Soft Topology Change</i> .....	17
3.4 Mecanismo de mobilidade dos terminais.....	18
Capítulo 4.....	21
Implementação da solução WiFIX 3.0.....	21
4.1 Nova métrica.....	22
4.2 Mecanismo de histerese.....	23
4.3 <i>Soft Topology Change</i> .....	24
Capítulo 5.....	27
Avaliação da Solução WiFIX 3.0.....	27
5.1 Comparação entre WiFIX 2.0 e WiFIX 3.0 (CPU Sobrecarregado).....	29
5.2 Comparação entre WiFIX 2.0 e WiFIX 3.0 (Ligação degradada).....	36
5.3 Tempos de mobilidade dos MAP.....	41
Capítulo 6.....	43
Conclusões.....	43
6.1 Concretização dos objectivos.....	43
6.2 Contribuições.....	44
6.3 Trabalho futuro.....	44
Referências.....	45

# Lista de Figuras

Figura 3.1 Exemplo da mobilidade dos MAPs .....	17
Figura 3.2 Exemplo da mobilidade de um terminal dentro da rede emalhada .....	19
Figura 4.1 Modelo das camadas protocolares para solução WiFIX .....	21
Figura 4.2 Fluxograma do mecanismo de histerese .....	23
Figura 4.3 Mecanismo de mobilidade dos MAPs em acção .....	25
Figura 4.4 Fluxograma da função de adição e remoção de grupos <i>multicast</i> .....	26
Figura 5.1 - Topologia da solução WiFIX 2.0 utilizada no <i>test-bed</i> .....	27
Figura 5.2 Estado inicial da rede utilizando a solução WiFIX 2.0 .....	29
Figura 5.3 <i>Unicast Throughput</i> .....	30
Figura 5.4 <i>Broadcast Throughput</i> .....	31
Figura 5.5 <i>Unicast Jitter</i> .....	32
Figura 5.6 <i>Broadcast Jitter</i> .....	33
Figura 5.7 <i>Unicast Packet Loss Ratio</i> .....	34
Figura 5.8 <i>Broadcast Packet Loss Ratio</i> .....	34
Figura 5.9 Latência a 2 saltos.....	35
Figura 5.10 Latência a 1 salto .....	35
Figura 5.11 <i>Unicast Throughput</i> .....	36
Figura 5.12 <i>Broadcast Throughput</i> .....	37
Figura 5.13 <i>Unicast Jitter</i> .....	38
Figura 5.14 <i>Broadcast Jitter</i> .....	38
Figura 5.15 <i>Unicast Packet Loss Ratio</i> .....	39
Figura 5.16 <i>Broadcast Packet Loss Ratio</i> .....	39
Figura 5.17 Latência a 2 saltos .....	40
Figura 5.18 Latência a 1 salto .....	40



# Lista de Tabelas

Tabela 4.1 Peso de cada parâmetro na métrica .....	22
Tabela 5.1 Tempo de Mobilidade dos MAPs .....	41





# Abreviaturas e Símbolos

ADMR	Adaptive Demand-Driven Multicast Routing
AP	Access Point
BASH	Backhaul-Aided Seamless Handoff Scheme
$B_t$	Number of bits in test frame
CAMP	Core-Assisted Mesh Protocol
CBT	CORE Based Tree
CPU	Central Processing Unit
DHCP	Dynamic Host Configuration Protocol
DR	Designated Receiver
$E_{pt}$	Frame error ratio for the test frame size
ENT	Effective Number of Transmissions
Eo11	Ethernet-over-802.11
ETT	Expected Transmission Time
ETX	Expected Transmission Count
FDR	Frame Delivery Ratio
iAWARE	Interference AWARE
IEEE	Institute of Electrical and Electronics Engineers
IGMP	Internet Group Management Protocol
IP	Internet Protocol
HWMP	Hybrid Wireless Mesh Protocol
LAM	Lightweight Adaptive Multicast
MAP	Mesh Access Point
Max_Load	Maximum Load
Mbps	Megabit per second
mETX	modified Expected Transmission Count
MIC	Metric of Interference and Channel-switching
Min_Bw	Minimum Bandwidth
ML	Minimum Loss

MM	Multi Metric
M3	Mesh Mobility Management
ODMRP	On-Demand Multicast Routing Protocol
OLSR	Optimized Link State Routing
OODMRP	Optimized On-Demand Multicast Routing Protocol
$O_{ca}$	Channel Access Overhead
$O_p$	Protocol Overhead
$r$	Bit rate em Mbps
RMTF	Reliable Multicast Transport Protocol
SMob	Selective Multicast with Mobility Support
SNR	Signal-to-Noise Ratio
SPT	Shortest Path Tree
TMIP	Transparent Mobile IP
TORA	Temporal-Ordered Routing Algorithm
$T_1$	Renewal Time
$T_2$	Rebinding Time
VoIP	Voice over IP
WCETT	Weighted Cumulative ETT
WiFIX	WiFi Network Infrastructure Extension
WiFIX 2.0	WiFi Network Infrastructure Extension Plus
WiFIX 3.0	WiFi Network Infrastructure Extension Enhanced
WiFi	Wireless Fidelity

# Capítulo 1

## Introdução

### 1.1 Contextualização

A proliferação de terminais móveis com acesso à Internet acentuou a importância das redes sem fios, nomeadamente das redes 802.11. A utilização de redes emalhadas 802.11 é vista como uma solução simples, flexível e de baixo custo para o aumento da cobertura 802.11. Uma rede emalhada 802.11 consiste num conjunto de nós, designados *Mesh Access Points* (MAPs), que se configuram automaticamente e comunicam via rádio; pelo menos um MAP está ligado directamente à rede estruturada, designando-se este(s) nó(s) por *gateway*. Com base no conceito de redes emalhadas foi proposto em [1] a solução *WiFi Network Infrastructure Extension* (WiFIX) para a entrega eficiente de tráfego *unicast*.

Paralelamente a diversificação recente dos serviços multimédia disponíveis ao utilizador, tais como videoconferências, VoIP ou *streaming* de vídeos, torna o uso do *multicast* essencial para a difusão desses conteúdos. Existem contudo problemas associados ao *multicast* em redes emalhadas sem fios, tais como a gestão eficiente dos recursos da rede e o uso de métricas de encaminhamento *multicast* adequadas ao contexto de uma rede emalhada 802.11. Este aspecto afecta directamente o desempenho e a escalabilidade da rede.

Este trabalho tem assim dois objectivos: 1) a definição de uma nova métrica de encaminhamento para a solução WiFIX, no sentido de garantir a entrega eficiente do tráfego *multicast* dentro de uma rede emalhada 802.11; 2) a definição de um mecanismo eficiente de suporte à mobilidade dos terminais.

## 1.2 Problema

Este trabalho visa resolver dois problemas no âmbito das redes emalhadas 802.11: 1) a alteração da topologia lógica da rede em função da identificação de ligações degradadas, sem prejuízo para a estabilidade da rede; 2) a gestão da mobilidade dos terminais que permita suportar serviços sensíveis ao atraso. Numa rede emalhada 802.11, os MAPs utilizam apenas rádio frequência para comunicar entre si. Por isso, o meio de transmissão caracteriza-se muitas vezes pela instabilidade e pelos recursos limitados, o que acentua a importância de transmitir apenas o tráfego necessário para obter o melhor desempenho possível da rede. Para conseguir isso é necessário criar uma topologia lógica sobre a qual seja possível calcular o caminho mais curto entre fonte e receptor(es) em tempo útil, sem que este apresente *loops*, utilizando métricas apropriadas. Estas métricas são usadas para caracterizar o custo das ligações formadas entre os MAPs.

Para a gestão de mobilidade dos terminais em redes emalhadas é necessário ter em consideração dois aspectos que podem causar atrasos excessivos: o primeiro está relacionado com o facto de os terminais terem de sondar cada canal que possam utilizar para descobrir qual o AP que apresenta a melhor ligação. Numa rede que suporte múltiplos canais este problema será o que mais contribui para o atraso durante a associação. O segundo está relacionado com a actualização das tabelas de encaminhamento dos MAPs da rede emalhada, para que estas reflectam a alteração da posição de um terminal.

Por fim, existe o problema de saber qual o tráfego a ser entregue no terminal, quando este se associa a um novo AP. Para o resolver é necessário que a rede tenha conhecimento de informação essencial tal como os grupos *multicast* aos quais o terminal estava associado no antigo AP e adopte medidas assim que perceba que o terminal se moveu, para que o atraso na entrega desse tráfego seja o menor possível. Desta forma é reduzida mais uma possível fonte de atraso.

## 1.3 Motivação

Com o papel cada vez mais importante de tráfego *multicast* como ferramenta utilizada para distribuir conteúdos multimédia para uma grande variedade de dispositivos móveis e fixos em ambiente IP, torna-se fundamental que existam mecanismos que suportem este tipo de tráfego de forma eficiente.

Por outro lado, a proliferação de terminais móveis e a falta, hoje em dia, de mecanismos de suporte à mobilidade nos protocolos adoptados em redes sem fios torna também este um assunto de grande importância, pois neste momento já existem serviços bastante exigentes em termos de latência e *jitter*, como por exemplo VoIP.

A resolução destes problemas representa a principal motivação para o trabalho aqui desenvolvido.

## 1.4 Objectivos

Nesta dissertação pretende-se melhorar a solução WiFIX 2.0 (também designado de WiFIX+) nomeadamente implementando uma métrica inteligente, que deve diferenciar as ligações entre MAPs com base na qualidade do caminho que apresenta até ao *gateway*. Este mecanismo deve evitar zonas de congestionamento de tráfego ou ligações com má qualidade, devido a possíveis interferências. Deve também ser criado um novo mecanismo para a gestão de mobilidade de terminais, mais eficiente do que apresentado na proposta do WiFIX 2.0.

## 1.5 Contribuições

### 1.5.1 Métrica de encaminhamento

No âmbito deste trabalho será descrita uma nova métrica inteligente de encaminhamento que identifica e evita ligações degradadas e/ou nós com elevada carga, distribuindo equitativamente a carga oferecida à rede pelos nós, não comprometendo a estabilidade da rede emalhada.

### 1.5.2 *Soft Topology Change*

A mobilidade dos MAPs traz problemas com a gestão das árvores *multicast* que se criam para cada grupo *multicast*. Estas baseiam-se na árvore topológica da rede, mas sempre que um MAP muda de pai, a topologia da rede altera-se, podendo desactualizar árvores *multicast*, o que implicaria uma nova associação aos grupos *multicast* por parte dos terminais. É portanto necessário proceder à actualização dessas árvores sempre que haja a mudança de um pai, de uma forma eficiente, sem introduzir um *overhead* de controlo significativo na rede, e de forma transparentes para os terminais. Este novo mecanismo que suporta a mobilidade dos MAPs é designado *Soft Topology Change*.

### 1.5.3 Implementação da métrica e do mecanismo *Soft Topology Change*

Com recurso ao código da solução WiFIX 2.0 pretende-se definir e implementar a nova métrica e o mecanismo *Soft Topology Change* para que seja possível validar esses mecanismos com testes práticos, simulando cenários reais, utilizando um *test-bed* criado para esse efeito.

## 1.6 Estrutura da Dissertação

Este documento está organizado em seis capítulos, incluindo o presente Capítulo 1 de Introdução. No Capítulo 2 é exposto o estado da arte em relação a métricas de encaminhamento em redes sem fios, encaminhamento de tráfego *multicast* em redes sem fios e gestão de mobilidade em geral. No Capítulo 3 é especificada a solução WiFIX 3.0, explicando os principais mecanismo criados para a nova solução. No Capítulo 4 detalha-se a implementação da solução WiFIX 3.0. O Capítulo 5 apresenta e discute os resultados obtidos nos testes realizados com o novo protocolo. Por fim, no Capítulo 6, são apresentadas as conclusões finais e discutido o trabalho futuro.

# Capítulo 2

## Estado da Arte

### 2.1 802.11s

O *draft* IEEE 802.11s [2] define uma solução para o encaminhamento de tráfego em redes emalhasadas 802.11, com o objectivo de estender uma rede com fios. Para tal, utiliza um protocolo de encaminhamento denominado *Hybrid Wireless Mesh Protocol* (HWMP), que cria uma topologia lógica em árvore, onde o AP que desempenha o papel de *root* tem uma ligação directa à rede com fios. Para a criação das ligações está proposta a métrica *airtime link*, que calcula o custo de uma ligação considerando o *data rate*, *overhead* e o *frame error ratio*.

Para lidar com o tráfego *multicast*, o IEEE 802.11s recorre a *flooding* puro, um método ineficaz e que compromete a escalabilidade da rede, pois cada MAP faz *broadcast* desse tráfego para os vizinhos, o que provoca repetição de tráfego nos MAPs e ocupação desnecessária da banda. Em compensação é fácil de implementar e suporta nativamente mobilidade dos terminais, visto que todos os MAPs recebem o tráfego de todos os grupos *multicast*.

### 2.2 WiFIX

A solução WiFIX, proposta em [1], concentra-se tal como o IEEE 802.11s no encaminhamento de tráfego *unicast*, embora apresente melhorias nessa área. Para tal introduz dois mecanismos: o *Active Topology Creation and Maintenance* (ATCM) é um mecanismo de encapsulamento das tramas dentro da rede WiFIX. O ATCM cria uma topologia lógica da rede em árvore hierárquica, onde cada nó tem um pai (excepto o *gateway*) e pode ou não ter filhos, utilizando uma mensagem única e periódica, designada *Topology Refresh* (TR). Esta mensagem transporta informações usadas para criar túneis *Ethernet over 802.11* (Eo11), que não são mais que ligações virtuais entre dois MAPs (*Mesh Access Points* - designação utilizada

no WiFIX para os APs) vizinhos, ou para informar se um determinado MAP será um pai ou filho de um MAP vizinho. Já o mecanismo de encapsulamento cria um novo cabeçalho nas tramas para permitir o encaminhamento *multi-hop* dentro da rede WiFIX. O novo cabeçalho tem informação sobre o nó de origem e de destino, entre outras informações essenciais para o encaminhamento. Dentro dessa nova trama existe o campo *Frame Body*, que transporta a trama original. Esta solução oferece melhorias na gestão de tramas *broadcast* pois apenas os nós com filhos irão encaminhar estas tramas, melhorando substancialmente o desempenho da rede.

## 2.3 WiFIX 2.0

Visto que o WiFIX não contempla a gestão de tráfego *multicast* (tratando como tráfego *broadcast*), foi proposta a solução WiFix+ [3], designada WiFIX 2.0 daqui em diante. Esta evolução pretende rectificar essa lacuna e adicionar gestão de mobilidade dos terminais na rede, com base no mecanismo *Selective Multicast with Mobility Support (SMob)*. Este constrói árvores únicas para cada grupo *multicast* diferente recorrendo a um mecanismo designado *IGMP snooping*, utilizando a árvore gerada pelo mecanismo ATCM como base. O mecanismo de *IGMP snooping* recorre à violação do princípio de independência entre camadas do modelo OSI para inspeccionar cabeçalhos dos níveis superiores, de maneira a identificar as mensagens *IGMP Report* enviadas pelas estações, permitindo a construção de uma tabela que estabelece a correspondência entre cada grupo *multicast* e os MAPs filho que os recebem. Isto permite um encaminhamento eficiente das tramas *multicast*, pois restringe este tráfego apenas aos nós necessários. Já a mobilidade dos terminais é suportada usando o mecanismo de *DHCP Snooping*, que consiste no envio de uma mensagem *IGMP Query general* aquando do envio da mensagem *DHCP ACK* para um terminal, após a associação com o respectivo MAP. Para que tal aconteça é necessário que antes seja consultada uma tabela de mobilidade existente em cada MAP, que permite determinar se o terminal se associou recentemente ao MAP actual. Caso seja esse o caso, o MAP envia uma mensagem *IGMP Query* em *unicast* e o terminal é forçado a reassociar-se aos grupos *multicast* que pretende receber. Contudo, esta solução usa como métrica simplesmente o número de saltos (*hop count*).

## 2.4 Métricas de encaminhamento

As métricas de encaminhamento existentes para redes com fios, tal como a métrica *hop count*, ou seja, o número de nós pelo qual um pacote terá de passar até chegar ao destino, não se adequam a redes sem fios. Isto porque nestas redes existem problemas mais exigentes como a latência ponto-a-ponto (mais elevada nestes meios), a qualidade dos canais (muito variável ao longo do tempo), a instabilidade do meio e a mobilidade dos terminais.

Para resolver esses problemas têm sido propostas várias métricas [4], tais como a *Expected Transmission Count (ETX)*, *Minimum Loss (ML)*, *Expected Transmission Time (ETT)*, *Weighted Cumulative ETT (WCETT)*, *Metric of Interference and Channel-switching (MIC)*, *modified ETX*



(mETX), *Effective Number of Transmissions* (ENT) e *Interference Aware* (iAWARE). As três primeiras métricas são consideradas simples de implementar, enquanto as restantes envolvem mais esforço computacional, já que consideram mais parâmetros para o cálculo do custo final. As métricas ETX e ML focam-se na qualidade dos canais entre nós vizinhos, usando o rácio de entrega de mensagens periódicas como único parâmetro, mas enquanto o ETX faz a escolha da ligação com menor custo em cada nó, o ML procura o percurso com menor custo no total. A simplicidade destas métricas permite reduzir o *overhead* de controlo existente na rede, mas ignora problemas como a variação frequente dos *data rates* nos canais. Para solucionar esse problema foi proposta a métrica ETT, que pode ser calculada de duas maneiras: a primeira consiste em utilizar a métrica ETX multiplicada por um valor  $t$  ( $ETT = ETX \times t$ ). Este valor é obtido dividindo o tamanho fixo de um pacote ( $S$ ) pela largura de banda estimada ( $B$ ) em cada ligação ( $t = \frac{S}{B}$ ). Essa estimativa é calculada enviando duas mensagens periódicas de tamanhos diferentes e fixos em *unicast* para cada vizinho. Estes calculam o intervalo de chegada entre as duas mensagens e enviam essa informação ao nó de origem, que calcula a largura de banda dividindo o tamanho da mensagem maior pelo menor intervalo obtido em cada ligação. Já na segunda forma é considerada a perda de *data* e *ACK frames* para cada vizinho. Para calcular a perda de *data frames* é feito o *broadcast* periódico de pacotes do mesmo tamanho que *data frames*, um pacote por cada *data rate* definido na norma usada. Já para estimar a perda de *ACK frames* são enviados pacotes do mesmo tamanho que estas tramas à frequência básica da rede. O custo final é o inverso da multiplicação do melhor *throughput* possível pela probabilidade de receber um *ACK* para confirmação. Portanto, a rota escolhida será a que apresente o menor custo. Mas nenhuma das métricas anteriores tem em conta a interferência gerada quando dois nós transmitem tráfego do mesmo fluxo (intra-fluxo), ou de fluxos diferentes (inter-fluxo). Para lidar com tráfego do mesmo fluxo foi proposto o WCETT, que tem em conta o atraso total e a diversidade de canais ao longo do percurso. No entanto, esta métrica não garante o caminho mais curto para o destino nem evita a interferência inter-fluxo. A métrica MIC corrige esses problemas, ao incorporar a métrica ETT, considerando o número de nós adjacentes para calcular a interferência e usando um esquema de nós virtuais para garantir percursos de custo mínimo. Já as métricas mETX e ENT consideram a rápida variação da qualidade dos canais sem fios, algo que métricas que usem a técnica da janela deslizante podem não conseguir controlar sem *overhead* excessivo. A mETX funciona ao nível do bit, enviando mensagens (com estrutura conhecida) periódicas e identificando onde ocorrem os erros e como estes se propagam entre transmissões sucessivas, calculando a partir daí a probabilidade de erro de bit. Já ENT contabiliza o número de retransmissões por ligação, e limita o cálculo de rotas a ligações que apresentem um número de retransmissões aceitável, que depende dos requisitos das camadas superiores. A métrica iAWARE utiliza a relação sinal-ruído (SNR) e o número de

retransmissões para cada vizinho para calcular o custo da métrica, o que faz com que este considere a interferência intra-fluxo e inter-fluxo, a instabilidade do meio e a latência.

No IEEE 802.11s é utilizada a métrica *airtime*, cuja descrição pode ser encontrada em [5], e que utiliza cinco parâmetros: o *overhead* do protocolo ( $O_p$ ) e do acesso ao canal ( $O_{ca}$ ), o número de *bits* na mensagem de teste utilizada ( $B_t$ ), o *bit rate* em Mbps ( $r$ ), e o número de mensagem de teste com erros ( $e_{pt}$ ). O custo de cada ligação é dado pela fórmula:

$$c_a = \left[ o_{ca} + o_p + \frac{B_t}{r} \right] \times \frac{1}{1 - e_{pt}}$$

Desta forma a métrica contabiliza a interferência no meio e a latência entre ligações, mas para utilizar esta métrica é necessário utilizar mensagens de teste que introduzem *overhead* significativo. Estas mensagens podem ser enviadas em *broadcast* ou *unicast*, e caso sejam enviadas em *broadcast* o *overhead* será reduzido, mas as mensagens serão enviadas usando o menor *data rate* possível, onde a perda de pacotes poderá ser menor. Se as mensagens forem enviadas em *unicast*, todas as ligações entre pares de APs serão testadas individualmente, o que aumenta ainda mais o *overhead*, afetando a escalabilidade da rede.

Por fim em [5] é também descrita uma nova métrica, designada *Multi-Metric* (MM). Esta utiliza três parâmetros: a largura de banda residual mínima disponível para um nó no percurso (*Min\_Bw*), a carga máxima de um nó no percurso (*Max\_Load*) e o rácio de entrega das tramas (*FDR*). O custo é dado pela fórmula:

$$c_a = \alpha \cdot Min_{BW} - \beta \cdot Max_{Load} + \gamma \cdot FDR$$

onde  $\alpha$ ,  $\beta$  e  $\gamma$  são factores de correcção e têm de satisfazer a condição:

$$|\alpha| + |\beta| + |\gamma| = 1$$

A mensagem de teste é transmitida em *broadcast* apenas quando não existe transmissão de dados, o que significa que apenas testa o *data rate* mínimo da rede, e de forma irregular, o que pode não reflectir o estado real da rede.

## 2.5 Protocolos de encaminhamento *multicast*

Os protocolos de encaminhamento *multicast* para redes sem fios podem ser separados em dois grupos, os que utilizam uma estrutura lógica em árvore (*tree-based*) e os que utilizam uma estrutura em malha (*mesh-based*). Os protocolos *tree-based*, propostos em [6] [7] [8] [9], são mais fáceis de implementar e implicam menor *overhead*, além de assegurarem percursos sem *loops*. Já os protocolos *mesh-based*, descritos em [10] [11] [12], são mais robustos a falhas de nós, pois oferecem diversos percursos diferentes para comunicação entre nós e suportam mobilidade entre nós.

Em [6] é proposto o protocolo *Adaptive Demand-Driven Multicast Routing* (ADMR) que tem como objectivo reduzir ao máximo as componentes pró-activas (*non-on-demand*) do protocolo. Isso permite reduzir o *overhead* de controlo, melhorando a eficiência do protocolo. Para atingir esse objectivo não utiliza mensagens periódicas de controlo, adapta o tempo de expiração das ligações entre APs consoante a aplicação, utiliza confirmação passiva, entre

outras características. Para lidar com a mobilidade frequente dos nós opta por fazer *flooding* na rede, retornando ao funcionamento normal após um determinado intervalo caso os nós estabilizem, e portanto não apresenta uma forma eficiente de lidar com a mobilidade constante dos terminais

Em [7] é criada uma integração dos protocolos *CORE Based Tree* (CBT), um protocolo de encaminhamento *multicast* e o *Temporal-Ordered Routing Algorithm* (TORA), designado como *Lightweight Adaptive Multicast* (LAM). O objectivo desta proposta é criar um protocolo para redes *ad-hoc* de grande dimensão em constante alteração, sem criar uma grande penalização no *overhead*. Este protocolo utiliza um nó especial chamado CORE, que será usado como centro para as árvores *multicast* criadas e portanto, caso um nó falhe, não existem muitos nós que dependam deste, ao contrário do que pode acontecer nas estruturas em árvore tradicionais. Também o tráfego de controlo do LAM é directamente proporcional ao número de alterações que ocorrem na rede, portanto caso a rede esteja estável não existem mensagens de controlo e, para simplificar, todos os nós (excepto o CORE) só utilizam três tipos de mensagens. A utilização do TORA, que atribui um parâmetro de identificação a cada nó chamado *height*, permite a criação de árvores sem *loops*. Algumas das desvantagens deste protocolo são a limitação imposta pelo uso de apenas um CORE por árvore *multicast*, o que pode criar um *bottleneck* na rede, a implementação deste protocolo em todos os terminais que queiram usar a rede, a dependência do TORA como protocolo para tráfego *unicast* e o fato de não haver um mecanismo para suportar a mobilidade dos terminais, nem o uso de métricas apropriadas para o encaminhamento de tráfego *multicast*.

Em [8] é descrito o *Reliable Multicast Transport Protocol* (RMTP). O RMTP recorre a três grandes funcionalidades: 1) a informação mantida pelos nós não depende do número total de nós presentes na árvore *multicast*, portanto não é necessário actualizar a informação em cada nó sempre que há uma alteração na topologia; 2) a responsabilidade de garantir um fluxo sequencial está do lado dos receptores, libertando assim recursos na fonte; 3) os receptores estão agrupados em regiões e cada região terá um *Designated Receiver* (DR), que é responsável por fazer *cache* dos pacotes recebidos pela fonte e responder aos pedidos de reenvio de pacotes dos receptores dessa região. Contudo, este protocolo não prevê o suporte de mobilidade de terminais, delegando essa responsabilidade num gestor de sessões (*Session Manager*) e, tal como no protocolo anterior, é necessário implementar o protocolo em todos os dispositivos que utilizem a rede.

A proposta apresentada em [9] descreve o MeshCAST, um protocolo *multicast* reactivo para redes emalhadas onde os APs da rede utilizam múltiplas interfaces. Para criação dos grupos *multicast* é utilizada uma topologia em árvore, e são apresentadas várias alternativas possíveis para a métrica de atribuição de canais nos APs e para a métrica de gestão das árvores *multicast*. Quando comparado com propostas que utilizam apenas um canal na rede emalhada, esta solução apresenta melhores resultados em termos de *throughput*, rácio de

perda de pacotes e atraso. No entanto, esta abordagem levanta problemas quando existe mobilidade dos terminais, devido ao uso de vários canais, algo que não é resolvido nesta proposta.

Em [10] é proposto o primeiro protocolo para redes *ad-hoc* a utilizar uma topologia *mesh-based*, o *Core-Assisted Mesh Protocol (CAMP)*. O principal objectivo da proposta é demonstrar que a topologia *mesh-based* é uma alternativa viável à topologia *tree-based*. Para isso define mecanismos que permitem criar caminhos sem *loops* e, dentro de um tempo finito, é possível encontrar o caminho mínimo na rede *mesh* criada. Para ser escalável utiliza nós especiais chamados *core* que limitam o tráfego de controlo dos receptores para se juntarem aos grupos *multicast*. Como desvantagens, esta solução apresenta a necessidade de um protocolo de encaminhamento *unicast* que forneça distâncias correctas para destinos conhecidos num tempo finito, não prevê um mecanismo para a mobilidade dos terminais e utiliza a métrica *hop count* para calcular as distâncias.

Em [11] é apresentado o protocolo *On-Demand Multicast Routing Protocol (ODMRP)*, um protocolo que utiliza técnicas reactivas para reduzir o *overhead* na rede e melhorar a escalabilidade. O ODMRP utiliza o conceito de *forwarding nodes*, nós que ficam responsáveis por reencaminhar os pacotes *multicast*, para construir uma topologia lógica em *mesh* que seja flexível e robusta à mobilidade dos nós. Este protocolo é capaz de lidar com tráfego *unicast* e *multicast*, mas também pode ser usado em conjunto com qualquer protocolo *unicast*. Apesar da robustez que o protocolo apresenta, não existe um mecanismo para lidar com a mobilidade dos nós, o que significa que pode haver atrasos excessivos na entrega de tráfego ao terminal após este ter efectuado uma transição entre APs. O protocolo também não apresenta uma métrica específica para lidar com o tráfego *multicast*.

Por fim, em [12] é descrito o protocolo *Optimized On-Demand Multicast Routing Protocol (OODMRP)* onde são propostas melhorias ao ODMRP para redes emalhadas, nomeadamente a redução de *forwarding nodes* em cada grupo *multicast*, aproveitando a estabilidade que advém de usar uma rede emalhada. Essa alteração minimiza a carga na rede e reduz o congestionamento da rede mas, tal como o ODMRP, este protocolo não introduz um mecanismo de gestão eficiente de mobilidade, nem apresenta métricas específicas para lidar com o tráfego *multicast*.

## 2.6 Gestão de mobilidade

Nesta secção são apresentadas soluções para o problema da gestão de mobilidade de terminais em redes emalhadas. Neste contexto, por um lado é necessário minimizar o tempo sem conectividade entre APs, quando o terminal se move; por outro, é necessário minimizar o tempo de actualização das tabelas de encaminhamento dos MAPs, para determinar a nova localização do terminal móvel.

Em [13] é apresentada a proposta *Backhaul-Aided Seamless Handoff Scheme (BASH)*, um mecanismo que permite fazer o *handoff* a nível 2 e facilitar o *handoff* a nível 3. Para tal

supõe-se que o terminal tem capacidades de modulação múltipla, ou seja, é capaz de suportar 802.11a, b e g, e portanto é capaz de trocar entre essas modulações quando necessário. A solução passa por explorar o canal de *backhaul*, ou seja, o canal usado pelos APs da rede emalhada, considerando que o canal usado entre AP e terminais será diferente. É usada uma solução *make-before-break*, onde o terminal (cujo pedido é enviado no canal de *backhaul*) ou o AP onde está registado podem iniciar o processo de *handoff*, e a escolha do novo AP será da responsabilidade do AP antigo, usando o SNR como métrica. Desta forma, a rede sabe antecipadamente a nova direcção do fluxo e pode tomar decisões *à priori*, minimizando o atraso de reconfiguração. Contudo, esta solução contempla apenas o tráfego *unicast*.

Em [14] é proposto o iMesh, uma solução de nível 3 que implementa mobilidade transparente, ou seja, não existem configurações adicionais nos terminais. Para tal utiliza o modo infraestrutura em cada AP para usar a mobilidade que o 802.11 implementa, no nível 2, que não é mais que o envio de uma trama *probe request* em *broadcast*, à qual todos os APs da área respondem com um *probe response*, e cuja escolha dependerá do SNR entre terminal e AP. Para mobilidade em nível 3 são usados dois protocolos, *Transparent Mobile IP* (TMIP) e *Optimized Link State Routing* (OLSR). O primeiro é semelhante ao protocolo *Mobile IP* (MIP), mas não necessita de ser implementado nos terminais. Tal como no MIP existe um *home AP* para cada terminal e um *Mobile Location Register* (MLR) que contém as relações entre terminais e *home APs*. Quando o terminal se move o novo AP irá contactar o MLR para determinar qual o *home AP* do terminal e posteriormente irá ser criada uma ligação entre os dois APs para que o *home AP* possa enviar tráfego que lhe chegue e seja dirigido ao terminal, enquanto as comunicações do terminal no AP visitado não necessitam de passar pelo *home AP*. Já o segundo protocolo cria em cada nó ligações lógicas para os vizinhos e uma tabela que faz o mapeamento IP-para-MAC de todos os terminais que a ele estão ligados, para garantir conectividade na rede. Cada nó terá um conjunto de endereços IP independentes, e sempre que um terminal se associe a um AP é enviado um *link update* com o endereço MAC do terminal para que todos os APs possam actualizar as suas tabelas com as associações entre endereços IP e MAC, e portanto saber onde se encontram os terminais em qualquer altura. É por essa mesma razão que é necessário ter um sistema distribuído de tabelas IP-para-MAC, pois quando um terminal se mover é necessário que o novo AP tenha conhecimento do seu AP antigo, para que possa actualizar as rotas correctamente e de forma transparente. Apesar de este sistema não necessitar de alterações nos terminais, tal como o anterior, não foi pensado para gerir tráfego *multicast*.

Em [15] é introduzido um mecanismo de *handoff* rápido, concebido para suportar aplicações com requisitos de tempo real, sem necessitar de alterações nos terminais, designado SMesh. Isso é conseguido garantindo que cada terminal recebe o mesmo endereço IP de todos os APs (é usada uma função *hash* juntamente com o endereço MAC para o calcular), criando a ilusão que a rede é apenas um AP de grande cobertura. Para monitorizar a mobilidade dos terminais

são usadas as mensagens definidas no protocolo DHCP, ajustando parâmetros como *renewal time* ( $T_1$ ), *rebinding* ( $T_2$ ) e *Lease Time*. Cada terminal está associado a um grupo *multicast* designado *Client Data Group*, onde também se associam os APs ao alcance. A mobilidade de um terminal é gerida pelos APs desse grupo, pois sempre que um AP considere que tem a melhor ligação ao terminal irá enviar uma *gratuitous ARP message* em *unicast* para que este altere o endereço MAC do seu *default gateway*. Desta forma os pacotes destinados a um determinado terminal serão enviados para o seu grupo *multicast*, que está sempre actualizado. Desta forma o terminal recebe pacotes repetidos; caso o protocolo de transporte seja o TCP este resolve, mas caso seja UDP a aplicação terá de resolver esse problema. Este mecanismo apresenta algumas desvantagens tais como a restrição de apenas um canal partilhado por todos os APs em toda a rede emalhada, a introdução de *overhead* adicional devido à gestão dos grupos *multicast* e dos pacotes repetidos, e a não existência de um mecanismo para gestão do tráfego *multicast*.

Em [16] é descrito o *Mesh Mobility Management* ( $M^3$ ), um mecanismo que propõe o uso de hierarquias para gerir a mobilidade dos terminais. Existem três tipos de nós definidos: os *gateways* (GW), que fazem a ligação entre as redes com/sem fios e representam o nível mais alto da hierarquia, os *superior routers* (SR), que guardam a informação de localização dos terminais, e os APs que fornecem a ligação aos terminais. Para fazer a entrega de tráfego do *gateway* até aos terminais são usadas técnicas de encapsulamento para acrescentar um cabeçalho adicional que especifica qual o AP a que se destinam. Estes cabeçalhos são posteriormente removidos no AP correspondente e os pacotes entregues aos terminais. Quando um terminal se move faz o pedido de *handoff* ao novo AP, que irá pedir a informação desse cliente ao AP antigo. Quando o AP antigo recebe a mensagem cria uma rota temporária para comunicar com esse cliente, que terá uma duração  $t$ . Após esse tempo  $t$ , a rota e a informação do cliente são apagados do antigo AP. Uma das desvantagens desta solução é a hierarquização que acrescenta complexidade ao sistema e *overhead* na rede. Esta solução também não apresenta soluções para melhorar o *handoff* a nível 2, e não foi testada usando tráfego *multicast*.

Por fim, é apresentada a proposta [17] que sugere dois mecanismos diferentes de *caching* para criar um mecanismo transparente de *handoff*, sem criar *overhead* adicional. O primeiro é baseado em *caching* ao longo do percurso (*en-route*), ou seja, todos os nós atravessados por um fluxo irão fazer *cache* desse tráfego. O segundo mecanismo assume que os APs estão a funcionar em modo promíscuo e portanto os vizinhos de um determinado AP irão fazer *cache* dos pacotes destinados para este, para que caso um terminal se associe a um vizinho este tenha a informação prontamente disponível. Apesar de estes mecanismos apresentarem baixa complexidade de implementação, não são suficientes para garantir mobilidade total e eficiente dentro de uma rede emalhada, pois no pior caso um terminal pode associar-se a um

AP que não esteja a fazer *cache* do tráfego do terminal, levando a atrasos excessivos na entrega do tráfego.

## 2.7 Conclusão

Em suma, apesar de existirem várias propostas para métricas em encaminhamento todas apresentam limitações. Enquanto umas se tentam focar na simplicidade (ETX, ML) para reduzir *overhead* de controlo, outras como ETT, WCETT, MIC, mETX, ENT, iAWARE, MM, *airtime metric* tornam-se demasiado complexas e geram demasiado tráfego de controlo para implementar em redes emalhadas 802.11. Os protocolos para o encaminhamento *multicast* [6] [7] [8] [9] [10] [11] [12] apresentam limitações nas métricas utilizadas e no suporte à mobilidade de terminais, apresentando também demasiada complexidade para o cenário proposto. Por fim, as soluções de gestão de mobilidade [13] [14] [15] utilizam mecanismos que funcionam ao nível três da camada protocolar OSI, a camada IP, que é transparente para a solução WiFIX. A solução proposta em [16] é demasiado complexa e gera demasiado *overhead* de controlo. Por fim, a solução [17] apresenta requisitos em termos de modo de funcionamento das placas de rede e recursos de *hardware* que introduzem limitações sérias na escalabilidade da rede.





# Capítulo 3

## Especificação da solução WiFIX 3.0

Como foi visto no Capítulo 2, não existe ainda nenhuma solução que permita a difusão eficiente de tráfego *multicast* em redes emalhadas, utilizando métricas apropriadas. Tendo em conta que o conceito das redes emalhadas 802.11 se tem tornado cada vez mais relevante dentro das opções que permitem fazer a extensão de uma rede infra-estruturada, torna-se importante que existam novas soluções adequadas a este contexto.

A solução WiFIX 2.0 apesar de apresentar ganhos significativos quando comparada com a solução 802.11s e contemplar a difusão de tráfego *multicast*, não especifica uma métrica inteligente capaz de reconhecer pontos de congestionamento na rede. Como a solução WiFIX utiliza uma topologia em árvore e uma métrica que se baseia no número de saltos até ao *gateway*, poderão existir cenários de degradação na qualidade das ligações entre MAPs vizinhos, ou até mesmo sobrecarga no MAP. É portanto necessário criar mecanismos adicionais no protocolo WiFIX 2.0 para lidar com estes problemas, e daí nasceu a solução WiFIX 3.0.

A especificação da solução WiFIX 3.0 foi dividida em quatro fases, que representam os mecanismos a criar ou a melhorar no protocolo. Numa primeira fase foi definida uma nova métrica, recorrendo a parâmetros como a carga no CPU dos MAPs, o *throughput* médio recebido do pai e o número de TRs de cada um dos vizinhos, para calcular o custo de uma ligação até ao *gateway*. Numa segunda fase foi criado um mecanismo de histerese para complementar a nova métrica, com o objectivo de limitar a alteração do pai de um MAP apenas a alterações prolongadas na qualidade da ligação e cujo impacto no custo da métrica seja significativo. Na terceira fase considerou-se a definição de um mecanismo de actualização rápida das árvores *multicast* na rede sempre que ocorra a mudança de pai por parte de um MAP. Por fim, na quarta e última fase foi especificado um mecanismo de mobilidade de terminais mais eficiente do que o apresentado na solução WiFIX 2.0. Cada um destes aspectos é detalhado de seguida.

### 3.1 Nova métrica

A métrica utilizada nas versões anteriores do protocolo WiFIX é o *hop-count*. Esta métrica apresenta limitações para uso em redes sem fios pois, por exemplo, não se consegue adaptar à degradação da qualidade das ligações entre MAPs. A nova métrica vem melhorar esse aspecto, utilizando parâmetros como a carga no CPU de cada MAP ( $C_{LOAD}$ ), a probabilidade de recepção de TRs ( $P_{TR}$ ) e o *throughput* médio recebido do pai ( $T$ ) para calcular o custo final da ligação entre cada vizinho e o *gateway*.

Os dois últimos parâmetros são baseados na métrica ETX mas adaptados ao protocolo WiFIX, onde se pretendeu utilizar as mensagens de controlo já existentes na construção da nova métrica para minimizar o *overhead* na rede. A nova métrica dá prioridade a custos mais baixos, e assim sendo os novos parâmetros foram organizados na equação seguinte:

$$\text{Custo total de uma ligação} = \min \left( \text{Custo anunciado pelo vizinho} + \frac{C_{LOAD}}{T \times P_{TR}} \right)$$

Cada um dos parâmetros tem um objectivo específico, sendo que o parâmetro de carga no CPU pretende distribuir a carga oferecida na rede de forma equitativa, evitando que um MAP se torne demasiado sobrecarregado, podendo também evitar equipamentos com problemas de desempenho caso estes impliquem uma maior carga no CPU. A probabilidade de recepção de TRs pretende avaliar a fiabilidade da ligação, contabilizando o número de TRs recebidos entre dois períodos de decisão para a escolha de um novo pai de um MAP e dividindo pelo número esperado de TRs durante esse mesmo período, que é um valor conhecido por todos os MAPs da rede. E como se pode perceber pela equação anteriormente apresentada, um maior número de TRs perdidos reflecte-se num aumento do custo final da ligação. Já o *throughput* contabiliza o tráfego médio recebido do pai durante o mesmo intervalo de tempo utilizado pelo parâmetro  $P_{TR}$  e serve para estimar qual será o impacto no mínimo custo recebido dos vizinhos, para perceber se é mais vantajoso mudar de pai ou manter o actual.

Utilizando estes três parâmetros em conjunto é possível perceber o estado de uma ligação já existente com o pai, assim como avaliar novas ligações possíveis para chegar ao destino pretendido que é o *gateway*, e traduzir essa percepção num valor quantificável que será transmitido nas mensagens TRs enviadas.

## 3.2 Mecanismo de histerese

Devido à introdução da nova métrica o valor do custo de cada ligação pode variar frequentemente (com um período mínimo igual ao período entre tempos de decisão de um novo pai de um MAP). Portanto foi necessário criar um mecanismo de histerese para impedir que alterações mínimas nos custos desencadeassem um processo de escolha de novo pai. O novo mecanismo considera o custo recebido de cada vizinho diferente e compara-o com o custo recebido pelo pai dentro do mesmo período de decisão.

Caso o custo mínimo recebido esteja abaixo ou acima de um certo limiar definível, que neste caso foi colocado a 50% do custo actual recebido do pai, e esse caso aconteça duas vezes consecutivas será desencadeado um processo de escolha para o novo pai.

Enquanto a primeira condição tenta evitar a escolha de um novo pai devido a pequenas flutuações, a segunda pretende evitar a escolha devido a casos em que o valor do custo ultrapassa o limiar definido devido a interferências externas graves mas transitórias ou devido à recepção de uma grande quantidade de dados vindos do MAP pai.

## 3.3 Soft Topology Change

Para fazer o encaminhamento de tráfego *multicast* utilizando túneis *unicast* é necessário manter uma tabela em cada MAP que faz a correspondência entre cada grupo *multicast* recebido e os vizinhos para onde deve ser enviado esse fluxo. Porque as árvores *multicast* são formadas a partir da topologia da rede emalhada, é necessário actualizar estas árvores sempre que um MAP escolhe um novo pai.

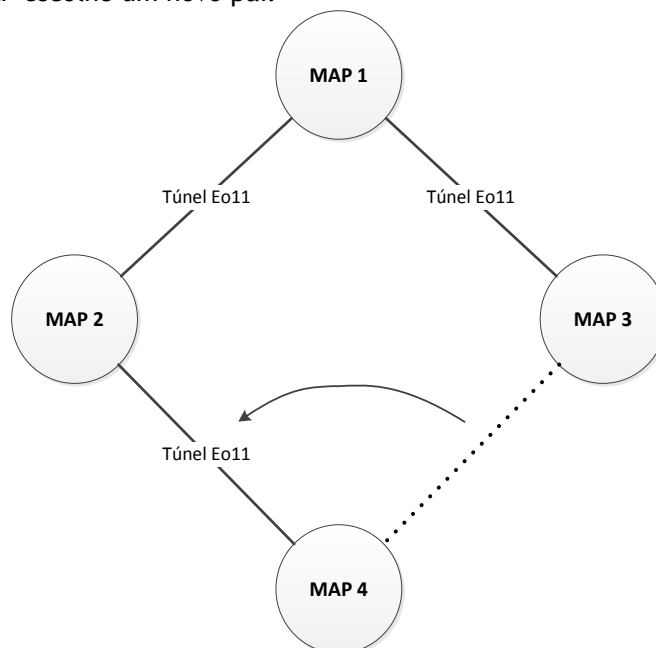


Figura 3.1 Exemplo da mobilidade dos MAPs

Suponhamos que no exemplo da Figura 3.1 o MAP 4 está a receber tráfego de um grupo *multicast*. Nesse caso existe uma entrada na tabela do MAP 1 e do MAP 3 para encaminhar o tráfego *multicast* até ao MAP 4. Quando este escolhe um novo pai, a topologia da rede altera-se e portanto a árvore *multicast* formada está desactualizada. Nesta solução cabe ao MAP 4, através de novo mecanismo, actualizar as tabelas *multicast* que contêm este grupo, de forma a reflectir a nova topologia da rede e por consequência da árvore *multicast*. Este novo mecanismo deve ser rápido, minimizando o número de pacotes perdidos durante a actualização e o impacto no tráfego de controlo deve ser mínimo. Foi portanto escolhido um mecanismo que envolve a criação e envio de duas mensagens IGMP pelo MAP que irá mudar de pai, sendo uma das mensagens recebida pelo pai antigo e a outra pelo novo pai. Ambas as mensagens são encaminhadas pelo percurso ascendente até ao *gateway* (onde o processo de encaminhamento termina) e, enquanto a mensagem recebida pelo antigo pai trata da dissociação dos grupos *multicast*, a mensagem recebida pelo novo pai trata por sua vez, da associação dos grupos.

### 3.4 Mecanismo de mobilidade dos terminais

Apesar de a solução WiFIX 2.0 especificar um mecanismo de mobilidade de terminais, com recurso à técnica de DHCP *Snooping*, os resultados obtidos para esse método não são satisfatórios nalguns casos, nomeadamente para o uso de aplicações com requisitos de tempo real. Desta forma, um dos objectivos deste trabalho é criar um novo mecanismo que permita o redireccionamento mais rápido dos fluxos *multicast* quando um terminal se move dentro da rede emalhada.

Foi elaborado um mecanismo que recorre a um programa instalado no terminal de cada utilizador, que monitorizando o SNR de cada MAP que se encontrem ao alcance, tem por objectivo tomar a decisão de *handoff* e informar a rede dessa transição, enviando uma mensagem de formato conhecido ao MAP a que está associado correntemente, onde inclui o MAC do novo MAP pretendido, e quebrando a ligação com esse MAP. Este irá então tratar da mobilidade do terminal na rede, actualizando as árvores *multicast* dos grupos aos quais o terminal pertença. Isso permite reduzir o tempo de reacquirição de fluxos no terminal, visto que enquanto esta actualização dentro da rede se processa, o terminal está a proceder à autenticação no novo MAP escolhido.

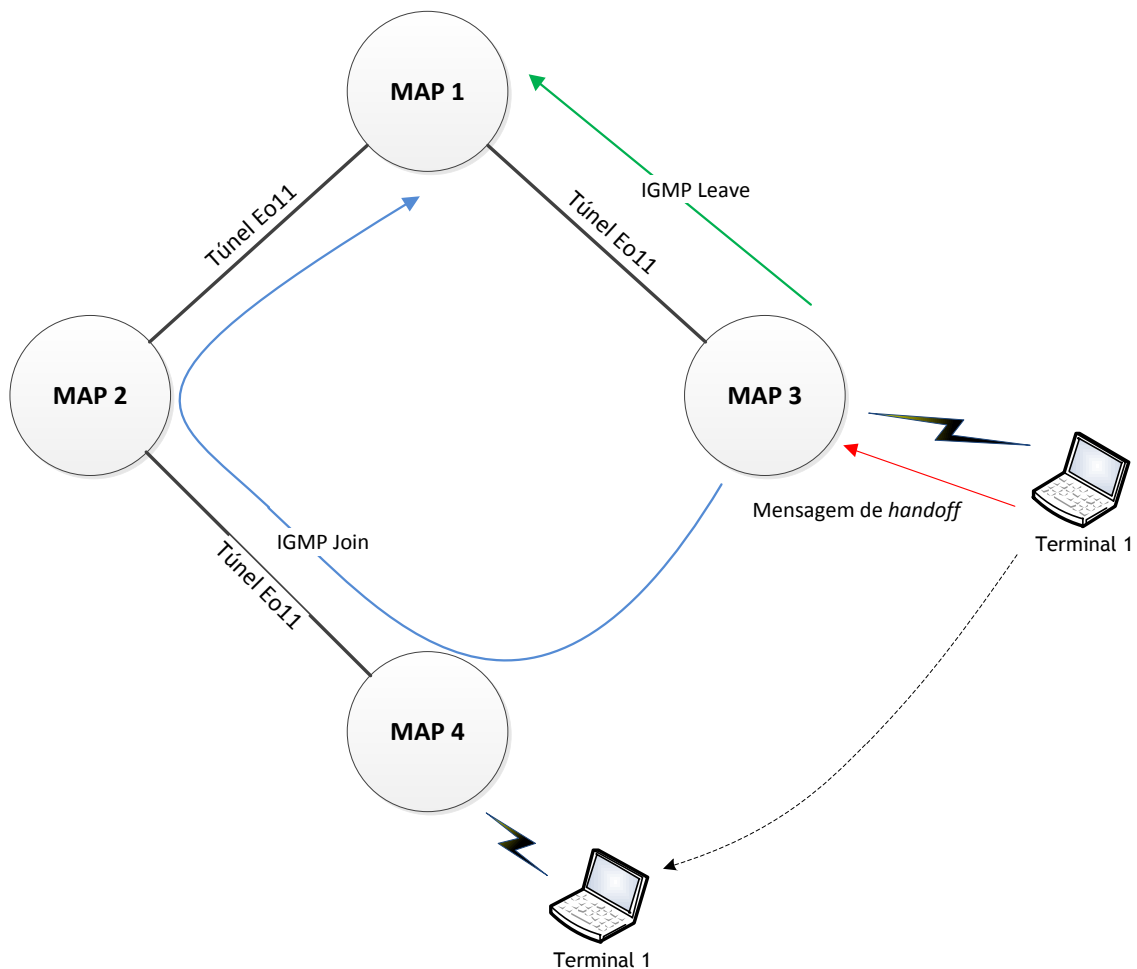


Figura 3.2 Exemplo da mobilidade de um terminal dentro da rede em malha

Na Figura 3.2 é exemplificado um caso mais simples do mecanismo em funcionamento. Supondo que o terminal 1 está a receber tráfego de um grupo *multicast* e pretende associar-se ao MAP 4 devido a um melhor SNR, é enviada uma mensagem de *handoff* para o MAP 3, que inclui o MAC do MAP 4. O MAP 3 responsabiliza-se por actualizar a árvore *multicast* na rede, utilizando duas mensagens IGMP. Uma das mensagens irá percorrer o caminho ascendente entre o MAP 3 até ao *gateway* desassociando o MAP 3 do grupo *multicast*, na tabela do MAP 1. A outra mensagem associa o MAP 4 e o MAP 2 ao grupo *multicast* pelo que o terminal 1 assim que terminar o processo de autenticação com o MAP 4 pode receber de imediato o tráfego *multicast*.



# Capítulo 4

## Implementação da solução WiFIX 3.0

Neste capítulo é descrita a implementação da solução WiFIX 3.0. A solução foi implementada em ambiente Linux, mais propriamente a distribuição OpenWRT compatível com sistemas embebidos.

Para a criação dos túneis virtuais Eo11 entre MAPs foram utilizados *drivers* que permitem a criação de *bridges* IEEE 802.11, nomeadamente a solução TUN/TAP, que se encontra integrada no *kernel* Linux. Esta solução cria interfaces virtuais que se designam por *taps*, que são vistos pelos MAPs como interfaces ponto-a-ponto.

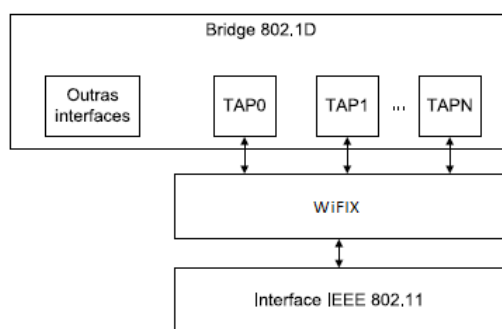


Figura 4.1 Modelo das camadas protocolares para solução WiFIX

O protocolo WiFIX está situado entre a camada física e a camada IP, e consiste em processar tramas recebidas na interface IEEE 802.11 e encaminhá-las para as *taps* correspondentes, assim como encaminhar as tramas recebidas nas *taps* para a interface 802.11, como mostra a Figura 4.1. A partir do momento em que uma trama é escrita numa *tap*, esta é tratada pelos *drivers* TUN/TAP, pelo que fica à responsabilidade do *kernel* do Linux processar e reencaminhar a trama para uma ou mais *taps* que não a original.

No entanto existem algumas mensagens que são tratadas ao nível do WiFIX, não sendo encaminhadas para as *taps*, pelo facto de serem mensagens de controlo introduzidas e descritas pelo protocolo WiFIX.

## 4.1 Nova métrica

Como descrito no Capítulo 3 a nova métrica assenta em três parâmetros: 1) a carga no CPU de cada MAP, 2) o *throughput* recebido do pai e 3) percentagem de mensagens TR recebidas de cada vizinho, sendo que os últimos dois parâmetros são calculados em cada intervalo de tempo entre dois períodos de escolha de um novo pai.

Para tal foram criadas duas listas, uma para o segundo parâmetro e a outra para o terceiro. A primeira lista guarda em cada entrada o endereço MAC, o tamanho (em bytes) e o *timestamp* de cada mensagem recebida. A segunda lista guarda por cada entrada o endereço MAC, e o *timestamp* de cada mensagem TR que receba e que cujo número de sequência seja diferente do anterior recebido desse vizinho.

Para obter a carga no CPU é lido o ficheiro que se encontra em “/proc/loadavg”, que está presente em todas as distribuições Linux, e como este ficheiro apresenta três valores, com as médias de cinco, dez e quinze minutos, é utilizado o primeiro valor, a média dos cinco minutos. Os valores estão numa escala de zero a infinito, onde o valor de 1.00 representa um CPU completamente carregado para um sistema com apenas um *core*, e cujos valores são arredondados às centésimas. Estes valores incluem as tarefas em execução e as em espera e portanto o valor 1.00 pode ser excedido (no caso de um sistema com um *core*), caso o CPU esteja completamente carregado e continue a receber pedidos. Apesar de esta não ser a maneira correcta de analisar apenas a carga activa no CPU, apresenta vantagens quanto à rapidez de processamento e indica com bastante precisão a carga no CPU para os equipamentos utilizados no *test-bed*.

$$\text{Custo total de uma ligação} = \min \left( \text{Custo anunciado pelo vizinho} + \frac{C_{LOAD}}{T \times P_{TR}} \right)$$

Na equação é apresentada a fórmula utilizada para calcular o custo final de cada ligação, mas durante a implementação foi necessário proceder a ajuste nos pesos de cada parâmetro para o custo final, pois o valor do custo dentro da mensagem TR só pode ocupar 3 bytes, o que impõe um valor máximo igual a 16777215.

Intervalo de valores	Parâmetro
0.0 - 10.0	Peso da carga no CPU na métrica
0.0 - 1.0	Peso da probabilidade de receber TRs
0.0 - ∞	Peso do <i>throughput</i> (valor em KB/s)

Tabela 4.1 Peso de cada parâmetro na métrica



Esses pesos estão descritos na Tabela 4.1, e como se pode verificar o parâmetro da carga no CPU foi multiplicado por um factor de 10, enquanto a escala do *throughput* foi convertida para KBytes por segundo.

## 4.2 Mecanismo de histerese

O objectivo do mecanismo de histerese é impedir que variações pequenas e frequentes ou rápidas e grandes no custo tornem a rede instável ao desencadear mudança de pai nos MAPs.

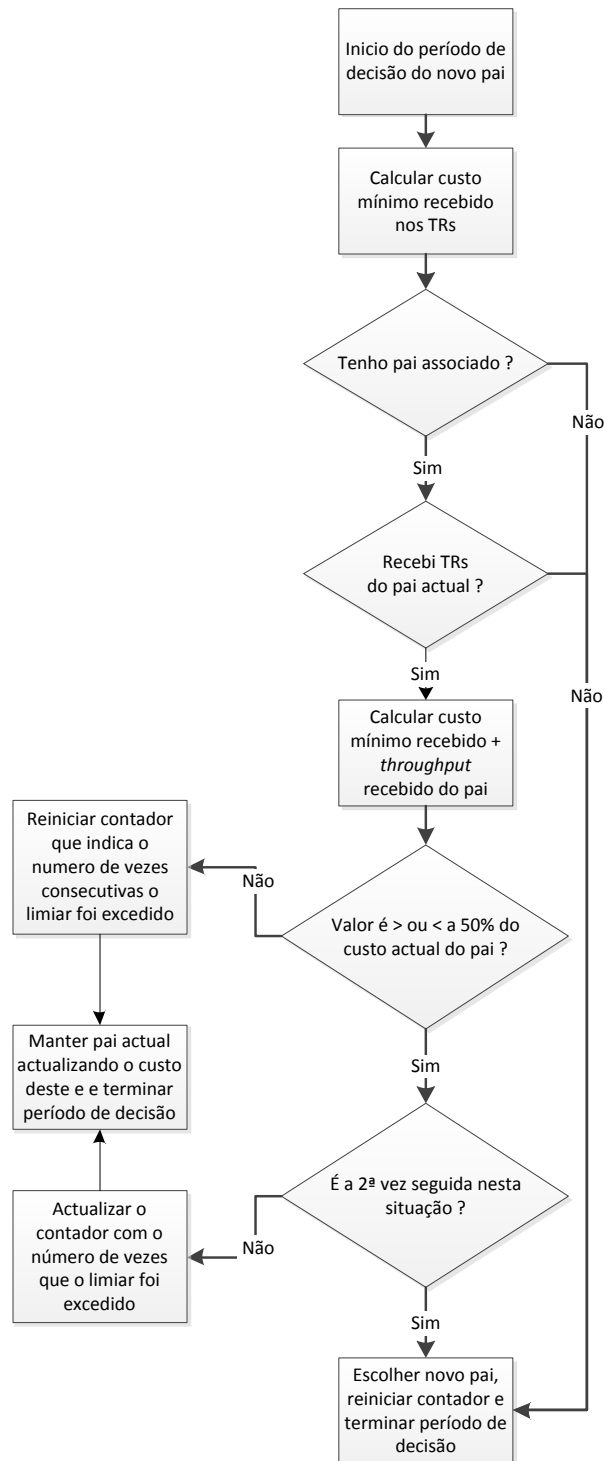


Figura 4.2 Fluxograma do mecanismo de histerese

Para evitar esse cenário, foi introduzido este mecanismo durante a escolha do novo pai, e este começa a começar a actuar assim que calcula o custo mínimo oferecido ao MAP, recebido pelos TRs, e adicionando o parâmetro do *throughput* do pai actual, caso o haja. Obtido esse custo mínimo é feita a comparação com o custo recebido pelo pai actual, e será feita a comparação entre os dois valores. Caso o custo mínimo seja superior a um limiar definível, que no caso do cenário de teste foi 50% do custo actual do pai, é feito o segundo teste, perceber se foi a segunda vez que o limiar foi ultrapassado. Um resultado negativo no primeiro teste termina o processo de escolha de pai, reiniciando a zero o contador que indica o número de vezes que o limiar foi ultrapassado consecutivamente e actualizando-se o custo do pai actual. No caso de resultado negativo no segundo teste o contador será incrementado, indicando que o limiar foi excedido, mas como ainda não corresponde ao segundo caso consecutivo o processo terminará apenas com a actualização do custo do pai actual. Caso o resultado seja positivo em ambos os testes, o processo de escolha de novo pai irá continuar sendo o contador reiniciado a zero. Por fim, se o MAP não tiver um pai associado, o mecanismo de histerese será ignorado procedendo-se imediatamente à escolha do novo pai, como mostra a Figura 4.2.

### **4.3 Soft Topology Change**

O mecanismo de suporte à mobilidade dos MAPs, ou *Soft Topology Change*, surge para solucionar o problema da actualização das árvores *multicast* sempre que haja uma mudança de pai por parte de um MAP.

A solução inicial proposta foi a de uma única mensagem actualizar todo o percurso entre o antigo pai e o novo pai no MAP, passando obrigatoriamente pelo *gateway*. No entanto essa solução implicaria criar uma mensagem com formato único, e podia levar algum tempo a propagar-se da origem até ao destino. Para tornar o processo mais rápido, optou-se então por criar duas mensagens IGMP padrão, enviadas em *unicast* pelo MAP filho para o novo e para o antigo pai.

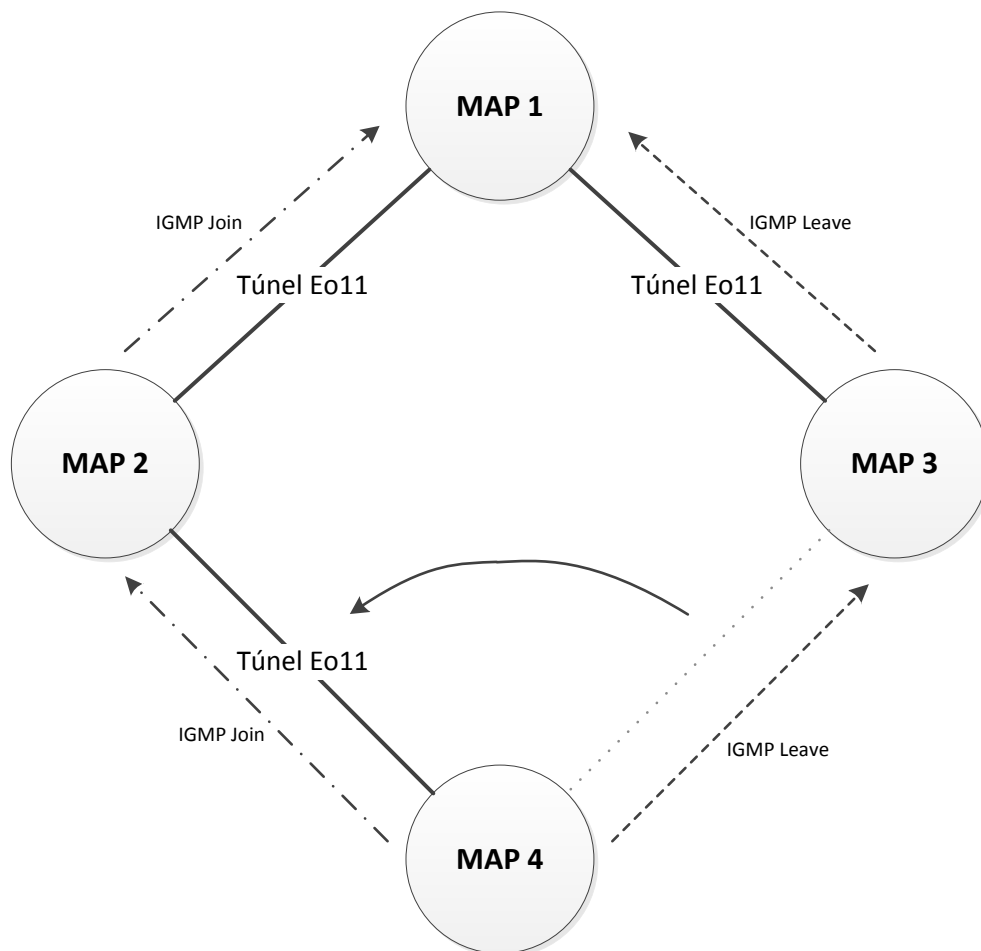


Figura 4.3 Mecanismo de mobilidade dos MAPs em acção

Ambas as mensagens contêm os grupos *multicast* a que o MAP está actualmente associado, mas uma das mensagens indica uma operação de associação aos grupos, que é enviada ao novo pai, enquanto a outra indica uma operação de dissociação, que é enviado ao antigo pai, como se pode ver na Figura 4.3.

Estas mensagens são processadas ao nível do WiFIX, sendo ambas encaminhadas até ao destino final, o *gateway*. Esta abordagem permite reduzir o tempo de propagação da mensagem de actualização, pois existem duas mensagens que em paralelo são encaminhadas até ao *gateway*, sendo esse percurso menor do que o percurso que uma única mensagem teria de percorrer. Este método apresenta como desvantagem o aumento do *overhead* de controlo na rede, pois são utilizadas duas mensagens em vez de uma, embora o mecanismo de histerese amenize este efeito, tornando as mudanças de pai pouco frequentes.

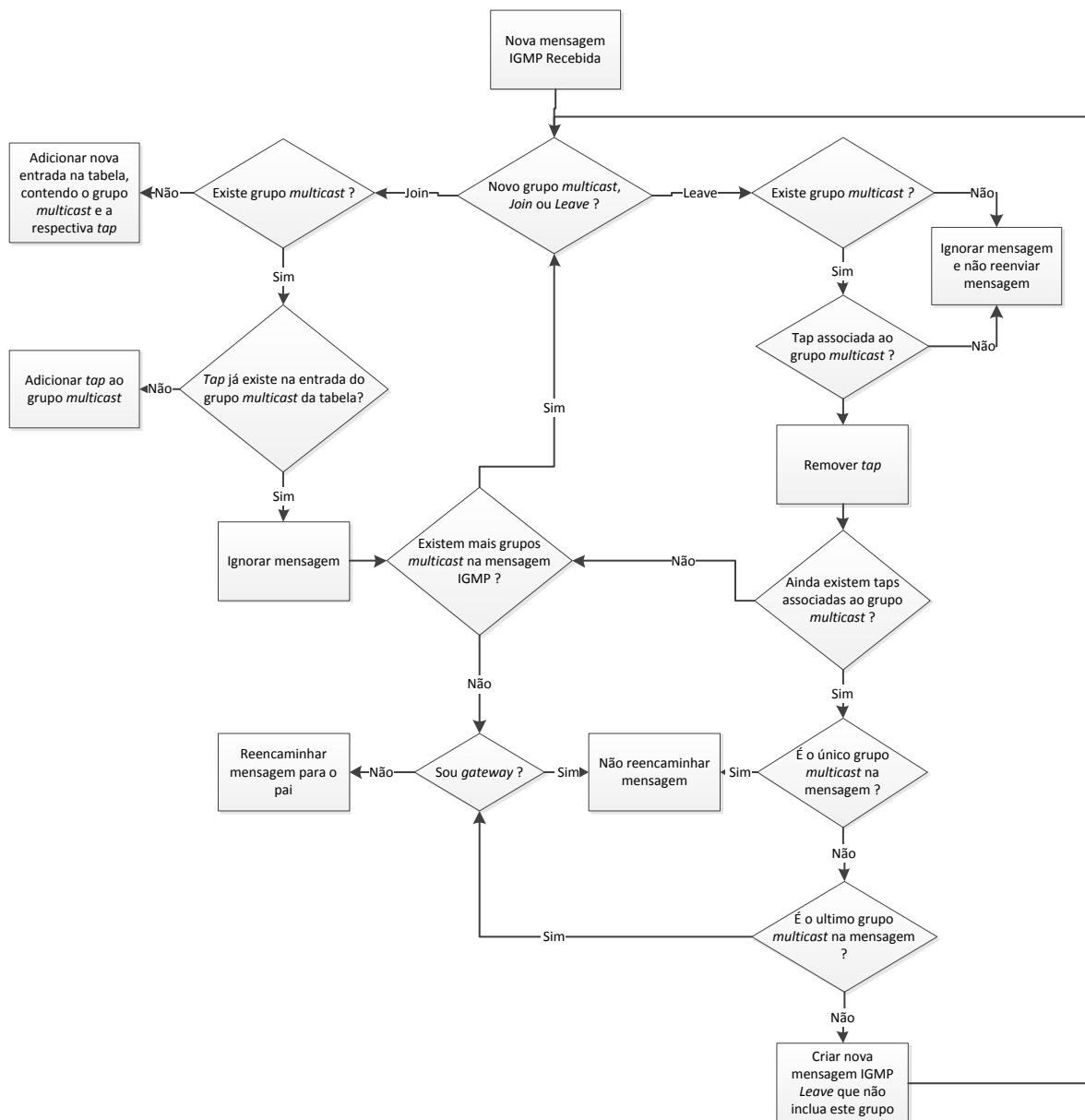


Figura 4.4 Fluxograma da função de adição e remoção de grupos *multicast*

Para evitar que a mensagem de dissociação remova grupos *multicast* necessários nos MAPs, foram criadas condições que terão de ser necessariamente obedecidas, e que são apresentadas na Figura 4.4, através de um fluxograma.

Como se pode observar, caso todos os grupos *multicast* mencionados na mensagem sejam necessários no MAP que a está a processar, esta mensagem não será encaminhada, reduzindo assim o *overhead* de controlo.

# Capítulo 5

## Avaliação da Solução WiFIX 3.0

Com o intuito de provar que as alterações propostas trazem melhorias significativas relativamente à solução WiFIX 2.0, foram realizados testes comparativos utilizando a implementação descrita no Capítulo 4 e um *test-bed* com 4 *routers* sem fios (*RouterStation Pro*), com o sistema operativo Linux, utilizando a distribuição OpenWRT. Os equipamentos foram colocados dentro de uma sala do edifício do INESC Porto, estando portanto todos ao alcance rádio entre si. Para tentar minimizar interferências, os equipamentos foram todos configurados para o canal 3 (2.422 GHz) e forçados a um débito máximo de 11 Mbit/s.

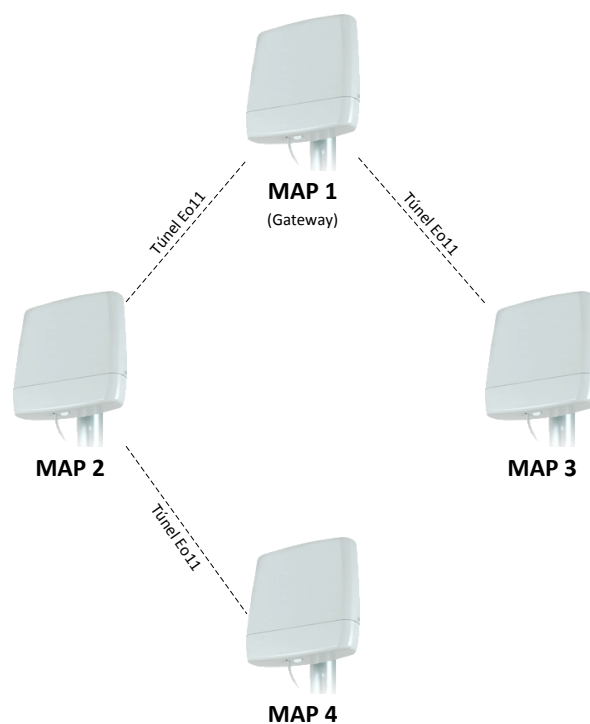


Figura 5.1 - Topologia da solução WiFIX 2.0 utilizada no *test-bed*

Nos testes foi comparado o desempenho dos protocolos WiFIX 2.0 e WiFIX 3.0. Como a solução WiFIX 2.0 cria topologias estáticas baseadas na métrica *hop-count*, foi forçada a topologia apresentada na Figura 5.1. Esta imposição da topologia não afecta os resultados finais, pois a topologia escolhida é uma das soluções possíveis, mas simplificou os testes garantindo que a topologia escolhida foi consistente ao longo dos testes. Para o WiFIX 3.0 foi apenas imposta uma topologia que não permitisse ao MAP 4 ligar-se directamente ao MAP 1, podendo portanto este escolher como pai o MAP 2 ou o MAP 3. Em ambos os casos foi utilizada uma técnica de filtragem de TRs, suportada desde da primeira versão do código WiFIX, que consiste em ignorar TRs cujo MAC esteja presente numa tabela com MACs bloqueados. Esta tabela é editável pelo utilizador a qualquer momento, enquanto o programa WiFIX esteja a correr.

Os testes foram realizados em três cenários distintos: no primeiro o MAP 2 foi carregado ao máximo, de modo a que o CPU estivesse em uso durante todo o tempo do teste. O segundo cenário envolve a simulação de uma ligação de má qualidade entre o MAP 2 e o MAP 4. No terceiro cenário foi utilizada apenas a solução WiFIX 3.0 em condições normais e foi forçada a transição de pai do MAP 4, enquanto este recebia fluxos *unicast* (e posteriormente *multicast*) para obter o tempo médio de reacquirição dos fluxos.

Para a obtenção dos resultados práticos foram utilizados os programas IPerf<sup>1</sup> e *ping*. O IPerf pode gerar tráfego TCP, UDP (*unicast* e *multicast*) sendo portanto a principal fonte de resultados. O *ping* foi utilizado para a medição da latência ponto-a-ponto. Para os testes em que é usado tráfego *multicast* o MAP 1 (*master*) foi a fonte do tráfego uma vez que se pretendeu simular a chegada de tráfego proveniente da rede infra-estruturada para nós na rede emalhada. Todos os valores apresentados neste capítulo foram obtidos utilizando a média de dez resultados experimentais, onde cada um destes foi obtido ao fim de sessenta segundos de simulação.

Este capítulo está dividido em três secções. Na Secção 5.1 são apresentados os resultados obtidos para o cenário em que o MAP 2 está sobrecarregado. Na Secção 5.2 são referidos os resultados no cenário de uma ligação com perdas entre o MAP 2 e o MAP 4. Por fim, na Secção 5.3 são apresentados os tempos de reacquirição de fluxos *multicast* quando um MAP muda de pai.

---

<sup>1</sup> Página Oficial - <http://iperf.sourceforge.net/>

## 5.1 Comparação entre WiFIX 2.0 e WiFIX 3.0 (CPU Sobrecarregado)

Nesta secção é apresentado o resultado da comparação entre o WiFIX 2.0 e WiFIX 3.0 para o cenário em que o MAP 2 está sobrecarregado. Neste cenário foi utilizado um programa em C, que faz apenas uma multiplicação simples num ciclo infinito, para sobrecarregar o CPU. Foi também utilizado o comando *nice* nativo ao sistema operativo Linux para colocar este programa como prioridade máxima, utilizando o valor da prioridade igual a 15.

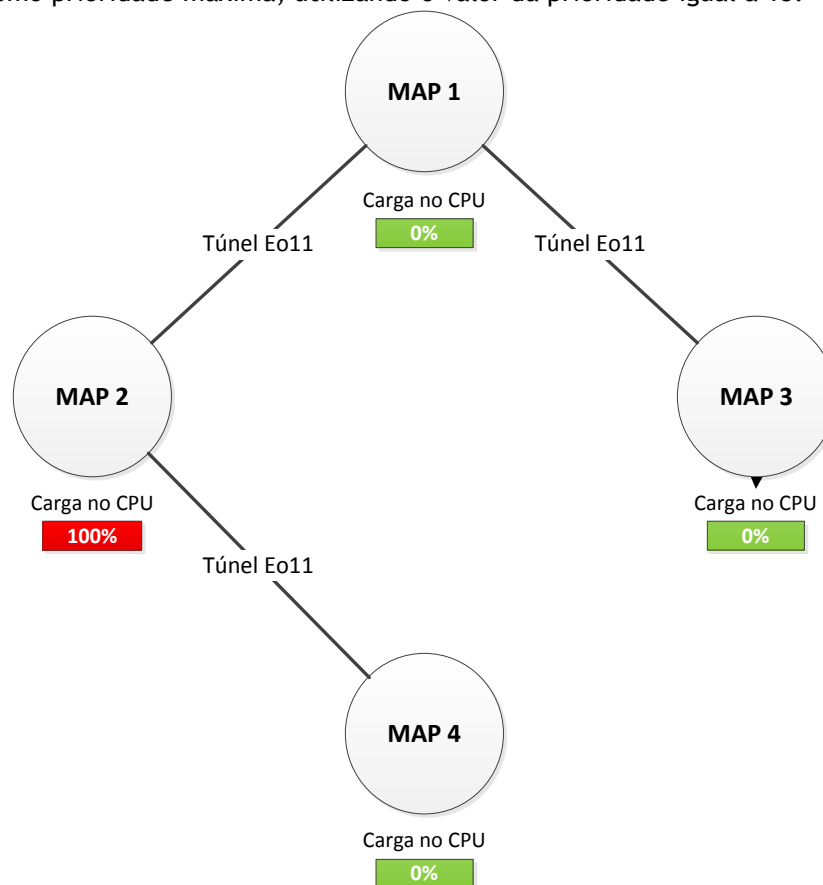


Figura 5.2 Estado inicial da rede utilizando a solução WiFIX 2.0

Na Figura 5.2 é exemplificada a configuração estática definida para a solução WiFIX 2.0. No caso da solução WiFIX 3.0 o pai do MAP 4 será o MAP 3, devido à nova métrica implementada. Foram realizados testes com três protocolos diferentes: TCP, UDP e ICMP. O protocolo TCP foi usado para descobrir o *throughput* máximo entre o *gateway* e o MAP 4. O protocolo UDP também foi utilizado em testes ponto-a-ponto, assim como ponto-a-multiponto (*multicast*), sendo utilizado para calcular parâmetros como o *throughput*, *jitter* e perda de pacotes. Por fim, o protocolo ICMP foi utilizado para determinar a latência a um salto (*gateway* e o pai do MAP 4) e a dois saltos (*gateway* e o MAP 4).

### 5.1.1 Throughput

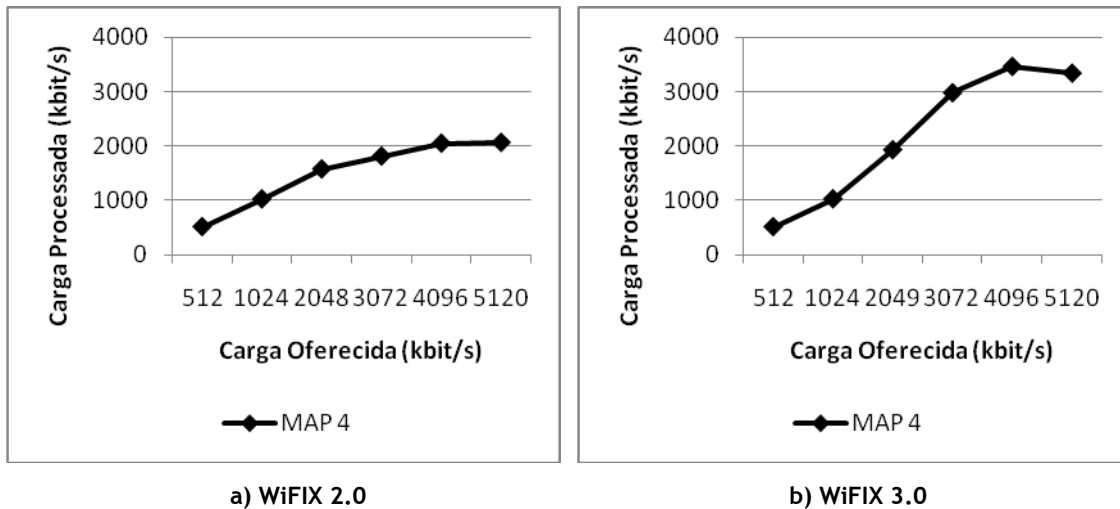
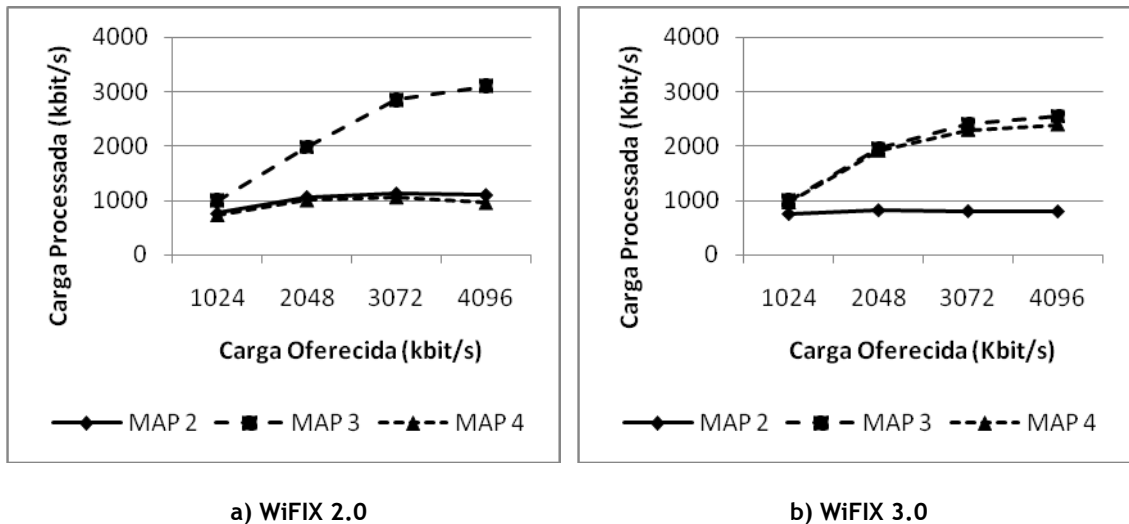


Figura 5.3 Unicast Throughput

Na Figura 5.3 é possível observar-se que o tráfego do WiFIX 3.0 satura aproximadamente aos 3,5 Mbit/s, enquanto o tráfego do WiFIX 2.0 atinge o seu máximo aproximadamente aos 2 Mbit/s. Na verdade, nas soluções de estado da arte os fluxos *multicast* estariam limitados ao *bit rate* mais baixo da norma utilizada (neste caso 1 Mbit/s). Contudo, a solução WiFIX utiliza túneis *unicast* para a difusão do tráfego *multicast* e portanto esse valor é significativamente superior. Já a diferença de débito entre as versões da solução devem-se ao facto de no WiFIX 2.0 o MAP 4 ter como pai o MAP 2, ou seja, o nó mais carregado da rede. Este nó, devido à carga excessiva a que está sujeito, perdeu um grande número de pacotes, o que afectou negativamente o MAP 4. Já na versão 3.0 do protocolo, esse problema foi evitado pois a nova métrica implementada reflecte no custo parâmetros que afectam a qualidade das ligações. Dessa forma, o MAP 4 escolhe o MAP 3 como seu pai, e portanto evita a ligação de má qualidade que passa pelo MAP 2. Os valores obtidos estão dentro dos valores esperados, pois o *throughput* máximo a 1 salto com tráfego UDP é de aproximadamente 7 Mbit/s. A diferença de 3,5 Mbit/s entre os testes a um salto e a dois saltos é explicada pelo facto de todos os MAPs estarem ao alcance rádio entre si. Isso implica que tanto o MAP 1 como o MAP 2 ou MAP 3 têm de partilhar o meio para transmitirem e retransmitirem, respectivamente, os pacotes UDP, o que implica um corte próximo de 50% da largura de banda disponível. Utilizando o protocolo TCP foi obtido um *throughput* de 1596 kbit/s e de 1901 kbit/s para a solução WiFIX 2.0 e WiFIX 3.0, respectivamente. Este desempenho inferior em relação ao protocolo UDP é esperado, pois em TCP cada pacote IP tem de ser confirmado antes de ser transmitido o seguinte. Apesar de este mecanismo garantir fiabilidade na transmissão também diminui o *throughput*, como se pode confirmar pelos resultados obtidos.





a) WiFIX 2.0

b) WiFIX 3.0

Figura 5.4 Broadcast Throughput

As limitações da solução WiFIX 2.0 são ainda mais visíveis quando se passa para um cenário de *broadcast*. Este cenário foi recriado utilizando tráfego *multicast*, que é recebido por todos os MAPs, excepto o MAP 1, que foi a origem dos dados. Na Figura 5.4 o *throughput* máximo obtido na solução WiFIX 2.0 e é aproximadamente 3 Mbit/s. No entanto este valor superior deve-se ao facto de nesta solução não existir tanta concorrência no meio, pois neste cenário ambos MAP 1 e MAP 2 encaminham tráfego *multicast*, mas o CPU do MAP 2 está sobrecarregado, pelo que o número de pacotes que envia para o meio será menor. Isto permite ao MAP 1 enviar os seus pacotes a um débito superior, razão pela qual o MAP 3 apresenta o maior *throughput*. Mas tal como no caso de *unicast* é possível notar uma melhoria significativa no *throughput* do MAP 4, que neste caso passa de 1 Mbit/s para 2,4 Mbit/s, ou seja, uma melhoria de aproximadamente 140%. Para explicar a diferença de valores no *throughput* máximo nos casos de UDP *unicast* e *multicast* é necessário entender um mecanismo presente na norma IEEE 802.11, utilizado no envio de tramas. Quando se envia uma mensagem em *unicast* de nível 2 (camada MAC), essa mensagem terá de ser confirmada com uma trama ACK (*acknowledgement frame*) pelo receptor. No caso do *multicast*, todos os pacotes são encapsulados em tramas *unicast*, o que significa que todos os MAPs confirmam as tramas que recebem correctamente. Essas tramas ACK, juntamente com as retransmissões feitas pelo MAP intermédio (MAP 2 para o WiFIX 2.0, MAP 3 para o WiFIX 3.0), no caminho entre o MAP 1 e o MAP 4, forçam a que haja uma concorrência muito elevada pelo meio. Isso traduz-se num menor *throughput* em todos os nós quando é usado *broadcast*, em comparação com *unicast*.

### 5.1.2 Jitter

O *jitter* é um parâmetro com especial importância para aplicações com requisitos de tempo real, tais como VoIP, que frequentemente utilizam tráfego *multicast*. Por essa razão este parâmetro torna-se importante para analisar o desempenho da rede.

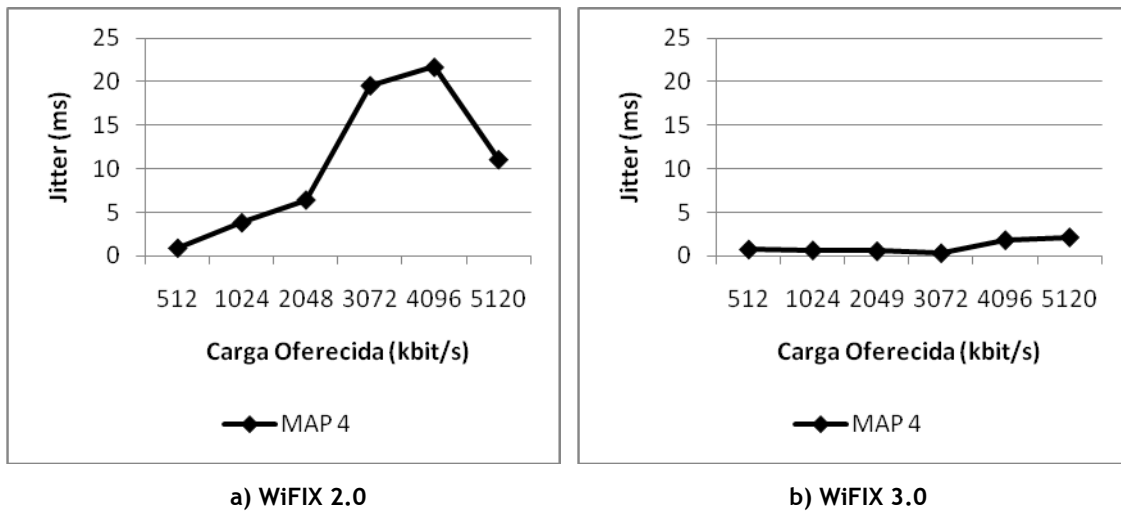
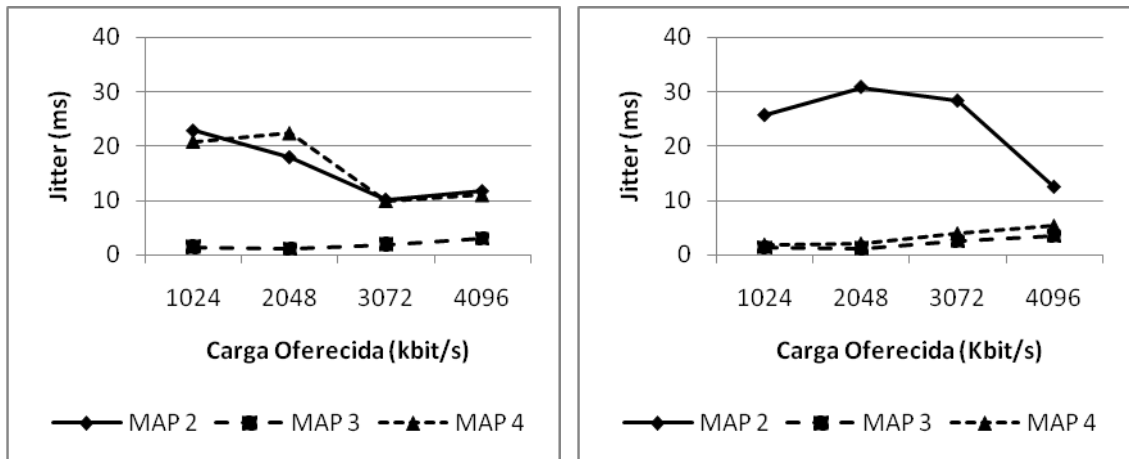


Figura 5.5 Unicast Jitter

Como se pode verificar na Figura 5.5, existe uma diferença significativa entre as duas versões do protocolo. Pode-se concluir que a carga elevada no CPU dos MAPs afecta não só o *throughput*, mas também o *jitter*. No WiFIX 2.0 o *jitter* chega a atingir valores próximos dos 22 ms para a carga oferecida de 4096 kbit/s, enquanto no WiFIX 3.0 esse valor não excede os 2,5 ms. Estes resultados são esperados sendo a explicação simples: com o aumento da carga no CPU, os pacotes enviados pelo MAP 1 e recebidos pelo MAP 2 ficam em fila de espera durante um maior período de tempo, o que influencia directamente o aumento do *jitter* observável no MAP 4. A razão para a quebra no valor do *jitter*, para uma carga oferecida de 5120 kbit/s nos resultados da solução WiFIX 2.0, deve-se ao facto de existirem muitos pacotes descartados na fila de espera do MAP 1 (a origem do tráfego *multicast*) e na fila do MAP 2 devido à elevada concorrência ao meio. Desta forma não é possível ao MAPs encaminharem todos os pacotes, e como o IPerf se baseia nos pacotes recebidos no receptor para calcular o *jitter*, isto leva a uma diminuição do valor calculado; uma vez que no MAP 4 é recebida uma menor quantidade de tráfego *multicast*, os pacotes de facto recebidos no destino não sofrem um atraso tão elevado como no caso de cargas oferecidas de 3072 e 4096 kbit/s.



a) WiFIX 2.0

b) WiFIX 3.0

Figura 5.6 Broadcast Jitter

Passando para o cenário de *broadcast*, é possível verificar se os tempos se agravam. Na Figura 5.6 pode-se verificar que o *jitter* do MAP 2 se agrava, atingindo quase 25 ms. Este aumento relaciona-se com a maior concorrência no meio, explicada no ponto 5.1.1, pois para além do atraso que os pacotes sofrem devido à carga excessiva, é também necessário somar o tempo que o MAP demora a escutar o meio para determinar se está livre. Esse tempo é aleatório e calculado pelo mecanismo DCF (*Distributed Coordination Function*), e que é representado pela sigla DIFS (*DCF Interframe Space*). Portanto, quanto maior for a concorrência no meio, maior será o *jitter*. Ainda assim, conclui-se que a maior contribuição para os resultados obtidos para a solução WiFIX 2.0 se deve à carga excessiva no CPU do MAP 2, uma situação evitada na solução WiFIX 3.0 que utilizando a nova métrica reconhece e evita o nó sobrecarregado, melhorando significativamente os resultados no MAP 4.

### 5.1.3 Packet Loss Ratio

O *packet loss ratio* é um parâmetro importante para perceber como reage a rede ao tráfego oferecido. Este parâmetro é muito importante, tanto para o protocolo UDP como para o TCP, por diferentes motivos. No caso do protocolo TCP, um maior *packet loss ratio* implica um maior número de retransmissões, diminuindo o *throughput*. Como no caso do UDP um pacote perdido não é retransmitido, o aumento do *packet loss ratio* pode afectar negativamente as aplicações que dependem deste protocolo.

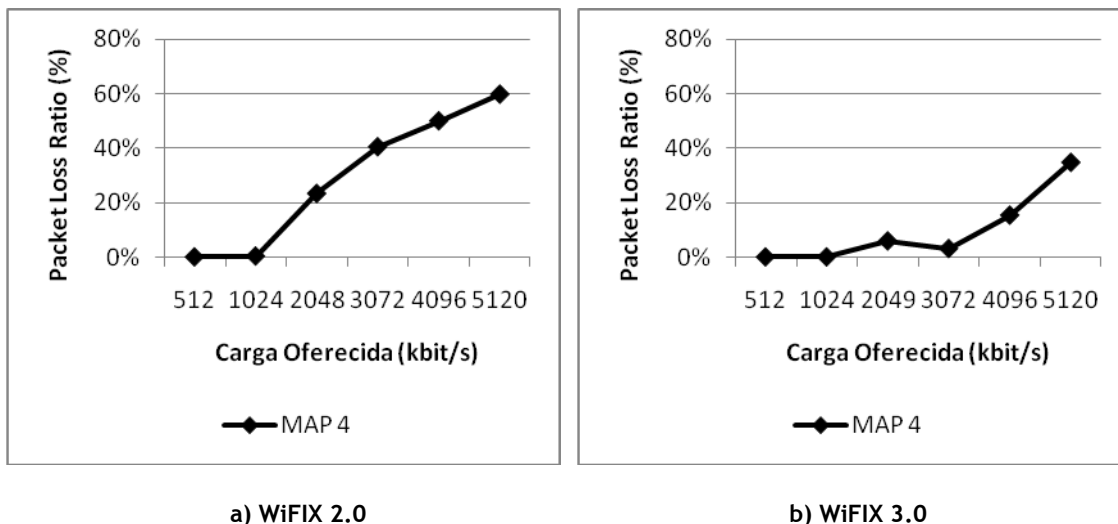


Figura 5.7 Unicast Packet Loss Ratio

Como se pode observar na Figura 5.7, existe uma grande perda de pacotes para débitos iguais ou superiores a 2 Mbit/s no caso do WiFIX 2.0. Já para o WiFIX 3.0 só a partir de 3 Mbit/s começam a existir perdas superiores a 10% que, ainda assim, são mais baixas que as da solução WiFIX 2.0. Este teste reafirma o facto de uma carga elevada no CPU levar a uma perda significativa de pacotes. Estes valores validam os resultados obtidos para o *throughput* máximo de cada uma das soluções (ver Secção 5.1.1).

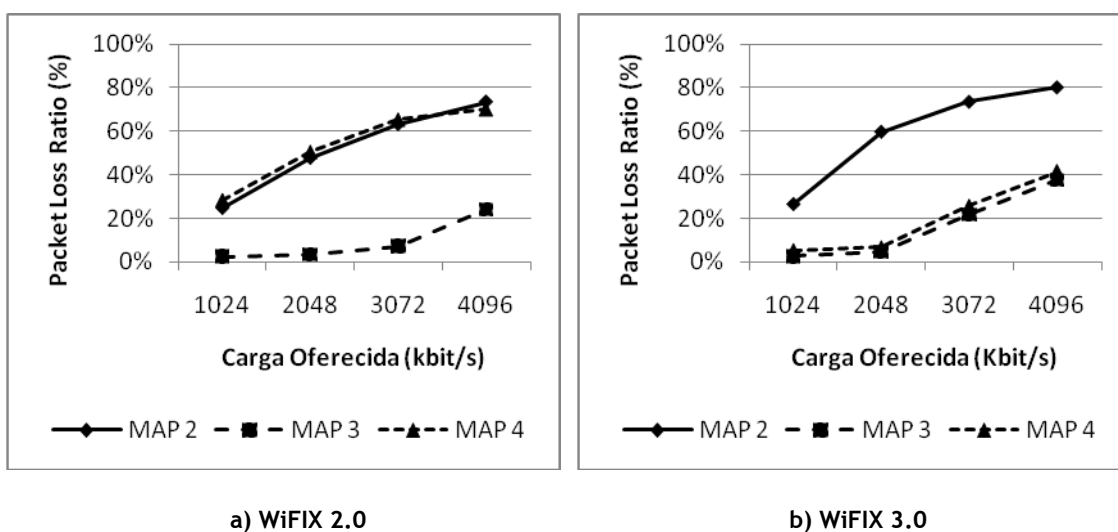


Figura 5.8 Broadcast Packet Loss Ratio

Em *broadcast* a Figura 5.8 mostra claramente as vantagens da nova métrica, pois existe uma diferença significativa entre as perdas verificadas no MAP 4 entre ambas as soluções, nomeadamente de perdas entre 20% até 70% na solução WiFIX 2.0 foi possível passar para perdas entre 0% e 40% na solução WiFIX 3.0.

### 5.1.4 Latência

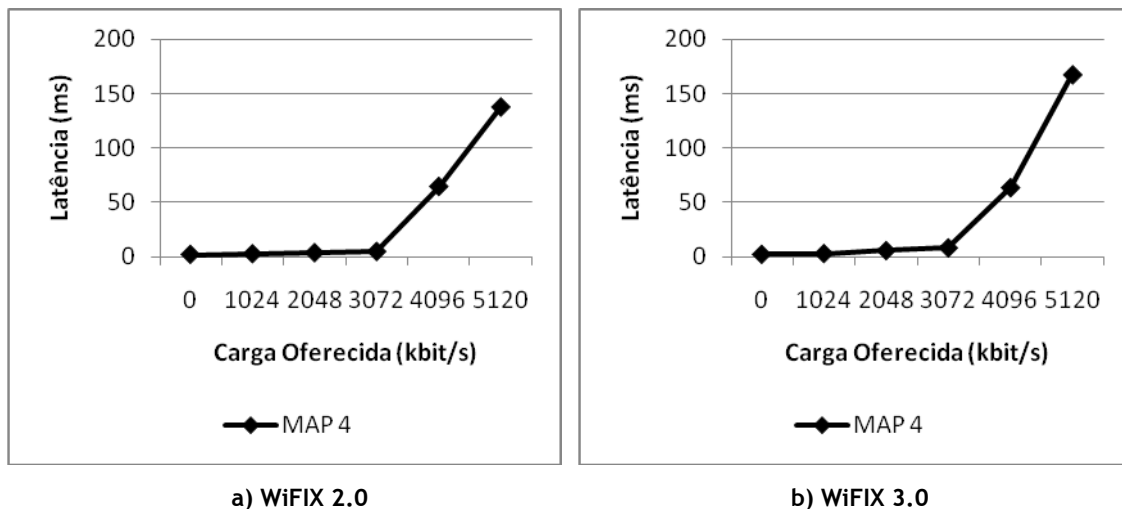


Figura 5.9 Latência a 2 saltos

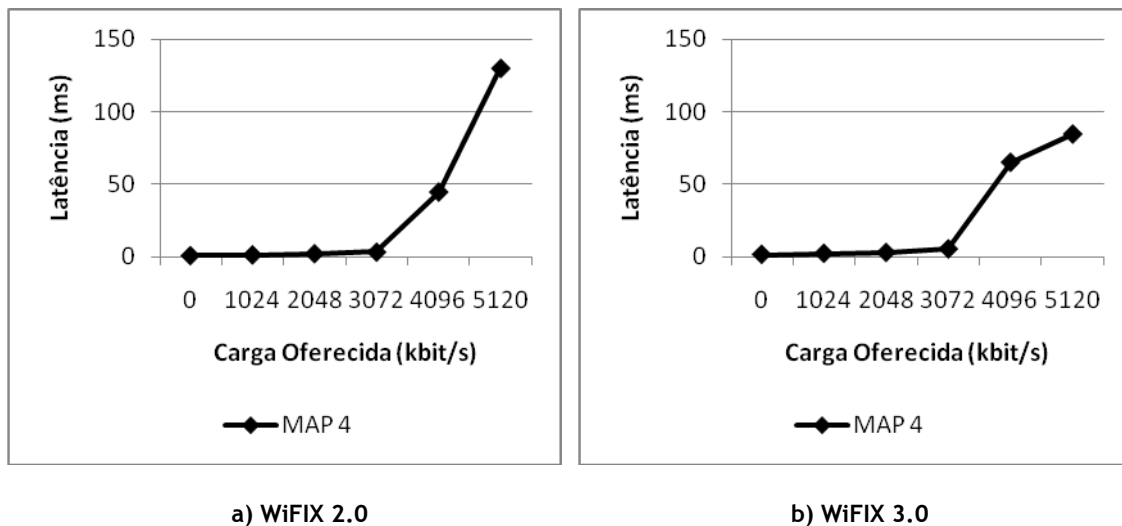


Figura 5.10 Latência a 1 salto

A latência é outro parâmetro essencial em aplicações de tempo real, e cujo controlo é importante para o bom desempenho da aplicação.

Para medir este parâmetro foi usada a ferramenta *ping*, que utiliza o protocolo ICMP. O teste foi realizado entre o MAP 1 e o pai do MAP 4 e entre o MAP 1 e o MAP 4; a variação da carga na rede foi feita utilizando a ferramenta IPerf e tráfego UDP *unicast* entre o MAP 1 e o MAP 4. Analisando a Figura 5.9 e a Figura 5.10 podemos concluir que até uma carga oferecida à rede de 3 Mbit/s existe uma latência aceitável (inferior a 10ms). Para uma carga superior, que coincide com o ponto de saturação da rede, é notório um aumento exponencial, um resultado esperado devido à carga excessiva presente em todos os nós. Esta carga provoca o enchimento das filas de espera dos MAPs, o que por sua vez provoca o descarte de pacotes. A carga aumenta também o número de colisões no meio, obrigando os MAPs a utilizar o

mecanismo DCF para minimizar essas colisões. A conjugação destes dois factores influencia directamente o aumento da latência dos pacotes.

## 5.2 Comparação entre WiFIX 2.0 e WiFIX 3.0 (Ligação degradada)

Nesta secção são apresentados os resultados para um cenário onde se pretende simular a degradação da qualidade da ligação entre o MAP 2 e o MAP 4. Para tal, foi criado um mecanismo que descarta uma percentagem (configurável) de pacotes prontos a enviar, embutido no código WiFIX. Este apresenta algumas limitações, pois não consegue simular um aumento do *jitter*. Mesmo assim, os parâmetros de *throughput* e *packet loss ratio* não são afectados pelo uso deste método.

Tal como no cenário anterior era expectável um melhor desempenho do WiFIX 3.0 para este caso, e de facto é o que sucede. Na versão mais recente do protocolo, o MAP 4 escolheu sempre como pai o MAP 3, pois este apresenta uma qualidade de ligação bastante superior. Como explicado no Capítulo 3, cada MAP irá avaliar a qualidade da ligação não só pelo custo que recebe de cada MAP vizinho, mas também pela quantidade de TRs que recebe desses vizinhos. Este último factor é preponderante neste cenário, pois trata-se de um cenário onde existem muitas perdas de pacotes (note-se que os TRs viajam em *broadcast* portanto não têm confirmação), o que em princípio implica perda de TRs, afectando portanto o custo final da métrica que cada MAP utiliza para escolher o seu pai.

Tal como na Secção 5.1 são usados os seguintes parâmetros para fazer a análise de desempenho: *throughput*, *jitter*, *packet loss ratio* e latência e será considerada uma perda de pacotes total de 50% na ligação entre o MAP 2 e MAP 4.

### 5.2.1 Throughput

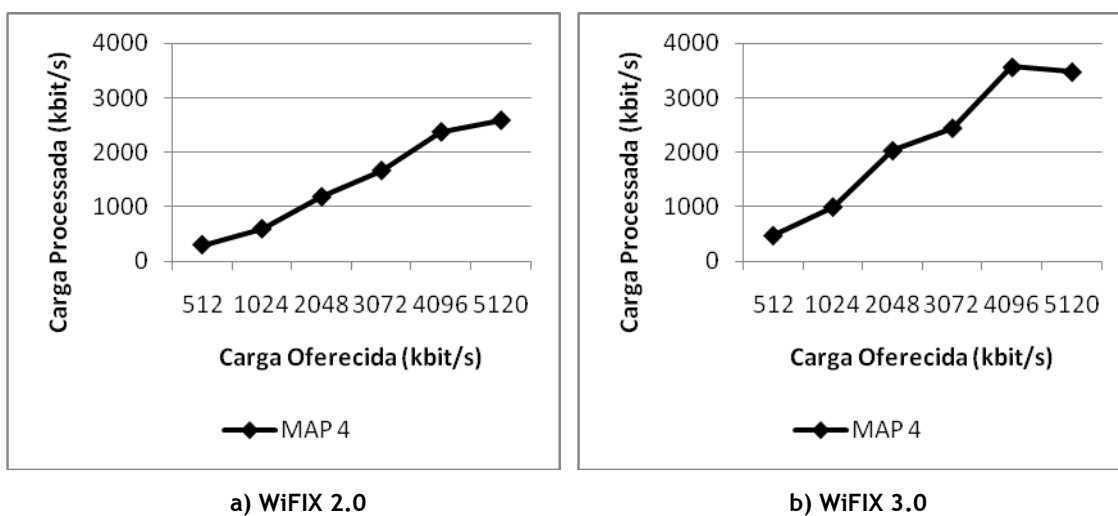


Figura 5.11 Unicast Throughput

Na Figura 5.11 é possível verificar que de facto existem vantagens na utilização da solução WiFIX 3.0. No WiFIX 2.0 o MAP 4 recebe sempre aproximadamente 60% do tráfego teórico, saturando por volta dos 2,5 Mbit/s. Já na versão 3.0 o ponto de saturação situa-se nos 3,5 Mbit/s. No caso da solução WiFIX 2.0 num cenário real os pacotes não seriam descartados, mas sim perdidos no meio. Portanto, os resultados seriam ainda piores do que os aqui apresentados. Para o protocolo TCP foram obtidos valores de *throughput* de 5,86 kbit/s e de 1918 kbit/s para a solução WiFIX 2.0 e WiFIX 3.0, respectivamente. O valor obtido na solução WiFIX 2.0 surge como consequência do mecanismo de confirmação de pacotes utilizado no protocolo TCP. Num meio com perdas, como é o caso deste cenário, o pacote ou a mensagem de confirmação podem ser perdidos várias vezes até serem recebidos correctamente no destino, diminuindo significativamente o *throughput*. No caso da solução WiFIX 3.0, a ligação com perdas entre o MAP 2 e o MAP 4 é evitada, obtendo-se um desempenho semelhante ao do cenário anterior (Secção 5.1.1 ).

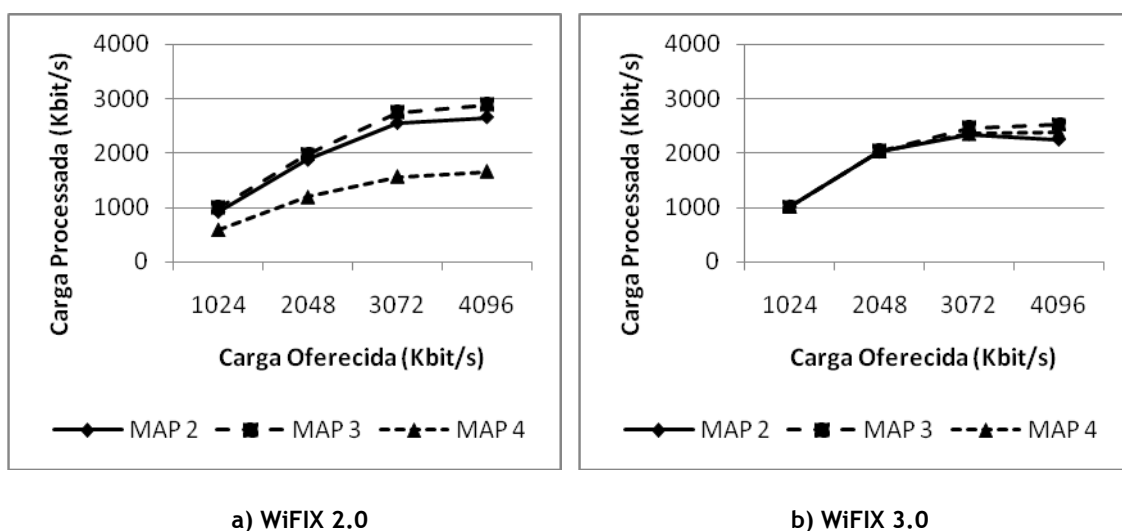
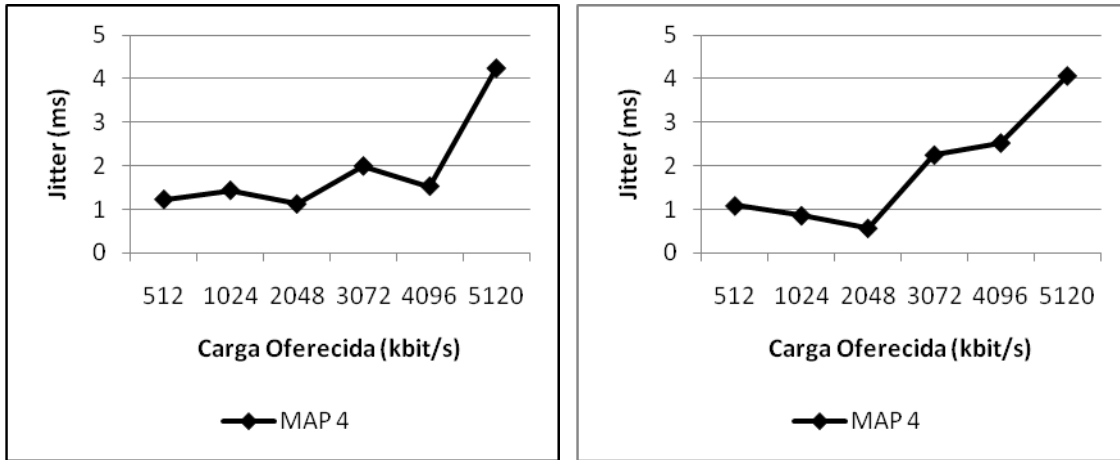


Figura 5.12 Broadcast Throughput

Para o caso do *broadcast*, na Figura 5.12 é visível o efeito da nova métrica introduzida no WiFIX 3.0, que evita a ligação com perdas dando origem a um desempenho semelhante para todos os MAPs. Nos testes realizados à solução WiFIX 2.0 é possível confirmar que apenas o MAP 4 sofre com a ligação de má qualidade e que o valor do *throughput* máximo obtido supera ligeiramente o valor obtido na solução WiFIX 3.0. Esta situação deve-se ao facto do MAP 2 descartar vários pacotes, devido ao mecanismo utilizado. Desta forma, existe uma menor concorrência ao meio quando comparado com a solução WiFIX 3.0 para o mesmo cenário, o que permitiu ao MAP 1 encaminhar pacotes a um débito superior culminando em valores de *throughput* superiores para os seus dois filhos, o MAP 2 e MAP 3.

## 5.2.2 Jitter

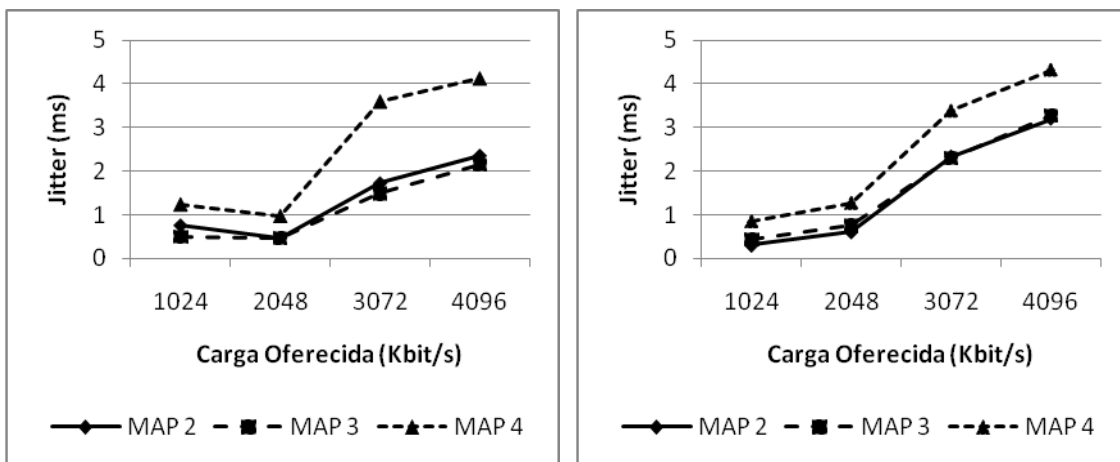


a) WiFIX 2.0

b) WiFIX 3.0

Figura 5.13 Unicast Jitter

Como é possível ver na Figura 5.13, em ambas as versões do protocolo, o *jitter* nunca excede os 5 ms no pior caso. Como mencionado no início da secção, este resultado deve-se ao método utilizado para simular o cenário. Num cenário real, os pacotes enviados para a rede são perdidos ou corrompidos durante o seu percurso e portanto existe um período em que o nó de origem fica à espera de receber a mensagem de confirmação das respectivas tramas. A soma do atraso sofrido no meio com o tempo necessário para concluir uma transmissão com sucesso de cada pacote afecta negativamente o *jitter*. Mas o método utilizado apenas descarta pacotes no nó de origem (neste caso o MAP 2), antes de eles serem enviados para o meio, e portanto esse nó não sofre os efeitos dos atrasos mencionados, condicionando os resultados do *jitter*.



a) WiFIX 2.0

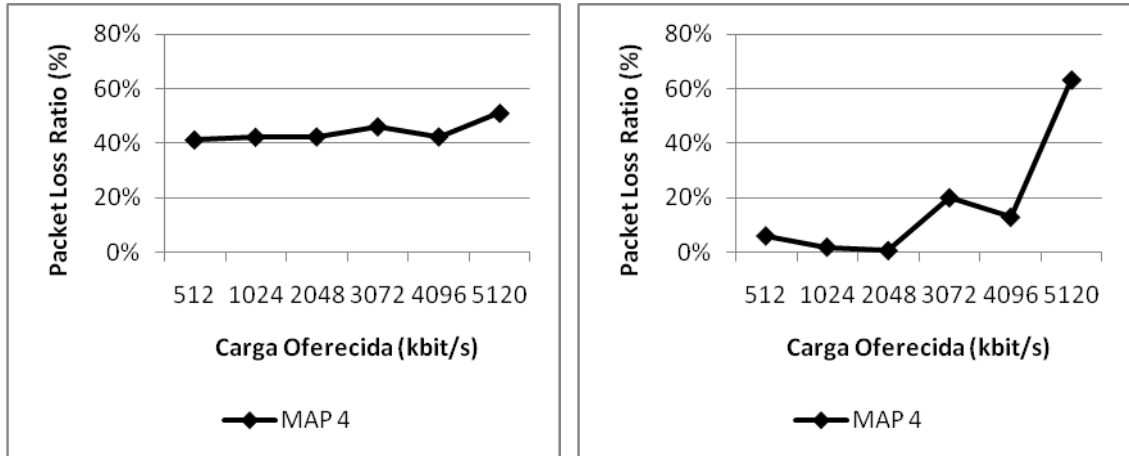
b) WiFIX 3.0

Figura 5.14 Broadcast Jitter



Analisando a Figura 5.14 observa-se um cenário semelhante, com o valor máximo do *jitter* a não exceder os 5 ms. É também possível verificar um valor médio do *jitter* superior no MAP 4, devido ao facto de este receber pacotes retransmitidos pelo MAP 2 ou MAP 3, consoante a versão do protocolo. Esses MAPs disputam o meio com o MAP 1, a origem do tráfego *multicast*, o que leva a um atraso adicional, aumentando o *jitter* apenas no MAP 4.

### 5.2.3 Packet Loss Ratio

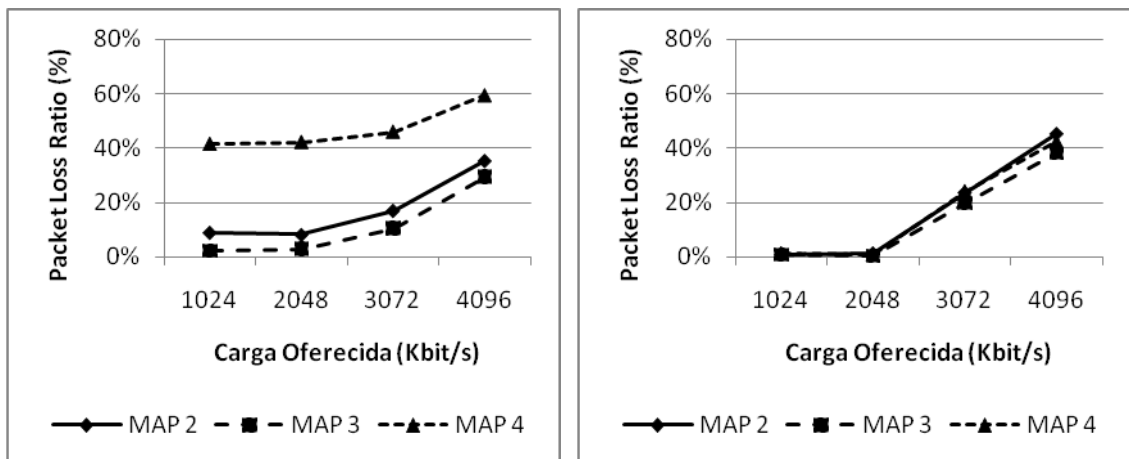


a) WiFIX 2.0

b) WiFIX 3.0

Figura 5.15 Unicast Packet Loss Ratio

Tanto em *unicast* como em *broadcast* é possível verificar que a nova métrica realmente melhora o desempenho do protocolo. No caso do *unicast* (Figura 5.15) foi possível passar de uma perda de pacotes de aproximadamente 45%, utilizando o WiFIX 2.0, para uma perda de pacotes inferior a 20% até uma carga oferecida de 4Mbit/s; a partir deste valor a rede satura.



a) WiFIX 2.0

b) WiFIX 3.0

Figura 5.16 Broadcast Packet Loss Ratio

Para o caso do *broadcast*, representado pela Figura 5.16 é também possível ganhos a nível do MAP 4, que passa de perdas de 40%-60% na solução WiFIX 2.0, para perdas 0%-40% na solução WiFIX 3.0, pelo que se pode concluir que a nova métrica introduz vantagens significativas, para um cenário tão provável como é o de uma má ligação.

## 5.2.4 Latência

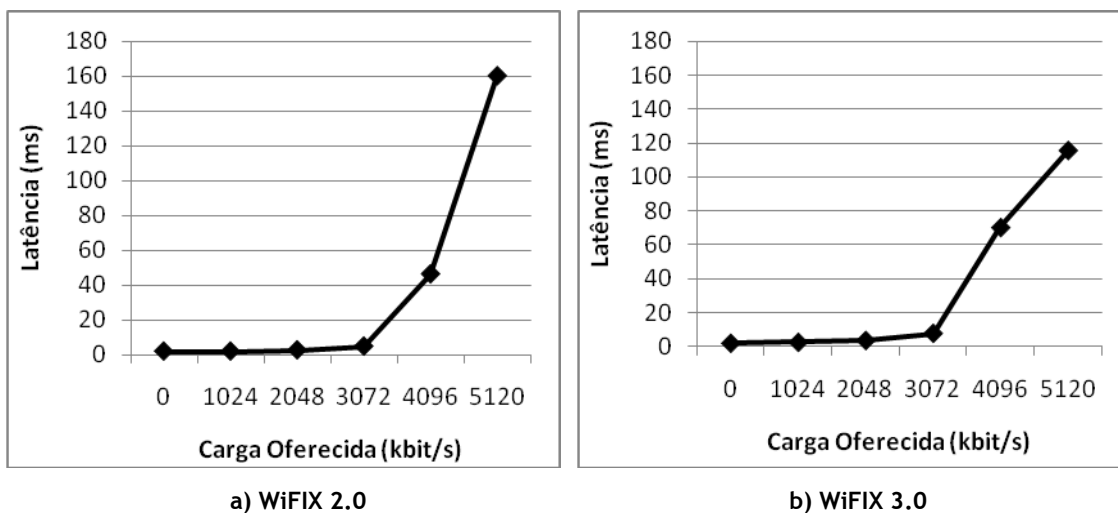


Figura 5.17 Latência a 2 saltos

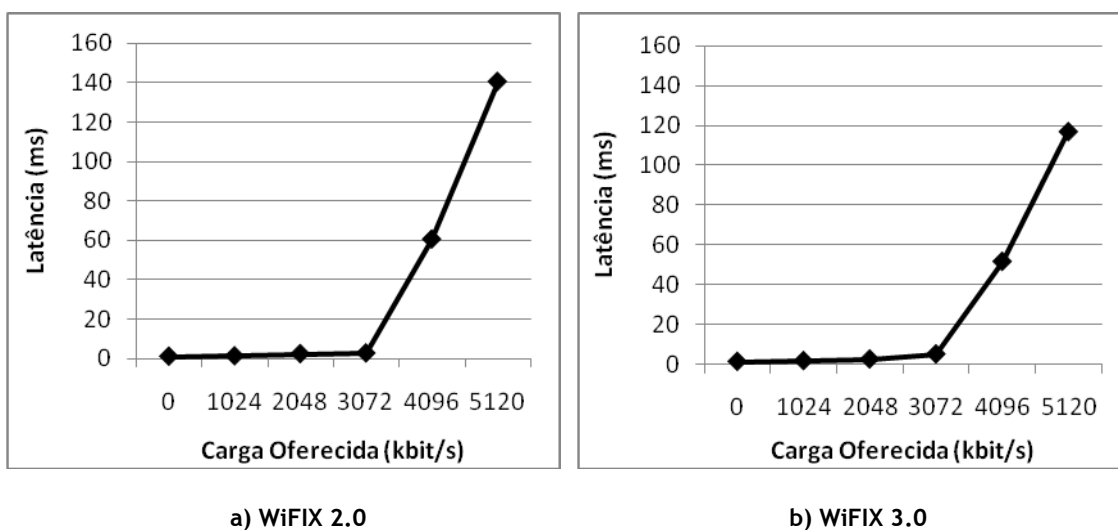


Figura 5.18 Latência a 1 salto

Tal como descrito na Secção 5.1.4 o método de teste para este parâmetro foi o de utilizar o IPerf, usando tráfego UDP *unicast* entre o MAP 1 e o MAP 4, para aumentar a carga na rede, e utilizar a ferramenta *ping* para calcular a latência a um e dois saltos.

Como é possível verificar, tal como no caso anterior, existe um desempenho aceitável no que refere à latência até uma carga oferecida de 3 Mbit/s. A partir daí assiste-se a um aumento muito significativo da latência, pelas razões discutidas na secção 5.1.4 .

### 5.3 Tempos de mobilidade dos MAP

Devido à introdução da nova métrica no protocolo, a mudança de pai por parte dos MAPs tornou-se mais frequente. Dessa forma, tomou-se em consideração o tempo de reaquisição dos fluxos num cenário de mudança de pai.

Tráfego UDP <i>unicast</i>	680 ms
Tráfego UDP <i>multicast</i>	840 ms

Tabela 5.1 Tempo de Mobilidade dos MAPs

Os resultados estão presentes na Tabela 5.1. Conclui-se que os valores são elevados, na ordem dos 680 ms para o caso do UDP *unicast*, e 840 ms no caso de UDP *multicast*. Na causa deste desempenho está a solução utilizada para a implementação dos túneis virtuais entre MAPs, que recorre ao mecanismo de *taps* disponível no *kernel* Linux. O mecanismo de mudança de pai tem três momentos importantes: (1) a abertura de uma ligação, criando um túnel virtual com o destino, (2) a leitura e/ou escrita para o túnel criado, e (3) o fecho desse túnel virtual. A destruição do túnel virtual contribui com aproximadamente 250ms para o atraso da reconfiguração total. Na implementação actual isto inviabiliza a possibilidade de suportar aplicações com requisitos de tempo real. O restante atraso de reconfiguração pode ser reduzido mediante optimizações do código WiFIX actual, algo que poderá ser feito como trabalho futuro.



# Capítulo 6

## Conclusões

Neste capítulo são apresentadas as conclusões relativas ao trabalho desenvolvido no âmbito desta dissertação, revistas as contribuições do trabalho e apresentados potenciais tópicos para trabalho futuro.

### 6.1 Concretização dos objectivos

Dos quatro objectivos propostos no Capítulo 3 foi apenas possível implementar três. Devido à falta de tempo provocada por contratemplos durante a construção e configuração do novo *test-bed* e a necessidade de corrigir alguns aspectos na implementação da solução WiFIX 2.0 que não estavam previstos à partida, o mecanismo de suporte à mobilidade dos terminais não foi implementado.

No entanto, os resultados obtidos com os mecanismos que foram implementados demonstram claramente as vantagens da solução WiFIX 3.0 em relação à versão anterior, evitando nós e ligações degradadas e até certo ponto balanceando a carga na rede emalhada. Estas vantagens resultaram em débitos superiores e valores de *jitter* e latência menores, quando comparados com a solução WiFIX 2.0. Já o mecanismo de suporte de mobilidade dos MAPs, na sua implementação actual tem um desempenho que torna inoportável o suporte de aplicações com requisitos de tempo real, pois apresenta um atraso nunca inferior a 600ms.

## 6.2 Contribuições

### 6.2.1 Métrica de encaminhamento

A nova métrica inteligente utilizada na solução WiFIX 3.0 foi uma contribuição concluída com sucesso que permite aos MAPs identificarem ligações degradadas e MAPs com carga elevada, podendo portanto evitar caminhos que os atravessem. Nesses casos foi possível obter melhores resultados na solução WiFIX 3.0, atingindo os objectivos proposto para este mecanismo.

### 6.2.2 *Soft Topology Change*

Esta contribuição, também concluída com sucesso, permitiu que a mudança de pai por parte de um MAP não só não corte a ligação dos fluxos *multicast*, devido ao facto das árvores *multicast* se basearem na topologia da rede, como o faz de uma forma transparente para todos os terminais associados a esse MAP. Desta forma é possível manter os fluxos *multicast* entre mudanças, apesar do desempenho durante essas mudanças não permitir um fluxo sem falhas para aplicações com requisitos de tempo real.

### 6.2.3 Implementação da métrica e do mecanismo *Soft Topology Change*

A implementação da nova métrica e do mecanismo *Soft Topology Change* foi mais uma contribuição concluída com sucesso que permitiu obter resultados experimentais num cenário real. Estes resultados permitiram validar a especificação das duas contribuições mencionadas anteriormente.

## 6.3 Trabalho futuro

Como trabalho futuro poderá ser considerada a implementação do mecanismo de gestão de mobilidade dos terminais especificado nesta solução, e uma reimplementação do código WiFIX utilizando um mecanismo diferente do utilizado actualmente, que recorre aos *drivers* TUN/TAP, para criar os túneis virtuais Eo11 entre MAPs. Seria também importante a realização de testes da solução WiFIX num *test-bed* de maiores dimensões e com equipamentos configurados para utilizar normas IEEE 802.11 superiores à norma IEEE 802.11b, como é o caso das normas IEEE 802.11g e 802.11n. O objectivo seria estudar a escalabilidade da solução proposta e analisar as suas vantagens numa rede de maior dimensão do que a utilizado nesta dissertação.

Por fim, seria também importante implementar o suporte para IPv6.

# Referências

- [1] R. Campos, *et al.*, "Network infrastructure extension using 802.1D-based wireless mesh networks," *Wireless Communications and Mobile Computing*, vol. 11, pp. 67-89, 2011.
- [2] G. R. Hiertz, *et al.*, "IEEE 802.11s: The WLAN Mesh Standard," *Wireless Communications, IEEE*, vol. 17, pp. 104-111, 2010.
- [3] R. Campos, *et al.*, "WiFIX+: A Multicast Solution for 802.11-based Wireless Mesh Networks," presented at the Proceedings of the 8th International Conference on Wireless Bardonecchia, Italy, 2011.
- [4] M. E. M. Campista, *et al.*, "Routing Metrics and Protocols for Wireless Mesh Networks," *Network, IEEE*, vol. 22, pp. 6-12, 2008.
- [5] S. Qiang and F. Xuming, "A Multi-metric AODV Routing in IEEE 802.11 s," in *Communication Technology, 2006. ICCT '06. International Conference on*, 2006, pp. 1-4.
- [6] J. G. Jetcheva and D. B. Johnson, "Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks," in *Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networking and Computing: MobiHoc 2001, October 4, 2001 - October 5, 2001*, Long Beach, CA, United states, 2001, pp. 33-44.
- [7] J. Lusheng and M. S. Corson, "A lightweight adaptive multicast algorithm," in *Global Telecommunications Conference, 1998. GLOBECOM 98. The Bridge to Global Integration. IEEE*, 1998, pp. 1036-1042 vol.2.
- [8] J. C. Lin and S. Paul, "RMTP: a reliable multicast transport protocol," in *INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, 1996, pp. 1414-1424 vol.3.
- [9] N. A. Gdoura, *et al.*, "MeshCAST: A Multi-channel Mult-interface Multicast Protocol for Mesh Networks," in *Wireless and Mobile Communications (ICWMC), 2010 6th International Conference on*, 2010, pp. 349-355.

- [10] J. J. Garcia-Luna-Aceves and E. L. Madruga, "A multicast routing protocol for ad-hoc networks," in *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 1999, pp. 784-792 vol.2.
- [11] L. Sung-Ju, *et al.*, "On-demand multicast routing protocol," in *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, 1999, pp. 1298-1302 vol.3.
- [12] S. Fellah and M. Kaddour, "A Multicast Routing Protocol adapted to the characteristics of Wireless Mesh Networks," in *Machine and Web Intelligence (ICMWI), 2010 International Conference on*, 2010, pp. 174-179.
- [13] H. Yan and D. Perkins, "BASH: A backhaul-aided seamless handoff scheme for Wireless Mesh Networks," in *World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a*, 2008, pp. 1-8.
- [14] V. Navda, *et al.*, "Design and evaluation of iMesh: an infrastructure-mode wireless mesh network," in *World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a*, 2005, pp. 164-170.
- [15] Y. Amir, *et al.*, "Fast handoff for seamless wireless mesh networks," presented at the Proceedings of the 4th international conference on Mobile systems, applications and services, Uppsala, Sweden, 2006.
- [16] H. Rongsheng, *et al.*, "A Mobility Management Scheme for Wireless Mesh Networks," in *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, 2007, pp. 5092-5096.
- [17] W. Hung-Yu, *et al.*, "Seamless Handoff Support in Wireless Mesh Networks," in *Operator-Assisted (Wireless Mesh) Community Networks, 2006 1st Workshop on*, 2006, pp. 1-8.