



**Universidade de  
Aveiro**  
2010

Departamento de Electrónica,  
Telecomunicações e Informática

**Carlos Manuel  
Martins Marques**

**Redes em Malha Virtuais para Optimização de  
Conectividade**





**Universidade de  
Aveiro**  
2010

Departamento de Electrónica,  
Telecomunicações e Informática

**Carlos Manuel  
Martins Marques**

**Redes em Malha Virtuais para Optimização de  
Conectividade**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica da Prof. Dra. Susana Sargento, Professora auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro



Dedico este trabalho aos meus pais pelo incansável apoio.



## **O júri**

Presidente

**Prof. Dr. António Rui de Oliveira e Silva Borges**

Professor associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Orientadora

**Prof. Dra. Susana Isabel Barreto de Miranda Sargento**

Professora Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Arguente

**Prof. Dra. Maria Solange Pires Ferreira Rito Lima**

Professora Auxiliar do Departamento de Informática da Escola de Engenharia da Universidade do Minho





## **Agradecimentos**

Desde logo gostaria de agradecer à minha família, todo o apoio dado, para conseguir superar mais esta etapa do meu percurso académico.

Gostaria também de agradecer, aos meus verdadeiros amigos que me apoiaram, sempre presentes nos melhores momentos mas também nos mais difíceis.

Gostaria de agradecer ao Ricardo Matos pela grande ajuda na realização deste trabalho, tornando-se não só um colaborador mas também um amigo. Sem ele não teria sido possível desenvolver o trabalho num tão curto espaço de tempo.

Gostaria ainda de agradecer à Prof. Susana Sargento, pela constante disponibilidade e orientação profissional, técnica e científica. Agradeço também toda a motivação e apoio dados para a realização deste trabalho.

Por fim um muito obrigado a todas as pessoas que por diferentes razões, contribuíram para a realização desta dissertação, sendo de destacar o Tiago Condeixa, o Luís Barreto e o Marcel Castro entre outros.



**Palavras-chave**

Redes baseadas em contexto, Redes em malha sem fios, Redes Virtuais, Serviços Personalizados, Gestão e controlo, Métricas de avaliação de desempenho, QoS, QoE, NS-2

**Resumo**

Actualmente, os utilizadores têm necessidades muito distintas, assim como os serviços a que acedem. Essas necessidades são constituídas por vários parâmetros, que se designam por contexto do utilizador e dos seus serviços, tais como, padrões de mobilidade, requisitos de qualidade de serviço, disponibilidade, confiança, segurança, custo, etc.

A dissertação apresentada encontra-se no âmbito de uma nova abordagem de rede que permite ter redes em malha sem fios muito adaptativas através do suporte de várias redes lógicas construídas sobre a mesma rede física. Nesta abordagem as redes em malha sem fios são virtualizadas, e cada rede virtual é específica para diferentes características de contexto mapeadas em níveis. De entre diversos desafios associados a esta nova abordagem, esta dissertação propõe-se a simular os seguintes mecanismos: i) configuração/mapeamento dos recursos/topologias de cada rede virtual, de acordo com as características de contexto dos utilizadores e aplicações a que se destinam; ii) associação de um determinado utilizador à rede virtual mais apropriada, introduzindo mecanismos de comparação entre o contexto do utilizador e as características das redes virtuais; iii) procura de um ponto de ligação à rede virtual através de mecanismos de descoberta locais ou globais (distribuídos), que envolvem diferentes mecanismos para mapeamento da informação de contexto nos elementos de rede.

De forma a implementar a proposta apresentada recorreu-se ao simulador de redes NS-2. Os resultados da avaliação do desempenho da rede foram obtidos usando diferentes cenários com a variação de diferentes parâmetros, tais como o número de redes virtuais e de utilizadores, o tamanho da rede física e das redes virtuais, os padrões de mobilidade e o tipo de tráfego dos utilizadores. Através das simulações efectuadas é possível concluir que aplicando a solução proposta, são fornecidos diferentes serviços aos utilizadores mediante os seus requisitos de contexto, sem que haja impacto significativo no desempenho da rede.



**keywords**

Contex-based Networks, Wireless Mesh Networks, Virtual Networks, Personalized Services, Management and Control, performance evaluation metrics, QoS, QoE, NS-2

**abstract**

Currently, users have different needs, as well as the services they access. These needs consist of various parameters designated by the user's context and its services, such as mobility patterns, requirements of quality of service, availability, reliability, safety, cost, etc.

This Thesis is in the scope of a new network approach that allows wireless mesh networks to be very adaptive through the support of multiple logical networks built on the same physical network. In this approach the wireless mesh networks are virtualized, and each virtual network is specific to different context requirements that are mapped in levels. Among the various challenges associated with this new approach, this dissertation proposes to simulate the following mechanisms: i) configuration/mapping of resources/topologies for each virtual network, according to the characteristics of the context of its users and applications; ii) association of a particular user to the most appropriate virtual network, introducing mechanisms to compare the user's context against the characteristics of virtual networks; iii) searching for a connection point to the virtual network through mechanisms for local or global (distributed) discovery, which involves different mechanisms for context information mapping in network elements.

In order to implement this architecture, the network simulator NS-2 was used. The evaluation results of the network performance were obtained using different scenarios with the variation of different parameters, such as the number of virtual networks and users, the WMN size and the size of the virtual networks, mobility patterns and traffic of the users. Through the implemented simulation setup, it is possible to conclude that with the deployment of the proposed solution, different services are provided to users without a significant impact on the network performance.



# Índice

<b>Índice</b> .....	<b>i</b>
<b>Acrónimos</b> .....	<b>v</b>
<b>Índice de Figuras</b> .....	<b>ix</b>
<b>Índice de Tabelas</b> .....	<b>xi</b>
<b>1. Introdução</b> .....	<b>1</b>
1.1 Motivação .....	1
1.2 Objectivos.....	2
1.3 Contributos deste trabalho.....	3
1.4 Organização da dissertação .....	3
<b>2. Estado da arte</b> .....	<b>5</b>
2.1 Introdução.....	5
2.2 Redes em malha sem fios .....	5
2.2.1 Arquitectura das redes em malha sem fios.....	6
2.2.2 Aplicações das redes em malha sem fios.....	8
2.2.3 Projectos com redes em malha sem fios .....	8
2.3 Virtualização de redes .....	11
2.3.1 Tipos de redes virtuais.....	11
2.3.2 Elementos da virtualização de redes.....	13
2.3.3 Técnicas de virtualização .....	14
2.3.4 Projectos com redes virtuais.....	17
2.4 Gestão de informação de contexto.....	19
2.4.1 Redes com informação de contexto .....	20
2.4.2 Modelação de contexto.....	20
2.4.3 Descoberta de informação de contexto .....	22
2.4.4 Projectos de redes em malha sem fios com contexto.....	22
2.5 Conclusão.....	23
<b>3. Redes em malha sem fios baseadas em contexto</b> .....	<b>25</b>
3.1 Introdução.....	25

3.2	Arquitectura proposta e objectivos, desafios inerentes.....	25
3.2.1	Plano de dados .....	26
3.2.2	Plano de controlo .....	27
3.3	Conclusão.....	32
<b>4.</b>	<b>Implementação.....</b>	<b>33</b>
4.1	Introdução.....	33
4.2	Network Simulator (NS 2.33) .....	33
4.2.1	Visão geral.....	33
4.2.2	Arquitectura.....	34
4.2.3	Implementação do conceito proposto no simulador.....	35
4.3	Detalhes de implementação .....	36
4.3.1	Plano de dados .....	36
4.3.1.1	Suporte de virtualização de redes .....	36
4.3.1.2	Atribuição de recursos apropriados.....	39
4.3.2	Plano de controlo .....	40
4.3.2.1	Mapeamento da informação de contexto .....	41
4.3.2.2	Configuração/adaptação dos recursos/topologias.....	41
4.3.2.3	Mecanismos de descoberta.....	42
4.3.3	Cooperação entre as diversas funcionalidades implementadas .....	50
4.4	Conclusão.....	51
<b>5.</b>	<b>Avaliação: resultados e discussão .....</b>	<b>53</b>
5.1	Introdução.....	53
5.2	Cenários e detalhes do simulador .....	53
5.2.1	Tamanho da rede.....	53
5.2.2	Parâmetros e níveis de contexto.....	54
5.2.3	Número de redes virtuais.....	55
5.2.4	Mapeamento dos parâmetros de contexto nas redes virtuais.....	55
5.2.5	Número de utilizadores .....	56
5.2.6	Descoberta das redes virtuais.....	56



5.2.7	Mobilidade dos utilizadores .....	56
5.2.8	Métricas de avaliação de desempenho .....	57
5.3	Influência da virtualização .....	57
5.3.1	Tráfego UDP .....	58
5.3.2	Tráfego TCP .....	62
5.4	Gestão local e global .....	67
5.4.1	Atraso .....	67
5.4.2	Impacto do controlo .....	70
5.5	Influência da mobilidade dos utilizadores .....	72
5.6	Discussão .....	75
5.7	Conclusão.....	78
<b>6.</b>	<b>Conclusões e linhas futuras de investigação .....</b>	<b>79</b>
	<b>Bibliografia.....</b>	<b>81</b>



## Acrónimos

---

### **8**

802.11e: QoS version of the IEEE Standard for Wireless LAN ·

---

### **A**

AODV: Ad hoc On-Demand Distance Vector ·

---

### **C**

CE: Customer Edge ·

CFB: Contention Free Burst ·

---

### **D**

DHT: Distributed Hash Table ·

---

### **E**

EDCA: Enhanced Distribution Coordinate Access ·

ETT: Estimated Estimated Transmission Time ·

ETX: Expected Transmission Count ·

---

### **I**

IP: Internet Protocol ·

ISP: Internet Service Provider ·

---

### **M**

MAC: Media Access Control ·

MIT: Massachusetts Institute of Technology ·

---

## ***N***

NAM: Network Animator ·

NGNs: Next Generation Networks ·

NS-2: Network Simulator version 2 ·

---

## ***O***

OLSR: Optimized Optimized Link State Routing ·

---

## ***P***

P2P: Peer-to-Peer ·

PE: Provider Edge ·

PPVPN: Provider-Provisioned VPN ·

---

## ***Q***

QoE: Quality of Experience ·

QoS: Quality of Service ·

---

## ***R***

RTT: Round-trip delay time ·

---

## ***S***

SGML: Standart Generic Markup Language ·

SIFS: Short Interframe Space ·

SP: Service Provider ·

---

## ***T***

TCP: Transmission Control Protocol ·

---

---

***U***

UDP: User Datagram Protocol ·

---

***V***

VN: Virtual Network ·

VPN: Virtual Private Network ·

---

***W***

WLAN: Wireless Local Area Network ·

WMN: Wireless Mesh Network ·



## Índice de Figuras

Figura 1: Arquitectura de uma rede em malha sem fios híbrida[5] .....	7
Figura 2: Aplicações das redes em malha: sistemas de transporte inteligente (esquerda) [6] e redes de vizinhança em áreas rurais (direita) [117].....	9
Figura 3: Arquitectura de virtualização de redes [29] .....	14
Figura 4: Solução para redes em malha sem fios baseadas em contexto [110].....	25
Figura 5: Processo de associação de um utilizador à VN pretendida[4].....	30
Figura 6: Integração das características de contexto das VNs no mecanismo de descoberta distribuído[5].....	31
Figura 7: Correspondência entre OTcl e C++ [111].....	34
Figura 8: Arquitectura do nó móvel original à esquerda. Arquitectura modificada do nó móvel, com suporte a múltiplas interfaces, à direita [112] .....	37
Figura 9: Arquitectura simplificada do AODV no NS-2 .....	39
Figura 10: Princípio de funcionamento do mecanismo de descoberta local .....	42
Figura 11: Troca de mensagens do protocolo .....	43
Figura 12: Diagrama de blocos da implementação do Bambo em NS-2 [113] .....	46
Figura 13: Tabela de encaminhamento do nó X. Os nós brancos são os nós da <i>leafset</i> do nó X, os arcos são as entradas da tabela de encaminhamento do nó X. [116] .....	47
Figura 14: Cooperação entre as diversas funcionalidades implementadas .....	49
Figura 15: Topologia da rede 4X4.....	54
Figura 16: Rede 4x4 com virtualização e tráfego UDP.....	59
Figura 17: Rede 4x4 sem virtualização e tráfego UDP .....	59
Figura 18: Rede 7x7 com virtualização e tráfego UDP .....	60
Figura 19: Rede 7x7 sem virtualização e tráfego UDP.....	60
Figura 20: Rede 10x10 com virtualização e tráfego UDP .....	61
Figura 21: Rede 10x10 sem virtualização e tráfego UDP .....	61
Figura 22: Rede 4x4 com virtualização e tráfego TCP .....	63
Figura 23: Rede 4x4 sem virtualização e tráfego TCP .....	63
Figura 24: Rede 7x7 com virtualização e tráfego TCP .....	64
Figura 25: Rede 7x7 sem virtualização e tráfego TCP .....	64
Figura 26: Rede 10x10 com virtualização e tráfego TCP .....	65
Figura 27: Rede 10x10 sem virtualização e tráfego TCP .....	65
Figura 28: Tempos de descoberta e extensão para uma rede 4x4. <i>Scenario 1</i> - Local, <i>Scenario 2</i> - Vizinhança, <i>Scenario 3</i> – Distribuído .....	68

Figura 29: Tempos de descoberta e extensão para uma rede 7x7. <i>Scenario 1</i> - Local, <i>Scenario 2</i> - Vizinhança, <i>Scenario 3</i> – Distribuído .....	69
Figura 30: Tempos de descoberta e extensão para uma rede 10x10. <i>Scenario 1</i> - Local, <i>Scenario 2</i> - Vizinhança, <i>Scenario 3</i> – Distribuído .....	70
Figura 31: <i>Overhead</i> de controlo do <i>Scenario 2</i> – Vizinhança, e do <i>Scenario 3</i> - Distribuído, para diferentes tamanhos da rede. ....	71
Figura 32: Tempos de descoberta e extensão (esquerda em cima), tempo de restabelecimento de conectividade (direita em cima), <i>packet loss</i> (esquerda em baixo), tempo de actualização do P2P <i>overlay</i> (direita em baixo) para uma rede 7x7 e 18 VNs (6 VNs por canal). <i>Scenario 1</i> - Local, <i>Scenario 2</i> - Vizinhança, <i>Scenario 3</i> – Distribuído ...	73



## Índice de Tabelas

Tabela 1: Diferentes projectos de virtualização de redes [29] .....	18
Tabela 2: Caracterização de cada utilizador através de parâmetros de contexto.....	28
Tabela 3: Caracterização de uma VN através de parâmetros de contexto.....	28
Tabela 4: Campos do cabeçalho das mensagens.....	44
Tabela 5: Mapeamento da informação de contexto na chave .....	48
Tabela 6: Mapeamento dos parâmetros de contexto nas VNs .....	56
Tabela 7: Número de pesquisas feitas (exprimidas em %) para cada um dos diferentes mecanismos de busca .....	72
Tabela 8: Número de pesquisas feitas (exprimidas em %) para cada um dos diferentes mecanismos de busca em função do número de utilizadores por VN para uma rede 7x7 com 18 VNs .....	74
Tabela 9: Número de vezes que o P2P <i>overlay</i> de controlo foi actualizado (exprimidas em %) para cada um dos diferentes canais, assim como o total dos diferentes canais, em função do número de utilizadores por VN para uma rede 7x7 com 18 VNs (6 VNs por canal).....	75



# 1. Introdução

## 1.1 Motivação

Hoje em dia, os utilizadores móveis estão cada vez mais exigentes ao pretenderem serviços personalizados, ou seja, têm necessidades, preferências, e requisitos muito distintos. As redes também têm características diferentes e suportam níveis diferentes dos parâmetros acima referidos. A limitação nas arquitecturas de redes existentes para trabalhar com serviços personalizados, motiva a comunidade científica para propor novas soluções.

A mobilidade faz parte das nossas vidas, e assume um papel mais relevante nas redes de futuro. Os clientes querem estar ligados em qualquer lugar a qualquer momento, sem perder a conectividade enquanto se movem. Uma arquitectura de rede terá que ter em conta os cenários totalmente móveis, mudando para uma nova configuração a cada momento.

A auto-organização e a auto-configuração das redes em malha sem fios (*Wireless Mesh Networks* - WMNs) [1] torna-as adequadas para lidar com os novos e desafiantes requisitos dos utilizadores, dispositivos e serviços. Esses requisitos definem o contexto dos utilizadores em relação a um determinado serviço, o qual pode conter parâmetros, tais como a Qualidade de Serviço (QoS) necessária, requisitos de segurança, preferências de custo, padrões de mobilidade, etc., todos eles proporcionando um determinado nível de Qualidade de Experiência (QoE) para os utilizadores.

A virtualização de redes [2] tem a capacidade de criar múltiplas redes virtuais numa única infra-estrutura. Este conceito pode ser usado para personalizar as redes até um grau elevado, providenciando redes virtuais (*Virtual Networks* - VNs) personalizadas para utilizadores e serviços, com um contexto específico. A estrutura heterogénea das WMNs, contendo um grande número de nós que integram diferentes requisitos de contexto com vários objectivos em conflito, pode beneficiar da separação dos seus clientes em VNs específicas, programáveis e adaptáveis.

O tema desta Dissertação é a avaliação de uma arquitectura de redes em malha sem fios, que através da virtualização possibilite a existência de várias VNs, cada uma específica para diferentes características de contexto. Para tal, a arquitectura deve-se adaptar dinamicamente à variação da informação de contexto dos utilizadores ou aplicações. A fim de realizar toda a gestão da solução foram criados mecanismos de controlo da mesma, para descobrir as VNs que mais se adequam ao utilizador ou aplicações (tanto a nível local como global), criar e reconfigurar VNs dinamicamente, e

para configuração/mapeamento dos recursos/topologias de cada VN, em conformidade com requisitos de contexto dos utilizadores.

## 1.2 Objectivos

Em [3][4][5] foi proposta uma nova abordagem de rede que permite ter redes em malha sem fios muito adaptativas através do suporte de VNs múltiplas (várias redes lógicas sobre a mesma rede física).

Esta Dissertação tem como objectivo implementar e avaliar no simulador de redes (*Network Simulator – NS-2*), a nova abordagem de rede acima referida, nomeadamente o conceito da virtualização de redes, o mapeamento do contexto em diferentes níveis para se poder seleccionar, criar e reconfigurar as VNs, e toda a gestão da rede de maneira a possibilitar a adaptação desta à mudança das preferências dos utilizadores.

Nesta abordagem é necessário integrar mecanismos de controlo da rede, para que seja possível incluir informação de contexto na mesma, criar e reconfigurar as redes virtuais, descobrir (de uma forma local e global) os nós/redes virtuais que mais se adequam às necessidades de utilizadores e dos seus serviços (ao seu contexto). Estes mecanismos permitem descobrir qual o nó da rede em malha mais próximo que pertence a uma rede virtual com as características necessárias para um determinado utilizador, e até que ponto será necessário reconfigurar essa rede virtual ou criar uma rede nova.

De uma forma sucinta, os objectivos desta Dissertação de Mestrado são os seguintes:

- Estudar o estado da arte dos diferentes conceitos envolvidos, tais como a virtualização, as redes em malha e os mecanismos de gestão da informação de contexto;
- Identificar a informação de contexto parametrizável e os mecanismos necessários para a implementação da abordagem de rede descrita anteriormente, assim como métricas comparativas que permitam avaliar o comportamento desses mecanismos;
- Adequar o conceito de virtualização ao simulador;
- Suportar mecanismos de descoberta, que possibilitem a descoberta de redes virtuais tanto a nível local como a nível global;
- Criar um mecanismo capaz de configurar/adaptar os recursos/topologia de cada rede virtual mediante a variação de contexto;
- Desenvolver cenários de teste dos mecanismos previamente desenvolvidos;
- Proceder à elaboração de testes, recolha de resultados e análise comparativa para melhoramento dos mecanismos propostos.

### **1.3 Contributos deste trabalho**

Como resultado da realização dos objectivos acima propostos, este trabalho fornece o seguinte conjunto de contribuições:

- O desenvolvimento de uma arquitectura de rede baseada em contexto no NS-2, que implementa a virtualização de rede, assim como os diferentes mecanismos de controlo e gestão da mesma, tais como o mecanismo de descoberta local e o mecanismo de descoberta global distribuído na rede;
- Uma avaliação, através de simulações, do desempenho da arquitectura global em diversas áreas e em resposta a diferentes situações, a fim de testar a robustez e fiabilidade da solução desenvolvida.

### **1.4 Organização da dissertação**

O trabalho de pesquisa desenvolvido nesta dissertação está organizado em 6 capítulos principais. Estes capítulos explicam detalhadamente o trabalho realizado em diferentes fases da pesquisa, introdução, estado da arte, ideias conceptuais, execução e avaliação da arquitectura, e conclusão/linhas de investigação futuras. É de salientar que as imagens apresentadas retiradas de referências se encontram em Inglês.

- O capítulo 1 contextualiza a dissertação, na situação actual dos paradigmas de nova geração. Apresenta também os objectivos e contribuições desta dissertação;
- O capítulo 2 fornece uma visão geral do estado da arte nesta área de pesquisa. Consiste numa visão geral dos conceitos de redes em malha, das redes virtuais, e da gestão de informação de contexto;
- O capítulo 3 apresenta as principais ideias, conceitos e componentes da solução proposta anteriormente, que serão implementados e avaliados no simulador;
- O capítulo 4 explica a implementação da solução proposta no capítulo 3, através do Network Simulator versão 2.33 (NS-2.33). É feita uma breve introdução ao simulador. Em seguida é efectuada uma descrição detalhada dos principais mecanismos implementados, assim como a explicação de como estes interagem;
- O capítulo 5 apresenta uma avaliação e discussão da implementação efectuada, através da realização de testes que permitem avaliar o desempenho da solução em cenários e condições específicas com base em diferentes métricas;
- O capítulo 6 sintetiza os principais resultados e fornece uma visão global de todo o trabalho realizado. Descreve também as linhas de investigação a desenvolver posteriormente a esta dissertação, como resultado de uma avaliação dos mecanismos ainda em falta e possíveis melhorias que poderão vir a ser realizadas.



## 2. Estado da arte

### 2.1 Introdução

Actualmente, os utilizadores têm necessidades distintas, exigindo cada vez mais serviços personalizados. As suas necessidades, exigências, preferências são caracterizadas em vários parâmetros designados de contexto. Para lidar com estes novos desafios dos utilizadores, as redes em malha sem fios, devido à sua auto-organização e auto-configuração, são promissoras e adequadas. Por outro lado, a virtualização de redes, pretende criar múltiplas redes virtuais numa única infra-estrutura. Por ultimo, e para gerir toda a informação de contexto, é necessário dotar os elementos de rede com inteligência. Deste modo é possível controlar e gerir toda a informação de contexto de uma maneira eficiente e distribuída na rede.

Na secção 2.2 são descritas as redes em malha sem fios, assim como as suas aplicações e projectos onde são utilizadas. Na secção 2.3 é descrito o conceito da virtualização de redes, assim como técnicas de virtualização e projectos onde é aplicado este conceito. Na secção 2.4 são descritas várias abordagens para a gestão e modelação do contexto, assim como mecanismos de descoberta do contexto na rede. Finalmente, a secção 2.5 apresenta um resumo do capítulo.

### 2.2 Redes em malha sem fios

As redes em malha sem fios (WMNs) surgiram como uma tecnologia chave para a próxima geração de redes sem fios.

As redes em malha sem fios são dinamicamente auto-organizadas e auto-configuradas, com os nós na rede a automaticamente estabelecerem uma rede *ad hoc* estática e a manterem a conectividade da rede através de ligações sem fios. Em vez de ser outro tipo de redes *ad hoc*, as WMNs diversificam as capacidades das redes *ad hoc*. Esta característica traz inúmeras vantagens para as WMNs, tais como um baixo custo inicial, uma fácil manutenção da rede, e elevada robustez.

Para além de serem amplamente aceites nos sectores tradicionais de aplicação de redes *ad hoc*, as WMNs estão a ser cada vez mais comercializadas em muitos outros cenários de aplicações, tais como: redes domésticas de banda larga; redes comunitárias; domótica; redes metropolitanas; redes corporativas de alta velocidade e redes de acesso sem fio à Internet pública oferecendo uma alternativa às tecnologias com fios (para fornecer acesso à Internet de banda larga especialmente em zonas rurais).

## 2.2.1 Arquitectura das redes em malha sem fios

- ***Routers e Clientes em malha sem fios***

As WMNs são compostas por dois tipos de nós: routers em malha sem fios (*Mesh Routers*) e clientes em malha sem fios (*Mesh Clients*).

Um router em malha sem fios contém funções de encaminhamento adicionais, em relação a um router sem fios tradicional. Através de comunicações com vários saltos, o mesmo nó pode ser alcançado por um router em malha sem fios com menos potência de transmissão. Para melhorar ainda mais a flexibilidade das redes em malha, um router em malha sem fios é normalmente equipado com múltiplas interfaces sem fios construídas na mesma ou em diferentes tecnologias de acesso. Apesar de todas estas diferenças, os routers em malha sem fios ou tradicionais são geralmente construídos com base numa plataforma de *hardware* semelhante. Routers em malha sem fios assumem uma mobilidade mínima e formam o *backbone* para os clientes em malha sem fios.

Embora os clientes em malha sem fios também possam funcionar como routers, a plataforma de *hardware* e *software* para eles pode ser muito mais simples do que para os routers em malha sem fios. Por exemplo, os protocolos de comunicação para os clientes em malha sem fios podem ser leves, visto que, funções de *gateway* ou *bridge* não existem em clientes em malha sem fios, e apenas uma única interface é necessária.

De revelar que para a rede em malha entre routers e clientes, as funcionalidades *gateway/bridge* nos routers permitem a integração de WMNs com várias outras redes.

- ***Tipos de redes em malha sem fios***

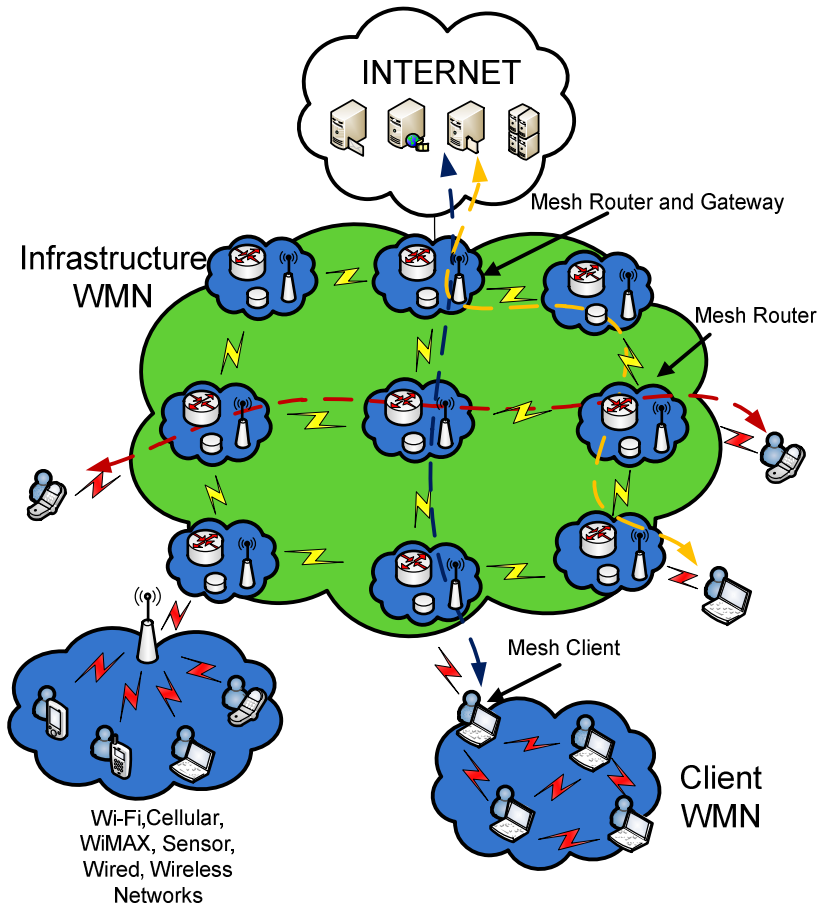
Três tipos de redes em malha sem fios podem ser identificados.

Nas *Infrastructure WMNs* os routers formam uma rede que oferece conectividade aos clientes. A rede é auto-configurável, auto-adaptável e oferece funcionalidades de *gateway* para ligações de redes com fios, e acesso à Internet.

As *Client WMNs* são redes *ad hoc* formadas por clientes que comunicam entre si. Não existe nenhum router ou infra-estrutura dedicada, de modo que os clientes têm de ser auto-configuráveis e actuar como routers.

As *Hybrid WMNs* combinam as vantagens das outras duas WMNs. Tanto os clientes como os routers participam no encaminhamento do tráfego, aumentando a cobertura e conectividade da rede. A infra-estrutura fornecida pelos routers forma o *backbone* para a rede e actua como *gateway* para redes com fios [1]. Um exemplo de uma WMN híbrida pode ser vista na *Figura 1*.





**Figura 1: Arquitectura de uma rede em malha sem fios híbrida [5]**

- **Características das redes em malha**

As características das WMNs são descritas seguidamente, onde se considerou a arquitectura híbrida uma vez que possui todas as vantagens das WMNs:

- As WMNs suportam redes *ad hoc*, e têm a capacidade de auto-formação, auto-adaptação e auto-organização;
- As WMNs são redes sem fios multi-salto, mas com a infra-estrutura/*backbone* fornecida pelos routers;
- Os routers têm mobilidade mínima e realizam encaminhamento dedicado e auto-configuração, o que diminui significativamente a carga de clientes e de outros nós finais;
- A mobilidade dos nós finais é facilmente suportada pela infra-estrutura sem fios;
- Os routers em malha integram redes homogéneas, incluindo com e sem fios. Existem múltiplos tipos de redes de acesso;
- As restrições ao consumo de energia são diferentes para routers e clientes;
- As WMNs não são autónomas e necessitam de ser compatíveis e interoperáveis com outras redes com e sem fios.

## 2.2.2 Aplicações das redes em malha sem fios

As aplicações comerciais existentes e emergentes de WMNs são tão diversas quanto as necessidades das comunicações. Exemplos destas aplicações são os ambientes difíceis, tais como situações de emergência, túneis, plataformas de petróleo, vigilância do campo de batalha em tempo real, aplicações de vídeo a bordo dos transportes públicos ou monitorização dos carros de corrida em tempo real.

Os chamados sistemas de transporte inteligente podem ser realizados de forma eficiente utilizando a tecnologia em malha para fornecer informações em tempo real sobre viagens relacionadas com o transporte público e/ou privados, tais como: tráfego, congestionamento das vias, obras, acidentes e estacionamento. Os veículos de transporte público podem ser equipados com rádios para comunicarem com routers sem fios, que comunicam uns com os outros para distribuir a informação em tempo real, como evidenciado na *Figura 2 (esquerda)* [6]. Além disso, as redes privadas usadas para serviços de segurança pública podem beneficiar da tecnologia em malha.

A aplicação comercial mais popular da tecnologia em malha é usar as WMNs como uma rede de acesso sem fios à Internet pública. A ideia neste caso é equipar casas em áreas escassamente povoadas, com routers que comunicam entre si por longas distâncias através de comunicações sem fios com múltiplos saltos, de modo a oferecer uma alternativa mais barata, flexível e robusta à rede com fios para acesso à Internet em banda larga nas áreas rurais (*Figura 2 (direita)*).

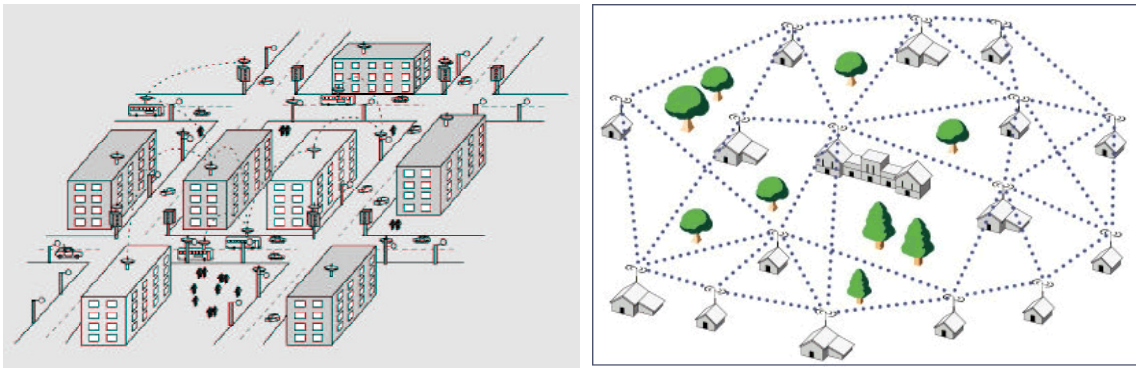
Outra aplicação importante refere-se a fornecer uma alternativa para a construção da rede de distribuição que liga as estações base WLAN (*Wireless Local Area Network*), dado que actualmente necessitam de estar ligadas entre si por ligações físicas.

## 2.2.3 Projectos com redes em malha sem fios

Há um interesse crescente na aplicação de tecnologias de redes em malha sem fios. Os exemplos mais típicos incluem produtos comerciais com o objectivo de oferecer acesso à Internet. Outro exemplo prático é o desenvolvimento de redes autónomas, como em casos de emergência/catástrofes onde as infra-estruturas foram destruídas. Por outro lado, as universidades estão a desenvolver redes de teste para o desenvolvimento de protocolos experimentais. Actualmente, os projectos em curso ainda só estão a usar a tecnologia 802.11[7], provavelmente devido ao elevado custo da tecnologia 802.16[8].

De seguida são apresentados os projectos principais de redes em malha.

O MIT Roofnet [9][10] é uma rede experimental em malha sem fios no MIT (*Massachusetts Institute of Technology* – Instituto de Tecnologia de Massachusetts) que



**Figura 2: Aplicações das redes em malha: sistemas de transporte inteligente (esquerda) [6] e redes de vizinhança em áreas rurais (direita) [117]**

possui 37 nós e cobre uma área de 4 Km<sup>2</sup>. Cada nó está equipado com um rádio simples e usa *hardware* 802.11b com uma antena omnidireccional. O protocolo de encaminhamento utiliza encaminhamento de origem com a métrica de custo de ligação ETT (*Estimated Transmission Time* – Estimativa de tempo de transmissão), que depende da probabilidade de retransmissão da ligação e da taxa realizada nessa ligação. O MIT Roofnet também possui um algoritmo para alterar a codificação e a taxa de transmissão de forma adaptativa.

Mesh@Purdue [11] é uma rede experimental em malha sem fios relativamente grande (32 nós). A rede utiliza antenas omnidireccionais e direccionais, com suporte para múltiplas interfaces de rádio [12]. O protocolo de encaminhamento é baseado no OLSR (*Optimized Link State Routing*) com a métrica de custo ETX (*Expected Transmission Count*).

O UCSB MeshNet [13] é uma rede experimental em malha sem fios na Universidade da Califórnia em Santa Barbara e possui 30 nós. Todos os nós possuem uma interface simples e usam o protocolo de encaminhamento AODV (*Ad hoc On-Demand Distance Vector*). A arquitectura MeshCluster [14] é uma extensão da arquitectura original MeshNet UCSB que permite ter múltiplas interfaces de rádio, e cujo protocolo de encaminhamento é modificado de modo a permitir a utilização de métricas de custo mais adequadas, tais como o ETT ou o ETX. São também propostos nesta extensão sistemas de gestão da mobilidade.

Hyacinth [15] é uma rede experimental em malha sem fios de menor escala (9 nós). No entanto, a arquitectura é bastante avançada, já que permite operar em cima de placas sem fios 802.11 tradicionais. Os nós da rede estão equipados com vários rádios. O protocolo de encaminhamento implementado resolve de forma distribuída a atribuição dos canais e o problema do balanceamento de carga através da utilização de informação de carga de tráfego a nível local.

Em [16] é apresentada uma rede experimental em malha sem fios para antenas direccionais com um único canal. O objectivo é construir uma rede sem fios auto-configurável, onde os nós se ligam entre si através de rádios 802.11 tradicionais equipados com antenas muito direccionais. Para o efeito, este projecto utiliza um esquema de endereçamento semelhante ao IPv6, em que a localização geográfica do nó está incluída no endereço.

Em [17] foram avaliados sistemas de gestão de mobilidade numa rede experimental em malha sem fios com 6 nós. São comparadas para o efeito duas abordagens. Numa abordagem é utilizado um protocolo de encaminhamento *ad hoc* tradicional, onde cada dispositivo cliente também tem um endereço IP e participa na construção da rota. A outra abordagem baseia-se numa solução onde os routers rastreiam a localização do cliente e encaminham os pacotes para um ponto de acesso.

Em [18] foi avaliada a qualidade do tráfego VoIP numa rede experimental em malha sem fios com 15 nós. O objectivo consistiu em investigar o impacto do uso de múltiplos canais/interfaces. Também foram estudados os benefícios da agregação de tráfego para combater a sobrecarga elevada gerada pelo protocolo sobre o tráfego VoIP.

Em [19] foi realizada uma simulação em larga escala, de uma área urbana realística com utilizadores móveis. O objectivo consistiu em investigar a viabilidade de uma rede 802.11e em malha sem fios baseada em padrões de mobilidade realísticos e condições realistas de propagação do sinal sem fios num mapa real de uma parte de Londres. Os resultados mostram que, para se fazer a cobertura adequada em ambientes urbanos, a densidade de routers 802.11 tem de ser necessariamente maior do que é usualmente previsto.

As redes em malha também estão a ser exploradas intensamente como um meio para fornecer redes que podem ser implementadas facilmente e que sejam muito robustas. Para isso a rede é projectada de modo a poder configurar-se automaticamente sem necessidade de infra-estruturas. Essas redes podem ser muito importantes em casos de emergência/calamidades/guerra onde a infra-estrutura tenha sido destruída, ou onde a mesma não exista. A CalMesh [20] e a RescueMesh [21] são duas redes em malha sem fios de emergência, em que as redes são principalmente usadas para o transporte de tráfego em tempo real, tanto de voz como de vídeo. A P3M Mesh Network [22] é uma rede em malha sem fios utilizada pelos militares dos Estados Unidos da América nas suas operações no terreno, que possibilita identificar em simultâneo a localização e o estado de cada soldado para coordenar as suas actividades sem intervenção de um mecanismo central.

## 2.3 Virtualização de redes

A Internet tem sido incrivelmente bem sucedida no mundo moderno, quer na forma como modela o acesso ao meio, quer na forma como troca informações. Ao longo das últimas três décadas, a arquitectura da Internet tem provado o seu valor através do suporte a uma multiplicidade de aplicações distribuídas, e do suporte a uma grande variedade de tecnologias de rede sobre a qual actualmente funciona.

No entanto, a sua popularidade tornou-se o maior impedimento para que pudesse continuar a crescer. Devido à sua natureza multi-fornecedor, a adopção de uma nova arquitectura ou a modificação das já existentes exige um consenso alargado entre todas as partes envolvidas, tornando a sua implementação praticamente impossível. Como resultado, as alterações na arquitectura da Internet são agora limitadas a actualizações simples, incrementais em vez de mudanças radicais com a introdução de novas tecnologias de rede [2]. Mesmo os defensores da arquitectura pura vêm a virtualização de redes como um meio para avaliar novas arquitecturas. De referir que a abordagem pluralista considera a virtualização como um atributo fundamental da própria arquitectura [2]. Segundo eles, a virtualização de redes pode enfraquecer a Internet actual e estimular a inovação e a diversidade, permitindo que múltiplas arquitecturas de redes heterogéneas partilhem a mesma camada física.

Em [23] foi proposta uma abordagem semelhante para virtualizar redes. Neste caso, o papel dos ISPs tradicionais foi dividido em dois: os fornecedores da infra-estrutura que gerem a infra-estrutura física, e os fornecedores de serviços que criam redes virtuais através da agregação de recursos de vários fornecedores de infra-estrutura, e que oferecem serviços fim-a-fim aos utilizadores finais. Tal ambiente irá promover a implementação de várias arquitecturas de rede heterogéneas coexistentes que não são restringidas pelas limitações inerentes existentes na Internet actual.

Em conclusão, ao permitir que múltiplas arquitecturas de redes heterogéneas, partilhem a mesma camada física, a virtualização de redes promove a diversidade, fornece flexibilidade, e promete maior segurança e gestão da rede.

### 2.3.1 Tipos de redes virtuais

As redes virtuais dividem-se em três tipos. Uma rede virtual privada, também conhecida como VPN, é uma rede virtual especializada que liga múltiplos sites distribuídos por meio de túneis através de redes públicas ou compartilhadas. Uma rede *overlay* é mais uma forma de virtualização de rede que normalmente é implementado na camada da aplicação, apesar de existirem várias implementações nas camadas mais baixas. Tem sido amplamente utilizado como uma ferramenta fraca, mas eficaz para

implantar novos recursos e correcções na Internet. Redes activas e programáveis, por outro lado, são um conceito que permite a personalização de elementos de rede com base em requisitos de fornecedores de serviços.

- ***Redes Virtuais Privadas***

Uma rede virtual privada (VPN) [24][25][26][26] pode ser pensada como uma rede de comunicações dedicada a uma ou mais empresas que estão distribuídas por vários sites ligados através de túneis sobre redes de comunicações compartilhadas ou publicas tal como a Internet. Se todos os sites numa VPN são propriedade da mesma empresa, a VPN é conhecida como uma Intranet corporativa. Por outro lado, se os sites são propriedade de diferentes empresas, a VPN é identificada como uma Extranet. Na prática, a maioria das VPNs são exemplos de intranets que ligam sites geograficamente distribuídos de grandes empresas corporativas.

Cada site VPN deve conter um ou mais dispositivos Customer Edge (CE) (por exemplo, terminais ou routers). Cada dispositivo CE está ligado, através de algum tipo de ligação do circuito, a um ou mais routers Provider Edge (PE). Routers na rede do fornecedor de serviço que não ligam aos dispositivos CE são conhecidos como “P” routers. Normalmente, uma VPN é gerida e fornecida por um fornecedor de serviços VPN (SP) e identificado como Provider-provisioned VPN (PPVPN) [27].

- ***Redes activas e programáveis***

A necessidade de criar, implantar e gerir novos serviços em tempo real, em resposta às procuras dos utilizadores foi o principal factor por trás do crescimento na pesquisa de redes programáveis. Para permitir serviços em tempo real, a separação das comunicações *hardware* do *software* de controlo é fundamental. Se a separação está em vigor, o *software* pode ser programado independentemente do *hardware* subjacente para entregar as funcionalidades necessárias. A comunidade de redes programáveis discute actualmente os passos a dar para que essa separação possa ser alcançada.

Os autores em [28] apresentam uma pesquisa de redes programáveis, juntamente com um modelo de rede programável generalizado, onde a programação é realizada através de mecanismos inseridos dentro da rede e por extensão da quantidade e alcance da mesma realizada em routers e *switches* existentes.

Colectivamente, os modelos de computação e comunicação formam uma rede programável, permitindo que um arquitecto de rede programe camadas individuais através do transporte, controlo e gestão de planos.

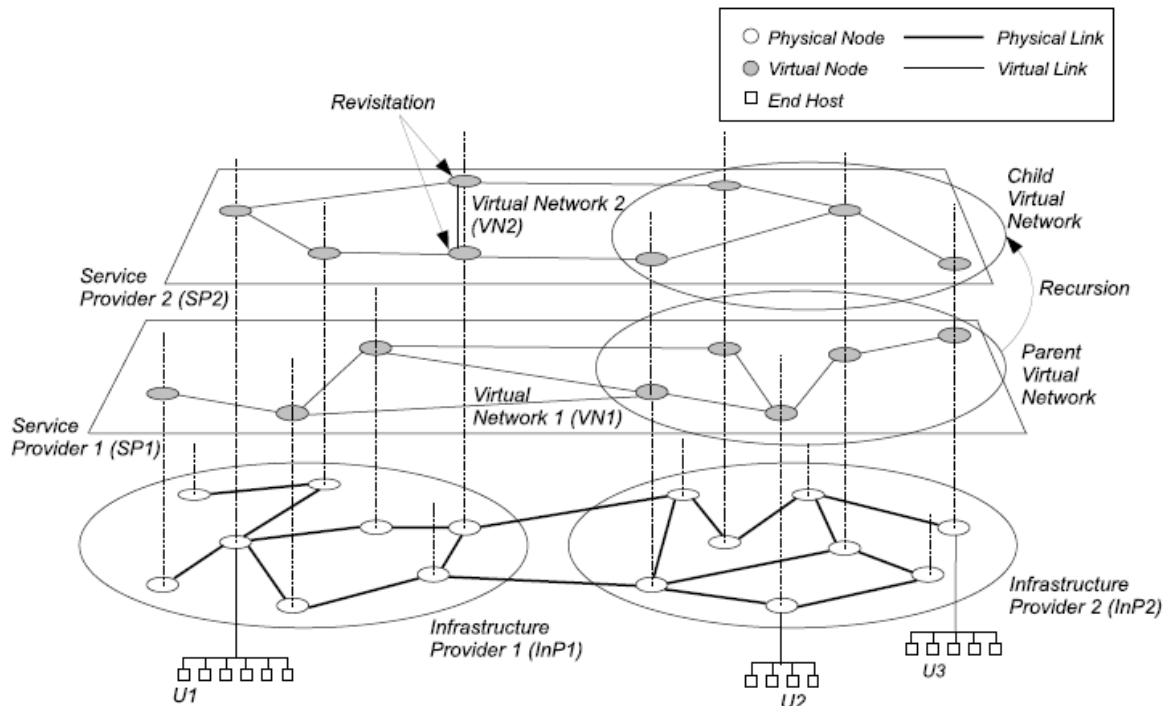
- **Redes overlay**

Uma rede *overlay* é uma rede virtual de computadores que cria uma topologia virtual em cima da topologia física de uma outra rede. Os nós numa rede *overlay* são ligados através de ligações virtuais que podem corresponder a um caminho, ligado por várias ligações físicas da rede subjacente. Os *overlays* não são geograficamente restritos e a participação é totalmente voluntária. Uma vez que os participantes emprestam voluntariamente os seus recursos à rede, os *overlays* não envolvem tipicamente gastos significativos. Além disso, são flexíveis e adaptáveis às mudanças e facilmente mobilizáveis, em comparação com qualquer outra rede.

### **2.3.2 Elementos da virtualização de redes**

A virtualização de redes é uma parte integrante da diversificada arquitectura da Internet, que suporta a coexistência de múltiplas arquitecturas de redes heterogéneas de diferentes fornecedores de serviços, partilhando uma camada física comum gerida por múltiplos fornecedores de infra-estrutura.

O modelo de virtualização de redes quebra com o tradicional modelo de redes. A principal diferença é a presença de dois fornecedores diferentes: os fornecedores de infra-estrutura e os fornecedores de serviços, que são representados por um único fornecedor: ISP no modelo convencional [23]. Do ponto de vista comercial, esta dissociação amortiza o elevado custo fixo em manutenção assegurado pela partilha de capital e das despesas operacionais em todos os múltiplos fornecedores de infra-estrutura. Os fornecedores de infra-estrutura implementam e gerem actualmente os recursos subjacentes à rede física num ambiente de virtualização, oferecendo os seus recursos através de interfaces programáveis aos fornecedores de serviços, que concedem os recursos facilmente a múltiplos fornecedores para criarem redes virtuais e implementarem protocolos personalizados através da programação dos recursos de rede alocados para oferecer serviços fim-a-fim aos utilizadores finais. Existem algumas entidades chamadas *brokers* que actuam como mediadores entre os fornecedores de infra-estrutura, os fornecedores de serviços e os utilizadores finais do mercado da virtualização de redes. Numa arquitectura de virtualização de redes, existem algumas notificações comuns que descrevem os diferentes aspectos, tais como: topologia física composta por ligações físicas, topologia virtual (ou topologia lógica) composta por nós virtuais (routers ou *hosts* virtuais) e ligações virtuais (que podem ser caracterizadas por parâmetros de QoS).



**Figura 3: Arquitectura de virtualização de redes [29]**

Na *Figura 3*, pode-se ver a topologia física composta por diferentes fornecedores de infra-estrutura (*Infrastructure Providers*), cada um possuindo diferentes nós e utilizadores (U). Na topologia virtual, pode-se ver que estes nós são virtualizados, através da divisão dos seus recursos pelos diferentes fornecedores de serviços (*Service Providers*), formando deste modo diferentes redes virtuais (*Virtual Networks*). Cada uma destas redes virtuais pode possuir nós e utilizadores de vários fornecedores de infra-estrutura.

As arquitecturas de virtualização possuem como aspectos fundamentais a compatibilidade, flexibilidade, capacidade de gestão, escalabilidade, isolamento, segurança, privacidade, programabilidade, heterogeneidade e facilidade de implementação.

### 2.3.3 Técnicas de virtualização

Para se adoptar novas ideias na Internet actual, existem algumas abordagens que começam a separar o *software* do *hardware*. Essas abordagens podem ser evidenciadas ao nível da virtualização de *switches*, da virtualização de routers ou da virtualização de interfaces sem fios.



- **Switches**

Ao nível da virtualização de *switches* surgiu a abordagem OpenFlow [30], que propõe um novo *switch* em que as suas principais funcionalidades se baseiam na programação de redes virtuais em *backbones* de campus universitários e em *wiring closets*. Esta abordagem possibilita a execução e controlo de protocolos experimentais com *switches* programáveis de baixo custo com portas de alta densidade, isolando o tráfego experimental do real. Um dos principais objectivos do OpenFlow é incentivar os fornecedores a adicionar o OpenFlow aos seus *switches*, não expondo o seu funcionamento interno. O OpenFlow permite que a tabela de fluxos do *switch* seja remotamente controlada. Há uma divisão entre os módulos de dados (tabela de fluxos) e do controlo, que comunicam entre si através de um canal seguro utilizando o protocolo OpenFlow.

Uma abordagem muito parecida com o OpenFlow é o NetFPGA [31], que propõe uma simples metodologia de design modular para rapidamente se criar protótipos de *hardware* de rede. É basicamente uma interface que traduz directamente o modo de processar pacotes num simples canal entre os módulos de *hardware*. Destina-se a fazer o design de *hardware* de redes mais reutilizáveis para professores e investigadores, e permitir a adição de novos módulos entre os módulos disponíveis dos designs NET-FPGA pré-construídos.

- **Routers**

Ao nível da virtualização de routers existem várias abordagens que lidam com o design dos routers virtuais.

Uma destas abordagens é o Click Modular Router [32] que permite a extensão e a configuração dos routers. Esta abordagem permite construir blocos que são módulos de processamento de pacotes, que implementam funções simples, tais como, classificação de pacotes, fila de espera, políticas de perda de pacotes, escalonador, e interfaces dos routers.

A abordagem VERA [33] propõe uma arquitectura que consiste numa mediação entre o *hardware* e as funcionalidades dos routers. Esta abordagem inclui uma camada de abstracção do router, que deve suportar as exigências dos routers IPv4, bem como novas funções de encaminhamento. VERA também inclui uma camada de abstracção de *hardware* que suporta a gama de *hardware* de interesse, expondo apenas detalhes de *hardware* suficientes para implementar eficientemente o router. No meio destas duas camadas de abstracção, trabalha um sistema operacional distribuído do router que faz a correspondência dessas abstracções de uma forma transparente e eficiente.

A abordagem XORP [34][35] baseia-se na abordagem modular Click para encaminhamento, onde cada processo de encaminhamento é composto por módulos de processamento, oferecendo uma forma flexível para controlar a transmissão através de routers programáveis. Cada protocolo de encaminhamento e função de gestão do encaminhamento é implementado num processo separado, adoptando um modelo orientado a objectos de *thread* simples, onde os eventos são gerados por temporizadores.

A abordagem Virtual Router Project [36][37][38][39][40] lida com a temática dos routers virtuais, deixando para trás a arquitectura convencional dos routers. Em [36] é avaliado o desempenho de um *software* do plano IP de encaminhamento do router dentro do ambiente XEN VMM[41], que é normalmente utilizado para a virtualização de servidores. Em [37], é explorada a ideia de ter um plano de encaminhamento reprogramável para routers virtuais, capaz de executar múltiplos encaminhamentos virtuais em paralelo. Em [38][39], é apresentada a avaliação de desempenho do *software* dos routers virtuais, usando o Click para o encaminhamento de pacotes. A avaliação mostra que o moderno *hardware* X86 pode constituir uma plataforma viável para suporte de alta performance com boa relação qualidade preço dos routers virtuais.

- **Interfaces sem fios**

Ao nível da virtualização de interfaces sem fios surgiu o projecto Madwifi [42], que é um conjunto de drivers de placas sem fios da Atheros para sistemas operativos Linux. Esta abordagem possibilita a criação de várias interfaces virtuais sem fios na mesma placa. Como é um *software* aberto, esta abordagem tem sido muito utilizada por investigadores em vários projectos [43] incluindo outras tecnologias como redes *ad hoc*, redes em malha, virtualização de redes, informação em tempo real, entre outras.

Uma outra abordagem de virtualização de interfaces é a MultiNet [44], cuja abordagem é actualmente desenvolvida pela Microsoft e que vem incluída no Windows 7. O MultiNet compartilha de forma transparente recursos de *hardware* por vários sistemas independentes, através de uma camada de *software* que abstrai o *hardware* da placa em vários adaptadores virtuais. O *software* faz o controlo das ligações de cada adaptador virtual, garantido que cada um receba uma “fatia” dos recursos do adaptador físico. Na prática não ocorrem ligações simultâneas em redes diferentes, pois o rádio transmissor/receptor não é capaz de executar essa tarefa, mas a alternância entre redes por divisão de tempo, torna o processo transparente para as aplicações e para o utilizador. A característica mais atraente desta tecnologia é a criação de redes *ad hoc* e redes em malha, em que numa rede em malha cada cliente actua também como um elemento de repetição, aumentando a área de cobertura da rede, à medida que novos clientes são conectados.

## 2.3.4 Projectos com redes virtuais

O termo “redes virtuais” tem sido usado de forma desmesurada por diferentes grupos de pesquisa para descreverem os seus trabalhos em redes virtuais privadas, redes *overlay* e redes activas e programáveis. Até há pouco tempo, poucas delas realmente seguiram a visão pluralista da virtualização de redes. Procedemos de seguida à realização de um resumo dos principais projectos de redes virtuais na *Tabela 1*. Para o efeito comparou-se as diferentes tecnologias através do seu objectivo principal, da tecnologia de rede usada, do tipo de camada de virtualização e do nível de virtualização fornecido por cada projecto. São também indicadas as referências para cada um dos projectos, em função das suas características próprias.

Projectos	Objectivo principal	Tecnologia de redes	Camada de virtualização	Nível de virtualização	Referências
<b>X-Bone</b>	Automatizar a implementação de <i>overlays</i> IP	IP	Aplicação	<b>Nó/ Ligação</b>	[45], [46]
<b>Tempest</b>	Permitir um controlo alternativo das arquitecturas	ATM	Ligação		[47], [48]
<b>UCLP</b>	Provisionamento dinâmico e reconfiguração de <i>lightpaths</i>	SONET/SDH	Física	<b>Ligação</b>	[49], [50], [51]
<b>VNET</b>	Máquina virtual de computação em grelha		Ligação	<b>Nó</b>	[52], [53]
<b>AGAVE</b>	Provisionamento de serviço fim-a-fim sensível ao QoS	IP	Rede		[54], [55], [56]
<b>VIOLIN</b>	Implementar a pedido serviços de valor acrescentado sobre <i>overlays</i> IP	IP	Aplicação	<b>Nó</b>	[57], [58]
<b>VNMRS</b>	Gestão de redes virtuais	ATM / IP		<b>Nó / Ligação</b>	[59], [60], [61]
<b>Darwin</b>	Integração de gestão de	IP			[62]

	recursos e serviços de valor acrescentado				
<b>NetScript</b>	Composição dinâmica de serviços	IP	Rede	<b>Nó</b>	[63], [64]
<b>Genesis</b>	Arquitecturas de redes <i>spawing</i> virtuais		Rede	<b>Nó / Ligação</b>	[65], [66], [67]
<b>PlanetLab</b>	Implementação e gestão de ambientes de testes baseados em <i>overlays</i>	IP	Aplicação	<b>Nó</b>	[68], [69]
<b>VINI</b>	Avaliar os protocolos e serviços num ambiente realista		Ligação		[70], [71]
<b>GENI</b>	Criação de redes virtuais personalizadas	Heterogénea			[72], [73]
<b>CABO</b>	Implementação de serviços fim-a-fim de valor acrescentado numa infraestrutura compartilhada	Heterogénea		<b>Completo</b>	[23]

**Tabela 1: Diferentes projectos de virtualização de redes [29]**

Hoje em dia, para além dos projectos apresentados, já se começam a desenvolver projectos de virtualização de redes sem fios, que exigem diferentes abordagens e desafios [74]. Embora já seja possível dividir uma rede sem fios, ainda subsistem problemas de interferências e partilha do meio físico no tempo, frequência e códigos, com impacto em diferentes tipos de aplicações. De seguida apresenta-se os projectos mais importantes em virtualização de redes sem fios.

Em [75] é apresentado o ORBIT que é um emulador de rede com uma malha de 400 nós sem fios com múltiplas interfaces sem fio por nó, onde os investigadores podem realizar e avaliar as suas próprias experiências.

Em [76], é validado e comparado o desempenho do UML contra a proposta OpenVZ em termos do isolamento entre as diferentes camadas, e do *overhead* na taxa de transferência e no atraso. O OpenVZ permite que um servidor físico execute múltiplas instâncias de sistema operacional conhecidas como um servidor virtual privado, e pode controlar outras características, tais como quotas do disco, memória, e unidades do CPU.

Em [77], é apresentado o WiSwitcher, que é um cliente sem fio capaz de se ligar a múltiplos pontos de acesso. Aumenta a estabilidade da percentagem de tempo atribuída aos escalonadores, mesmo se a estação transmite em modo de saturação, atinge uma alta taxa de transferência agregada sobre os pontos de acesso ligados, e transmite perfeitamente o tráfego TCP em cenários controlados.

## 2.4 Gestão de informação de contexto

O contexto é uma característica cada vez mais utilizada na Internet do futuro. De acordo com a definição de contexto em [78], o contexto é entendido como qualquer informação que pode ser utilizada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, lugar, ou objecto que é considerado relevante para a interacção entre utilizadores e aplicações.

Existem vários tipos de contexto com importância no desempenho da rede:

- Contexto de utilizador;
- Contexto de dispositivo;
- Contexto de aplicações/serviços;
- Contexto de rede;
- Contexto de controlo do fluxo;
- Contexto de mobilidade;
- Contexto de segurança;
- Contexto de preço;
- Contexto de previsão.

Os protocolos de redes e soluções de gestão podem tirar partido da informação de contexto de todas as entidades. A informação de contexto deve ser monitorizada de forma independente para construir redes com contexto específico, melhores que as que não têm contexto, de forma a fornecer soluções individualizadas em termos de latência, privacidade, estado de um dispositivo particular, preço, requisitos de QoS, mobilidade, entre outros.

### **2.4.1 Redes com informação de contexto**

A informação de contexto pode ser utilizada para construir redes lógicas (*overlays*) sobre a mesma rede física de forma independente. Estas redes devem por isso ser formadas com base em informações de contexto, ou seja, interligar utilizadores que partilhem os mesmos interesses e/ou que solicitem serviços semelhantes. Desta forma, estas redes necessitam de otimizar o transporte, assim como fornecer abordagens orientadas aos serviços. De seguida são apresentadas algumas soluções bibliográficas que pretendem fornecer serviços sensíveis ao respectivo contexto.

Têm sido propostos *overlays* sensíveis a contexto para ambientes de redes [79], onde o contexto é obtido a partir de sensores. Em [80][81], é proposta uma camada para a gestão da criação, destruição e adaptação das redes *overlay* sensíveis ao contexto, onde os *overlays* são criados com base no contexto do utilizador, do fornecedor de serviços, e do fornecedor de redes.

Em [82][83][84][85], é apresentada uma abordagem onde a gestão é feita sem intervenção humana para criar, configurar, adaptar, contextualizar e mudar redes *overlay* sensíveis a contexto, que permitem a composição e a entrega dos serviços aos utilizadores finais.

Em [86], o projecto C-CAST [87] apresenta uma arquitectura sensível ao contexto para sessões de múltiplos utilizadores e controlo da rede, que se adapta dinamicamente às mudanças de contexto e mantém a conectividade com os requisitos pretendidos. São considerados os conceitos de sessões dinâmicas e de redes controladas pelo contexto, sendo introduzido o conceito de árvores abstractas na rede para aumentar a estabilidade da rede a qualquer mudança de contexto.

Em [88], é apresentado o projecto NEON da Nokia Siemens, que fornece um ambiente com um operador de rede virtual para suportar serviços sensíveis ao contexto. O operador de rede virtual recolhe informações de contexto do utilizador, e suporta diferentes tipos de comunicações sobre redes heterogéneas multi-acesso baseadas em IP, abstraindo-se da complexidade da rede física, e oferecendo pacotes de serviço modulares.

### **2.4.2 Modelação de contexto**

Existem muitos tipos de informações de contexto, cada um com diferentes propriedades, o que leva a diferentes formas de os expressar e modelar. Até ao momento, cada sistema usa a sua própria maneira de modelar a informação de contexto, não permitindo que haja troca de informações entre diferentes sistemas, nem mesmo através das aplicações. De seguida são apresentadas algumas estruturas de dados

utilizadas para expressar e trocar informação de contexto, em vários sistemas sensíveis ao contexto [89]. Embora a maioria dos sistemas use estruturas de dados *ad hoc*, elas podem ser enquadradas em diferentes categorias.

Uma das categorias é a das etiquetas, onde o contexto é modelado como uma etiqueta com os seus respectivos campos. A Stick-e note é baseada no SGML (*Standart Generic Markup Language*) [90][91], e os seus campos podem recursivamente conter outras etiquetas e campos correspondentes. Este modelo evoluiu para ConteXtML [92], que contém um protocolo simples baseado em XML (linguagem de programação baseada em etiquetas) para a troca de informação de contexto entre um cliente móvel e um servidor.

Outra categoria promissora é a chave-valor, onde a informação de contexto é modelada num par chave-valor. Schilit [93] e Mobisaic [94], usam esta abordagem onde uma variável actua como chave e o valor da variável contém os dados do contexto actual. Esse modelo simples permite assegurar a correspondência entre os diferentes padrões considerados.

O Modelo baseado em lógica é outra categoria que se poderá evidenciar. O sistema multimédia orientado à localização [95] exprime as informações de contexto existentes numa base de dados centralizada utilizando um modelo de dados entidade-relação implementado em Prolog (é uma linguagem de programação baseada em lógica). Os dados de contexto são expressos como elementos de um sistema baseado em regras, tornando possível adicionar novas regras assim como de submeter pedidos à base de dados.

Uma outra categoria é o Modelo orientado a objectos. O sistema GUIDE [96][97] fornece um modelo de informação especialmente projectado para contexto de localização, onde a informação de contexto é incorporado como os estados do objecto, e o objecto fornece métodos para aceder e modificar os respectivos estados.

Por último, poder-se-á ainda considerar uma categoria baseada em camadas. O projecto TEA [98] modela os dados de contexto numa estrutura em camadas. A primeira camada é dos dados de contexto em bruto obtidos a partir dos sensores. A segunda camada gera valores simbólicos, de entre um conjunto de valores possíveis, a partir da primeira camada. A terceira camada obtém o contexto a partir das camadas inferiores, e é descrito como um conjunto de vectores bidireccionais, onde cada vector é composto por um valor simbólico que descreve as situações e um número que indica a probabilidade respectiva. Por fim, alguns scripts primitivos são fornecidos para a camada mais alta do aplicativo.

### 2.4.3 Descoberta de informação de contexto

Com a quantidade de informação de contexto disponível, é importante fornecer métodos eficientes para descobrir na rede o contexto apropriado a determinado serviço. Para o efeito existem as abordagens centralizadas e as abordagens descentralizadas.

Nas abordagens centralizadas, apenas um elemento da rede faz toda a gestão de contexto, o que provoca problemas de pontos de falha, de estrangulamento e de escalabilidade da rede. As abordagens descentralizadas permitem que a gestão da rede seja distribuída por todos os elementos da rede, ficando todos os elementos da rede responsáveis por gerir a rede. Desta forma não possui os problemas descritos anteriormente, reagindo mais facilmente a mudanças de contexto de utilizadores ou aplicações. De referir, todavia a desvantagem existente de gerar mais tráfego de controlo.

Uma das arquitecturas distribuídas que fornece mais características vantajosas para lidar com o contexto é a arquitectura baseada em P2P (*Peer-to-Peer*), que para além de ser distribuída é também auto-organizada. Neste tipo de arquitectura, a informação de contexto é distribuída na rede, e é descoberta através de um mecanismo distribuído que usa as DHTs (*Distributed Hash Tables*). A DHT é mantida através da cooperação de vários nós de uma forma descentralizada e autónoma [99].

Apresentam-se de seguida algumas destas abordagens. A abordagem CSON-D [100][101][102], propõe uma nova solução de disseminação de contexto baseada em P2P *overlay* para sistemas autónomos em ambientes heterogéneos e dinâmicos. Esta abordagem para além de suportar todos os tipos de contexto, implementa uma arquitectura específica para contexto, dinâmica e justa de base semântica para redes *overlay* hierárquicas e com múltiplos níveis. A abordagem descrita em [103][104], propõe uma estrutura P2P distribuída para facilitar a pesquisa dos nós, que se enquadram nos requisitos de contexto de um processo de descoberta. Nesta solução, a rede é dividida em grupos, onde cada grupo pertence a um determinado grupo semântico, de acordo com o maior conjunto de contexto que o caracteriza. Estes grupos semânticos são organizados numa estrutura em anel, que possibilita a existência de atalhos entre eles de forma a otimizar o algoritmo de busca. Este sistema permite ainda que os grupos semânticos se dividam ou se juntem mediante o número de chaves associadas.

### 2.4.4 Projectos de redes em malha sem fios com contexto

Para redes em malha sem fios existem muitos parâmetros de contexto que são importantes: os pontos de acesso e a sua disponibilidade, o número de *gateways* disponíveis, as condições da rede, as rotas e caminhos disponíveis, a quantidade de tráfego, e a mobilidade e previsão da mesma. Além destes, os utilizadores e as redes têm



muitos requisitos heterogéneos, que são parâmetros de contexto importantes, tais como, diferentes preferências, redes de acesso sem fios com largura de banda diferente, diferentes dispositivos com diferentes capacidades, que suportam diferentes aplicações com diferentes requisitos de QoS.

A investigação em redes em malha sem fios com contexto tem incidido principalmente em mecanismos de encaminhamento sensíveis à QoS. Em [105] é proposto um de muitos esquemas que consideram a qualidade de uma ligação de comunicação, através da integração em métricas de medidas de interferência do sinal.

Sobre os outros factores de contexto, tais como, robustez, estabilidade, mobilidade, segurança e preferências do utilizador, tem havido pouca investigação. No entanto, em [106] é proposta uma arquitectura para suportar comunicações em tempo real sobre WMN, que se adapta às condições da rede actual, mediante o histórico de dados da rede. Através de métricas de disponibilidade de largura de banda, atraso e falhas de rede, é possível determinar a robustez dos caminhos. Em [107] são relacionadas as métricas de encaminhamento, tais como a contagem de retransmissões, contagem de saltos, e interferência nas ligações, com dados de contexto, ou seja, prioridade de tráfego, nível de segurança, e mobilidade, através de regras configuradas manualmente. Em [108] é proposto um algoritmo de encaminhamento adaptativo, que reage com a intensidade da mobilidade no interior da WMN. Dependendo da mobilidade assumida, é aplicado um algoritmo de encaminhamento reactivo ou pró-activo. A posição actual dos clientes na malha é determinada pela atribuição de canais e algoritmos de selecção do router proposto em [109].

## 2.5 Conclusão

Neste capítulo foi apresentado um breve resumo do conceito das redes em malha sem fios, assim como as principais aplicações e projectos deste tipo de redes. Através deste estudo pode-se constatar que as redes em malha mostram um grande potencial para cumprir com os requisitos das redes de nova geração (NGNs – *Next Generation Networks*). Através do conceito da virtualização, estas redes podem-se tornar muito promissoras para o fornecimento de redes personalizadas baseadas em contexto. Para tal foi descrito este conceito denominado de virtualização, assim como exemplos práticos onde o mesmo já é aplicado.

Para efectuar a gestão das redes com base em informação de contexto, foram estudadas várias técnicas, dando-se ênfase às redes que se auto-configuram e se adaptam mediante as mudanças de contexto do utilizador ou aplicação, assim como aos mecanismos de descoberta distribuídos para encontrar na rede o contexto apropriado a determinado serviço ou aplicação.



### 3. Redes em malha sem fios baseadas em contexto

#### 3.1 Introdução

A solução avaliada nesta dissertação aborda a introdução de contexto em WMNs, dividindo as WMNs em várias VNs baseadas em contexto, utilizando para isso a virtualização e mecanismos de gestão de contexto distribuídos na rede. O facto de os utilizadores serem cada vez mais móveis e quererem ser diferenciados em termos do serviço, implica o desenvolvimento de mecanismos mais eficientes que providenciem os requisitos pedidos, dinamicamente. Este capítulo descreve em detalhe a solução avaliada.

Na secção 3.2 são descritos os conceitos gerais e os desafios da solução que se propõe. Finalmente, a secção 3.3 apresenta um resumo o capítulo.

#### 3.2 Arquitectura proposta e objectivos, desafios inerentes

Na *Figura 4* pode-se ver um primeiro esquema da solução proposta.

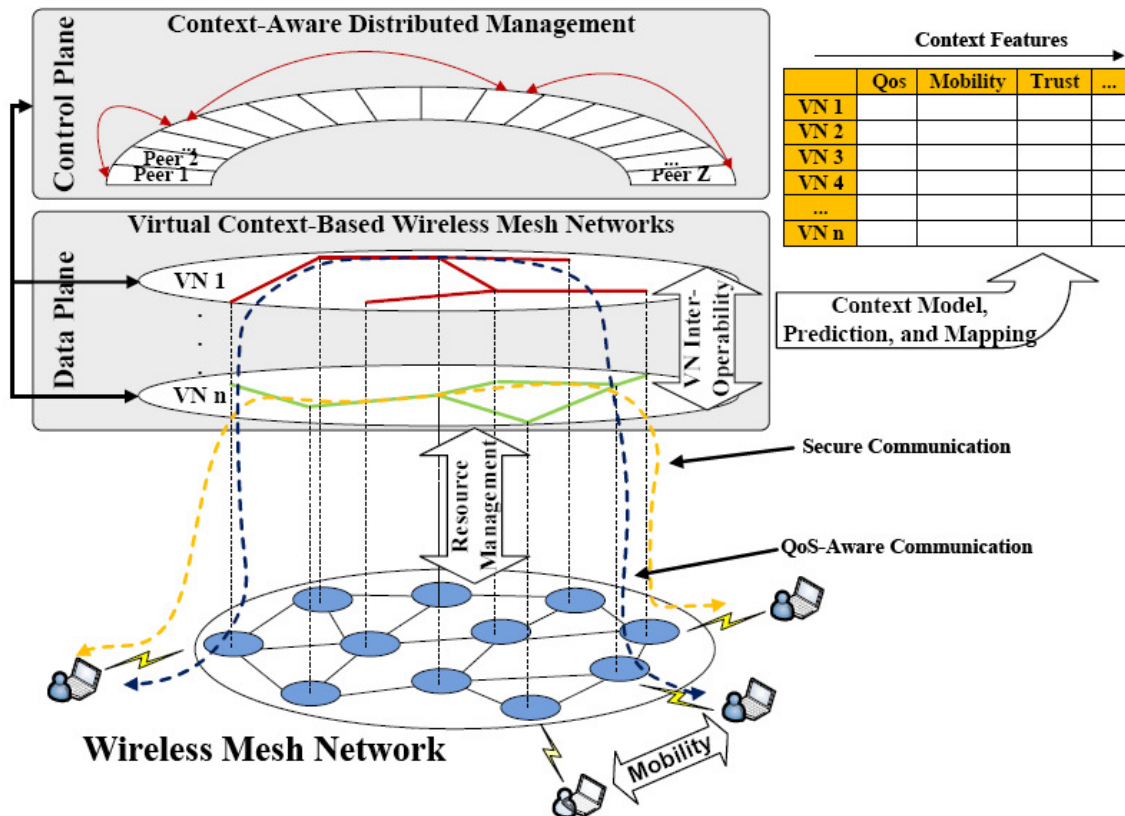


Figura 4: Solução para redes em malha sem fios baseadas em contexto [110]

A solução proposta baseia-se na divisão de uma rede física WMN num diferente número de redes lógicas (VNs). Cada uma dessas VNs é específica para certos níveis das características de contexto dos utilizadores e suas aplicações. O objectivo é o de fornecer redes personalizadas, para os utilizadores, dispositivos, ou aplicações, que melhor satisfaçam os requisitos ou expectativas pretendidas.

Para lidar com essas exigências, as WMNs já mostraram que têm muito potencial, visto que se auto-organizam à medida que a rede muda dinamicamente. As suas topologias, mecanismos e protocolos são flexíveis o suficiente para promover o estabelecimento e alteração de rotas para diferentes comunicações caracterizadas por diferentes parâmetros de contexto.

As várias dimensões disponíveis para a informação de contexto, e a constante mudança de tais informações nas WMNs, p. ex., preferências e localizações de clientes, e recursos exigidos pelas aplicações, destacaram a necessidade da implementação de vários mecanismos de controlo e gestão das VNs baseadas em contexto, tornando-as utilizáveis, adaptáveis e escaláveis.

### **3.2.1 Plano de dados**

Cada VN é caracterizada por um conjunto de nós e ligações, que compõem uma topologia adequada, para uma comunicação específica baseada em contexto: requisitos de segurança; mobilidade; custo; qualidade de serviço; etc. Os recursos atribuídos a esses nós e ligações, assim como os protocolos que correm em cada VN (encaminhamento reactivo/proactivo, métricas de encaminhamento, mecanismos de transporte UDP/TCP, etc.) devem ser apropriados ao contexto de cada rede. Portanto, a solução deve ser dotada com a programabilidade para criar e configurar as VNs, e adaptá-las, à medida que mudam os utilizadores e aplicações, e os diferentes níveis de contexto.

Para o suporte dos conceitos apresentados tem que ser considerada uma ferramenta que permita criar, configurar e adaptar as VNs. Além das bem conhecidas vantagens da virtualização de redes para diferentes áreas de pesquisa, torna-se necessário aceder a características particulares das WMNs, e avaliar se (e como) a virtualização de redes é vantajosa quando lhes é aplicada.

Visto que os clientes se movem constantemente de um nó para outro nó, ou mudam as suas preferências e expectativas, as VNs disponíveis têm de se estender dinamicamente para se incorporarem num nó físico específico. Portanto, essas VNs precisam de ser directamente mapeadas na topologia física, e não podem ser abstraídas dela, tal como acontece quando se aplica a virtualização de redes num ambiente de larga

escala. A programabilidade, o isolamento e a redução da interferência entre diferentes comunicações são ganhos que claramente podem ser alcançados através da aplicação da virtualização de redes em WMNs. Diferentes comunicações com requisitos de qualidade distintos podem ser melhor suportadas por uma VN isolada e dedicada, sem ter interferência de outras comunicações suportadas por outras VNs.

Porém, o *overhead* de controlo, a latência introduzida pelo mecanismo de virtualização de rede, e a falta de recursos em ambientes sem fios são vistos como potenciais obstáculos para operacionalizar as VNs baseadas em contexto. Neste caso, uma solução poderá ser uma emulação das VNs, ou seja, agrupar as VNs segundo o parâmetro de contexto mais prioritário, de modo que numa mesma interface (*overlay*) possam estar várias VNs com requisitos semelhantes, reduzindo o *overhead* e otimizando os recursos.

### 3.2.2 Plano de controlo

Como se está a lidar com uma solução baseada em informação de contexto, é necessário uma plataforma para monitorizar, adquirir e prever a informação de contexto, e quantificar essa informação em diferentes níveis que possam ser directamente mapeados em topologias e recursos adequados. Este trabalho vai-se concentrar em como lidar com a informação de contexto presente na rede, de forma a permitir o aparecimento de vários mecanismos de controlo e gestão orientados ao contexto. Além de outros, vai-se concentrar na gestão das topologias e recursos das VNs, nos mecanismos de correspondência entre VNs e utilizadores, e nos mecanismos de descoberta de VNs/nós apropriados a um utilizador.

- ***Mapeamento do contexto***

Para se conseguir fazer a correspondência entre as características dos parâmetros de contexto dos utilizadores e aplicações, descritas na *Tabela 2*, com as VNs com diferentes características de contexto disponíveis na rede, descritas na *Tabela 3*, decidiu-se mapear os parâmetros dos utilizadores e aplicações, assim como as características das VNs em diferentes níveis.

Para isso definiu-se que o número máximo de níveis para cada parâmetro de contexto seria 4. O nível 0 é sempre o melhor caso, e o 3 é o pior. Por exemplo, pode-se ter 3 padrões de segurança e ter 4 padrões de mobilidade. Por exemplo para uma VN caracterizada por possuir muita segurança, terá o nível 0 no parâmetro de contexto que caracteriza a segurança, assim como um utilizador ou serviço que deseje ter muita segurança, terá a caracterizá-lo no parâmetro de contexto relativo à segurança, o nível 0. O número de níveis é no entanto reconfigurável e depende do cenário de estudo.

Desta forma, quando se quiser atribuir uma VN a um utilizador ou serviço, é fácil através de uma simples comparação de todas as métricas de contexto, descobrir se a mesma existe ou não na rede.

	QoS	Segurança	Mobilidade	Confiança	...
Utilizador 1					
Utilizador 2					
...					
Utilizador k					

**Tabela 2: Caracterização de cada utilizador através de parâmetros de contexto**

	QoS	Segurança	Mobilidade	Confiança	...
VN 1					
VN 2					
VN 3					
...					
VN n					

**Tabela 3: Caracterização de uma VN através de parâmetros de contexto**

- ***Configuração/adaptação dos recursos/topologias***

A criação da topologia adequada para cada VN baseada em contexto, e a atribuição dos recursos a cada VN, de acordo com o seu contexto, são outros desafios desta solução. É crucial providenciar *thresholds* para uma adaptação dinâmica dos recursos e topologias de cada VN de acordo com as mudanças de contexto, tais como, requisitos de QoS das aplicações e preferências dos utilizadores. Esses *thresholds* devem ser configurados e adaptados de acordo com a variação dos padrões dos parâmetros de contexto considerados. Finalmente, as mudanças de contexto também podem desencadear a criação de outras VNs/nós. O compromisso entre o número de VNs existentes, o custo da sua criação, e a QoE oferecida aos utilizadores necessita de uma análise efectiva. Diversos mecanismos necessitam de estar presentes na rede:

- Seleção de VNs. Quando um utilizador chega à rede constituída por uma multiplicidade de VNs, a melhor deve ser escolhida para ligar o utilizador. Assim, a rede tem de conter informações das características de cada VN e comparar essas informações com as características de contexto do utilizador, para encontrar a VN mais adequada para cada utilizador. É necessário integrar a informação de contexto nos elementos de rede através de uma abordagem distribuída, e

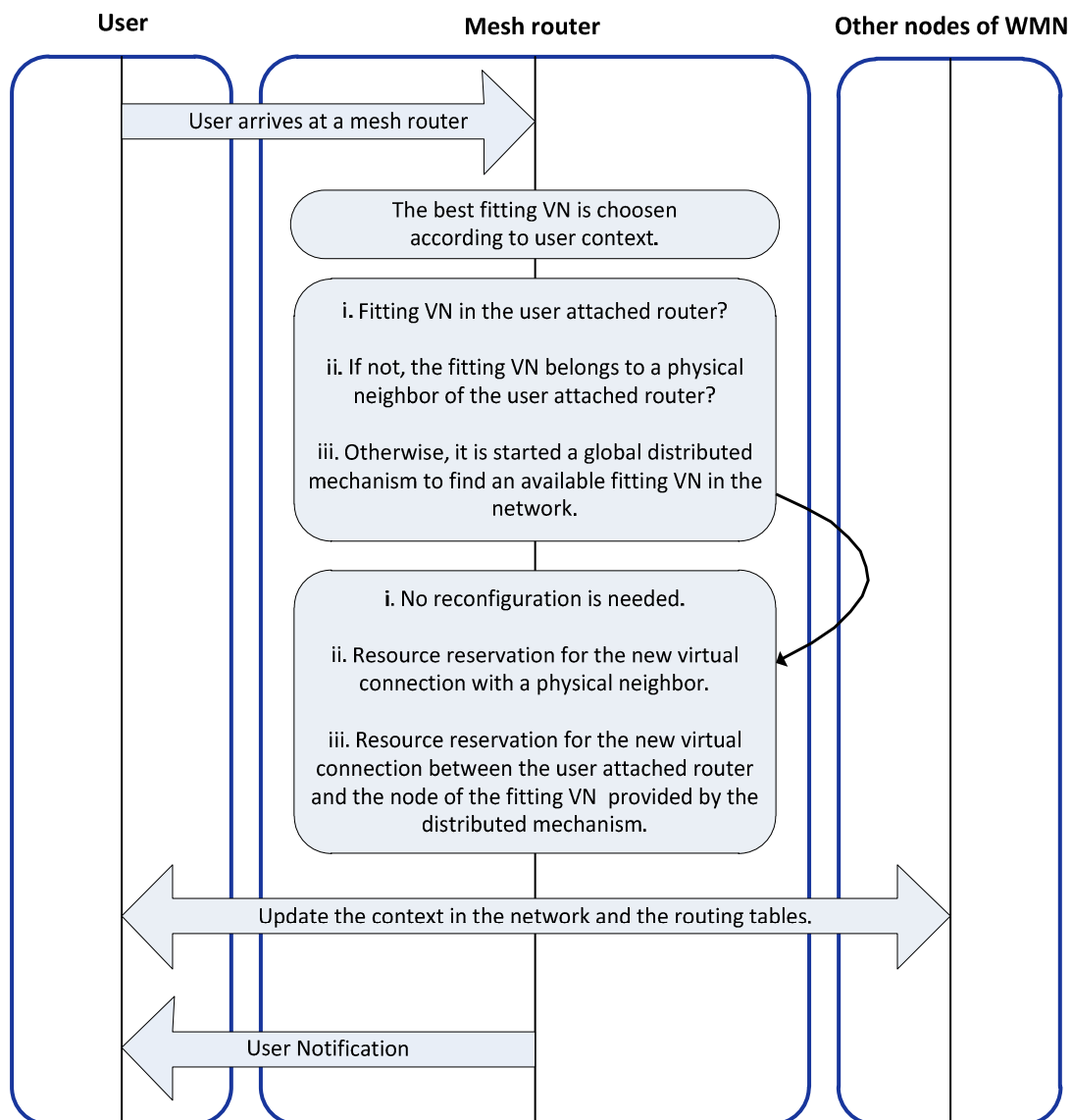
introduzir mecanismos para encontrar o nó mais próximo que contenha a VN pretendida.

- Criação de VNs. Se um utilizador exigir uma VN que não exista na rede, ou seja, todas as VNs disponíveis na rede apresentam características que não correspondem às exigidas pelo utilizador, pode ser escolhida a VN que se adapte melhor, ou criada uma nova VN com as características pretendidas. A decisão da criação de uma nova VN e o seu processo de criação deve considerar o contexto de todos os elementos de rede, assim como dos recursos de rede envolvidos.
- Reconfiguração de VNs. As VNs podem ser reconfiguradas de uma forma proactiva baseando-se no contexto, nas preferências e na gestão da mobilidade. Um caso possível de uma reconfiguração é o caso de um utilizador estar ligado a um router em malha que não possui a VN desejada pelo utilizador, mas existindo esta VN algures na rede. Este processo requer a extensão da VN existente de modo a integrar este nó na VN, através da criação de um nó virtual no nó físico (inclui a criação de nós virtuais com as características de contexto da VN, nos nós intermédios, que passam a fazer parte da VN).

- ***Mecanismos de descoberta***

A descoberta de VN/nós específicos que melhor se adequam ao utilizador ou aos requisitos da aplicação é um desafio importante que tem de ser tratado quando se lida com redes adequadas para requisitos específicos de contexto. Esses mecanismos de descoberta, para optimizarem o seu desempenho, devem ser divididos em dois tipos, um mecanismo local e um mecanismo global. O mecanismo local deve fazer a descoberta de VN/nós específicos que se encontrem na sua vizinhança. Caso não estejam VNs disponíveis na sua vizinhança, é desencadeado o mecanismo global de descoberta. Este mecanismo deve ser distribuído ou descentralizado, de modo a poder efectuar a descoberta mais rapidamente, para redes de média/elevada dimensão. Deste modo, o mecanismo local deve fornecer a informação da rede ao mecanismo global, de modo que quando o mecanismo global necessitar de efectuar uma descoberta de VN/nós específicos, contenha toda a informação da rede. Para se perceber melhor todas as hipóteses possíveis de descoberta de VNs específicas efectuadas pelos mecanismos mencionados, é apresentada a *Figura 5* que é descrita seguidamente.

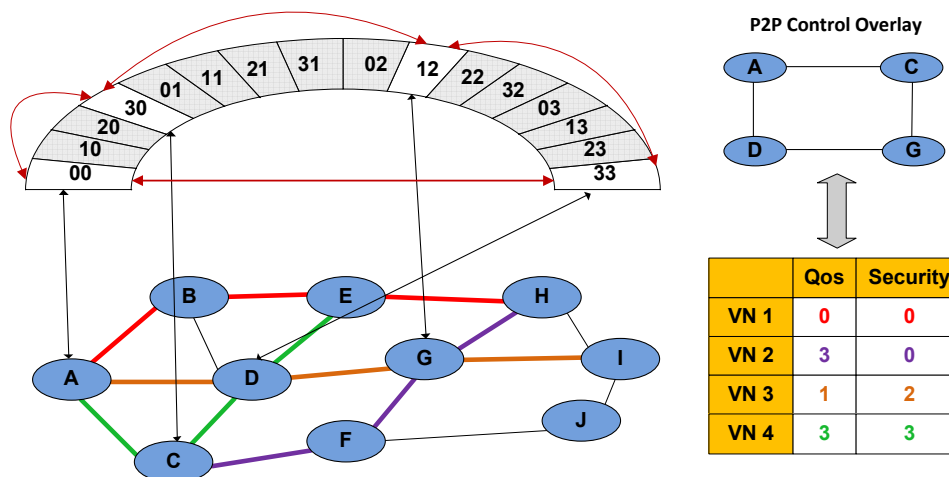
- i. Em primeiro lugar, é verificado se o nó ao qual o utilizador se ligou contém a VN pretendida. Se for verdade, não há necessidade de chamar nenhum mecanismo de descoberta e o algoritmo é interrompido. A VN simplesmente tem de efectuar uma ligação virtual entre o nó e o utilizador, de acordo com as características de contexto consideradas.



**Figura 5: Processo de associação de um utilizador à VN pretendida [4]**

- ii. Se o nó ao qual o utilizador se ligou não contém a VN pretendida, mas um dos seus vizinhos a contém, então o algoritmo é interrompido. A VN tem de ser reconfigurada e estender-se ao nó do utilizador, através da criação de um novo nó virtual no nó físico, da criação de uma ligação virtual entre os dois nós, e do novo nó virtual ao utilizador. Por último é necessário actualizar as rotas em conformidade.
- iii. Se o nó ao qual o utilizador se ligou não contém a VN pretendida, e nenhum dos seus vizinhos contém essa VN, um mecanismo de decisão distribuída necessita de encontrar na rede, a VN pretendida pelo utilizador. Isto pode ser realizado através de DHTs por exemplo. O tempo de associar um utilizador a uma VN





**Figura 6: Integração das características de contexto das VNs no mecanismo de descoberta distribuído [5]**

específica, e o *overhead*/desempenho do custo de reconfiguração das VNs, precisam de ser cuidadosamente avaliadas para serem tomadas as melhores decisões.

Uma reconfiguração da rede requer a actualização do contexto na rede e das tabelas de encaminhamento nas VNs. O encaminhamento dentro de cada VN pode ser um protocolo de encaminhamento tradicional para WMNs. As características da virtualização de redes podem também desempenhar um papel importante no controlo geral e no processo de gestão de modo a isolar o plano de controlo do plano de dados, uma vez que estes dois planos podem ser realizados por entidades diferentes.

- ***Integração do contexto no mecanismo de descoberta***

A abordagem mais tradicional para armazenar dados de contexto e resolver os pedidos de descoberta é a utilização de uma solução centralizada. Embora esta abordagem possa fornecer respostas rápidas para redes pequenas, tem limitações como a escalabilidade, pontos de falha, e estrangulamento. Por outro lado, as abordagens distribuídas com base nos conceitos P2P têm sido propostas para superar estes desafios. Tipicamente elas usam DHTs e mecanismos para dirigir um pedido de descoberta para os nós específicos, através de uma rede *overlay* estruturada. Para este efeito, é necessário o conhecimento das VNs e das suas principais características. A solução que foi utilizada para integrar o contexto através de DHTs, de maneira a facilitar a descoberta da VN mais adequada para um utilizador, pode ser vista na *Figura 6*, e é descrita seguidamente.

- Cada chave descreve uma VN, sendo cada algarismo da chave representativo de um parâmetro de contexto. Num exemplo simples com apenas dois parâmetros de

contexto, o algoritmo menos significativo pode representar a segurança e o algoritmo seguinte a QoS. Desta maneira a chave terá tantos algoritmos, quantos os parâmetros de contexto considerados.

- ii. Cada parâmetro de contexto pode ter diferentes níveis de contexto. Num exemplo simples com apenas 4 níveis de contexto tem-se o nível 0 como o melhor e o nível 3 como o pior.
- iii. Cada chave tem de estar associada a um dos nós pertencentes à VN que caracteriza. Como se pode ver no exemplo, a chave da VN1 está associada ao nó A, que é um nó que pertence à VN1; caso esse nó deixe de pertencer à VN1, a chave tem de mudar para um nó que faça parte da mesma VN.

Através deste processo é possível de forma eficiente e directa encontrar várias VNs que podem ser associadas a um utilizador. Por exemplo, se um utilizador chega a um nó que não contém a VN pretendida, e o mecanismo de descoberta local também não a encontrou na vizinhança, é necessário iniciar um procedimento de descoberta distribuído na rede, utilizando para isso o P2P *overlay* de controlo descrito anteriormente. Em primeiro lugar, uma chave tem de ser encontrada para inicializar o processo de descoberta, que pode residir no nó no qual o utilizador se ligou. Em seguida, é necessário mover-se ao longo do P2P *overlay* de controlo até encontrar a chave procurada. Finalmente, é necessário estender a VN entre o nó ao qual o utilizador está ligado e o nó mais próximo da VN escolhida. A rede é estendida através da criação de um novo nó virtual no nó físico ao qual o utilizador está ligado, da criação de uma ligação virtual entre os dois nós (os nós intermédios também necessitam de criar um novo nó virtual no nó físico e naturalmente passam a fazer parte da VN considerada) e do novo nó virtual ao utilizador.

### 3.3 Conclusão

Este capítulo descreve, conceptualmente, uma solução proposta para redes em malha sem fios baseada em contexto, em que através da virtualização, a rede física é dividida em várias VNs, cada uma delas possuindo níveis diferentes das características de contexto. Estas VNs são criadas e reconfiguradas através de vários mecanismos de controlo e de gestão, consoante a informação de contexto disponível e a mudança dessa informação ao longo do tempo, tornando assim essas VNs, adaptáveis e escaláveis. Este capítulo descreve também os mecanismos de controlo, os mecanismos de descoberta, essenciais para a correspondência entre as VNs disponíveis baseadas em contexto, e o conjunto dos requisitos de contexto que caracterizam um utilizador ou aplicação particular, ou para distribuir a informação adquirida num processo de decisão global, considerando um processo distribuído.

## **4. Implementação**

### **4.1 Introdução**

Com o intuito de avaliar a solução proposta, descrita no Capítulo 3, a mesma foi implementada no NS-2 (Network Simulator). Este capítulo apresenta o trabalho desenvolvido, descrevendo todas as alterações efectuadas ao simulador, assim como os mecanismos implementados para o controlo e gestão da rede baseada em contexto.

Na secção 4.2 é explicado o simulador de redes utilizado, NS-2.33, assim como as limitações do simulador e o trabalho de fundo necessário para iniciar o desenvolvimento da solução proposta.

Na secção 4.3 são explicadas todas as alterações efectuadas no NS, para que se possa avaliar a solução proposta. Esta secção é dividida em três subsecções: a primeira explica os módulos implementados ao nível do plano de dados; a segunda explica os módulos implementados ao nível do plano de controlo e a terceira e última explica a interligação de todos os módulos implementados, permitindo uma melhor compreensão da implementação total efectuada no simulador.

Finalmente, na secção 4.4, é resumido o trabalho desenvolvido nesta dissertação para avaliar a solução proposta de WMNs baseadas em contexto.

### **4.2 Network Simulator (NS 2.33)**

#### **4.2.1 Visão geral**

O Network Simulator – NS-2.33 é um simulador de eventos discretos, que tem como principais objectivos apoiar redes de investigação e educação, fornecendo ferramentas para concepção de protocolos e estudo de redes.

Outro objectivo deste simulador é fornecer um ambiente colaborativo disponibilizando o código-fonte, permitindo facilmente comparar protocolos desenvolvidos para aumentar a credibilidade dos resultados. O simulador tem como objectivo modelar protocolos de rede para cenários de redes com ou sem fios desde a camada física até à camada aplicacional. Para além disso, possibilita testar novas funcionalidades ao nível do controlo e gestão da rede.

O NS produz um ou mais ficheiros de texto que contêm dados da simulação detalhados. Os dados podem ser utilizados como entrada para uma ferramenta de simulação gráfica chamada Network Animator (NAM), ou para análise de simulações através de scripts (awk, perl, python) que permitem análises profundas de simulações.

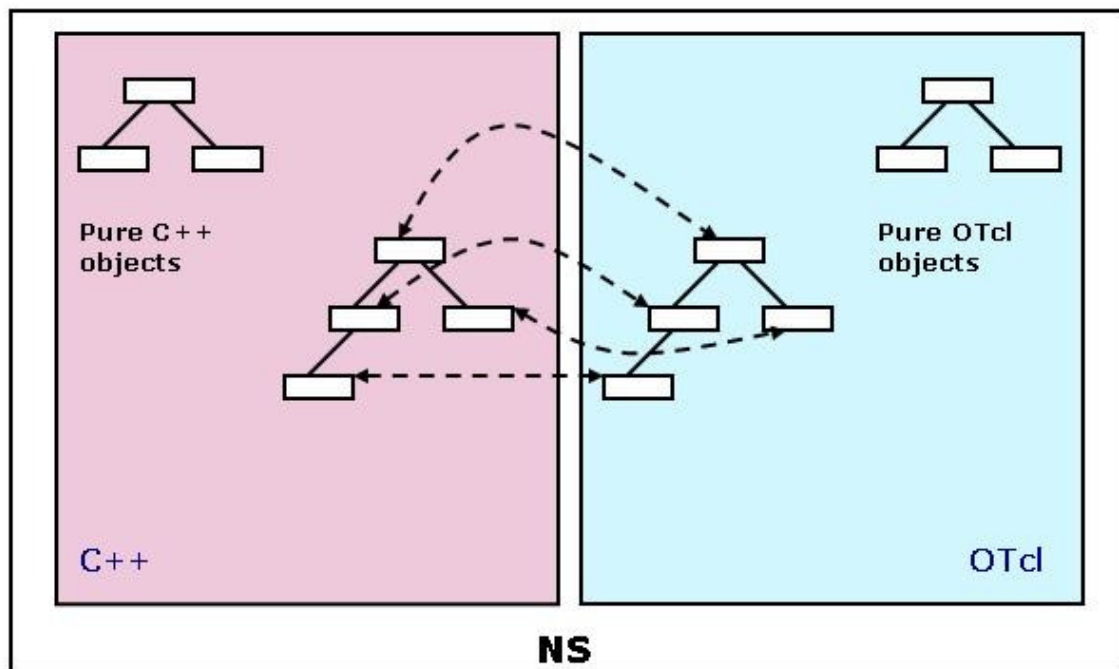
## 4.2.2 Arquitectura

O NS é um interpretador de script Tcl orientado a objectos (OTcl). Esta biblioteca contém objectos de escalonamento de eventos, objectos de componentes de rede e módulos de ajuda de configuração de rede.

Para criar e configurar um simulador de rede é necessário construir um script OTcl que inicia um escalonador de eventos, configura a topologia da rede utilizando as bibliotecas disponíveis e configura as fontes de tráfego.

Para reduzir o tempo de processamento de eventos e pacotes, o escalonador de eventos e os objectos que compõem a rede básica são escritos usando C++.

Os objectos compilados são disponibilizados para o interpretador de OTcl por uma ligação OTcl que cria uma correspondência do objecto OTcl para cada um dos objectos C++, tal como se pode ver na *Figura 7*.



**Figura 7: Correspondência entre OTcl e C++ [111]**

Esta ligação também permite que as funções de controlo e as variáveis de configuração especificadas pelo objecto C++ actuem como funções membro e variáveis configuráveis do objecto OTcl correspondente. Desta forma, os objectos C++ são controlados pelo OTcl.

Como o OTcl é um interpretador, não é um compilador, qualquer mudança num script OTcl não necessita de ser compilado. Por isso, não converte o código em linguagem máquina, e cada linha de código necessita de mais tempo de execução. Isto

faz com que o OTcl seja mais lento para correr, mas mais rápido para mudar do que o C++. Da mesma forma, C++ é mais lento para mudar, porque requer a compilação do código.

### **4.2.3 Implementação do conceito proposto no simulador**

A versão do NS usada (2.33), assim como a mais recente (2.34), possui sérias limitações à solução proposta anteriormente no capítulo 3.

Ao nível da virtualização, necessária para fornecer redes personalizadas para utilizadores e serviços, o NS apesar de possuir vários canais diferentes, só permite uma interface, facto que impossibilita a virtualização e como tal assume-se como uma séria limitação efectiva. Ao nível da qualidade de serviço em ambientes sem fios, necessário para a discriminação de VNs na atribuição de recursos, o NS não possui o protocolo 802.11e, embora existam algumas extensões para versões anteriores que podem ser adaptadas.

Ao nível das características de contexto, p. ex. requisitos da largura de banda das aplicações ou preferências dos utilizadores, não existe qualquer elemento implementado no NS, pelo que teve de ser desenvolvida uma solução que integrasse diferentes níveis das características dos parâmetros de contexto orientado ao utilizador ou aplicações subjacentes.

Ao nível da configuração e adaptação dos recursos e topologias das VNs mediante a variação do contexto, necessário para gerir a rede de uma forma dinâmica, não existe implementação disponível no NS, pelo que teve de ser desenvolvido um mecanismo que, mediante alterações nas características de contexto dos utilizadores e aplicações, reconfigurasse a VN ou até mudasse a VN associada ao utilizador ou aplicação. Este mecanismo necessita de descobrir a VN mais apropriada ao utilizador ou aplicação. Para o efeito, foram implementados mecanismos de descoberta: um de descoberta local e outro global.

Ao nível do mecanismo de descoberta local, necessário para descobrir uma VN específica na vizinhança, este teve que ser implementado de raiz.

Ao nível do mecanismo de descoberta global descentralizado, necessário para descobrir uma VN específica na rede, existem algumas extensões para versões anteriores que podem ser adaptadas.

## 4.3 Detalhes de implementação

Para avaliar a solução explicada anteriormente, esta secção irá descrever a implementação no NS-2 de alguns componentes da solução proposta a qual se encontra dividida em 2 secções: plano de dados e plano de controlo.

### 4.3.1 Plano de dados

Para implementar as funcionalidades descritas no Capítulo 3, foi necessário implementar a virtualização e a atribuição de recursos apropriados. Como o NS não possui suporte à virtualização a mesma teve de ser implementada com base numa extensão existente que permita múltiplas interfaces. Em relação à atribuição dos recursos apropriados, foi adaptada uma extensão para uma versão anterior do NS, a qual possibilita a introdução da Qualidade de serviço, e a implementação da diferenciação da largura de banda para cada interface. As subsecções seguintes descrevem em pormenor as extensões usadas e as alterações efectuadas.

#### 4.3.1.1 Suporte de virtualização de redes

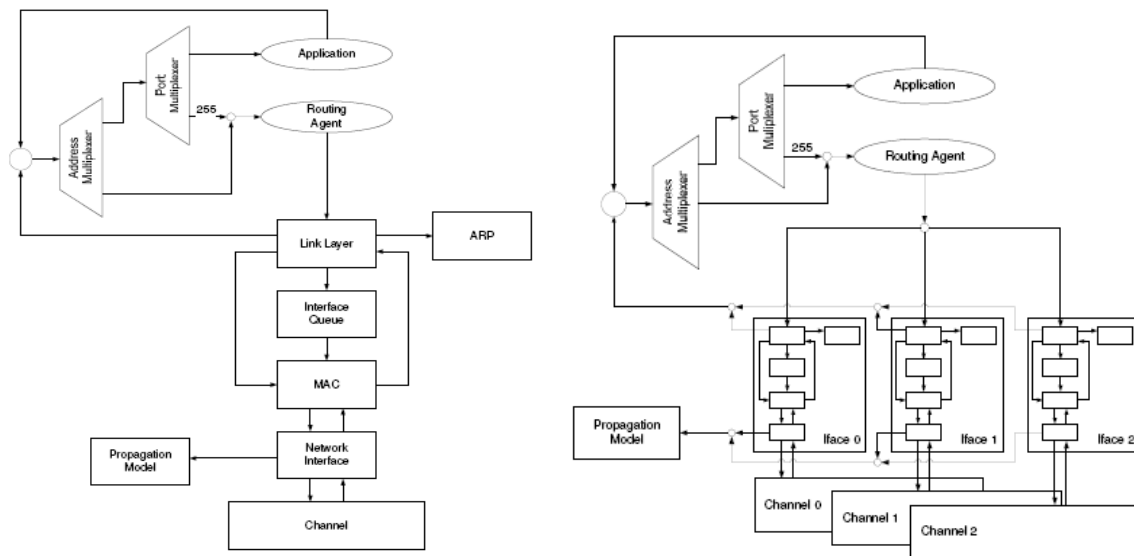
##### ***Descrição da extensão já existente***

Para implementar a virtualização de redes foi utilizada a informação do documento “Adding Multiple Interface Support in NS-2” [112], o qual mostra de uma forma intuitiva como extender o NS-2 para adicionar multiplas interfaces, de modo a resolver uma solução genérica. Não sendo completo, obriga no entanto a que seja necessario efectuar modificações para problemas mais especificos.

As principais características são:

- O número de canais num cenário particular é modificável;
- O número de interfaces por nó é variável, e não necessita de ser o mesmo para todos os nós num único cenário;
- Cada nó do mesmo cenário pode-se ligar a um número diferente de canais (dos que tinham sido previamente definidos);
- Os agentes de encaminhamento tiram vantagens do modelo modificado, mas a operacionalidade do simulador é preservada, para permitir compatibilidade.

Ao nível do NS-2 pode-se ver na *Figura 8* à esquerda, a arquitectura original do nó móvel, que consiste, abaixo do “*Routing Agent*”, por uma série de módulos que emulam as pilhas dos diferentes protocolos que existem na vida real em qualquer *host*:



**Figura 8: Arquitectura do nó móvel original à esquerda. Arquitectura modificada do nó móvel, com suporte a múltiplas interfaces, à direita [112]**

“Link Layer”, “MAC”, “ARP”, “Interface Queue”, “Network Interface”, todos eles ligados ao mesmo canal partilhado. Além disso o “Propagation Model” é usado para simular os efeitos dos canais reais no sinal transmitido, mais especificamente, a *propagation loss* é modelada, tanto de forma determinística como de forma aleatória.

Pode-se agora ver na *Figura 8* à direita, ao nível do NS-2, a arquitectura modificada do nó móvel, em que cada nó tem tantas cópias da série de módulos original (mostrada na *Figura 8* à esquerda), como interfaces. Além disso, o único módulo que não se repete é o “Propagation Model”, uma vez que em redes IEEE 802.11 pode-se utilizar mais de um canal ao mesmo tempo.

Para o tráfego de entrada não há muitas diferenças para o funcionamento original do simulador. Os pacotes de entrada chegam através do canal correspondente e percorrem os diferentes módulos por ordem crescente até ao último módulo de cada interface, o “Link Layer”, que está ligado ao mesmo ponto comum (o “Address Multiplexer”, sendo que todos os pacotes são tratados pelo agente apropriado) independentemente da interface pelo qual chegou.

Para o tráfego de saída, é de destacar que a inteligência de escolher a interface apropriada precisa de ser dentro do agente de encaminhamento.

### **Funcionalidades Implementadas**

Como dito anteriormente, as múltiplas interfaces estão preparadas para resolver um problema genérico, mas necessita-se de resolver um problema específico chamado de virtualização, que possui os seguintes requisitos adicionais:

- Total Independência entre interfaces;
- Redução de interferência entre diferentes comunicações com contexto.

Pode parecer que com as modificações apresentadas anteriormente já existe independência entre as diferentes interfaces, devido a termos uma série de módulos independentes para cada interface. Mas o módulo do “*Routing Agent*” continua a ser apenas um, de modo que quando se estabelece a ligação entre dois nós, apenas vai ser usada uma ligação durante toda a simulação (excepto se houver perda de rota) que usa apenas uma das interfaces, ficando as outras interfaces incomunicáveis.

O que se pretende é, durante toda a simulação, poder ter várias ligações entre dois nós, cada uma delas associada a uma interface diferente, de modo a se poder usar todas as interfaces simultaneamente.

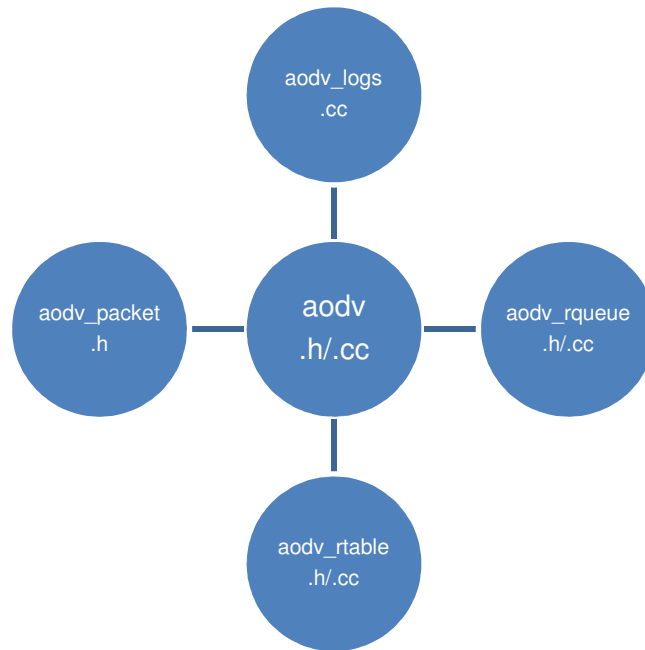
O módulo do “*Routing Agent*” que se alterou foi o AODV, pois já era o que possuía algumas alterações para suportar as múltiplas interfaces.

A arquitectura simplificada do AODV está representada na *Figura 9*, onde se pode ver os ficheiros presentes no NS-2: o ficheiro `aodv_logs.cc` é o ficheiro onde, tal como o nome indica, se fazem os logs; o ficheiro `aodv_packet.h` é onde se encontram definidos os cabeçalhos dos pacotes aodv; o ficheiro `aodv_rqueue.h/cc` é onde se encontra definida uma fila de espera onde são armazenados os pacotes quando não existe rota para o destino pretendido; o ficheiro `aodv_rtable.h/cc` é onde se encontram definidas as rotas, assim como uma lista de vizinhos e uma lista de precursores; no ficheiro `aodv.h/cc` é onde se encontra definido todo o funcionamento do protocolo, assim como as funções de recepção de mensagens, envio de mensagens, encaminhamento de mensagens, actualização de rotas e uma lista de broadcastID.

Sendo assim, alterou-se no `aodv_rtable.h/cc` as tabelas de encaminhamento de modo a possuírem para o mesmo destino várias rotas, uma rota por cada interface, assim como a lista de vizinhos e a lista de precursores. Alterou-se também, no `aodv_rqueue.h/cc`, o mecanismo da fila de espera de modo a passar a haver uma fila de espera para cada interface. Mediante estas alterações tiveram que ser alteradas quase todas as funções em termos de parâmetros de entrada (adicionar a interface), tanto nestes dois ficheiros como no `aodv.cc`.

No `aodv.cc` adicionou-se a capacidade de, mediante o identificador do fluxo e/ou o tipo de pacote, escolher a interface apropriada nas funções `rt_resolve`, `rt_ll_failed` e `forward` para os pacotes *broadcast*. Alterou-se o identificador do fluxo dos pacotes AODV para a interface pelo qual são enviados para, aquando da recepção, serem tratados na interface correcta; alterou-se também o *broadcast* dos pacotes AODV para passar a enviar só pela interface respectiva (excepção feita aos pacotes Hello que vão para todas as interfaces, mas como estão desactivados, este processo não foi modificado).





**Figura 9: Arquitectura simplificada do AODV no NS-2**

Para se reduzir a interferência entre comunicações diferentes com contexto, optou-se por dividir a rede física em várias VNs (VNs) isoladas e dedicadas a comunicações com confiança específica e requisitos de qualidade, sem interferências com outros requisitos de comunicações suportadas por outras VNs.

#### **4.3.1.2 Atribuição de recursos apropriados**

##### ***Descrição da extensão já existente***

Para implementar a atribuição de recursos apropriados, primeiro foi implementada a qualidade de serviço (QoS) no NS-2 através da instalação de um patch do 802.11e EDCA (*Enhanced Distribution Coordinate Access – Acesso aprimorado de Coordenação Distribuída*) [113][114][115] para uma versão anterior do NS e modificado por forma a funcionar com a versão do NS usada.

Foi escolhido este modelo visto que possui boa documentação, eficiência de funcionamento comprovado, grande quantidade de trabalhos científicos publicados (que referenciam o modelo), e pelo facto de implementar a versão final do 802.11e aprovado pelo IEEE.

Este modelo possibilita o escalonamento de eventos entre as filas da camada MAC, implementa o CFB (*Contention Free Burst*) que permite o envio de tramas em rajadas separados pelo intervalo SIFS (*Short Interframe Space*) durante um período livre de contenção defenido.

Ao modelo 802.11 do NS-2 foram introduzidas quatro filas na camada MAC, cada uma caracterizada por uma prioridade. Esta prioridade é especificada pelo campo *prio* do cabeçalho IP, em que zero representa a prioridade maior e três a prioridade menor. A técnica de descarte de pacotes usada é a *drop-tail*. Quando chega um pacote de encaminhamento, esse é encaminhado para o início da fila de maior prioridade.

### **Funcionalidades Implementadas**

Para a solução proposta anteriormente são necessários os seguintes requisitos adicionais:

- Largura de banda diferente para cada interface;
- Diferenciação do parâmetro de contexto associado à largura de banda.

Para se poder diferenciar comunicações com base no parâmetro de contexto denominado de largura de banda, é necessário que cada VN possa ter diferentes valores da largura de banda atribuída.

Para o efeito, alterou-se no ficheiro `mac_802_11e.h/.cc` a variável `datarate_` para um array de modo a se poder ter um `datarate` diferente para cada interface. Adicionou-se na função `sendData` a capacidade de, mediante o identificador do fluxo e/ou o tipo de pacote, escolher o `datarate` apropriado, à semelhança do que se faz no encaminhamento já explicado anteriormente. Os valores da largura de banda para cada interface (o `datarate_`) são fornecidos através do script TCL.

Os parâmetros das filas podem ser alterados internamente, mas dado a eficiência de resultados já acima mencionada, não foram alterados os parâmetros, usando-se o campo *prio* no script para diferenciar o tráfego em termos do parâmetro de contexto denominado por QoS.

### **4.3.2 Plano de controlo**

Nesta secção, tal como descrito no Capítulo 3, foi necessário implementar o mapeamento da informação de contexto, a configuração/adaptação dos recursos/topologias e os mecanismos de descoberta. Ao nível das características de contexto não existe implementação prévia no NS, pelo que se teve de implementar uma forma de o integrar e mapear na rede. Em relação à configuração e adaptação dos recursos e topologias das VNs mediante a variação do contexto, este também teve de ser implementado de raiz. Para os mecanismos de descoberta foi implementado um mecanismo local e um global, sendo que o local devido à sua especificidade foi desenvolvido de raiz, de modo a poder descobrir a VN nos seus vizinhos. O mecanismo de descoberta global foi adaptado a partir de uma extensão para uma versão anterior do NS, o qual possibilita a descoberta de uma forma distribuída (através de DHTs), e foi

implementado o mapeamento do contexto na chave, a adequação à reconfiguração/ extensão da rede, a inserção da chave num nó específico e a adequação ao tamanho da rede. As subsecções seguintes descrevem em pormenor as implementações realizadas, as extensões usadas e as alterações efectuadas às extensões.

#### **4.3.2.1 Mapeamento da informação de contexto**

Para se conseguir fazer a correspondência entre as características dos parâmetros de contexto dos utilizadores e aplicações com as VNs disponíveis na rede, decidiu-se mapear os parâmetros dos utilizadores e aplicações, assim como das VNs em níveis diferentes.

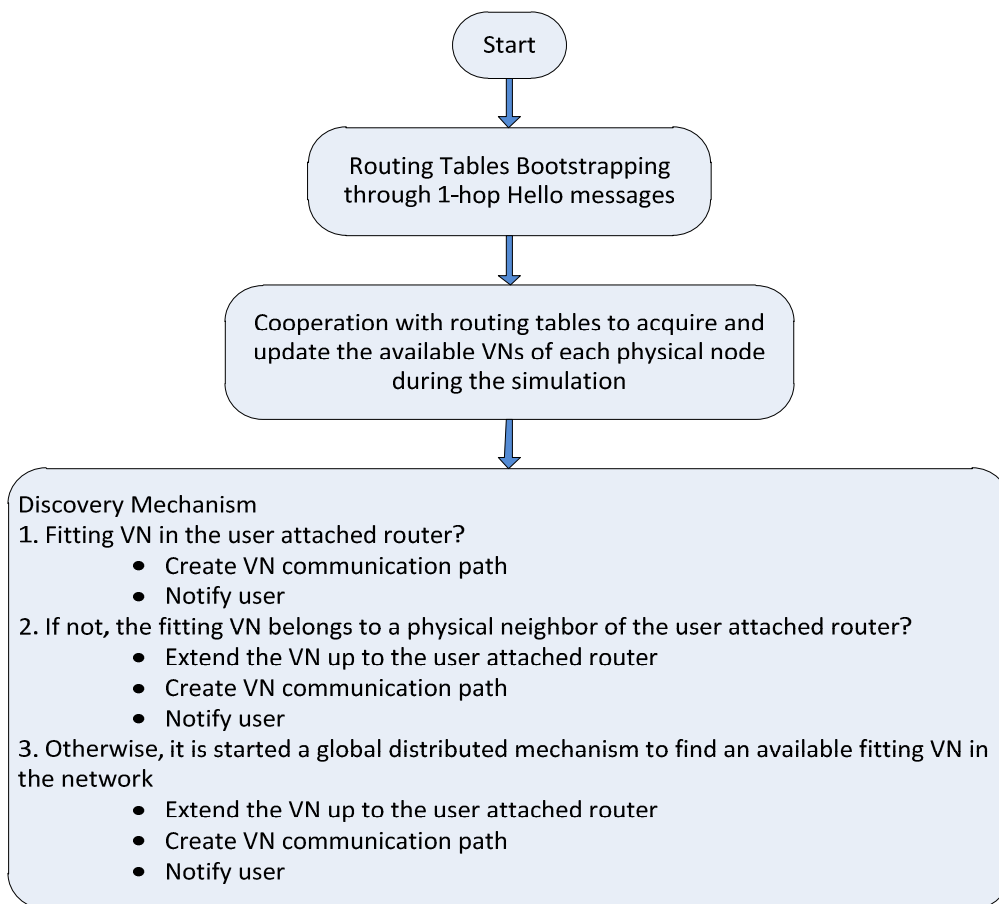
Para isso, definiu-se um número variável de níveis para cada parâmetro de contexto. No entanto, nesta implementação apenas se utilizaram três níveis para cada parâmetro de contexto, em que 0 é o melhor caso e 2 é o pior. Por exemplo, uma VN caracterizada por possuir muito pouco atraso terá o nível 0 no parâmetro de contexto que caracteriza o atraso, assim como um utilizador ou serviço que deseje ter um atraso muito baixo (comunicações de voz, por exemplo), terá a caracterizá-lo no parâmetro de contexto relativo ao atraso, o nível 0.

Desta forma, quando se quiser atribuir uma VN a um utilizador ou serviço, é fácil através de uma simples comparação de todas as métricas de contexto, descobrir se a mesma existe ou não na rede.

Todo o mapeamento descrito acima foi realizado no script TCL, excepto o mecanismo de descoberta que será explicado posteriormente.

#### **4.3.2.2 Configuração/adaptação dos recursos/topologias**

O facto de se atribuir ao utilizador ou aplicação uma determinada VN, esta não pode ser definitiva, ou seja, o utilizador ou aplicação pode de um momento para o outro alterar as suas características de contexto. Quando isto acontece é necessário recomençar o mecanismo de descoberta da VN que melhor se adequa aos novos parâmetros de contexto. Seguidamente, caso o nó a que o utilizador está ligado não possua essa VN, é necessário efectuar a extensão da VN para passar a conter esse nó, assim como a actualização do encaminhamento para passar a conter essa rota. Um exemplo, que mesmo sem mudar o nível de contexto, utiliza frequentemente o mecanismo descrito acima é a mobilidade. Cada vez que um utilizador ao mover-se, muda a sua ligação de nó, é necessário recomençar todo o mecanismo de descoberta e extensão da rede de novo, assim como a actualização do encaminhamento caso a rota não esteja ainda definida.



**Figura 10: Princípio de funcionamento do mecanismo de descoberta local**

O mecanismo descrito acima foi implementado no script TCL, sendo que o mecanismo de descoberta é despoletado no script TCL, o qual retorna a melhor VN (e o nó a que se vai ligar), e a extensão da rede é despoletada no script TCL caso o nó não pertença à VN. Sendo ainda de realçar que a actualização do encaminhamento é efectuado pelo AODV, de forma dinâmica, sempre que tal for necessário, sem intervenção do script TCL.

### 4.3.2.3 Mecanismos de descoberta

Para implementar o mecanismo de descoberta de VNs/nós específicos que melhor se adequem ao utilizador ou aos requisitos da aplicação, foi decidido dividir-se este mecanismo em dois, ou seja, um local denominado *Discover VN* e um global distribuído na rede denominado por *Bamboo* [116][117].

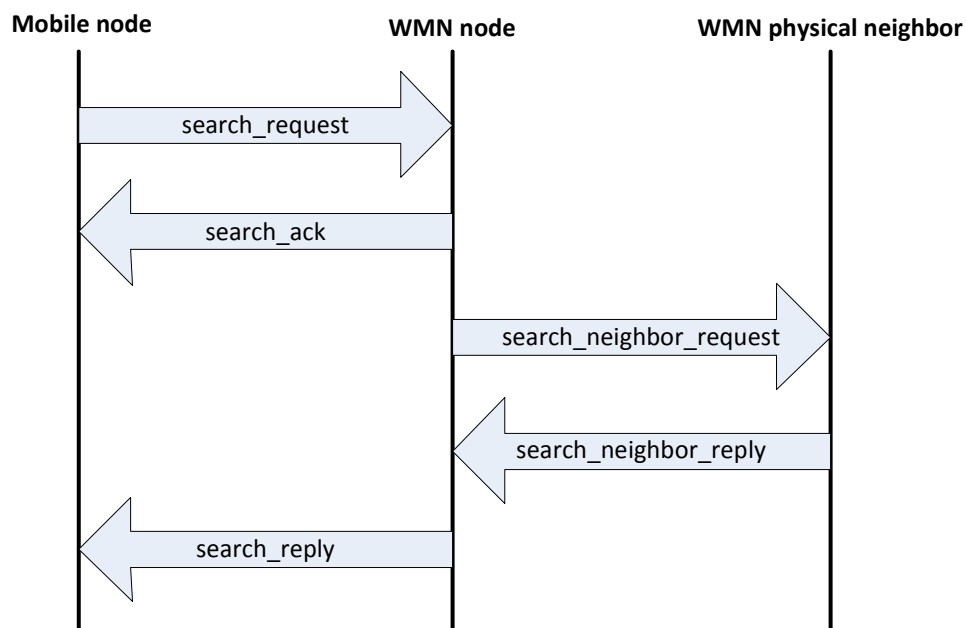
- **Descoberta Local**

Este mecanismo de descoberta local baseia-se na procura, entre as VNs disponíveis localmente (no próprio nó e nos nós vizinhos) baseadas em contexto, de uma VN que seja adequada ao conjunto de requisitos de contexto que caracterizam um utilizador ou aplicação particular.

Este mecanismo trabalha numa interface de controlo separada do restante tráfego da rede, e separada do tráfego de controlo do *Bamboo*, para não haver interferência com as restantes comunicações presentes na rede.

O seu princípio de funcionamento está explicado na *Figura 10*.

Para se implementar o mecanismo de busca, foi criado um protocolo chamado *Discover VN*. Este protocolo é responsável por toda a troca de mensagens necessárias ao funcionamento do mecanismo de busca ao nível local. Para se perceber as mensagens trocadas está representado na *Figura 11* uma troca típica de mensagens de quando acontece uma busca e o contexto desejado está no nó vizinho.



**Figura 11: Troca de mensagens do protocolo**

As mensagens trocadas pelo protocolo descrito têm as seguintes funcionalidades:

- “`search_request`” – Leva o contexto desejado pelo utilizador e despoleta o mecanismo de busca no nó em malha ao qual se ligou.
- “`search_ack`” – É enviado no segundo caso, quando o contexto não está no nó, avisa o nó móvel que não possui o contexto, mas que o processo de busca foi para os vizinhos.

- “search\_neighbor\_request” – É enviado no segundo caso, quando o contexto não está no nó, leva o contexto desejado pelo utilizador aos nós vizinhos.
- “search\_neighbor\_reply” – É enviado em resposta ao “search\_neighbor\_request”, somente quando o nó fixo vizinho possui o contexto.
- “search\_reply” – É enviado em resposta ao “search\_request”, somente quando o processo de busca termina, com a informação de que o contexto pretendido existe ou não, e se existe qual o nó que o contém.
- “discoverVN\_pkt” – É enviado no início da simulação por *broadcast*, para preenchimento da tabela de encaminhamento (dos vizinhos).

O cabeçalho das mensagens é diferente para cada tipo de mensagem, devido às especificações de cada tipo, mantendo no entanto alguns campos iguais, tal como se pode ver na *Tabela 4*.

Campo do cabeçalho	Tipo de mensagens	Função
ah_type_	Todos	Tipo do pacote.
pkt_src_	Todos	Nó origem do pacote.
pkt_seq_num_	Todos	<i>Sequence number</i> do pacote.
size	Todos	Tamanho do cabeçalho, varia com o número de campos.
VN_	search_request, search_neighbor_request	Contexto pretendido pelo utilizador.
Id_	search_request	Identificador da pesquisa, para fins estatísticos.
pkt_dest_	search_reply, search_ack, search_neighbor_request, search_neighbor_reply	Nó destino do pacote.
flag_	search_reply, search_neighbor_reply	Identificador da validade do campo node_ (verdadeiro ou falso).
node_	search_reply, search_neighbor_reply	Nó que contém o contexto pretendido.
pkt_src_search_	search_neighbor_request, search_neighbor_reply	Nó da malha que origina a busca.

**Tabela 4: Campos do cabeçalho das mensagens**

Mediante o conhecimento do princípio de funcionamento e do tipo de mensagens, procede-se de seguida e em detalhe à explicação do mecanismo de troca de mensagens.

No início da simulação cada nó envia um pacote de *broadcast*. Os nós que recebem essa mensagem preenchem uma tabela de encaminhamento com o endereço origem da mensagem recebida, ficando assim com uma tabela com todos os vizinhos acessíveis. Como na solução proposta os nós em malha são fixos, este processo é feito apenas no início da simulação, mas está preparado para ser feito periodicamente se for preciso.

Quando um nó móvel chega à rede e decide ligar-se, envia uma mensagem *search\_request*, com o contexto pretendido, por *broadcast* periodicamente até receber a mensagem *search\_reply*. Caso receba a mensagem *search\_ack*, isto significa que o contexto desejado não está no nó ao qual se ligou, mas que a procura foi para os vizinhos. Por isso espera um *timeout* definido pela mensagem *search\_reply*, ao fim do qual se não tiver recebido a mensagem *search\_reply* volta a enviar periodicamente o *search\_request*.

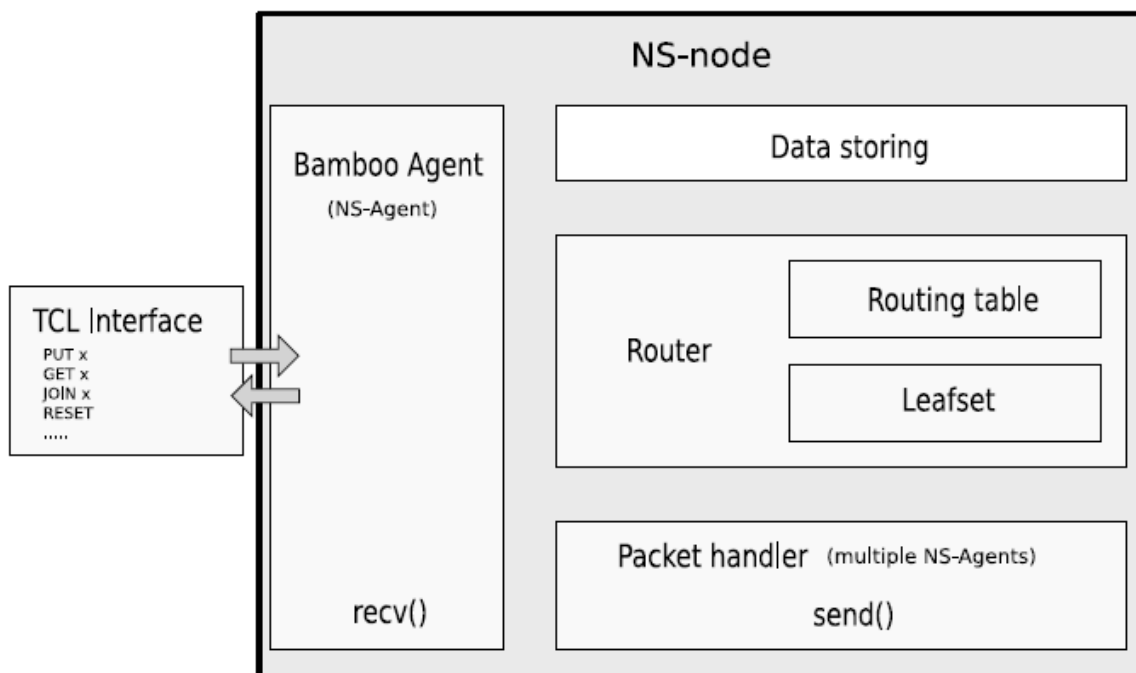
Quando um nó da rede recebe a mensagem *search\_request*, vai verificar se possui o contexto pedido na tabela de contexto; caso possua, envia a mensagem *search\_reply* ao nó móvel e faz a interacção com o TCL para efectuar a ligação. Caso não possua, envia uma mensagem *search\_ack* ao nó móvel a informar que a busca foi para os nós vizinhos. Envia também periodicamente a mensagem *search\_neighbor\_request* para cada um dos seus vizinhos presentes na tabela de encaminhamento, até receber a mensagem *search\_neighbor\_reply* ou já ter enviado para todos os vizinhos. Se recebe uma mensagem *search\_neighbor\_reply*, faz a interacção com o TCL para efectuar a ligação e faz a extensão; caso contrário, faz a interacção com o TCL para a procura ir para o *Bamboo*. Em ambos os casos é enviada uma mensagem *search\_reply* ao nó móvel informando-o que o processo está concluído; informa também se a procura foi para o *Bamboo* ou não.

Quando um nó da rede recebe uma mensagem *search\_neighbor\_request*, ele vai verificar se possui o contexto pedido na tabela de contexto; caso possua, responde com uma mensagem *search\_neighbor\_reply*; caso contrário não responde.

- ***Descoberta Global***

### ***Descrição da extensão já existente***

Este mecanismo de descoberta global baseia-se na procura, entre as VNs disponíveis (em toda a rede) baseadas em contexto, de uma VN que seja adequada ao conjunto de requisitos de contexto que caracterizam um utilizador ou aplicação particular.



**Figura 12: Diagrama de blocos da implementação do Bambo em NS-2 [113]**

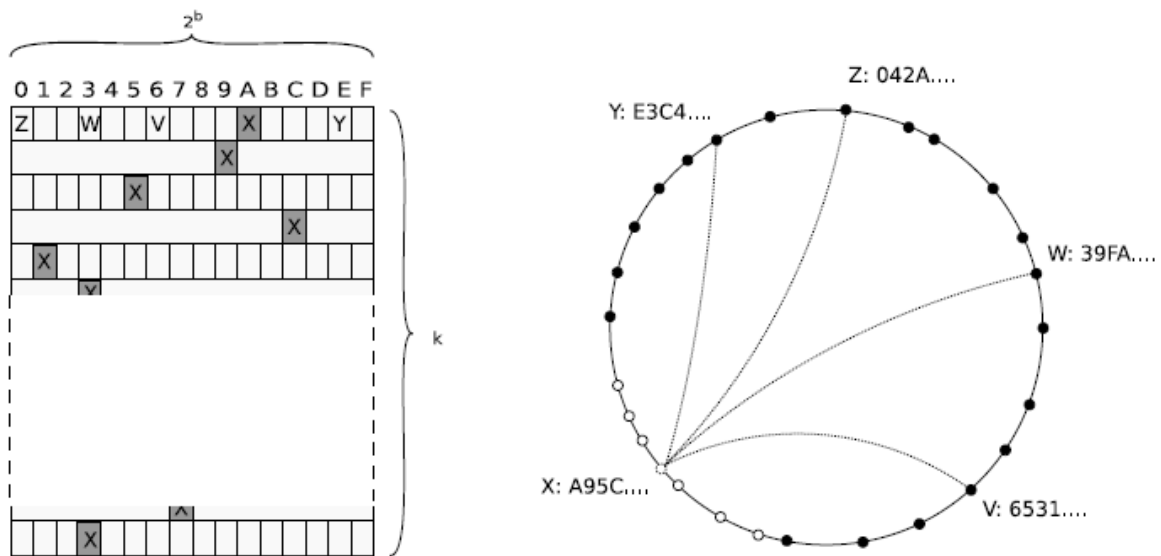
Para implementar este mecanismo no NS-2 foi usado a versão do *Bamboo* para NS-2 [116], que é uma solução distribuída denominada por P2P, cujo diagrama de blocos pode ser visto na *Figura 12*.

O *Bamboo* [117] é referido como uma DHT de terceira geração [118].

Para manter a estrutura da rede, o *Bamboo* mantém dois conjuntos de informação dos vizinhos em cada nó, a *leafset* e a tabela de encaminhamento (*Routing table*). A *leafset* consiste numa lista dos predecessores e sucessores que estão mais perto no espaço de chaves. A tabela de encaminhamento é usada para otimizar o processo de busca, contendo atalhos para chaves com prefixos diferentes, como pode ser visto na *Figura 13*. Nesta figura, do lado direito está representado o espaço de chaves, onde os nós brancos são os nós da *leafset* do nó X, e os arcos são os atalhos da tabela de encaminhamento do nó X, que pode ser vista do lado esquerdo. As chaves, embora possam estar perto (na *leafset*), na rede podem estar fisicamente longe.

Quando a chave é armazenada na DHT usando o comando *PUT*, os dados são encaminhados através da DHT para o nó principal responsável por armazenar os dados. Quando o nó responsável obtém os dados, ele armazena-os em *cache* e nos nós vizinhos da *leafset*, mediante o número de cópias desejado. Cada nó periodicamente escolhe um nó da *leafset* e sincroniza as chaves armazenadas com ele.





**Figura 13: Tabela de encaminhamento do nó X. Os nós brancos são os nós da *leafset* do nó X, os arcos são as entradas da tabela de encaminhamento do nó X.**

[116]

A gestão da rede é efectuada periodicamente em todas as camadas. Cada nó gera *pings* entre vizinhos, para se certificar que o vizinho ainda pode ser alcançável e para obter o RTT para cálculos de *timeout*. Tais valores são usados para saber se um pacote está perdido e precisa de ser reenviado. Os nós também actualizam a tabela de *leafset* periodicamente, sincronizando com um nó aleatório da *leafset*.

O *Bamboo* considera dois nós no mesmo nível se um nó estiver na tabela de encaminhamento do outro, por isso a actualização da tabela de encaminhamento é usada para trocar informações dos nós no mesmo nível. Se um nó obtém informações de outro que está no mesmo nível, verifica a acessibilidade e o RTT; caso esteja acessível, verifica se o campo na tabela de encaminhamento está ocupado; se não estiver adiciona; caso contrário, adiciona se tiver menor latência. É importante referir que a tabela de encaminhamento optimizada não influencia a exactidão da pesquisa, mas apenas a latência da pesquisa.

O número máximo de chaves na rede é dado por  $2^{(b \cdot k)}$ , sendo  $b$  e  $k$  configuráveis.

O *Bamboo* foi avaliado através de simulação [118] e na utilização de redes experimentais como o PlanetLab [119].

### **Funcionalidades Implementadas**

Para a solução proposta é essencial tomar em conta os seguintes requisitos:

- Mapeamento das características dos parâmetros de contexto das VNs (p. ex. atraso, largura de banda) na chave;

- Inserção da chave num dos nós da VN;
- Adequação à reconfiguração/extensão da rede;
- Adequação ao tamanho da rede;
- Total independência relativamente ao restante tráfego.

Para o mapeamento das características dos parâmetros de contexto da VN na chave, decidiu-se que cada um dos algarismos da chave representava um dos parâmetros de contexto, ficando o algarismo menos significativo para descrever o nível da largura de banda, o algarismo seguinte para descrever o nível de mobilidade dos utilizadores, e o algarismo seguinte para descrever o nível do atraso. Se houvesse mais parâmetros de contexto, esses seriam mapeados da forma descrita acima, obtendo-se assim uma chave com tantos algarismos como parâmetros de contexto, sendo de referir que o número máximo possível de chaves no *Bamboo* é configurável como já mencionado acima.

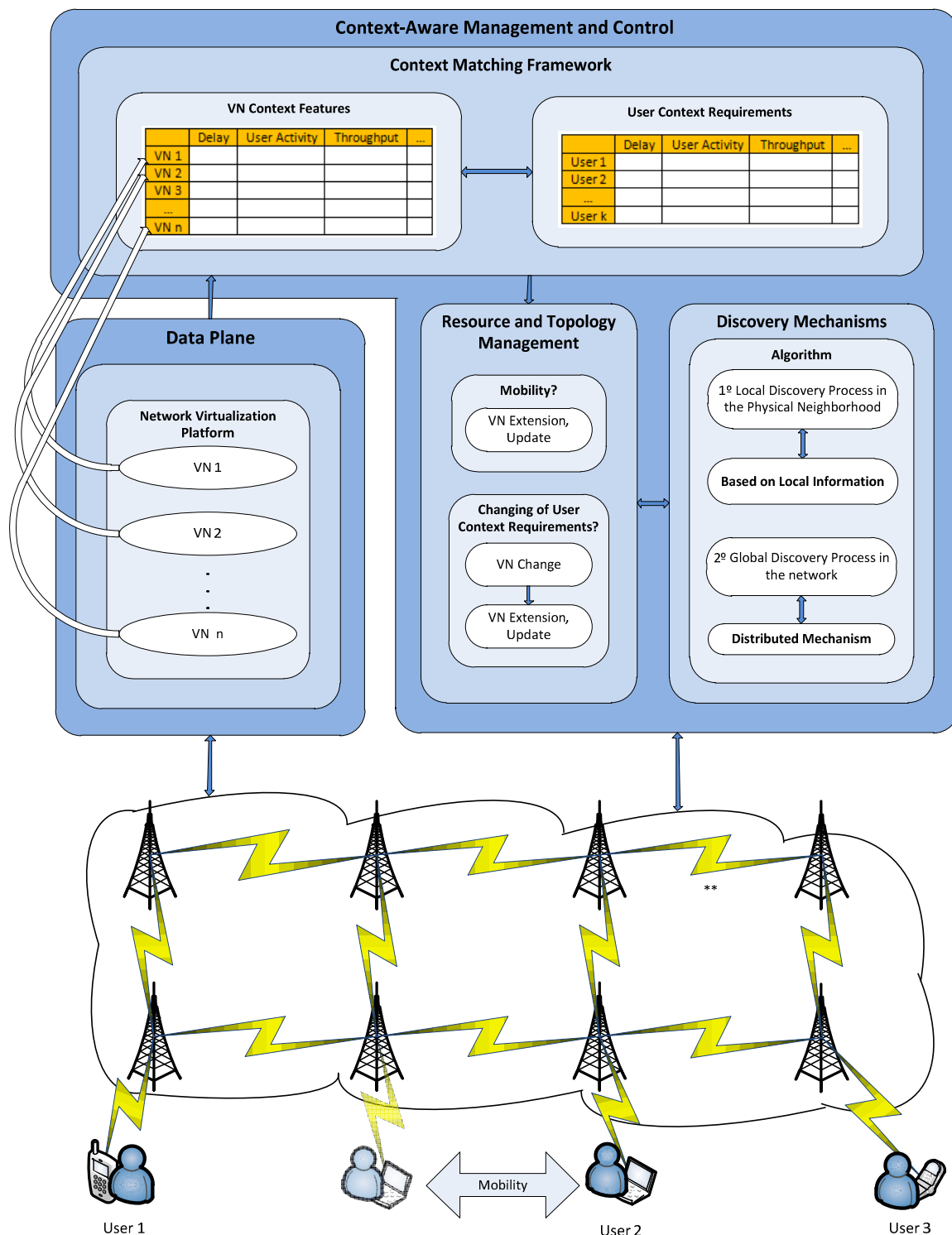
Para se inserir a chave na VN, surgiram complicações visto que como dito anteriormente, o *Bamboo* insere as chaves no nó que contém a chave numericamente mais próxima e não num nó específico da VN como se pretende. Para resolver este problema resolveu-se mapear o nó nos dois algarismos mais significativos da chave, sendo que quando se processa a reconfiguração da rede e se necessita de mudar a chave de localização, este muda o mapeamento do nó na chave.

Definiram-se três níveis para cada medida de contexto, em que 0 é o melhor caso e 2 é o pior. Por exemplo, para uma VN com muita largura de banda, baixa mobilidade dos utilizadores, atraso médio e que foi colocada no nó 15 (um dos nós da VN) possui a chave 15120, como se pode ver na *Tabela 5*.

	Nó da chave	...	Atraso	Mobilidade dos utilizadores	Largura de Banda
Chave da VN1		...			
Chave da VN2		...			
...		...			
Chave exemplo	15	...	1	2	0

**Tabela 5: Mapeamento da informação de contexto na chave**

Para se adequar o *Bamboo* à reconfiguração da VN, caso a VN deixe de conter o nó onde está guardada a chave, optou-se por inserir a chave num dos nós da VN, e alterar o mapeamento do nó na chave para o novo nó, que passa a caracterizar a VN.



**Figura 14: Cooperação entre as diversas funcionalidades implementadas**

Para se adequar o *Bamboo* ao tamanho da rede, alterou-se o tamanho da *leafset*, mediante o tamanho da rede, mais precisamente para 16, 49 e 100 nós na rede, para haver respectivamente 4, 6 e 8 nós na *leafset*. A tabela de encaminhamento não foi

alterada devido ao facto de levar a uma grande degradação da eficiência de funcionamento do *Bamboo*.

Para o *Bamboo* ter total independência relativamente ao restante tráfego da rede, atribuiu-se-lhe uma interface de controlo, em que nas funções já explicadas anteriormente nas múltiplas interfaces, se o pacote for do tipo *Bamboo* é encaminhado para a interface reservada ao controlo P2P.

### **4.3.3 Cooperação entre as diversas funcionalidades implementadas**

Para se perceber melhor como todas as funcionalidades implementadas interagem para criar a solução proposta, está representado na *Figura 14* um diagrama geral.

Neste diagrama pode-se ver que a rede física WMN é dividida em várias VNs, cada uma delas sendo caracterizada por níveis diferentes de parâmetros de contexto, como por exemplo, atraso, largura de banda, segurança, QoS, etc.

Para gerir estas VNs foram criados mecanismos de controlo. Primeiro, é necessário fazer a correspondência entre as características dos parâmetros de contexto desejado pelo utilizador e as características dos parâmetros de contexto das VNs disponíveis, para se seleccionar a VN mais adequada ao utilizador. Segundo, é necessário adaptar a VN para passar a reservar os recursos necessários, isto é feito através da reconfiguração da rede. Terceiro, é necessário um mecanismo de descoberta, para encontrar na rede, a VN pretendida, para isso existe o mecanismo local e o global.

O mecanismo local, denominado de *Discover VN*, é responsável de a nível local (no próprio nó e vizinhos) corresponder as VNs disponíveis baseadas em contexto, com o conjunto dos requisitos de contexto que caracterizam um utilizador ou aplicação particular. No caso de não se encontrar a nível local nenhuma VN que se adeque ao pretendido, é o mecanismo global distribuído, denominado de *Bamboo*, que vai fazer essa correspondência, visto que possui o conhecimento de toda a rede, fornecido pelo mecanismo local.

Estes mecanismos de controlo são chamados, pela ordem acima evidenciada, não só quando chega um novo utilizador à rede, mas também caso se alterem os parâmetros de contexto de um utilizador, e no caso da mobilidade. Neste último caso, a adaptação da rede será simplesmente uma extensão ou reconfiguração da VN. Caso se alterem parâmetros de contexto e se na selecção de rede for escolhida outra VN, terá de se mudar de VN e libertar os recursos que estavam reservados, e reconfigurar ou estender a nova VN.

## 4.4 Conclusão

Este capítulo apresenta as extensões feitas no NS, para desenvolver e avaliar a solução proposta.

Não obstante dos desafios e das dificuldades associadas à introdução de novas funcionalidades na arquitectura do NS, é possível concluir que as funcionalidades básicas da solução proposta foram aplicadas de forma satisfatória.

Um módulo importante implementado é o do suporte da virtualização de redes, que é responsável por garantir a total independência entre as diferentes VNs, cada uma das quais com diferentes parâmetros de contexto. Este módulo e o módulo da atribuição de recursos dinâmicos, que permitiu a introdução da Qualidade de Serviço e da diferenciação da largura de banda, são os módulos do plano de dados da solução proposta.

A nível do plano de controlo foram implementados quatro módulos: o módulo do mapeamento do contexto dos utilizadores e aplicações, essencial para se proceder à selecção da VN que mais se adequa aos utilizadores; o módulo da adaptação dinâmica dos recursos e topologia da rede em função do contexto, essencial para reconfigurar as VNs, caso as características dos parâmetros de contexto dos utilizadores ou aplicações mudem; e dois módulos dedicados à descoberta de VN/nós na rede. O módulo do mecanismo de descoberta local (*Discover VN*) é responsável pela descoberta a nível local da VN que melhor se adapte às características de contexto pretendidas por um utilizador ou aplicação particular. O módulo do mecanismo de descoberta global (*Bamboo*) é uma solução descentralizada, distribuída na rede baseada em P2P; cabe-lhe a ele a descoberta a nível global, da VN que melhor se adapte às exigências de um utilizador ou aplicação particular, em termos das características dos parâmetros de contexto.

Por fim, os módulos foram combinados de forma a poderem implementar a solução proposta e descrita no Capítulo 3.



## **5. Avaliação: resultados e discussão**

### **5.1 Introdução**

Este capítulo expõe uma avaliação de desempenho da solução proposta, implementada através de diferentes cenários.

Na secção 5.2 são explicados os cenários e a topologia utilizada no NS-2.33 para avaliar diferentes aspectos. As secções seguintes fornecem os resultados e as conclusões das diferentes avaliações, medindo o desempenho e comparando os resultados obtidos para parâmetros de configuração diferentes.

Na secção 5.3 é avaliada a influência da virtualização no plano de dados com tráfego UDP e tráfego TCP. Na secção 5.4 são avaliados os mecanismos de descoberta implementados nesta dissertação. Na secção 5.5 é avaliada a influência da mobilidade dos utilizadores na solução de redes em malha proposta.

Na secção 5.6 são resumidos e discutidos os resultados obtidos nas avaliações feitas. Finalmente, na secção 5.7 são fornecidas as conclusões.

### **5.2 Cenários e detalhes do simulador**

Com o propósito de executar avaliações diferentes da solução proposta, foi desenvolvido um cenário genérico. De maneira a obter resultados em diferentes situações mais facilmente, o cenário possui parâmetros de entrada, tais como o tamanho da rede, número de VNs, número de utilizadores, etc. Estes parâmetros, devido à sua importância serão explicados detalhadamente.

Um parâmetro importante em NS é a semente utilizada na simulação, uma vez que os resultados são os mesmos para cada semente. Para se obter resultados diferentes para o mesmo cenário é necessário alterar a semente, a fim de se poder tratá-los estatisticamente (valores médios e intervalos de confiança a 90%). A semente é utilizada para alterar todos os parâmetros aleatórios do NS, provocando alterações nas rotas entre os nós, assim como nas condições de tráfego e na mobilidade dos utilizadores.

#### **5.2.1 Tamanho da rede**

O tamanho da rede nos cenários varia entre três hipóteses: 4x4, 7x7 ou 10x10 nós da rede, sempre com a mesma topologia, de forma a se poder comparar mais facilmente os resultados. Um exemplo da topologia 4x4 pode ser visualizado na *Figura 15*.



**Figura 15: Topologia da rede 4X4**

Os nós em malha distam uns dos outros 100 metros e possuem o alcance de 100 metros. Por exemplo, o nó 5 apenas consegue comunicar com os nós 1, 4, 6 e 9, e estes nós serão por isso considerados seus vizinhos.

Os nós móveis possuem um alcance de 40 metros e ligam-se aleatoriamente na rede estabelecendo comunicações entre eles, dependendo do cenário considerado.

### **5.2.2 Parâmetros e níveis de contexto**

É crucial providenciar limites para uma topologia eficaz e para a adaptação de recursos a cada VN, de acordo com as mudanças de contexto. Esses limites foram impostos através da criação de níveis, em que 0 é o melhor caso e 2 o pior, e devem ser adaptados à variação dos padrões dos parâmetros de contexto considerados, tais como a largura de banda, actividade de utilizadores e atraso.

O parâmetro 'largura de banda' é diferenciado através da interface à qual o utilizador ou aplicação se vai ligar (e onde está a VN). Cada interface tem diferentes larguras de banda associadas e diferentes tipos de tráfegos. Cada tráfego representa a comunicação efectuada por um utilizador ou aplicação com os parâmetros de contexto considerados.

O parâmetro 'nível de actividade dos utilizadores' é diferenciado através da prioridade atribuída a cada VN, ou seja, um utilizador que possua mais actividade será ligado a uma VN que terá maior prioridade que as restantes VNs.



O parâmetro 'atraso' é diferenciado através do número de saltos da VN na rede, ou seja, um utilizador ou aplicação que necessite de pouco atraso será ligado a uma VN que terá menos saltos de extensão do que as outras VNs.

### 5.2.3 Número de redes virtuais

O número de VNs varia entre três hipóteses: 3, 6 ou 9 por interface (cada interface tem um canal diferente associado), o que dá um total de 9, 18 ou 27 na rede. No caso de não haver virtualização, como só há uma interface, as VNs vão estar todas nessa interface.

### 5.2.4 Mapeamento dos parâmetros de contexto nas redes virtuais

Para se simular o parâmetro da 'largura de banda', foi dividida a largura de banda total por 5 interfaces quando há virtualização (3 para dados e 2 para controlo), sendo que o nível 0 é associado à interface 0 que possui maior largura de banda, enquanto o nível 2 é associado à interface 2 que possui menor largura de banda. A largura de banda pretendida era de 54 Mb/s, mas visto que no NS o tempo de simulação aumenta exponencialmente com o aumento do tráfego, utilizou-se uma largura de banda de 11 Mb/s. Associado à divisão da largura de banda por interfaces, foram considerados tipos de tráfego diferentes para cada uma das interfaces de dados, com as características em termos de tamanho do pacote e taxa (por utilizador) apresentadas na *Tabela 6* (É de notar que estes valores foram convertidos dos valores típicos de uma largura de banda de 54 Mb/s para 11Mb/s, devido ao tempo de simulação, como já explicado acima).

Para se simular o parâmetro 'nível de actividade dos utilizadores', este foi diferenciado através da prioridade atribuída a cada VN, em que o nível de prioridade é igual ao nível do parâmetro 'nível de actividade dos utilizadores'.

Para se simular o parâmetro 'atraso', este foi diferenciado através do número de saltos na rede que o fluxo tem de percorrer, em que esse número é proporcional ao tamanho da rede, ou seja, considerando o número máximo de saltos possíveis entre dois nós da rede (pelo percurso mínimo) como  $\Psi$ , o nível 0 corresponde a saltos inferiores ou iguais que  $\Psi / 4$ , o nível 1 corresponde a saltos superiores a  $\Psi / 4$  e inferiores ou iguais a  $\Psi / 2$ , o nível 2 corresponde a saltos superiores a  $\Psi / 2$ .

O resumo do mapeamento dos parâmetros de contexto nas VNs pode ser visto na *Tabela 6*.

Níveis	Parâmetros	Nível de actividade dos utilizadores	Atraso
	Largura de Banda		
0	512 Bytes	Prioridade 0	Ligação com [1, $\Psi/4$ ] saltos
	32 Kb/s		
1	256 Bytes	Prioridade 1	Ligação com ] $\Psi/4$ , $\Psi/2$ ] saltos
	8 Kb/s		
2	64 Bytes	Prioridade 2	Ligação com ] $\Psi/2$ , $\Psi$ ] saltos
	2 Kb/s		

**Tabela 6: Mapeamento dos parâmetros de contexto nas VNs**

### 5.2.5 Número de utilizadores

O número de utilizadores por VN varia entre quatro hipóteses: 1, 2, 3 ou 4 sendo que estes valores são mapeados em fluxos tendo em conta o tamanho dos pacotes e a taxa, definida na *Tabela 6*.

### 5.2.6 Descoberta das redes virtuais

Para se simular os mecanismos de descoberta local/global, cada nó possui um nó móvel associado inicialmente desactivado. O nó móvel não possui associado nenhum mecanismo de controlo: quando o nó chega à rede (é activado via script TCL) liga-se a um nó em malha e esse nó em malha é que possui os mecanismos de controlo e gestão da rede. Cada nó móvel vai-se tentar ligar a uma VN específica, para se testar os tempos de descoberta, e de reconfiguração das VNs, de modo que todos os nós móveis activos, caso se movam para outro nó em malha, iniciam o processo de descoberta de novo. No caso de haver mobilidade, testa-se também os tempos do restabelecimento da conectividade e actualização do mecanismo global.

O protocolo de encaminhamento usado foi o AODV, visto ser um protocolo adaptativo a cenários com alta mobilidade.

### 5.2.7 Mobilidade dos utilizadores

Para se simular a mobilidade dos utilizadores, foi considerado que quando o utilizador se move, movimenta-se a determinada velocidade que lhe permite dar dois “saltos” na rede. Ou seja, quando o nó móvel perde a ligação e se volta a ligar já está à

distância de dois saltos de onde estava ligado. Assumiu-se dois saltos e não apenas um salto, pois em um salto o mecanismo de gestão distribuído nunca seria chamado (seria chamado sempre o mecanismo local, pois a VN estaria sempre num dos nós vizinhos).

### 5.2.8 Métricas de avaliação de desempenho

Ao longo das secções seguintes, são apresentadas algumas medidas de desempenho (*Overhead, Delay, Packet Loss, Throughput*).

O *Overhead* é apresentado em percentagem  $\left(\frac{\text{número de bytes de controlo}}{\text{número de bytes total}}\right)$  e representa a percentagem do tráfego que é usado para controlo, sendo de notar que uma menor percentagem não é necessariamente menos bytes de controlo.

O *Delay* em todos os cenários é medido em mili-segundos e pode representar o atraso de uma comunicação, o tempo de descoberta de determinada VN, o tempo de extensão de determinada VN, e o atraso no restabelecimento de conectividade.

O *Packet Loss* é apresentado em percentagem  $\left(\frac{\text{número de pacotes/bytes perdidos}}{\text{número de pacotes/bytes total}}\right)$  e é outra medida importante para avaliar a rede e a solução proposta, representa a percentagem de dados que se perdeu na rede.

O *Throughput* em todos os cenários é apresentado em kbps  $\left(\frac{\text{número de bites enviados}}{\text{tempo de transmissão}}\right)$  e é outra medida importante que permite avaliar a saturação da rede, representa a quantidade de tráfego enviada que chegou ao destino.

## 5.3 Influência da virtualização

Uma das principais características da solução proposta é a virtualização, que se pretende que aumente o desempenho da rede. Para se quantificar o desempenho da rede, realizou-se a simulação num ambiente com virtualização e num ambiente limpo, variando-se os parâmetros já explicados anteriormente, nomeadamente o tamanho da rede, o número de utilizadores na rede (o número de fluxos), o tipo de tráfego (UDP ou TCP) e o número de VNs por interface (cada interface possui um canal diferente). No caso da virtualização são utilizadas três interfaces; sem virtualização é apenas utilizada uma interface (com a largura de banda e o tráfego a ser o somatório da largura de banda e do tráfego de todas as interfaces na virtualização).

### 5.3.1 Tráfego UDP

- **Rede 4x4 nós**

Os parâmetros *Delay*, *Throughput* e *Packet Loss*, são importantes para se avaliar os ganhos de desempenho da aplicação da virtualização, aplicados à solução proposta. A *Figura 16* apresenta uma rede 4x4 variando o número de VNs e o número de utilizadores por VN em termos de quantidade de fluxo. São apresentados gráficos para os três canais reservados para dados, canal 0 para tráfego sensível à largura de banda, canal 1 para tráfego sensível ao nível de actividade dos utilizadores e canal 2 para tráfego sensível ao atraso.

A *Figura 17* apresenta uma rede 4x4 variando o número de VNs e o número de utilizadores por VN em termos de quantidade de fluxos. Apesar de só haver um canal, para se poder comparar com a solução sem virtualização (*Figura 16*), são apresentados gráficos para os três tipos de tráfego de dados.

Os resultados apresentados na *Figura 16* mostram que em termos de atraso entre os diferentes canais, este diminui do tráfego sensível à largura de banda, para o tráfego sensível ao nível de actividade dos utilizadores, e deste para o tráfego sensível ao atraso. Em termos de atraso, no mesmo canal, este sobe com o aumento do número de VNs. Para o mesmo número de VNs pode-se constatar que caso não haja saturação da rede, a variação do atraso com o número de fluxos depende de os fluxos criados atravessarem ligações mais ou menos congestionadas nas diferentes simulações. Caso a rede esteja saturada, o atraso aumenta. Em termos do *Throughput* pode-se ver que tanto o tráfego sensível à largura de banda, como o tráfego sensível ao nível de actividade dos utilizadores, começa a saturar para 9 VNs e 4 fluxos por VN. Em termos de *Packet Loss*, este sobe com o aumento do número de VNs, e para o mesmo número de VNs sobe com o aumento do número de fluxos. Pode-se constatar que no caso em que começa a saturar a rede, o *Packet Loss* aumenta abruptamente.

Comparando agora a *Figura 16* com a *Figura 17*, ao nível do atraso verifica-se que quando há saturação na rede, no caso da virtualização o atraso é inferior relativamente ao NS limpo. No entanto, quando não há saturação da rede, o atraso na virtualização é um pouco maior do que o NS limpo. Ao nível da saturação da rede, verifica-se que o NS limpo satura mais cedo do que com o uso da virtualização, e isto é um grande ganho do desempenho da virtualização pois consegue acomodar mais carga na rede. Ao nível do *Packet Loss*, verifica-se que com a virtualização as perdas diminuem drasticamente relativamente ao NS limpo; isto deve-se ao facto de haver menos tráfego em cada interface, ou seja, existem menos colisões.

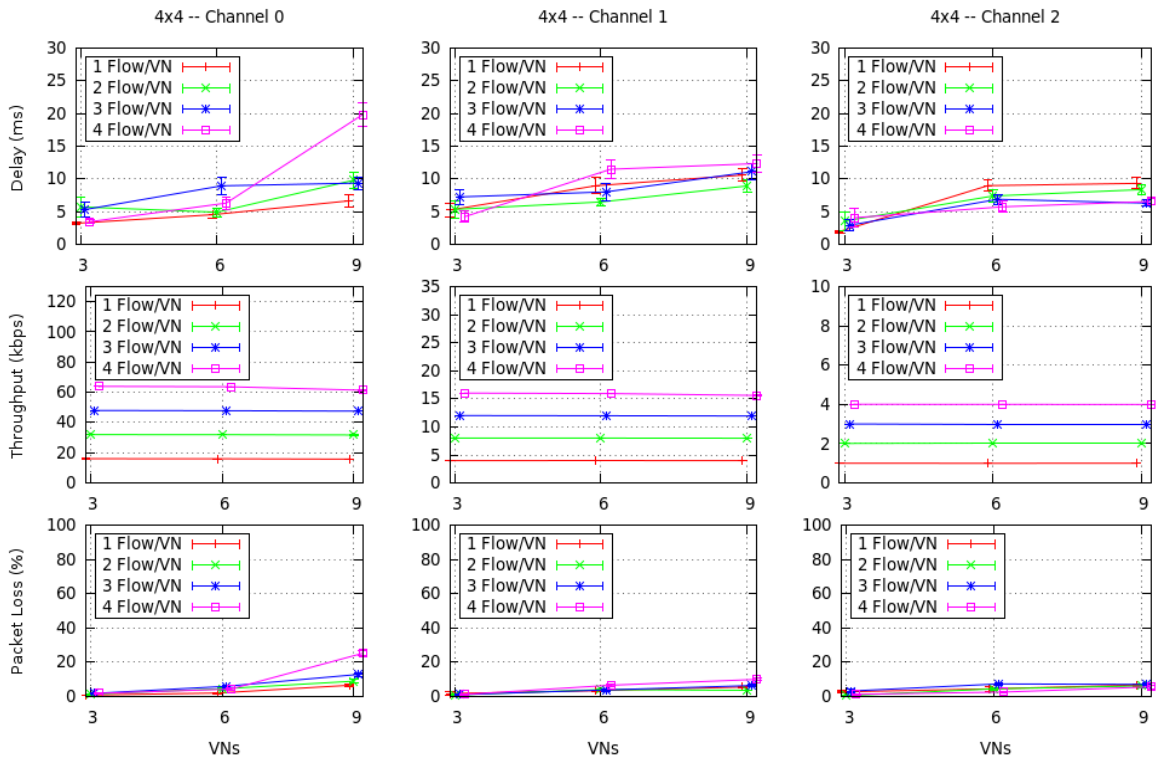


Figura 16: Rede 4x4 com virtualização e tráfego UDP

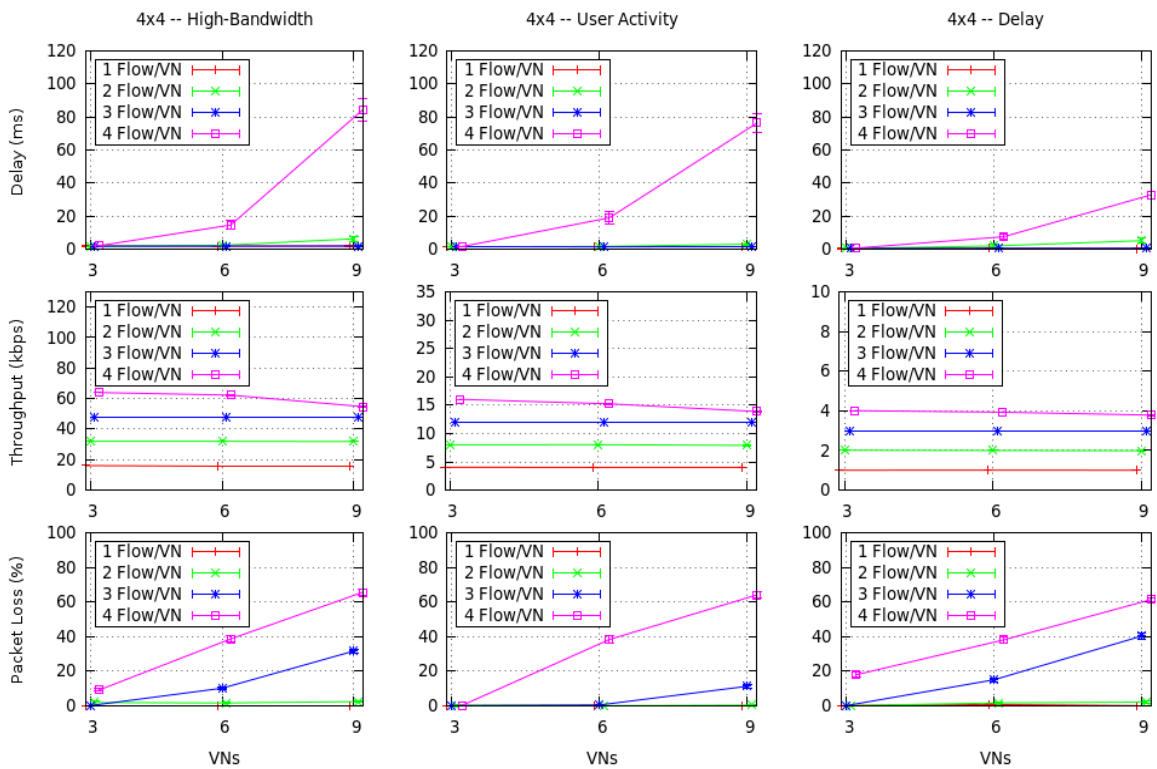


Figura 17: Rede 4x4 sem virtualização e tráfego UDP

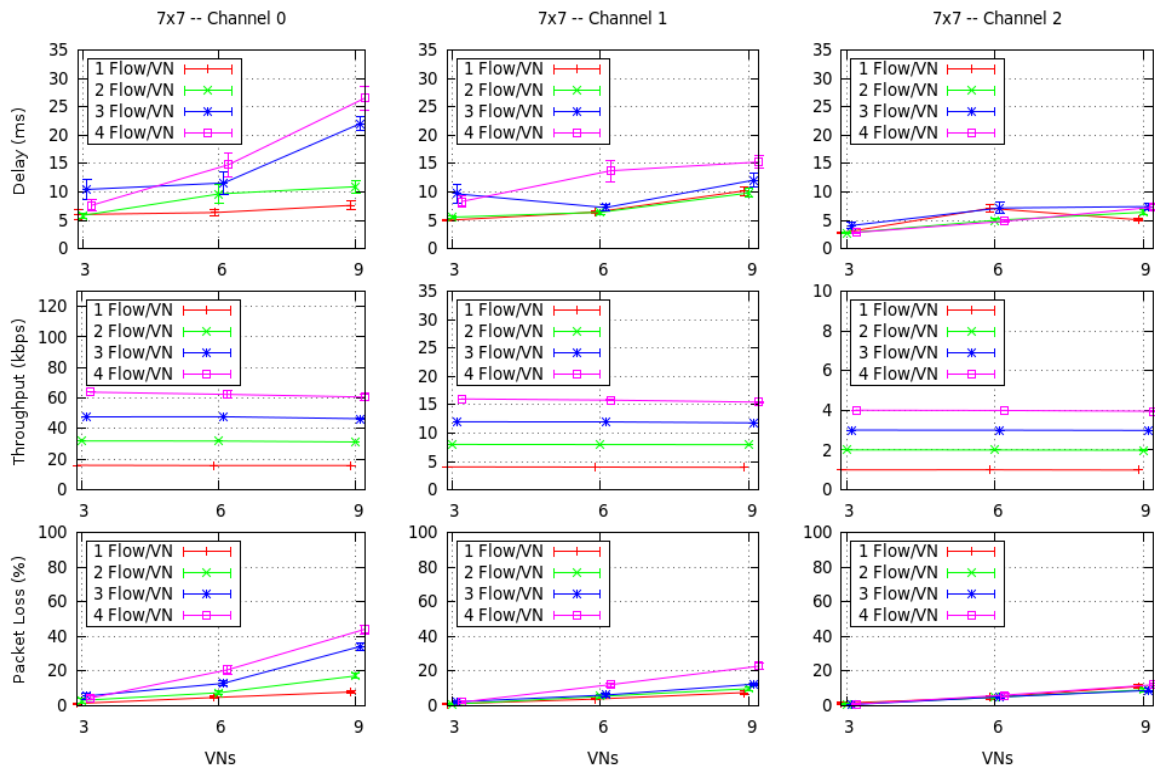


Figura 18: Rede 7x7 com virtualização e tráfego UDP

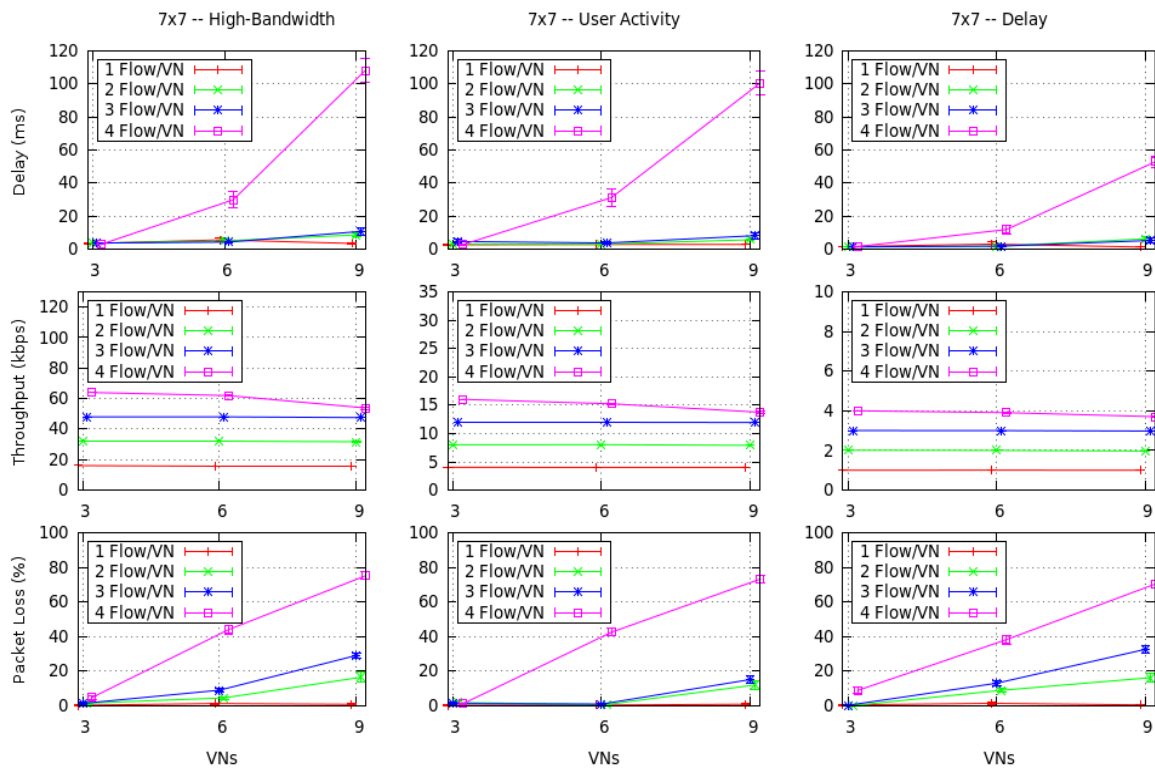


Figura 19: Rede 7x7 sem virtualização e tráfego UDP

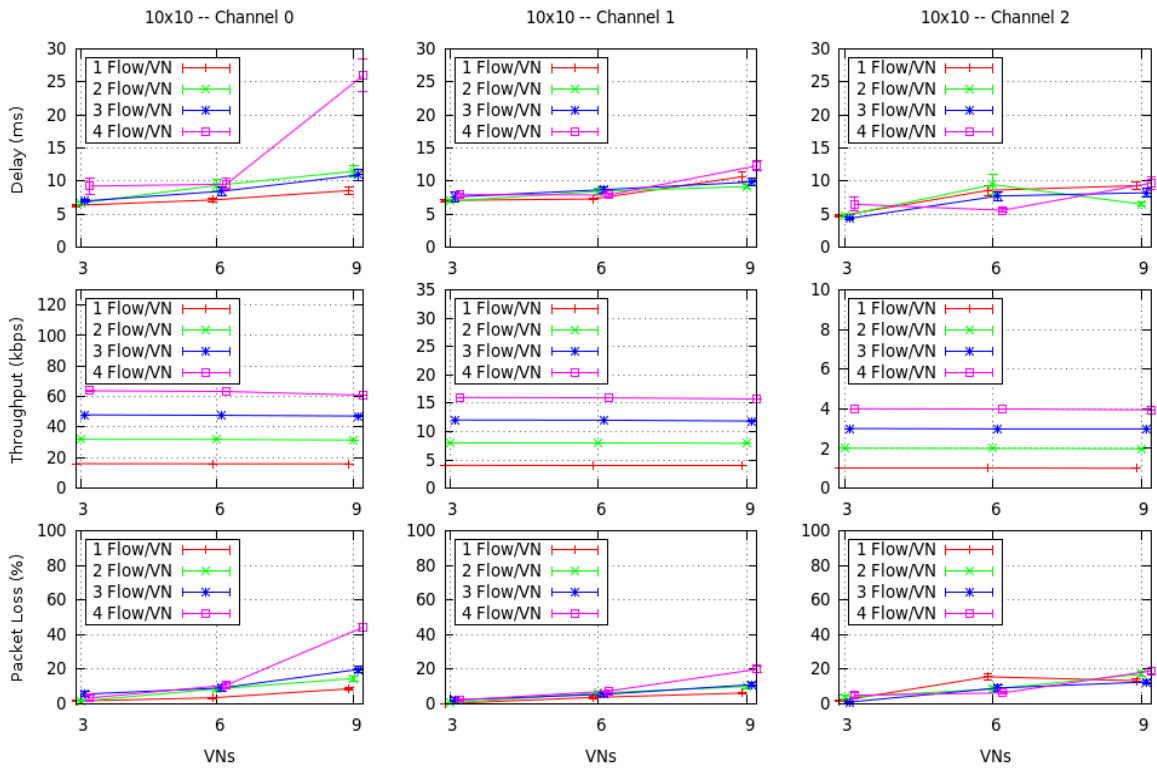


Figura 20: Rede 10x10 com virtualização e tráfego UDP

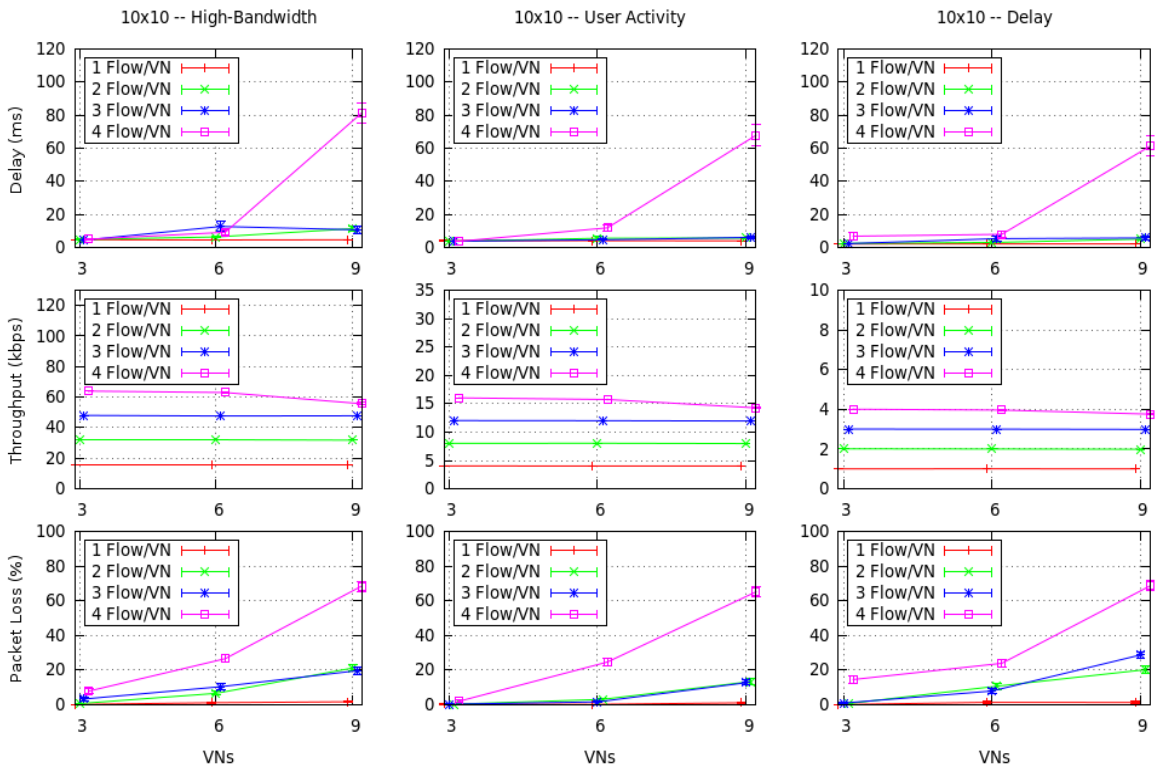


Figura 21: Rede 10x10 sem virtualização e tráfego UDP

Outra característica importante para se avaliar os ganhos de desempenho da aplicação da virtualização, aplicados à solução proposta é a escalabilidade. Para isso voltaram-se a repetir as experiências anteriores para um tamanho de rede de 7x7 nós e para um tamanho de rede de 10x10 nós.

- **Rede 7x7 nós**

A *Figura 18* apresenta uma rede 7x7 variando o número de VNs e o número de utilizadores por VN em termos de quantidade de fluxo. São apresentados gráficos para os três canais reservados a dados, canal 0 para tráfego sensível à largura de banda, canal 1 para tráfego sensível ao nível de actividade dos utilizadores e canal 2 para tráfego sensível ao atraso.

A *Figura 19* apresenta uma rede 7x7 variando o número de VNs e o número de utilizadores por VN em termos de quantidade de fluxo. Neste caso só existe um canal. Para se poder comparar com a solução sem virtualização, são apresentados gráficos para os três tipos de tráfego de dados.

Comparando agora a *Figura 18* com a *Figura 19*, verifica-se que os resultados são semelhantes aos da rede 4x4.

- **Rede 10x10 nós**

A *Figura 20* e a *Figura 21* apresentam uma rede 10x10 nas mesmas condições dos cenários anteriores. Os resultados também são semelhantes aos anteriores.

Comparando agora, a rede 4x4 com a rede 7x7 e com a rede 10x10 no caso em que há virtualização, pode-se verificar que com o aumento da rede, o atraso nos diferentes canais se mantém relativamente constante. A saturação da rede nos diferentes canais mantém-se no mesmo número de utilizadores, e o *Packet Loss* nos diferentes canais aumenta um pouco, mas nada que comprometa o desempenho da rede.

Com base nestes resultados, afirma-se que a solução proposta da virtualização funciona bem em termos de escalabilidade, uma vez que os diferentes parâmetros avaliados, não mudaram os seus valores quando se varia o tamanho da rede.

### 5.3.2 Tráfego TCP

Nesta secção analisa-se tráfego TCP para se poder concluir sobre o desempenho de virtualização com os dois mecanismos de transporte.



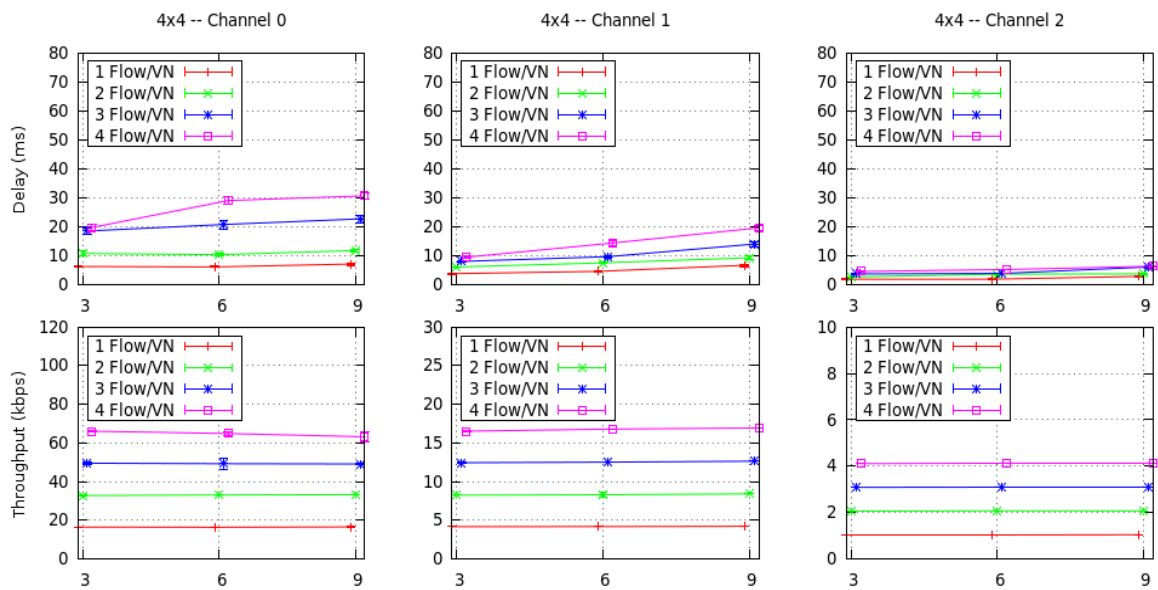


Figura 22: Rede 4x4 com virtualização e tráfego TCP

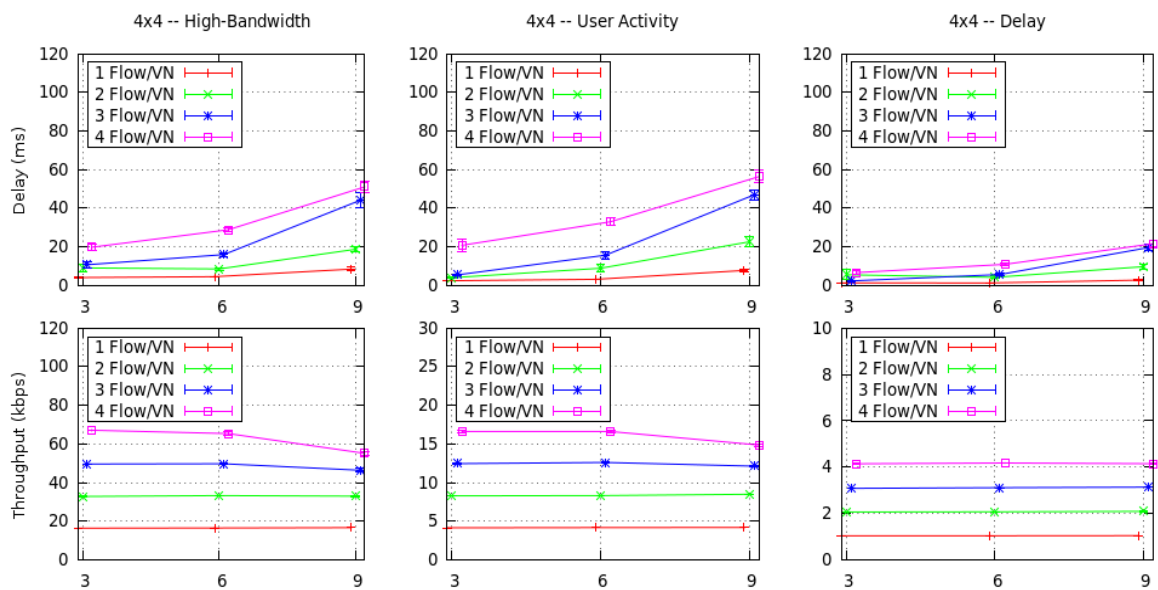


Figura 23: Rede 4x4 sem virtualização e tráfego TCP

- **Rede 4x4 nós**

A Figura 22 apresenta uma rede 4x4 variando o número de VNs e o número de utilizadores por VN em termos de quantidade de fluxo. São apresentados gráficos para os três canais reservados a dados, canal 0 para tráfego sensível à largura de banda, canal 1 para tráfego sensível ao nível de actividade dos utilizadores e canal 2 para tráfego sensível ao atraso.

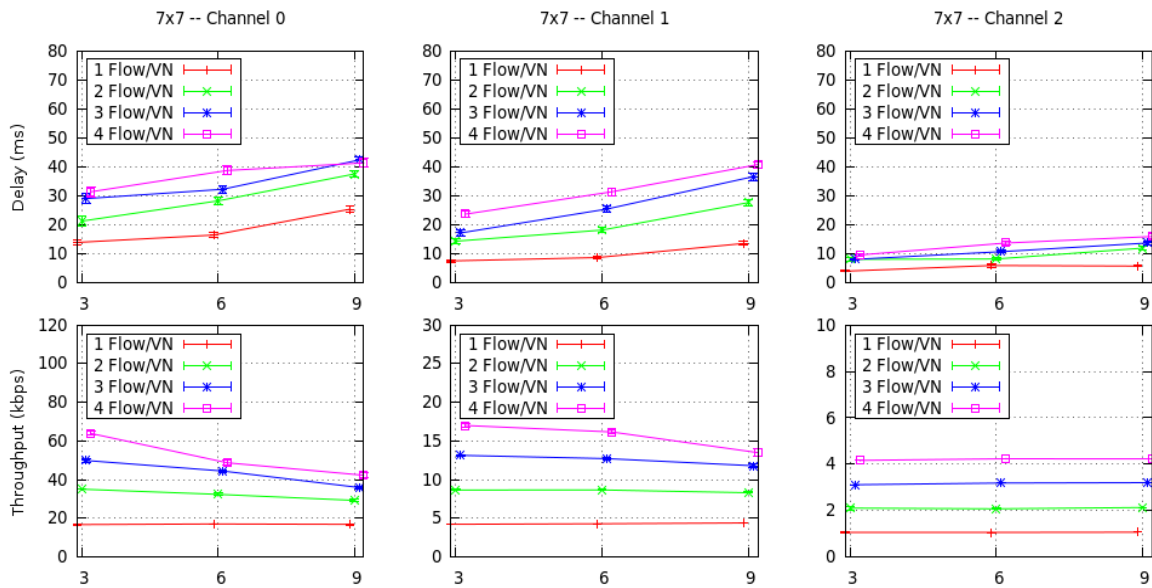


Figura 24: Rede 7x7 com virtualização e tráfego TCP

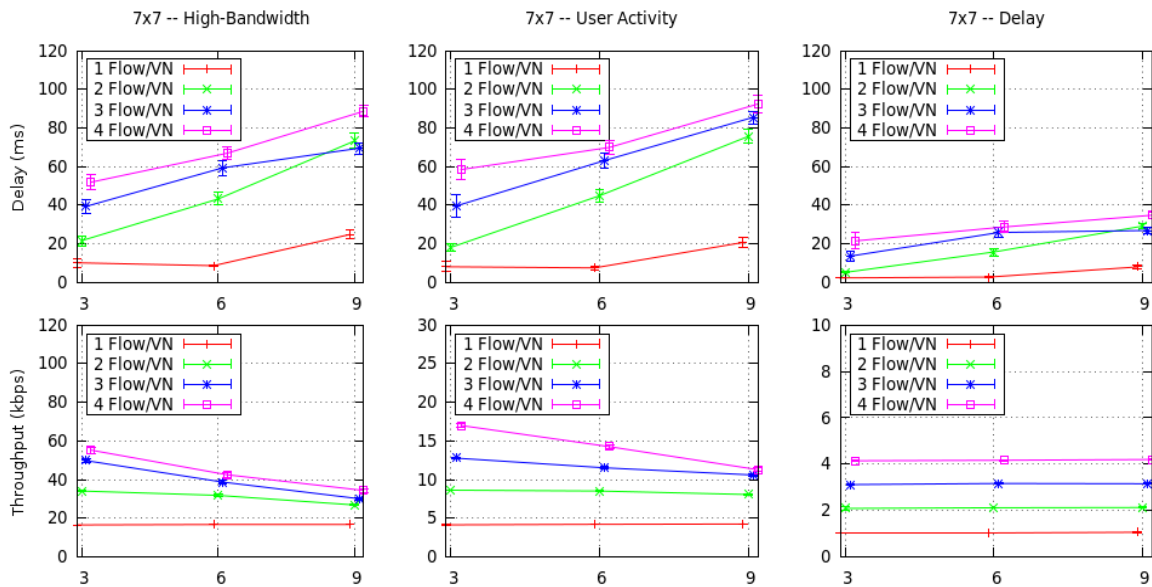
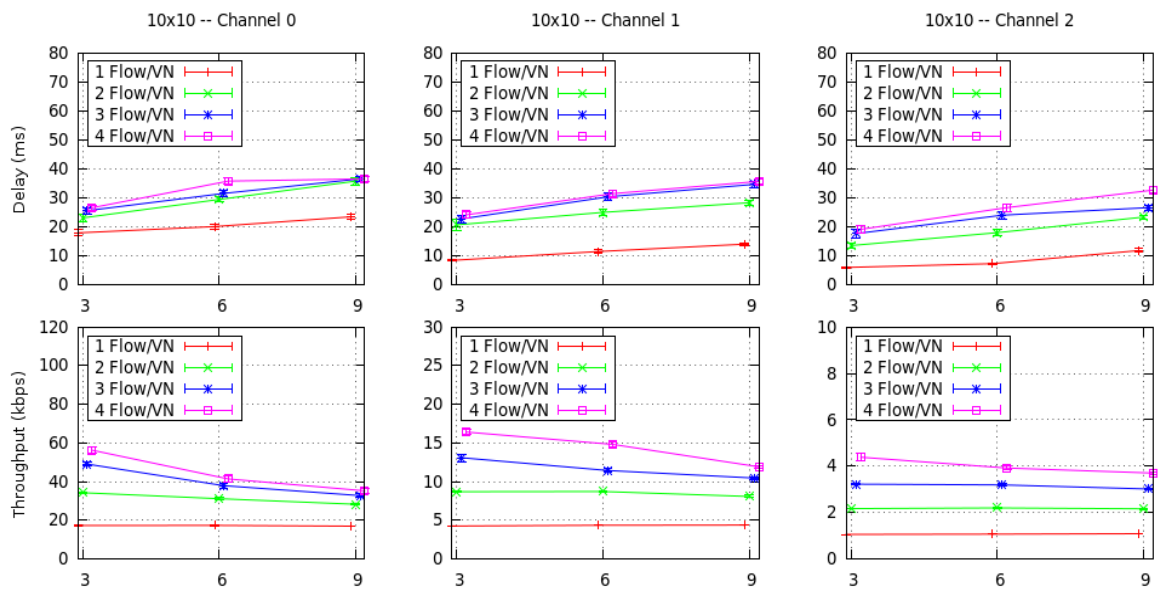


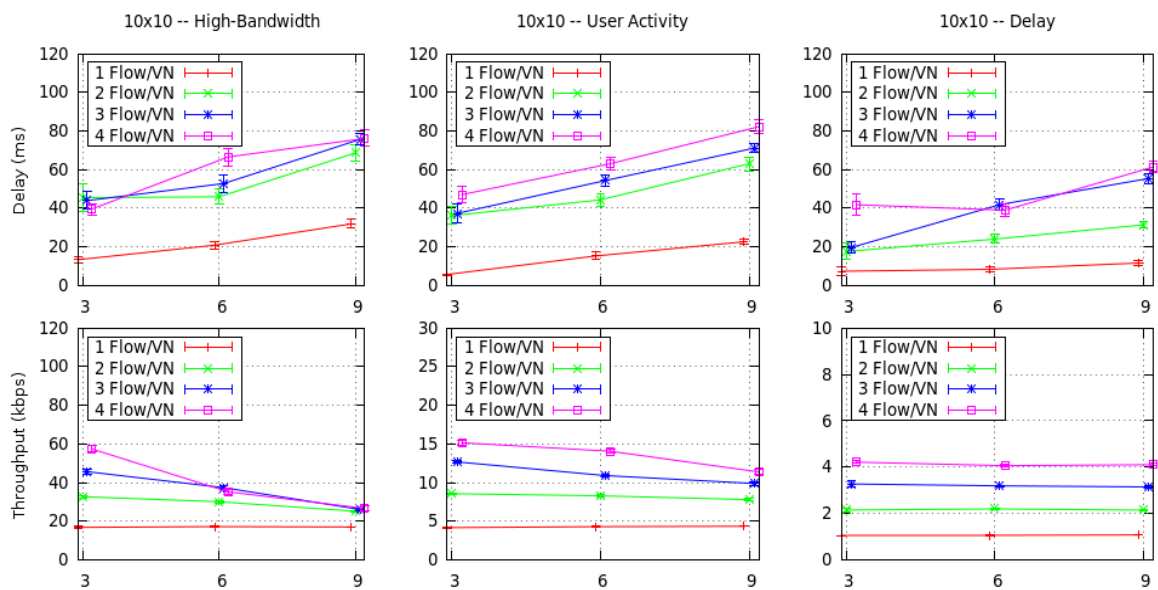
Figura 25: Rede 7x7 sem virtualização e tráfego TCP

A Figura 23 apresenta uma rede 4x4 variando o número de VNs e o número de utilizadores por VN em termos de quantidade de fluxos. Apesar de só haver um canal para se poder comparar com a solução sem virtualização, são apresentados gráficos para os três tipos de tráfego de dados.

Os resultados apresentados na Figura 22 mostram que em termos de atraso entre os diferentes canais, este diminui do tráfego sensível à largura de banda, para o tráfego



**Figura 26: Rede 10x10 com virtualização e tráfego TCP**



**Figura 27: Rede 10x10 sem virtualização e tráfego TCP**

sensível ao nível de actividade dos utilizadores, e deste para o tráfego sensível ao atraso. Em termos de atraso, no mesmo canal, este sobe com o aumento do número de VNs, e para o mesmo número de VNs sobe com o aumento do número de fluxos. Em termos do *Throughput* pode-se ver que o tráfego sensível à largura de banda começa a saturar para 9 VNs e 4 fluxos por VN.

Comparando agora a *Figura 22* com a *Figura 23*, ao nível do atraso verifica-se que quando há saturação na rede, no caso da virtualização o atraso é inferior

relativamente ao NS limpo. No entanto, quando não há saturação da rede, o atraso na virtualização é um pouco maior do que o NS limpo. Ao nível da saturação da rede, verifica-se que o NS limpo satura mais cedo do que com o uso da virtualização, e isto é um grande ganho do desempenho da virtualização pois consegue acomodar mais carga na rede.

Outra característica importante, para se avaliar os ganhos de desempenho da aplicação da virtualização, aplicados à solução proposta, é a escalabilidade. Para isso voltou-se a repetir as experiências anteriores para um tamanho de rede de 7x7 nós e para um tamanho de rede de 10x10 nós.

- **Rede 7x7 nós**

A *Figura 24* e a *Figura 25* apresentam uma rede 7x7 nas mesmas condições dos cenários anteriores. Os resultados são semelhantes aos da rede 4x4.

- **Rede 10x10 nós**

A *Figura 26* e a *Figura 27* apresentam uma rede 10x10 nas mesmas condições dos cenários anteriores. Os resultados também são semelhantes aos anteriores.

Comparando agora, a rede 4x4 com a rede 7x7 e com a rede 10x10 no caso em que há virtualização, pode-se verificar que com o aumento da rede, o atraso nos diferentes canais aumenta um pouco. A saturação da rede nos diferentes canais começa com um menor número de utilizadores, sendo o pior caso o tráfego sensível à largura de banda, que na rede 4x4 começa a saturar para 6 VNs e 4 fluxos por VN, e na rede 10x10 começa a saturar para 6 VNs e 2 fluxos por VN, o que corresponde a reduzir-se o número utilizadores na rede, para metade.

Comparando o tráfego UDP com o tráfego TCP, verifica-se que no TCP a saturação ocorre mais cedo, o que era de esperar visto que no TCP como não há perdas, há mais tráfego na rede devido às confirmações e às retransmissões. Como há mais tráfego na rede e devido à janela de congestionamento, o atraso no TCP é maior que no UDP, principalmente nos casos em que o TCP está saturado e o UDP ainda não. Sendo ainda de referir que ao nível do atraso, no tráfego TCP, para o mesmo número de VNs o atraso sobe com o aumento do número de fluxos; no tráfego UDP, para o mesmo número de VNs, caso não haja saturação da rede, a variação do atraso com o número de fluxos depende dos fluxos criados atravessarem nas diferentes simulações ligações mais ou menos congestionadas. Este resultado pode ser explicado com base na janela de congestionamento que o TCP usa, limitando as perdas através da redução do tráfego, reduzindo por isso a variação do atraso e consequentemente aumentando o atraso.

Em relação à escalabilidade, no tráfego UDP, ao aumentar-se o tamanho da rede, o atraso e o início da saturação da rede mantinham-se constantes. No caso do

TCP, ao aumentar-se o tamanho da rede, o atraso aumentou e a rede começou a saturar mais cedo. Este resultado pode-se explicar com base no aumento do *Packet Loss* no UDP, o que ao não poder acontecer no TCP pela própria definição, aumenta o número de retransmissões e por isso o tráfego na rede, suportando por isso menos utilizadores.

## 5.4 Gestão local e global

Outra característica muito importante da solução proposta centra-se nos mecanismos de descoberta, que se pretende que realizem a pesquisa da VN pretendida de uma forma eficiente. Para se quantificar o desempenho dos mecanismos de descoberta na rede, alteraram-se os parâmetros já explicados anteriormente, nomeadamente o tamanho da rede, o número de utilizadores na rede (o número de fluxos) e o número de VNs por interface (cada interface possui um canal diferente, três interfaces para dados e duas para os mecanismos de descoberta).

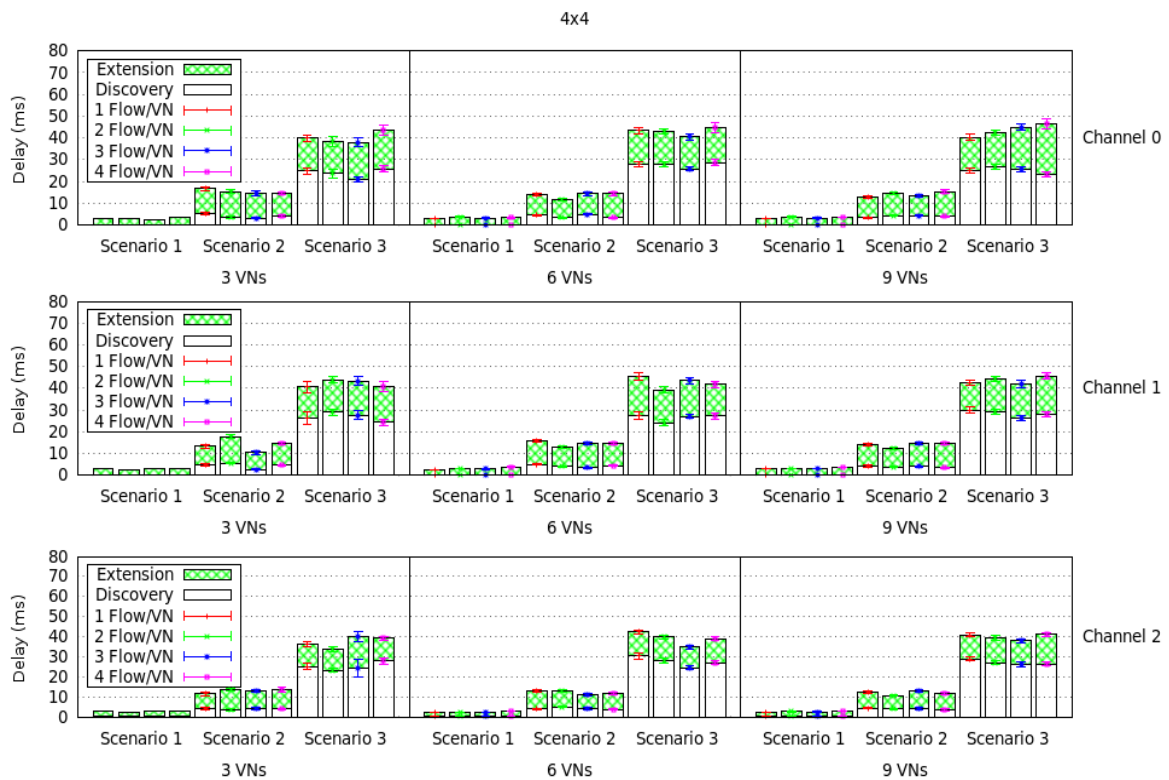
Cada nó móvel vai-se tentar ligar a uma VN específica, para se testar os tempos de busca, e de reconfiguração das VNs, de modo que todos os nós móveis se tentam ligar a todas as VNs possíveis.

### 5.4.1 Atraso

O parâmetro *Delay* é importante para se avaliar o desempenho dos mecanismos de descoberta, aplicados à solução proposta. A *Figura 28* apresenta os tempos de descoberta e extensão numa rede 4x4 variando o número de VNs e o número de utilizadores por VN em termos de quantidade de fluxo. São apresentados gráficos para os três canais reservados a dados, canal 0 para tráfego sensível à largura de banda, canal 1 para tráfego sensível ao nível de actividade dos utilizadores e canal 2 para tráfego sensível ao atraso. Nos gráficos apresentados, *Scenario 1* representa a descoberta local, *Scenario 2* representa a descoberta através do mecanismo da vizinhança, e *Scenario 3* representa a descoberta através do mecanismo distribuído.

A *Figura 29* apresenta os tempos de descoberta e extensão numa rede 7x7. E a *Figura 30* apresenta os tempos de descoberta e extensão numa rede 10x10.

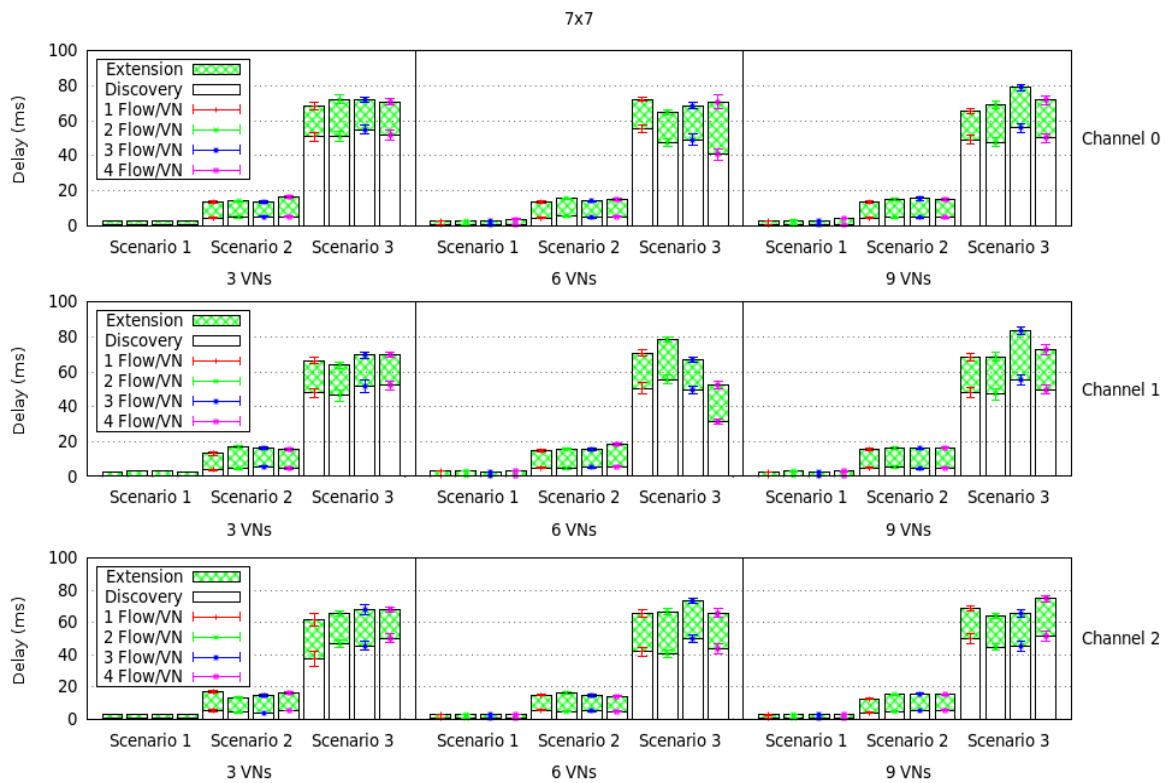
Os resultados apresentados na *Figura 28* mostram que em termos do tempo de descoberta, no mesmo canal, o mecanismo local é o mais rápido, e o mecanismo distribuído (*Bamboo*) é o mais lento. O mecanismo local só trabalha no próprio nó, o mecanismo da vizinhança efectua a descoberta nos nós vizinhos e o mecanismo distribuído tem de fazer a descoberta na rede toda. Em relação aos tempos de descoberta, no mesmo mecanismo de descoberta, pode-se ver que são independentes



**Figura 28: Tempos de descoberta e extensão para uma rede 4x4. *Scenario 1* - Local, *Scenario 2* - Vizinhança, *Scenario 3* – Distribuído**

do número de VNs e do número de utilizadores na rede: tal deve-se ao facto de trabalharem em canais separados, não tendo por isso interferência com o tráfego da rede, nem entre os mecanismos de descoberta. O mesmo motivo leva ao facto de os tempos de descoberta nos três canais, no mesmo mecanismo de descoberta, serem semelhantes.

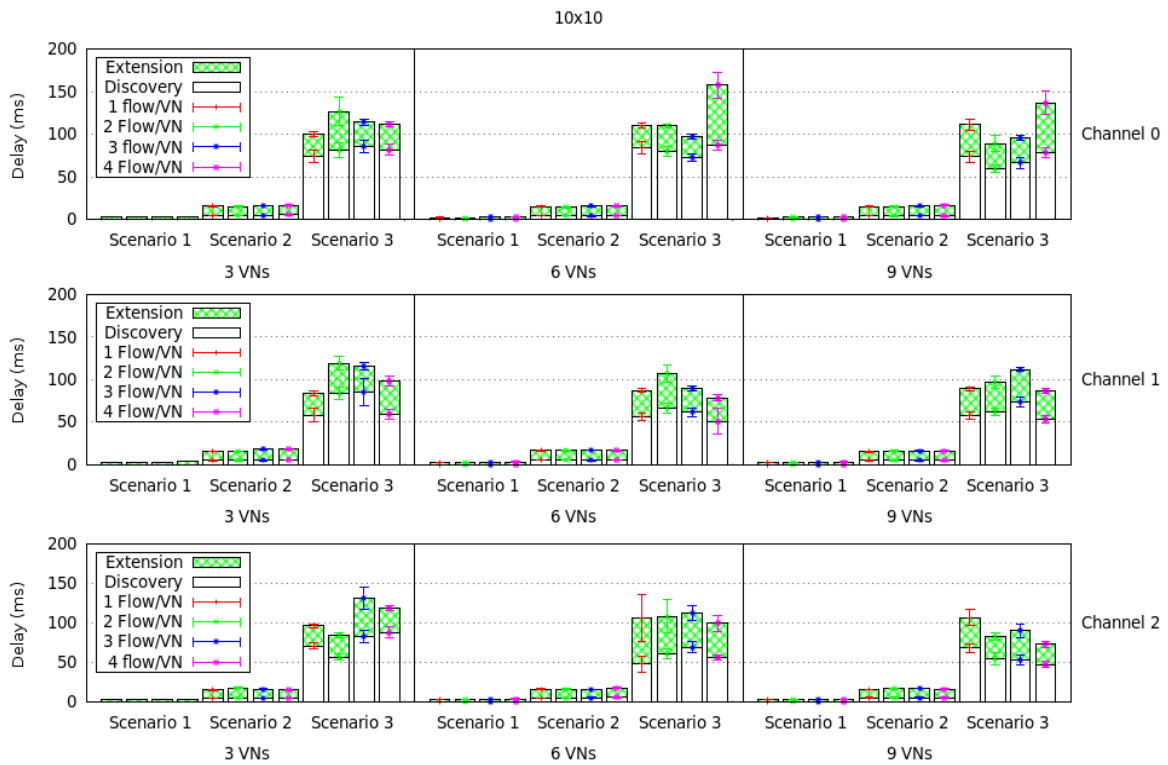
Em relação aos tempos de extensão pode-se ver que, no mesmo canal, o tempo de extensão para o mecanismo local é o menor, seguindo-se o mecanismo da vizinhança, e por fim o mecanismo distribuído que é o que possui maior tempo de extensão. Estes resultados são explicados com base da proximidade da VN pretendida. Para o mecanismo local é uma ligação de um salto, para o mecanismo da vizinhança é uma ligação de dois saltos, e para o mecanismo distribuído é uma ligação de três ou mais saltos. Em relação aos tempos de extensão, no mesmo mecanismo de descoberta, pode-se ver que aumentam com o aumento do número de VNs, e para o mesmo número de VNs aumenta com o aumento do número de utilizadores na rede. Tal deve-se ao aumento de tráfego na rede, que implica um atraso maior na nova ligação. Pode-se constatar que o tempo de extensão é mais elevado para o canal do tráfego sensível à



**Figura 29: Tempos de descoberta e extensão para uma rede 7x7. *Scenario 1* - Local, *Scenario 2* - Vizinhança, *Scenario 3* – Distribuído**

largura de banda, no caso de 9 VNs e 4 fluxos por VN, visto que a rede já está saturada (como se pode ver na *Figura 16* do cenário anterior). Os tempos de extensão, no mesmo mecanismo de descoberta, diminuem do canal destinado ao tráfego sensível à largura de banda, para o canal destinado ao tráfego sensível ao atraso. Este facto é esperado devido às características do tráfego, como já explicado em pormenor aquando da avaliação da virtualização no cenário anterior.

Comparando agora, a rede 4x4 com a rede 7x7 e com a rede 10x10, pode-se verificar que com o aumento da rede, os tempos de descoberta no mecanismo local e no mecanismo da vizinhança mantêm-se, enquanto no mecanismo distribuído os tempos aumentam. Esta diferença explica-se pelo facto de o mecanismo distribuído ser um mecanismo global distribuído na rede, e por isso com o aumento da rede, aumenta a complexidade da procura e conseqüentemente o tempo da descoberta. Em relação os tempos de extensão, estes aumentam com o aumento do tamanho da rede no caso de associados ao mecanismo distribuído, e são semelhantes no caso do mecanismo local e do mecanismo da vizinhança. Tal deve-se ao facto de o número de saltos da extensão



**Figura 30: Tempos de descoberta e extensão para uma rede 10x10. *Scenario 1 - Local, Scenario 2 - Vizinhaça, Scenario 3 – Distribuído***

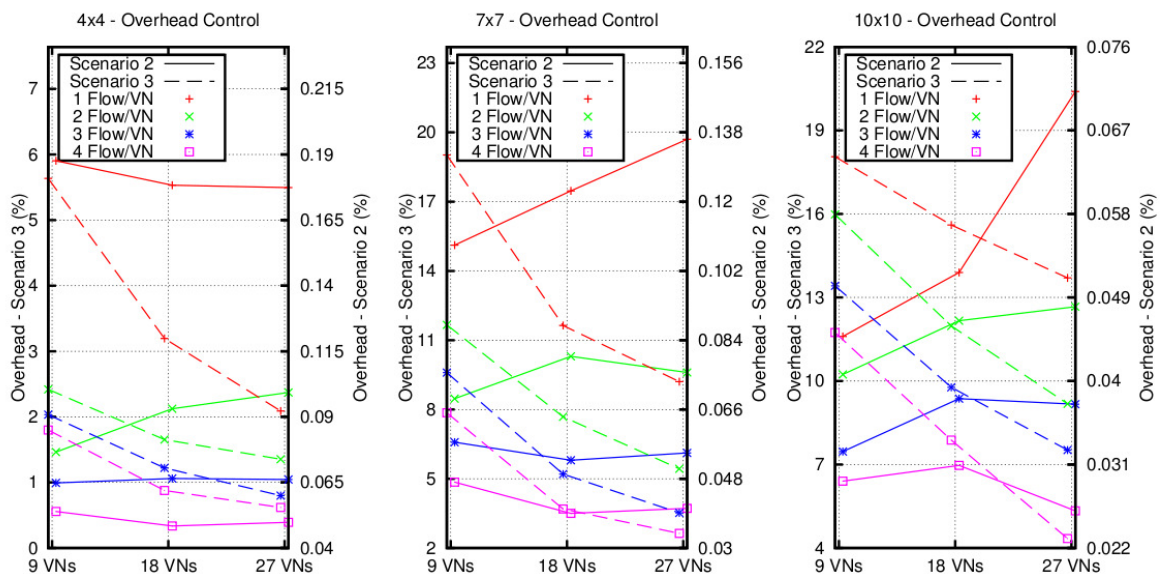
aumentar no mecanismo distribuído, enquanto nos outros mecanismos se mantém. De revelar, que quando a rede está saturada, o tempo de extensão é mais elevado (comparando com o cenário anterior).

Com base nestes resultados, poder-se-á afirmar que os mecanismos de descoberta propostos funcionam bem em termos de escalabilidade, uma vez que os diferentes parâmetros avaliados aumentam os seus valores apenas no caso do mecanismo distribuído, e tal era inevitável dado que o mecanismo de descoberta é global e passa a conter muito mais informação.

## 5.4.2 Impacto do controlo

O parâmetro *Overhead* é importante para se avaliar a carga gerada na rede pelos mecanismos de controlo. A *Figura 31* apresenta o *Overhead* de controlo variando o tamanho da rede, o número de VNs e o número de utilizadores por VN em termos de quantidade de fluxo. Nos gráficos apresentados, as linhas a tracejado estão associadas à escala da esquerda (*Overhead - Scenario 3*), e as linhas a cheio estão associadas à escala da direita (*Overhead – Scenario 2*).





**Figura 31: Overhead de controlo do *Scenario 2* – Vizinhança, e do *Scenario 3* – Distribuído, para diferentes tamanhos da rede.**

Os resultados apresentados na *Figura 31* mostram que o *Overhead* do mecanismo de descoberta distribuído (*Bamboo*) é muito superior ao do mecanismo de descoberta da vizinhança (*Discover VN*). Este resultado era esperado pois o mecanismo de descoberta distribuído é global e é efectuado em toda a rede, enquanto o mecanismo de descoberta da vizinhança só é efectuado nos seus vizinhos. O *Overhead* do mecanismo distribuído diminui com o aumento do número de VN's, e para o mesmo número de VN's, diminui com o aumento do número de utilizadores na rede. Quanto ao *Overhead* do mecanismo da vizinhança, este aumenta ou mantém-se com o aumento do número de VN's, e para o mesmo número de VN's, diminui com o aumento do número de utilizadores na rede. Estes resultados podem ser explicados pelo facto de, para o mesmo número de VN's, o aumento de utilizadores, como está traduzido em fluxos, não gera mais pesquisas, logo aumenta a quantidade de bytes de dados, mantendo a quantidade de bytes de controlo. Já para o aumento do número de VN's, o *Overhead* do mecanismo da vizinhança aumenta devido ao aumento do número de pesquisas na rede, aumentando mais a quantidade de bytes de controlo do que a quantidade de bytes de dados. O *Overhead* do mecanismo distribuído diminui devido ao aumento do tráfego na rede, aumentando mais a quantidade de bytes de dados do que a quantidade de bytes de controlo, visto que grande parte do tráfego de controlo do mecanismo distribuído é para manter a estrutura da rede, e não para efectuar as pesquisas propriamente ditas.

A *Tabela 7* apresenta o número de pesquisas efectuadas (expressadas em percentagem) para cada um dos diferentes mecanismos de busca, variando o tamanho

da rede, o número de VNs e o número de utilizadores por VN em termos de quantidade de fluxo.

Tamanho da rede	4X4			7x7			10x10		
VNs	9	18	27	9	18	27	9	18	27
Local	6.94%	6.25%	6.25%	2.72%	2.95%	2.80%	2.11%	1.83%	1.96%
Vizinhança	11.11%	10.76%	10.65%	4.99%	5.33%	5.14%	3.78%	3.33%	3.41%
Distribuído	81.94%	82.99%	83.10%	92.29%	91.72%	92.06%	94.11%	94.83%	94.63%

**Tabela 7: Número de pesquisas feitas (exprimidas em %) para cada um dos diferentes mecanismos de busca**

Os resultados apresentados mostram que a grande maioria das descobertas são efectuadas pelo mecanismo distribuído, e que esta percentagem vai aumentando com o aumento do tamanho da rede. Pode-se também verificar que para o mesmo tamanho da rede, a percentagem das pesquisas efectuadas por cada um dos mecanismos de descoberta não varia com o número de VNs.

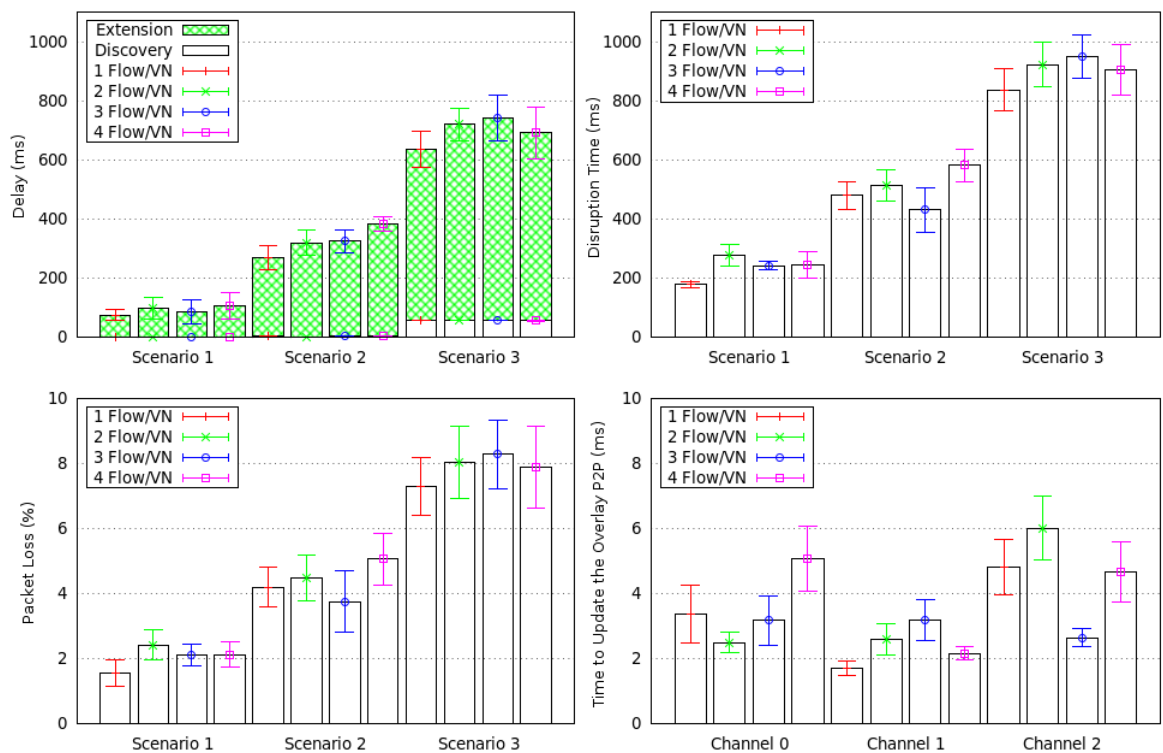
## 5.5 Influência da mobilidade dos utilizadores

Outra característica muito importante da solução proposta é avaliar o desempenho da solução de controlo com a mobilidade, através de uma rede dinâmica, que à medida que o utilizador se move, possibilita uma nova reconfiguração da rede.

Neste cenário, o tamanho da rede é de 7x7 nós na rede e de 6 VNs em cada interface de dados (18 VNs no total). Os nós movem-se aleatoriamente dentro da rede, tanto ao nível da posição para onde se deslocam, como ao nível do instante de tempo em que se decidem começar a mover, só com a restrição de que a posição para onde se movem corresponde a dois saltos físicos na rede (os nós movem-se sempre à mesma velocidade de 15 m/seg).

A *Figura 32* apresenta os resultados obtidos quando se varia o número de utilizadores por VN num cenário de mobilidade.

Comparando o gráfico da esquerda em cima da *Figura 32* com os gráficos da *Figura 29* para 6 VNs (18VNs na rede), pode-se constatar que a mobilidade não tem influência nos mecanismos de descoberta. Tal deve-se ao facto de os mecanismos de descoberta trabalharem em interfaces separadas das interfaces dos dados, com largura de banda suficiente para suportarem o aumento do número de pesquisas provocadas pela mobilidade. Já quanto aos mecanismos de extensão/reconfiguração, a mobilidade provoca um grande aumento dos tempos. Esta situação deve-se ao facto de várias



**Figura 32: Tempos de descoberta e extensão (esquerda em cima), tempo de restabelecimento de conectividade (direita em cima), *packet loss* (esquerda em baixo), tempo de actualização do P2P overlay (direita em baixo) para uma rede 7x7 e 18 VNs (6 VNs por canal). *Scenario 1* - Local, *Scenario 2* - Vizinhança, *Scenario 3* – Distribuído**

reconfigurações poderem ser desencadeadas ao mesmo tempo na rede, provavelmente no mesmo nó físico e nas mesmas ligações físicas. Esse facto, associado ao facto das VNs aumentarem muito o seu tamanho (em saltos), provoca o aumento do número de colisões, congestionando mais a rede e consequentemente dificultando o estabelecimento de novas extensões.

No gráfico da direita em cima da Figura 32 pode-se ver o tempo médio de restabelecimento da ligação, que é o tempo que vai desde que o utilizador perdeu a ligação devido à sua mobilidade até se ligar de novo com a aplicação ao qual estava antes ligado (através da VN). Pode-se ver que o gráfico é parecido com o gráfico da esquerda em cima da Figura 32. Tal situação deve-se ao facto de a soma do tempo de extensão com o tempo dispendido na descoberta serem a maior parcela do tempo de restabelecimento, uma vez que o tempo que demora desde que perde a ligação até iniciar o mecanismo de busca é muito diminuto.

No gráfico da esquerda em baixo da Figura 32 pode-se ver o número de pacotes perdidos pela aplicação devido à perda de ligação do utilizador, o qual é directamente

proporcional ao tempo de restabelecimento de conectividade, e por isso os gráficos possuem uma forma parecida (também se deve a estar-se a considerar fluxos constantes). Este gráfico mostra que, embora os tempos de extensão tenham aumentado significativamente, esse valor não se reflecte nas perdas, visto estas serem inferiores a 10% e por isso poderem ser consideradas aceitáveis. Pode-se por isso dizer que a nossa solução reage bem à mobilidade tal como se pretendia.

No gráfico da direita em baixo da *Figura 32* pode-se ver o tempo de actualização do P2P *overlay* de controlo. Este tempo apenas considera a mudança da chave na rede, visto a actualização das tabelas de encaminhamento serem periódicas no mecanismo global utilizado (*Bamboo*). Sendo assim, seria de esperar que o tempo obtido fosse pequeno. Pode-se verificar que os tempos obtidos não variam consoante o canal, nem com o número de utilizadores na rede. Tal deve-se ao facto da actualização da posição da chave se efectuar na interface de controlo, independente do tráfego de dados. Os tempos são um pouco diferentes uns dos outros devido ao tempo de actualização depender da distância dos nós no *overlay* (podem estar perto fisicamente e longe no *overlay*).

A *Tabela 8* apresenta o número de pesquisas efectuadas (exprimidas em percentagem) para cada um dos diferentes mecanismos de busca.

	1 Fluxo/VN	2 Fluxos/VN	3 Fluxos/VN	4 Fluxos/VN
Local	24.45%	28.33%	26.95%	26.94%
Vizinhança	9.72%	7.50%	15.83%	14.44%
Distribuído	65.83%	64.17%	57.22%	58.61%

**Tabela 8: Número de pesquisas feitas (exprimidas em %) para cada um dos diferentes mecanismos de busca em função do número de utilizadores por VN para uma rede 7x7 com 18 VNs**

Comparando os resultados apresentados na *Tabela 8*, com os resultados da *Tabela 7*, pode-se constatar que no caso da mobilidade o número de pesquisas efectuadas pelo mecanismo distribuído é menor, o que provoca um aumento do número de pesquisas locais e na vizinhança. Tal deve-se ao facto de na mobilidade o utilizador se deslocar para um nó próximo, em que a probabilidade desse nó, ou os seus vizinhos, conterem a VN a que estava ligado, é muito maior. Os resultados apresentados na *Tabela 8* mostram ainda que a percentagem das descobertas serem efectuadas pelo mecanismo distribuído vai diminuindo com o aumento do número de utilizadores na rede. Nesta situação, com o aumento do número de utilizadores, as VNs são maiores, e por isso a probabilidade de encontrar a VN num determinado nó, ou nos seus vizinhos, também é maior.

A *Tabela 9* apresenta o número de vezes que o P2P *overlay* de controlo foi actualizado (exprimidas em percentagem em relação ao número de vezes que os mecanismos de descoberta foram executados) para cada um dos diferentes canais, assim como para a rede (a média dos diferentes canais).

	1 Fluxo/VN	2 Fluxos/VN	3 Fluxos/VN	4 Fluxos/VN
Canal 0	70.00%	69.17%	67.50%	66.67%
Canal 1	73.33%	70.83%	65.00%	66.67%
Canal 2	68.33%	63.33%	62.50%	62.83%
Total	70.56%	67.78%	65.00%	66.39%

**Tabela 9: Número de vezes que o P2P *overlay* de controlo foi actualizado (exprimidas em %) para cada um dos diferentes canais, assim como o total dos diferentes canais, em função do número de utilizadores por VN para uma rede 7x7 com 18 VNs (6 VNs por canal)**

Os resultados apresentados mostram que o número de vezes que o P2P *overlay* é actualizado em relação ao número de vezes possíveis (que corresponde a todas as vezes que os mecanismos de descoberta foram executados) é elevado. Estes resultados devem-se ao facto de ser necessário actualizar o P2P *overlay* de controlo sempre que o mecanismo de descoberta distribuído é utilizado. Ao comparar com a *Tabela 8*, verifica-se que a percentagem do número de vezes que o P2P *overlay* de controlo é actualizado, é proporcional e maior que o número de vezes que o mecanismo de descoberta distribuído é executado.

Os resultados apresentados na *Tabela 9* mostram ainda que a percentagem do número de actualizações do P2P *overlay* de controlo vai diminuindo com o aumento do número de utilizadores na rede. Este facto pode ser explicado devido à percentagem de vezes que o mecanismo distribuído é executado também diminuir. Ou seja, com o aumento de utilizadores, as VNs são maiores, e por isso a probabilidade de encontrar a VN num determinado nó, ou nos seus vizinhos, também é maior, não necessitando de actualizar tantas vezes o P2P *overlay* de controlo.

## 5.6 Discussão

Os resultados obtidos, através da realização de diferentes simulações sobre diferentes cenários, estão de acordo com o esperado, na maioria dos casos.

Quanto à influência da virtualização no desempenho da rede, a avaliação realizada demonstrou que a virtualização da rede possibilita maior carga na rede (satura mais tarde), devido à menor carga por interface provocar menos colisões. Demonstra

também um ligeiro aumento no atraso, quando não há saturação, justificado pela menor largura de banda disponível para cada interface, que é compensado por um menor atraso.

Verificou-se também que as diferentes interfaces estão bem dimensionadas para o tipo de tráfego baseado em contexto que suportam (tráfego sensível à largura de banda, ao atraso, ao nível de actividade dos utilizadores), visto que possuem as características dos parâmetros de contexto desejado pelos utilizadores.

Quanto à influência do número de VNs na rede, verificou-se, como já esperado, que o aumento do número de VNs na rede aumenta ligeiramente o atraso e a perda de pacotes, diminuindo um pouco o desempenho na rede.

Quanto à influência do número de utilizadores na rede para cada VN, constatou-se, como já esperado, que o aumento do número de utilizadores na rede provoca mais atrasos e mais perda de pacotes, diminuindo o desempenho na rede.

Comparando o tráfego através do protocolo UDP com o tráfego através do protocolo TCP, verificou-se que através do protocolo UDP, a nossa solução funciona bem em termos de escalabilidade, enquanto através do protocolo TCP já não funciona tão bem. Tal situação deve-se ao facto de, pela definição, o protocolo TCP não ter perdas e por isso possuir confirmações e retransmissões que aumentam a carga na rede. Esta sobrecarga na rede aumenta com o tamanho da rede, pois aumenta a distância entre os nós móveis e conseqüentemente o número de ligações que o tráfego precisa de atravessar. É de constatar que não é um problema da virtualização, pois sem virtualização este fenómeno também se verifica.

Os resultados dos mecanismos de descoberta revelam, como era esperado, que o tempo de descoberta no mecanismo de busca local é menor do que o tempo de descoberta no mecanismo da vizinhança (*Discover VN*), este é inferior ao tempo de descoberta no mecanismo da vizinhança é inferior ao tempo de descoberta no mecanismo distribuído (*Bamboo*). O mecanismo local só trabalha no próprio nó, e por isso é mais rápido na descoberta que o mecanismo da vizinhança que efectua a descoberta nos nós vizinhos. Por sua vez, este torna-se mais rápido que o mecanismo distribuído, que tem de fazer a descoberta na rede toda. Verifica-se também que a grande maioria das descobertas são efectuadas pelo mecanismo distribuído, e que esta percentagem vai aumentando com o aumento do tamanho da rede. Pode-se também verificar que para o mesmo tamanho da rede, a percentagem das pesquisas efectuadas por cada um dos mecanismos de descoberta não varia com o número de VNs.

Em termos de escalabilidade verificou-se que os tempos de descoberta no mecanismo local e no mecanismo da vizinhança não são afectados, visto que continuam a fazer a pesquisa no próprio nó e nos vizinhos respectivamente. Já o mecanismo

distribuído, como tem que efectuar a pesquisa em toda a rede, e esta aumenta, vai demorar por si só mais tempo.

Em termos de variação do número de VNs e do número de utilizadores dentro de cada VN, verificou-se que tal não afecta os tempos de descoberta, visto que os mecanismos de descoberta funcionam em interfaces separadas independentes do tráfego de dados na rede.

Em relação aos tempos de extensão das VNs, estes aumentam com o aumento do número de VNs. Para o mesmo número de VNs, estes aumentam com o aumento do número de utilizadores. Tal deve-se ao facto de as extensões serem feitas nas interfaces de dados e por isso estarem sujeitas ao tráfego na rede, aumentando muito em caso de saturação.

Os tempos de extensão no mecanismo local são menores do que no mecanismo da vizinhança e menores deste para o mecanismo distribuído. Tal deve-se ao facto de a extensão efectuada para o mecanismo local ser apenas de um salto, para o mecanismo da vizinhança ser apenas de dois saltos, e para o mecanismo distribuído ser mais de dois saltos.

Em relação ao *Overhead* na rede, o mecanismo de descoberta da vizinhança tem um *Overhead* muito baixo, pois em percentagem esse mecanismo tem um uso muito baixo. Já o mecanismo de descoberta distribuído tem um *Overhead* muito alto. De notar que o mecanismo distribuído gera muito tráfego para manutenção da estrutura da rede. É por isso que com o aumento do tráfego de dados, o *Overhead* do mecanismo distribuído diminui, pois as pesquisas não aumentam muito o tráfego de controlo. Com o aumento da rede, tal como esperado, o mecanismo distribuído necessita de gerar mais tráfego para manutenção da estrutura da rede, e por isso o *Overhead* aumenta.

Os resultados da influência da mobilidade mostram, como era esperado, que a mobilidade não tem influência nos tempos dos mecanismos de descoberta, visto estes trabalharem em interfaces separadas das interfaces de dados. Já quanto aos mecanismos de extensão/reconfiguração, a mobilidade provoca um grande aumento dos tempos. Esta situação deve-se ao facto de várias reconfigurações poderem ser desencadeadas ao mesmo tempo na rede, provavelmente no mesmo nó físico e nas mesmas ligações físicas. Esse facto, associado ao facto das VNs aumentarem muito o seu tamanho (em saltos), provoca o aumento do número de colisões, congestionando mais a rede e consequentemente dificultando o estabelecimento de novas extensões.

Com a introdução da mobilidade a percentagem do número de pesquisas efectuadas pelo mecanismo distribuído diminui, mas continuam a ser a maior parte das pesquisas efectuadas na rede. Tal deve-se ao facto de, na mobilidade o utilizador se

deslocar para um nó próximo, em que a probabilidade desse nó, ou dos seus vizinhos, conterem a VN a que estava ligado, é muito maior.

Em relação ao tempo de restabelecimento de conectividade, este é proporcional à soma do tempo de descoberta com o tempo de extensão, visto que estes dois tempos são a maior parcela do tempo de restabelecimento. O tempo que demora desde que perde a ligação até iniciar o mecanismo de busca é muito diminuto.

O número de pacotes perdidos devido à mobilidade é directamente proporcional ao tempo de restabelecimento de conectividade e, embora os tempos de extensão tenham aumentado significativamente, as perdas são inferiores a 10%.

Quanto ao tempo de actualização do P2P *overlay* de controlo, como apenas considera a mudança da chave na rede, este tempo é pequeno. Este tempo não varia consoante o canal, nem com o número de utilizadores na rede. Tal deve-se ao facto de a actualização da posição da chave se efectuar na interface de controlo, independentemente do tráfego de dados que possa existir.

O número de vezes que o P2P *overlay* é actualizado em relação ao número de vezes possíveis é elevado. Tal deve-se ao facto de sempre que o mecanismo de descoberta distribuído for utilizado, é necessário actualizar o P2P *overlay* de controlo (e esse mecanismo representa a maior parte das pesquisas entretanto efectuadas).

## 5.7 Conclusão

Este capítulo apresentou os cenários e os testes utilizados para avaliar a solução proposta nesta dissertação. A análise dos resultados permitiu concluir que a solução apresentada é eficiente em termos dos parâmetros de contexto utilizados, e que os mesmos estão de acordo com os esperados.

Foi avaliada a influência da virtualização na solução proposta, e a mesma revelou que a introdução da virtualização permite uma independência total entre os diferentes grupos de utilizadores, agrupados de acordo com características dos parâmetros de contexto semelhantes.

Foram avaliados dois mecanismos de descoberta na rede, um local que faz a procura na vizinhança e um distribuído que faz a procura na rede toda. Estes mecanismos possibilitam aos utilizadores encontrarem a rede virtual que melhor se enquadra nas suas preferências, de entre as redes virtuais disponíveis.

A solução proposta foi também avaliada em termos da mobilidade dos utilizadores, nomeadamente o tempo de recuperação da conectividade, a actualização dos mecanismos de descoberta e o tempo de descoberta das redes virtuais pretendidas.



## 6. Conclusões e linhas futuras de investigação

Nesta dissertação foi desenvolvida e avaliada uma nova abordagem de rede que permite ter redes em malha sem fios muito adaptativas através do suporte de múltiplas redes lógicas, baseadas em informação de contexto. Para tal foi usada a virtualização que possibilita a programabilidade e flexibilidade necessárias para se poder configurar/adaptar os recursos/topologias das redes virtuais mediante a variação de contexto, sendo imprescindível a utilização de um mecanismo que permita gerir toda a rede tanto a nível local como global.

A arquitectura foi desenvolvida e testada num ambiente de simulação utilizando o NS-2. O conceito de virtualização, assim como os mecanismos de gestão e controlo, foram implementados, e diversos cenários foram criados para testar o comportamento da arquitectura, em diferentes situações.

A solução proposta permite que cada utilizador ou aplicação se ligue à VN que melhor se adapte às suas características de contexto, possibilitando ainda que, ao variar o contexto do utilizador ou aplicação, os recursos/topologias associados sejam adaptados ao contexto.

O uso da virtualização demonstra que, para além de permitir uma caracterização efectiva dos recursos/topologias, também aumenta o desempenho, permitindo maior carga na rede.

O uso do mecanismo de descoberta local demonstrou um desempenho muito superior ao do mecanismo global, como se esperava, sendo que a maioria das descobertas necessitam de ser efectuadas pelo mecanismo global distribuído e por isso possui também um *Overhead* muito superior.

A introdução da mobilidade na rede provoca um aumento dos tempos de reconfiguração, devido ao facto de várias reconfigurações poderem ser desencadeadas ao mesmo tempo na rede, provavelmente no mesmo nó físico e nas mesmas ligações físicas.

No final deste trabalho verifica-se que ainda existe muito trabalho a desenvolver nesta área: os vários mecanismos devem ser melhorados/desenvolvidos, para melhorar a eficiência, gestão e controlo da rede. Em primeiro lugar, é necessário melhorar o mecanismo de selecção de rede, para ter em conta o compromisso entre o custo do ajuste perfeito entre o número de redes virtuais possíveis e a qualidade de experiência realmente oferecida ao utilizador ou aplicação. Em segundo lugar, é necessário desenvolver um mecanismo capaz de, sem perda de conectividade, adaptar dinamicamente os recursos/topologias das VNs existentes, tendo em conta as alterações de parâmetros de contexto e as preferências de um utilizador, os recursos necessários

para as aplicações dos utilizadores, e a mobilidade dos mesmos. Em terceiro lugar, é necessário melhorar o mecanismo de mobilidade implementado, para permitir mais padrões de mobilidade dos utilizadores (os utilizadores poderem variar a velocidade de deslocação), e para prever a deslocação do utilizador, de modo a não haver perda de conectividade. É também importante considerar outros parâmetros de contexto, tais como disponibilidade, confiança, custo, etc.

Para aumentar a escalabilidade, é imperativo hierarquizar os mecanismos de gestão da rede, de modo a diminuir o *Overhead* de controlo, assim como os tempos de descoberta.

## Bibliografia

1. **I.F. Akyildiz, X.Wang.** A survey on wireless mesh networks. *IEE Communications Magazine*. 2005. Vols. 43, no.9, pp. S23-S30.
2. **T.Anderson, L.Peterson, S.Shenker, J.Turner.** Overcoming the Internet impasse through virtualization. *IEEE Computer Society*. April, 2005.
3. **R.Matos, S.Sergento, K.Hummel, A.Hess, K.Tutschku, H.Meer.** Context-based wireless mesh networks: A case for network virtualization. *Springer Telecommunication Systems Journal*. s.l. : Accepted for Publication, 2010.
4. **R.Matos, S.Sergento.** Context-aware connectivity and mobility in wireless mesh networks. *in ICST MOMAMI*. October 2009.
5. **Ricardo Matos.** Context-based wireless mesh networks. *PhD Thesis Proposal*. University of Aveiro,' Department of Electronics, Telecommunications and Informatics', March 2010.
6. **T. Huovila\*, P. Lassila‡, J. Manner\*, and A. Penttinen‡.** State of the Art Analysis of Wireless Mesh Technologies. *ABI project technical report*. ‡University of Helsinki, \*Helsinki University of Technology, November 2006.
7. IEEE 802.11. <http://www.ieee802.org/11/>. July 2010.
8. IEEE 802.16. <http://www.ieee802.org/16/>. July 2010.
9. **J. Bicket, D. Aguayo, S. Biswas, and R. Morris.** Architecture and evaluation of an unplanned 802.11b mesh network. *In Proceedings of ACM MobiCom*. August 2005. pp. 31 - 42.
10. **D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris.** Link-level measurements from an 802.11b mesh network. *In Proceedings of ACM SIGCOMM*. August 2004. pp. 121 - 131.
11. Mesh@Purdue - Purdue university wireless mesh network testbed. <http://calmesh.calit2.net/index.html>. 2006.
12. **S. M. Das, H. Pucha, D. Koutsonikolas, Y. C. Hu, and D. Peroulis.** DMesh: incorporating practical directional antennas in multi-channel wireless mesh networks. *To appear in IEEE Journal on Selected Areas in Communications special issue on Multi-Hop Wireless Mesh Networks*. 2005.
13. **H. Lundgren, K. Ramachandran E. M. Belding-Royer, K. Almeroth, M. Benny, A. Hewatt, A. Touma, and A. Jardosh.** Experiences from the design, deployment, and usage of the UCSB MeshNet testbed. *IEEE Communications Magazine*. April 2006. Vols. 44, no. 4, pp. 18 - 29.

14. **K. N. Ramachandran, M. M. Buddhikot, G. Chandranmenon, S. Miller, E. M. Belding-Royer, and K. C. Almeroth.** On the design and implementation of infrastructure mesh networks. *In Proceedings of First IEEE Workshop on Wireless Mesh Networks.* September 2005.
15. **A. Raniwala and T. Chiueh.** Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. *In Proceedings of IEEE INFOCOM.* March 2005. pp. 2223 - 2234.
16. **B.-N. Cheng, S. Kalyanaraman, and M. Klein.** A geography -ware scalable community wireless network testbed. *In Proceedings of 1st International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities.* February 2005. pp. 82 - 91.
17. **V. Navda, A. Kashyap, and S. R. Das.** Design and evaluation of iMesh: an infrastructure-mode wireless mesh network. *In Proceedings of 6th IEEE International Symposium on aWorld ofWireless Mobile and Multimedia Networks.* June 2005. pp. 164 - 170.
18. **D.Ñiculescu, S. Ganguly, K. Kim, and R. Izmailov .** Performance of VoIP in a 802.11 wireless mesh network. *In Proceedings of IEEE INFOCOM.* April 2006.
19. **V. Sridhara, J. Kim, and S. Bohacek.** Performance of urban mesh networks. *In Proceedings of ACM MSWiM.* October 2005. pp. 269 - 277.
20. Calmesh. <http://calmesh.calit2.net>. June 2010.
21. **R. B. Dilmaghani, B. S. Manoj, B. Jafarian, and R. R. Rao.** Performance evaluation of RescueMesh: a metro-scale hybrid wireless network. *In Proceedings of First IEEE Workshop on Wireless Mesh Networks.* September 2005.
22. P3M Mesh Network. <http://www.meshdynamics.com/military-mesh-networks.html>. June 2010.
23. **N.Feamster, L.Gao, J.Rexford.** How to lease the internet in your spare time. *SIGCOMM Computer Communication Review.* 2007. Vols. 37, no. 1, pp. 61-64.
24. **P.Ferguson, G.Huston.** What is a VPN? *Cisco Systems, Tech. Rep.* 1998.
25. **E.Rosen, Y.Rekhter.** BGP/MPLS VPNs. *RFC 2547.* March 1999.
26. BGP/MPLS IP Virtual Private Networks (VPNs). *RFC 4364.* February 2006.
27. **L.Andersson, T.Madsen.** Provider Provisioned Virtual Private Network (VPN) Terminology. *RFC 4026.* March 2005.
28. **A.T.Campbell, H.G.D.Meer, M.E.Kounavis, K.Miki, J.B.Vicente, D.Villela.** A survey of programmable networks. *SIGCOMM Computer Communication Review.* 1999. Vols. 29, no. 2, pp. 7-23.

29. **N.M. Mosharaf Kabir Chowdhury, Raouf Boutaba.** A Survey of Network Virtualization. *Technical Report CS-2008-25*. University of Waterloo, Oct. 2008.
30. **N. McKeown et al.** Openflow: Enabling innovation in campus networks. *ACM SIGCOMM Comput. Commun. Rev.* 2008. Vols. 38, no. 2, pp. 68-74.
31. **J. Naous, G. Gibb, S. Bolouki, N. Mc Keown.** NetFPGA: reusable router architecture for experimental research. *ACM PRESTO*. 2008. pp. 1-7.
32. **R. Morris, E. Kohler, J. Jannotti, M. F. Kaashoek.** The Click modular router. *ACM SIGOPS Oper. Syst. Rev.* 1999. Vols. 33, no. 5, pp. 217–231.
33. **S. Karlin and L. Peterson.** Vera: an extensible router architecture. *Elsevier Comput. Netw.* 2002. Vols. 38, no. 3, pp. 277–293.
34. **M. Handley, O. Hodson, E. Kohler.** XORP: an open platform for network research. *ACM SIGCOMM Comput. Commun. Rev.* 2003. Vols. 33, no. 1, pp. 53–57.
35. **M. Handley et al.** Designing extensible IP router software. *USENIX NSDI*. 2005. pp. 189–202.
36. **N. Egi et al.** Evaluating Xen for Router Virtualization. *ICCCN*. 2007.
37. A flexible and performant virtual router. *Poster in IWSOS*. 2007.
38. Fairness issues in software virtual routers. *ACM PRESTO*. 2008. pp. 33–38.
39. Towards high performance virtual routers on commodity hardware. *ACM CoNEXT*. 2008. pp. 1–12.
40. A platform for high performance and flexible virtual routers on commodity hardware. *ACM SIGCOMM Comput. Commun. Rev.* 2010. Vols. 40, no. 1, pp. 127–128.
41. XEN VMM. <http://www.cl.cam.ac.uk/research/srg/netos/xen/>. July 2010.
42. Madwifi. <http://madwifi.org/>. June 2010.
43. Projects of Madwifi. <http://madwifi-project.org/wiki/Publicity>. June 2010.
44. **Paramvir Bahl, Pradeep Bahl, and Ranveer Chandra.** MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card. *Technical Report MSR-TR-2003-46*. August 2003.
45. **J. Touch, S. Hotz.** The X-Bone. *In Proceedings of the Third Global Internet Mini-Conference at GLOBECOM'98*. 1998. pp. 44-5.
46. **J. D. Touch, Y.-S. Wang, L. Eggert, G. Finn.** A virtual internet architecture. *Tech. Rep. TR-570*. USC/Information Sciences Institute, 2003.
47. **J. E. van der Merwe, S. Rooney, I. Leslie, S. Crosby.** The Tempest - A practical framework for network programmability. *IEEE Network Magazine*. 1998. Vols. 12, no. 3, pp. 20-28.

48. **J. E. van der Merwe, I. M. Leslie.** Switchlets and dynamic virtual ATM networks. *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management (IM'97)*. 1997. pp. 355-368.
49. User controlled lightpaths project. <http://uclp.uwaterloo.ca/>. U. of Waterloo
50. **J. Recio, E. Grasa, S. Figuerola, G. Junyent.** Evolution of the user controlled lightpath provisioning system. *Proceedings of 7th International Conference on Transparent Optical Networks*. July 2005. Vol. 1, pp. 263-266.
51. **B. Nandy, D. Bennett, I. Ahmad, S. Majumdar, B. St.Arnaud.** User controlled lightpath management system based on a service oriented architecture. <http://www.solananetworks.com/UCLP/files/UCLPv2-SOA.pdf>. June 2010.
52. Virtuoso: Resource management and prediction for distributed computing using virtual machines. <http://virtuoso.cs.northwestern.edu/>.
53. **A. Sundararaj, P. Dinda.** Towards virtual networks for virtual machine grid computing. *Proceedings of the 3rd USENIX Virtual Machine Research and Technology Symposium (VM'04)*. 2004.
54. Parallel Internets framework. *AGAVE Deliverable (Id: AGAVE/WP1/FTRD/D1.1/public)*. 2006.
55. A framework for end-to-end service differentiation: Network planes and parallel Internets. *IEEE Communications*. September. Vols. 45, no. 9, pp. 134-143.
56. **N. Wang, D. Griffin, J. Spencer, J. Griem, J. R. Sanchez, M. Boucadair, E. Mykoniati, B. Quoitin, M. Howarth, G. Pavlou, A. J. Elizondo, M. L. G. Osma, P. Georgatsos.** A framework for lightweight QoS provisioning: Network planes and parallel Internets. *Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management (IM'07)*. 2007. pp. 797-800.
57. **X. Jiang, D. Xu.** VIOLIN: Virtual internetworking on overlay infrastructure. *Tech. Rep. TR-03-027*. Purdue University, 2003.
58. **P. Ruth, X. Jiang, D. Xu, S. Goasguen.** Virtual distributed environments in a shared infrastructure. *Computer*. 2005. Vols. 38, no. 5, pp. 63-69.
59. **A. Jun and A. Leon-Garcia.** A virtual network approach to network resources management. *Proceedings of the Canadian Conference on Broadband Research (CCBR'98)*. June 1998.
60. Virtual network resources management: A divide-and-conquer approach for the control of future networks. *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'98)*. 1998. Vol. 2, pp. 1065-1070.

61. **W. Ng, R. Boutaba, A. Leon-Garcia.** Provision and customization of ATM virtual networks for supporting IP services. *Proceedings of the IEEE ATM Workshop'1999*. 1999. pp. 205-210.
62. **P. Chandra, A. Fisher, C. Kosak, T. S. E. Ng, P. Steenkiste, E. Takahashi, H. Zhang.** Darwin: Customizable resource management for value-added network services. *IEEE Network*. 2001. Vols. 15, no. 1, pp. 22-35.
63. **S. da Silva, Y. Yemini, D. Florissi.** The NetScript active network system. *IEEE Journal on Selected Areas in Communication*. 2001. Vols. 19, no. 3, pp. 538-551.
64. **S. da Silva, D. Florissi, Y. Yemini.** NetScript: A language-based approach to active networks. *Tech. Rep.* Columbia University, January 1998.
65. **M. Kounavis, A. Campbell, S. Chou, F. Modoux, J. Vicente, H. Zhuang.** The Genesis Kernel: A programming system for spawning network architectures. *IEEE Journal on Selected Areas in Communications*. 2001. Vols. 19, no. 3, pp. 511-526.
66. **A. A. Lazar, A. T. Campbell.** Spawning networking architectures (White Paper). *Tech. Rep.* Columbia University, 1998.
67. **A. T. Campbell, M. E. Kounavis, D. A. Villela, J. Vicente, K. Miki, H. G. D. Meer, and K. S. Kalaichelvan.** Spawning networks. *IEEE Network Magazine*. 1999. Vols. 13, no. 4, pp. 16-30.
68. PlanetLab: An open platform for developing, deploying, and accessing planetary-scale services. <http://www.planet-lab.org/>. June 2010.
69. **L. Peterson, T. Anderson, D. Culler, T. Roscoe.** A blueprint for introducing disruptive technology into the Internet. *SIGCOMM Computer Communication Review*. 2003. Vols. 33, no. 1, pp. 59-64.
70. VINI: A virtual network infrastructure. <http://www.vini-veritas.net/>. June 2010.
71. **A. Bavier, N. Feamster, M. Huang, L. Peterson, J. Rexford.** In VINI veritas: Realistic and controlled network experimentation. *Proceedings of SIGCOMM'06*. New York, NY, USA : ACM, 2006. pp. 3-14.
72. GENI: Global Environment for Network Innovations. <http://www.geni.net/>.
73. **Group, G. P.** GENI design principles. *Computer*. 2006. Vols. 39, no. 9, pp. 102-105.
74. Group-GENI. *Technical document on wireless virtualization*. Sept. 2006.
75. **D. Raychaudhuri, M. Ott, and I. Secker.** ORBIT Radio Grid Tested for Evaluation of Next-Generation Wireless Network Protocols. *TRIDENTCOM*. 2005. pp. 308 - 309.
76. **I. Z. Y. R. D. Bhanage, G. Sesar.** Evaluation Of OpenVZ Based Wireless Testbed Virtualization. *Tech. Rep.* Rutgers University
77. **D. Giustiniano, E. Goma, A. Lopez, and P. Rodriguez.** WiSwitcher: an efficient client for managing multiple APs. *ACM PRESTO*. 2009. pp. 43 - 48.

78. **Dey, A.K.** Understanding and Using Context. *Personal and Ubiquitous Computing Journal*. 2001. Vol. 5 (1).
79. Ambient networks. <http://www.ambient-networks.org/>.
80. **I. Al-Oqily and A. Karmouch.** Policy-Based Context-Aware Overlay Networks. *GIIS*. 2007.
81. Automating Overlay Networks Management. *AINA*. 2007.
82. **B. Mathieu et al.** Self-Management of Context-Aware Overlay Ambient Networks. *IFIP/IEEE International Symposium on Integrated Network Management*. 2007.
83. **B. Mathieu, M. Song, A. Galis, L. Cheng, K. Jean, R. Ocampo, h. Lai, M. Brunner, M. Stiernerling, M. Cassini, and M. Kampmann.** Autonomic Management of Context-Aware Ambient Overlay Networks. *ChinaCom*. 2007.
84. **K. Jean, L. Cheng, R. Ocampo, and A. Galis.** Contextualisation of Management Overlays in Ambient Networks.
85. **L. Cheng, K. J. anf R. Ocampo, and A. Galis.** Service-aware Overlay Adaptation in Ambient Networks. *Proceedings of International Multi-Conference on Computing in the Global Information Technology*. 2006.
86. **Neto A., Sargento S., Logota E., Antoniou J., Pinto F.** Multiparty session and network resource control in the context casting (c-cast) project. *Proceedings of 2nd International Workshop on Future Multimedia Networking*. 2009.
87. C-CAST - Context casting. <http://www.ict-ccast.eu/>. June 2010.
88. **O. Riva, V.-M. Teittinen, S. Siikavirta, and L. Huovinen.** A Next Generation Operator Environment to Turn Context-Aware Services into a Commercial Reality. *IEEE MDM*. 2008. pp. 90 - 97.
89. **Guanling Chen and David Kotz.** A Survey of Context-Aware Mobile Computing Research. *Dartmouth Computer Science Technical Report TR2000-381*. November 2000.
90. **Peter J. Brown, John D. Bovey, and Xian Chen.** Context-aware applications: from the laboratory to the marketplace. *IEEE Personal Communications*. October 1997. Vols. 4, no.5, pp. 58 - 64.
91. **Jason Pascoe.** The Stick-e note architecture: Extending the interface beyond the user. *Proceedings of the 1997 International Conference on Intelligent User Interfaces*. January 1997. pp. 261 - 264.
92. ConteXtML. <http://www.cs.kent.ac.uk/projects/mobicomp/fnc/ConteXtML.html>. June 2010.
93. **Bill N. Schilit, Marvin M. Theimer, and Brent B. Welch.** Customizing mobile applications. *Proceedings of USENIX Mobile & Location-Independent Computing Symposium*. Cambridge, Massachusetts, August 1993. pp. 129 - 138.



94. **Geoffrey M. Voelker and Brian N. Bershad.** Mobisaic: An information system for a mobile wireless computing environment. *In Proceedings of IEEE Workshop on Mobile Computing Systems and Applications.* Santa Cruz, California, December 1994. pp. 185 - 190.
95. **Jean Bacon, John Bates, and David Halls.** Location-oriented multimedia. *IEEE Personal Communications.* October 1997. Vols. 4, no.5, pp. 48 - 57.
96. **Keith Cheverst, Keith Mitchell, and Nigel Davies.** Design of an object model for a context sensitive tourist GUIDE. *In Proceedings of Workshop on Interactive Applications of Mobile Computing.* Rostock, Germany, November 1998.
97. **Nigel Davies, Keith Cheverst, Keith Mitchell, and Adrian Friday.** Caches in the air: Disseminating tourist information in the GUIDE system. *In Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications.* New Orleans, Louisiana, February 1999.
98. **Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, and Walter Van de Velde.** Advanced interaction in context. *In Proceedings of First International Symposium on Handheld and Ubiquitous Computing.* Karlsruhe, Germany, September 1999. pp. 89 - 101.
99. **G. Manku.** Dipsea: A Modular Distributed Hash Table. *PhD thesis.* Stanford University, August 2004.
100. **D. Balakrishnan and A. Nayak.** CSON-D: Context dissemination in autonomous systems using overlays. *International Conference on Intelligent Environments.* July 2008.
101. An Efficient Context-Specific Pure Overlay Space for Context Dissemination in Ambient Networks. *IEEE CSE.* 2008. pp. 23 - 30.
102. **D. Balakrishnan, A. Nayak, and P. Dhar.** Towards a realistic context dissemination protocol using pure multi-level overlay networks. *IEEE ICON.* Dec. 2008.
103. **T. Gu, H. Pung, and D. Zhang.** A peer-to-peer overlay for context information search. *IEEE ICCCN.* Oct 2005. pp. 395 - 400.
104. **T. Gu, E. Tan, H. Pung, and D. Zhang.** A peer-to-peer architecture for context lookup. *IEEE MobiQuitous.* July 2005. pp. 333 - 341.
105. **A. Subramanian, M. Buddhikot, and S. Miller.** Interference aware routing in multi-radio wireless mesh networks. *IEEE WiMesh.* Sept. 2006. pp. 55 - 63.
106. **T. B. Thomas Staub.** Atom: Adaptive transport over multipaths in wireless mesh networks. *2nd ERCIM Workshop on eMobility.* May 2008.
107. **P. Hu, M. Portmann, R. Robinson, and J. Indulska.** Context-aware routing in wireless mesh networks. *ACM CASEMANS.* 2008. pp. 16 - 23.

108. **M. Oh.** An Adaptive Routing Algorithm for Wireless Mesh Networks. *ICACT*. February 2008.
109. **J. Määttä and T. Bräysy.** A novel approach to fair routing in wireless mesh networks. *EURASIP J. Wirel. Commun. Netw.* 2009.
110. **Ricardo Matos and Susana Sargento.** Data Communications over Context-Based WMNs: Delay Performance Evaluation. *3rd IFIP/IEEE International Workshop on Bandwidth on Demand and Federation Economics (BoD), Co-located with NOMS Symposium*. Osaka, Japan, April 2010.
111. NS-2 in the Wikipedia. [http://en.wikipedia.org/wiki/Ns\\_\(simulator\)](http://en.wikipedia.org/wiki/Ns_(simulator)). June 2010.
112. **Ramón Agüero Calvo, Jesús Pérez Campo.** Adding Multiple Interface Support in NS-2. *Howto*. University of Cantabria, January, 2007.
113. **Sven Wiethölter, Christian Hoene.** Design and Verification of an IEEE 802.11e EDCF Simulation Model in ns-2.26. *Technical Report TKN-03-019*. Telecommunication Networks Group, Technische Universität Berlin, November de 2003.
114. **Sven Wiethölter, Christian Hoene, Adam Wolisz.** Perceptual Quality of Internet Telephony over IEEE 802.11e Supporting Enhanced DCF and Contention Free Bursting. *Technical Report TKN-04-11*. Telecommunication Networks Group, Technische Universität Berlin, September 2004.
115. **Sven Wiethölter, M. Emmelmann, Christian Hoene, Adam Wolisz.** TKN EDCA Model for ns-2. *Technical Report TKN-06-003*. Telecommunication Networks Group, Technische Universität Berlin, June 2006.
116. **Olof Rensfelt, Lars-Ake Larzon.** A bandwidth study of a DHT in a heterogeneous. *Technical Report 2007-017*. IT Department, Uppsala University, May 2007.
117. **S. Rhea, D. Geels, T. Roscoe, and J. Kubiawicz.** Handling churn in a DHT. *Proc. of the 2004 USENIX Technical Conference*. Boston, MA, USA, June 2004.
118. **Marcel C. Castro‡, Eva Villanueva‡, Iraide Ruiz‡, Susana Sargento\*, and Andreas J. Kasser‡.** Performance Evaluation of Structured P2P over. *The Second International Conference on Sensor Technologies and Applications*. ‡University of Karlstad, Sweden,\*Instituto de Telecomunicações – Aveiro, Portugal, August 2008.
119. **B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman.** PlanetLab: An Overlay Testbed for Broad-Coverage Services. *ACM SIGCOMM Computer Communication Review*. July 2003.
120. Opave. <http://opave.com/nokia/>. June 2010.