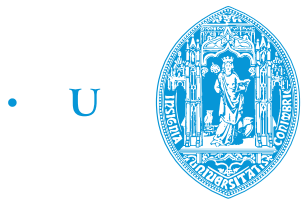


Universidade de Coimbra
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática

Proposta e estudo de soluções para otimização de rotas em ambientes de mobilidade de redes

Pedro Alexandre Vale Pinheiro

Coimbra
Janeiro 2013

Tese submetida à
Universidade de Coimbra
para a obtenção do grau de
Doutor em Engenharia Informática

Proposta e estudo de soluções para otimização de rotas em ambientes de mobilidade de redes

Pedro Alexandre Vale Pinheiro

Universidade de Coimbra
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática
Janeiro 2013

Tese realizada sob a orientação do
Professor Doutor Fernando Boavida
Professor Catedrático
do Departamento de Engenharia Informática
da Faculdade de Ciências e Tecnologia
da Universidade de Coimbra

Esta tese foi parcialmente financiada pelos projetos
IST-FP6-0384239: CONTENT – Network of Excellence on Content Networks and Services for Home Users,
e
FCT PTDC/EIA –EIA/116173/2009: CoFiMoM - Combate a Incêndios com Multihoming e Mobilidade.

À minha esposa e filho, Neusa e Gabriel

Aos meus pais, Manuel e Tina

Agradecimentos

Esta tese não seria uma realidade sem o apoio incondicional do Professor Doutor Fernando Boavida. Agradeço, por um lado, por ter conseguido manter constante o entusiasmo neste trabalho ao longo dos anos, mesmo nas alturas mais difíceis e, por outro, pela sua amizade, paciência, dedicação e companhia.

Agradeço também aos estudantes que contribuíram para este trabalho, com particular destaque ao Shivam Jain e ao André Carvalho.

Uma palavra de apreço é devida aos Professor Doutor Pedro Vieira Alberto, Eng. Luís Pinto e Eng. Pedro Almeida, do Laboratório de Computação Avançada da Universidade de Coimbra, por facultarem acesso ao cluster Milipeia, sem o qual não teria sido possível a execução das simulações presentes nesta tese.

Um agradecimento muito especial ao Eng. Mário Bernardes, tanto pela grande amizade como pela motivação e aconselhamento constantes.

Não posso deixar de manifestar o meu apreço por todo o pessoal do serviço de Gestão de Sistemas de Informação e Infraestruturas de Redes da Universidade de Coimbra, GSIIC, pela simpatia e acolhimento caloroso com que fui recebido desde o primeiro instante.

Agradeço, também, a todos os docentes do Departamento de Engenharia Informática que, de uma forma ou doutra, fizeram o que atualmente sou a nível profissional. Uma nota muito especial é dedicada a todos os docentes que me apoiaram mais de perto e que me influenciaram mais profundamente.

Um agradecimento muito especial e sentido vai para a minha esposa, Neusa, que me deu o seu apoio imediato, constante e incondicional para esta ousada aventura. Outro agradecimento, com a mesma intensidade, é dirigido ao meu filho, Gabriel, a quem tive que roubar tempo de brincadeira, pinturas e outras atividades bem mais agradáveis aos seus

olhos. O pai depois explica-te os motivos quando fores maior. Sem eles esta tese não existiria.

Uma palavra muito especial vai para a minha família, em especial ao meu pai e à minha mãe pelo seu amor, carinho e dedicação. Sei, em particular, que o meu pai iria adorar folhear este documento e tecer doutas considerações. Agradeço também ao meu irmão e família por estarem sempre presentes, e aos meus sogros e cunhado pelo carinho com que me acolheram.

Agradeço ainda aos meus amigos por estarem sempre presentes dando mais sentido à vida.

A todos os que não mencionei e que, de uma maneira ou de outra, me ajudaram na escrita deste trabalho, o meu muito obrigado.

Resumo

No mundo de hoje, no qual se acentua a tendência para que todo o tipo de comunicações recorra à arquitetura TCP/IP e crescem, em número e tipo, os dispositivos que utilizam ligações sem fios, a mobilidade em ambiente IP assume um papel de extrema importância. Por esse facto, tem sido grande a atenção da comunidade científica à proposta e desenvolvimento de soluções de mobilidade IP de nós individuais e de redes.

O NEMO Basic Support Protocol, IETF RFC 3963, foi desenvolvido com o objetivo de fornecer mobilidade de redes de forma imediata e transparente para a Internet atual. Contudo, a sua simplicidade está na génese das suas maiores limitações, que resultam em claros problemas de desempenho. Por outro lado, nenhuma das alternativas propostas com o intuito de resolver estas limitações conseguiu reunir consenso.

Nesta tese é apresentada uma mudança de paradigma, que consiste em envolver os dispositivos finais nos processos de mobilidade de redes. A proposta *Optimised Mobility for Enhanced Networking*, OMEN, faculta os mecanismos necessários para que os dispositivos finais tomem consciência da sua condição de mobilidade e possam recorrer aos mecanismos de otimização de rotas já previstos no MIPv6, de forma a não estarem sujeitos às limitações do RFC 3963. Com esta medida consegue-se resolver o problema da decisão da altura ideal para otimizar a rota de um determinado fluxo e, ao mesmo tempo, permitir que os elementos da infraestrutura de rede móvel fiquem dedicados às suas funções de encaminhamento de pacotes, resultando num incremento acentuado do desempenho da rede e num decréscimo do consumo de energia. As simulações realizadas mostram que a proposta OMEN apresenta valores de desempenho de comunicação e de perda de pacotes substancialmente melhores que as restantes soluções existentes, corroborando as vantagens da mudança de paradigma.

Para a realização dos diversos estudos de comparação das soluções foi necessário desenvolver um emulador que permitisse resolver as limitações de falta de implementação das soluções de mobilidade de redes e, ao mesmo tempo, permitir simulações de larga escala e de carga na rede. O emulador desenvolvido, denominado mobSim, foi executado num *cluster* de grandes dimensões, dado o tamanho e complexidade dos cenários de simulação.

Abstract

In the current world, in which there is a growing trend to use the TCP/IP protocol suite in all types of communication networks, and the number and type of devices using wireless connections is growing, IP mobility of both nodes and networks is of extreme importance. This is the main reason why the scientific community has paid and is paying special attention to the proposal and development of IP mobility solutions.

The NEMO Basic Support Protocol, IETF RFC 3963, was developed with the objective of readily allowing transparent network mobility in the current Internet. Nevertheless, the simplicity of this solution is at the basis of its limitations, which severely affect its performance. On the other hand, none of the proposed alternatives is gathering enough consensus of the community.

In this thesis, a paradigm shift is proposed, consisting of involving end nodes in the network mobility process. The proposal, named Optimised Mobility for Enhanced Networking, OMEN, establishes the necessary means for informing end nodes of their mobility condition, which can then use existing MIPv6 route optimisation mechanisms in order for them not to be subject to the limitations of RFC 3963. In this way, the problem of deciding which and when to optimise flows is left to the end nodes, which are in the best position to decide. At the same time, mobile routers are freed from all tasks concerning the mobility management of a potentially large number of flows, making them lighter and with lower power requirements.

The performed simulations show that the OMEN proposal leads to better performance then existing network mobility solutions, confirming the advantages of the paradigm shift.

The performed studies were carried out using a specially built network mobility emulator, in order to overcome the lack of support for this type of mobility and the scalability limitations of existing simulators. The developed emulator, named mobSim, ran in a large cluster, due to the size and complexity of the simulated scenarios.

Palavras Chave / Keywords: IPv6, Network Mobility, Route Optimisation, Simulation

Índice

Lista de Tabelas

Lista de Figuras

Acrónimos

AP	*Access Point*
AR	*Access Router*
ATS	*Air Trafic Services*
BA	*Binding Acknowledgment*
BEX	*HIP Base Exchange*
BR	*Binding Router*
BU	*Binding Update*
CE	*Correspondent Entity*
CN	*Correspondent Node*
CoA	*Care-of Address*
CoT	*Care-of Test*
CoTi	*Care-of Test init*
DAD	*Duplicate Address Detection*
DHT	*Distributed Hash Table*
DNS	*Domain Name Server*
EID	*Endpoint Identifiers*
ESP	*Encapsulation Security Payload*
ETR	*Egress Tunnel Router*
HA	*Home Agent*
Global HAHA	*Global Home Agent to Home Agent Protocol*
HIP	*Host Identity Protocol*
HMIPv6	*Hierarchical Mobile IPv6*
HoA	*Home Address*
HoT	*Home Test*
HoTi	*Home Test init*
IEEE	*The Institute of Electrical and Electronics Engineers*
IETF	*The Internet Engineering Task Force*
IP	*Internet Protocol Address*
ISP	*Internet Service Provider*
ITR	*Ingress Tunnel Router*
LFN	*Local Fixed Node*
LISP	*Locator/ID Separation Protocol*
LMN	*Local Mobile Node*
MAP	*Mobility Anchor Point*
MEXT	*Mobility Extensions for IPv6*
MIP	*Mobile IP*

MIPv6	*Mobile IPv6*
MIRON	*Mobile IPv6 RO for Network Mobility*
MMUSIC	*working group Multiparty Multimedia*
MN	*Mobile Node (MIPv6)*
MNN	*Mobile Network Node (NEMO)*
MNP	*Mobile Network Prefix*
MR	*Mobile Router*
MRHA	*Tunnel Mobile Router to Home Agent*
MS	*Multilink Subnet (RO ND-Proxy)*
MSR	*Multilink Subnet Router (RO ND-Proxy)*
MTU	*Maximum Transmission Unit*
MX	*Mail Exchange (DNS)*
NAT	*Network Address Translation*
ND	*Neighbor Discovery*
NEMO	*Network Mobility*
ORC	*Optimized Route Cache Managment Protocol*
OSI	*Open Systems Interconnection*
PAN	*Personal Area Network*
PANA	*Protocol for Carrying Authentication for Network Access*
PCH	*Path Control Header*
RA	*Router Advertisement*
RFC	*Request For Comment*
RLOC	*Routing Locators (LISP)*
RO	*Route Optimisation*
RR	*Resource Record (DNS)*
RR	*Return Routability (MIPv6)*
RS	*Router Solicitation*
RTT	*Round Trip Time*
SA	*Security Association (IPSec ESP)*
SCTP	*Stream Control Transmission Protocol*
SIP	*Session Initiation Protocol*
SLD	*Second Level Domain (DNS)*
SMTP	*Simple Mail Transport Protocol*
SRV	*Server (DNS)*
TCP	*Transmission Control Protocol*
UDP	*User Datagram Protocol*
URI	*Uniform Resource Identifier*
URL	*Uniform Resource Locator*
VMN	*Visiting Mobile Node*
WG	*IETF Work Group*
LC	*Legacy-compatible (LC) network mobility paradigm*

1. Introdução

A Internet está a evoluir para uma rede com requisitos de acesso ubíquo e ininterrupto. O acentuado crescimento do número e tipo de dispositivos que requerem acesso à Internet é uma clara indicação desta tendência.

De modo a possibilitar o acesso à Internet de dispositivos a partir de qualquer ponto em qualquer altura, é fundamental lidar com mudanças de localização sucessivas. Contudo, sempre que o ponto de acesso à Internet é modificado isto traduz-se na quebra de ligação das comunicações em curso. Para que tal não aconteça, [Perkins10] aponta como requisitos duas possibilidades:

- o dispositivo tem que modificar o seu endereço IP sempre que muda o seu ponto de ligação à Internet;
- ou devem ser propagadas as rotas específicas para este dispositivo ao longo dos elementos da rede na Internet

Contudo, [Perkins10] alerta também para o facto de ambas as soluções terem problemas. A primeira solução torna impossível a manutenção das ligações da camada de transporte e superiores sempre que o endereço IP é alterado. Por outro lado, a segunda opção tem problemas óbvios de escalabilidade, especialmente relevantes tendo em consideração o forte aumento do número de equipamentos móveis. Do mesmo modo, [Stallings05] afirma que, a par da realidade de que o protocolo IP não foi concebido para ter mobilidade, também é possível constatar que para haver conectividade é necessário que os dispositivos usem um endereço IP topologicamente correto, o que não acontece sempre que existe uma transição entre diferentes fornecedores de serviço (Internet Service Providers, ISP).

Esta problemática assume uma nova e mais abrangente perspetiva quando analisada sob o prisma da mobilidade de uma rede IP. Embora um dispositivo dentro dessa rede móvel mantenha o seu endereço IP, o equipamento responsável por ligar a rede à Internet (normalmente um *router*) vai obtendo diversos endereços IP enquanto atravessa os diversos ISPs [Gai07].

De modo a resolver este problema, o IETF NEMO Working Group[1] publicou o RFC 3963 com o título NEMO Basic Support Protocol [Devarapalli05], que permite implementar de imediato a mobilidade de redes na Internet atual. Contudo, esta proposta apresenta diversas limitações que inviabilizam a sua utilização em larga escala.

Foram apresentadas diversas propostas para resolver as limitações do RFC 3963. Não obstante já terem sido apresentadas há algum tempo, existe falta de consenso sobre qual a melhor solução a adotar. Deste modo pode-se concluir, com base no estado da arte, que a tecnologia não é suficientemente madura de modo a ser implementada de forma estável e com fins comerciais.

Esta lacuna e falta de consenso abrem espaço à investigação e ao desenvolvimento de uma nova solução que vá de encontro aos requisitos da mobilidade de redes. De seguida são apresentados os objetivos da presente tese tendo em consideração o exposto.

1.1. Objetivos

A implementação da mobilidade IP pode ser efetuada em diversas camadas da pilha protocolar TCP/IP. Contudo, para o caso da mobilidade de redes a utilização da camada IP é a que reúne maior consenso da comunidade científica.

A proposta do IETF NEMO Working Group para a mobilidade de redes, NEMO Basic Support Protocol, consiste numa extensão do protocolo Mobile IPv6 [Perkins11], embora sem suporte do procedimento de otimização de rotas do MIPv6. Este procedimento consiste em utilizar o novo endereço IP, obtido quando em mobilidade, para criar uma comunicação mais direta entre os dispositivos finais. A não utilização deste caminho otimizado leva a que os pacotes entre os extremos tenham que atravessar um agente localizado na rede original da rede móvel, o que resulta em diversos problemas, conforme referido por diversos autores, dos quais se destacam os [Ng07], [Ng07a] e [Bernardos05a].

Neste contexto, o objetivo desta tese consiste em analisar a temática da mobilidade de redes, nas suas diversas perspetivas. Após uma análise do estado da arte, pretende-se apresentar uma proposta que permita resolver as limitações inerentes à proposta NEMO Basic Support Protocol, assim como preencher as lacunas deixadas em aberto pelas soluções existentes. Por fim, é também objetivo da tese avaliar e comparar a proposta apresentada com as soluções de mobilidade de rede já existentes.

[1] http://tools.ietf.org/wg/nemo/

1.2. Contribuições

Tendo em consideração a importância crucial de existir uma solução de otimização de rotas para ambientes de mobilidade de redes, salienta-se que a contribuição principal desta tese consiste na concepção de uma nova solução de mobilidade de redes genérica conhecida como *Optimised Mobility for Enhanced Networking*, OMEN. Esta nova proposta traduz-se numa mudança de paradigma de mobilidade de redes.

O paradigma atual consiste em tornar tão transparente quando possível aos dispositivos finais todos os procedimentos de mobilidade IP. No caso da proposta NEMO Basic Support Protocol, esta funcionalidade é conseguida à custa de túneis bidirecionais entre o agente na rede móvel e o *router* móvel.

As soluções que se propõem a resolver as limitações do RFC 3963, continuam a manter o paradigma de transparência da mobilidade. Assim, são propostas diversas soluções em que os mecanismos de otimização de rotas são realizados pelos elementos da infraestrutura. Estas propostas levam a um impacto indesejável na infraestrutura de redes e, consequentemente, na comunicação entre os dispositivos finais.

A proposta OMEN vai no sentido de mudar o paradigma atual, focando-se na necessidade de dar a conhecer a condição de mobilidade aos dispositivos móveis finais. Deste modo, estes dispositivos poderão beneficiar, se assim o desejarem, deste conhecimento, para criarem rotas otimizadas para as ligações que delas necessitem.

Com esta mudança de paradigma espera-se uma melhoria significativa do desempenho das comunicações, já que o esforço da execução dos procedimentos de mobilidade é distribuído pelos dispositivos móveis finais.

Por outro lado, a necessidade de avaliar o desempenho desta proposta em comparação com outras soluções existentes e em ambientes de grande escala tornou claras as limitações dos simuladores existentes. Assim, uma outra contribuição relevante consistiu no desenvolvimento de um emulador que suporta, de forma nativa, a mobilidade de nós e de redes, permitindo também a avaliação do desempenho das soluções em ambientes de larga escala ou em situações de carga moderada a elevada.

Por último, embora não menos importante, este trabalho permitiu obter resultados que sustentam a corrente emergente de que a mobilidade de redes deverá ser conhecida por parte dos dispositivos finais, corrente essa que, no contexto dos trabalhos do IETF, se designa por mobilidade baseada no cliente (client-based mobility).

1.3. Organização do documento

A presente dissertação está organizada em seis capítulos e três anexos. Para além do capítulo corrente, que apresenta a motivação, os objetivos, as contribuições e a organização do texto, este documento começa por avaliar o estado da arte no capítulo 2. A análise da mobilidade ao nível das diversas camadas da arquitetura TCP/IP, assim como as diversas soluções enquadradas nessas camadas, iniciam o capítulo. Posteriormente, é canalizada a atenção do capítulo para as questões da otimização de rotas e as soluções que a suportam. Também é feita uma análise dos requisitos de mobilidade de redes de modo a enquadrar a margem de manobra para a criação de novas soluções. O capítulo termina com a caracterização dos paradigmas de mobilidade de redes.

O capítulo 3 está centrado na solução proposta no âmbito desta tese. É feita uma descrição do OMEN e são avaliados os requisitos em função dos tipos de dispositivos que podem existir numa rede móvel. Uma parte importante do capítulo é dedicada à especificação semiformal da proposta. Na parte final são discutidos aspetos de validação e de segurança.

O capítulo 4 começa com uma análise das ferramentas de simulação existentes, como forma de justificar o desenvolvimento de uma nova ferramenta. O núcleo do capítulo é dedicado à apresentação das funcionalidades, arquitetura, características, aspetos de implementação e de utilização do emulador mobSim.

O capítulo 5 faz uma análise exaustiva do comportamento do OMEN quando comparado com restantes soluções de mobilidade de redes. As avaliações contemplam aspetos funcionais, a comparação de paradigmas em ambientes de larga escala, a realização de testes de carga e, finalmente, a inclusão de ligações reais sem fios nos cenários de emulação.

O capítulo 6 apresenta as conclusões e identifica linhas para trabalho futuro.

Nos anexos encontram-se alguns scripts relacionados com o emulador implementado.

2. Estado da arte

O sucesso das redes celulares tem impulsionado a já de si crescente necessidade de um acesso à Internet ubíquo e ininterrupto. Neste contexto, a mobilidade de dispositivos apresenta-se como um tema atual e urgente. Esta área de estudo tem sido alvo de considerável atenção nas últimas décadas, tendo em vista encontrar soluções adequadas para o facto de os protocolos da arquitetura TCP/IP, desenvolvida em grande parte nas décadas de 1970 e 1980, terem sido concebidos para equipamentos e redes fixos. A génese e paradigmas subjacentes à atual Internet são reconhecidos por diversos autores como os principais obstáculos à mobilidade de sistemas terminais e redes. Desses autores, referem-se, a título de exemplo, os trabalhos [Eddy04], [Ratola04] e [Bernardos05a], que procuram focar-se nas implicações da mobilidade.

Do ponto de vista técnico, os endereços IP que constam nos campos *source IP* e *destination IP* são imutáveis durante uma ligação TCP, já que esta é identificada por esses dois endereços em conjunto com os campos *source port* e *destination port*. É esta premissa que [Henderson03] considera ser o calcanhar de Aquiles que dificulta a implementação da mobilidade.

Com efeito, sempre que um dispositivo muda de rede, o seu endereço IP original passa a estar topologicamente incorreto e, desta forma, o dispositivo deixa de estar contatável. A mudança de localização traduz-se, então, na perda de conectividade. Além disso, tal como já foi referido, a comunicação entre dois dispositivos é conseguida com recurso aos respetivos endereços IPs, pelo que qualquer modificação destes valores leva a uma quebra de ligação.

De facto, de forma simples, é esta a base de todo o problema: numa Internet desenvolvida para dispositivos e redes estáticos, os endereços IP foram concebidos de forma a desempenharem o duplo papel de localizadores e identificadores.

Esta dualidade de papéis dificulta a implementação da mobilidade de dispositivos, já que ao modificar-se a localização de um dispositivo, a sua identificação é automaticamente modificada, o que, naturalmente, é indesejável. O que se pretende, então, é conseguir uma maneira de modificar a localização mantendo a identificação do dispositivo.

No seguimento da análise desta problemática, [Eddy04] propõe três princípios básicos a serem seguidos para a implementação de possíveis soluções para a mobilidade de dispositivos:

- ***Transições não penosas***: sempre que houver mudança de ponto de ligação à Internet, esta não deverá traduzir-se em perda substancial de dados críticos para as aplicações, não deve ser tão pesada que incremente o tempo de inacessibilidade, e deve ser transparente para as aplicações de modo a não induzir perda de sessão para aplicações de longa duração;
- ***Gestão de localização***: o dispositivo móvel deve estar sempre acessível através algum identificador estático, independentemente da sua localização física;
- ***Infraestrutura leve***: uma solução de mobilidade deve ser aplicada tão próxima do extremo quanto possível; deve-se evitar a aplicação de atualizações em equipamentos de encaminhamento intermédios, sob pena de inviabilizar a sua implementação.

Tendo estas condicionantes em consideração, Wesley Eddy analisa a aplicação da mobilidade nas diversas camadas da arquitetura TCP/IP, de que resulta a Tabela 2.1.

	Transições	**Gestão da localização**	**Infraestrutura**
Camada de Rede	A camada de transporte deve lidar com perdas e mudanças de caminho	Incluída	Implementação de *Home Agents* e *routers* com capacidade para *fast / smooth handover*
Camada de Transporte	Incluídas	Requer gestão de localização externa	Pouca ou nenhuma
Camada de Sessão	Incluídas	Pode ser incluída	Pouca ou nenhuma

Tabela 2.1 – Sumário de diferenças entre soluções de mobilidade em função das camadas TCP/IP [Eddy04]

Quando a mobilidade é implementada na camada de rede, é necessário que a infraestrutura seja dotada de equipamentos adicionais que façam encaminhamento de pacotes e gestão da localização, enquanto que as camadas superiores deverão cuidar de eventuais perdas de pacotes. No caso da mobilidade ser aplicada ao nível da camada de transporte, então a gestão de localização deve ser alicerçada por equipamentos externos que auxiliem e validem a identificação. Para o caso da camada de sessão, considera o autor que é necessário alterar as aplicações de modo a suportarem esta funcionalidade.

Embora todas as opções apresentadas sejam válidas, a presente tese analisa a mobilidade na perspetiva da camada de rede, já que esta é, também, a opção mais consensual na

comunidade científica. Neste capítulo começa-se, então, pela apresentação das soluções que recorrem à camada de rede, na secção 2.1. Esta análise aborda as propostas tanto do ponto de vista de um dispositivo (nó), como de uma rede isolada ou, ainda, como de redes imbricadas. A secção 2.2 aborda a problemática da localização e identificação, através de um conjunto de soluções que recorrem a funcionalidade que, estando acima da camada de rede, não estão na camada de transporte. Estas soluções são, frequentemente, referidas como soluções da camada 3 e ½ , sendo, no fundo, uma modificação das soluções baseadas na camada de transporte. A secção 2.3 analisa as soluções baseadas na camada de transporte e camadas superiores. A secção 2.4 aprofunda o estudo de soluções de mobilidade que usem procedimentos de otimização de rotas, ao nível da camada de rede. A secção 2.5 apresenta uma análise de requisitos comuns e requisitos específicos (i.e., dependentes da área de aplicação) para a mobilidade de redes. A secção 2.6 conclui o capítulo, apresentando uma visão geral e uma classificação das soluções de mobilidade de rede.

Como nota final, é importante referir que foi tomada a decisão de focar o presente estudo no protocolo IP versão 6. Embora o estudo da mobilidade de dispositivos se tenha iniciado com o protocolo IP versão 4, esta funcionalidade foi implementada de raiz na nova geração do protocolo IP, a versão 6 (IPv6). Dado que adicionar mobilidade acrescenta um grau de complexidade que apenas é justificável se houver a previsão de um elevado número de equipamentos que possa beneficiar dela, o alargamento dos horizontes só é possível de forma transparente e simples com recurso ao elevado número de endereços que o IPv6 faculta. Por outro lado, para além das questões de eficácia e eficiência, as questões de segurança na mobilidade assumem um papel preponderante, já que vários dos mecanismos necessários à mobilidade podem abrir um caminho para novos ataques. Uma vez mais, o IPv6 foi concebido com as ferramentas adequadas deste ponto de vista. Foi por estes motivos que, no presente trabalho, se optou por estudar a mobilidade de redes apenas para ambiente IP versão 6.

2.1. Mobilidade na camada de rede

A mobilidade ao nível da camada de rede pode ser conseguida com recurso a dois tipos distintos de soluções:

a) modificação das rotas para cada nó móvel, de modo a que o endereço IP permaneça topologicamente correto;
b) reencaminhamento do tráfego destinado a um nó móvel através de agentes responsáveis por fazerem chegar os pacotes ao destino.

O primeiro tipo de solução levanta problemas de escalabilidade e exequibilidade, já que não é prático alterar configurações de encaminhamento em tempo real para conseguir fornecer rotas para os dispositivos móveis. Tal implicaria não só um grande *overhead* em termos de protocolos de encaminhamento, muitos dos quais teriam que ser alterados, mas também uma sobrecarga dos *routers* em termos de processamento.

O segundo tipo de solução é o que, na prática, tem vindo a ser investigado. A sua implementação mais conhecida é o protocolo *Mobile IPv6*, MIPv6 [Perkins11] [Stallings05], que será descrito na secção 2.1.1.

A perspetiva do problema adquire um nível de complexidade adicional se passar do movimento de um simples nó móvel para toda uma rede móvel. Este é, de facto, o objeto da presente tese. Naturalmente, neste cenário muitos conceitos e ideias podem ser transpostos do MIPv6. A proposta mais conhecida para mobilidade de redes é designada *Network Mobility* – NEMO, sendo documentada no RFC 3963 [Devarapalli05]. A secção 2.1.2 aborda este protocolo.

O complexidade aumenta ainda mais quando se verifica que uma rede móvel pode aceder à Internet através de outra rede móvel. Esta situação é conhecida como imbricação de redes. Justifica-se uma secção dedicada a este tema.

O *Internet Engineering Task Force* MEXT *Working Group* (IETF-MEXT WG)[2] é o grupo de trabalho que desenvolveu diversos documentos que serviram de base às soluções apresentadas no presente capítulo.

2.1.1. Mobilidade de nós – MIPv6

O protocolo *Mobility Support in IP Version 6*, mais conhecido como MIPv6, especificado no *Request For Comments* (RFC) 6275 [Perkins11], define a terminologia e os componentes que estão envolvidos na mobilidade. A Figura 2.1 ilustra os elementos existentes no RFC, dos quais se destacam os seguintes [Perkins11] [Manner04]:

- ***Mobile node, nó móvel* – MN**, um nó autónomo, equipado com pelo menos uma interface de acesso a redes sem fios, e com suporte de MIPv6[3];

[2] http://tools.ietf.org/wg/mext/

[3] Em boa verdade, todo o equipamento que tenha suporte de IPv6 é um potencial utilizador de MIPv6, pelo que dizer que um *mobile node* é um nó com capacidade de MIPv6 é redundante. Apesar disso, optou-se por colocar esta referência para reforçar esta funcionalidade.

- ***Home agent* – HA**, um *router* existente na rede original do nó móvel que encaminha os pacotes para os nós que se encontrem em redes externas. Este router mantém uma tabela que relaciona o IP original com o IP na rede externa;
- ***Correspondent node, nó correspondente* – CN**, um nó que está em comunicação com o nó móvel. Este nó pode ser móvel ou estacionário.

As iniciais AR da Figura 2.1 designam o *access router* e referenciam o *router* da rede visitada, que fornece ao nó móvel o meio de acesso à Internet.
A par destes elementos, é necessário conhecer outros termos que foram criados e que são essenciais para uma boa compreensão deste protocolo,

- ***Home address* – HoA**, um endereço pertencente à rede origem, e que está permanentemente associado ao nó móvel;
- ***Care-of address* – CoA**, um endereço associado à rede visitada pelo nó móvel. Dado que o CoA é topologicamente correto, o seu prefixo é o da rede visitada (*foreign subnet prefix*);
- ***Binding***, a associação do *home address* do nó móvel com o *care-of address* adquirido na rede visitada;
- ***Registration, registo***, operação durante a qual um nó móvel envia pacotes de *binding* para o seu *home agent* ou para o seu nó correspondente;
- ***Return Routability procedure* – RR**, operação que permite autorizar registos, com recurso a troca de testemunhos encriptados.

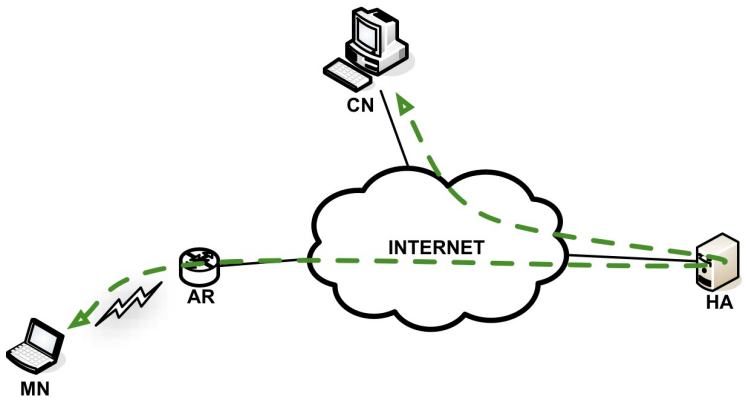


Figura 2.1 – Entidades envolvidas no MIPv6

Um nó móvel possui, por norma, dois endereços IP, o *home address* (HoA) e o *care-of address* (CoA), que lhe permitem definir a sua identidade e a sua localização,

respectivamente. Exceptua-se o caso em que o nó móvel se encontra na rede origem, pois nessa situação não possui *care-of address*, tendo o *home address* as funções de localizador e identificador.

O endereço IP *home address* é atribuído de forma automática ou manual ao nó móvel aquando da sua inicialização, com base no prefixo da rede origem, permanecendo inalterado. Por outro lado, o *care-of address* é atribuído sempre que o nó móvel se associa a uma nova rede estrangeira, quer através de *IPv6 stateless address autoconfiguration* [Thomson07], quer através de *dynamic host configuration protocol for IPv6*, DHCPv6 [Droms03]. O prefixo utilizado é o da rede estrangeira. Este endereço IP vai ser utilizado para as operações de *binding*, para reencaminhar os pacotes destinados ao *home address* enviando-os através de um túnel bidirecional [Conta98] [Perkins96] estabelecido entre o CoA e o *home agent*, e para a comunicação otimizada entre o nó móvel e o seu nó correspondente.

Assim que o nó móvel se associa a uma rede estrangeira e adquire o seu *care-of address*, o protocolo define que o nó móvel deve registar o seu novo endereço, ou o que for definido como o *care-of address* principal, com um *router* na sua rede origem. A esta operação designa-se *binding*. O nó móvel começa por enviar um pacote de *binding update* (BU) para o *router* requerendo que este se comporte como um *home agent* e que passe a encaminhar os pacotes destinados ao *home address* do nó móvel através do túnel bidirecional para o novo endereço IP. O *router* deverá responder com um *binding acknowledgement* (BA) a confirmar o registo. A partir deste momento, todos os pacotes destinados ao endereço *home address* serão interceptados pelo *home agent* e encapsulados no túnel bidirecional com destino ao *care-of address*. A resposta a ser enviada pelo nó móvel deverá seguir o caminho inverso, utilizando o mesmo túnel bidirecional. As questões de segurança relacionadas com estas operações foram amplamente estudadas em [Ferguson00] [Aura02] [Nordmark01] [Roe02] [Savola02] [Nikander05].

Assim que houver comunicação, em qualquer dos sentidos, entre o nó móvel e o seu nó correspondente, pode-se dar o início do procedimento de otimização de rotas, evitando que o tráfego seja forçado a circular pela rede origem, a que haja sobrecarga do seu *home agent* e a que fique sujeito à diminuição do *Maximum Transmission Unit* (MTU). Com a comunicação através da rota otimizada, o tráfego entre os dois nós passa a processar-se pelo caminho mais curto, conforme se pode ver na Figura 2.2. O procedimento de *Return Routability* (RR), descrito no que se segue, vai assegurar a segurança da execução deste processo.

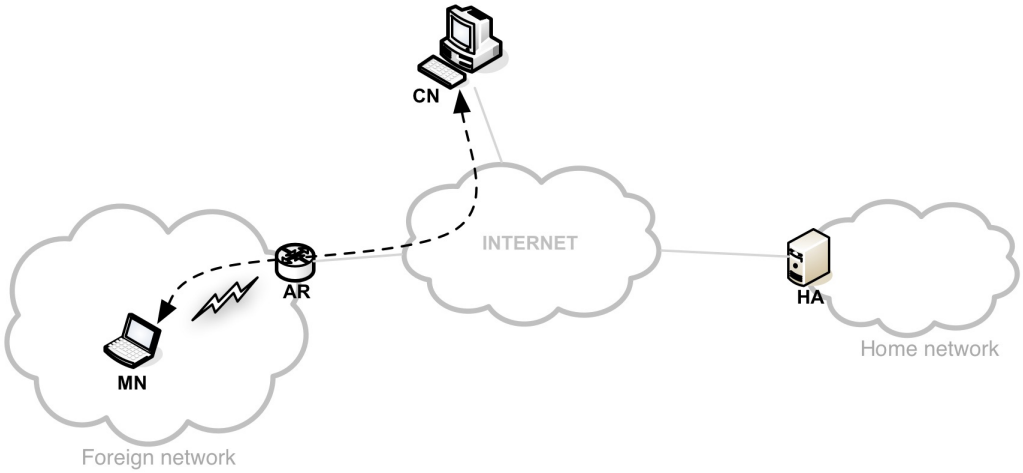


Figura 2.2 – Comunicação otimizada entre MN e CN

A Figura 2.3 representa os passos do processo de *Return Routability*. Na figura pode ver-se que a comunicação se processa entre o nó móvel (MN) e o seu nó correspondente (CN). O *home agent* (HA) é o *router* responsável por fazer os pacotes chegarem ao nó móvel MN. Os endereços IP do nó móvel são o *home address* MN e o *care-of address* CoA, este último adquirido na rede estrangeira. O endereço IP do *home agent* é o HA, enquanto o endereço IP do nó correspondente é CN.

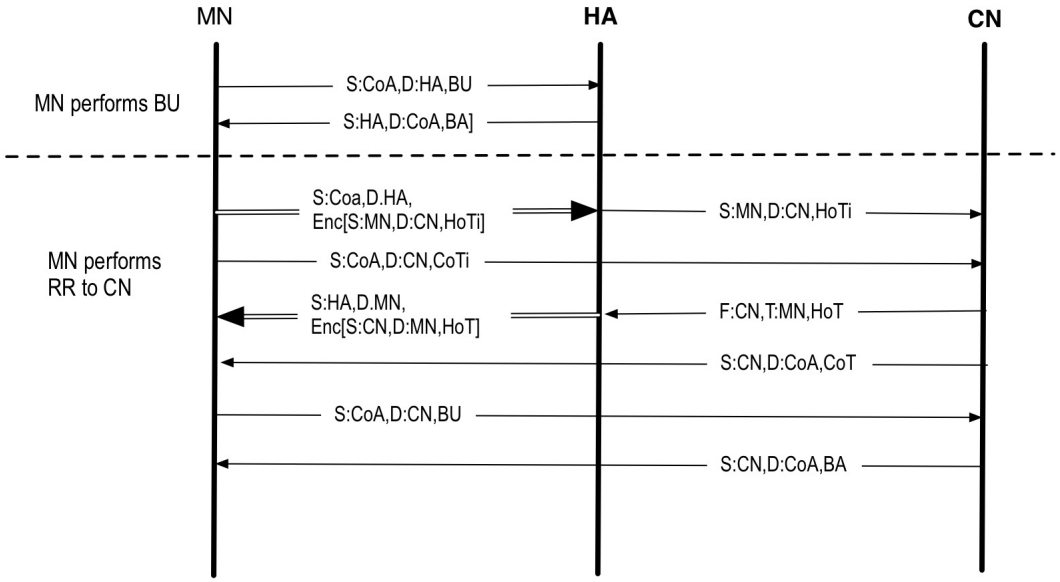


Figura 2.3 – Passos do procedimento de *Return Routability*

As setas definem o sentido dos pacotes. No caso da primeira ligação, cujo descritivo é `S:CoA,D:HA,BU`, o `S` define a origem (*source IP*), enquanto o `D` define o destino (*destination IP*). Neste exemplo, o descritivo da seta in dica que o pacote, com origem no endereço IP `CoA` do nó móvel, é destinado ao endereço IP do *home agent* `HA` e contém como *data* o *binding update* (BU). Neste caso, a ligação é efetuada com recurso ao IPSec [Arkko04] [Devarapalli07] [Aura02] [Roe02]. As setas representadas por linha dupla correspondem a comunicação encapsulada, isto é, comunicação através do túnel bidirecional.

O primeiro bloco, constituído pelas duas setas, que se encontra identificado na figura como `MN performs BU`, mostra que o nó móvel executa o processo de *binding* com o seu *home agent*, tendo como resposta um *binding acknowledgement* (BA). No fim deste processo, o *home agent* passou a estar na posse do *care-of address* do nó móvel, e fez o mapeamento com respetivo *home address*. O processo de *return routability* só é possível se existir o *binding* entre o nó móvel e o seu *home agent*, dado que alguns pacotes terão que circular pela rede origem, de forma a garantir as questões de segurança [Kent05a] [Kent05b] [Kent05c].

De seguida, o nó móvel envia dois pacotes para o seu nó correspondente, designados *home test init* (`HoTi`) e o *care-of test init* (`CoTi`). O `HoTi` é enviado através do túnel bidirecional e sai com o *home address* do nó móvel, `MN`. O `CoTi` é enviado diretamente para o nó correspondente utilizando o IP *care-of address*, `CoA`. Estes pacotes vão conter um *token* que servirá para garantir que é o nó móvel que está a tentar estabelecer a otimização de rota com o nó correspondente.

Quando o `CN` recebe os dois pacotes deve enviar, como resposta, um *token* baseado no que recebeu dentro dos dois pacotes *home test* (`HoT`) e *care-of test* (`CoT`). Estes pacotes devem ser enviados, respetivamente, para o *home address* e *care-of address* do nó móvel.

Quando estiver na posse do novo *token*, o nó móvel está em condições de enviar o *binding update* contendo uma cifra que apenas os dois nós conhecem. Como resposta, o nó correspondente devolve um *binding acknowledgement*. A partir deste instante, a comunicação entre os dois nós é feita de forma otimizada. Isto significa que cada pacote é enviado com o endereço IP de localização do nó móvel, *care-of address* `CoA`, contendo um cabeçalho adicional com o *home address* do nó móvel, `MN`.

Se o pacote transitar do nó móvel para o nó correspondente, o campo a ser adicionado é o *home address option* [Perkins11], que deve transportar no *destination option extension header* o *home address* do nó móvel. Desta forma, o nó correspondente consegue relacionar o *care-of address* e o *home address*.

No sentido inverso, o nó correspondente deverá acrescentar o campo *type 2 routing header* (T2RH) [Deering98], preenchendo o *destination option extension header* com o próximo salto (*next hop*), ou seja, o *home address* do nó móvel. Assim, quando o nó móvel receber o

pacote destinado ao endereço *care-of address* CoA, deve consultar o campo T2RH para verificar se o *home address* é o seu.

Apesar do protocolo MIPv6 resolver o problema da mobilidade de forma eficaz, nem sempre consegue ser eficiente. Os principais problemas dizem respeito ao acréscimo de sinalização (*signaling overhead*) e à latência na ligação após quebra de conectividade (*latency handhover*). Várias soluções foram apresentadas para resolver estas questões, estando o seu estudo esteja fora do âmbito deste trabalho.

2.1.2. Mobilidade de Redes – NEMO

É acrescentado um factor de complexidade quando se transita da mobilidade de um nó para a problemática da mobilidade de uma rede contendo diversos nós. O IETF NEMO *Working Group*[4] elaborou o RFC 3963 [Devarapalli05], *Network Mobility (NEMO) Basic Support Protocol*, simplesmente conhecido como NEMO[5]. A abordagem adotada visa obter uma solução de imediata implementação e utilização, deixando uma abertura para uma solução mais abrangente, que será denominada *Network mobility (NEMO) Extended Support Protocol*.

Entretanto, o NEMO WG deu lugar ao IETF MEXT *Working Group*[6], um agregador dos diversos problemas relacionados com mobilidade IP e de redes.

Dado o surgimento de novos componentes, torna-se necessário atualizar o léxico relacionado com a mobilidade de redes, embora se mantenha toda a terminologia utilizada no MIPv6.

A Figura 2.4 ilustra uma rede móvel e os principais intervenientes, dos quais se salientam os seguintes [Devarapalli05] [Kent98] [Manner04] [Ernst07]:

- ***Mobile network,* rede móvel – NEMO**, uma rede inteira que se move de forma una, mudando de forma dinâmica o seu ponto de ligação à Internet e, consequentemente, o seu ponto de alcance na topologia; um ou mais *routers* móveis permitem a sua ligação ao exterior;
- ***Mobile router, router* móvel – MR**, um *router* com capacidade de mudar o seu ponto de ligação à Internet, movendo-se de rede em rede; este *router* é capaz de encaminhar pacotes entre duas ou mais interfaces de rede;

[4] http://tools.ietf.org/wg/nemo/

[5] NEMO pode referir-se ao protocolo, assim como pode referir-se a uma rede móvel, *NEtwork MObility*, dependendo do contexto

[6] http://tools.ietf.org/wg/mext/

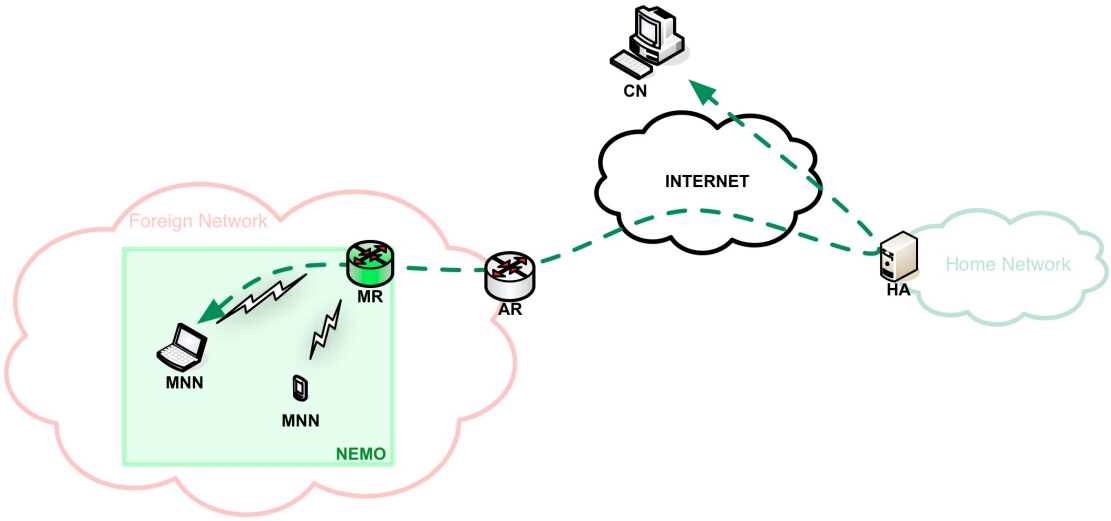


Figura 2.4 – Exemplo de uma rede móvel

- ***Mobile network prefix* , prefixo da rede móvel – MNP**, conjunto inicial de *bits* de um endereço IP que identifica a rede móvel inteira, de forma topologicamente correta na Internet; todos os nós dentro da rede móvel possuem necessariamente um endereço que contém este prefixo;
- ***Mobile network node*, nó da rede móvel – MNN**, um nó, máquina ou *router*, localizado dentro da rede móvel, tanto de forma permanente como temporária;
- ***Correspondent node*, nó correspondente – CN**, um nó que está a comunicar com um ou mais nós da rede móvel; um nó correspondente pode estar localizado numa rede fixa ou numa rede móvel, e pode suportar mobilidade IP.

É interessante notar que foi criada uma diferenciação entre o nó móvel (MIPv6) e o nó da rede móvel (NEMO). Assim, torna-se possível afirmar que um nó móvel (MIPv6), ou seja, fora de qualquer rede móvel, está a comunicar com um nó da rede móvel (NEMO). O nó da rede móvel ainda pode ser definido, com maior precisão, da seguinte forma:

- ***Local fixed node*, Nó local fixo – LFN**, um nó fixo, tanto dispositivo como *router*, que pertence à rede móvel e não tem capacidade de mudar o ponto de ligação à rede;
- ***Local mobile node*, nó local móvel – LMN**, tanto um nó móvel como um *router* móvel, associado à rede móvel e que possui capacidade de mudar o seu ponto de ligação à rede, enquanto mantém as ligações em curso;

- ***Visiting mobile node,* nó móvel visitante – VMN**, tanto um nó móvel como um *router* móvel que não pertence à rede móvel e que tem capacidade de mudar o seu ponto de ligação à rede, enquanto mantém as ligações em curso.

Os termos *home agent* – HA, nó correspondente – CN e *access router* – AR mantêm a designação já existente no MIPv6.
A comunicação entre o nó da rede móvel e o nó correspondente tem que atravessar a Internet através de um túnel conhecido como túnel MRHA, que é um túnel bidirecional entre o *router* móvel e o seu *home agent*.

Quando um pacote do nó da rede móvel é interceptado pelo *router* móvel, o MR encapsula o pacote dentro do túnel MRHA com destino ao *home agent*. Quando o HA recebe o pacote, efetua a operação inversa e envia o pacote original para a rede, sem qualquer encapsulação. No sentido contrário o processo processa-se de forma similar.
Segundo [Ernst07a], os objectivos principais que motivaram a criação do NEMO Basic Support Protocol foram a transparência na migração, mínimo impacto introduzido pela funcionalidade de mobilidade, transparência no suporte de mobilidade, transparência operacional, suporte de configurações diversificadas (redes de qualquer dimensão, mobilidade de nós, redes com múltiplos ponto de ligação, redes imbricadas, ...), mobilidade local e global, escalabilidade, retro-compatibilidade, segurança, privacidade na localização, e mínimo impacto no encaminhamento.
O RFC 3963 cumpre os requisitos impostos com grande simplicidade, facultando o acesso imediato à mobilidade de redes, sem necessidade de modificações ao nível de outros equipamentos que não sejam o *router* móvel e o *home agent*.
Do mesmo modo que no MIPv6, o túnel bidirecional entre o *router* móvel e o *home agent* garante as questões de segurança, assim como permite ultrapassar possíveis *firewall* que pudessem constituir um entrave à mobilidade de redes.

Com a terminologia atualizada e esclarecidos os principais objetivos que motivaram a criação do NEMO Basic Support Protocol, segue-se a explicação do protocolo. Conforme se pode depreender da terminologia, uma rede móvel é constituída por um *router* móvel (MR) e pelo seu prefixo da rede móvel (MNP), que define o endereçamento utilizado pelos nós da rede móvel (MNN). A Figura 2.5 apresenta os principais passos do protocolo NEMO.
Para além do *router* móvel e dos nós da rede móvel, temos como intervenientes neste processo o *access router* (AR), o *home* agent (HA) e o nó correspondente (CN). Os pacotes que transitam do nó da rede móvel para o nó correspondente têm que, forçosamente,

circular por todos estes equipamentos intermédios, para além dos equipamentos de *routing* existentes na Internet,

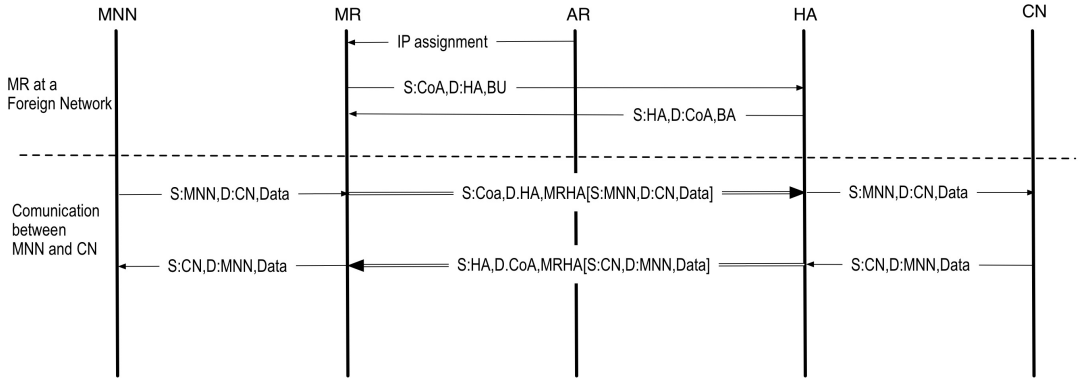


Figura 2.5 – Passos do protocolo NEMO

As setas indicam o sentido dos pacotes. A título de exemplo, a primeira seta indica que o *access router* enviou um, ou mais, pacotes para atribuir um endereço IP ao *router* móvel. O endereço obtido vai corresponder ao *care-of address*. A este processo, que se resolveu designar por *IP assignment*, pode corresponder o processo de *IPv6 stateless address autoconfiguration* [Thomson07] ou o processo de *dynamic host configuration protocol for IPv6*, DHCPv6 [Droms03].

Considerou-se, para ajudar à compreensão da figura, que cada equipamento possui um endereço IP igual ao seu nome. Assim, o endereço IP do nó da rede móvel é `MNN`. Os endereços *egress* do *router* móvel são `MR` para o *home address* e `CoA` para o *care-of address*. O *home agent* vai possuir o endereço `HA`, enquanto o endereço IP do nó correspondente vai ser o `CN`.

Na figura, cada seta possui um descritivo que vai conter os campos mais relevantes do cabeçalho IP, para o presente texto. A título de exemplo, considere-se o caso da segunda seta, que possui o descritivo `S:CoA,D:HA,BU`, e que consiste na operação de *binding update* do *router* móvel para o seu *home agent*. Pode-se verificar que a primeira parte – `S:CoA` – define como origem (*source*, `S`) o endereço IP `CoA`. O destino (*destination*, `D`) – representado por `D:HA` – indica o endereço IP do *home agent*, `HA`. O campo *data* vai conter a informação relativa ao *binding update* (`BU`). Tal como no caso da secção anterior, as setas de linha dupla representam comunicação encapsulada.

No exemplo da figura, a comunicação `S:CoA,D:HA,MRHA[S:MNN,D:CN,Data]` indica que o pacote tem origem no *router* móvel, através do seu novo endereço IP `CoA`, é destinado ao *home agent*, e que o túnel bidirecional MRHA encapsulou o pacote `S:MNN,D:CN,Data`.

O bloco MR at a Foreign Network representa, então, o processo de aquisição de um novo endereço IP, CoA, na rede visitada, seguido do *binding update* (BU) e *binding acknowledgement* (BA) entre o *router* móvel MR e o seu *home agent*, HA. Quando o *home agent* envia o pacote BA, anuncia para a sua rede que todos os pacotes destinados ao prefixo da rede móvel, MNP, deverão ser encaminhados para si. Este processo vai permitir que a rede móvel volte a estar contactável, através do túnel MRHA. Normalmente, o papel de *home agent* é desempenhado pelo *access router* do *router* móvel, quando este está na rede origem, de modo a evitar um processo mais complexo de desvio do tráfego destinado à rede móvel.
O segundo bloco, Communication between MNN and CN, ilustra como se processa a comunicação entre o nó da rede móvel, MNN, e o nó correspondente, CN. O nó da rede móvel começa por enviar um pacote com destino ao nó correspondente, S:MNN,D:CN,Data. Quando o pacote chega ao *router* móvel, este encapsula-o e envia-o através do túnel bidirecional MRHA para o *home agent*. Quando o *home agent* recebe o pacote, extrai o pacote encapsulado e encaminha-o para a rede como se tivesse tido origem na *home network*. Quando o nó correspondente responde, o pacote segue o caminho inverso, utilizando uma vez mais o túnel bidirecional MRHA.
No protocolo NEMO não há lugar a otimização de rotas, pelo que a utilização desta solução encontra-se sujeita às limitações que são analisadas na secção 2.4. Optimização de rotas.

2.1.3. Mobilidade imbricada de redes

O que identifica uma rede móvel como imbricada é o facto do seu *router* móvel ser um nó visitante móvel de outra rede NEMO. Para o *router* móvel imbricado poder aceder à Internet tem que atravessar a rede NEMO onde se encontra imbricado. A Figura 2.6 ilustra o exemplo de uma rede móvel imbricada, em que o *router* móvel 2, MR2, está dependente do *router* móvel 1, MR1. A rede móvel servida pelo MR1 é conhecida como NEMO1, enquanto a rede móvel do MR2 é conhecida como NEMO2. Nesta figura os túneis MRHA são representados com linhas duplas. A ligação representada com a cor vermelha diz respeito ao túnel MRHA entre o MR2 e o HA2, enquanto que a ligação cinza diz respeito ao túnel MRHA entre o MR1 e o HA1. Fora do túnel, linha a tracejado azul, não há qualquer encapsulação do pacote.
As redes móveis imbricadas acrescentam um conjunto de novos termos que são importantes para compreender este tema, dos quais se destacam os seguintes,

- ***Nested mobile networks* ou *nested* NEMO, rede móvel imbricada**, uma rede móvel diz-se imbricada se estiver ligada a outra rede móvel;

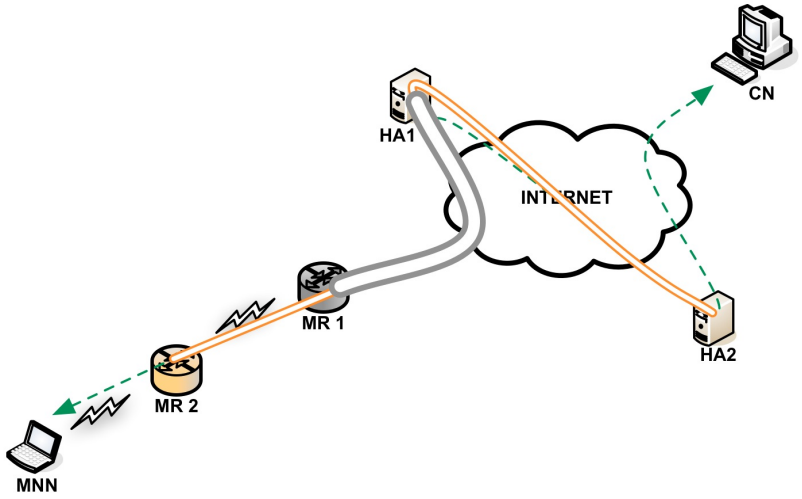


Figura 2.6 – Exemplo de rede móvel imbricada

- ***Root-NEMO***, a rede móvel que está no topo da hierarquia das redes móveis imbricadas, e a que fornece o acesso ao exterior. No exemplo da Figura 2.6, a *root*-NEMO da rede `NEMO2` é a `NEMO1`;
- ***Parent-NEMO***, a rede móvel que fornece acesso a outra rede móvel que está abaixo na hierarquia de acesso ao exterior. No exemplo, o *parent*-NEMO do `NEMO2` é o `NEMO1`;
- ***Sub-NEMO***, a rede móvel que liga à rede móvel acima na hierarquia de acesso ao exterior. No exemplo, `NEMO2` é *sub*-NEMO do `NEMO1`;
- ***Root-MR, parent-MR, sub-MR,*** a posição do *router* móvel na hierarquia da rede imbricada. No exemplo, o `MR1` é o *root*-MR e, também, o *parent*-MR do `MR2`, enquanto o `MR2` é o *sub*-MR do `MR1`;
- ***Profundidade ou nível,*** indica o número de sub-MR que um pacote tem que atravessar desde o nó da rede móvel até ao *root*-MR. Uma rede imbricada de nível 3 indica que são necessários 3 *routers* móveis até que o pacote chegue ao *root-MR*. No exemplo da figura, diz-se que o MR2 está com uma imbricação de nível 1.

O modo de funcionamento do NEMO Basic Support Protocol para o caso das redes imbricadas mantém-se inalterado. Isto é, cada pacote que chegue a um *router* móvel deverá ser encapsulado e enviado para o seu *home agent*, sendo posteriormente removido o encapsulamento e enviado o pacote original para a Internet. No sentido inverso, o pacote é sujeito ao mesmo procedimento.

A Figura 2.7 mostra os passos para a comunicação entre o nó da rede móvel e o seu nó correspondente, de acordo com o exemplo da Figura 2.6.

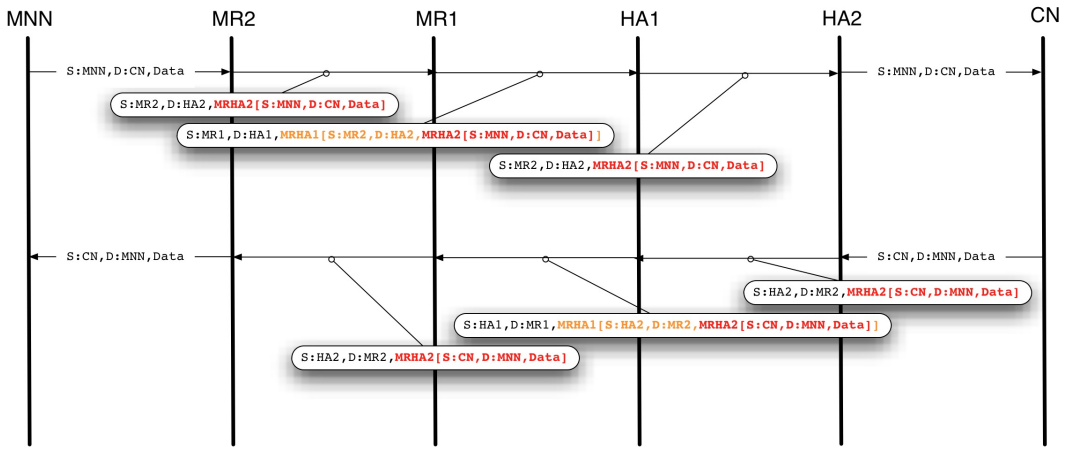


Figura 2.7 – Comunicação entre um MNN e um CN

A representação da figura é semelhante à da secção anterior, sendo que foram acrescentadas caixas com descritivos coloridos de modo a tornar mais visível a informação relevante.

Neste exemplo, o nó da rede móvel MNN envia um pacote com destino ao nó correspondente CN, representado pelo descritivo S:MNN,D:CN,Data. Quando o *router* móvel 2, MR2, recebe o pacote, procede ao seu encapsulamento com destino ao seu *home agent*, HA2 – representado por S:MR2,D:HA2,**MRHA2[S:MNN,D:CN,Data]**.

Quando o pacote alcança o *root*-MR, MR1, este encapsula-o com destino ao seu *home agent*, HA1, como se pode ver pelo descritivo S:MR1,D:HA1,**MRHA1[S:MR2,D:HA2,MRHA2[S:MNN,D:CN,Data]]**. Quando o pacote chega ao *home agent* HA1, este remove o conteúdo do pacote encapsulado e envia-o para o HA2 que, por sua vez, extrai o pacote original e envia-o com destino ao nó correspondente CN.

No sentido inverso, o pacote atravessa os mesmos intervenientes, sendo executadas as mesmas operações, com as devidas adaptações.

A solução apresentada, integrada no NEMO Basic Support Protocol – RFC 3963, é simples e funcional, tendo aplicabilidade imediata na Internet atual. Contudo, a sua simplicidade traz problemas que não são desprezáveis e que merecem um estudo mais aprofundado, apresentado na secção 2.4 Optimização de rotas.

2.2. Separação da localização e identificação

A questão fulcral da problemática da mobilidade está no facto de o endereço IP assumir dupla função: a de localização e a de identificação [Eddy04] [Ratola04]. A localização obtida com o endereço IP corresponde ao caminho topologicamente correto para que um pacote TCP/IP possa atingir o seu destino. Por outro lado, o endereço IP também identifica o destino (interface da máquina) com quem se pretende comunicar.

Na Internet inicial, na qual os equipamentos eram estáticos, fazia sentido que um endereço IP indicasse a forma de alcançar o destino (isto é, a localização) assim como que identificasse o destino pretendido (ou seja, servisse também de identificação). No entanto, a mobilidade de dispositivos veio expor a fragilidade e limitações desta dualidade de funções, que constitui um verdadeiro obstáculo, e apontar para a necessidade de encontrar soluções para que a localização e a identificação sejam separadas. O estudo da mobilidade de equipamentos por este prisma é conhecido como *Loc/ID split* [Meyer08]. Esta secção aborda as soluções que se baseiam nessa separação de funções.

2.2.1. *Locator/ID Separation Protocol* – LISP

O *Locator/ID Separation Protocol* (LISP) [Farinacci12] [Meyer08] [Farinacci11] [Iannone11] [Farinacci11a] [Farinacci11b] tem como base a ideia de que é necessário separar o endereço de encaminhamento para a localização, *Routing Locators* (RLOCs), do endereço que serve de identificação, *Endpoint Identifiers* (EIDs).

Para atingir este fim, o LISP utiliza o método conhecido como *map-and-encap*, que consiste em procurar a localização atual do nó pretendido, associá-la à sua localização e transmitir os pacotes no formato encapsulado até atingir o destino.

O protocolo define, ainda, dois elementos de rede essenciais: o *Egress Tunnel Router* (ETR) e o *Ingress Tunnel Router* (ITR). O ITR é responsável por receber os pacotes de equipamentos que suportem o protocolo LISP, encapsulando-os com destino ao ETR topologicamente mais perto do nó correspondente. Quando chega ao ETR, o pacote original é extraído e encaminhado para o nó correspondente.

O cabeçalho LISP utilizado é constituído pelas duas primeiras secções da Figura 2.8, `UDP packet header` e `LISP`, sendo que a secção do `Internet header` corresponde ao cabeçalho do pacote original.

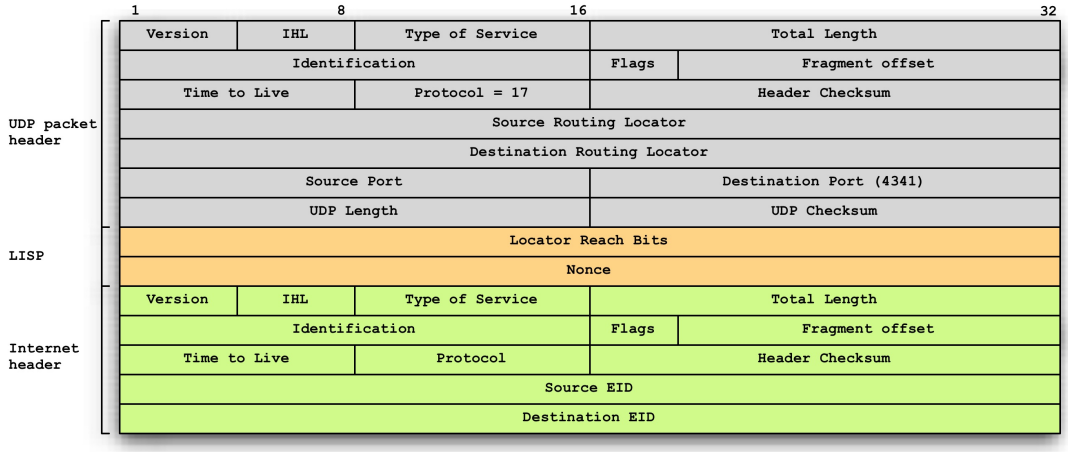


Figura 2.8 – Exemplo de um cabeçalho IPv4 de um pacote LISP

O fluxo da comunicação entre dois nós que utilizem LISP começa com um pedido de DNS regular, inquirindo o IP do nó correspondente. Na resposta, é devolvido um *resource record* A (IPv4) ou AAAA (IPv6) contendo o EID do nó correspondente. Este endereço corresponde ao que foi adquirido na rede visitada.

O nó emissor envia, então, o pacote no formato *standard* que vai ser interceptado por um ITR que é responsável por inquirir a localização (RLOC) associada ao EID, através de um *Map-Request* [Farinacci11c]. O pedido de *Map-Request* vai ser interceptado pelo ETR responsável pela localização topologicamente correta do nó correspondente e deverá devolver um *Map-Reply*.

Quando o ITR recebe o *Map-Reply*, guarda a informação relativa à localização do nó correspondente na sua tabela, para tratamento de pacotes futuros. Os subsequentes pacotes levarão o cabeçalho adicional LISP quando em circulação entre os ITR e ETR.

As questões de segurança são amplamente analisadas em [Madison98] [Eastlake05] [Eastlake11] [Lear96] [Jakab11] [Maino11] [Meyer09].

Os problemas resultados desta proposta concentram-se essencialmente no aumento do tamanho dos pacotes quando encapsulados, associado à adição dos cabeçalhos LISP. Este aumento traduz-se numa diminuição da parcela útil da *Maximum Transmission Unit*, MTU.

O LISP também tem algumas questões não resolvidas no que se refere ao mapeamento EID-to-RLOC, tanto ao nível da latência como ao nível do impacto introduzido pela perda de pacotes, dado que as comunicações se processam por UDP. Para além disto, exige a presença de elementos externos como sejam o ITR e o ETR, aumentando a complexidade da solução.

2.2.2. *Host Identity Protocol* – HIP

O *Host Identity Protocol* (HIP) [Moskowitz08] [Gurtov09] [Moskowitz06] [Nikander08] propõe uma alternativa para resolver o problema da dualidade de funções dos endereços IP. Enquanto o LISP está concebido para resolver o problema da separação da localização e identificação ao nível da rede, o Host Identity Protocol (HIP) está focado em resolver o problema ao nível do nó (*host*) [Gurtov09].

Neste protocolo, cada nó é responsável por criar um, ou mais, par de chaves encriptadas (uma chave pública e outra privada). A parte pública do par de chaves é disponibilizada com recurso ao *Domain Name Server*, DNS [Mockapetris87a] [Mockapetris87b] [Saltzer93] [Gieben04] ou a Distributed Hash Table (DHT) [Gurtov09].

À componente pública da chave dá-se o nome de *Host Identity*. A aplicação de uma função de encriptação sobre a parte pública da chave produz o identificador do nó, conhecido como *Host Identifier Tag*, HIT, que é disponibilizada pelos meios referidos anteriormente.

A comunicação entre dois nós inicia-se estabelecendo uma associação HIP, recorrendo a um processo conhecido como *HIP Base Exchange* (BEX). Através deste processo torna-se possível criar um par de *IPSec Encapsulation Security Payload* (ESP) *Security Association* (SA), um para cada ligação, bem como garantir a identidade do nó correspondente. O cabeçalho do protocolo HIP é apresentado na Figura 2.9.

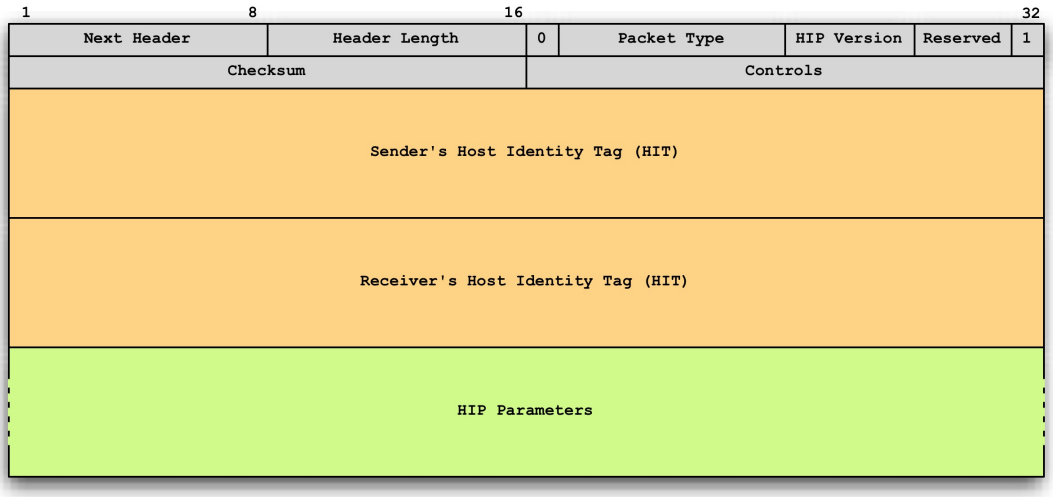


Figura 2.9 – Cabeçalho HIP

Dado que as ligações ao nível da camada de transporte bem como as associações de segurança criadas pelo *HIP Base Exchange* não estão dependentes do endereço IP, um

dispositivo pode mudar o seu endereço de localização mantendo a comunicação segura (através do ESP) com o seu nó correspondente, conforme se pode ver na Figura 2.10.

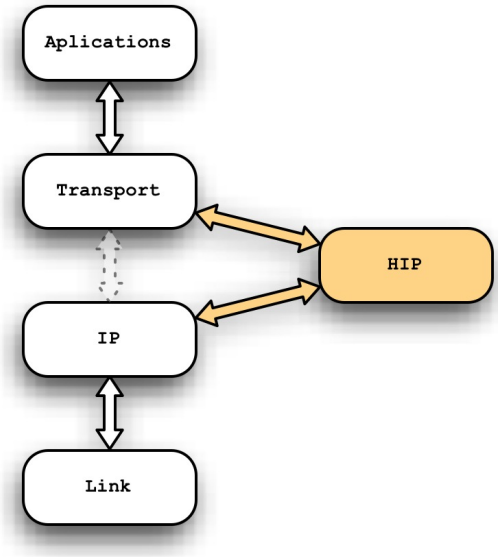


Figura 2.10 – Camada HIP em função ao TCP/IP

A mudança de IP por parte de um nó pode ser atualizada utilizando o *end-to-end three-way UPDATE signaling mechanism* [Nikander08a]. Para o caso em que ambos os nós se movem, deve ser utilizado um servidor *rendezvous* [Laganier08].

O *Host Identity Protocol* permite a identificação dos nós, mesmo no caso em que estes estejam debaixo de uma rede servida com *Network Address Translation* (NAT) [Srisuresh01].

Outra vantagem do HIP consiste no facto das camadas superiores, de transporte e aplicacional, não sofrerem com a alteração do endereço IP, já que para as camadas superiores o endereço com quem estão a comunicar é identificado pelo *host identifier tag*.

Apesar das aplicações da Internet poderem funcionar já com o protocolo HIP, somente as que utilizam o protocolo podem tirar verdadeiro proveito desta solução [Henderson08].

O protocolo implementa diversas medidas que previnem ataques de *denial-of-service* (DoS) e *man-in-the-middle* (MitM). Não obstante, foram criadas as condições para novos ataques de DoS e MitM relacionados com a própria solução que, potencialmente, podem ser mais prejudiciais [Moskowitz06].

2.2.3. Shim6

O Level 3 Multihoming Shim Protocol for IPv6, mais conhecido como shim6 [Nordmark09] [Nordmark05] [Bagnulo07] [Komu08] [Arkko09] [Bagnulo09] [Bagnulo05] [Bagnulo08] [Abley03] [Barré11], tem como principal objetivo fornecer um mecanismo de localização para situações de *multihoming*. Este protocolo faculta uma forma de garantir que as comunicações em curso não são interrompidas enquanto, pelo menos, uma das ligações à Internet estiver ativa.

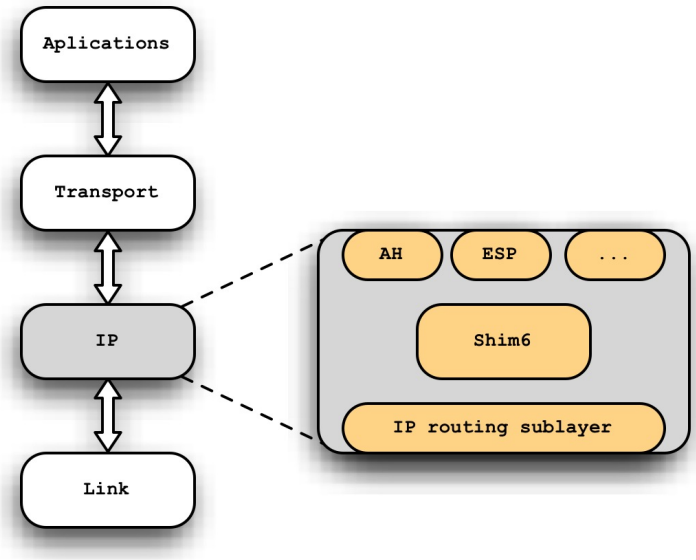


Figura 2.11 – Arquitetura do shim6

Esta solução não é compatível com situações de mudança de IP, pelo que não é objetivo deste protocolo resolver problemas relacionados com mobilidade de nós [Nordmark09]. Contudo, o shim6 pode vir a ser um componente muito útil em futuras soluções de mobilidade, nomeadamente para as questões de otimização de rotas.

O shim6 tem por base o HIP, pelo que adota a aproximação *Loc/ID split*. As camadas de transporte e superiores interagem com o seu nó correspondente através de um endereço IP conhecido como *Upper-layer Identifier*, ULID, enquanto o IP *Locator* é definido ao nível da subcamada *IP routing layer*, conforme mostra a Figura 2.11.

Com esta aproximação, as modificações dos IP *Locator* não têm influência nas camadas superiores, sendo que estas apenas utilizam o ULID. Assim, o nó pode saltar entre as diversas interfaces sem que a comunicação seja interrompida, desde que haja pelo menos uma interface com acesso ao exterior em todos os momentos.

2.3. Soluções acima da camada de rede

As soluções de mobilidade baseadas em mecanismos implementados acima da camada de rede assumem que nesta camada são tratadas todas as questões relacionadas com a localização do nó [Eddy04]. A obtenção de um endereço IP topologicamente correto é conseguida através da funcionalidade de autoconfiguração existente nativamente no IPv6 ou através do DHCPv6. Após a obtenção dos novos endereços, as ligações em curso necessitam de ser atualizadas.

A utilização do já existente *Dynamic Updates in the Domain Name System* [Vixie97] é uma possibilidade viável e com aplicação imediata.

Algumas das vantagens de implementar a mobilidade nas camadas de transporte ou de aplicação são: utilização nativa de otimização de rotas, já que é sempre utilizado o IP topologicamente correto; a não existência do conceito de rede original (*home network*); a inexistência de requisitos ao nível da infraestrutura, para além dos serviços de atribuição de endereços IP já existentes; e a possibilidade de suspensão temporária, sem disrupção, da ligação no caso de se prever a interrupção no acesso ao exterior.

Por outro lado, os problemas resultantes desta aproximação são: a dependência da camada de rede para a gestão da localização; e a necessidade de atualização de diversos protocolos para que esta solução seja uma realidade.

Esta secção analisa as soluções baseadas, direta ou indiretamente, na resolução do problema da mobilidade com recurso à camada de transporte.

2.3.1. *Mobile Stream Control Transmission Protocol* – mSCTP

O *Stream Control Transmission Protocol*, SCTP, [Stewart04] [Stewart00] [Stone02] [Stewart06] [Tuexen07] [Tuexen11] possui capacidades de *multi-streaming* e de *multi-homing*. No caso particular do *multi-homing* torna-se possível utilizar o SCTP como um protocolo para suporte de mobilidade IP, sem necessidade de modificar a infraestrutura de rede existente.

O mobile SCTP, mais conhecido como mSCTP [Koh04] [Imtiaz11], utiliza o *ADDIP extension* [Koh08] [Budzisz08] [Ahmed10] [Stewart07] de modo a que um nó possa notificar eventuais mudanças do seu endereço IP ao nó correspondente.

Como o SCTP pertence à camada de transporte, conforme se pode ver na Figura 2.12, as aplicações têm que ser alteradas de modo a usufruírem deste protocolo.

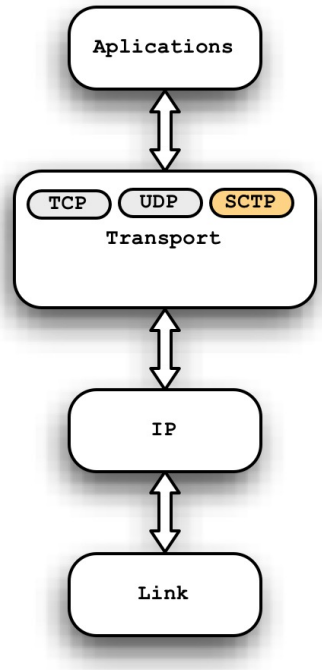


Figura 2.12 – Arquitetura do SCTP

Este protocolo pressupõe que o nó correspondente já conhece pelo menos um dos endereços IP que o nó móvel possui. Existem algumas propostas para resolver o caso em que isso não se verifica [Imtiaz11].

O procedimento de mobilidade com recurso ao mSCTP é constituído por diversos passos que são realizados de forma cíclica, sempre que o nó muda de rede, dos quais se destacam:

- Obtenção do novo endereço IP, enquanto ainda se encontra numa fase de transição (*multi-homing*);
- Adição do novo endereço à associação SCTP; esta informação é transmitida para o nó correspondente através do envio de um pacote SCTP ASCONF; o nó móvel deve receber um ASCONF-ACK como resposta do nó correspondente;
- A determinada altura, quer seja por perder contacto com a rede antiga, quer seja por qualquer outra razão, o nó móvel modifica o seu endereço IP primário para o novo endereço obtido na rede estrangeira;
- Remoção do endereço IP referente à anterior rede de acesso.

Com esta aproximação, o nó móvel pode alterar o seu endereço IP sem que as ligações sejam interrompidas. A comunicação entre o nó móvel e o seu nó correspondente deve ser feita de forma segura [Tuexen05].

De facto, esta solução parece promissora dado que não requer a intervenção da infraestrutura para obter a mobilidade dos nós. O aspecto negativo desta proposta reside na necessidade, inerente ao SCTP, de modificar todas as aplicações de modo a passarem dos largamente utilizados protocolos TCP e UDP para o SCTP, devendo contemplar a variante da extensão mSCTP.

2.3.2. Mobilidade baseada em SIP

O *Session Initiation Protocol* (SIP) [Rosenberg02] [Handley99] [Schulzrinne99] [Rosenberg99] [Schulzrinne00] é um protocolo de sinalização que faculta os mecanismos necessários para que dois ou mais participantes possam criar e gerir sessões múltiplas de multimédia [Prasad05] [Schulzrinne00a]. Este protocolo foi desenvolvido no âmbito do *Multiparty Multimedia* (MMUSIC[7]) *working group*, no IETF.

O SIP é usado para mobilidade de equipamentos porque suporta os serviços de mapeamento e redireccionamento de nomes [Pandya95] [Rosenberg02a] [Marples00] [Wedlund99].

Os endereços SIP são do mesmo formato que o utilizado para o correio electrónico, i.e., `sip:utilizar@realm.tld`. Aproveitou-se, mesmo, para reutilizar alguma da infraestrutura de entrega de mensagens electrónicas, tal como o registo de DNS (*resource record*, RR) *mail exchange* (MX) ou o SMTP EXPN para expansão de endereços. Embora não seja forçoso que assim seja, tem-se verificado que os utilizadores utilizam pelo menos um dos seus endereços de correio electrónico como *uniform resource locator* (URL) SIP. A Figura 2.13 ilustra uma comunicação SIP.

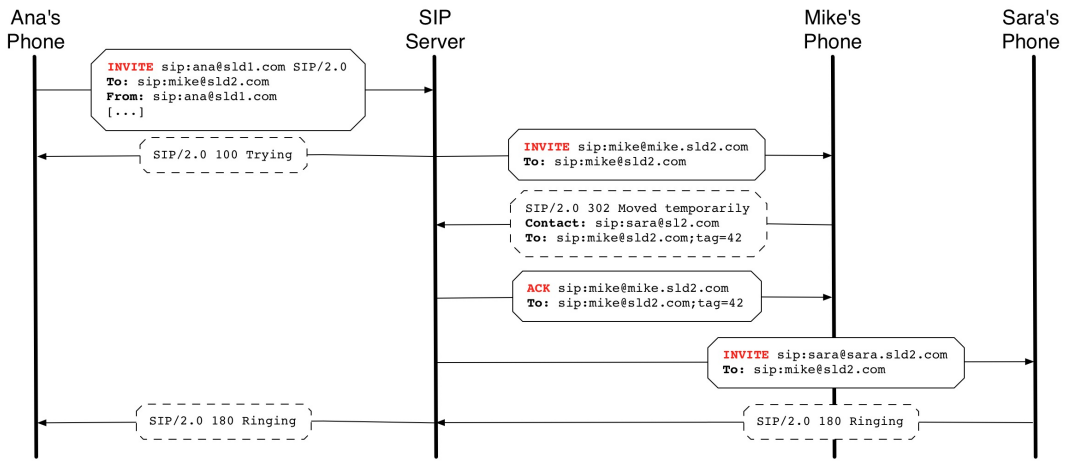


Figura 2.13 – Início de sessão multimédia com recurso ao SIP

[7] http://datatracker.ietf.org/wg/mmusic/charter/

O SIP é independente da camada de transporte, podendo ser utilizado um mecanismo de entrega não resiliente, já que é o próprio protocolo a garantir a fiabilidade da comunicação. Contudo, por norma é utilizado o protocolo UDP ou o TCP.
Os pedidos e respostas SIP consistem num cabeçalho de texto e corpo no formato MIME, muito similar ao formato dos pedidos HTTP.
O SIP define o seguinte conjunto de entidades lógicas:

- ***user agent***, são as entidades responsáveis por iniciar e terminar os pedidos SIP, tais como *software* de conferência ou de voz;
- ***redirect server***, recebe os pedidos e devolve as respostas a indicar para onde devem ser realizados os pedidos seguintes, se houver necessidade de tal;
- ***proxy server***, pode ser um simples encaminhador de pedidos para outro servidor (*stateless proxy*) ou pode manter o estado de uma transação, os pedidos e respetivas respostas que pertencem a este pedido (*stateful proxy*);
- ***registrar server***, aceita pedidos de registo e coloca a informação que receber desses pedidos num serviço de localização para o domínio pelo qual está responsável.

Usualmente, um servidor SIP implementa simultaneamente um *registar*, *redirect* e *proxy server*. A localização de um servidor SIP encontra-se definida num DNS *resource record* (RR) SRV [Gulbrandsen00]. Os *user agents* registam-se junto de um *registrar* local [Kempf00].
O SIP utiliza o seguinte conjunto de métodos na especificação base:

- ***INVITE***, para iniciar uma sessão;
- ***ACK***, para confirmar o estabelecimento de uma sessão;
- ***BYE***, para terminar uma sessão;
- ***OPTIONS***, para definições adicionais;
- ***CANCEL***, para terminar uma sessão em processo de estabelecimento.

No que concerne a operação de registo entre o nó móvel e seu nó correspondente, existe alguma semelhança com a mobilidade IP [Perkins10]. Contudo, enquanto o MIP faz um mapeamento entre o IP permanente, *home address*, e o IP temporariamente atribuído na nova rede, *care-of address*, o SIP faz a associação entre o identificador do utilizador e os endereços IP que o nó móvel for adquirindo.
Deste modo, quando um nó correspondente pretender comunicar pode fazer uso das propriedades do SIP para localizar o IP atualmente em uso pelo nó móvel, estabelecendo sempre uma comunicação optimizada.

Existe outra proposta que aborda a utilização do SIP para a mobilidade específica de redes, conhecida como SIP-NEMO [Lee06]. Neste caso, o *router* móvel, aqui designado SIP *Network Mobility Server* (SIP-NMS), executa as operações de optimização de rotas baseadas em SIP por vez dos clientes finais e faz operações de *network address translation* (NAT) de modo a que os nós da rede móvel não tenham que saber da sua condição de mobilidade.

Como entrave para a utilização da solução de mobilidade baseada em SIP está a necessidade das aplicações terem que suportar SIP e fazerem uso dela para comunicar com os outros equipamentos.

2.3.3. Mobilidade de nós baseada em DNS

A mobilidade de dispositivos baseada em DNS explora as vantagens que este serviço tem em termos de mapeamento entre nomes (isto é, identificadores) e endereços IP [Conti01] [Cheng05] [Adjie-Winoto00].

Esta proposta preconiza que a comunicação entre dois nós seja sempre feita com recurso ao serviço de nomeação, *domain name server* (DNS). Um dispositivo realiza um *query* de DNS, com a operação *gethostbyname*, para obter o endereço IP do nó que pretende contactar e, posteriormente, utiliza a resposta obtida para preenchimento do cabeçalho IP. Deste modo, é determinada a localização do nó com base num protocolo que já se encontra bem cimentado na Internet – o protocolo DNS. A simplicidade do mecanismo torna-o bastante aliciante como solução para a determinação da localização de nós móveis, embora levante vários problemas.

Neste contexto são, tipicamente, considerados dois tipos de mobilidade: a fraca e a forte.

Na mobilidade fraca (*weak mobility*) as ligações são interrompidas quando o nó se move, sendo necessário o restabelecimento das comunicações utilizando o novo endereço IP, entretanto fornecido pelo DNS. Se um nó móvel está a transferir um ficheiro utilizando o protocolo *file transfer protocol*, FTP, por exemplo, e adquire um novo IP, na mobilidade fraca torna-se necessário reiniciar o processo de transferência do ficheiro.

Na mobilidade forte, não há quebra de conectividade IP, mesmo quando o nó se move. Este cenário contempla situações como a de *streaming* de vídeo, chamadas VoIP, etc, nas quais não é desejável que a transferência de informação seja interrompida.

Para a mobilidade fraca a solução proposta implica a alteração dos servidores de DNS, de modo a contemplar a informação relativa à localização do nó móvel. Sempre que o nó móvel adquire um novo endereço IP, isto é, um *care-of address, CoA*, deve informar o seu servidor de DNS, sendo essa informação guardada durante um determinado tempo (*TTLbinding*). Durante este período, sempre que for feito algum *query* a resposta conterá o *care-of address*.

É da responsabilidade do nó móvel, MH, enviar uma atualização, *binding update*, a informar que ainda está a utilizar o CoA. Se não o fizer, este expirará e o servidor de DNS passará a responder com o IP original, *home address – HoA*.

A configuração destes endereços é feita sem recurso a *cache*, TTL=0, pelo que os servidores de DNS intermediários nunca guardarão esta informação.

Por motivos de escalabilidade, é aconselhada a criação de um *second level domain*, SLD, específico para este propósito, como mostrado na Figura 2.14. No exemplo, o *top level domain*, `TLD`, criou um `SLD mobile` e, abaixo desse, cria subdomínios que pode delegar noutras unidades orgânicas como, no exemplo, Universidade de Coimbra (`uc`).

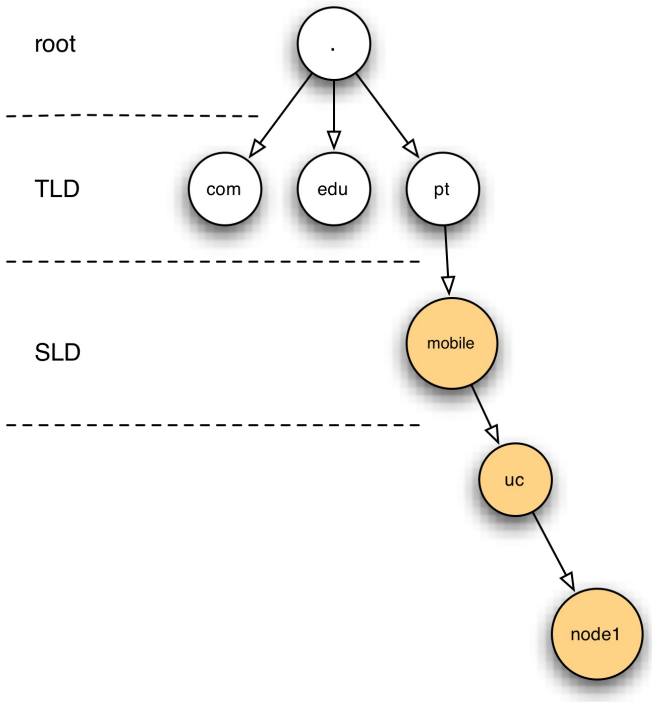


Figura 2.14 – Exemplo de árvore de nomeação para o nó móvel node1.uc.mobile.pt.

Para a mobilidade forte a solução proposta vai de encontro à que é utilizada no *mobile IP*, MIP. De facto, a manutenção da sessão é um dos desafios para esta aproximação [Henderson03a]. Para que tal seja possível sem recurso a uma solução tipo MIP é necessária a modificação dos nós finais.

Mesmo que se optasse por esta solução, seria necessário inquirir o DNS antes de enviar cada pacote, o que seria um incremento inaceitável de tráfego.

É importante referir que o protocolo de nomeação DNS não é o único existente (outros possíveis são: INS, JINI, UPnP) pelo que a implementação desta solução pode não abranger a mobilidade para toda a Internet.

Para contornar esta limitação chegou-se a propor uma nova aproximação à mobilidade da Internet [Snoeren01] com recurso a uma arquitetura chamada *Migrate*. Nesta solução, muito ao estilo do mSCTP, é aconselhada a extensão do protocolo TCP de modo a incluir o controlo de sessões. Claro está que o contra desta solução é a necessidade de alterar as aplicações para suportar esta funcionalidade.

2.4. Optimização de rotas

As soluções propostas na secção 2.3 visam resolver o problema da mobilidade tanto para um dispositivo isolado como para uma rede. Optou-se por alargar a abrangência do estudo de modo a confirmar se alguma das soluções de mobilidade para um nó também serviria para as redes móveis. No entanto, nenhuma das soluções apresentadas é comummente aceite como solução para a mobilidade de redes.

Analisadas as diferentes soluções de mobilidade nas diversas camadas protocolares, optou-se por centrar o presente trabalho na mobilidade baseada na camada de rede, não só por ser essa a camada na qual a comunidade científica da área aposta mais fortemente, mas também porque é nessa camada que as principais soluções normalizadas de mobilidade operam. No entanto, não significa isto que soluções baseadas noutras camadas não tenham um potencial interessante, nomeadamente as que se baseiam na camada de transporte.

Conforme foi explicado na secção 2.1.2, a solução consensual para esta área é o *network mobility*, NEMO. Apesar de ser uma solução simples e funcional, com aplicabilidade imediata na Internet, apresenta alguns problemas que são bastante limitativos.

Esta secção começa por analisar estes problemas, para depois apresentar as soluções mais relevantes para os contornar, na vertente de otimização de rotas ao nível da camada de rede.

2.4.1. Análise do problema

As limitações do NEMO Basic Support Protocol, tais como a *triangular routing*, potencial estrangulamento na *home network* e amplificação da falta de otimização para redes imbricadas, são motivo de um estudo aprofundado em [Ng07], [Ng07a] e [Bernardos05a]. Os problemas desta solução são, no essencial, devidos à extrema simplicidade do protocolo proposto no RFC 3963 que, sendo o seu ponto forte reflete, simultaneamente, o seu lado mais fraco.

No topo da lista das consequências da não utilização da optimização de rotas aparece o ***Triangular routing*** que, conforme se pode verificar na Figura 2.15, obriga a que as

comunicações entre o nó da rede móvel MNN e o seu nó correspondente CN passem pelo *home agent* HA, através de um túnel encapsulado conhecido como *MRHA tunnel*. Este problema também é conhecido como ***pinball route***.

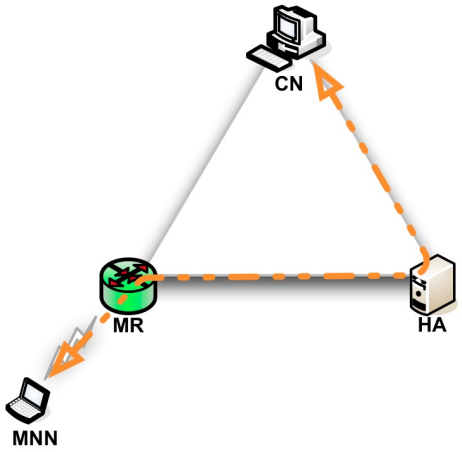


Figura 2.15 – *Triangular routing* no NEMO Basic Protocol

Os efeitos que advêm desta condição são:

- ***Caminhos mais compridos*** – como o pacote que circula entre o nó da rede móvel e o nó correspondente tem que passar pelo *home agent* da rede móvel, o tempo de trânsito é superior ao que levaria se o pacote fosse diretamente de um extremo ao outro, sem passar pela *home network*; este caminho mais longo pode trazer problemas a aplicações com necessidades de transmissão de dados em tempo real ou, até, ao protocolo TCP, cuja cadência de envio de pacotes é determinada pelo *round trip time* (RTT); dado que são atravessados mais equipamentos para a comunicação entre os dois nós, torna-se perceptível o impacto introduzido na infraestrutura pela utilização desta solução em comparação com a comunicação direta entre os dois nós;
- ***Processamento adicional do pacote*** – o túnel encapsulado entre o *router* móvel e o *home agent* força o incremento do tamanho dos pacotes, dado que os pacotes originais são encapsulados em novos pacotes, que acrescentam outro cabeçalho IP e, opcionalmente, um conjunto de extensões; naturalmente, a eficiência da comunicação é afetada por este incremento no tamanho; por outro lado, o processo de encapsular e desencapsular um pacote introduz um atraso na transmissão pois é necessário executar estas tarefas; o processamento destes pacotes leva a um incremento na carga do CPU dos *router* móvel e *home agent*;

- ***Maior probabilidade de fragmentação*** – o incremento do tamanho dos pacotes devido ao túnel MRHA aumenta a probabilidade de ocorrer fragmentação, principalmente se não for realizada uma operação de descoberta do *maximum transmission unit* (MTU), ou se não for tido em consideração o túnel MRHA que, por sinal, é transparente para os nós finais (ou seja, é muito provável que não seja tido em consideração); a fragmentação de pacotes traduz-se num incremento ainda maior do atraso na comunicação, numa redução adicional da eficiência da comunicação, na maior probabilidade de ocorrência de problemas de processamento na reconstrução dos pacotes ou, ainda, de problemas de *firewall* que não permitam a correta transmissão dos pacotes fragmentados;
- ***Maior exposição a falhas de ligações*** – dado que o pacote tem que atravessar um caminho mais longo, através do *home agent*, torna-se evidente o incremento no número de equipamentos que intervêm no processo, sendo que a probabilidade de haver uma falha na comunicação aumenta.

Outra das consequências de não se optar por uma solução de otimização de rotas é o potencial **estrangulamento** que pode ocorrer na *home network*.

Numa solução de implementação do protocolo NEMO em contexto real, podemos considerar um cenário como o exposto na Figura 2.16. Neste exemplo, considera-se que um *Internet Service Provider* (ISP) fornece acesso a várias redes móveis. Parece fazer sentido que a opção economicamente viável seja a de agregar diversos *routers* móveis ao mesmo *home agent*, fazendo com que o tráfego tenha que atravessar o mesmo ponto central: o *home agent*.

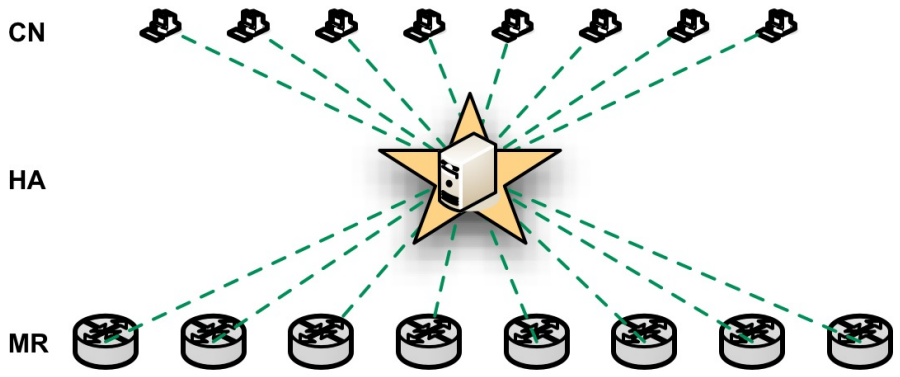


Figura 2.16 – Estrangulamento na *home network*

Devido ao estrangulamento na rede origem, vários problemas podem ocorrer:

- ***Congestionamento na* home network** – ao forçar o tráfego de diversas redes móveis a atravessar o mesmo *home agent* vai-se criar um potencial ponto de congestionamento, fazendo com que o tráfego esteja sujeito a atrasos ou mesmo a não processamento de pacotes e sua consequente perda;
- **Processamento adicional** – por exemplo, devido a operações de verificação de segurança – *autorizo o pacote para uma determinada rede móvel?* –, intercepção de pacotes destinados a redes móveis em trânsito, encapsulação/desencapsulação de pacotes, recepção dos pacotes de *binding* e respetiva resposta; este processamento adicional torna o funcionamento menos optimizado do que o simples encaminhamento do tráfego;
- **Ponto único de falha** – dado que todo o tráfego das diversas redes móveis tem que atravessar o mesmo *home* agent, cria-se um ponto único de falha na rede; qualquer perda de conectividade neste ponto traduz-se na perda irrecuperável de comunicação até que o problema seja resolvido;
- **Atrasos ou perdas de pacotes de *binding*** – podem ser um factor adicional de problemas no estabelecimento de novos túneis ou podem mesmo resultar na quebra de conectividade de túneis existentes.

A problemática da falta de otimização de rotas assume um papel mais acentuado quando se analisa a situação do ponto de vista das redes imbricadas. Estas questões são designadas de **amplificação da falta de otimização em redes imbricadas**.

Conforme referido anteriormente, o NEMO Basic Support Protocol permite que redes móveis se juntem a outras redes móveis, de forma sucessiva, criando vários níveis de imbricação. Esta possibilidade faz com que a falta de otimização de rotas e o subsequente efeito *triangular routing* seja amplificado a cada nível de imbricação. Na Figura 2.17 é possível visualizar um exemplo do problema.

O *router* móvel de topo, `MR1`, tem três *routers* móveis imbricados, `MR2`, `MR3` e `MR4`. Qualquer pacote com origem no `MNN3` com destino ao nó da rede móvel `MNN4`, vai ser encapsulado pelo `MR3` com destino ao `HA3`. Quando o pacote chega ao `MR2`, este volta a ser encapsulado no túnel MRHA agora com destino ao `HA2`. Por fim, o *root-MR* `MR1` vai encapsular o pacote proveniente do `MR2` com destino ao `HA1`.

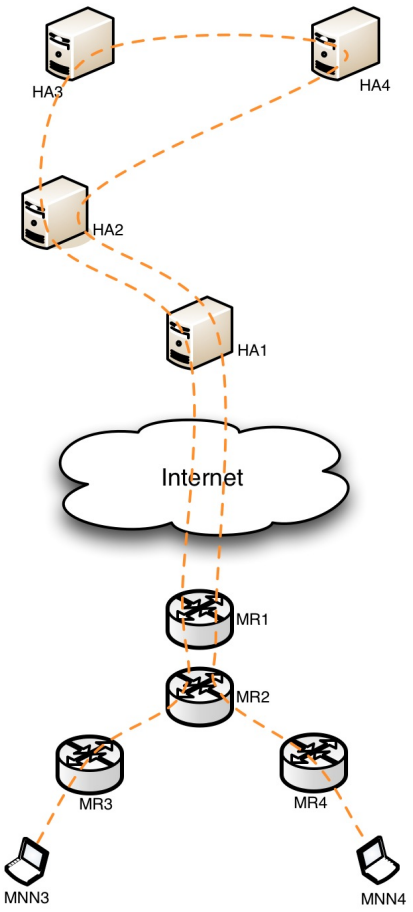


Figura 2.17 – Amplificação da falta de otimização em redes imbricadas

Sempre que um pacote é encapsulado, o seu tamanho aumenta, conforme se pode verificar na Figura 2.18. O `encapsulation #1` diz respeito ao pacote quando encapsulado pelo `MR3`, enquanto o `encapsulation #2` diz respeito ao encapsular pelo `MR2` e o `encapsulation #3` diz respeito ao túnel MRHA do `MR1`.

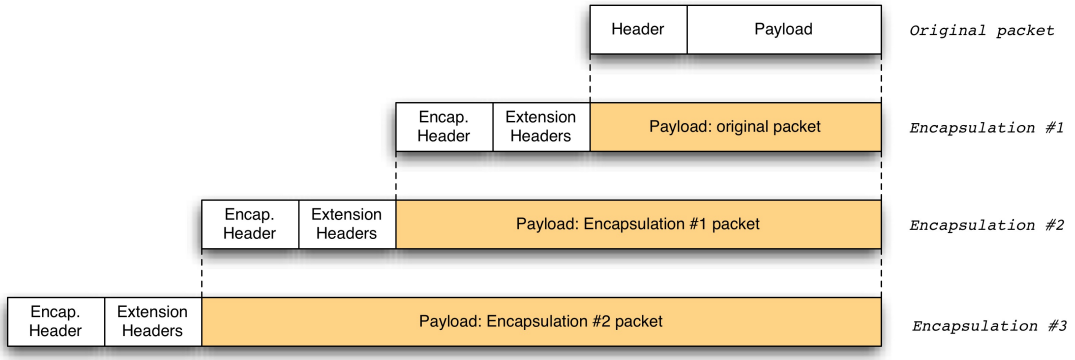


Figura 2.18 – Encapsulação dos túneis MRHA para redes imbricadas

Quando o pacote atinge o HA1, procede-se à sua extração e consequente envio para a Internet, neste caso com destino ao HA2. Quando finalmente atinge o HA3, o pacote que vai para a Internet é o originalmente enviado pelo nó da rede móvel MNN3 e segue com destino à rede original do *router* móvel MR4.

Chegando a essa rede é interceptado pelo *home agent* HA4, e segue-se o processo de encapsulação e extração sucessiva até atingir o nó da rede móvel MNN4.

Todo este processo diz respeito apenas ao caminho que o pacote levou no sentido do nó móvel MNN3 para o nó móvel MNN4. A resposta deverá seguir o caminho inverso.

Neste pequeno exemplo é visível o impacto que a solução NEMO Basic Support Protocol introduz na comunicação, fazendo com que o caminho entre os dois equipamentos incremente em função da profundidade da imbricação, O tamanho do pacote é incrementado em função ao nível de imbricação, incrementando a probabilidade de ocorrerem os problemas abordados anteriormente.

É importante, ainda, notar que se a comunicação do MR1 para o exterior for interrompida, a comunicação no interior das redes imbricadas é igualmente interrompida, mesmo que o caminho interno esteja operacional.

O problema anterior aplica-se também ao caso de um **nó móvel visitante, *visiting mobile node* – VMN**. A Figura 2.19 tem um exemplo do percurso necessário para a comunicação entre o nó móvel visitante e o seu nó correspondente.

No caso de não haver nenhuma otimização de rotas, então o caminho entre os dois nós é o indicado pela linha 1. Este caminho é essencial se for necessário recorrer ao procedimento de *Return Routability*.

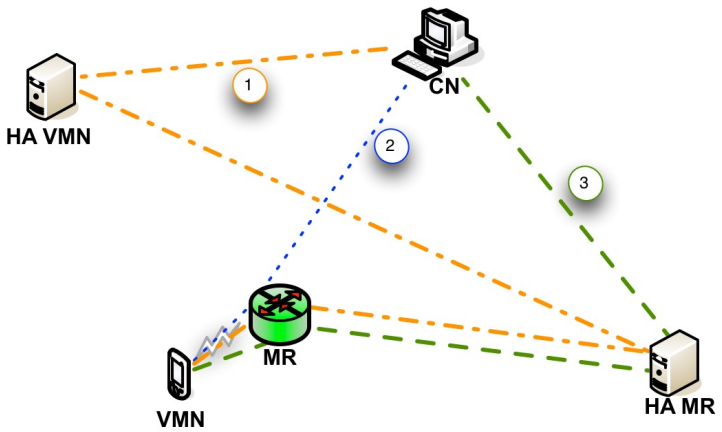


Figura 2.19 – Possíveis percursos para comunicação entre um VMN e CN

Mesmo após o estabelecimento da otimização de rotas, utilizando o MIPv6, o caminho nunca poderá ser o da linha 2, porque a comunicação entre os dois nós tem que atravessar sempre o túnel MRHA entre o MR e o HA MR. Assim, no melhor dos cenários, em que a comunicação entre os dois nós é otimizada, o caminho é o indicado pela linha 3.
Naturalmente, as consequências abordadas na amplificação da falta de otimização nas redes imbricadas assumem um papel mais agravado neste cenário, já que o esforço adicional de otimização entre os dois nós não tem o efeito desejado. Pior ainda, não existe forma do nó móvel visitante ter consciência desta situação.

Estas questões motivam a implementação de mecanismos de otimização de rotas que resolvam ou mitiguem os problemas de encaminhamento não otimizado do NEMO Basic Support Protocol. De seguida, são abordadas algumas propostas nesse sentido.

2.4.2. *Path Control Header*, PCH

O *Path Control Header*, PCH [Na04] [Na04a] [Na04b] fornece um mecanismo de estabelecimento de rotas otimizadas com recurso à inclusão de informação essencial no pacote original a circular dentro do túnel MRHA.

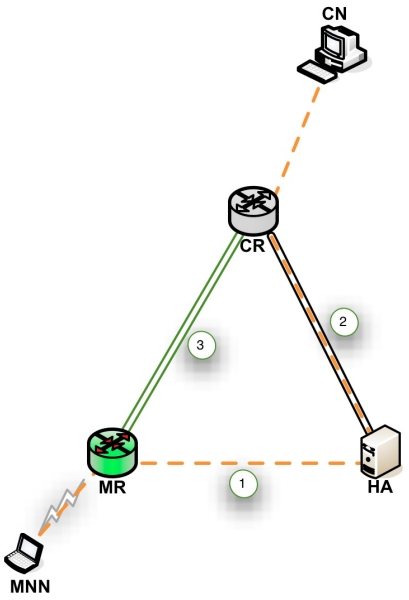


Figura 2.20 – Funcionamento do *Path Control Header*

Nesta proposta é assumido que a infraestrutura deve ser alterada de modo a criar o melhor caminho (otimizado) entre os equipamentos finais, de forma transparente e automática.
Sempre que um nó começa a comunicar, a infraestrutura criará um túnel otimizado entre dois pontos centrais, de modo a que subsequentes pacotes tomem o caminho otimizado.
Nesta proposta, os intervenientes principais são o *home agent* e o *router* correspondente, *correspondent router* – CR. Na Figura 2.20 pode-se ver um exemplo de uma comunicação com recurso ao *path control header* e os diversos passos do processo.
Quando o nó da rede móvel MNN envia o primeiro pacote com destino ao nó correspondente CN este toma o caminho identificado pela linha 1.
Quando o pacote é extraído do túnel MRHA pelo *home agent* HA este acrescenta ao cabeçalho o campo opcional *Path Control Header* (linha 2) contendo o *Care-of Address*, CoA, do *router* móvel MR. Se no caminho até ao nó correspondente existir algum *router* correspondente que compreenda PCH, então deve utilizar esta informação de modo a criar uma rota otimizada até ao *router* móvel, identificado na figura pela linha 3.
Quando contatado, o autor e principal mentor, JongKeun Na, informou que acabara por abandonar esta solução por existirem diversos problemas técnicos, acrescida da dificuldade em implementar esta solução em larga escala e da falta de capacidade em conseguir avaliar de forma imparcial a sua performance.

2.4.3. *Optimised Route Cache Managment Protocol,* ORC

O *Optimized Route Cache Managment Protocol*, ORC [Wakikawa03] [Wakikawa04] possui algumas semelhanças com a PCH. Esta proposta também faz recurso da infraestrutura para otimizar, de forma transparente, o caminho entre os equipamentos terminais.
O mecanismo de otimização inicia-se sempre que haja tráfego para uma rede, sem confirmação da existência, ou não, de um *router* com capacidade de ORC. Neste caso, o *router* móvel envia um *Binding Router* (BR) para o endereço *anycast* da rede a alcançar. Se no trajeto existir algum *router* com capacidade de ORC, então este responde criando um túnel com o *router* móvel. A partir desse instante, todo o tráfego entre os dois equipamentos terminais circula dentro do túnel ORC.
Sempre que o *router* móvel adquire um novo *Care-of Address*, CoA, deve enviar um *Binding Router* (BR) para todos os ORC *routers* com quem tenha estabelecido o túnel.
As questões relacionadas com a implementação em larga escala de *routers* ORC, intercepção dos pacotes de *binding* pelos *routers* ORC mais próximos do nó correspondente ou questões de performance em larga escala não têm solução fácil e são, claramente, limitações da proposta.

2.4.4. Otimização de rotas baseada em ND-Proxy

A proposta de otimização de rotas baseada em *Neighbor Discovery Proxy* (ND-Proxy) [Jeong04] [Jeong04a] pretende fazer uso das funcionalidades existentes no *Mobile IPv6*, passando para o nó da rede móvel o ónus de otimizar as rotas com os seus nós correspondentes.

Para tal, o *router* móvel faz o reencaminhamento dos pacotes de *Neighbor Discovery* para dentro da sua rede móvel, permitindo que os nós da rede móvel obtenham um endereço IP com o prefixo da rede visitada.

Na Figura 2.21 pode-se ver o exemplo de uma rede móvel que se associa à rede do *router* de acesso AR. Após o *router* móvel obter o endereço *Care-of Address*, CoA, começa a propagar os pacotes de *Neighbor Discovery* da rede visitada para dentro da sua rede móvel.

A partir desse instante, a rede 2 fica com o mesmo prefixo da rede 1. A estas redes os autores chamam de *Multilink Subnet* (MS), consistindo em várias redes distintas mas que partilham o mesmo prefixo.

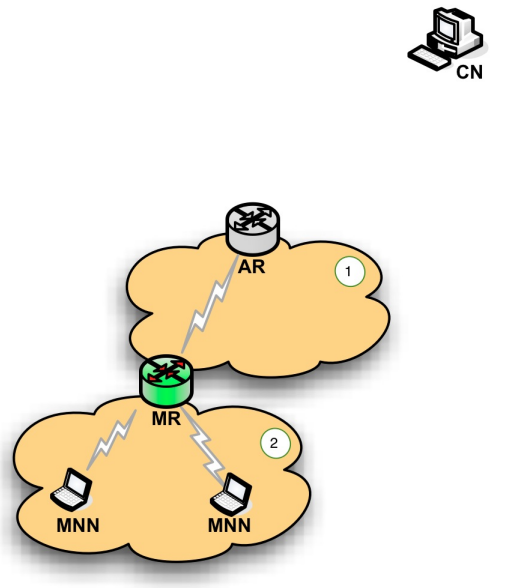


Figura 2.21 – Otimização de rotas baseada em ND-Proxy

Neste caso, o *router* móvel assume o papel de *Multilink Subnet Router* (MSR), um equipamento com capacidade de reencaminhar pacotes de *Neighbor Discovery* (ND), e de encaminhar pacotes dos nós da rede móvel.

Assim que o nó da rede móvel adquire um IP da rede visitada, este passa a ser o seu *Care-of Address*, CoA, que poderá utilizar para estabelecer as rotas otimizadas com os nós correspondentes.

Qualquer pacote enviado pelo nó da rede móvel que seja recebido pelo *Multilink Subnet Router* deve ser analisado de modo a verificar se deve ser encaminhado para o exterior, para a rede móvel interna ou encapsulado dentro do túnel MRHA.

Esta proposta, apresentada em 2004, implica a alteração dos equipamentos finais de modo a reconhecer esta nova funcionalidade. Nessa altura, o foco da investigação estava vocacionado para a maior transparência possível da mobilidade para os nós da rede móvel. Por outro lado, a questão de encaminhamento dos pacotes com o prefixo da rede visitada dentro de outras redes apresentava-se como um desafio técnico complexo.

2.4.5. Global HA to HA Protocol, Global HAHA

O *Global HA to HA Protocol* [Thubert06] [Thubert09] [Ayaz09], que se encontra baseado no *Inter Home Agents Protocol* [Wakikawa06], serve para resolver os problemas do *triangular routing*, para mitigar as questões de escalabilidade e para garantir a redundância do *home agente*.

Esta proposta está vocacionada para situações em que a distância entre os *router* móvel (MR), *home agent* (HA) e correspondent node (CN) é tal que ||MR,HA||+||HA,CN|| é muito maior que ||MR,CN|| , estando-se a pensar, neste caso, numa dimensão global. Normalmente, os cenários que justificam esta proposta são aqueles cujo *router* móvel ou o nó móvel se encontram fora do país de origem.

O *Global* HAHA define alguns novos conceitos importantes, como o de HA primário, que é o primeiro *home agent* ao qual o *router* móvel se associa, ou seja, é o seu *home agent* original. No exemplo da Figura 2.22, o *home agent* primário é o HA MR. O *home agent* secundário é o *home agent* ao qual o *router* móvel se vai associar quando se encontra fora da sua rede topologicamente correta. Por fim, o *proxy home agent* é todo o *home agent* que encaminha pacotes para o *home agent* primário ou secundário.

O funcionamento do *Global* HAHA tem início quando o *router* móvel se encontra fora da sua rede e procura um *proxy home agent* próximo de si. No caso de existir, o MR regista-se nesse agente, que se encarrega de informar o *home agent* primário e informar os *proxy home agent* da existência deste *router* móvel, bem como os respetivos prefixos utilizados. Assim, passa a estar garantido o *routing* interno dentro dos *routers* HAHA.

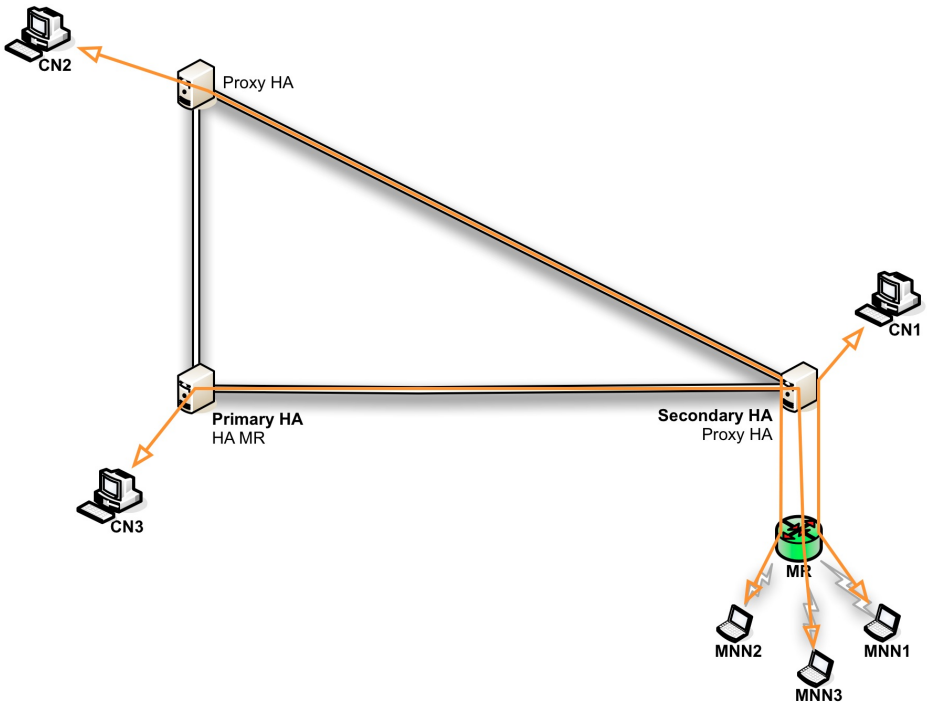


Figura 2.22 – Exemplo de comunicação usando Global HAHA

Após esta fase de registo (*binding update* e *binding acknowledgement*) do *router* móvel com o *home agent* e entre os *home agents* envolvidos, a comunicação entre os equipamentos `MNN1` e `CN1` não necessitará de passar pelo *home agent* principal do *router* móvel. A comunicação entre o nó `MNN2` e o seu nó correspondente, `CN2`, também não necessitará de passar pelo HA principal.

Nesta proposta, é assumido que a sobrecarga de processamento para encapsular e extrair os pacotes pelos *home agent* é desprezável, pelo que a comunicação entre o `MNN3` e o `CN3` segue o caminho ||`MR`,`Secondary HA`||+||`Secondary HA`,`Primary HA`||.

O facto da solução estar vocacionada para longas distâncias, razão pelo qual é amplamente discutida nos cenários de aviação, limita um pouco o âmbito de utilização quando se pretende obter uma solução global e independente do cenário.

2.4.6. *Hierarchical Mobile IPv6*, HMIPv6

O *Hierarchical Mobile IPv6*, HMIPv6 [Soliman08] [Perez-Costa03], é uma proposta que visa proporcionar novas funcionalidades ao protocolo *Mobile IPv6*, e que pretende resolver dois problemas distintos: o dos nós que se movem muito rapidamente entre redes diferentes adquirindo, sucessivamente, novos *Care-of Address;* e, por outro lado, o de equipamentos que

se encontrem muito distantes do seu *home agent*, estando sujeitos a uma latência significativa devido ao *triangular routing*.

Para se poder falar sobre HMIPv6 é importante tomar conhecimento dos seguintes elementos que foram adicionados à terminologia de mobilidade de nós,

- ***Mobility Anchor Point*, MAP** – *router* localizado na rede visitada pelo nó móvel e que tem capacidade de operar como um *home agent* local;
- ***Regional Care-of Address*, RCoA** – um endereço IP atribuído pelo *router* MAP ao nó móvel para ser usado como *Care-of Address*;
- ***On-Link Care-of Address*, LCoA** – é o endereço IP atribuído pelo *router* de acesso da rede onde o nó móvel se ligou; este endereço é diferente do RCoA;
- ***Local Binding Update*** – a operação realizada pelo nó móvel para se registar no MAP.

Este protocolo não é exclusivo, podendo ser usado em simultâneo com o MIPv6. Um nó móvel que se associe a uma rede que suporte HMIPv6 pode usar, para falar com o nó correspondente, os seguintes endereços IP: o *home address* – neste caso, a comunicação com o nó correspondente deve passar pelo túnel entre o *home agent* e o nó móvel, através do *router MAP*; o *On-Link Care-of Address*, LCoA, podendo criar rotas otimizadas MIPv6 com os nós correspondentes; ou, por fim, o *Regional Care-of Address*, RCoA, em que a otimização de rotas MIPv6 é realizada pelo *router* MAP em vez do nó móvel.

O procedimento do HMIPv6 começa quando o nó móvel chega a uma rede que suporte este protocolo, de acordo com o qual o MN deve procurar um MAP *router*, utilizando o procedimento conhecido como *MAP Discovery*.

No caso de encontrar um MAP *router*, o nó móvel deve realizar a operação de *binding update* utilizando o seu LCoA, e obtendo um novo RCoA, que deve utilizar para fazer o *binding update* com o *home agent* do nó móvel.

No exemplo da Figura 2.23 pode-se ver o nó móvel `MN` que se encontra, inicialmente, na rede do *router* `MAP1`, associado ao *access router* `AR` mais à esquerda. Assim que as operações de *binding* são concluídas com sucesso, a comunicação entre o `MN` e o `CN` é processada através do caminho indicado pela linha `1`, se for utilizada a rota otimizada usando o protocolo HMIPv6.

Se a comunicação entre o `MN` e o `CN` se processar usando o *home address*, o caminho utilizado é o indicado pela linha `h`.

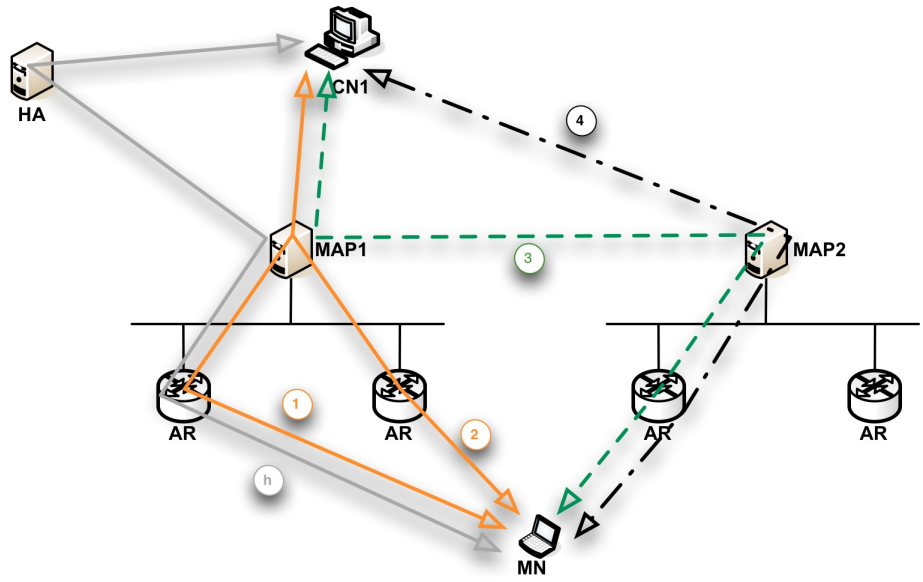


Figura 2.23 – Vários cenários de HMIPv6

Quando o MN transitar de *router* AR dentro da mesma rede gerida pelo MAP1, a única operação que o nó móvel tem que fazer é o *local binding update* com o MAP1 a informar o seu novo *On-Link Care-of Address* LCoA. Neste caso, as comunicações continuam a processar-se sem qualquer alteração da parte do HA ou do CN, utilizando o caminho 2.

No caso do nó móvel transitar para uma rede servida por outro MAP *router*, no exemplo o MAP2, então o MN deve registar o novo LCoA no novo MAP *router*, obtendo o novo RCoA e registando-se no seu *home agent* HA e no antigo MAP *router*, MAP1.

Depois de concluídas estas operações, os pacotes que tenham sido enviados durante o *handoff*, e enquanto não for estabelecida a nova rota otimizada, seguem o caminho indicado pela linha 3. Após a otimização de rota usando o novo RCoA estar concluída, o caminho a ser usado passará a ser o da linha 4.

Existe uma proposta para utilizar o HMIPv6 nas redes NEMO [Ohnishi03] [Park07], que consiste numa extensão do HMIPv6 para suportar o NEMO Basic Support Protocol, de acordo com a qual o *router* móvel e os nós da rede móvel farão uso do MAP que exista nas redes visitadas.

2.4.7. *Mobile IPv6 RO for Network Mobility*, MIRON

O *Mobile IPv6 Route Optimization for Network Mobility*, MIRON [Bernardos07] [Bernardos04], é uma proposta simples e eficaz para adaptar o protocolo MIPv6 à realidade das rede móveis, de acordo com a qual o *router* móvel executa todas as tarefas de otimização de rotas em vez dos nós da rede móvel.

Nesta proposta, o objetivo principal é facultar a mobilidade aos equipamentos móveis de uma forma transparente. O *router* móvel passa a funcionar como um *proxy* MIPv6 e desempenha o papel principal nesta solução.

O procedimento inicia-se assim que o primeiro pacote não otimizado chegar ao *router* móvel. Considere-se o exemplo da Figura 2.24, em que há um pacote que vai do nó da rede móvel `MNN` para o nó correspondente `CN`, ou vice-versa.

Neste caso, o *router* móvel encaminha o pacote original através do caminho identificado pela linha `1`, iniciando em simultâneo o procedimento de otimização de rotas com o nó correspondente em vez do nó da rede móvel.

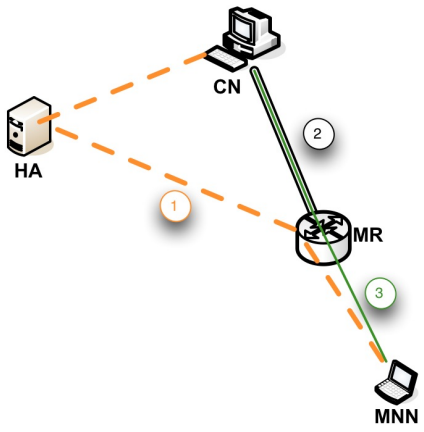


Figura 2.24 – Funcionamento do MIRON

Assim que o procedimento de *return routability* seja concluído com sucesso, é estabelecido o túnel RO identificado na figura pela linha `2`. Os pacotes subsequentes passarão a usar o caminho da linha `3`, que utiliza esse túnel.

O comportamento apresentado funciona para os nós da rede móvel que sejam locais (*Local Fixed Nodes*, LFN, ou *Local Mobile Nodes*, LMN), em que o *router* móvel assume as funções de *home agent*.

Contudo, para o caso de um nó móvel visitante (*Visiting Mobile Node*, VMN) o comportamento é diferente. Assim que um nó móvel visitante se associa à rede móvel,

obtém um endereço IP com o prefixo desta rede móvel. Posteriormente, inicia o procedimento de *binding update* que irá ser interceptado pelo *router* móvel MR. O MR vai utilizar o *Protocol for Carrying Authentication for Network Access*, PANA [Jayaraman08], para obter um endereço pertencente à rede visitada, de modo a atribuí-lo ao nó da rede visitante. Deste modo, o nó móvel visitante passa a ter um endereço IP da rede visitada pela rede móvel.

Assim que está na posse do novo endereço, o nó de rede visitante vai realizar o *binding update* com o novo *Care-of Address*. Para que tal seja possível, o *router* móvel tem que utilizar *source address routing* para conseguir encaminhar os pacotes originados pelo nó de rede visitante.

Deste modo, o nó de rede visitante não está sujeito ao problema do *triangular routing* porque está a usar um IP topologicamente correto.

Para o caso das redes imbricadas o comportamento do MIRON é muito similar ao dos nós de rede visitantes, em que os endereços são obtidos da rede visitada pelo *root*-MR e é distribuído por todos os *sub*-MR que venham a aninhar no *root*-NEMO.

Entre os problemas que existem para esta solução encontram-se a dificuldade em decidir quais os fluxos a otimizar e os problemas de escalabilidade do *router* móvel inerentes à necessidade de realizar todo o trabalho associado ao MIPv6 em vez dos nós de rede móvel.

2.5. Requisitos de mobilidade de rede

Nas secções anteriores apresentou-se um leque de possíveis soluções para a mobilidade de redes. Optou-se por também estudar a mobilidade dos equipamentos finais com o objetivo de avaliar a possível existência de uma solução que se enquadrasse na problemática da mobilidade de redes.

Até ao momento, no entanto, nenhuma proposta ou solução conseguiu reunir consenso ou satisfazer as necessidades de mobilidade de redes, que estão cada vez mais sujeitas a um tráfego mais intenso e exigente como, por exemplo, o de *streaming*, comunicações em tempo real ou outro tipo de comunicações não tolerantes a longas latências ou problemas de conectividade.

Por forma a melhor caracterizar a área de investigação, nesta secção identificação-se os requisitos da mobilidade de redes. A primeira subsecção é dedicada à identificação dos requisitos comuns, isto é, dos requisitos aplicáveis a todos os cenários. No sentido de alargar o âmbito, na subsecção seguinte foi feita uma análise de requisitos específicos das indústrias automóvel, aeronáutica e electrónica de consumo.

2.5.1. Requisitos comuns da mobilidade de redes

Thierry Ernst elaborou, no RFC 4886 [Ernst07a], um conjunto de requisitos que devem ser alcançados quando se aborda a temática da mobilidade de redes, numa perspectiva tão geral quanto possível.
O RFC centra-se no facto dos nós da rede móvel, MNN, não poderem modificar o seu endereço IP quando o *router* móvel muda de rede, mesmo apesar destes endereços terem deixado de estar topologicamente corretos. Assim, o objetivo principal deste documento foi fornecer linhas de orientação para que o acesso à rede fosse reposto de forma imediata e com o mínimo impacto.

O primeiro requisito, obviamente, é o da garantida da **transparência na operação de mobilidade de rede**. É importante que a comunicação de todos os nós da rede móvel seja reposta e que os seus endereços IP se mantenham contactáveis de modo a que não haja quebra de sessões. Este processo não deve ser influenciado pelos **mecanismos operacionais** necessários para que a rede móvel obtenha um acesso topologicamente correto.

Para que a interferência nos nós da rede final seja mínimo, é importante que a migração seja efetuada **sem impacto ao nível do desempenho e sem sobressaltos**. A ligação à rede deve ser restabelecida sem que haja consequências originadas pelo esforço introduzido pela sinalização específica da operação de mobilidade de redes. Assim, deve ser minimizado o tempo sem conectividade para que não se traduza em perdas ou atrasos na entrega dos pacotes às aplicações.

Deve-se assumir a **transparência no suporte da mobilidade de redes**, dado que os nós da rede móvel não estão sujeitos a alterações quando a sua rede transita para outra infraestrutura. Deste modo, não faz sentido impor alterações nos equipamentos terminais para que tenham acesso à Internet. Por isso, aconselha-se que as operações de mobilidade sejam efetuadas apenas pelos *routers* móveis.
Este aspeto torna-se mais relevante quando se consideram alguns casos de equipamentos antigos que não tenham qualquer possibilidade de efetuar atualizações ou, ainda, equipamentos que não tenham capacidade de processamento para outras funcionalidades que não aquelas para que foram concebidos, tais como sensores.

Estas operações podem, também, traduzir-se em impacto ao nível de consumo de energia, o que pode ser comprometedor da eficiência do nó final.

É importante que o suporte de mobilidade seja **agnóstico em relação à configuração** das redes móveis. Deve ser contemplado desde o caso mais simples, de apenas um *router* móvel e o seu nó da rede móvel, até casos mais complexos que tenham várias ligações à rede ou que estejam em diversos níveis de imbricação, que no total possam somar vários milhares de nós.
A lista de potenciais configurações não deve ser limitada. Como exemplos referem-se:

- redes móveis de qualquer tamanho, que contemplem várias sub-redes e um número elevado de nós;
- nós que possam mudar o seu ponto de ligação à rede;
- nós móveis de outras redes que possam associar-se à rede móvel;
- redes com múltiplos pontos de ligação à rede;
- redes imbricadas;
- suporte de vários tipos de acesso ao meio.

Naturalmente, deve ser garantida a **escalabilidade**, de modo a suportar um número elevado de redes móveis, independentemente das suas configurações.

As questões de **segurança** e **privacidade** devem ser asseguradas de modo a que a solução que venha a ser proposta não comprometa o acesso à informação ou a usurpação de identidade. Para além disso, devem ser garantidos mecanismos que permitam preservar a privacidade do utilizador final.

Por fim, deve-se garantir o **mínimo impacto na infraestrutura de encaminhamento**, de modo a que uma solução proposta não implique um esforço adicional e, por vezes, excessivo para fornecer a mobilidade a uma rede.

2.5.2. Requisitos específicos da mobilidade de redes

Os requisitos de mobilidade que são analisados nesta secção dizem respeito a necessidades específicas das indústrias de automóveis [Baldessari09], de electrónica de consumo [Ng08] e da aeronáutica [Eddy09].

O leque de funcionalidades que podem ser usadas com a mobilidade de redes na **indústria automóvel**, e que determinam as suas necessidades específicas, são: serviços de notificações – tais como estado do tempo, tráfego ou notícias; aplicações *peer-to-peer* – como, por exemplo, as mensagens instantâneas, VoIP ou transferência de ficheiros entre veículos; transferência de serviços; monitorização de veículos; aplicações de *infotainment*; e, por fim, aplicações de navegação.
Embora possam ocorrer casos de redes móveis imbricadas, estas não estão na lista de prioridades da indústria automóvel, sendo que a maioria dos requisitos se foca apenas na otimização de rotas entre o *router* móvel e a entidade correspondente (*correspondent entity*, CE).

O primeiro requisito é o da **independência da otimização de rota dos fluxos**, sendo imperativa a possibilidade da realização de otimização de rotas apenas para os fluxos que estejam pré-definidos numa determinada política de RO.

O segundo requisito é a garantia da **segurança no procedimento de otimização de rotas**, de acordo com o qual deve ser garantida a identidade do requerente de *binding update*, bem como as operações de atualização do prefixo da rede móvel (*mobile network prefix*, MNP).
Desejavelmente, qualquer mecanismo de encriptação, a ser adotado, deverá ser baseado num que já se encontre implementado, de modo a evitar um esforço adicional na criação de novos mecanismos de encriptação para os diversos componentes da indústria automóvel.

As questões de **privacidade** assumem um papel importante para esta indústria, devendo ser garantida a confidencialidade das operações de *binding* – *Care-of Address*, prefixo da rede móvel, *home address* –, sendo que estas só devem ser do conhecimento da entidade correspondente.

O suporte de ***multihoming*** também é um requisito importante. As soluções de mobilidade de redes devem suportar a possibilidade do *router* móvel estar ligado a múltiplas redes de acesso e, consequentemente, de poder possuir múltiplos *Care-of Addresses*. Para além disso, deve ser contemplada a possibilidade da rede móvel possuir múltiplos prefixos. Por fim, deve-se permitir a possibilidade de se poder utilizar múltiplos *home agents*.

O último requisito específico da indústria automóvel é ter uma **sinalização eficiente**, ou seja, que a sinalização necessária para obter a rota otimizada seja reduzida ao mínimo indispensável para que sejam cumpridos os requisitos apresentados.

Os problemas que afetam a **indústria aeronáutica** são diferentes dos apresentados anteriormente. Os aviões circulam por áreas geográficas muito dispersas, muito vezes entre países distintos ou, até mesmo, entre continentes. Durante um voo, uma aeronave poderá mudar de ponto de acesso a cada 30 a 60 minutos, aproximadamente, adquirindo um novo *Care-of Address* de cada vez que isso acontece.
Um dos cenários relevantes para esta indústria é o dos serviços de tráfego de ar (*air traffic services*, ATS), que podem ser críticos para a segurança das pessoas. Neste cenário, a maioria dos equipamentos utilizados são locais e fixos (*local fixed nodes*, LFN), podendo existir alguns nós locais móveis (*local mobile node*, LMN).
Em relação aos utilizadores finais, considera-se que a maioria serão nós móveis locais (LMN) ou nós visitantes móveis (*visiting mobile nodes*, VMN), não havendo qualquer controlo do tipo de equipamento ou sistema operativo, não estando prevista a possibilidade de impor algum requisito especial. As aplicações existentes para os utilizadores finais são as existentes na Internet.

Os requisitos de **independência da otimização de rota dos fluxos**, **segurança no procedimento de otimização de rotas**, ***multihoming*** e **sinalização eficiente** são similares aos da indústria automóvel.

Acresce a esta lista a **tolerância à latência**, dado que podem ocorrer vários períodos de inacessibilidade ao exterior. Assim, é importante que os mecanismos de otimização de rotas estejam conscientes desta limitação.
Integrada com este requisito, está a necessidade de garantir a **disponibilidade**, sendo importante que seja possível voltar ao mecanismo do NEMO Basic Support Protocol no caso de algum elemento de otimização de rotas se tornar inacessível.

Dada a elevada probabilidade da instabilidade no acesso ao exterior, é importante que seja garantida a **tolerância à perda de pacotes**, sendo que o mecanismo de otimização de rotas não deve gerar uma maior perda de pacotes ou produzir duplicação de pacotes de dados.

A questão da **escalabilidade** assume um papel preponderante, sendo essencial que um *router* móvel deva ser capaz de suportar várias comunicações sem que tenha atrasos significativos. Para além disso, os pontos de acesso terrestres devem conseguir fornecer acesso à rede a centenas ou milhares de aeronaves.

Por fim, a **adaptabilidade** é um requisito importante porque a indústria da aeronáutica tem receio que venha a ter novos requisitos de protocolos de transporte, mecanismos de encriptação ou, até mesmo, novos campos e opções ao nível dos protocolos TCP/IP.

As preocupações da **indústria da electrónica de consumo** concentram-se nos dispositivos móveis, mais concretamente os *smartphones*. Estes podem funcionar como um nó móvel visitante (VMN) ou como um *router* móvel de uma rede de área pessoal (*personal area network*, PAN).
Os componentes electrónicos que facultem o acesso ao exterior podem estar localizados, também, ao nível de um *router* num componente móvel (por exemplo, um carro) ou num componente fixo (por exemplo, em casa).
O tipo de acesso pretendido para esta indústria engloba as aplicações já existentes na Internet.

O primeiro requisito é que os nós fixos da rede (LFN) **não tenham que ser modificados** para que possam beneficiar do acesso à funcionalidade de otimização de rotas. Esta funcionalidade é importante porque os nós finais são dispositivos simples e sem grande capacidade de suportar novas funcionalidades que sejam exigentes em termos de processamento ou consumo de bateria. Assim, qualquer proposta de otimização de rotas deve ser **pouco consumidora de recursos**.

A questão da segurança é similar à já apresentada para as duas indústrias anteriores.

Por fim, a **harmonia no protocolo** deve ser garantida, de modo a que não haja problemas na utilização dos protocolos existentes atualmente.

A tabela seguinte sumaria os requisitos específicos das diversas indústrias abordadas.

	Automóvel	Aeronáutica	Electrónica de Consumo
Independência fluxos	✔	✔	
Segurança	✔	✔	✔
Privacidade	✔		
Multihoming	✔	✔	
Sinalização eficiente	✔	✔	
Tolerância à latência		✔	
Disponibilidade		✔	
Perda de pacotes		✔	
Escalabilidade		✔	
Adaptabilidade		✔	
LFN não modificados			✔
Consumo de recursos			✔
Harmonia no protocolo			✔

Tabela 2.2 – Comparação dos requisitos para a mobilidade de redes por parte das indústrias

2.6. Conclusão

Ao longo deste capítulo foram brevemente apresentadas várias soluções de mobilidade de redes operando em diferentes camadas protocolares, foi analisado o problema da otimização de rotas, e identificados os requisitos chave da mobilidade de redes. Como conclusão deste capítulo pretende-se agora propor uma classificação das diversas soluções, em função do paradigma no qual assentam. Essa classificação é importante pois permite uma visão clara das características arquiteturais chave de cada solução.

É possível enquadrar as propostas e soluções de mobilidade de redes em três paradigmas distintos: centrado nos nós antigos, baseado na rede e baseado no cliente.

2.6.1. Paradigma centrado nos equipamentos antigos

A ideia subjacente ao paradigma centrado nos equipamentos antigos, *Legacy-compatible (LC) network mobility paradigm*, consiste no fornecimento imediato do acesso à rede sem que os nós da rede móvel e os nós correspondentes tenham que estar sujeitos a alterações. Este

paradigma vai de encontro ao especificado por [Ernst07a], nomeadamente no que diz respeito à "garantia da transparência na operação de mobilidade de rede".

Os protocolos Mobile IP, sem a componente de otimização de rotas, e NEMO Basic Support Protocol são as referências para este paradigma.

O objetivo deste paradigma é garantir que os pacotes são encaminhados pelos caminhos topologicamente corretos, sendo que qualquer situação de mobilidade forçará a passagem através de túneis encapsulados.

A Figura 2.25 ilustra um exemplo de um paradigma centrado nos equipamentos antigos, para o caso da mobilidade de redes. Quando um pacote é enviado pelo nó correspondente CN com destino ao prefixo da rede móvel, MNP, este é interceptado por um agente estrategicamente colocado. No exemplo, está identificado como sendo o *home agent* HA.

Este agente apenas procederá desta forma no caso da rede móvel estar fora da sua rede origem. Naturalmente, este processo só é efetuado se tiver ocorrido o registo de mobilidade neste *home agent*.

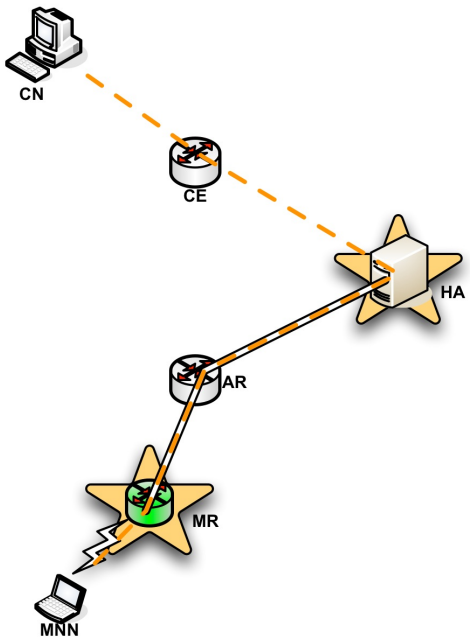


Figura 2.25 – Paradigma centrado nos nós antigos

O agente necessita de guardar a informação relativa a todos os prefixos que está a monitorizar e a interceptar. De igual modo, necessita de conhecer a nova localização dos *routers* móveis.

Quando um pacote é interceptado pelo *home agent*, este procede à sua encapsulação e encaminha-o, através do túnel MRHA, para o endereço topologicamente correto da presente localização do *router* móvel.

Quando o *router* móvel recebe o pacote, procede à sua extração e encaminha o conteúdo original para a rede até atingir o destino final.

As respostas do nó da rede móvel MNN seguem o caminho inverso, sendo os pacotes encapsulados pelo *router* móvel, enviados dentro do túnel MRHA e extraídos pelo agente HA, para depois circularem na Internet até ao nó correspondente CN.

Embora extremamente simples, e totalmente compatível com os equipamentos antigos, estas soluções sofrem de diversos problemas importantes que comprometem a sua implementação em larga escala, tais como o *triangular routing*, estrangulamento na rede origem e a amplificação da falta de otimização em redes imbricadas, tal como discutido anteriormente.

No fundo, todos os problemas que afetam este paradigma estão relacionados com a falta da criação de um caminho melhor para a comunicação entre os nós finais.

2.6.2. Paradigma baseado na rede

No caso do paradigma baseado na rede, as operações relacionadas com a mobilidade são realizadas pelos elementos da infraestrutura de rede – tais como, *routers* móveis, *home agents* ou entidades correspondentes.

A principal ideia deste paradigma está em conseguir que os nós finais possam beneficiar de uma solução de otimização de rotas, sem necessidade modificações. Contudo, a fatura a pagar para obter este benefício consiste no incremento substancial da complexidade e esforço introduzidos na infraestrutura.

Com este paradigma é possível obter o melhor do mundo da mobilidade, sem que para isso se esteja sujeito aos problemas do paradigma centrado no equipamento antigo. Outra diferença em relação ao paradigma anterior é que neste se assume a possibilidade de se realizarem alterações ao nível do nó correspondente, passando a suportar a otimização de rotas.

Os mecanismos de otimização de rotas, contemplam sempre a necessidade de uma entidade externa ao nó móvel, responsável pela realização dos procedimentos de RO em vez destes.

Alguns exemplos de soluções deste tipo para a mobilidade IP e para a mobilidade de redes são: ORC, ND-Proxy, *Global* HAHA, *Hierarchical MIPv6*, MIRON, *Proxy MIPv6* [Gundavelli08] [Soto10], e ainda o caso do *network-based Distributed Mobility Managment* [Patil11].

A otimização de rotas para o caso da mobilidade de redes depende do tipo de nó da rede móvel. O caso mais simples é o do nó fixo local, LFN, que é similar ao do nó móvel local, LMN, para o caso de ainda estar na rede móvel original.

O exemplo da Figura 2.26 ilustra o caso geral de mobilidade de redes utilizando paradigma baseado na rede. Pode-se verificar que todo o tráfego originado por um nó de rede móvel MNN – LFN ou LMN –, deve ser otimizado por um elemento de rede, que no caso é o *router* móvel MR.

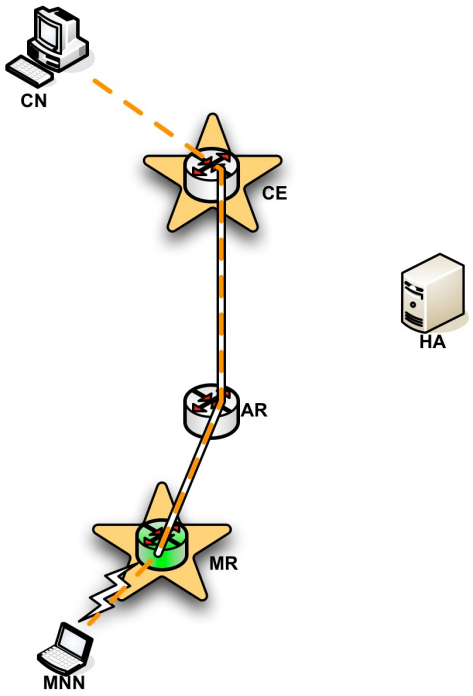


Figura 2.26 – Paradigma baseado na rede

Pode-se verificar que a comunicação está otimizada entre o MR e a entidade correspondente CE. Esta entidade pode ser um *router* no caminho ou, no limite, pode ser o nó correspondente.

Isto significa que o *router* móvel deve manter o registo de todas as ligações entre os nós da rede móvel e as respectivas entidades correspondentes, bem como proceder à atualização de informação de otimização de rotas para todas as ligações sempre que a rede se mover.

Para o caso dos nós móveis visitantes, VMN, o procedimento tem que ser outro sob pena de não viabilizar o procedimento de otimização de rota (qual a *Care-of Address* a usar? O do VMN enquanto está na rede móvel ou o do MR?). Para estes casos opta-se por atribuir endereços topologicamente corretos, pertencentes às redes visitadas pelos *routers* móveis, em que se torna necessário garantir o encaminhamento interno.

Para os casos das redes imbricadas é dado um tratamento similar ao dos VMN.

2.6.3. Paradigma baseado no cliente

O *Internet Engineering Task Force*, IETF, está consciente das limitações introduzidas por um paradigma baseado na infraestrutura pelo que, atualmente, também considera algumas soluções de mobilidade baseadas no cliente [Chan11] [Bernardos11].

No caso do paradigma baseado no cliente, os nós finais estão conscientes da sua condição de mobilidade. Isto significa que os nós finais podem tomar parte ativa nas tarefas de gestão de mobilidade como, por exemplo, na otimização de rotas, aliviando a infraestrutura.

A Figura 2.27 ilustra um exemplo em que o nó da rede móvel MNN toma a iniciativa de otimizar a rota com a entidade correspondente CE, que também poderia ser o nó correspondente CN.

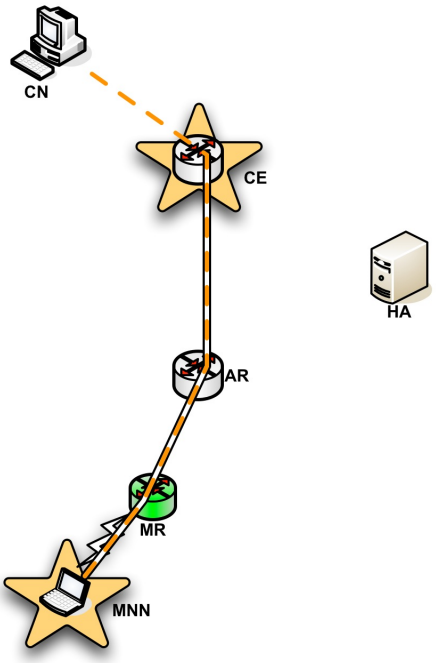


Figura 2.27 – Paradigma baseado no cliente

A partir deste instante, os nós móveis podem decidir quando otimizar as rotas e, quando necessitam, realizar os procedimentos de otimização. Por outro lado, os *routers* móveis passam a restringir-se exclusivamente à tarefa de encaminhamento de pacotes.

As propostas do IETF vão no sentido de utilizar o *Dynamic Mobility Managment protocol* para alcançar este objetivo. A solução apresentada nesta tese, *Optimised Mobility for Enhanced*

Networking, OMEN, também usa este paradigma, conforme se vai explicar no próximo capítulo.

3. *Optimised Mobility for Enhanced Networking, OMEN*

A mobilidade de redes informáticas apresenta desafios técnicos que não são triviais, conforme foi possível constatar no capítulo anterior. O RFC 3963, NEMO Basic Support Protocol [Devarapalli05], foi apresentado como uma solução de mobilidade de redes transparente e com aplicação imediata [Ernst07a].

Os seus aspetos positivos, a simplicidade e a falta de requisitos nos dispositivos finais, estão na base dos seus maiores problemas, conforme se pode constatar nos trabalhos [Ng07], [Ng07a] e [Bernardos05a]. Alguns exemplos que mais se destacam são: *triangular routing*, estrangulamento na rede origem, amplificação da falta de otimização em redes imbricadas. A estes problemas acrescentam os efeitos colaterais, já discutidos anteriormente.

É, então, imperativo olhar para o paradigma atual de forma crítica. Numa altura em que a proliferação de equipamentos móveis é uma realidade incontornável, torna-se fundamental tornar os dispositivos e as redes conscientes da sua condição de mobilidade. Não o fazer é "esconder a cabeça debaixo da areia" ou, o que é pior, forçar a Internet atual a operar e a comportar-se da mesma maneira que a Internet dos anos 70.

Não obstante, é crucial manter a Internet simples e os seus protocolos tão inalterados quanto possível. Contudo, os equipamentos deverão ser objeto de alterações de modo a fornecer-lhes a "inteligência" suficiente para que tenham conhecimento da sua condição de mobilidade e reagir em conformidade.

Alinhado com esta nova perspetiva do problema, o *Optimised Mobility for Enhanced Networking* (OMEN) [Vapi08] [Vapi09] [Vapi10] [Vapi11a] [Vapi11b] [Vapi12] pretende apresentar-se como uma proposta de mobilidade de redes que está baseada no pressuposto de que os equipamentos e as redes passam a estar conscientes da sua condição de mobilidade. A mudança de paradigma permite a criação de uma solução de mobilidade que é, simultaneamente, simples, eficiente e efetiva.

Todo o trabalho é focado exclusivamente em IPv6 por se considerar que uma solução de mobilidade de larga escala apenas faz sentido se for aplicada neste novo protocolo, que já incorpora suporte de diversas funcionalidades importantes de forma nativa, para além de

aumentar o número de endereços IP disponíveis, o que é indispensável para que a proliferação de equipamentos móveis seja uma realidade.

Este capítulo começa com uma descrição geral do OMEN, seguida de uma apresentação do seu funcionamento. Subsequentemente, a proposta é detalhada, do ponto de vista dos diversos tipos de equipamentos que podem usufruir da mobilidade. A descrição do OMEN é complementada por uma especificação semiformal. Na parte final do capítulo inclui-se uma secção sobre validação e outra sobre aspetos de segurança. A conclusão sumariza a proposta e estabelece a ponte para o trabalho a apresentar nos capítulos seguintes.

3.1. Descrição geral do OMEN

O ponto central da proposta OMEN está assente na necessidade de mudança do paradigma de controlo de mobilidade. É interessante notar que, alguns anos após a primeira publicação do OMEN, o IETF passou a orientar as suas atenções no sentido de mudar o paradigma da mobilidade [Chan11] [Bernardos11], devido às mesmas razões.

Partindo do pressuposto de que os nós da rede móvel devem estar conscientes da mobilidade, o OMEN propõe que seja utilizado o *router* móvel como sistema responsável pela anúncio do estado de mobilidade dos nós. Assim, todos os elementos da rede são informados da localização topologicamente correta da rede móvel em que se encontram, podendo, subsequentemente, realizar as operações de otimização de rotas, se assim o entenderem, usando o *Care-of Address* do *router* móvel.

Ao invés de usar um novo protocolo para passar esta mensagem, optou-se por utilizar o *Neighbor Discovery Protocol*, RFC 4861 [Narten07].

O comportamento perante esta informação vai depender do tipo de nó da rede móvel (nó local fixo antigo, nó local fixo, nó local móvel, nó móvel visitante ou outro NEMO) e da necessidade das suas aplicações beneficiarem da otimização de rotas.

Esta secção vai analisar estes aspetos em pormenor, a começar pelas motivações para a mudança do paradigma.

3.1.1. Motivação para a mudança de paradigma

A mudança para o paradigma baseado no cliente vem resolver alguns dos problemas existentes.

O paradigma baseado no cliente antigo possui diversos problemas que já foram amplamente discutidos, enquanto o paradigma baseado na rede tem, pelo menos, dois problemas, que se relembram seguidamente.

O primeiro é a incapacidade de determinar a altura ideal para realizar o procedimento de otimização de rotas, já que é aceite que nem todo o tráfego necessita ser otimizado. De facto, existem situações em que a otimização de rotas pode ser contraproducente, principalmente quando o processo demorar mais tempo que o próprio tráfego, como é o caso do tráfego de DNS.

O segundo problema deste paradigma é o impacto no desempenho dos *routers* móveis, já que estes têm de efetuar todas as operações de otimização de rotas em vez dos nós da rede móvel. Para pequenas redes estas soluções podem ser exequíveis mas, se aplicadas em larga escala, a carga pode ser excessiva para os *routers* móveis.

Naturalmente que a capacidade de processamento dos *routers* tem incrementado ao longo dos tempos, mas não se deve descurar o facto de que quanto mais leve for o *router* móvel menores serão os seus requisitos ao nível de *hardware* e, por conseguinte, menores os requisitos energéticos.

Considerando que estamos a falar em implementações em ambientes de total mobilidade, estas questões assumem um papel preponderante.

O paradigma baseado no cliente final vem resolver estas questões. O cliente, consciente da sua condição de mobilidade, toma a decisão de quando realizar, ou não, a otimização de rotas. Deste modo, não está sujeito a uma política global de realização indiferenciada de otimização de rotas.

Por outro lado, os *routers* móveis passam a dedicar-se quase em exclusivo às tarefas de comutação e encaminhamento, tornando-se assim extremamente leves e com menores necessidades de processamento.

Com estas alterações, o cliente final é informado de que está numa rede móvel, sendo-lhe fornecida toda a informação necessária para a otimização de rotas. Os mecanismos já existentes no MIPv6 vão assumir um papel fundamental para estas operações. Outro aspeto positivo é o facto da solução ser complementar, não impedindo que os equipamentos antigos possam continuar a beneficiar de outras soluções já existentes.

O preço a pagar por esta aproximação é baixo. Os nós locais móveis e os nós locais fixos necessitam de ser atualizados, de modo a poderem usufruir das vantagens do OMEN e, consequentemente, da mobilidade de redes. Os nós visitantes móveis e os *routers* móveis que imbriquem na rede móvel também necessitam de atualização.

Não obstante a necessidade destas alterações, os protocolos – o ponto mais crítico – não precisam ser modificados, já que o OMEN apenas faz uso das funcionalidades já existentes.

Deste modo, torna-se importante analisar quais são as funcionalidades que podem ser úteis para a solução apresentada.

3.1.2. Funcionalidades disponíveis

O OMEN faz uso das funcionalidades de mobilidade de dispositivos e de redes disponíveis nos protocolos *Mobile IPv6*, RFC 6275 [Perkins11] e NEMO Basic Support Protocol, RFC 3963 [Devarapalli05]. O protocolo *Neighbor Discovery*, RFC 4861 [Narten07] também é utilizado, sendo a peça fundamental para o fornecimento de informação relativa à rede móvel. Decidiu-se utilizar este protocolo por já contemplar nativamente o suporte de mobilidade.

Nesta subseção são abordadas algumas funcionalidades destes protocolos que são importantes para o funcionamento do OMEN.

3.1.2.1 *Home address option*

A secção 6.3 do RFC 6275 apresenta o *Home Address Option*, sendo explicado o seu formato e a forma de utilização desta opção do cabeçalho. Este campo vai ser útil para que o OMEN possa transportar a informação sobre o endereço original do nó móvel. Se, por exemplo, houver perda de conectividade com o exterior (*handoff*), então este campo vai fornecer a informação necessária para o envio do pacote ICMP *unreachable*.

Segundo o RFC, esta informação é incluída no pacote enviado pelo nó móvel enquanto está fora da rede origem. O objetivo é o de informar o recipiente de qual é o endereço IP original do nó móvel.

O formato desta opção está ilustrado na Figura 3.1. Eta opção deve ser inserida dentro do campo *Destination Option extension header*, utilizando o *next header* com valor 60.

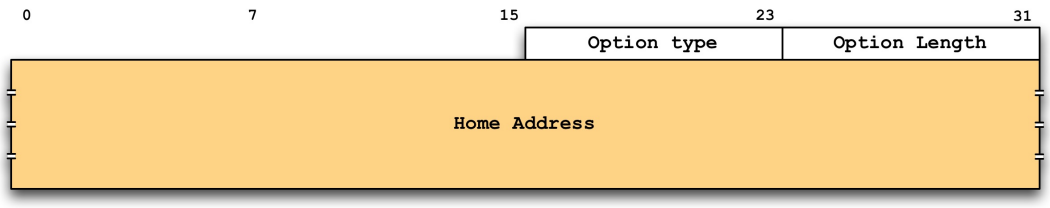


Figura 3.1 – Formato da opção *Home Address Option*

Qualquer nó IPv6 que não reconheça esta opção deverá descartar o pacote com o erro *ICMP Parameter Problem, code 2*, com destino ao endereço *source* IP. No caso do destino do pacote original ser *multicast*, então não deve ser retornado nenhum erro.

O valor do campo *home address option* apenas é aplicável ao pacote que o contém, não devendo ser criado nenhum estado ou modificada a forma de recepção dos pacotes subsequentes.

É com base nesta informação que o *router* móvel OMEN está em condições de entrar em contacto com o emissor do pacote, processo esse que será detalhado na secção 3.4.

3.1.2.2 *Type 2 Routing Header*, T2RH

Na secção 6.4 do RFC 6275 é explicada uma nova variante do cabeçalho de *routing*, criada pelo MIPv6. Este campo permite que um pacote enviado por um nó correspondente para o *Care-of Address* de um nó móvel contenha a informação do endereço destino original – o *home address* do nó móvel. Este campo funciona como *next hop*, indicando o próximo passo a seguir quando for atingido o *Care-of address*.

A utilização deste cabeçalho é feita quando o nó correspondente constrói o pacote com o *Care-of address* do nó móvel no campo *destination* IP, e o *home address* do mesmo nó móvel no campo *Type 2 Routing Header*. É este campo que permite que o router móvel possa determinar o nó ao qual o pacote se destina, encaminhando-o para este. Quando o pacote é recebido pelo nó móvel, este extrai o seu *home address* e utiliza-o para entregar o pacote às camadas superiores.

A Figura 3.2 ilustra o formato do *Type 2 Routing Header*. É usado um tipo de cabeçalho diferente do definido por defeito para o encaminhamento IP, permitindo, desta forma, que as *firewalls* possam aplicar filtros a pacotes que não usem o *Mobile IPv6*.

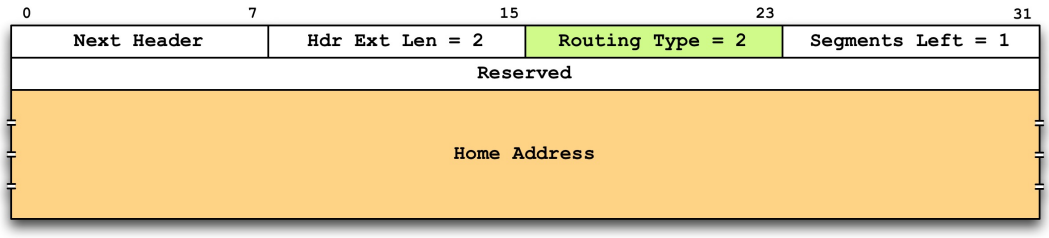


Figura 3.2 – Formato do *Type 2 Routing Header*

O protocolo define que este novo campo apenas pode levar um endereço IPv6.

Por motivos de segurança, todos os nós móveis que processem o cabeçalho de *routing* devem confirmar se o endereço contido dentro deste cabeçalho é o seu.

Conforme já referido, no OMEN os nós utilizam o *Care-of address* do *router* móvel. Esta opção leva a que todos os pacotes otimizados sejam encaminhados para o *Care-of address* do *router* móvel. O campo *Type 2 Routing Header* vai ser a peça fundamental para que o *router* móvel saiba a quem entregar um pacote otimizado.

Com base neste novo cabeçalho torna-se possível a um *router* móvel com suporte de OMEN determinar qual o destino final dos pacotes destinados ao seu *Care-of Address*. Este campo também assume um papel muito importante para a comunicação entre dois nós móveis que usem o OMEN, conforme detalhado na secção 3.4.

3.1.2.3 Neighbor Discovery

O *Neighbor Discovery for IP version 6*, RFC 2461, já contempla funcionalidades específicas para a mobilidade IP. A título de exemplo, na terminologia existe o conceito de *proxy* que designa um equipamento que responde ao pedidos de *Neighbor Discovery* por vez de outro. Na descrição desta funcionalidade é usado o exemplo de um nó móvel.

Outro exemplo é o conceito de *proxy advertisement*, que se define como um equipamento que recebe pacotes por vez de um nó móvel.

Na secção 5.1, *Conceptual Data Structures*, define-se claramente que outros protocolos como, por exemplo, o MIPv6, podem adicionar estruturas conceptuais de dados.

> "Note also that other protocols (e.g., Mobile IPv6) might add additional conceptual data structures. An implementation is at liberty to implement such data structures in any way it pleases. For example, an implementation could merge all conceptual data structures into a single routing table."
>
> In RFC 4861, *section 5.1*

Deste modo, encontra-se aberta uma porta para que o OMEN possa usufruir de um mecanismo de difusão de informação relevante para a mobilidade de redes, nomeadamente a difusão de informação entre *router* móvel e nós finais.

3.2. Funcionamento do OMEN

O OMEN está focado numa mobilidade baseada no cliente final, em que o ónus da realização das tarefas de otimização de rotas é transferido para os nós da rede móvel. Deste modo, as

tarefas de otimização de rotas podem ser realizadas pelos próprios nós quando e se o entenderem.

Para que tal seja possível, em primeiro lugar é necessário que os nós da rede móvel estejam na posse das informações necessárias para a execução das tarefas de mobilidade. Da análise das diversas soluções apresentadas, tanto para a mobilidade IP como para a mobilidade de redes, constatou-se que existe um ponto comum na comunicação móvel: o acesso ao novo endereço IP, conhecido como *Care-of Address*, CoA.

É interessante constatar que este novo endereço IP fornece a informação relativa à localização do dispositivo móvel, enquanto que o *home address* fornece a informação relativa à sua identificação.

Assim, o objetivo do OMEN é fazer com que o *router* móvel, `MR`, informe os seus nós da rede móvel que estão inseridos numa rede NEMO, de modo a que, caso pretendam tirar proveito disso, usem o *Care-of Address* do `MR`.

A Figura 3.3 ilustra uma configuração inicial prevista na proposta OMEN. Neste exemplo, a rede móvel, cujo *gateway* é o *router* móvel `MR`, transita para uma nova rede estrangeira.

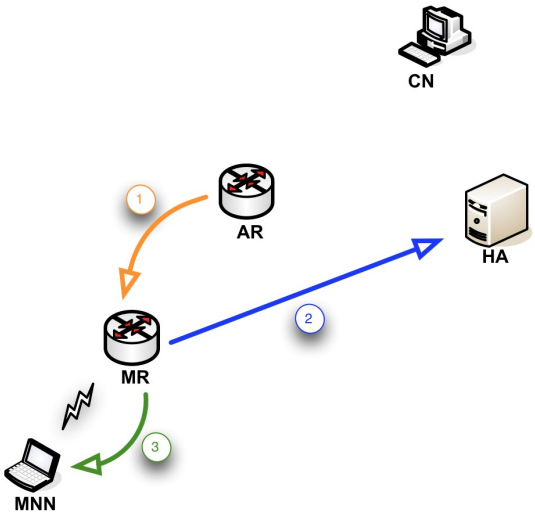


Figura 3.3 – Configuração inicial do OMEN

O *router* de acesso, `AR`, fornece um novo endereço IP topologicamente correto (passo `1` da figura). Este passo pode ser realizado através do *IPv6 stateless address autoconfiguration* [Thomson07] ou do *Dynamic Host Configuration Protocol for IPv6*, DHCPv6 [Droms03].

Na posse do seu novo endereço IP, doravante conhecido como *Care-of Address* – `CoA`, o *router* móvel informa o seu *home agent* da sua nova localização (passo 2). Assim que o processo de *binding* é concluído com sucesso, então passa a existir um canal de comunicação

para a nova localização da rede móvel através do túnel MRHA. Até ao momento, este processo é em tudo igual ao especificado pelo NEMO Basic Support Protocol.

O passo 3 da figura mostra a primeira diferença do OMEN em relação ao RFC 3963. Neste passo o *router* móvel MR comunica aos nós da sua rede móvel o seu *Care-of Address*, informando-os, consequentemente, de que estão numa rede móvel.

Em vez de criar um novo protocolo para passar esta informação para os nós da rede móvel, o OMEN utiliza o *Neighbor Discovery*, ND [Narten07]. Deste modo, em resposta a uma mensagem de *Router Solicitation* ou por sua própria iniciativa, o *router* móvel pode enviar uma mensagem de *Router Advertisement* com o seu *Care-of Address*.

Para a otimização de rotas, os nós da rede móvel deverão utilizar o *Care-of address* do *router* móvel. Na resposta, o *router* móvel deve conferir o *home address* que consta no campo *Type 2 Routing Header*, verificando se pertence ao prefixo da rede móvel ou se se encontra registado na sua tabela de *routing*. O encaminhamento para o nó da rede móvel final é feito com recurso a esta informação.

A Figura 3.4 ilustra um exemplo da comunicação entre um nó correspondente e um nó da rede móvel.

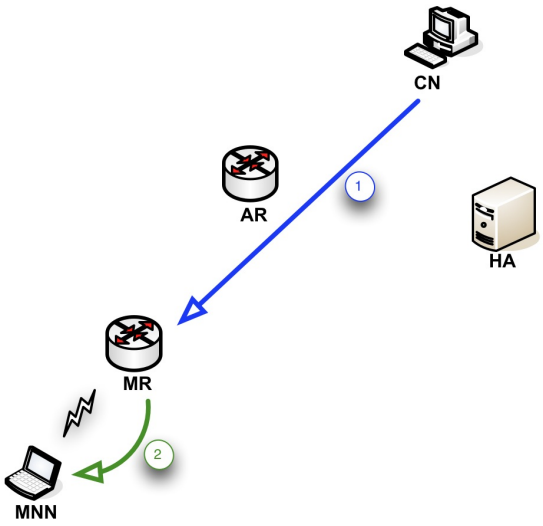


Figura 3.4 – Comunicação entre o CN e o MNN

O CN começa por enviar um pacote com destino ao *Care-of Address* do *router* móvel (passo 1). O MR verifica que o endereço que consta no *Type 2 Routing Header* é o do MNN, pelo que no passo 2 o MR encaminha o pacote para o nó da rede móvel respectivo.

Contudo, para que possa ser possível utilizar a comunicação otimizada, é necessário verificar se o procedimento *Return Routability* consegue ser executado com sucesso. Este

procedimento, definido no protocolo *Mobile IPv6* [Perkins11], pode ser executado dado que as rotas entre o nó correspondente e os endereços IP *Care-of Address* e *Home Address* do nó da rede móvel estão disponíveis através da comunicação direta e através do túnel MRHA, respetivamente.

Um pacote destinado ao *care-of address* atravessa o caminho topologicamente correto até atingir o *router* móvel e, a partir daí, chega ao nó da rede móvel através da informação constante no campo T2RH. Já o caminho até ao *home address* do MNN segue até à rede original do NEMO sendo interceptado pelo *home agent* do *router* móvel. Este agente encapsula o pacote e envia-o através do túnel MRHA com destino ao MR. O *router* móvel procede à extração do pacote original e encaminha-o para o MNN.

Está, assim, garantida a comunicação entre o nó da rede móvel e o nó correspondente.

É de salientar que o nó correspondente CN não consegue distinguir se está a comunicar com um dispositivo pertencente a uma rede móvel ou se está a comunicar com um simples equipamento com suporte de MIPv6.

Sempre que a rede móvel se movimenta, os nós locais fixos ou móveis, LFN ou LMN, não necessitam de proceder a qualquer notificação do *home agent*. Este processo é realizado pelo *router* móvel MR e beneficia diretamente os seus nós da rede móvel. Deste modo, sabendo que apenas após o *binding* com o seu *home agent* é que o *router* móvel informa os seus nós da rede móvel, torna-se óbvio que estes apenas necessitam de atualizar a informação de *binding* com os seus nós correspondentes.

Nos casos de nós móveis visitantes ou de redes móveis imbricadas, como ilustrado na Figura 3.5, a solução funciona de uma forma similar ao explicado anteriormente.

No exemplo da figura, o *router* móvel MR2 juntou-se à rede do MR1. Os passos 1, 2 e 3 são iguais ao caso de um nó local fixo ou móvel.

O passo 4 processa-se após o MR2 receber a informação de que está imbricado numa rede móvel e que o *router* de topo, MR1, tem o *Care-of Address* CoA.

O MR2 envia um *Binding Update* com o novo *Care-of Address* para o seu *home agent*, HA2. Após o registo concluído com sucesso, o MR2 regista o seu prefixo da rede móvel no seu *router* móvel, sub-MR. Este registo é necessário para que o sub-*router* possa tomar conhecimento do prefixo da rede móvel imbricada.

Apenas após este passo é que o MR2 pode informar os seus nós da rede móvel sobre o novo *Care-of Address* (passo 5).

Naturalmente, o nó móvel visitante não regista o prefixo da rede móvel. Contudo, tem que registar o seu *home address* para que o *router* móvel saiba quando encaminhar um pacote destinado a este endereço IP.

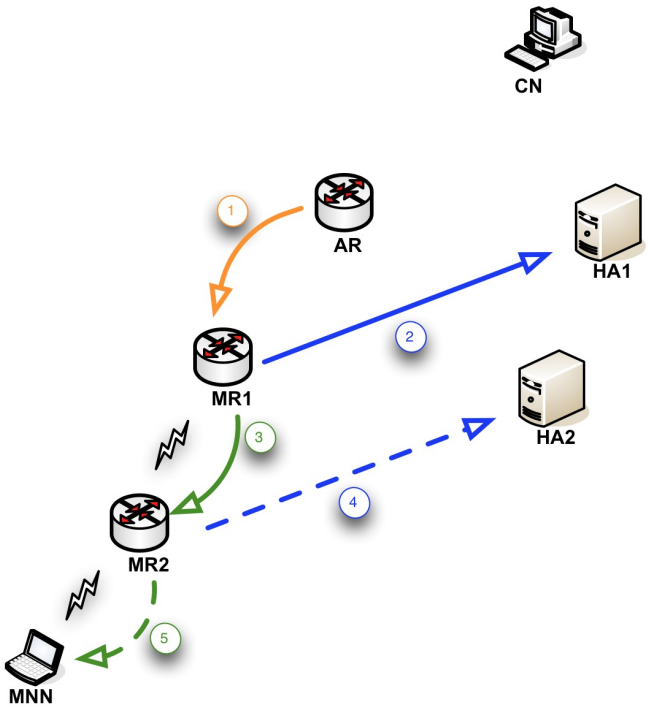


Figura 3.5 – Configuração inicial do OMEN com uma rede imbricada

A comunicação entre um nó correspondente CN e um nó da rede móvel imbricada MNN também se processa de forma similar, conforme se pode ver na Figura 3.6.
O pacote é enviado para o CoA do MR1, que após o receber e confirmar que o *next hop*, incluído no campo *Type 2 Routing Header*, pertence ao prefixo de uma rede imbricada, encaminha-o para o MR2, que o faz chegar ao nó final, MNN. No caso do *next hop* ser o *home address* de um nó móvel visitante, então o *router* móvel encaminha-o para o VMN.

No OMEN, são os nós da rede móvel quem decide quando otimizar as rotas. No caso de necessitarem, podem realizar esta operação por si próprios. Desta forma, os *router* móveis não são afetados por esta tarefa. Não têm que guardar estados, informação sobre fluxos, ou qualquer outro registo sobre as rotas otimizadas pelos seus nós da rede móvel.
Por outro lado, dado que os nós da rede móvel utilizam o *Care-of Address* do *router* de topo da rede móvel, não há necessidade de requerer um novo endereço IP para cada nó visitante móvel ou para cada rede imbricada, como no caso de algumas soluções apresentadas no âmbito do paradigma baseado na rede.

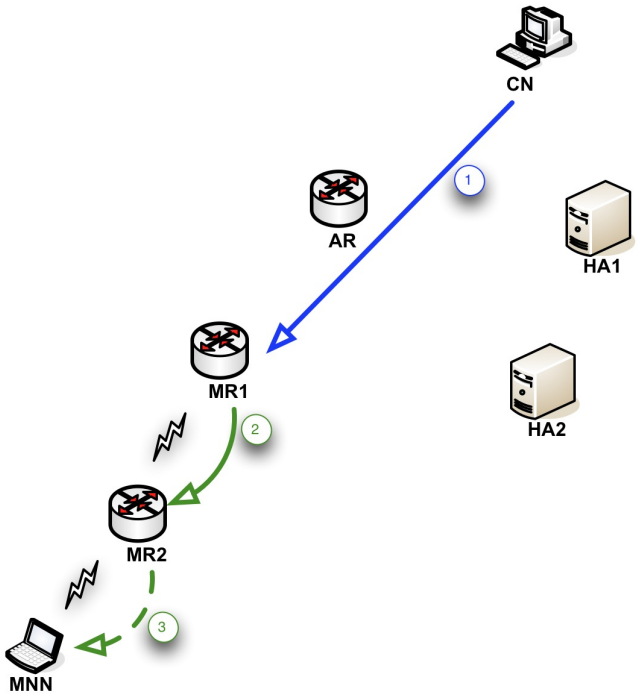


Figura 3.6 – Comunicação entre o CN e o MNN

Quando o pacote chega ao *router* de topo, ou a qualquer *sub-router*, estes sabem como encaminhar o pacote com base na informação constante no *Type 2 Routing Header*, que indica o *next hop*, sem haver a necessidade de realizar processamentos adicionais ao pacote ou terminar túneis.

Por último, dado que todos os nós otimizam a rota utilizando o mesmo *Care-of Address* do *router* de topo, torna-se possível determinar se um nó correspondente se encontra dentro da mesma rede móvel. Este efeito secundário desejável abre caminho para que a comunicação entre os equipamentos finais se processe dentro da rede móvel, mesmo não havendo conectividade com exterior (*handoff*). Repare-se que, apesar de não haver lugar ao procedimento de *Return Routability*, as questões de segurança estão garantidas dado que a comunicação neste caso será realizada sem otimização de rotas.

Conforme é possível constatar, o OMEN tem um comportamento diferenciado em função do tipo de nó de rede móvel, pelo que a próxima secção vai analisar cada caso em particular.

3.3. Tipos de nós móveis

O OMEN utiliza diferentes aproximações, dependendo do tipo de nó da rede móvel, tal como se encontra ilustrado na Figura 3.7.

Os diferentes casos são: nós locais fixos antigos, conhecidos como `LFN legacy`; nós locais fixos com suporte de MIPv6, `LFN`; e os nós locais móveis, `LMN`. Também são considerados os casos de equipamentos externos à rede móvel como, por exemplo, o nó móvel visitante, `VMN`, ou outra rede móvel, `nested MR`.

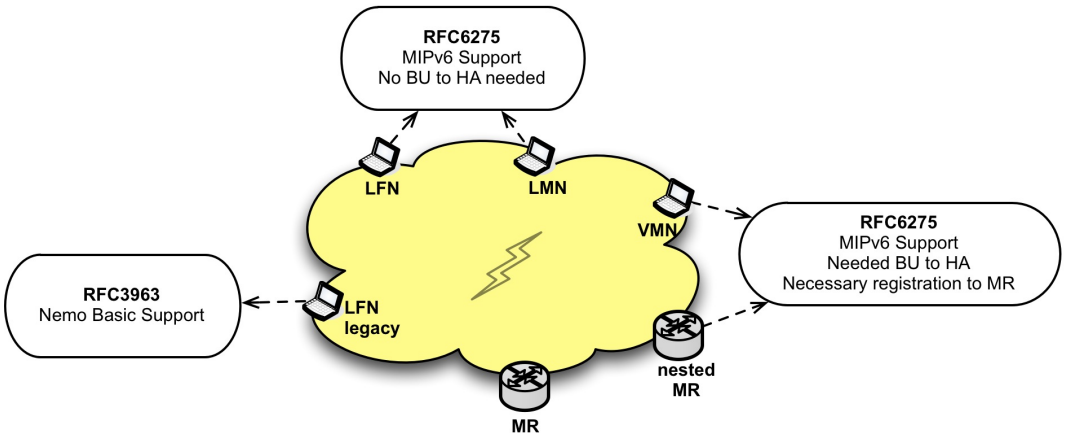


Figura 3.7 – Exemplos de tipos de nós móveis para o OMEN

A figura ilustra as soluções que o OMEN propõe para cada tipo de nó da rede móvel. De seguida é analisada em detalhe cada uma das possibilidades.

3.3.1. LFN *legacy*

Os nós locais fixos antigos, *LFN legacy*, não estão conscientes da condição de mobilidade, pelo que o OMEN não tem em consideração este tipo de equipamentos. Considera-se que nesta categoria se enquadram os equipamentos antigos ou leves, que nunca terão capacidade de implementar o OMEN. Para estes, a solução a adoptar é utilizar o NEMO Basic Support Protocol e, deste modo, ficar sujeito às limitações e problemas inerentes a esta solução.

Dado que o OMEN é complementar relativamente a outras propostas, existe também a possibilidade de se aplicar outra solução de otimização de rotas como, por exemplo, uma solução baseada na rede, e permitir que estes equipamentos antigos estejam sujeitos a este paradigma.

3.3.2. LMN ou LFN com suporte MIPv6

O caso de um nó local móvel, LMN, é em tudo semelhante ao caso de um nó local fixo, LFN, com suporte de MIPv6. Deste modo, opta-se por referi-los a ambos como sendo um nó local móvel, LMN, por motivos de simplificação do texto.

Um equipamento nestas condições deve aceitar como seu o *Care-of Address*, `CoA`, anunciado através do *Router Advertisement* pelo *router* móvel. Utilizando este CoA, o nó local móvel deverá criar rotas otimizadas com os seus nós correspondentes.

O nó local móvel nunca deverá anunciar o *Care-of Address* ao *home agent* da rede móvel, já que esta tarefa é executa pelo *router* móvel. Isto só é possível porque, apesar da rede móvel estar em trânsito, o LMN se encontra dentro da sua rede original (móvel).

No caso do nó local móvel abandonar a sua rede móvel, isto é, a rede NEMO, então deverá passar a seguir o *standard* MIPv6. É importante salientar que quando um nó móvel local sai da sua rede móvel passa a ser considerado um nó móvel MIPv6 ou um nó móvel visitante NEMO. Nestes casos, o LMN deve realizar o *binding* com o seu *home agent* e, também, o procedimento de *Return Routability*, de modo a poder utilizar a rota otimizada com o seu nó correspondente.

3.3.3. VMN

O caso de um nó visitante móvel, VMN, tanto pode dizer respeito a um nó móvel que apenas tem suporte de MIPv6, como pode ser outro nó local móvel, LMN, de outra rede, que transita para a presente rede móvel.

O nó visitante móvel deve aceitar como seu o *Care-of Address*, enviado pelo *router* móvel da rede onde se encontra, da mesma forma que no caso do LMN. O VMN deve registar o CoA junto do seu *home agent*, conforme ilustrado no passo `1` da Figura 3.8.

O registo do *Home Address*, HoA, do nó visitante móvel só pode ocorrer após a conclusão com sucesso do *Binding Update* junto do seu *Home Agent*, por motivos de segurança. É que a melhor forma do VMN garantir que o HoA é seu é recorrendo ao procedimento de *Return Routability* junto do *router* móvel.

Para que tal possa ocorrer, é necessário que a comunicação com o seu *Home Address* esteja garantida através do *Home Agent* do nó visitante móvel.

Este processo está ilustrado, de forma simplificada, no passo `2` da Figura 3.8.

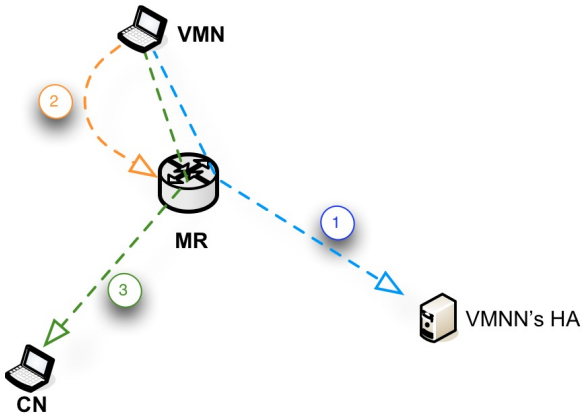


Figura 3.8 – Um VMN a utilizar o OMEN

Após o registo com sucesso no *router* móvel, este último deve criar e gerir uma tabela de encaminhamento que contenha o mapeamento entre o *Home Address* do nó visitante móvel e o respetivo endereço *Link Local* adquirido com o prefixo da rede móvel.

Dado que o campo *Type 2 Routing Header* vai conter sempre o *Home Address* do VMN, o *router* móvel consegue, usando a sua tabela, determinar a que endereço de *Link Local* deve ser enviado o pacote.

O VMN deve executar todos os procedimentos de otimização de rotas de acordo com o RFC 6275 utilizando o novo *Care-of Address*, CoA.

3.3.4. NEMO imbricado

No caso de uma rede NEMO transitar para uma rede móvel, o processo de utilização do OMEN é muito similar ao VMN, explicado anteriormente.

No exemplo da Figura 3.9 o *router* móvel MR2 encontra-se imbricado no MR1.

O primeiro passo (1) deve ser dado pelo MR2, enviando um *Binding Update* ao seu *Home Agent* HA2. Uma vez mais, este passo é necessário por motivos de segurança, conforme explicado anteriormente.

O segundo passo (2) consiste no registo do *Home Address* do *router* móvel MR2 junto do MR1. Neste processo está também incluído o registo do prefixo de rede móvel MNP2.

Com base nestas duas informações, o MR1 saberá sempre como encaminhar pacotes cujo *Home Address* conste no *Type 2 Routing Header* e que digam respeito ao *Home Address* do MR2 ou ao um endereço IP com o prefixo da rede móvel NEMO2.

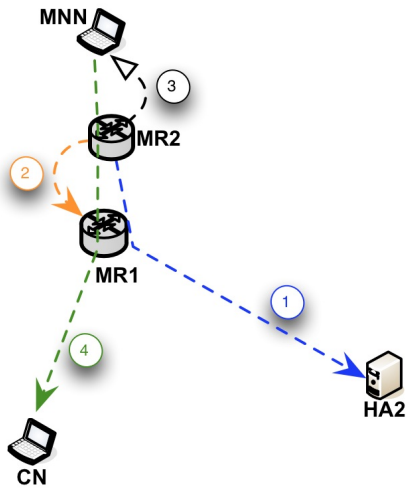


Figura 3.9 – Uma rede imbricada a usar o OMEN

De seguida, o MR2 deve enviar um *Router Advertisement* aos seus MNN a informar do novo *Care-of Address*. Este passo está ilustrado na figura como sendo o passo 3.

Após este passo, o *router* móvel MR2, ou qualquer um dos seus nós da rede móvel, passarão a estar em condições de utilizar os protocolos NEMO Basic Support Protocol ou Mobile IPv6 com recurso ao *Care-of Address* do *router* de topo MR1.

3.4. Especificação semiformal

A descrição do OMEN apresentada anteriormente deve ser complementada com uma especificação semiformal do seu funcionamento. Esta secção vai analisar em pormenor diversos passos críticos inerentes à proposta OMEN.

Por forma a possibilitar uma correta compreensão da especificação semiformal, apresenta-se, no que se segue, uma breve explicação da convenção seguida nas figuras que a compõem.

Para o exemplo da Figura 3.10, a comunicação em causa envolve os seguintes elementos: nó da rede móvel MNN; *router* móvel MR; *router* de acesso AR; o *home agent* do MR, HA; e o nó correspondente CN. Estes elementos aparecem sempre no topo das linhas verticais.

A sequência dos fluxos tem uma relação temporal. Assim, o fluxo mais perto do topo ocorre antes do fluxo logo abaixo dele, e assim por diante.

Por uma questão de conveniência, convencionou-se que o fluxo número 1 é o que está mais perto do topo, e mais perto do lado esquerdo. Os fluxos subsequentes processam-se sempre da esquerda para a direita, e do topo para o fundo.

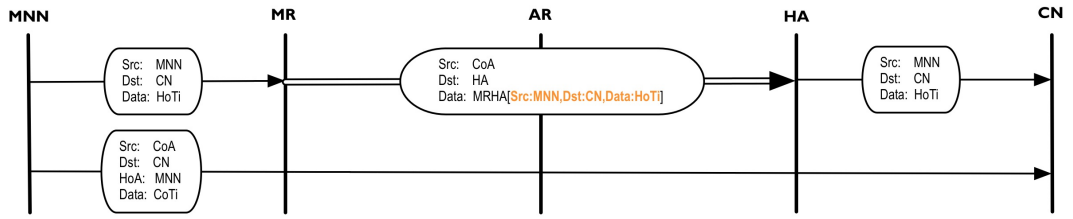


Figura 3.10 – Exemplo de uma especificação

Considerou-se que a colocação dos fluxos num ângulo que denotasse o tempo que o pacote leva a chegar de um equipamento a outro não traria valor acrescentado e traduzir-se-ia num impacto ao nível do tamanho das figuras, optando-se por não o fazer.

Ainda na Figura 3.10, o primeiro fluxo diz respeito à comunicação entre o `MNN` e o `MR`. Após esta comunicação, o segundo fluxo mostra o `MR` a enviar um pacote encapsulado dentro de um túnel MRHA para o `HA`. Como se pode intuir, as linhas duplas indicam que um fluxo é encapsulado.

Neste mesmo fluxo pode-se verificar que dentro do túnel aparece o fluxo original identificado por outra cor, no exemplo a laranja.

Dentro de cada caixa identificativa do fluxo encontra-se a informação que consta no pacote IP em circulação. Optou-se por apenas fazer constar os campos que se consideram mais relevantes para o caso em estudo, de modo a não sobrecarregar as figuras.

No mesmo exemplo, em cada fluxo é visível a origem (`src` = *source IP*), o destinado (`dst` = *destination IP*) e os dados (`data`). No exemplo em causa, a informação que consta dentro do pacote é relativa a um *Home Test init*, `HoTi`.

Por motivos de simplificação da leitura dos endereços IP, optou-se por associar o nome do dispositivo ao endereço IP.

Assim, no caso do exemplo, quando se disser que um pacote vai de `src:MNN` para `dst:CN`, assume-se que o endereço IP do `MNN` é `MNN`, e o do `CN` é `CN`.

Não se considera relevante distinguir o prefixo, dado que para o estudo das redes móveis aquilo que importa é o endereço IPv6 completo, independentemente de qual o conteúdo dos 64 bits mais significativos.

De modo a simplificar a compreensão da proposta OMEN como um todo, tentou-se seccionar a especificação em casos tão elementares quanto possível permitindo, desta forma, a reutilização dos diversos passos.

Por exemplo, a configuração inicial é um passo comum a todos os nós da rede móvel, independentemente do seu tipo. Assim, quando se fala num caso de um tipo de equipamento

em particular, pode-se fazer a chamada à configuração inicial já explicada na secção para o efeito, reduzindo o número de figuras e tornando os desenhos mais leves.

3.4.1. Configuração inicial

Dado que não faz sentido falar em otimização de rotas quando a rede móvel se encontra na sua rede de origem (*home network*), considera-se como configuração inicial aquela que o *router* móvel adquire quando se move para outra rede.

A partir do momento em que se consegue ligar à nova rede, o *router* móvel MR adquire o *Care-of Address* CoA através do *IPv6 stateless address autoconfiguration* ou do *Dynamic Host Configuration Protocol for IPv6*, DHCPv6, identificado pelo primeiro fluxo da Figura 3.11.

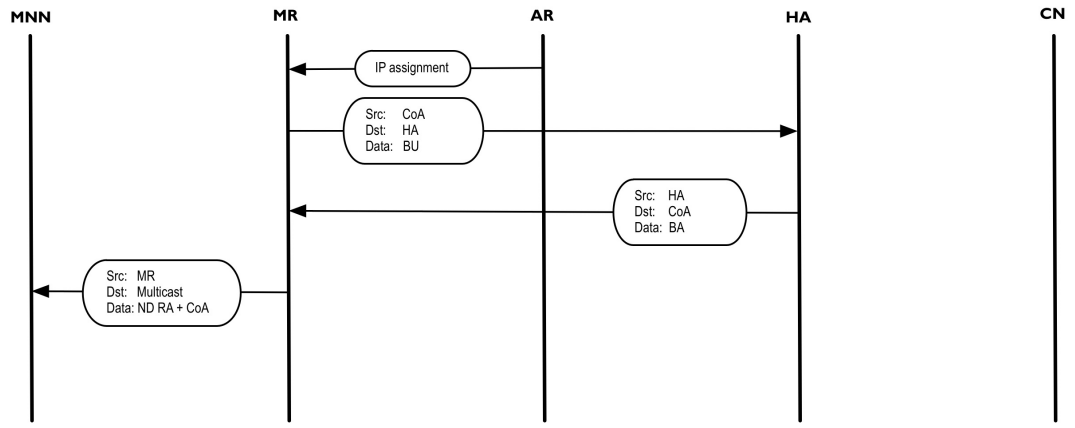


Figura 3.11 – Configuração inicial

Na posse do seu novo endereço, o MR tem que o anunciar ao seu *home agent*, HA.

Os segundo e o terceiro fluxos dizem respeito aos *Binding Update* e *Binding Acknowledgement*, respetivamente. Este processo está definido no NEMO Basic Support Protocol, RFC 3963.

O OMEN inicia-se no fluxo número 4, quando o MR envia um *Router Advertisement* para a sua rede móvel anunciando o novo *care-of address* CoA, de modo a que todos os nós da rede móvel possam usar este novo endereço.

O *Router Advertisement* cumpre os requisitos do *Neighbor Discovery Protocol*, RFC 4861, sendo acrescentado o campo opcional dedicado ao CoA.

Os equipamentos mais antigos podem ignorar este campo opcional, fazendo uso apenas dos campos que conheçam. Os equipamentos que suportem OMEN deverão guardar esta informação para que a possam utilizar no caso de necessitarem otimizar rotas.

Para a realização do procedimento de RO apenas necessitarão do que já se encontra definido no protocolo MIPv6, RFC 6275. O próximo passo, apresentado na próxima secção, é comum para todos os tipos de nós da rede móvel que pretendam otimizar as rotas.

3.4.2. RR MNN → CN

Sempre que um nó da rede móvel, MNN, pretenda proceder à otimização de rotas com o seu nó correspondente, CN, deverá executar o procedimento de *Return Routability*, RR.

Durante este procedimento, as comunicações entre os dois equipamentos finais, MNN e CN, são asseguradas por dois campos essenciais do protocolo MIPv6: *Home Address Option* e *Type 2 Routing Header*, T2RH.

O primeiro encontra-se definido na Figura 3.12 como o campo HoA:, enquanto o segundo aparece como o campo T2RH:. A inclusão destes campos não implica nenhuma alteração em relação ao definido no RFC 6275.

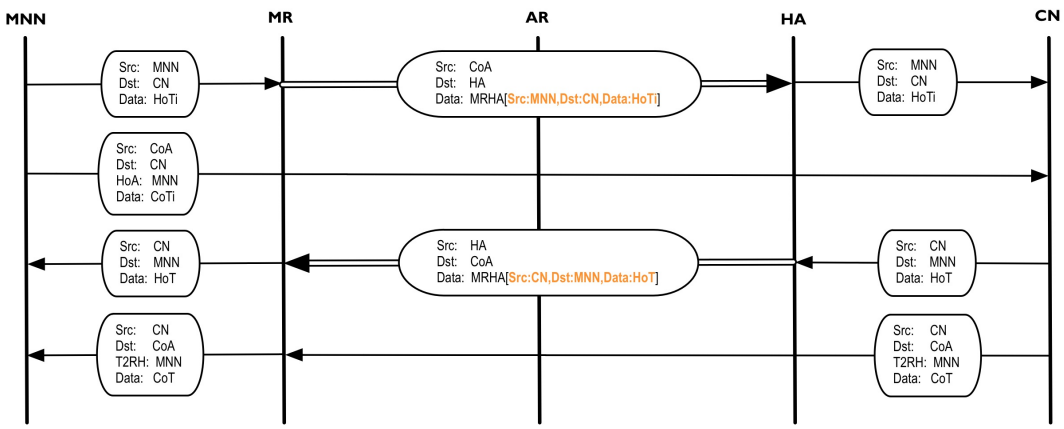


Figura 3.12 – Procedimento *Return Routability* executado por um MNN

De modo a executar o procedimento de *Return Routability*, o nó da rede móvel MNN envia dois pacotes para o seu nó correspondente:

- um pacote que contém o *Home Test init* (HoTi) enviado com o seu endereço original (MNN) e que segue através do túnel MRHA, ou seja, seguindo o NEMO Basic Support Protocol, RFC 3963;
- um pacote contendo o *Care-of Test init* (CoTi) utilizando o *Care-of Address* CoA, enviado diretamente para o CN.

Ambos os pacotes contêm um *token* que identifica o nó da rede móvel MNN. Estes dois pacotes são identificados pelos quatro primeiros fluxos da figura.

Quando o nó correspondente CN recebe os dois pacotes, deve responder a cada um deles enviado de volta um *Home Test* (HoT) através do túnel MRHA e um *Care-of Test* (CoT) diretamente para o CoA do MNN.

Ambos os pacotes contém a resposta ao *token* enviado nos pacotes HoTi e CoTi.

O HoT está identificado pelos fluxos 5, 6 e 7, enquanto que o CoT está identificado pelos fluxos 8 e 9.

O oitavo fluxo está destinado ao *Care-of Address* do *router* móvel MR, pelo que quando o *router* recebe este pacote analisa o conteúdo do campo T2RH: e encaminha-o para o nó da rede móvel MNN. O último fluxo mostra esse passo.

A seguir a este passo, é necessário efetuar o *binding update* com nó correspondente, conforme se pode ver a seguir.

3.4.3. RO BU MNN → CN

O procedimento de *Route Optimization*, RO, contempla o passo de *Binding Update*, BU, que permite concluir o procedimento de *Return Routability*, RR, e que também faculta um mecanismo de passagem de informação do novo *Care-of Address*.

Para os casos dos equipamentos que já tenham a rota otimizada e que, entretanto, tenham adquirido um novo *Care-of Address*, torna-se apenas necessário enviar um *Binding Update*, poupando desta forma muito tráfego.

Assim, seguidamente vai-se ilustrar como se processa o *Binding* sempre que é anunciado um novo *Care-of Address*. Conforme se disse, este passo pode ser repetido sempre que for anunciado um novo CoA, sem necessidade de repetir todo o procedimento de *return routability*, razão pela qual se optou por dedicar uma secção exclusiva a este passo.

A Figura 3.13 ilustra o nó da rede móvel MNN a enviar um pacote de *Binding Update* para o seu nó correspondente CN. O seu endereço IP origem é o CoA, enquanto o campo HoA: vai conter o seu IP original MNN.

Quando o nó correspondente CN recebe o pacote, consegue verificar se o *token* que acompanha o pacote valida o endereço IP que consta no campo HoA:. Em caso afirmativo, o CN envia um *Binding Acknowledgement* destinado ao CoA, contendo no campo T2RH: o enderço IP do MNN.

Uma vez mais, o *router* móvel MR sabe como encaminhar o pacote com base no conteúdo deste campo.

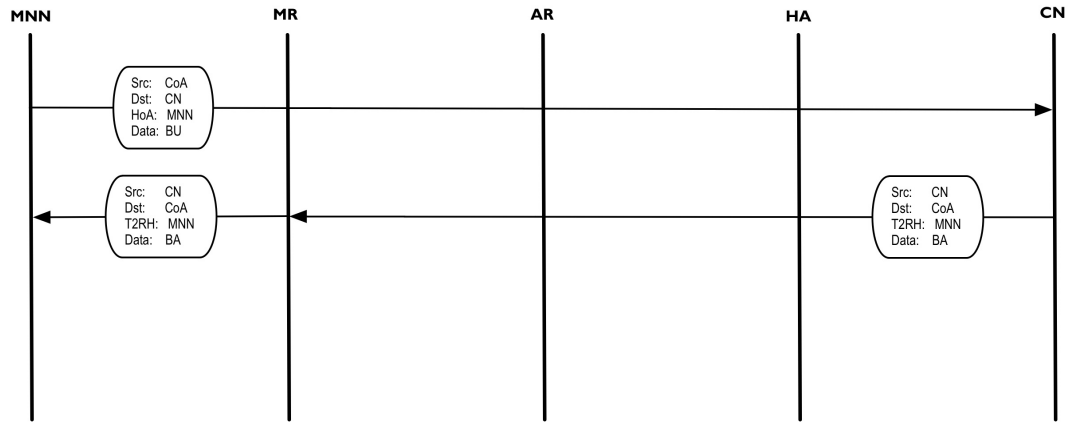


Figura 3.13 – Um MNN envia um *Binding Update* a um CN

O passo a seguir é o processo de comunicação entre o nó da rede móvel e o nó correspondente, quando em situação de otimização de rotas.

3.4.4. Comunicação RO MNN → CN

Após a rota estar otimizada, a comunicação entre os dois equipamentos finais pode ser realizada diretamente utilizando o *Care-of Address* `CoA` como o endereço do `MNN`, desde que devidamente complementado pelos campos `T2RH:` e `HoA:`, de modo a identificar o *next hop* e o endereço origem, respetivamente.

A Figura 3.14 ilustra a comunicação entre o nó da rede móvel `MNN` e o seu nó correspondente `CN`.

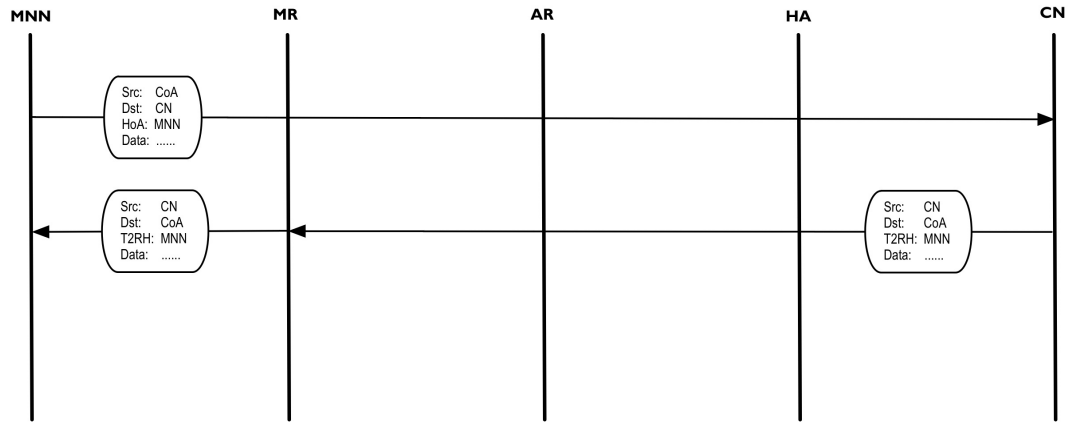


Figura 3.14 – Comunicação otimizada entre o `MNN` e o `CN`

A comunicação entre o MNN e o CN processa-se de forma direta, sem passar por nenhum túnel. Assim que o CN recebe o pacote, analisa o campo HoA: para verificar quem é o verdadeiro emissor. Na resposta, o CN envia o pacote destinado ao CoA, contendo no campo T2RH: o endereço IP MNN. Deste modo, o *router* móvel MR consegue entregar o pacote ao destino correto.

3.4.5. Comunicação entre dois MNN inter-NEMO

Por comunicação inter-NEMO entende-se toda a passagem de tráfego entre duas redes móveis que não estejam imbricadas entre si. Assim, o que é pretendido analisar é a comunicação entre um nó da rede móvel de uma rede NEMO com o nó da rede móvel de outra rede NEMO.

Este tipo de situações tem apresentado alguns desafios técnicos devido à complexidade acrescida. A proposta OMEN consegue dar resposta a este problema desde que sejam cumpridos os requisitos do MIPv6.

A Figura 3.15 ilustra a comunicação entre dois nós de redes móveis, MNN1 e MNN2, servidos pelos *routers* móveis MR1 e MR2, respetivamente.

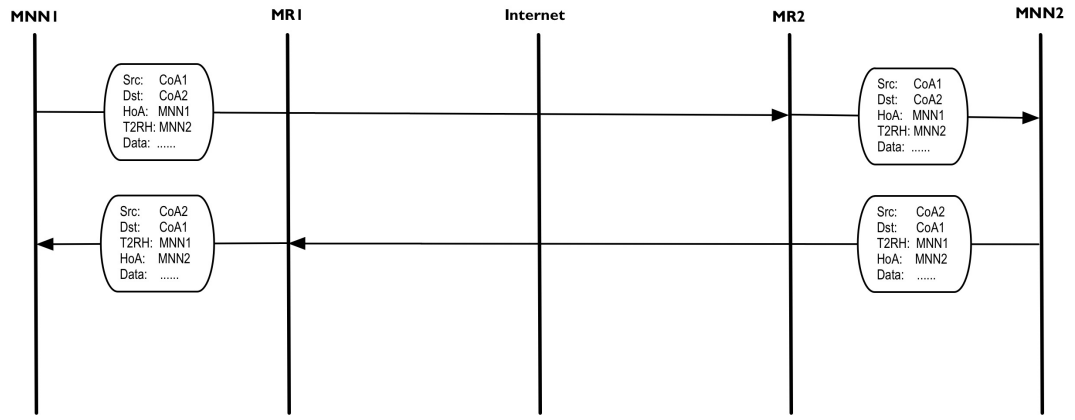


Figura 3.15 – Comunicação entre dois MNN de redes NEMO distintas

Quando o MNN1 envia um pacote para o MNN2 insere no campo do endereço IP origem o CoA do MR1, CoA1, enquanto que no campo do destino vai o CoA do MR2, CoA2. O campo HoA: tem que identificar o seu endereço IP origem, MNN1, enquanto que o campo T2RH: tem que indicar o *next hop*, no caso o IP origem do MNN2.

Quando o MR2 recebe o pacote sabe como reencaminhar para o MNN2 (através do campo T2RH:). Por outro lado, o MNN2 sabe a proveniência do pacote a partir do campo HoA:.

Na resposta, o processo é o inverso do explicado.

3.4.6. LMN move-se para outra rede móvel

No caso de um nó local móvel, LMN, transitar para outra rede móvel, mesmo que não tenha estabelecido nenhuma rota otimizada, passa a comportar-se como um nó móvel visitante para a outra rede. Assim, a forma como se deve proceder será explicada na secção 3.4.8 Registo de um VMN.

3.4.7. MR perde acesso ao exterior (*handoff*)

A comunicação não otimizada entre os nós da rede móvel e os nós correspondentes processa-se sempre de acordo com o RFC 3963. Contudo, a comunicação otimizada no protocolo OMEN processa-se utilizando o *Care-of Address* do *router* móvel de topo.

Esta secção explica qual o comportamento esperado quando é necessário processar algum pacote de erro, como por exemplo um *Internet Control Message Protocol*, ICMPv6 [Conta06].

O ICMP pode ser gerado a qualquer momento da comunicação entre os dois equipamentos finais.

A Figura 3.16 ilustra o `MNN` a enviar um pacote e o `MR` a não conseguir processar, necessitando de gerar um `ICMP`.

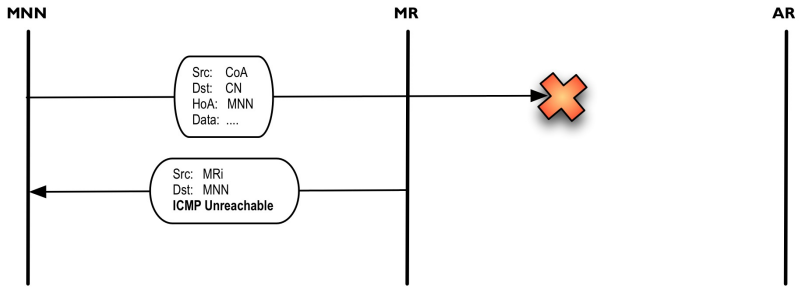


Figura 3.16 – O *router* móvel perde acesso ao exterior (*handoff*)

Neste caso simples, em que o `MR` possui o pacote original, consegue determinar através do campo `HoA:` qual o endereço IP para onde deve encaminhar o pacote ICMP.

Contudo, no caso do erro ocorrer quando o pacote se encontra em trânsito, o ICMP irá ser gerado por outro *router* e encaminhado para o *Care-of Address* do *router* móvel `MR`. Quando o *router* móvel receber este ICMP, deve analisar o seu conteúdo para determinar o

valor do campo HoA:. Segundo o RFC 4443 esta informação deve ser incluída no pacote ICMP.

> Inclusion of, at least, the start of the invoking packet is intended to allow the originator of a packet that has resulted in an ICMPv6 error message to identify the upper-layer protocol and process that sent the packet.
>
> In RFC 4443, section 2.1 Message General Format

3.4.8. Registo de um VMN

Sempre que um equipamento com suporte de mobilidade, quer seja um nó móvel com suporte de MIPv6 ou um nó da rede móvel que transitou de outra rede móvel, se associa a uma rede com suporte de OMEN, deverá proceder ao registo do seu endereço IP *Home Address*. Esta informação é útil para que o *router* móvel possa encaminhar os pacotes dentro da rede móvel.

O registo do endereço *Home Address* por parte do nó móvel visitante torna-se mais complexo devido à necessidade de garantir a identidade do equipamento sem modificar nenhum protocolo existente.

Este processo inicia-se com a aquisição de um endereço IP dentro da rede móvel e subsequente obtenção do *Care-of Address* CoA da rede móvel, A partir deste instante, o nó móvel visitante toma conhecimento da condição de mobilidade da rede, conforme explicado no passo da secção de Configuração inicial.

De seguida, o VMN deve registar o seu endereço IP adquirido na rede móvel junto do seu *home agent*. A Figura 3.17 ilustra este passo.

Assumiu-se que o nó móvel visitante VMN adquire o endereço IP VMN na rede móvel e que o seu IP original é HoAVMN.

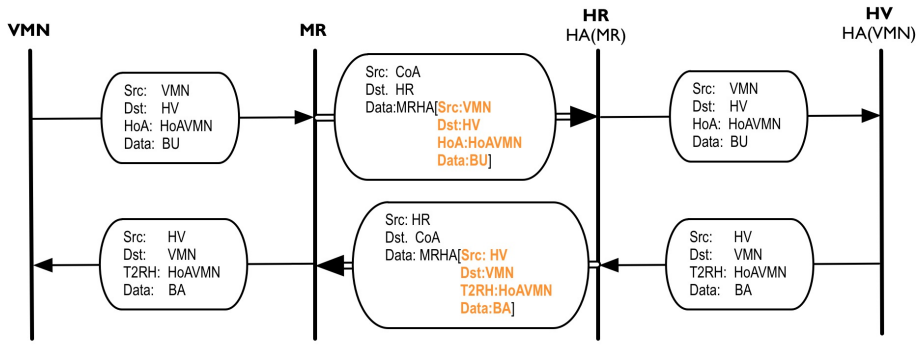


Figura 3.17 – VMN realiza a operação de *Binding Update* com o seu *Home Agent*

O *home agent* do *router* móvel MR é o HA(MR) que tem o endereço IP HR, enquanto o *home agent* do VMN é o HA(VMN) e tem o endereço IP HV.

O *Binding Update* do nó móvel visitante é realizado como se não estivesse dentro da rede móvel, pelo que os pacotes seguem encapsulados entre o *router* móvel MR e o *home agent* HA(MR).

Após a recepção do *Binding Acknowledgement* do HA(VMN), o nó móvel visitante deve tomar a iniciativa de otimizar a rota com o *Care-of Address* CoA, de modo a efetuar o registo do seu endereço IP original HoAVMN.

É necessário realizar o procedimento de otimização de rotas para o endereço IP *egress* do *router* móvel de topo, *root*-MR, de modo a garantir que é criada uma rota interna até ao endereço IP HoAVMN.

Na Figura 3.18 é possível ver que o VMN inicia procedimento de *Return Routability* enviando os pacotes de *Home Test init* (HoTi) e *Care-of Test init* (CoTi).

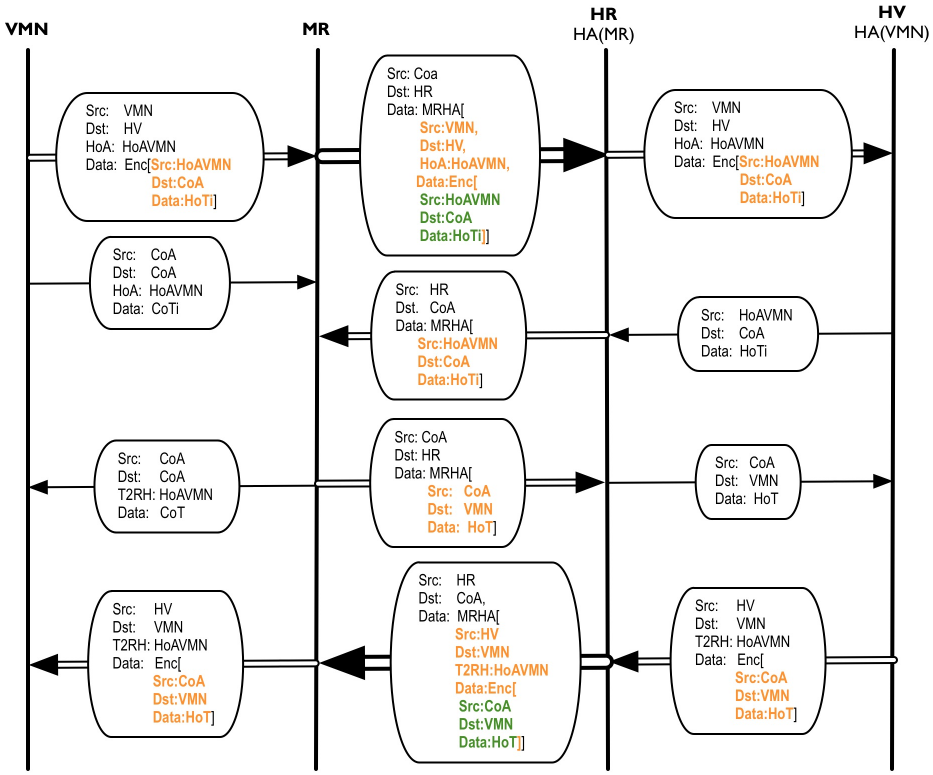


Figura 3.18 – Procedimento de *Return Routability* do VMN com o CoA do *router* móvel MR

Enquanto o CoTi chega rapidamente ao *router* móvel, o HoTi segue encapsulado com destino ao *Home Agent* HA(VMN). Quando passa pelo túnel MRHA entre o MR e o HA(MR) o pacote é duplamente encapsulado.

O procedimento de *Return Routability* termina quando o *router* móvel envia os pacotes de resposta *Care-of Test* CoT e *Home Test* HoT e que são devidamente recebidos pelo nó móvel visitante VMN.

O processo de registo do *Home Address* HoAVMN é feito quando o nó móvel visitante efetua o *Binding Update*, ilustrado na Figura 3.19.

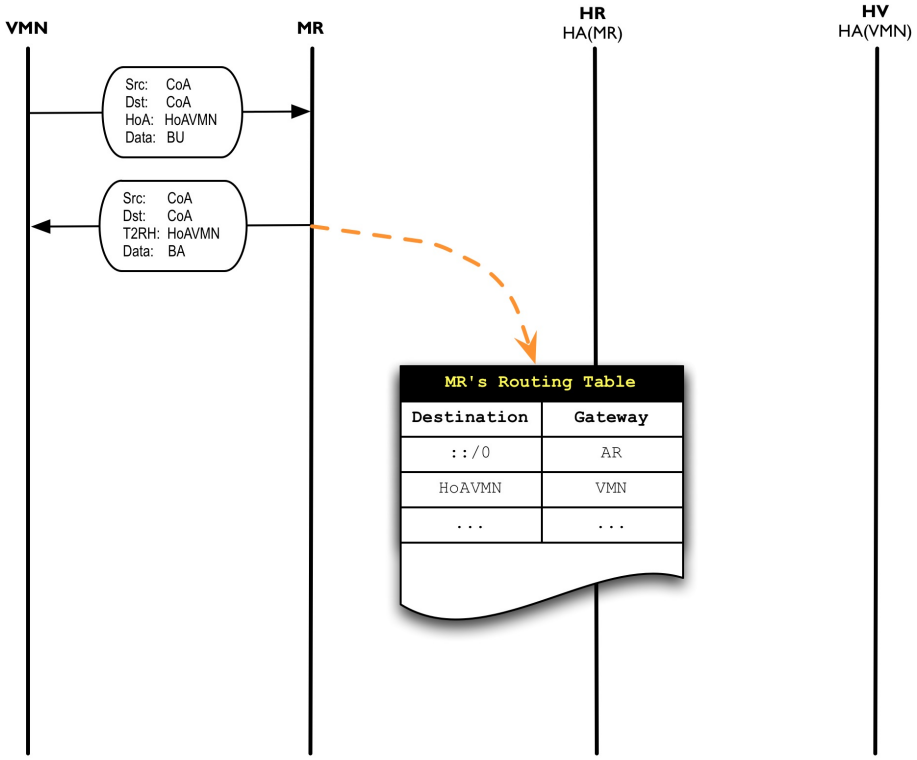


Figura 3.19 – VMN regista-se no MR através de um *Binding Update*

A partir deste instante, o *router* móvel MR sabe que está a falar com o legítimo detentor do endereço HoAVMN e pode atualizar a sua tabela de *routing* para passar a conter uma rota do endereço IP HoAVMN para o endereço IP VMN.

Agora o nó móvel visitante já pode registar o *Care-of Address* do *router* móvel de topo junto do seu *Home Agent* HA(VMN), conforme se pode ver na Figura 3.20. De seguida, pode começar a utilizar o CoA junto dos seus nós correspondentes.

Este processo pode-se tornar mais simples no caso de se proceder à alteração do protocolo NEMO Basic Support Protocol e MIPv6 para que seja possível o envio de um pacote autenticado para efetuar o registo do endereço origem do nó móvel visitante. Contudo, este estudo está fora do âmbito desta tese.

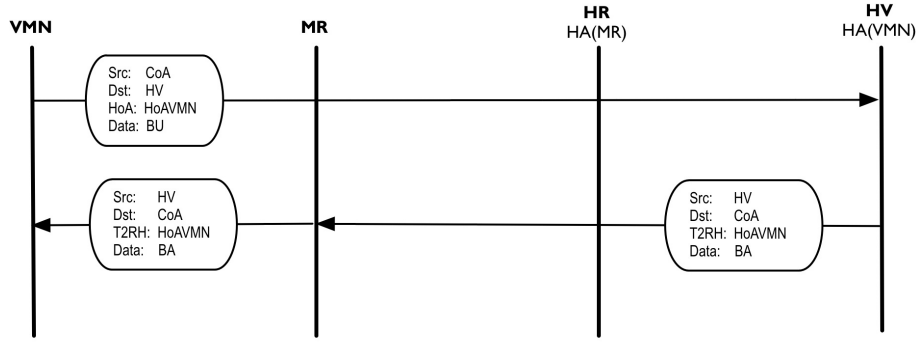


Figura 3.20 – VMN realiza *Binding Update* usando o seu novo CoA

3.4.9. Registo de uma rede NEMO

Sempre que uma rede móvel imbrica noutra, o OMEN obriga ao registo dos novos endereços IP, passando a permitir o encaminhamento dos pacotes. Os endereços IP que devem ser registados são o endereço IP origem do *router* móvel e o prefixo da rede móvel imbricada.

Para efeitos de exemplificação, assume-se que um *router* MR2 imbricou num *router* de topo MR1. O prefixo da rede móvel servida pelo MR2 é MNP2. O endereço IP origem do *router* móvel MR2 é HoAMR2 e o endereço IP adquirido na rede móvel de topo é MR2.

O registo do endereço IP HoAMR2 deve ser realizado da mesma forma que para o nó móvel visitante, conforme explicado anteriormente. Assim, o *router* móvel MR2 começa por realizar o *Binding Update* do novo endereço IP adquirido, MR2, junto do seu *home agent*. Após a atualização do endereço IP junto do HA(MR2), o MR2 deve recorrer ao procedimento *Return Routability* e subsequente *Binding Update* junto do *Care-of Address* do *router* de topo MR1.

A partir deste ponto, é acrescentado o endereço HoAMR2 à tabela de *routing* do MR1, conforme se pode ver na linha verde da Figura 3.21.

Para efetuar o registo do prefixo da rede móvel imbricada, o *router* móvel MR2 deve recorrer ao procedimento utilizado pelo nó móvel visitante para os dois endereços do extremo do prefixo da rede móvel: MNP2::/64 e MNP2:FFFF:FFFF:FFFF:FFFF/64. Deve-se, assim, realizar o procedimento de *Return Routability* e de *Binding Update* junto do *router* móvel de topo para estes dois endereços.

Obviamente, não se justifica fazer o passo de *Binding Update* junto do *Home Agent* do MR2 dado que não se pretende associar estes endereços IP ao *Care-of Address*.

Este processo apenas serve para provar que o MR2 é responsável pelo prefixo da rede móvel MNP2.

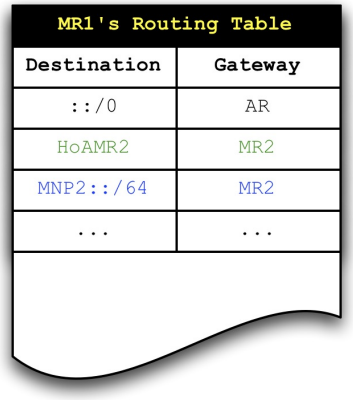


Figura 3.21 – Tabela de *Routing* do root-MR após registo de um sub-MR

A partir deste instante, a tabela de *routing* do *router* móvel `MR1` fica com a entrada identificada a azul na figura, relativa à rede `MNP2::/64`.

3.4.10. Comunicação intra-NEMO em *handoff*

Por comunicação intra-NEMO entende-se toda a comunicação entre dois equipamentos que estejam dentro de redes imbricadas e cujo tráfego otimizado entre elas não chega a sair da rede de topo.

A Figura 3.22 mostra que para o nó da rede móvel `MNN1`, que está na rede do *router* móvel `MR1`, comunicar com o nó da rede móvel `MNN2`, servido pelo *router* móvel `MR2`, não necessita de sair para fora da rede servida pelo *router* móvel `MR`.

Se for utilizada a solução NEMO Basic Support Protocol, a comunicação entre o nó `MNN1` e `MNN2` tem sempre que vir ao exterior, até aos *home agents* dos *routers* `MR`, `MR1` e `MR2`, atravessar os túneis `MRHA` respetivos, antes de atingir o dispositivo final.

As soluções de otimização de rotas dos paradigmas baseados em rede obrigam, na sua maioria, a que a comunicação entre o *router* de topo e o exterior exista.

Já o OMEN permite que a comunicação entre os dois equipamentos móveis se processe de forma direta e segura, sem que haja necessidade de ter acesso ao exterior em momento algum.

Para tal, assume-se que todos os equipamentos já se encontram na posse do *Care-of Address* correto e que estão em condições de iniciar o processo de *Route Optimization* um com o outro.

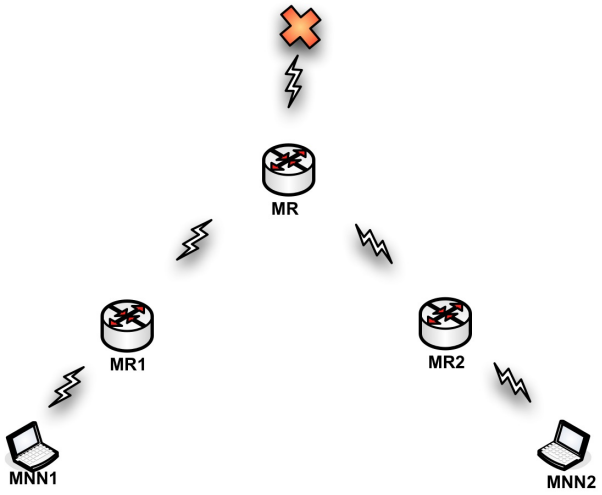


Figura 3.22 – Cenário de comunicação entre dois MNN em redes imbricadas

A Figura 3.23 apresenta os fluxos entres os nós da rede móvel.

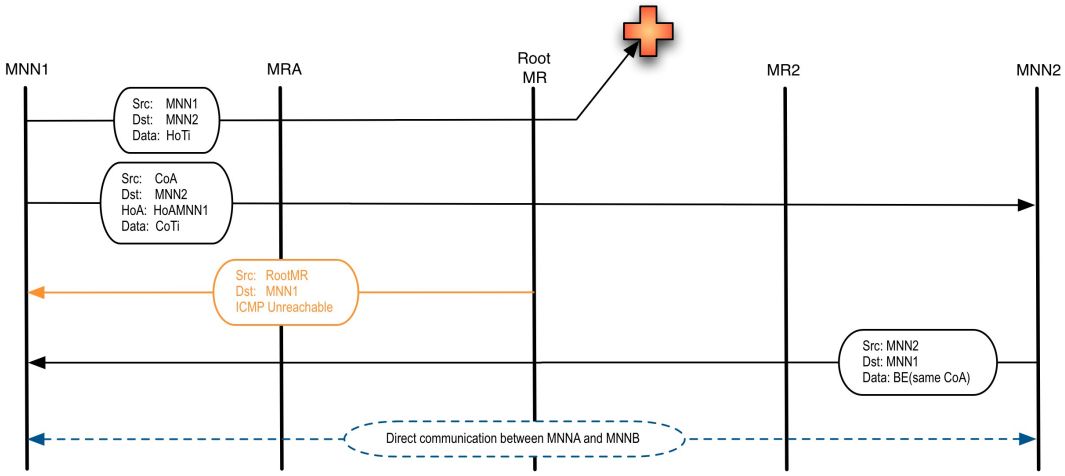


Figura 3.23 – Comunicação inter-imbricada em *handoff*

Para iniciar o processo de *Return Routability* entre o nó da rede móvel MNN1 e o MNN2, o primeiro envia os dois pacotes de *Home Test init* (HoTi) e de *Care-of Test init* (CoTi) com destino ao segundo.

Enquanto o CoTi tem caminho direto até ao equipamento MNN2, o HoTi é barrado pelo *handoff* quando chega ao *router* móvel de topo MR. Este tem que gerar um ICMP unreachable para o MNN1, indicando que não tem conectividade ao exterior.

Quando o nó da rede móvel MNN2 recebe o pacote CoTi, consegue perceber que o *Care-of Address* CoA é igual ao seu, indicando que se encontra na mesma rede móvel que o MNN1.

Deste modo, é enviado um pacote `Binding Error` informando que ambos possuem o mesmo `CoA`. Esta opção tem que ser adicionada ao RFC 3963, estando o protocolo NEMO aberto a esta inclusão. Contudo, enquanto não se procede à alteração do protocolo, o `MNN2` pode enviar um `CoTi` com o `CoA` ao `MNN1`, indicando-lhe que está na mesma rede. Em alternativa, o nó da rede móvel `MNN2` pode ignorar subsequentes pedidos de otimização de rotas, passando a comunicar diretamente como `MNN1`.
Assim que o nó da rede móvel `MNN1` recebe esta informação, começa a comunicar sem otimização de rotas com o `MNN2` já que tem a certeza que o caminho entre os dois nós é direto, não havendo necessidade de otimizar rotas.

No caso de uma nova rede NEMO imbricar na rede servida pelo *router* móvel `MR`, este procedimento não é possível enquanto o `MR` não conseguir ter acesso ao exterior, dado que para o novo *router* móvel registar o seu *Home Address* e o respetivo prefixo da rede móvel é necessário ter acesso ao seu *Home Agent* e, por conseguinte, acesso à Internet.

3.5. Validação

Não sendo objetivo do presente capítulo validar a proposta, é importante fornecer desde já uma visão geral do que foi feito nesse sentido. A informação detalhada sobre validação e avaliação será fornecida nos capítulos seguintes.
Para a verificação da consistência da proposta OMEN foi utilizado um emulador criado pelo autor da presente tese. Este emulador consiste num mecanismo muito simples de construção de pacotes IP sobre IP e respetivo envio para a rede.
Para que os pacotes sejam encaminhados é necessário que sejam cumpridos os requisitos de *routing* IP. Aos módulos que fazem *routing* foram adicionadas as funcionalidades de três soluções de mobilidade de rede, uma de cada paradigma: NEMO Basic Support Protocol; uma solução de otimização de rotas baseada em rede (MIRON); e a proposta OMEN.
Utilizando a ferramenta desenvolvida foi realizado um conjunto muito extenso de emulações. O objectivo das primeiras emulações foi o de verificar a consistência da solução e obter uma análise comparativa preliminar entre as três soluções mais representativas dos paradigmas de mobilidade de rede. Seguiram-se várias outras, a detalhar no Capítulo 5.
A Figura 3.24 mostra o primeiro cenário implementado para a validação da solução OMEN.

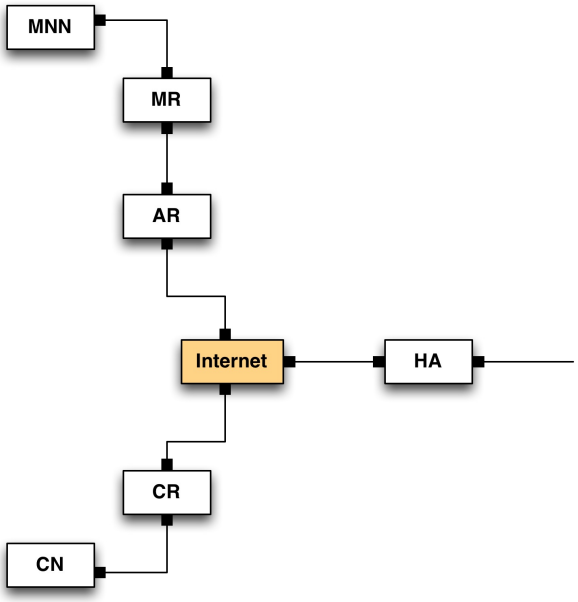


Figura 3.24 – Cenário para verificação de consistência da proposta OMEN

Foram criados dois dispositivos com suporte de MIPv6: o nó da rede móvel `MNN` e o nó correspondente `CN`.

O *router* móvel `MR` liga-se ao *router* de acesso `AR`, que por seu lado liga a um *router* central designado de `Internet`. Este *router* conhece as rotas para todas os prefixos em uso.

O nó correspondente `CN` liga-se à Internet através de um *router* de acesso identificado como *Correspondent Router*, `CR`.

Cada linha da figura identifica uma ligação entre os dois equipamentos, enquanto os quadrados terminais das linhas identificam interfaces distintas de cada equipamento. Assim, os *routers* têm todos duas interfaces, enquanto os nós finais apenas possuem uma.

O *Home Agent* `HA` tem uma ligação que foi interrompida porque o seu *router* móvel se moveu para a rede do *router* `AR`.

O túnel encapsulado `MRHA` entre o `MR1` e o `HA1` é conseguido com recurso à operação de BASE64 prevista em unix aplicada ao campo `DATA` do pacote, garantido desta forma que nenhum equipamento intermédio consegue ter acesso à informação dentro do túnel.

Com esta emulação conseguiu-se analisar em pormenor todos os pacotes em trânsito e validar a consistência do OMEN. Tendo-se cumprido os procedimentos dos protocolos MIPv6 e NEMO Basic Support Protocol, foi possível verificar que o OMEN funcionava de acordo com o definido. Também foram realizados testes com o protocolo NEMO Basic Support Protocol e o MIRON de modo a garantir a consistência do emulador.

Para verificar o funcionamento do OMEN em ambientes imbricados, foi criado o cenário ilustrado na Figura 3.25, que é igual ao anterior mas agora com mais um *router de acesso* `AR1`, mais uma rede móvel `MR1` e o seu respetivo *Home Agent* `HA1`.

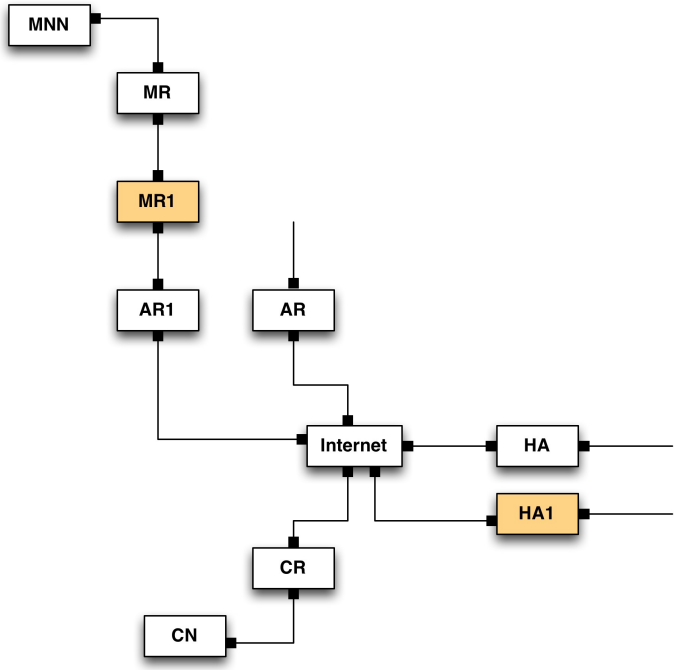


Figura 3.25 – Cenário para verificação de consistência do OMEN imbricado

Para a realização do duplo túnel encapsulado MRHA entre o `MR1` e o `AR1` foi realizado uma operação BASE64 sobre o pacote inteiro do primeiro túnel MRHA. Deste modo, há uma aproximação ao túnel MRHA utilizado no NEMO Basic Support Protocol.

Uma vez mais, o OMEN funcionou sem problemas, dado que os protocolos MIPv6 e NEMO Basic Support Protocol estavam implementados de acordo com o especificado.

A questão de segurança do OMEN também foi tida em consideração, tendo sido objeto de uma análise profunda. Não sendo a segurança um aspeto central desta tese, a secção seguinte resume as conclusões dessa análise.

3.6. Análise de segurança

3.6.1. Segurança no envio de CoA via ND

O envio do *Care-of Address* inserido dentro do pacote de *Router Advertisement* do protocolo Neighbor Discovery está sujeito aos mesmos tipos de problemas que o protocolo RFC 4861. Os tipos de ataques a que pode estar sujeito são: *Denial of Service* (DoS); *address spoofing*; *router spoofing*.

Em qualquer dos casos, o OMEN está sujeito a este tipo de problemas, sendo que o seu impacto é igual ao que seria para uma rede não móvel.

As soluções que venham a ser propostas para resolver o problema do *Neighbor Discovery* servirão igualmente para o OMEN.

3.6.2. Segurança no Registo de um VMN

O procedimento de registo de um nó móvel visitante, VMN, conforme explicado anteriormente, contém um nível adicional de passos para garantir a identidade do nó da rede móvel, sem que haja necessidade de modificar o RFC 6275 ou o 3963.

No caso de se justificar, pode-se proceder à alteração destes protocolos para passarem a incluir o OMEN e, desta forma, ser possível realizar o registo do nó móvel visitante de uma forma mais célere.

A segurança do registo e utilização do OMEN por parte de um nó móvel visitante está sujeita apenas aos problemas inerentes à solução MIPv6, não acrescentando o OMEN qualquer nível de insegurança.

3.6.3. Segurança no registo de uma rede NEMO

O registo do endereço IP original do *router* de uma rede móvel imbricada junto do seu *parent-router* está sujeito às mesmas questões de segurança de um nó móvel visitante, VMN, conforme explicado anteriormente.

Para a passagem de informação sobre o prefixo da rede móvel, MNP, em que se optou por realizar a mesma operação de registo de um VMN para os endereços dos limites da rede móvel, MNP::/64 e MNP::FFFF:FFFF:FFFF:FFFF/64, existe o problema de não ser garantida a autenticidade do *router*.

É possível que um nó móvel, MN, requeira estes dois endereços IP e, quando se mover para outra rede, consiga fazer o registo do prefixo da rede como se fosse um *router* móvel, possibilitando o ataque de *Denial of Service* para todo o tráfego dentro da rede para o prefixo registado.

A melhor forma de resolver este problema seria adicionar um mecanismo de registo do prefixo da rede móvel ao protocolo NEMO Basic Support Protocol, RFC 3963 junto de uma rede móvel quando imbricado.

3.7. Conclusão

No presente capítulo foi apresentada a proposta OMEN para mobilidade de redes. Trata-se de uma proposta que se enquadra no paradigma de mobilidade baseada no cliente, ao contrário da maioria das soluções atuais, que são baseadas na infraestrutura de rede.

Após uma descrição geral da proposta detalhou-se o seu funcionamento e apresentou-se uma especificação. Abordaram-se, também, os cenários para validação inicial da proposta e as questões de segurança.

Com base na informação apresentada, é possível antever alguns dos potenciais aspetos positivos do OMEN, dos quais se salientam a não necessidade de alteração de protocolos já existentes e a não sobrecarga dos *routers* móveis.

4. mobSim – ferramenta para emulação de mobilidade

O suporte nativo de mobilidade de redes por parte dos simuladores existentes encontra-se sujeito a limitações consideráveis. Este défice resulta não só da falta de funcionalidades necessárias, mas também da dificuldade em suportar cenários de dimensão razoável.

Tipicamente, os simuladores existentes lidam apenas com cenários de pequena dimensão, para os quais o impacto das diferenças de desempenho decorrentes das arquiteturas em estudo são dificilmente visíveis ou até mesmo inexistentes. No entanto, o facto dessas diferenças não serem visíveis em cenários de pequena dimensão não significa que elas não existam, podendo, até, ter um impacto determinante em cenário de maior escala.

Dadas as limitações referidas, foi criada uma nova ferramenta com o objetivo de emular cenários que podem atingir grande dimensão, com suporte nativo de mobilidade de nós e de redes. Dado que é uma ferramenta com características de escalabilidade horizontal, o cenário encontra-se apenas limitado pelo tamanho do *cluster* onde esta é executada. Naturalmente, a ferramenta também é adequada a cenários de pequena dimensão, como o que foi utilizado para estudar a viabilidade da proposta OMEN.

Este capítulo inicia-se com uma caracterização das soluções de simulação existentes, no que toca a mobilidade de nós e redes, a que se segue a identificação da motivação para a criação da ferramenta mobSim. Segue-se a apresentação das funcionalidades, arquitetura e características da ferramenta. Por fim, são apresentados os principais aspetos de implementação e utilização do mobSim.

4.1. Soluções de simulação existentes

No início do presente trabalho, quando foi necessário estudar a viabilidade da solução proposta, constatou-se que os simuladores existentes tinham suporte nativo de mobilidade de nós. Contudo, eventualmente por ser um tema demasiado recente, havia uma lacuna enorme no que respeitava o suporte de mobilidade de redes.

Nessa mesma altura, ficou claro que seria necessário avaliar o desempenho da proposta em cenários de alguma dimensão, tendo em vista um estudo aprofundado das soluções existentes. A dimensão dos cenários a estudar excluía a hipótese de implementação real. Por outro lado, a ferramenta de simulação a utilizar teria que comportar cenários dessa dimensão. Na procura do simulador adequado, teve-se o cuidado de considerar estas questões.

4.1.1. Network Simulator version 2, ns-2

O Network Simulator version 2[8], conhecido como ns-2, é uma das ferramentas de simulação mais extensamente utilizadas. Trata-se de um simulador de eventos discretos, que utiliza C++ e TCL. É altamente portável, com extensa documentação e algum suporte nativo para a mobilidade de nós através do MobiWan [Inria08].

Contudo, não existe muito suporte para cenários de mobilidade de redes, obrigando os utilizadores a usarem o módulo MobiWan como base para implementar a funcionalidade de NEMO Basic Support Protocol [Shahriar07].

O ns-2 tem algumas capacidades interessantes no que concerne cenários de larga escala [Henderson08a]. Contudo, esta escalabilidade é vertical, ou seja, o tamanho dos cenários é limitado pelo *hardware* do sistema, mais concretamente pela sua memória.

Por outro lado, a documentação do ns-2 alerta para o facto de que um número elevado de nós combinado com uma carga elevada do sistema pode levar a resultados erróneos.

4.1.2. Network Simulator version 3, ns-3

A 3ª versão do Network Simulator [ns-308] [ns-308a] é uma evolução do simulador anteriormente apresentado. É desenvolvido em C++, com suporte opcional para uma interface em python. É uma ferramenta de simulação com um futuro promissor, que já conta com uma documentação extensa e bastante completa.

O seu objetivo é permitir simulações de elevada precisão e confiança para ambientes académicos e, ao mesmo tempo, resolver algumas limitações existentes no ns-2, com particular enfoque na gestão da memória e análise de erros (*debugging*).

De modo a melhorar a fiabilidade e obter a melhor performance possível, o ns-3 permite a integração da ferramenta com componentes reais do sistema, tais como módulos do *kernel*,

[8] http://www.isi.edu/nsnam/ns/

interfaces ou programas. Adicionalmente, é possível ter várias instâncias ns-3 a executar em diversas máquinas distintas.

Contudo, dado que em 2008 ainda era um produto recente, apenas possuía suporte de mobilidade de nós ao nível da camada 2. Existem alguns trabalhos para implementar a mobilidade IP [Mauchie10], mas não existe nenhum para implementar a mobilidade de redes.

4.1.3. OMNet++

Outro simulador bem conhecido é o OMNET++ [OMNet09], que consiste num conjunto de bibliotecas desenvolvidas em C++. A arquitetura modular do simulador torna-o extensível.

Possui uma documentação completa e existem diversas contribuições externas, o que demonstra uma atividade intensa à volta desta ferramenta.

No que respeita à mobilidade, possui suporte nativo para a mobilidade de nós. Adicionalmente, existe um módulo para mobilidade de nós que foi desenvolvido externamente, conhecido como *extensible MIPv6* (xMIPv6) [Yousaf08]. Contudo, não existe suporte para mobilidade de redes.

As preocupações com a escalabilidade foram a base para a criação de um projeto para processamento paralelo utilizando este simulador, em 2003 [Sekercioglu03]. Contudo, não existe muita documentação sobre este produto e não parece claro que seja oficialmente suportado.

4.1.4. OPNET Modeler

O OPNET Modeler [OPNET08] é uma ferramenta de simulação orientada a objetos, de eventos discretos, com suporte nativo de mobilidade de nós. Apesar de não suportar de forma nativa o NEMO Basic Support Protocol, existem alguns projetos que fornecem extensões para a mobilidade de redes [KCL08]. Por exemplo, o [OPNETSC06] fornece suporte limitado e incompleto para a variante de NEMO para redes veiculares (VANET).

O OPNET tem boa escalabilidade vertical, com capacidade de explorar processadores multicores ou máquinas multiprocessador de modo a acelerar a o tempo de execução das simulações [OPNET08a].

Não possui escalabilidade horizontal, pelo que fica limitado ao *hardware* onde for executado.

4.1.5. Resumo das soluções existentes

A Tabela 4.1 apresenta a comparação entre as diversas ferramentas de simulação de redes.

	ns-2	ns-3	OMNet++	OPNet
Suporte nativo de MIPv6	✔[9]	✖	✔	✔
Suporte nativo de NEMO	✖	✖	✖	✔[9]
Escalabilidade	vertical	horizontal	vertical	vertical

Tabela 4.1 – Lista das ferramentas de simulação analisadas

Claramente, para todos os simuladores referidos existe uma enorme preocupação em relação à fiabilidade dos resultados.

Embora seja comum o suporte para a mobilidade de dispositivos, o mesmo não se verifica no que respeita a mobilidade de redes.

É evidente que existe algum cuidado com as questões de escalabilidade. Contudo, nenhum dos simuladores apresentados tem capacidade para implementar de forma efetiva cenários de mobilidade de rede de larga escala.

4.2. Motivação

A mobilidade de nós e redes é uma área bastante ativa em termos de investigação científica. Muitos dos estudos efetuados nesta área recorrem a ferramentas de simulação. As razões centram-se em dois fatores cruciais: por um lado, antevê-se que a Internet do futuro seja cada vez mais móvel e, por outro lado, não é viável a implementação, em laboratório ou em ambiente real, de propostas de soluções genéricas de mobilidade para cenários de larga escala.

No entanto, o estudo de soluções de mobilidade de redes existentes ou emergentes com recurso a simulação encontra-se largamente condicionado pelas ferramentas de simulação existentes que, como vimos anteriormente, têm diversas limitações.

É importante notar que cenários de pequena e média escala levam a que as diferenças entre as várias propostas se tornem irrelevantes, permitindo que uma proposta menos boa não se comporte tão mal e que uma boa proposta não sobressaia de forma relevante, comprometendo a escolha da melhor solução.

[9] Suporte através de um módulo externo

Tal como ficou patente na secção anterior, nenhuma das ferramentas de simulação existentes atualmente tem enquadramento para simulações de mobilidade de redes em larga escala, envolvendo dezenas de milhares de redes e equipamentos finais. Dada a necessidade e interesse em estudar este tipo de cenários, optou-se pelo desenvolvimento de uma nova ferramenta, de simulação ou de emulação, com capacidade de lidar com estas dimensões.
Naturalmente, tendo em consideração os enormes requisitos de processamento e fidelidade dos resultados, a ferramenta deverá:

1) ser construída e ser optimizada para operar em *clusters*, tirando partido das capacidades de processamento paralelo;
2) comportar detalhes de simulação/e,mulação muito finos, incluindo os campos do cabeçalho do protocolo, a operação dos mecanismos de mobilidade e, ainda, a configuração dos equipamentos finais individuais, *routers*, redes e cenários globais;
3) ser altamente flexível em termos de definição de cenários, incluindo a especificação das rede fixas e móveis, equipamentos finais fixos e móveis, topologias e comportamentos dinâmicos;
4) Permitir a utilização dos mesmos parâmetros e condições para as diferentes soluções que venham a ser analisadas.

Destas necessidades nasceu a ferramenta, chamada *mobSim*, que é um emulador de mobilidade de redes.
O facto do mobSim ser um emulador conduz a uma vantagem substancial em relação aos restantes simuladores, porque os mecanismos e tráfego são reais e, deste modo, refletem melhor o funcionamento das soluções em estudo.

4.3. Funcionalidades implementadas

O mobSim foi desenvolvido tendo em vista a comparação dos três paradigmas de mobilidade de redes identificados no Capítulo 2: centrado nos equipamentos antigos, baseado na rede e baseado no cliente final. Dado que estes paradigmas usam protocolos normalizados, tornou-se essencial cumprir de forma tão correta quanto possível as especificações dos respetivos RFC.
A implementação dos protocolos no mobSim vai ao pormenor dos campos dos cabeçalhos, aproximando-o muito a uma implementação.

A Tabela 4.2 apresenta a lista de funcionalidades normalizadas implementadas pelo mobSim.

IPv6 Basic Support Protocol (RFC 2460)
Hop limit
Next Header implementations
ICMPv6 (RFC 4443)
ICMP echo, reply, unreachable, time exceeded
IPv6 encapsulation (RFC 2473)
Mobility Header (RFC 6275)
Type 2 Routing Header (routing type equals 2 for MIPv6 MIPv6 final hop HoA)
Home Address option
Mobile IPv6 (RFC 6275)
Binding Update
Binding Acknowledgement (binding accept, reject)
Return Routability procedure
Home Test init
Care-of Test init
Home Test
Care-of Test
Nonce utilisation (RFC 6275 section 5.2.2)
Binding Refresh
Neighbor Discovery (RFC 4861)
Router Advertisement
Router Solicitation
Network Mobility (NEMO) Basic Support Protocol (RFC 3963)
Bidirectional tunnel (MRHA tunnel)
Binding Update
Binding Acknowledgement
Home Agent implementation
Mobile Router implementation

Tabela 4.2 – Funcionalidades standard do mobSim

Estas funcionalidades foram usadas para construir as implementações genéricas dos três paradigmas de mobilidade de redes.

O paradigma centrado nos equipamentos antigos é implementado respeitando o protocolo NEMO Basic Support Protocol (RFC 3963).

As funções de mobilidade do paradigma baseado na rede – incluindo a otimização de rotas – são realizados pelos *routers* móveis, em vez dos nós da rede móvel. A implementação deste paradigma é feita com recurso a duas soluções representativas: MIRON e ORC.

Já a implementação do paradigma baseado no cliente segue a proposta OMEN, em que as operações de mobilidade são executadas pelos nós da rede móvel.

Naturalmente, o mobSim está construído de modo a permitir a implementação de outras soluções de mobilidade.

4.4. Arquitetura do emulador

A Figura 4.1 apresenta uma visão geral da arquitetura da ferramenta de emulação mobSim.

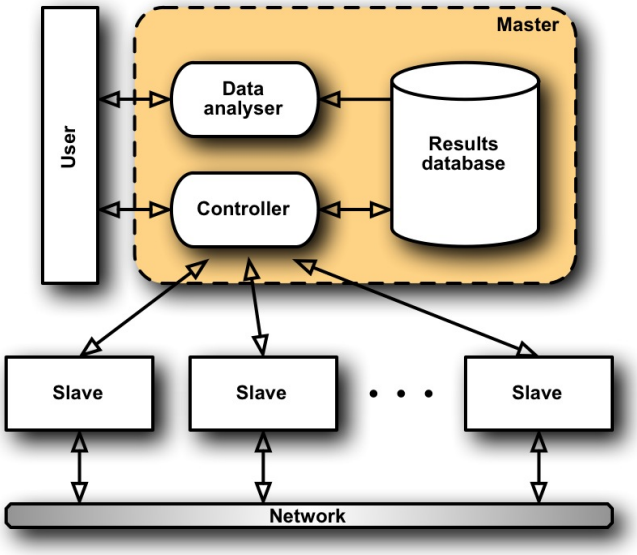


Figura 4.1 – Arquitetura mobSim

Como esta ferramenta foi especificamente concebida para ser executada em *clusters*, o desenho da arquitetura do mobSim reflete naturalmente esta opção.

Os seus componentes principais são o nó principal (`Master`) e os nós operacionais (`Slave`), cujo número é apenas limitado pelo número de servidores físicos disponíveis no *cluster*. No caso do *cluster* Milipeia [LCA08], utilizado nas simulações, a comunicação entre o nó principal e os nós operacionais é realizada através de uma rede de alta velocidade, nomeadamente a 1Gbps.

O nó principal é composto por três módulos distintos: o módulo controlador (`Controller`), o módulo de análise de dados (`Data analyser`) e a base de dados dos resultados (`Results database`).

O controlador (`Controller`) é a componente mais importante do simulador. É responsável por criar o cenário de simulação de acordo com as instruções do utilizador, construir e enviar o código para ser executado por cada nó operacional, iniciar a simulação, obter os resultados dos nós operacionais, guardar os resultados na base de dados e terminar a execução de todo o cenário no fim da simulação.

Dada a enorme quantidade de tarefas, a extrema importância e a criticidade das operações realizadas pelo controlador, decidiu-se que o nó principal teria um servidor físico dedicado. Este nó não deve executar operações externas de modo a não correr o risco de vir a ter alguma interferência nos resultados finais.

Quando a simulação é iniciada pelo *cluster*, um dos servidores físicos é eleito como o controlador, ficando os restantes servidores à espera de instruções dele provenientes.

Os nós operacionais executam o cenário de simulação. É mais correto afirmar que os nós operacionais executam emulação dado que, de facto, o que acontece é que o são construídos pacotes IPv6 e são enviados para a rede.

De facto, pode-se considerar que é criado um *overlay* por onde circulam os pacotes em estudo, e que são enviados sobre a rede real, conforme se pode ver na Figura 4.2.

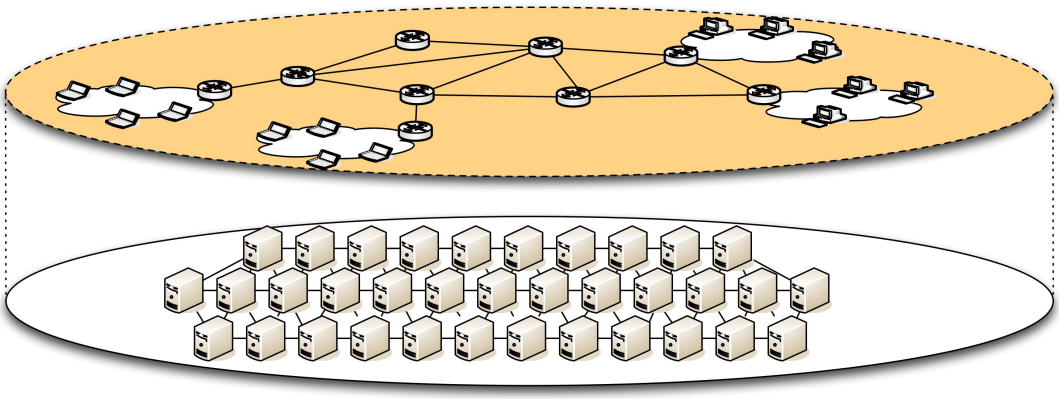


Figura 4.2 – *Overlay* mobSim

Quando um pacote é transmitido entre dois equipamentos virtuais, este é enviado através da rede física que interliga os servidores físicos do *cluster*. Na Figura 4.3 é exemplificada a comunicação entre equipamentos virtuais.

Os traços entre os nós do *cluster* (linhas marcadas com o número 2) identificam os cabos de rede física. Os traços marcados com o número 3 representam as comunicações entre os equipamentos virtuais.

Quando um equipamento virtual envia um pacote para outro equipamento virtual, este segue o caminho marcado com o número 1. É possível verificar que o pacote é enviado para o servidor físico do *cluster* e, posteriormente, encaminhado pela rede física até atingir o servidor físico pretendido, que será responsável por encaminhar para o equipamento virtual seguinte.

Deste modo, torna-se possível a comunicação entre equipamentos virtuais utilizando redes reais.

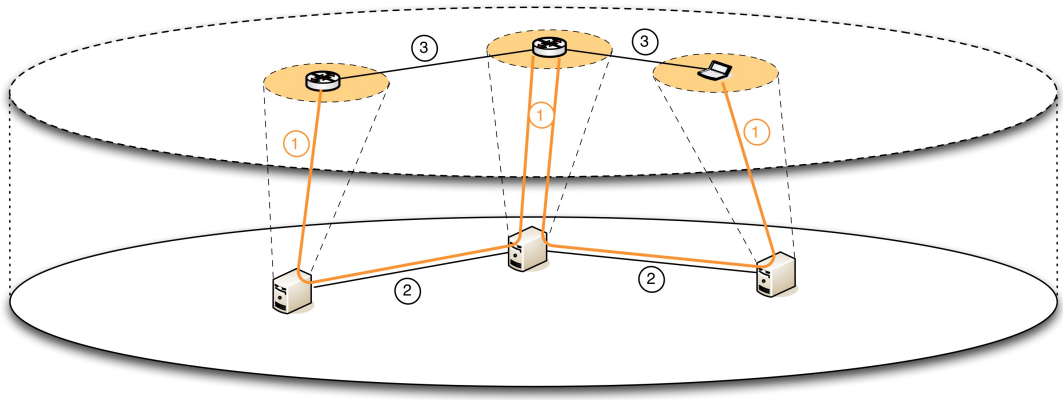


Figura 4.3 – Exemplo de comunicação entre equipamentos no mobSim

Cada nó operacional pode executar o código de várias centenas de equipamentos virtuais. Na prática, é como se fossem equipamentos reais comunicando entre si. Pacotes reais são construídos e enviados através dos equipamentos físicos. Existem tabelas de *routing* e os mecanismos de mobilidade seguem os respetivos RFCs.

Se o *routing* determina que os equipamentos comunicantes residem dentro do mesmo nó operacional, então é utilizado a interface *localhost* para o envio dos pacotes. Se estes residem em nós operacionais diferentes, então é utilizada a rede a 1Gbps.

O controlador determina que equipamentos virtuais deverão ser executados em que nós operacionais, no início de cada simulação.

Os nós operacionais enviam os dados relativos à simulação para o controlador. É da responsabilidade do nó principal a manipulação desta informação a partir da base de dados de resultados.

Após a execução da simulação, o utilizador pode fazer uso do módulo de análise de dados de modo a poder extrair a informação relevante. Alguns dos valores possíveis são o *round trip time* (RTT) dos pacotes, tempos de *handoff*, tempos necessários para criar os túneis bidirecionais, tempos para a realização do procedimento *return routability*, entre outros. É interessante notar que novos valores ou métricas podem ser adicionadas ao emulador, enriquecendo o seu funcionamento.

O módulo de análise de dados também permite dar acesso aos *logs* que poderão ser utilizados para uma análise mais fina, tal como o caminho tomado pelos pacotes, as ações que foram realizadas para a criação dos túneis bidirecionais, ou para efeitos de análise de problemas.

De igual modo, é possível ter acesso aos dados em bruto para análises de mais baixo nível, ou para *debugging*.

4.5. Características do mobSim

O mobSim é extremamente flexível e permite a criação de qualquer topologia de rede, com qualquer número de redes, *routers* e equipamentos virtuais, assim como com qualquer número de níveis de imbricação de redes móveis.
A maioria dos parâmetros são configuráveis ao nível do nó ou do *router*. É possível definir parâmetros específicos para cada nó e/ou rede.

4.5.1. Tempos

A definição do valor *tempo* serve para introduzir atrasos de modo a simular o tempo que levaria para executar determinada tarefa. Por exemplo, existe um valor específico para simular o tempo que demora a executar a operação de encapsulamento de um pacote.
A configuração dos tempos inclui, nativamente, os seguintes:

- *handoff* da camada de *data link*;
- obtenção do novo endereço IP (através de DHCP ou por autoconfiguração);
- procedimento *binding update*;
- procedimento de *binding acknowledgement*;
- velocidade de comunicação da linha;
- tratamento dos pacotes
    - HoTi;
    - CoTi;
    - HoT;
    - CoT;
- encapsular e extrair dados de túneis.

4.5.2. *Routing*

A criação das tabelas de *routing* de cada equipamento virtual é feita de forma automática pelo nó controlador, com base na topologia de rede definida pelo utilizador.

Seria bastante complexo definir de forma manual e precisa a rede para diversas dezenas de milhar de redes. Assim, optou-se por delegar essa função no nó controlador.
Dada a natureza do simulador e os objetivos em jogo, não se implementou qualquer mecanismo de *routing* dinâmico, embora este não seja difícil de se alcançar.

4.5.3. Gestão dos comportamentos

O utilizador pode definir as características de tráfego e quais os nós que o geram. Também é possível definir o comportamento dinâmico dos cenários, isto é, o movimento dos nós e redes.
Os parâmetros, topologia e comportamentos dinâmicos são mantidos na execução das simulações das diversas soluções de mobilidade em estudo. Assim, torna-se possível comparar o comportamento das diversas propostas perante os diversos casos em estudo, já que os cenários repetem-se exatamente da mesma forma.

4.5.4. Tipos de equipamentos

O mobSim suporta duas categorias de dispositivos: dispositivo de *routing* e dispositivo terminal. O primeiro inclui os *router* de topo, *routers* ordinários, *routers* móveis (MR) e *home agents* (HA). O segundo comporta os nós fixos da rede (LFN), nós móveis locais (LMN), nós móveis visitantes (VMN), e nós correspondentes (CN). Os nós correspondentes podem ser fixos ou móveis (LFN, LMN ou VMN).
Por defeito, todos os dispositivos virtuais suportam MIPv6.

4.5.5. Controlo das simulações

Dado que o emulador foi desenvolvido para ser utilizado em grandes *clusters*, em que a utilização dos recursos tem que ser otimizada e o tempo de CPU é muito caro, foi necessário ter algum cuidado de modo a minimizar o tempo necessário para cada simulação.
Por este motivo, o arranque das simulações é feito *off-line*, e a inicialização dos diversos nós operacionais é realizada de forma automática pelo nó controlador, que também controla a execução e término da simulação.
Durante a execução da simulação não há qualquer intervenção humana. De facto, em geral não é possível determinar, com absoluta certeza, quando é que a simulação será executada, pois tal depende da política de operação do cluster que for utilizado. Tipicamente, assim que

for solicitada a execução da simulação, esta entra numa fila espera de programas e pode ser necessário esperar até alguns dias pela sua execução.

4.6. Implementação do mobSim

O mobSim foi desenvolvido em perl [Perl08], sendo o seu núcleo constituído por menos de 2000 linhas de código. O nó controlador consiste num conjunto de *scripts* que são responsáveis por construir o código para cada cenário, transmiti-los para os nós operacionais e controlar as operações de gestão – iniciar simulação, parar simulação, enviar tráfego, receber os resultados.

A criação de dispositivos virtuais com capacidade de *routing* é feita com base num *script* genérico que abre dois *sockets* UDP. Este código tem as funcionalidades básicas de um *router*. Se for pretendido ter alguma caraterística específica, tal como ser um *home agent* ou um *router* móvel, então o nó controlador acrescenta o código necessário.

A criação dos dispositivos virtuais que vão representar os nós móveis é feita com base num *script* que apenas abre um *socket* UDP. Da mesma maneira que o caso anterior, sempre que é preciso adicionar suporte de alguma funcionalidade específica, o nó controlador modifica o seu código para suportar as novas operações.

De seguida, são analisados os *scripts* mais importantes do simulador mobSim.

4.6.1. *Script* mobsim_master

O *script* mobsim_master é o responsável por criar o código que irá ser executado em cada nó operacional. Este código é parte integrante do nó controlador.

O programa começa por ler a informação base do cenário em estudo, ou seja, todos os endereços IP utilizados e todos os atrasos definidos por defeito. Posteriormente, obtém a listagem dos dispositivos virtuais a criar.

A parte central do código é apresentada na Figura 4.4.

```
open FIND, "find $f -name \"*.conf\" | sort |";
while ($file=<FIND>) {
    [...]
}
close FIND;
```

Figura 4.4: Parte central do código

Cada ficheiro `*.conf` corresponde a um equipamento virtual.

Os limites dos diversos comandos unix, tais como o *rm* ou o *ls*, introduziram problemas não esperados na fase inicial da concepção do programa. Dado que poderão existir dezenas de milhar de equipamentos virtuais, em que cada um é representado por um ficheiro de texto, constatou-se que a utilização destes comandos levava a problemas de memória e a que o programa central terminasse com um erro de sistema. Também se experimentou o comando interno do perl de obtenção dos ficheiros numa determinada diretoria, `<$f/*.conf>`, uma vez mais sem sucesso.

Para contornar este problema, foi necessário recorrer ao comando `find`. Este comando consegue fazer uma gestão eficiente da memória do sistema, nunca trazendo qualquer limite de utilização.

Este *script* realiza várias verificações de consistência de modo a criar o dispositivo virtual. Por exemplo, não deixa criar o código se não estiver definido o seu *default gateway* ou se não estiver definido o seu endereço IP.

A inserção de código adicional é feito com base na opção `extracode`. Após verificar que o dispositivo virtual necessita uma determinada função, o mobsim_master vai buscar o código através das seguintes linhas:

```
if ($fconfs{"extracode"}) {
        open EXTRACODE, $f."/".$fconfs{"extracode"};
        @extracode=<EXTRACODE>;
        close EXTRACODE;
}
```

Figura 4.5: Obtenção do código adicional

O código extra é inserido na secção correta através das linhas:

```
$code=""; $extracode=0;
open INSERTCODE, $conf{"bin_dir"}."/".$fconfs{"prog"};
while (<INSERTCODE>) {
        if (/^#INSERT#CODE#HERE#/) {
                [...]
        }
        if ($extracode) {
                if (/^#END#INSERT#CODE#HERE#/) {
                    [...]
                }
        }
        $code.=$_;
}
close(INSERTCODE);
```

Figura 4.6: Inserção de código adicional

A variável `$code` vai conter o código devidamente construído, quer tenha código adicional ou não. O resultado final desta operação é que constitui a informação que será enviada para os nós operacionais do *cluster*, através do código da Figura 4.7.

```
$prog{$aux1}=$aux."-".$aux1.".pl";
my $sock = create_sock($auxipaddr{$fconfs{"ipaddr1"}},
                       65000);
die "Couldn't contact destination port\n" unless ($sock);
print $sock "$prog{$aux1}\n";
print $sock $code;
close $sock;
```

Figura 4.7: Envio do código aos nós operacionais

4.6.2. *script* create_ipaddresseslst

Este é um dos *scripts* mais importantes do nó controlador, pois é ele que cria as rotas, associa os endereços IP físicos aos endereços IP virtuais, cria os ficheiros de configuração de cada equipamento virtual e cria as bases para que todo o simulador possa ser executado.

Este *script* começa por obter a listagem dos nós do *cluster* que pode usar. Esta listagem é dinâmica e depende do *cluster* utilizado (no caso do presente trabalho, o cluster Milipeia, da Universidade de Coimbra).

É com base nos endereços IP dos nós do *cluster* que o *script* consegue fazer a associação do endereço IP real ao endereço IP virtual: Deste modo é possível constatar que este emulador é adaptável a novos *clusters* sem modificações adicionais.

Para poder construir o cenário, o create_ipaddresseslst lê um ficheiro de configuração que inclui o cenário que o utilizador pretende implementar. Este ficheiro será explicado mais adiante.

Na posse desta informação, o *script* passa a ter toda a informação relativa às redes que deverão ser criadas, assim como de todos os equipamentos virtuais a implementar.

Em função dos diversos tipos de equipamentos, o programa vai adicionando diversas informações especificas (1 ou 2 endereços IP, *default gateway*, etc.). O seguinte código ilustra este passo:

```
open AX, "$f/$s/$file";
while (<AX>) {
        [...]
        # Specific sections
        if ($type eq "toprouter") {
            [...]
        }
        if ($type =~ m"^(mobile)?router$") {
            [...]
        }
```

```
        if ($type eq "mobilerouter") {
            [...]
        }
        if ($type eq "node" or $type=~/^multiplenode:/) {
            [...]
        }
        if ($toprouter{$myinfo{"gateway"}}) {
            [...]
        }
}
close AX;
```

Figura 4.8: Configuração dos dispositivos virtuais

Dado que o emulador precisa de conhecer todos os endereços IP que podem ser utilizados, este *script* ainda analisa todos os movimentos que os diversos nós e *routers* móveis realizam, reservando endereços IP para estas situações. Por exemplo, se um *router* móvel se desloca para as redes A, B e C, então este script reserva endereços topologicamente corretos para atribuir o *care-of address* em cada uma destas redes.

O código que implementa esta componente é:

```
open AX, "$scriptfile";
while (<AX>) {
        next unless (/^move;[^;]+;(.*)\n$/);
        $mrmove{$1}=1;
}
close AX;
```

Figura 4.9: Reserva de endereços virtuais para mobilidade

Agora que o *script* já possui todas as informações que necessita, pode construir os ficheiros específicos de cada dispositivo virtual. Estes ficheiros têm uma configuração do género `*.conf`. Também é criado o ficheiro que faz a associação do endereço IP real ao endereço IP virtual. As linhas de código que realizam estas operações estão a seguir sumariadas:

```
foreach $k (keys %final) {
    [...]
}
open IPADDRLST, ">$f/$paradigm/ipaddresses.lst";
print IPADDRLST "# All basic IPs needed\n";
print IPADDRLST "LOG:1 $myipaddris:1051\n";
foreach $k (keys %ip) {
    [...]
}
```

Figura 4.10: Associação do endereço IP real ao endereço IP virtual

4.6.3. *Script* mobsim_2ports_udp

Este é o *script* que serve de base a todos os dispositivos virtuais que emulam equipamentos com capacidade de *routing*.

Este programa começa por abrir dois *sockets* UDP, nos portos definidos pelo *script* create_ipaddresseslst. Esta operação é realizada com as linhas da Figura 4.11.

```
$sock=create_sock($conf{"port1"});
$check_socks=new IO::Select($sock);
$save_sock{$conf{'port1'}}=$sock;

$sock1=create_sock($conf{"port2"});
$check_socks->add($sock1);
$save_sock{$conf{'port2'}}=$sock1;
```

Figura 4.11: Abertura de dois *sockets* UDP

Por defeito, o modo de funcionamento de um *socket* consiste em abrir um porto, ficando o programa suspenso à espera de entrada de dados. Contudo, para o caso do *routing*, não é possível saber em que porto vai dar entrada um pacote. Deste modo, foi necessário conceber o *script* de modo a que este esteja permanentemente à escuta em ambos os portos, conforme se mostra a seguir:

```
while (@ready=$check_socks->can_read) {
        foreach $k (@ready) {
            [...]
        }
}
```

Figura 4.12: Obtenção de pacotes dos *sockets* UDP

Sempre que é recebido um pacote, é chamada a função `process_packet($buf);` que consiste num processamento básico do pacote. Este procedimento decide se opta pelo encaminhamento do pacote ou pela geração de um erro ICMP.

Também é esta a função que é reescrita para contemplar os casos particulares *router* móvel ou *home agent*. No caso de não haver modificação deste código, ou seja, quando o *router* não é nem um *router* móvel nem um *home agent*, o processamento do pacote é feito da seguinte maneira:

```
sub process_packet {
    my $packet=$_[0];

    if ($packet=~/h\[ipv6;[^;]+;[^;]+;[^;]+;[^;]+;[^;]+;[^;]+;([^;\]]+)/) {
                if ($conf{"ipaddr1"} eq $1 or $conf{"ipaddr2"} eq $1) {
```

```
                        myprint("<MENSAGEM>\n");
                } else {
                        route_packet($packet);
                }
        } else {
                myprint("Unknown packet: $packet. Droping it!\n");
                return;
    }
}
```

Figura 4.13: Processamento básico do pacote

A primeira condição `if` vai confirmar se o cabeçalho cumpre o formato *standard* de um pacote. Se não estiver em conformidade então o pacote é descartado, sendo registada nos *logs* a informação "`Unknown packet: $packet. Droping it!`".
No caso do pacote ser destinado ao *router*, na versão mais básica, o equipamento virtual não saberá o que fazer com este pacote e terminará com um erro, representado no código acima com o texto "**<MENSAGEM>**", que no código é traduzido por "`This packet destination was to me... I don't know what to do with it! $packet`". A variável `$packet` contém o conteúdo do pacote descartado e também é apresentada nos *logs*.
Por fim, a função `route_packet($packet);` é a responsável por encaminhar o pacote para o *next hop*.

No caso de ser necessário gerar um ICMP *unreachable*, a função responsável por esta tarefa é a seguinte

```
sub icmpunreach {
    my $packet=$_[0];

    if ($packet=~/h\[ipv6;[^;]+;[^;]+;[^;]+;[^;]+;[^;]+;[^;]+;[^;\]]+;1;/) {
                myprint("<MENSAGEM>\n");
                return;
    } elsif ($packet =~
 /h\[ipv6;[^;]+;[^;]+;[^;]+;[^;]+;([^;]+);[^;\]]+.*\]d\[([^\]]+)\]/) {
                my $msg=encode_base64($packet,"");
                sendpacket($1,
                   "h[ipv6;0;0;0;0;64;$conf{'ipaddr1'};$1;1;0;0;0]d[$msg]");
                return;
        } else {
                myprint("<MENSAGEM>\n");
    }
}
```

Figura 4.14: Geração de ICMP *unreachable*

Neste código são verificadas as condições impostas no RFC que não permitem o envio de um ICMP unreachable em resposta a outro ICMP unreachable. De igual modo, apenas é enviado um pacote ICMP se for possível determinar o endereço IP do emissor. O texto que

aparece quando ocorre um erro foi substituído, no código apresentado acima, por "**<MENSAGEM>**" para não tornar o código muito pesado devido ao longo tamanho destas mensagens.

Todo o envio de pacotes é realizado com recurso ao protocolo UDP, já que este acrescenta um *overhead* desprezável em relação ao protocolo IP. Contudo, as informações de *logging* são enviadas com o protocolo TCP. Esta informação é enviada por um processo separado para não interferir com a recepção de novos pacotes UDP e para que haja garantia na entrega.

4.6.4. *Script* mobsim_1port_udp

Conceptualmente, este *script* é muito semelhante ao anterior. Contudo, o objetivo deste é fornecer o código base para um nó final. Como este código está orientado para os nós finais, apenas se considerou a possibilidade de ter uma interface. Cada interface de rede está associada a uma porta UDP.
Na sua versão base, este dispositivo virtual tem suporte de MIPv6 (RFC 6275), incluindo o procedimento *return routability*.

A inserção de código adicional é feita de uma forma muito similar à anterior.
De igual modo, este equipamento tem a possibilidade de gerar pacotes ICMP em caso de problemas, de acordo com o RFC.

4.7. Utilização do mobSim

A utilização do emulador mobSim pressupõe um estudo prévio que clarifique qual o âmbito do teste. É importante ter a noção do que se pretende observar ou comparar. Posteriormente, é necessário configurar a ferramenta para a sua utilização na forma não supervisionada.
Definiu-se um conjunto de passos que auxiliam a correta utilização desta ferramenta:

1. Definição geral do cenário
2. Configuração dos equipamentos virtuais
3. Configuração dos cenários
4. |Configuração da simulação

4.7.1. Definição geral do cenário

O primeiro passo, **definição geral do cenário**, deve ser realizado pelo utilizador da ferramenta de modo a obter uma ideia clara de quais os parâmetros ou fatores a estudar, quais os testes a realizar e quais os resultados que pretende visualizar. É uma fase preliminar, que vai servir de base para as configurações a executar nos passos seguintes.

Imagine-se que o objetivo de um determinado teste é avaliar a viabilidade das soluções num ambiente de imbricação. A Figura 4.15 ilustra como se poderia configurar um cenário para ajudar a avaliar as soluções de mobilidade imbricada.

Neste ambiente de exemplo são criadas duas redes de topo. Os *routers* de topo `RA` e `RB` interligam, por exemplo, países. Cada *router* de topo tem dois *routers* de segundo nível. A cada *router* de segundo nível, existe uma ligação a apenas um *router* de terceiro nível. Este último é o *home agent* para os *routers* móveis que se encontram ligados a ele.

Este cenário contempla 14 *routers* e 2 *routers* móveis. Os nós finais não são mostrados para não adensar a imagem.

Depois de definido o cenário em termos gerais, o passo seguinte consiste em configurar os equipamentos que irão compor o ambiente de estudo.

Posteriormente, deve-se configurar o cenário de acordo com o desenho do primeiro passo.

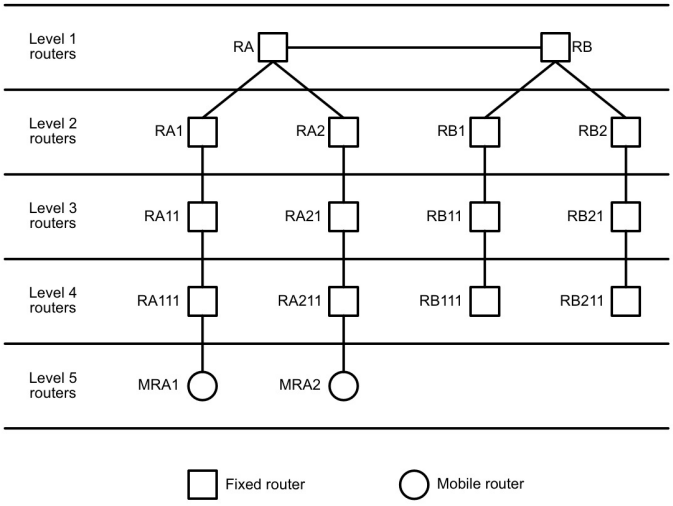


Figura 4.15 – Exemplo de cenário

Finalmente, deve-se configurar a simulação: que equipamentos se movem, qual o tipo de tráfego, qual a intensidade de tráfego, etc.

4.7.2. Configuração dos equipamentos virtuais

A configuração dos equipamentos virtuais é obtida através de ficheiros `.conf` que contêm os diversos parâmetros específicos de cada dispositivo virtual. Estes permitem a definição do endereço IP, o endereço de *gateway*, o servidor de *log*, o nível de *debug*, localização do ficheiro de código adicional, entre outras informações relevantes. Normalmente, estes ficheiros são construídos automaticamente pelo nó controlador.

A Figura 4.16 apresenta um exemplo de uma configuração, neste caso a do *router* `ra111` que é, ao mesmo tempo, o *home agent* do *router* móvel `mra1`.

```
prog=mobsim_2ports_udp.pl
logaddr=LOG:1
debug=0
logfile=/tmp/vapirun/logs/log_ra111
extracode=HA.extracode

ipaddr1=ra111:ra111
ipaddr2=ra11:ra111
gateway=ra11:ra11
routing=mra1::ra111:mra1
```

Figura 4.16 – Configuração do *home agent* ra111

As opções que podem existir no ficheiro de configuração de cada equipamento virtual são:

- **prog**, tipo de programa que será utilizado pelo programa virtual; até ao momento, apenas existem as possibilidades: mobsim_2ports_udp.pl, mobsim_1port_udp.pl, mobsim_2ports_tcp.pl, mobsim_1port_tcp.pl;
- **logaddr**, endereço do servidor de logs e *debug*;
- **debug**, nível de *debug*; quanto maior for o nível, maior é a quantidade de informação fornecida; em produção este valor deve estar a 0 (zero);
- **logfile**, no caso de haver algum problema, o programa utilizará este ficheiro como alternativa para guardar informação útil para *debug*;
- **extracode**, localização do ficheiro com o código que é adicionado ao código inicial definido na variável **prog**;
- **ipaddr1**, endereço IP virtual da primeira interface de rede;
- **ipaddr2**, endereço IP virtual da segunda interface de rede (apenas utilizado em *routers*);
- **gateway**, endereço do *router* responsável por encaminhar os pacotes cujo destino não conste na tabela de *routing* do equipamento virtual;

- **routing**, tabela de *routing* do equipamento virtual; o formato desta tabela é <endereço de rede>::<*next hop*>. Para o exemplo em causa, `mra1::ra111:mra1`, significa que pacotes que tenham como destino a rede `mra1::/64`, deverão ser entregues no equipamento `ra111:mra1`;
- **delay_network**, atraso a ser introduzido, antes de encaminhar o pacote;
- **delay_handoff**, atraso a ser introduzido sempre que houver um *handoff*;
- **delay_dhcp**, atraso a ser introduzido sempre que houver algum pedido de *dhcp*;
- **delay_bu**, atraso a ser introduzido sempre que houver um pedido de *binding update*;
- **delay_ba**, atraso a ser introduzido sempre que houver um pedido de *binding acknowledgement*;
- **delay_rr**, atraso a ser introduzido sempre que se recorrer ao procedimento de *return routability*;
- **delay_hoti**, atraso a ser introduzido sempre que houver um *home test init*;
- **delay_coti**, atraso a ser introduzido sempre que houver um *care-of test init*;
- **delay_hot**, atraso a ser introduzido sempre que houver um *home test*;
- **delay_cot**, atraso a ser introduzido sempre que houver um *care-of test*;
- **delay_mrha**, atraso a ser introduzido sempre que for usado um túnel MRHA;
- **delay_panareauth**, atraso a ser introduzido sempre que houver um pedido PANA *re-authentication*;
- **delay_omen_nd**, atraso a ser introduzido sempre que houver um pedido de *neighbor discovery*;

Existe sempre a possibilidade de criar novos parâmetros que poderão ser utilizados por algum código extra que será criado para alguma solução em particular.

4.7.3. Configuração dos cenários

A configuração dos cenários é feita com recurso a *scripts* com a extensão .scn. Na definição de um cenário são contempladas as questões como quantas redes existem, qual a topologia utilizada, ou onde está cada dispositivo ligado.

A Figura 4.17 ilustra a configuração para o exemplo em estudo neste capítulo.

```
# Network A
toprouter|ra
router|ra1;gateway|ra
router|ra11;gateway|ra1
```

```
router|ra111;gateway|ra11;extracode|HA.extracode
mobilerouter|mra1;gateway|ra111;extracode|mr.extracode
node|mnna1;gateway|mra1;prog|mobsim_1port_udp.pl;extracode|mipv6.extracode

router|ra2;gateway|ra
router|ra21;gateway|ra2
router|ra211;gateway|ra21;extracode|HA.extracode
mobilerouter|mra2;gateway|ra211;extracode|mr.extracode

# Network B
toprouter|rb
router|rb1;gateway|rb
router|rb11;gateway|rb1
router|rb111;gateway|rb11;extracode|HA.extracode

router|rb2;gateway|rb
router|rb21;gateway|rb2
router|rb211;gateway|rb21;extracode|HA.extracode
node|mnnb1;gateway|rb211;prog|mobsim_1port_udp.pl;extracode|mipv6.extracode

LOGS|00logs;prog=mobsim_1port_tcp.pl;ipaddr1=LOG:1;logfile=/tmp/vapirun/logs
/log_log;extracode=logs.extracode;debug=6;logaddr=
```

Figura 4.17 – Exemplo de configuração do equipamentos virtuais

É com base no script de cenário que são criados todos os dispositivos virtuais. Cada linha do ficheiro de configuração do cenário identifica um dispositivo virtual e as respetivas configurações. O cardinal (#) identifica um comentário e é ignorado pelo programa.
O formato de configuração de cada equipamento é:

```
<operação>|<valor>[;<operação1>|<valor1>[...]]
```

Figura 4.18: Formato de configuração dos equipamentos

Os valores possíveis para o campo `operação` são todos os parâmetros que compõem um equipamento, tais como endereço IP, endereço do *gateway*, etc.
É, também, nesta opção que é definido o tipo de equipamento. Os possíveis tipos que atualmente o mobSim suporta são:

- `toprouter`: *router* de topo, geralmente utilizado para interligar grandes redes; este tipo de dispositivo virtual conhece as rotas para as outras redes de topo;
- `router`: um dispositivo virtual fixo com capacidade de encaminhamento de pacotes;
- `mobilerouter`: um dispositivo virtual móvel com capacidade de encaminhamento de pacotes;
- `node`: um equipamento terminal (portátil, sensor, etc.);
- `multiplenode`: permite a criação de múltiplos equipamentos terminais, com as mesmas caraterísticas; este comando é útil para a criação de várias centenas de equipamentos terminais debaixo do mesmo *router* móvel, por exemplo;

- `LOGS`: equipamento central para obtenção de estatísticas para a simulação e relatórios de problemas ou de *debug*.

A linha que configura o *router* de topo `RA` é `toprouter|ra`. O ficheiro de configuração que será criado para este equipamento encontra-se na Figura 4.19.

```
prog=mobsim_2ports_udp.pl
logaddr=LOG:1
debug=0
logfile=/tmp/vapirun/logs/log_ra

ipaddr1=ra:ra
ipaddr2=TOP:1

routing=rb::TOP:2;rb1::TOP:2;rb11::TOP:2;rb111::TOP:2;rb2::TOP:2;rb21::TOP:2
;rb211::TOP:2;ra1::ra:ra1;ra2::ra:ra2;ra1::ra:ra1;ra11::ra:ra1;ra111::ra:ra1
;mra1::ra:ra1;ra2::ra:ra2;ra21::ra:ra2;ra211::ra:ra2;mra2::ra:ra2
```

Figura 4.19 – Configuração do *router* de topo `ra`

A tabela de `routing` é construída automaticamente com todas as redes que existem no cenário em estudo. A rota para cada rede tem como *next hop* o equipamento mais próximo do *router* `ra`.
A título de exemplo, para encaminhar um pacote para a rede do *router* `rb111`, o `ra` sabe que o *next hop* é o endereço IP `TOP:2`, que corresponde ao *router* de topo `rb`.

Outro exemplo é a definição do *router* `ra1`, que é conseguido através do comando `router|ra1;gateway|ra`. Este comando cria o equipamento virtual com o nome `ra1`, contendo o endereço IP do *gateway* igual a `ra`. Deste modo, o mobSim sabe que o equipamento `ra1` está por baixo do `ra`.
Com base nesta linha, o módulo de controlo vai criar o ficheiro de configuração do *router* `ra1` que é ilustrado na Figura 4.20. A tabela de `routing` é construída com as rotas para as redes dos *routers* `ra11`, `ra111` e `mra1`.

```
prog=mobsim_2ports_udp.pl
logaddr=LOG:1
debug=0
logfile=/tmp/vapirun/logs/log_ra1

ipaddr1=ra1:ra1
ipaddr2=ra:ra1
gateway=ra:ra

routing=ra11::ra1:ra11;ra111::ra1:ra11;mra1::ra1:ra11
```

Figura 4.20 – Ficheiro de configuração do *router* `ra1`

A configuração do equipamento terminal `mnna1` consiste na seguinte linha de código,

```
node|mnna1;gateway|mra1;prog|mobsim_1port_udp.pl;extracode|mipv6.extracode
```

Figura 4.21: Configuração do equipamento `mnna1`

O nome `mnna1` é atribuído ao nó terminal com o comando `node`. A identificação do endereço de *gateway* é obtida através do comando `gateway|ra1`.

Dado que é um equipamento com apenas uma interface de rede, deve-se usar o *script* de `prog mobsim_1port_udp.pl`.

O código adicional `mipv6.extracode` identifica que o equipamento tem suporte de MIPv6 para criação de novas ligações otimizadas.

O ficheiro de configuração que é gerado para este equipamento encontra-se na Figura 4.22.

```
prog=mobsim_1port_udp.pl
extracode=mipv6.extracode
logaddr=LOG:1
debug=0
logfile=/tmp/vapirun/logs/log_mnna1

ipaddr1=mra1:mnna1
gateway=mra1:mra1
```

Figura 4.22 – Configuração do nó da rede móvel `mnna1`

É importante notar que se torna possível atribuir diferentes caraterísticas a um determinado equipamento. Por exemplo, para definir que um determinado *router* tem um atraso no processamento de pacotes, pode-se adicionar o parâmetro `delay_network`.

Os outros possíveis comandos incluem `delay_handoff`, `delay_dhcp`, `delay_bu`, `delay_ba`, `delay_rr`, `delay_hoti`, `delay_coti`, `delay_hot`, `delay_cot`, `delay_mrha`, `delay_panareauth`, `delay_omen_nd`.

Existe sempre a possibilidade de adicionar novos comandos e parâmetros que terão impacto ao nível do emulador.

Dado que os pacotes são enviados através da Internet, é necessário que o programa tenha alguma forma de mapear o endereço IP virtual no endereço IP que realmente será utilizado. Esta associação é feita com recurso a uma tabela que é construída pelo módulo de controlo.

Na Figura 4.23 pode ver-se que o endereço IP virtual `ra1:ra1` está associado ao endereço IP `2001:690:2180:120:a00:27ff:fed4:ecfd`, utilizando a porta `20101`.

```
LOG:1          2001:690:2180:120:a00:27ff:fe18:d7c4:1051
ra2:ra21       2001:690:2180:120:a00:27ff:fed4:ecfd:20100
ra1:ra1        2001:690:2180:120:a00:27ff:fed4:ecfd:20101
ra211:mra2     2001:690:2180:120:a00:27ff:fed4:ecfd:20102
rb21:rb211     2001:690:2180:120:a00:27ff:febe:9eb8:20100
```

Figura 4.23 – Lista de associação do endereço IP virtual ao endereço IP usado

Por fim, é necessário garantir que o *routing* está a funcionar corretamente. Para isso, e apenas para validação do emulador, ativou-se a funcionalidade de *debug* e verificou-se o percurso que um pacote toma quando percorre os equipamentos.

A Figura 4.24 mostra o caminho que um pacote percorre desde o equipamento `mnna1` até ao `mnnb1`, para o cenário de exemplo. De modo a não tornar a figura muito densa, foram removidos os cabeçalhos e respetivos conteúdos dos pacotes.

```
[mnna1] Sending to rb211:mnnb1 (final destination)
[mra1] Sending to mra2:mra2
[mra2] Sending to rb111:rb111
[rb111] Sending to rb11:rb11
[rb11] Sending to rb1:rb1
[rb1] Sending to rb:rb
[rb] Sending to TOP:1
[ra] Sending to ra:ra2
[ra2] Sending to ra2:ra21
[ra21] Sending to ra21:ra211
[ra211] Sending to ra21:ra21
[ra21] Sending to ra2:ra2
[ra2] Sending to ra:ra
[ra] Sending to ra:ra1
[ra1] Sending to ra1:ra11
[ra11] Sending to ra11:ra111
[ra111] Sending to ra11:ra11
[ra11] Sending to ra1:ra1
[ra1] Sending to ra:ra
[ra] Sending to TOP:2
[rb] Sending to rb:rb2
[rb2] Sending to rb2:rb21
[rb21] Sending to rb21:rb211
[rb211] Sending to rb211:mnnb1
```

Figura 4.24 – Percurso de um pacote desde o `mnna1` até ao `mnnb1`

Cada linha da figura apresenta uma informação fornecida pelo equipamento referido entre parêntesis retos. Cada equipamento informa para onde envia o pacote. No caso de um nó final, como o `mnna1`, este apresenta o destino final, enquanto cada equipamento de *routing* mostra o próximo passo (*next hop*).

Assim, torna-se possível confirmar que o equipamento `mnna1` pretende enviar um pacote para o `mnnb1`. Este pacote aparece no *router* `mra1`, que é o *default gateway* do `mnna1`, e é posteriormente encaminhado para a interface *ingress* do `mra2` (`mra2:mra2`).

Com esta opção de *debug* é possível verificar qual o percurso que o pacote faz desde que sai do `mnna1` até ao seu destino final.

4.7.4. Configuração das simulações

A configuração das simulações também é feita com recurso a um conjunto de *scripts*, que irão garantir que serão executadas exatamente da mesma maneira para todas as soluções de mobilidade em estudo.

Enquanto os *scripts* de cenário apenas definem as condições estáticas da simulação (isto é, topologia da rede, número e tipos de equipamentos), os *scripts* de simulação definem o comportamento dinâmico, tal como tipo e quantidade de pacotes enviados, intensidade com que são gerados os pacotes, origem e destino dos pacotes, movimento dos nós e *routers* móveis.

A Figura 4.25 apresenta o cenário que irá ser estudado neste exemplo.

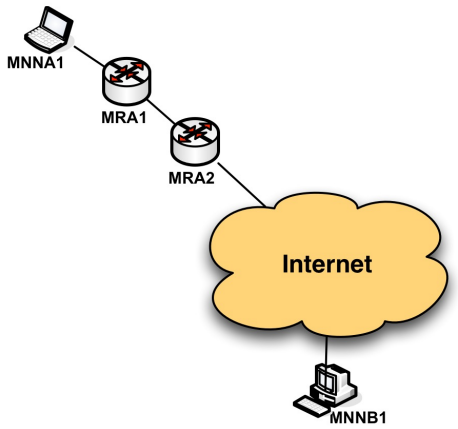


Figura 4.25 – Cenário implementado para efeitos de exemplo

Neste exemplo pretende-se que o *router* móvel `MRA1` se encontre numa situação de imbricação de nível 1, dentro da rede do *router* móvel `MRA2`. Neste contexto, vai-se confirmar que o equipamento `MNNA1` consegue enviar um PING para o nó correspondente `MNNB1`.

Neste cenário considera-se que o *router* `MRA2` se encontra numa rede estrangeira, isto é, em situação de mobilidade. Se assim não fosse, o *router* `MRA1` não se encontraria numa situação de imbricação mas numa rede estrangeira simples. Por outro lado, considera-se que o `MRA2` criou já os túneis MRHA necessários para ter conectividade, antes do `MRA1` imbricar nele. Apenas após a conclusão com sucesso deste passo é que o `MRA1` pode criar o seu próprio

túnel MRHA. Com estes pressupostos, pode então realizar-se o teste de conectividade com o `MNNB1`.

A Figura 4.26 ilustra o ficheiro de configuração .scr do cenário de simulação discutido.

```
scenario;example
paradigm;nemo
paradigm;miron
paradigm;omen


# MRA2 network move
move;ra211:mra2;rb111:mra2
waitmove;40

# MRA2 network move
move;ra111:mra1;mra2:mra1
waitmove;40

# Ping test
ping;1;100000;mra1:mnna1;rb211:mnnb1
waitping;40
ping;1;100000;mra1:mnna1;rb211:mnnb1
waitping;40
```

Figura 4.26 – Configuração da simulação do exemplo em estudo

A primeira linha identifica o cenário a ser utilizado. No nosso caso, o cenário em uso designa-se `example`. De seguida, indicam-se quais as soluções que se pretendem estudar. No exemplo, serão estudados os `nemo`, `miron` e `omen`. Estas são as linhas que determinam a configuração do cenário estático que será utilizado pelo emulador. As linhas seguintes indicam a componente dinâmica da simulação.

O comando `move;ra211:mra2;rb111:mra2` indica que o *router* `mra2`, com o endereço IP `ra211:mra2` deve mover-se para baixo do *router* `rb111`. Aí, ele deverá usar o endereço IP `rb111:mra2`.

A simulação apenas deve continuar após este movimento ter sido concluído. Esta garantia é conseguida com o comando `waitmove;40`, que indica que deve esperar que o *router* `mra2` informe que concluiu o procedimento de *binding update*.

De seguida, o *router* `mra1` vai passar para dentro da rede móvel do `mra2` e adquirirá o endereço IP `mra2:mra1`.

Por fim, é enviado um PING com o comando `ping;1;100000;mra1:mnna1;rb211:mnnb1`, que indica que deve ser enviado um pacote do tipo `PING`, do endereço `mra1:mnna1` para o endereço `rb211:mnnb1`. O valor `100000` indica que deve haver um intervalo de 100ms entre o primeiro PING e o segundo. Como apenas é enviado um pacote, esta opção é ignorada.

O *script* vai esperar pela correta recepção do pacote antes de enviar outro PING. Depois de esperar pela recepção do segundo pacote, a simulação termina.

Sempre que o simulador é executado é apresentado um conjunto de relatórios do género do que se pode visualizar na Figura 4.27. Esta informação é gerada com base no *script* de configuração da simulação.

```
Cleaning:
       2001:690:2180:120:a00:27ff:fed4:ecfd
       2001:690:2180:120:a00:27ff:febe:9eb8
       2001:690:2180:120:a00:27ff:fe18:d7c4
------------------------------------------------------
Starting simulation for script ../confs/scripts/example.scr!
Scenario: example
Paradigms: nemo; miron; omen;
Servers:
       2001:690:2180:120:a00:27ff:fed4:ecfd;
       2001:690:2180:120:a00:27ff:febe:9eb8;
       2001:690:2180:120:a00:27ff:fe18:d7c4;
------------------------------------------------------
Generating IPs: Done!
       Nr of hosts per server:
              2001:690:2180:120:a00:27ff:fed4:ecfd:   8
              2001:690:2180:120:a00:27ff:febe:9eb8:   10
       Total number of hosts: 18
       Finding nodes: 3 Done!
       Finding ports: 34 Done!
------------------------------------------------------
starting controler daemons: DONE


======================================================================
= Starting paradigm nemo
======================================================================
! Starting simulator: 18 hosts with 34 ports: [0][34]Done!
! Actions:
nemo]    + move;ra211:mra2;rb111:mra2
nemo]    + waitmove;40:      Waiting for 1 moves: [0][1]Done!
nemo]    + move;ra111:mra1;mra2:mra1
nemo]    + waitmove;40:      Waiting for 2 moves: [1][2]Done!
nemo]    + ping;1;100000;mra1:mnna1;rb211:mnnb1
nemo]    + waitping;40:      Waiting for 1 pings: [1]Done!
nemo]    + ping;1;100000;mra1:mnna1;rb211:mnnb1
nemo]    + waitping;40:      Waiting for 2 pings: [2]Done!
- Shutting down the nemo paradigm. Waiting for 18 hosts: [0][18]Done!


======================================================================
= Starting paradigm miron
======================================================================
! Starting simulator: 18 hosts with 34 ports: [0][34]Done!
! Actions:
miron]    + move;ra211:mra2;rb111:mra2
miron]    + waitmove;40:      Waiting for 1 moves: [0][1]Done!
miron]    + move;ra111:mra1;mra2:mra1
miron]    + waitmove;40:      Waiting for 2 moves: [1][1][2]Done!
miron]    + ping;1;100000;mra1:mnna1;rb211:mnnb1
miron]    + waitping;40:      Waiting for 1 pings: [1]Done!
miron]    + ping;1;100000;mra1:mnna1;rb211:mnnb1
miron]    + waitping;40:      Waiting for 2 pings: [2]Done!
- Shutting down the miron paradigm. Waiting for 18 hosts: [0][18]Done!


======================================================================
= Starting paradigm omen
======================================================================
! Starting simulator: 18 hosts with 34 ports: [0][34]Done!
! Actions:
omen]    + move;ra211:mra2;rb111:mra2
```

```
omen]    + waitmove;40:      Waiting for 1 moves: [0][1]Done!
omen]    + move;ra111:mra1;mra2:mra1
omen]    + waitmove;40:      Waiting for 2 moves: [1][2]Done!
omen]    + ping;1;100000;mra1:mnna1;rb211:mnnb1
omen]    + waitping;40:      Waiting for 1 pings: [1]Done!
omen]    + ping;1;100000;mra1:mnna1;rb211:mnnb1
omen]    + waitping;40:      Waiting for 2 pings: [2]Done!
- Shutting down the omen paradigm. Waiting for 18 hosts: [0][18]Done!

------------------------------------------------------
Stopping the overall system (including myself!):
```

Figura 4.27 – Resultado da simulação

Pode-se observar que é feita uma análise para verificar a consistência das configurações e instruções de simulação. Se estiver tudo em ordem, o mobSim vai avaliar a distribuição das máquinas virtuais pelos servidores disponíveis. No nosso exemplo, não é permitido nenhum equipamento virtual no servidor principal.

De seguida, o emulador realiza os diversos testes de acordo com a configuração do ficheiro de simulação e termina instruindo todos os equipamentos virtuais que devem parar a execução.

Os resultados das operações de PING são obtidos nos ficheiros de LOG centralizado. A Figura 4.28 mostra um exemplo de um resultado para o caso do NEMO.

```
1344098198.22418 [mra2] changing IP from ra211:mra2
                        to rb111:mra2 gw rb111:rb111
1344098199.15255 [mra2] MRHA active from ra211:ra211 to ra211:mra2
1344098200.25432 [mra1] changing IP from ra111:mra1
                        to mra2:mra1 gw mra2:mra2
1344098201.27376 [mra1] MRHA active from ra111:ra111 to ra111:mra1
1344098205.63706 [mnna1] RES 1-PING_mra1:mnna1_rb211:mnnb1: 0.20
1344098205.95269 [mnna1] RES 2-PING_mra1:mnna1_rb211:mnnb1: 0.19
```

Figura 4.28 – Resultado do PING usando o NEMO

Neste exemplo, pode-se verificar que o equipamento `mra2` iniciou a mobilidade da sua rede origem (`ra211:mra2`) para a rede estrangeira (`rb111`) no instante `1344098198.22418`. No instante `1344098199.15255` o túnel MRHA encontrava-se concluído.

É também possível verificar que os dois PINGs foram executados (`1-PING` e `2-PING`) e que demoraram exatamente `0,20` e `0,19` segundos.

4.8. Conclusão

Neste capítulo foi apresentada a ferramenta de emulação de mobilidade de redes mobSim, especialmente desenvolvida para o estudo das soluções de mobilidade alvo da presente tese. Como características chave do mobSim referem-se a sua escalabilidade, a conformidade com mecanismos normalizados, o seu nível de detalhe e a sua versatilidade. Todos estes aspetos foram abordados no presente capítulo.

Por se tratar de uma ferramenta de características ímpares, o mobSim é uma das contribuições relevantes do presente trabalho, permitindo não só o estudo comparativo dos três paradigmas de mobilidade ao nível da camada IP, mas também uma análise muito aprofundada das características, comportamento, potencial e problemas de cada um desses paradigmas, quando encarados individualmente.

Esta ferramenta está na base da avaliação da proposta OMEN, apresentada no capítulo seguinte.

5. Avaliação do OMEN

A avaliação da proposta OMEN foi feita de quatro perspectivas distintas. Inicialmente, foi efetuada uma avaliação preliminar num cenário de pequena dimensão, tendo em vista uma validação do ponto de vista funcional e uma confirmação das principais características esperadas para a proposta em face das opções arquiteturais que ela comporta, em particular a opção pelo paradigma de mobilidade com base no cliente. Apesar da reduzida dimensão do cenário, os resultados obtidos apontaram para o facto de o OMEN ser uma solução bastante leve, como esperado, e com vantagens do ponto de vista de desempenho.

No seguimento destes resultados decidiu-se avançar para uma comparação de muito grande dimensão dos paradigmas existentes, na ordem das dezenas de milhar de equipamentos finais e redes. Com esta avaliação pretendeu-se aferir o potencial do OMEN para funcionamento numa Internet que é cada vez mais móvel e que, num futuro que se antevê não muito distante, incluirá largos milhares de redes móveis. Dada a dimensão do cenário, os testes foram realizados com recurso ao emulador mobSim, executado no *cluster* Milipeia[10]. Os resultados desta avaliação permitiram confirmar as diferenças de desempenho entre a proposta OMEN e as soluções de mobilidade que se enquadram nos outros dois paradigmas. Não obstante a clareza dos resultados, prosseguiu-se a análise da solução OMEN na perspetiva do stress introduzido por tráfego intensivo, tal como videoconferência, transferência de ficheiros ou semelhantes, num cenário de dimensão razoável. Naturalmente, o estudo contemplou a comparação com soluções de mobilidade de referência. Uma vez mais, a proposta do paradigma baseado no cliente final apresentou resultados claramente positivos.

Os estudos comparativos acima referidos foram concebidos e executados de modo a que os cenários fossem o mais neutros possível. Nesse sentido, foram evitados quaisquer pontos de estrangulamento na rede ou a introdução de variáveis que distorcessem a eficiência de uma solução em detrimento de outra. Por outro lado, a análise das soluções foi realizada de uma forma agnóstica em relação às tecnologias de redes. Por este motivo, foram sempre utilizadas redes físicas cabladas, a 1Gbps, com baixa probabilidade de perda de pacotes. O

[10] O *cluster* Milipeia é constituído por 130 nós de computação Sun Fire X4100 com o sistema operativo CentOS. O *site* do Laboratório de Computação Avançada (www.lca.uc.pt) contém informação relativa a este *cluster*.

objetivo desta aproximação foi o de assegurar que as diferenças de resultados entre os diversos paradigmas em estudo (por exemplo, em termos de atraso de trânsito ou perdas de pacotes) decorressem apenas das características arquiteturais intrínsecas de cada um.

A última perspectiva de avaliação contemplou a utilização de ligações reais sem fios nos cenários de emulação. Neste caso, os pacotes gerados pelo emulador mobSim foram enviados através de redes Wi-Fi sempre que tinham por origem ou destino um sistema móvel. Neste tipo de avaliação, foram analisadas as soluções de mobilidade de redes em cenários de carga e dimensão moderadas.

Nas secções seguintes são detalhados os quatro tipos de avaliações referidos.

5.1. Avaliação preliminar

Os principais objetivos da avaliação preliminar foram, por um lado, a obtenção de dados que apontassem para uma confirmação das melhorias esperadas teoricamente para a proposta OMEN, decorrentes da sua arquitetura, e, por outro, identificar os aspetos a explorar e detalhar em avaliações subsequentes. Com este objetivos, foram realizados testes comparativos de desempenho envolvendo as soluções NEMO Basic Support Protocol, MIRON e OMEN.

Para a realização desta avaliação foi construído um pequeno simulador em Perl, que implementa os aspetos principais das soluções em estudo. Embora extremamente flexível e leve, este simulador não fornece valores absolutos, sendo o seu objetivo o de obter valores relativos, adequados apenas para efeitos de comparação das soluções em causa.

As subseções seguintes apresentam os resultados dos testes, que se encontram divididos em dois cenários diferentes: um cenário sem imbricação e outro cenário com imbricação. Ambos os cenários são de pequena dimensão. Cada teste analisa o *round trip time* (RTT) desde o nó da rede móvel (MNN), passando pelo seu nó correspondente (CN) e terminando no MNN. Cada bloco de simulações é composto de 600 testes individuais, cada um enviando 400 pacotes, a que corresponde um envio de um total de 240.000 pacotes.

5.1.1. Cenário sem imbricação

No cenário sem imbricação, estudou-se o comportamento das soluções NEMO, MIRON e OMEN, considerando que a rede móvel se desloca de uma rede distante para redes sucessivamente mais perto do nó correspondente.

A Figura 5.1 ilustra o teste realizado, em que se pode verificar que o salto 1 e o 2 fazem o MR aproximar-se do nó correspondente CN. No salto número 3 o *router* móvel fica a ser

servido pelo mesmo *router* do nó correspondente. O *router* `net1000` é o *home agent* do *router* móvel `MR`.

Neste cenário, o conceito de "distância" é implementado através dos atrasos introduzidos na entrega dos pacotes. Deste modo, o *router* `net1000` atrasa a entrega dos pacotes que por ele atravessam em 1000μs, enquanto o *router* `net500` atrasa 500μs, e por aí adiante.

Assim, se um *router* móvel com suporte de NEMO Basic Support Protocol estiver debaixo do *router* `net500`, e o nó da rede móvel `MNN` enviar um pacote destinado ao nó correspondente `CN`, este pacote terá que atravessar os *routers* `net500`, `net1000` e `net100`, respetivamente, fazendo com que o pacote demore `500μs + 1000μs + 100μs + tempo de processamento do pacote por cada equipamento virtual` até atingir o destino.

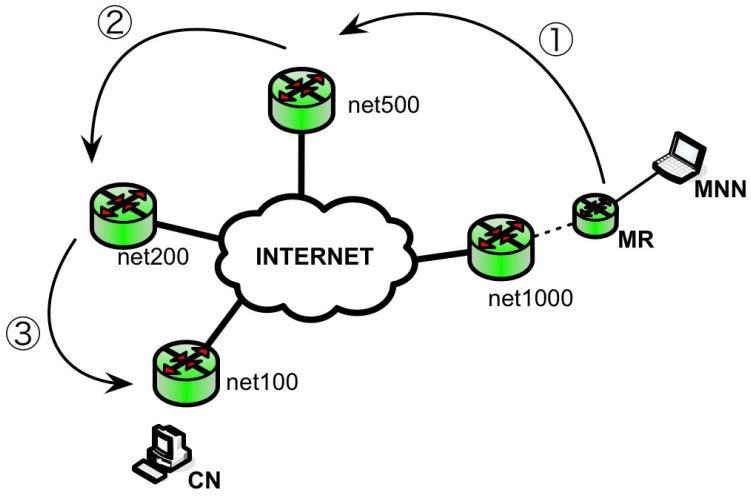


Figura 5.1 – Movimento NEMO para o cenário sem imbricação

A simulação começa com o nó da rede móvel `MNN` a enviar 100 pacotes, enquanto o *router* móvel `MR` está na rede origem. Após o envio dos 100 pacotes, a rede móvel desloca-se para a rede do *router* `net500`, e o `MNN` envia mais 100 pacotes. Sucede o mesmo nos saltos para as redes `net200` e `net100`. Na rede `net100`, o `MR` está na mesma rede que o nó correspondente `CN`.

A Figura 5.2 apresenta os tempos de *round trip time* obtidos.

Como se pode verificar na figura, conforme o *router* móvel com suporte de NEMO Basic Support Protocol se afasta da sua rede origem, maior é o seu atraso na entrega do pacote. Devido ao problema da triangulação, o tempo necessário para um pacote percorrer o

caminho desde o nó móvel até ao nó correspondente e voltar pode demorar até 700 milissegundos, quando o *router* móvel está na rede `net100`.

Já o MIRON e o OMEN conseguem uma melhoria no desempenho à medida que se aproximam da rede do nó correspondente, dado que implementam mecanismos de otimização de rotas.

Neste cenário bastante simplificado não é possível verificar qualquer diferença entre a proposta MIRON e OMEN.

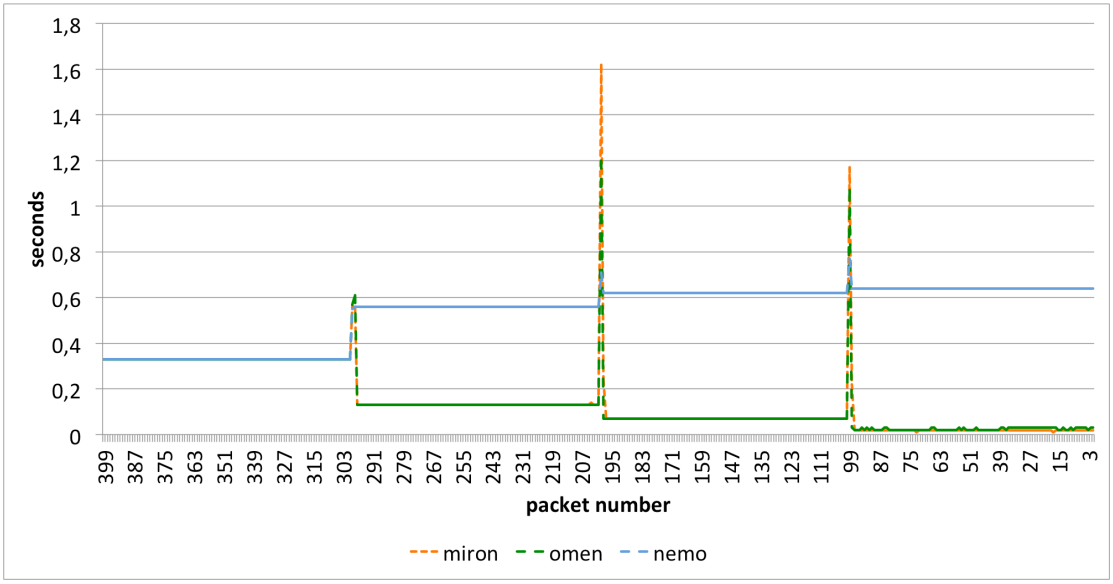


Figura 5.2 – Round trip time entre o MNN e o CN (cenário não imbricado)

A Figura 5.3 apresenta os tempos de *handoff* de cada solução. Os tempos de *handoff* analisados neste estudo dizem respeito ao período que vai desde a mudança do *router* móvel MR para a nova rede, aquisição do novo IP, realização das operações de *binding update*, criação de túneis MRHA, e otimização de rotas, quando aplicável. Apenas quando todas as instruções relacionadas com a mobilidade de redes estão concluídas é que é parado o contador do tempo necessário de *handoff*.

Pela figura é possível verificar que o tempo de *handoff* do NEMO é sempre inferior ao das restantes soluções. Este facto é lógico, já que as restantes soluções precisam de ter conectividade à rede origem através do NEMO Basic Support Protocol para estabelecerem as otimizações de rotas.

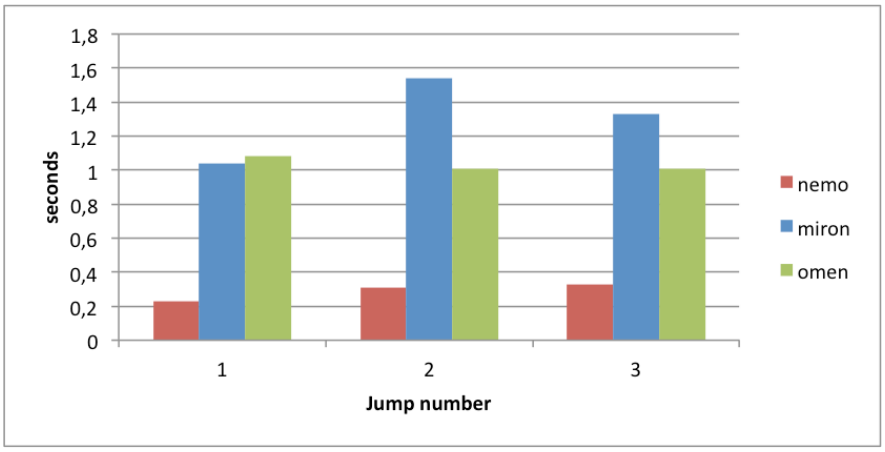


Figura 5.3 – Tempo de *Handoff* (cenário não imbricado)

5.1.2. Cenário com imbricação

Para o cenário com imbricação apenas foram comparados o NEMO com o OMEN. Tal deveu-se a limitações do simulador, que não implementa o suporte de imbricação no caso do MIRON devido à complexidade envolvida. Dado que posteriormente foram feitos testes mais detalhados de comparação que incluíram o MIRON com recurso ao emulador mobSim, optou-se por não refazer este conjunto de testes.

A Figura 5.4 apresenta o cenário utilizado na simulação em ambiente com imbricação.

Como se pode verificar, a rede utilizada para os testes foi sempre a `net400`, tendo um atraso de 400μs. Os *routers* móveis `MR200`, `MR300`, `MR500` têm como *home agents* os *routers* `net200`, `net300` e `net500`, com os respetivos atrasos de 200μs, 300μs e 500μs.

Da mesma forma que nos testes anteriores, foram enviados 100 pacotes pelo nó móvel `MNN`, sendo posteriormente efetuado o salto número `1`. De seguida foram enviados mais 100 pacotes e efetuou-se o segundo salto. O último salto processa-se da mesma maneira.

O primeiro salto colocou o *router* móvel `MR` num estado imbricado de nível 1. Os saltos 2 e 3 fazem com que o `MR` fique com níveis 2 e 3 de imbricação, respetivamente.

Neste estudo pretendeu-se comparar o OMEN com o NEMO Basic Support Protocol, no que respeita ao comportamento em ambientes imbricados. Em testes subsequentes foi possível verificar de forma inequívoca a ineficiência do protocolo NEMO Basic Support Protocol para cenários com imbricação, através de mais testes realizados no âmbito desta tese bem como através de trabalhos de outros autores.

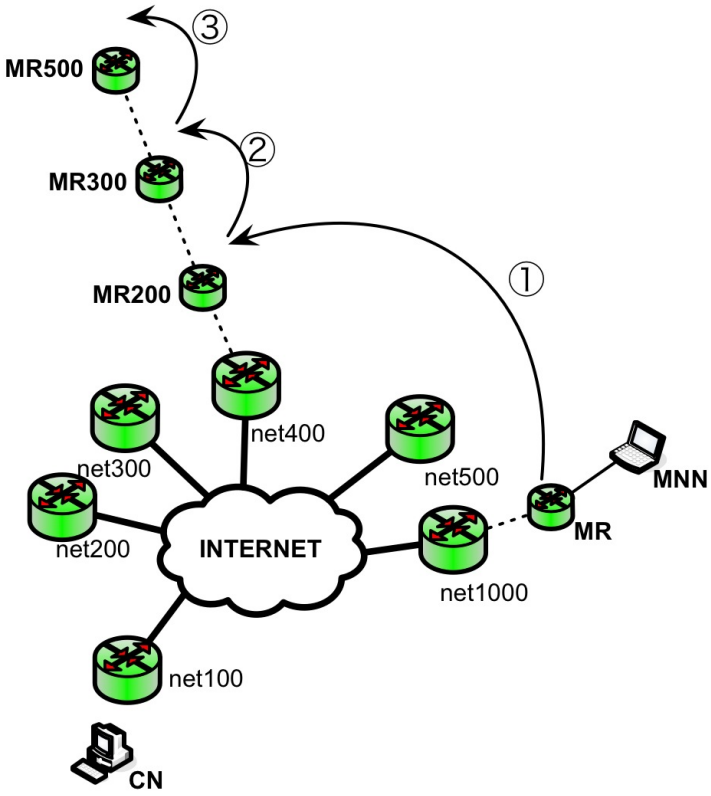


Figura 5.4 – Movimento NEMO num ambiente imbricado

A Figura 5.5 apresenta os tempos de *round trip time* dos pacotes do nó de rede móvel MNN para o nó correspondente CN.

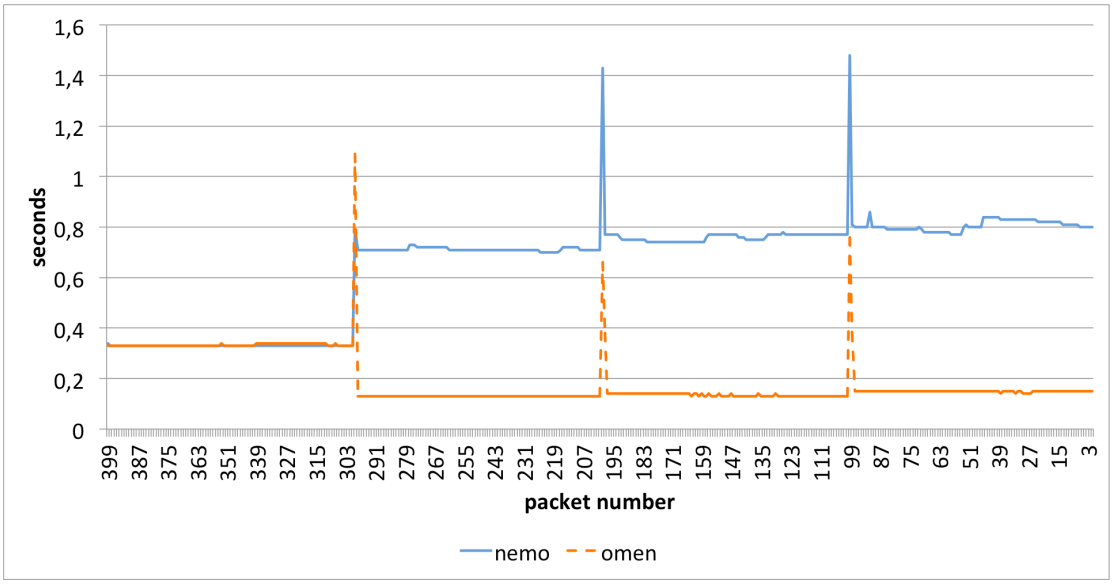


Figura 5.5 – Round trip time entre o MNN e o CN (cenário imbricado)

Conforme se pode verificar, o desempenho do NEMO Basic Support Protocol vai piorando à medida que aumenta o nível de imbricação.

Por outro lado, o OMEN consegue lidar bem com a imbricação, não sendo particularmente afetado por ela. Dado que com esta solução existe otimização de rotas, não há necessidade de percorrer os diversos *home agents* de cada *router* móvel. Assim, o tempo necessário para percorrer o caminho entre o nó da rede móvel e o nó correspondente é apenas ditado pelo caminho mais curto entre os dois.

A Figura 5.6 apresenta os tempos de *handoff*.

Para efeitos de comparação, resolveu-se separar o tempo de *handoff* do *router* móvel `MR` – identificado no gráfico como `omen`, e o tempo necessário para estabelecer a rota otimizada entre o `MNN` e o `CN` – identificado como `omen-RO`.

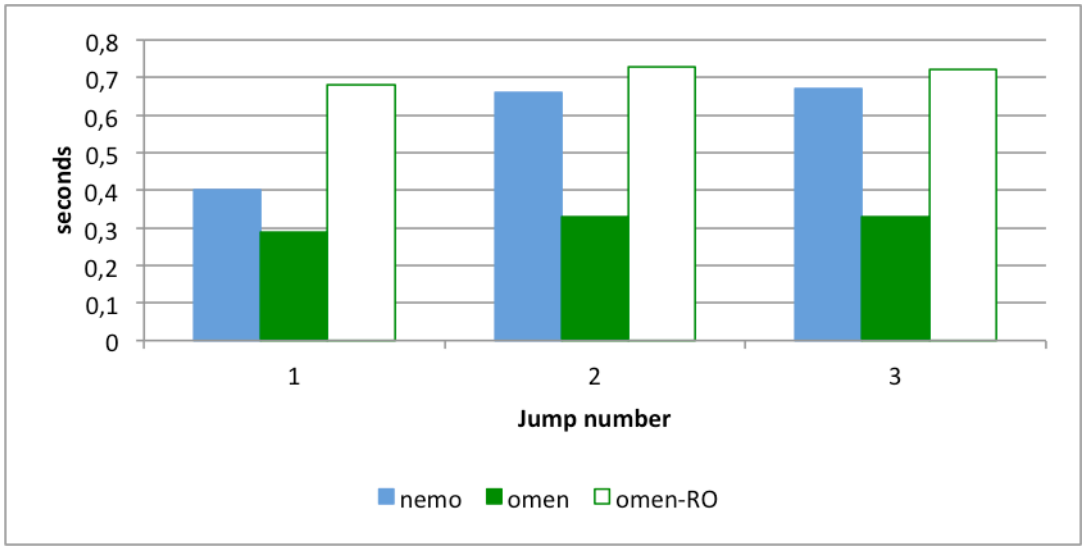


Figura 5.6 – Tempo de *handoff* e tempo de otimização de rota (cenário imbricado)

Como se pode verificar, o tempo de *handoff* para o NEMO vai piorando conforme se vai incrementando o nível de imbricação. Por outro lado, o OMEN mal sofre qualquer alteração. De igual modo, a otimização de rotas também não é particularmente afetada pelo nível de imbricação no OMEN.

5.1.3. Perspetivas decorrentes da avaliação preliminar

Nesta fase de avaliação, o OMEN foi comparado com duas soluções chave para a mobilidade de redes – NEMO Basic Support Protocol e MIRON – em dois cenários simplificados. Mais do que obter resultados relativos aos detalhes das soluções, a ideia foi a de confirmar a intuição inicial, por forma a decidir sobre o potencial da proposta.

Os resultados alcançados apontaram para um bom potencial de melhoria, pelo que se decidiu passar a fases de avaliação mais fina e com recurso a uma ferramenta mais poderosa.
Os testes a realizar nessas fases deverão ser orientados para a avaliação das potenciais vantagens do OMEN, decorrentes dos seguintes aspetos chave desta proposta:

- A otimização de rotas é estabelecida entre os nós da rede móvel e os seus nós correspondentes, sendo que o *router* móvel funciona como um equipamento de *routing*, sem requisitos específicos tanto ao nível protocolar como de desempenho;
- As decisões de otimização de rotas são tomadas pelos nós da rede móvel, quando e apenas se necessitarem de otimização;
- Como cada nó da rede móvel pode agir por sua própria iniciativa, a possibilidade de ocorrer um estrangulamento é inferior à de um cenário em que o *router* móvel otimize as rotas de todos os fluxos de todos os seus MNN;
- Como a otimização de rotas é realizada utilizando o *care-of address* CoA, isto é, um endereço IP topologicamente correto, não há qualquer necessidade de atravessar os *home agents* e, deste modo, os pacotes não estão sujeitos à triangulação.

É interessante relembrar que o OMEN satisfaz muitos dos requisitos e funcionalidade desejáveis definidas no [Ng08], nomeadamente, os requisitos de baixo consumo de processamento (Req2), segurança (Req3), otimização de rotas MR-para-MR (Des1), Otimização de rotas em ambientes imbricados (Des2) e comunicação intra-imbricado (Des3), pelo que se espera que isso se reflita no melhor desempenho da proposta quando comparada com outras soluções.
De forma resumida, o OMEN leva a *routers* mais leves, não requer alterações aos protocolos existentes e tem melhor desempenho que o NEMO Basic Support Protocol. Para além disso, o OMEN não tem, teoricamente, limites de nível de imbricação.

5.2. Avaliação em cenários de muito grande dimensão

Os paradigmas de otimização de rotas considerados neste estudo estão em sintonia com os que foram apresentados no fim do capítulo 2, que são: paradigma centrado nos equipamentos antigos, paradigma baseado na rede e paradigma baseado no cliente final.
O objetivo principal do paradigma centrado nos equipamentos antigos (*legacy-compatible*, LC) é permitir a implementação imediata da mobilidade de redes sem necessidade de alteração

dos equipamentos finais. A referência para este paradigma é o NEMO Basic Support Protocol.

No caso do paradigma baseado na rede (infrastructure-centric, IC) as operações de mobilidade de redes são executadas pelos elementos da infraestrutura, tais como *routers* móveis, *home agents* e entidades correspondentes. A ideia subjacente a este paradigma é que os equipamentos finais sejam mantidos tão inalterados quanto possível e beneficiando da otimização de rotas, embora com o incremento de complexidade ao nível da infraestrutura de rede. Apesar de existirem várias propostas para este paradigma, foram consideradas neste estudo as soluções ORC e MIRON, por serem representativas.

Por fim, no caso do paradigma baseado no cliente final (end-node-centric, EC) os equipamentos finais tomam conhecimento da sua condição de mobilidade. Isto significa que os nós da rede móvel passam a ter parte ativa nas operações de mobilidade, tais como otimização de rotas, aliviando os *routers* móveis desta tarefa. O OMEN enquadra-se neste paradigma.

O paradigma baseado no cliente final pretende eliminar várias problemas inerentes aos paradigmas LC e IC. Um dos problemas associado ao paradigma IC é decidir quando se deve otimizar a rota, ou não, já que é reconhecido que nem todo o tráfego necessita ser otimizado. De facto, em alguns casos a otimização de rotas pode ser contraproducente, já que o processo pode demorar mais tempo que o trânsito do fluxo em si, como é o caso da comunicação DNS. Outro problema está relacionado com o impacto introduzido nos *routers* móveis, que pode ser excessivo se este equipamento tiver que realizar todas as operações de otimização de rotas em vez dos nós da rede móvel. Para redes móveis de pequena dimensão este impacto pode ser aceitável, mas para redes de maior dimensão o esforço pode ser excessivo para os *routers* móveis.

Embora a capacidade de processamento dos equipamentos de *routing* tenha sido alvo de melhorias significativas ao longo dos tempos, a condição de mobilidade implica atenções redobradas às questões de eficiência energética e de processamento excessivo em pontos chave da cadeia de mobilidade de redes.

A vantagem do paradigma baseado no cliente é que os nós finais podem, por si só, tomar a decisão de quando realizar a operação de otimização de rotas, ou não, em função das suas necessidades, e não estarem sujeitos a uma otimização de rotas indiferenciada, generalista e global. Naturalmente, as potenciais vantagens do paradigma EC serão apenas visíveis e significativas em cenários de mobilidade generalizada, tal como se prevê que venha a ser a Internet do futuro, na qual uma maior percentagem de equipamentos será móvel. Assim, torna-se essencial estudar e comparar o comportamento dos três paradigmas de mobilidade de redes em ambientes de larga escala.

Dada a dimensão dos cenários de avaliação envolvidos, os testes apresentados só puderam ser realizados com recurso a simulação/emulação e utilizando um *cluster* com grande capacidade de processamento.

Em concreto, a comparação dos paradigmas LC, IC e EC em larga escala só foi possível com recurso ao emulador mobSim, que utilizou as capacidades do ambiente de processamento paralelo existente no *cluster* Milipeia para permitir extrair as conclusões do comportamento dos diversos paradigmas em termos de *round trip time*, tempos necessários para otimização de rotas e sinalização adicional.

O emulador utilizou o RFC 3963, NEMO Basic Support Protocol, como solução de paradigma centrado nos equipamentos antigos. O paradigma baseado na rede foi baseado na proposta MIRON e, para os problemas que esta proposta não resolve, foi implementada a solução ORC. Para o paradigma baseado no cliente foi utilizada a proposta OMEN.

As subseções seguintes apresentam a configuração dos testes e os respetivos resultados.

5.2.1. Cenários de simulação

A Figura 5.7 apresenta a topologia base da rede global utilizada em todos os cenários de simulação deste conjunto de testes.

Esta topologia base é constituída por redes fixas e móveis. As redes móveis podem alterar o seu ponto de ligação à rede no decurso das simulações. As situações de mobilidade podem ocorrer em ambientes com ou sem imbricação. Para simplificação da imagem, a rede é representada apenas pelos seus *routers* de fronteira. Ou seja, os equipamentos finais não são apresentados.

A rede global consiste numa estrutura em forma de árvore com sete níveis. O primeiro nível da hierarquia consiste em 50 *routers* de topo. Cada *router* de topo está diretamente ligado aos outros 49 *routers* de nível 1. Da mesma forma que os *routers* de topo na Internet, estes possuem rotas para as todas as gamas de endereços IP em uso, tendo sempre como *next hop* outro *router* de nível 1.

Cada um destes *routers* está ligado a 5 *routers* de nível 2. Por seu lado, cada um dos *routers* de nível 2 está ligado a 3 *routers* de nível 3, e por aí adiante, conforme se pode ver na figura.

Com esta configuração, existem 456 *routers* debaixo de cada *router* de topo. Adicionalmente, o número de equipamentos finais debaixo de cada *router* de nível 1 foi configurado para variar entre 225 e aproximadamente 54.000. No final, o cenário simulado era composto por 22.800 *routers*, 11.250 redes, e mais de 27.000 equipamentos finais (fixos e móveis).

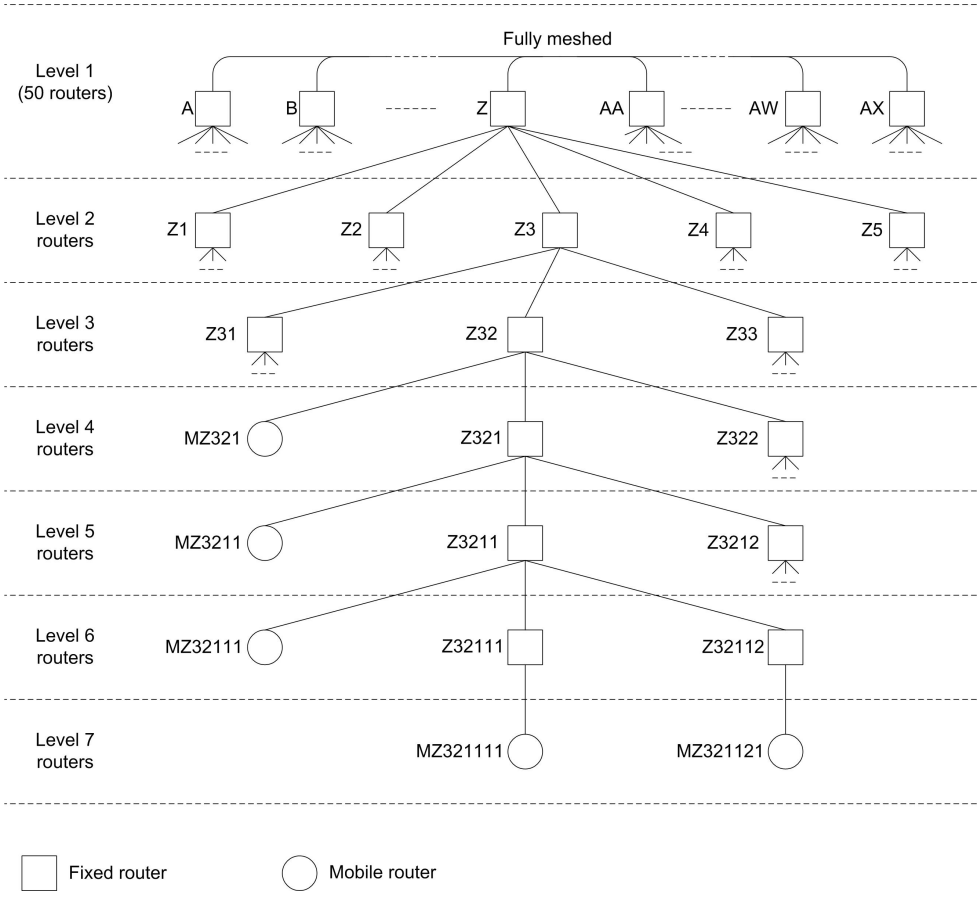


Figura 5.7 – Topologia base da rede global

Foram criados dezassete cenários de simulação. No primeiro cenário não foi utilizado qualquer nível de imbricação, ou seja, todas as redes móveis estavam diretamente ligadas às redes fixas. Para os cenários 2 a 17, foram utilizados níveis de imbricação desde 1 a 16 respetivamente.

Em todos os cenários, até 1.000 nós de rede móvel podiam transmitir e/ou receber tráfego em simultâneo, embora todos os equipamentos finais estivessem ativos e com atividade esporádica no decorrer dos testes.

Cada equipamento móvel transmitiu aproximadamente 140 pacotes. Exatamente as mesmas condições e comportamentos foram aplicados para cada um dos paradigmas em estudo.

Adicionalmente, foram usados diversos parâmetros de atraso em todas as simulações, com os valores indicados na Tabela 5.1.

Estes atrasos pretendem representar o tempo necessário para realizar as várias ações associadas à mobilidade de redes. Os valores apresentados são aproximados aos valores medidos num protótipo laboratorial.

Parâmetro	Descrição	Valor
delay_dhcp	Atraso DHCP	300 ms
delay_rr	Atraso *return routability*	200 ms
delay_hoti	Atraso *home Test init message*	100 ms
delay_coti	Atraso *care-of Test init message*	100 ms
delay_hot	Atraso *home Test message*	100 ms
delay_cot	Atraso *care-of Test message*	100 ms
delay_handoff	Quando o nó muda o seu ponto de ligação à rede, tempo que medeia a perda e conectividade ao nível da camada de *link*	500 ms
delay_mrha	Atraso para configurar o túnel MRHA	10 ms
delay_ba	Atraso *binding acknowledgement*	10 ms
delay_bu	Atraso *binding update*	10 ms

Tabela 5.1 – Valores dos parâmetros de atraso

Outro aspeto importante das simulações, para o paradigma baseado no cliente, EC, foi a utilização de diferentes rácios (ratios) de número de equipamentos finais que não realizam a operação de otimização de rotas em função ao número de nós que realizam esta operação. Na Figura 5.8 encontram-se exemplificados vários rácios.

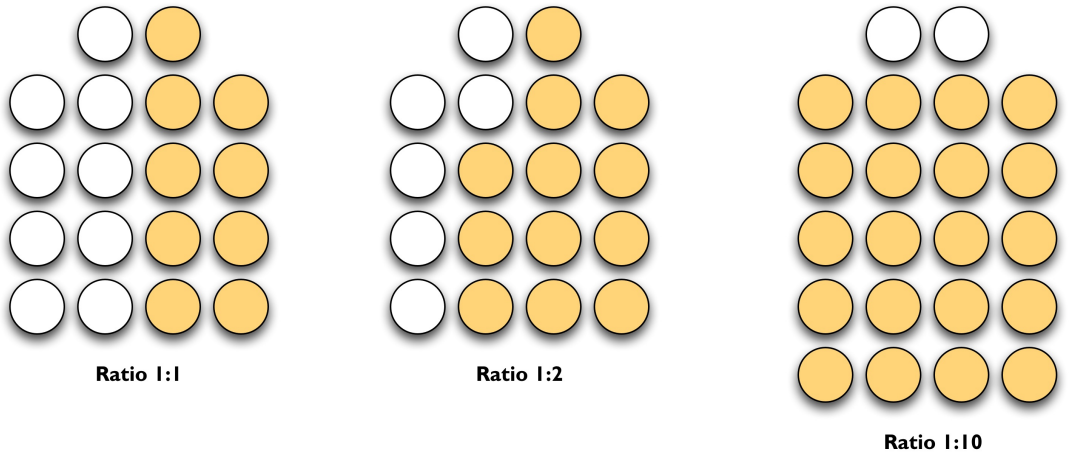


Figura 5.8 – Exemplos de rácios

A título de exemplo, um rácio de 1:2 significa que para cada nó que não realiza a operação de otimização de rotas existem dois equipamentos que irão realizar esta operação.

É importante notar que, por definição, o paradigma centrado no equipamento antigo, LC, e o paradigma baseado na rede, IC, não estão sujeitos a este parâmetro já que para o primeiro não há qualquer otimização de rotas e para o segundo o procedimento de otimização de rotas é realizado para todos os equipamentos, quer o tráfego necessite ou não dessa funcionalidade.

Este fator tem, obviamente, impacto ao nível da quantidade de sinalização necessária e do desempenho do *router* móvel.

Finalmente, é importante notar que as simulações tiveram por objetivo obter resultados que mostrassem de forma inequívoca o impacto que cada paradigma tem nos *routers* e nas redes. Este impacto apenas pode ser avaliado se não existirem outros fatores que possam distorcer os resultados. É por este motivo que todos os cenários consideram apenas redes não sobrecarregadas.

Naturalmente, se um dado paradigma tem pior desempenho que outro numa rede não sobrecarregada, o seu comportamento será ainda pior numa rede que tenha tráfego adicional, na qual, possivelmente, alguns *links* terão pouca capacidade disponível e alguns elementos de rede estarão sobrecarregados.

Acresce que não é objetivo deste estudo executar testes de stress aos diversos paradigmas em estudo, pelo que o número de pacotes enviados por cada nó foi relativamente baixo. Os testes de stress foram alvo de um outro conjunto de simulações e são apresentados numa seção posterior.

5.2.2. Resultados de simulação

Foram realizados três conjuntos de simulações, abrangendo os dezassete cenários já referidos. O primeiro conjunto de simulações teve em vista o estudo do efeito de cada paradigma no tempo médio de ida e volta (*average round trip time*) dos pacotes. O segundo conjunto endereçou o estudo do tempo médio de otimização de rotas. O terceiro e último bloco de testes visou a análise da sobrecarga de sinalização imposta pelos procedimentos de mobilidade.

As subseções seguintes apresentam e discutem os resultados de cada conjunto de simulações em detalhe.

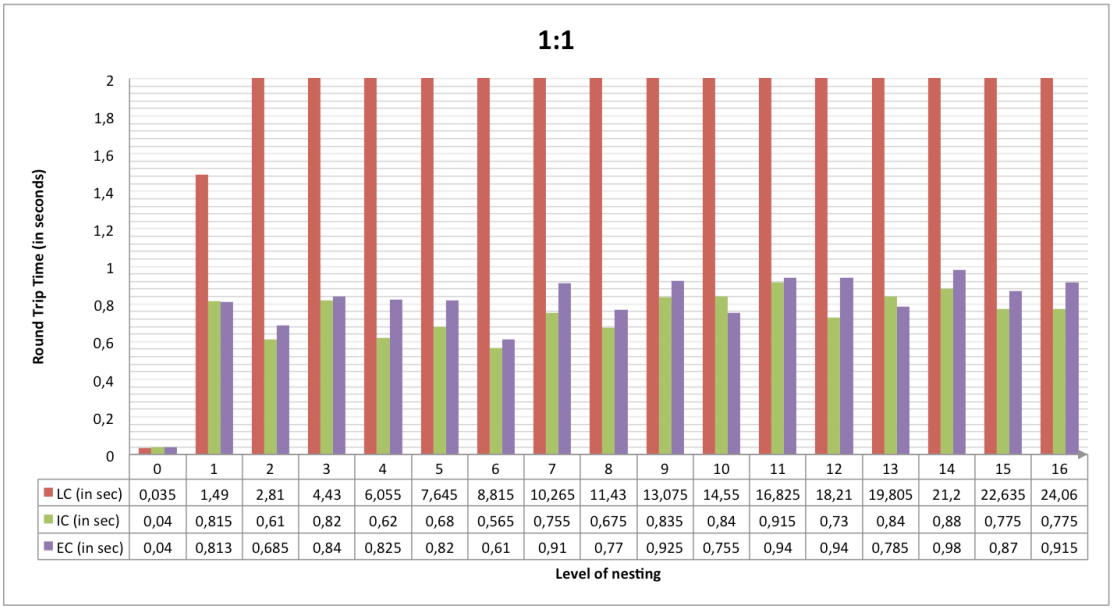
5.2.2.1 Tempo médio de ida e volta

Este conjunto de testes compreendeu um número considerável de simulações, para cada um dos paradigmas em estudo e para cada cenário (1 a 17). Adicionalmente, cada simulação foi executada para um determinado rácio de equipamentos que realizam ou não otimização de rotas. Os vários rácios utilizados fora 1:1, 1:2, 1:10, 1:100 e 1:1000.

As imagens presentes desde a Figura 5.9 até à Figura 5.13 apresentam os resultados obtidos.

A primeira conclusão que se extrai destes resultados é que a média do *round trip time* incrementa em função do nível de imbricação para o caso do paradigma centrado no equipamento antigo, LC.

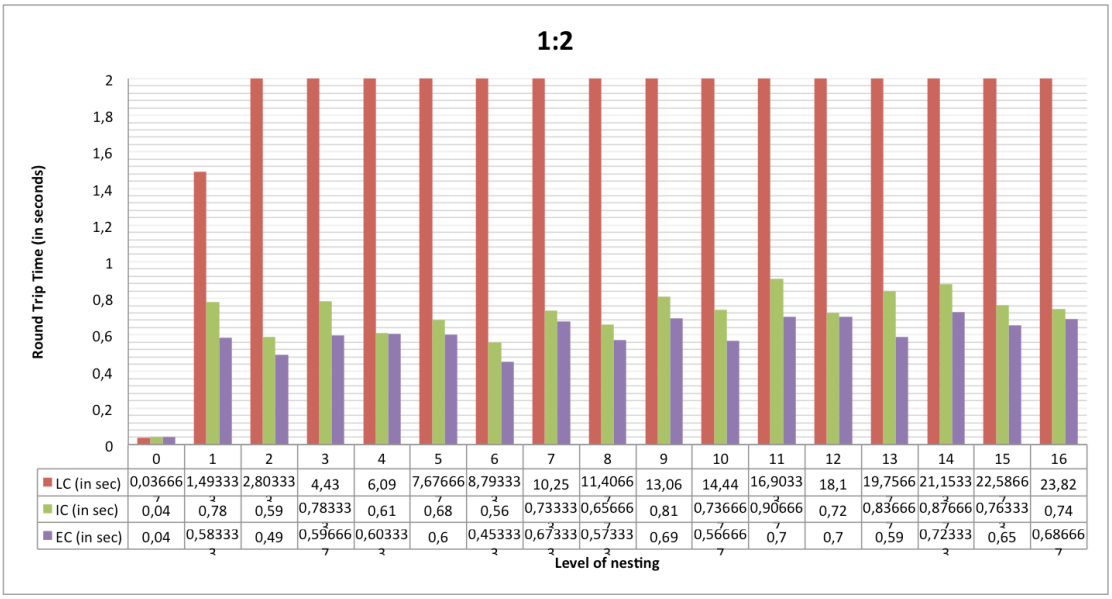
Claro que quanto maior é o nível de imbricação, maior é o número de túneis que dentro de túneis e, consequentemente, maior é o número de redes que os pacotes terão que atravessar de modo a atingir o seu destino. Esta é, obviamente, uma confirmação do que seria expectável.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LC (in sec)	0,035	1,49	2,81	4,43	6,055	7,645	8,815	10,265	11,43	13,075	14,55	16,825	18,21	19,805	21,2	22,635	24,06
IC (in sec)	0,04	0,815	0,61	0,82	0,62	0,68	0,565	0,755	0,675	0,835	0,84	0,915	0,73	0,84	0,88	0,775	0,775
EC (in sec)	0,04	0,813	0,685	0,84	0,825	0,82	0,61	0,91	0,77	0,925	0,755	0,94	0,94	0,785	0,98	0,87	0,915

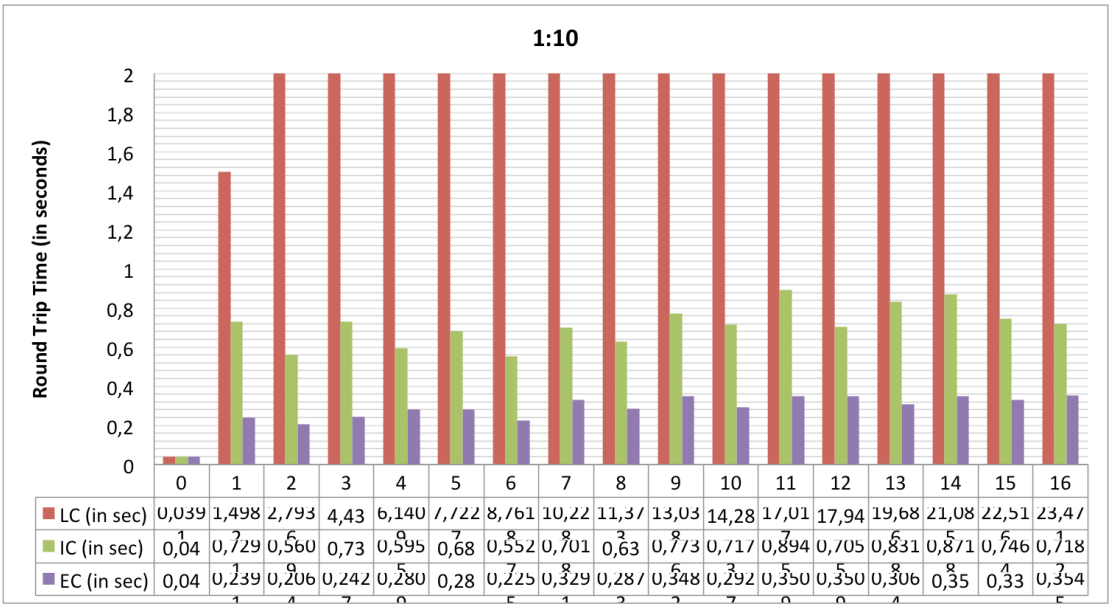
Figura 5.9 – RTT médio, para os três paradigmas, para todos os cenários e rácio de otimização de 1:1

É importante notar que, embora os resultados apenas apresentem níveis de imbricação até uma profundidade de 16, os testes realizados foram muito acima de 20. Contudo, a partir do nível de imbricação 17 a perda de pacotes era de tal modo elevada que inviabilizava a realização de testes.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LC (in sec)	0,036667	1,493333	2,803333	4,43	6,09	7,676667	8,793333	10,25	11,40667	13,06	14,44	16,90333	18,1	19,75667	21,15333	22,58667	23,82
IC (in sec)	0,04	0,78	0,59	0,783333	0,61	0,68	0,56	0,733333	0,656667	0,81	0,736667	0,906667	0,72	0,836667	0,876667	0,763333	0,74
EC (in sec)	0,04	0,583333	0,49	0,596667	0,603333	0,6	0,453333	0,673333	0,573333	0,69	0,566667	0,7	0,7	0,59	0,723333	0,65	0,686667

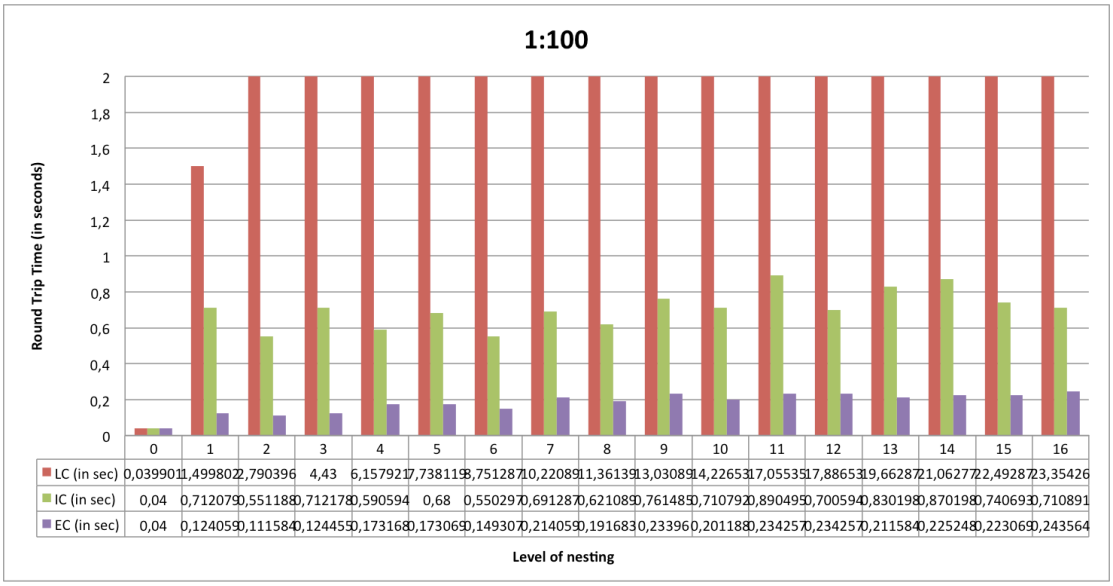
Figura 5.10 – RTT médio, para os três paradigmas, para todos os cenários e rácio de otimização de 1:2



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LC (in sec)	0,039	1,498	2,793	4,43	6,140	7,722	8,761	10,22	11,37	13,03	14,28	17,01	17,94	19,68	21,08	22,51	23,47
IC (in sec)	0,04	0,729	0,560	0,73	0,595	0,68	0,552	0,701	0,63	0,773	0,717	0,894	0,705	0,831	0,871	0,746	0,718
EC (in sec)	0,04	0,239	0,206	0,242	0,280	0,28	0,225	0,329	0,287	0,348	0,292	0,350	0,350	0,306	0,35	0,33	0,354

Figura 5.11 – RTT médio, para os três paradigmas, para todos os cenários e rácio de otimização de 1:10

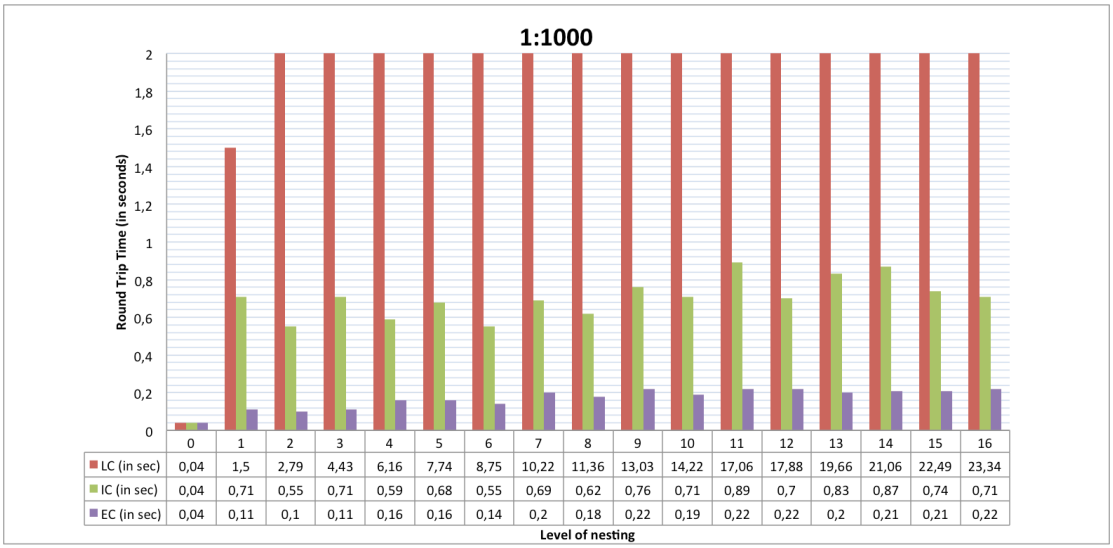
O segundo aspeto que se pode observar é que para qualquer um dos outros paradigmas, baseado na rede (IC) ou no cliente (EC), o nível de imbricação não afeta de forma significativa a média do *round trip time*.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LC (in sec)	0,039901	1,499802	2,790396	4,43	6,157921	7,738119	8,751287	10,22089	11,36139	13,03089	14,22653	17,05535	17,88653	19,66287	21,06277	22,49287	23,35426
IC (in sec)	0,04	0,712079	0,551188	0,712178	0,590594	0,68	0,550297	0,691287	0,621089	0,761485	0,710792	0,890495	0,700594	0,830198	0,870198	0,740693	0,710891
EC (in sec)	0,04	0,124059	0,111584	0,124455	0,173168	0,173069	0,149307	0,214059	0,191683	0,233960	0,201188	0,234257	0,234257	0,211584	0,225248	0,223069	0,243564

Figura 5.12 – RTT médio, para os três paradigmas, para todos os cenários e rácio de otimização de 1:100

Uma vez mais, este resultado serve de confirmação de que as simulações são consistentes, já que ambos os paradigmas usam otimização de rotas e, deste modo, o nível de imbricação não deveria afetar o RTT.

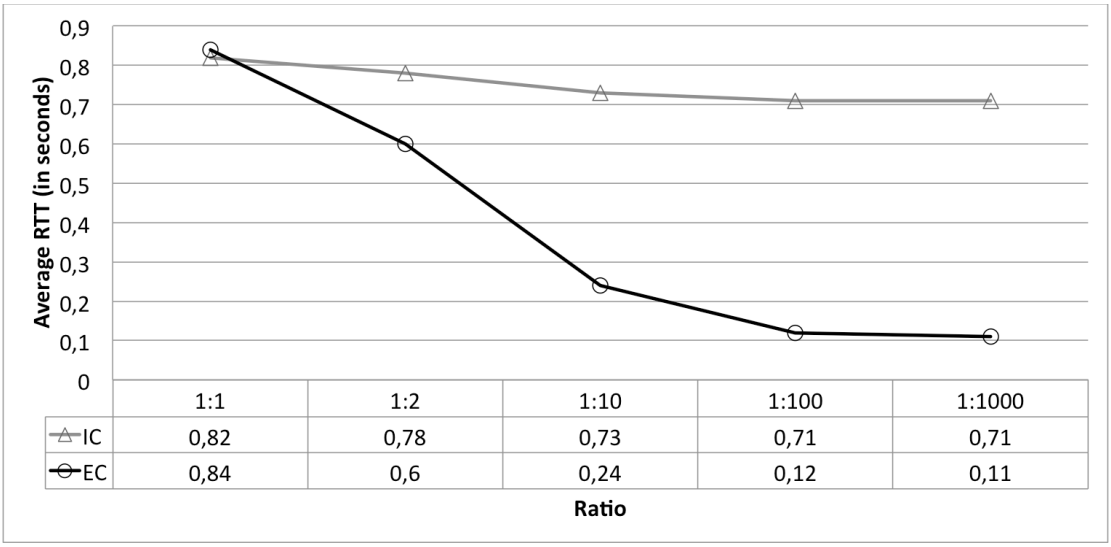


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LC (in sec)	0,04	1,5	2,79	4,43	6,16	7,74	8,75	10,22	11,36	13,03	14,22	17,06	17,88	19,66	21,06	22,49	23,34
IC (in sec)	0,04	0,71	0,55	0,71	0,59	0,68	0,55	0,69	0,62	0,76	0,71	0,89	0,7	0,83	0,87	0,74	0,71
EC (in sec)	0,04	0,11	0,1	0,11	0,16	0,16	0,14	0,2	0,18	0,22	0,19	0,22	0,22	0,2	0,21	0,21	0,22

Figura 5.13 – RTT médio, para os três paradigmas, para todos os cenários e rácio de otimização de 1:1000

Outra conclusão que se pode extrair destes dados está relacionada com o paradigma baseado no cliente, EC. Conforme se incrementa o número de equipamentos finais que realizam a operação de otimização de rotas (desde a Figura 5.9 até a Figura 5.13) a média do *round trip time* decrementa, devido ao facto que cada vez mais pacotes seguem o caminho ótimo.

Este facto torna-se mais evidente ao analisar-se a Figura 5.14, que apresenta a evolução da média do RTT, para um nível de imbricação 3.
É visível que, com o incremento no número de nós que realiza a operação de otimização de rotas, o paradigma baseado no cliente (EC) conduz a melhores resultados.



	1:1	1:2	1:10	1:100	1:1000
IC	0,82	0,78	0,73	0,71	0,71
EC	0,84	0,6	0,24	0,12	0,11

Figura 5.14 – RTT médio, para o nível de imbricação 3 e diferentes rácios de otimização de rotas

Por outro lado, no caso do paradigma baseado na rede, não há alteração significativa em função deste rácio já que os *routers* móveis têm que realizar a otimização de rotas para todos os equipamentos e fluxos. É mesmo interessante notar que todas as simulações mostram que este processamento adicional nos *routers* móveis leva a uma média de RTT mais fraca para o paradigma IC em todos os casos excepto para o caso do ratio 1:1. Para este último caso, a média de RTT do paradigma EC é pior porque metade dos pacotes não seguem o caminho ótimo, ao contrário do IC onde todos os pacotes seguem o melhor caminho possível.
No paradigma baseado no cliente, se um equipamento final não otimizar a rota, então fará uso do NEMO Basic Support Protocol. Contudo, o OMEN leva a que, mesmo que o nó da rede móvel esteja num nível de imbricação superior, o tempo médio do RTT dum pacote nesta situação seja sempre igual ao paradigma LC no nível de imbricação 1. Isto é sempre verdade porque o paradigma EC resolve o problema da profundidade de imbricação.

O último aspeto a salientar é que, quando o número de equipamentos finais que requerem otimização de rotas incrementa, o paradigma baseado no cliente obtém melhores resultados do que o paradigma baseado na rede. Isto pode ser facilmente visto nas Figura 5.10 a Figura 5.13.

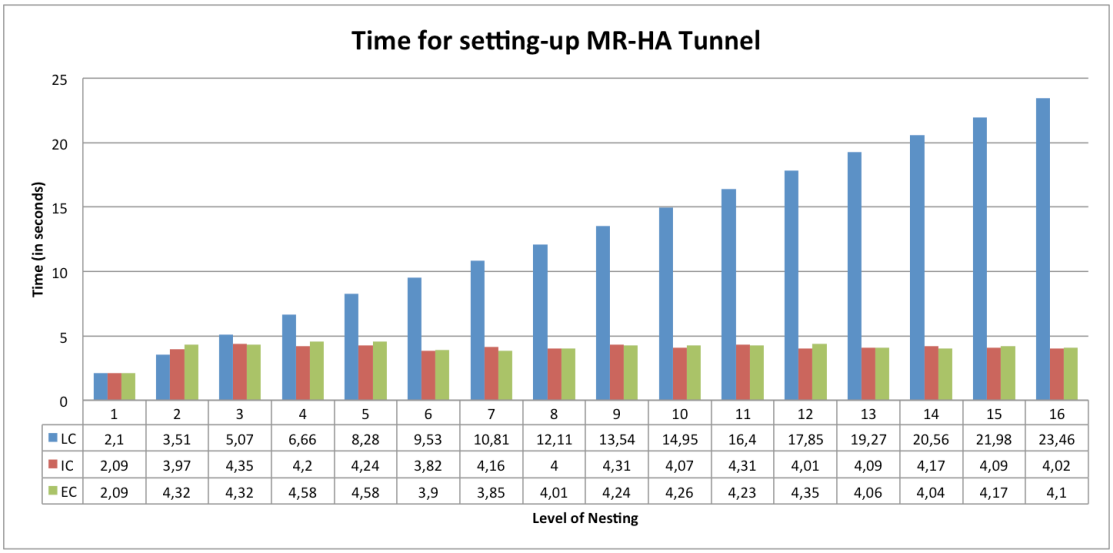
A razão para este comportamento reside no rácio de nós que realizam a operação de otimização de rotas. Quanto maior for o número deste tipo de equipamentos, menor será o esforço introduzido no *router* móvel. Por oposição, no caso do paradigma IC o *router* móvel necessita realizar todas as operações de otimização de rotas para todos os equipamentos finais, pelo que o esforço é necessariamente maior.

5.2.2.2 Tempo médio de otimização de rotas

O segundo conjunto de simulações endereçou as questões de configuração do túnel MRHA e da otimização de rotas.

Uma vez mais, por uma questão de confirmação da consistência, começou-se por obter os dados relativos à configuração do túnel MRHA. Teoricamente, este tempo deverá incrementar em função do nível de imbricação para o caso do paradigma baseado nos equipamentos antigos (LC), e não deverá variar significativamente para os paradigmas baseados na rede (IC) e no cliente final (EC).

De facto, os resultados obtidos vêm confirmar que este comportamento é consistente com a teoria. A Figura 5.15 mostra os resultados da simulação.

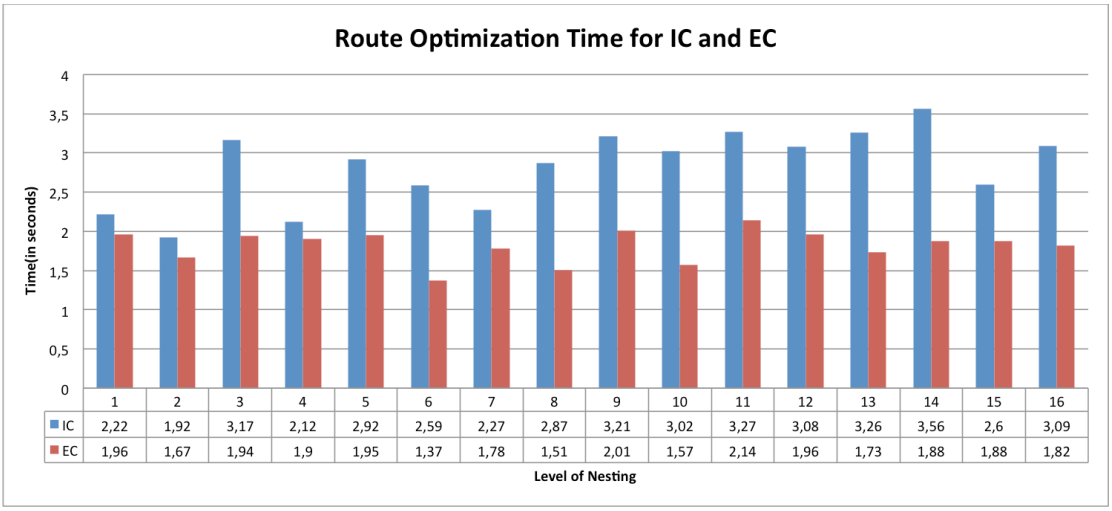


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LC	2,1	3,51	5,07	6,66	8,28	9,53	10,81	12,11	13,54	14,95	16,4	17,85	19,27	20,56	21,98	23,46
IC	2,09	3,97	4,35	4,2	4,24	3,82	4,16	4	4,31	4,07	4,31	4,01	4,09	4,17	4,09	4,02
EC	2,09	4,32	4,32	4,58	4,58	3,9	3,85	4,01	4,24	4,26	4,23	4,35	4,06	4,04	4,17	4,1

Figura 5.15 – Tempo necessário para configurar o túnel MRHA

Com a confirmação dos resultados obtidos, as simulações a seguir tiveram por objetivo obter o tempo médio necessário para o estabelecimento da otimização de rotas, naturalmente apenas para os paradigmas baseados na rede e no cliente final, já que o paradigma baseado no equipamento antigo não incluí qualquer funcionalidade de optimização de rotas.

Para o caso da otimização de rotas, dois aspetos foram tidos em consideração: o primeiro foi o tempo médio como uma função dos níveis de imbricação, enquanto que o segundo foi o tempo médio em função do número de nós que requerem otimização de rotas. Os resultados respeitantes ao primeiro aspeto são apresentados na Figura 5.16.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
IC	2,22	1,92	3,17	2,12	2,92	2,59	2,27	2,87	3,21	3,02	3,27	3,08	3,26	3,56	2,6	3,09
EC	1,96	1,67	1,94	1,9	1,95	1,37	1,78	1,51	2,01	1,57	2,14	1,96	1,73	1,88	1,88	1,82

Figura 5.16 – Tempo médio para otimização de rotas, em função ao nível de imbricação

Como pode ser verificado, não há uma variação significativa do tempo médio da otimização de rotas em função do nível de imbricação, tal como seria de esperar.

Não obstante, pode-se observar que os valores para o paradigma baseado na rede (IC) são consistentemente superiores aos valores do paradigma baseado no cliente final. Isto deve-se ao facto de que para o caso do paradigma baseado na rede o processamento necessário para realizar as operações de otimização de rotas é superior nos *routers* móveis, já que todas as tarefas de otimização de rotas são realizadas por estes elementos da rede. Este comportamento não se verifica no paradigma baseado no cliente final, no qual as operações de otimização de rotas são realizadas pelos equipamentos finais.

Para confirmar esta tese, foram realizadas simulações adicionais com o objetivo de determinar a evolução do tempo necessário para o estabelecimento da otimização de rotas se for incrementado o número de equipamentos que requerem esta funcionalidade. A Figura 5.17 apresenta esta evolução, que vai desde 1 equipamento até um máximo de 51 equipamentos que requerem otimização de rotas.

Os resultados parecem confirmar que, no caso do paradigma baseado na rede (IC), quanto maior é o número de equipamentos que requerem otimização de rotas, maior é o tempo médio para a otimização de rotas, registando-se um aumento significativo. Já no caso do paradigma EC, esse tempo mantém-se estável.

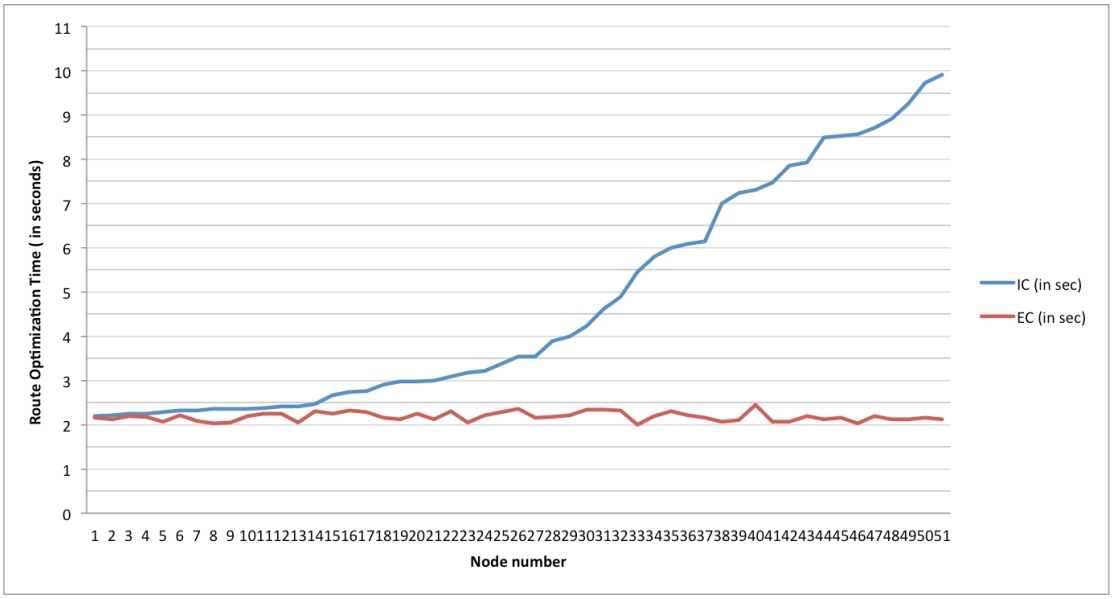


Figura 5.17 – Tempo médio para otimização de rotas (RO), em função ao número de nós que requerem RO

5.2.2.3 Sobrecarga de sinalização

O conjunto final da presente série de testes endereçou a questão da sobrecarga de sinalização, decorrente da otimização de rotas. Esta sinalização inclui os pacotes relacionados com o procedimento de *return routability* ou de anúncio da nova localização, tais como os *home-test init* (HoTi), *care-of test init* (CoTi), *home test* (HoT), *care-of test* (CoT), *binding update* (BU), *binding acknowledgement* (BA).

Existem dois fatores determinantes que afetam a sobrecarga de sinalização: o número de saltos de mobilidade e o rácio de equipamentos que requerem/não requerem otimização de rotas. De uma maneira geral, a sinalização aumenta com

- o incremento do número de saltos de mobilidade do *router* móvel;
- o incremento do número de equipamentos que requerem a otimização de rotas.

A Figura 5.18 apresenta resultados para a sobrecarga de sinalização para o caso em que o ponto de acesso à Internet de um *router* móvel é alterado 5 vezes, para diferentes rácios de otimização de rotas.

Como se pode verificar, os resultados da simulação mostram claramente que a sobrecarga de sinalização para o paradigma baseado no equipamento final (EC) é sempre inferior à do paradigma baseado na rede (IC). Como já se viu, o paradigma IC realiza o procedimento de otimização de rotas para todos os nós da rede móvel, independentemente da sua real necessidade.

	10:1	1:1	1:2	1:5	1:10	1:100
IC	110%	110%	110%	110%	110%	110%
EC	101%	105%	107%	108%	109%	110%
Data packets	100%	100%	100%	100%	100%	100%

Figura 5.18 – Sinalização adicional para 5 saltos e para diferentes ratios de otimização

Quando a percentagem de equipamentos que requerem otimização de rotas incrementa, a sinalização adicional do paradigma baseado no cliente final (EC) aproxima-se do paradigma baseado na rede (IC).

5.2.3. Conclusões da avaliação em cenários de muito grande dimensão

Foi apresentada uma comparação dos três paradigmas de referência para a mobilidade de redes – baseado nos equipamentos antigos, baseado na rede e baseado no cliente final. A comparação visou um cenário de muito grande dimensão, tendo em mente uma implementação da mobilidade de redes em larga escala, tal como se espera que seja a Internet do futuro.

Os resultados apresentados ajudam a extrair várias conclusões. A primeira é que o paradigma baseado no equipamento antigo tem diversas limitações de desempenho e, deste modo, dificilmente poderá ser utilizado em cenários de grande dimensão. Claramente, esta solução foi desenvolvida a pensar nos equipamentos antigos e em cenários em que a mobilidade é a exceção.

Os resultados da simulação em relação ao paradigma baseado na rede mostram claramente que a concentração de funcionalidade nos elementos de rede afeta o desempenho global do sistema e, consequentemente, a qualidade da comunicação entre os nós.

Por outro lado, as simulações realizadas também mostram claramente que o paradigma baseado no cliente final tem potencial. Ao libertar os *routers* móveis, e outros elementos da

rede, das tarefas de mobilidade, beneficia-se significativamente o desempenho, quer em termos de tempos de *round trip time*, quer em termos de tempos de otimização de rotas e, ainda, em termos de sinalização.

5.3. Avaliação em cenários de carga elevada

Os testes realizados na secção anterior, 5.2. Avaliação , focaram a avaliação do desempenho de diversos paradigmas em função ao número de equipamentos móveis, num cenário de muito grande dimensão. Não obstante o elevado número de equipamentos presentes no estudo, não foram realizados testes de carga.

Isto significa que os paradigmas foram avaliados na perspetiva de um elevado número de equipamentos que se comportavam de forma ordeira. Nesse estudo, verificou-se que os paradigmas têm comportamentos diferenciados em termos de desempenho, escalabilidade e sinalização adicional.

Parece natural que o próximo passo seja a avaliação do comportamento dos paradigmas quando confrontados com elevadas cargas de tráfego. Ou seja, as perguntas que se colocam são: se, quando sujeitos a uma reduzida carga de tráfego em ambientes de pequena dimensão, todas as soluções de mobilidade em estudo apresentam comportamentos aceitáveis, como será que estas mesmas soluções se irão comportar perante cargas de tráfego elevadas? Em que medida é que as escolhas arquiteturais influenciarão o seu comportamento perante situações de stress na rede? É objectivo desta secção analisar os diversos paradigmas nesta nova perspetiva.

Assim, foi criado um cenário realístico, de média dimensão, em que foram emulados diversos níveis de carga. Para ajudar a compreender o problema de vários pontos de vista procedeu-se à variação do número de nós móveis, do intervalo entre pacotes, dos níveis de imbricação e da percentagem de fluxos de tráfego que requerem otimização de rotas.

É importante notar que se optou por isolar os cenários de testes dos factores externos à mobilidade, tal como tráfego de fundo, utilização de diferentes modelos de tráfego ou a criação de zonas de estrangulamento em pontos fixos da rede. Deste modo, sabe-se que os resultados apresentados dizem respeito exclusivamente às opções arquiteturais de cada uma das soluções de mobilidade em estudo.

5.3.1. Cenários de simulação

Os testes de stress de carga de tráfego foram avaliados com recurso ao emulador de mobilidade de redes *mobSim*. Foi utilizado o emulador mobSim porque tem uma arquitetura

desenhada especificamente para cenários de emulação de larga dimensão e/ou de intenso tráfego, tendo sido construído com o objetivo de tirar proveito do processamento paralelo de servidores ou clusters.

O cenário da Figura 5.19 foi criado para este efeito e permitiu avaliar o desempenho das soluções de mobilidade de redes mais relevantes: NEMO Basic Support Protocol, que se encaixa no paradigma dos equipamentos antigos (LC); MIRON, representativo do paradigma baseado na rede (IC); e o OMEN, que pertence ao paradigma baseado no equipamento final (EC).

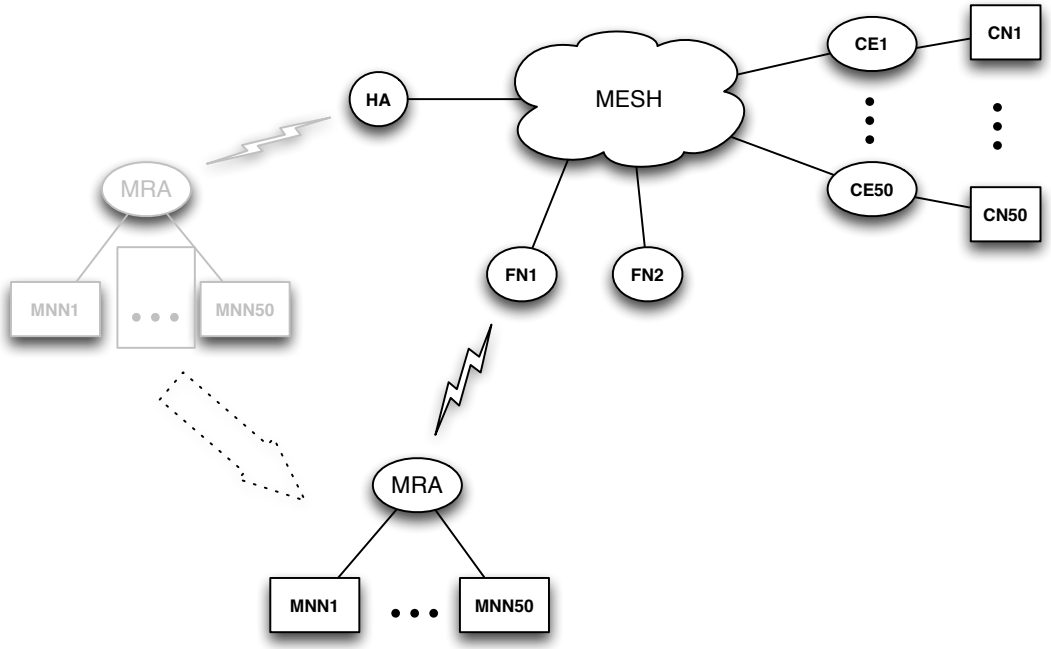


Figura 5.19 – Topologia base dos cenários de emulação

Foi adotada uma topologia em estrela, constituída por 56 *routers* de topo, interligados por uma malha total. A ideia subjacente a esta topologia é garantir que não existem pontos de estrangulamento ou outros fatores indesejáveis na rede.

Este cenário contém 4 *home networks* (HN), cada uma contendo uma rede móvel e respetivo *router* móvel. Existem 2 redes estrangeiras (*Foreign Network*, `FN`).

Por fim, existem 50 redes fixas, contendo o *router* `CE1` até ao `CE50`, cada uma contendo apenas um simples nó correspondente (`CN1` até `CN50`, respetivamente).

A rede móvel `A`, cujo *router* móvel de acesso é o `MRA` – contém até um máximo de 50 nós da rede móvel. As restantes redes móveis, que não se encontram representadas na figura por uma questão de simplificação, são usadas apenas para construção de ambientes imbricados e não contêm qualquer nó móvel.

Cada nó da rede móvel comunica com um único nó correspondente. Assim, o nó MNN1 apenas comunica com o CN1, o MNN2 comunica com o CN2, e por aí adiante. Uma vez mais, o objetivo é conseguir evitar qualquer ponto de estrangulamento ou sobrecarga num determinado ponto da infraestrutura, dado que isso poderia influenciar os resultados com factores externos às soluções de mobilidade de redes em estudo. Pela mesma razão, foi decidido não introduzir nenhum tráfego adicional de fundo na rede. Os valores de desempenho que venham a ser obtidos serão, deste modo, exclusivamente derivados das características intrínsecas e capacidades das soluções de mobilidade de rede em estudo.

Os testes que foram levados a cabo cobrem todas as combinações dos seguintes valores para os parâmetros em análise, para cada um dos paradigmas de mobilidade de rede:

- Tempo médio de intervalo entre pacotes: 1ms, 5ms, 10ms, 50ms, 100ms, 1000ms; naturalmente, quanto menor for o intervalo entre os pacotes maior é a carga de tráfego na rede;
- Variação de número de nós dentro da rede móvel A (cujo *router* móvel é o MRA): 5, 10, 15, 20, 25, 30, 35, 40, 45 e 50;
- Variação do rácio de fluxos de pacotes para os quais existe otimização de rotas (RO); um ratio de optimização de rotas de n:m significa que, para um total de n+m fluxos, n fluxos não seguem uma rota otimizada, enquanto que m fluxos seguem uma rota otimizada; este parâmetro apenas tem impacto no paradigma baseado no cliente final; foram usados os seguintes ratios: 1:1, 1:5, 1:10 e 1:25;
- Variação do nível de imbricação; foram considerados 5 nível de imbricação no estudo: sem imbricação e a rede móvel estava dentro da sua rede nativa, sem imbricação e a rede móvel estava numa rede estrangeira, 1 nível de imbricação (i.e., a rede móvel estava dentro de uma rede móvel não imbricada), 2 níveis de imbricação e 3 níveis de imbricação.

Assim, foram realizadas 1200 simulações diferentes, para cada solução de mobilidade de redes. Cada simulação foi executada 3 vezes para cada uma das 3 soluções de mobilidade de redes, perfazendo um total de 10 800 simulações realizadas.

5.3.2. Resultados de simulação

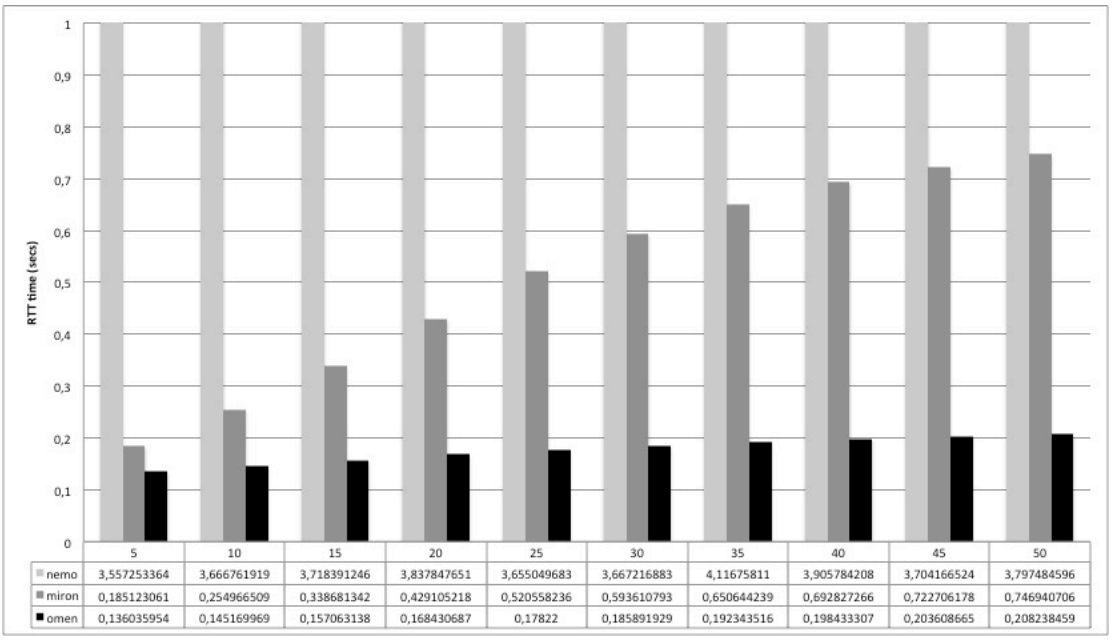
As seguintes subsecções apresentam os resultados dos testes de acordo com as configurações apresentadas anteriormente.

O primeiro aspeto a ser analisado diz respeito às implicações da variação do número de equipamentos móveis. Posteriormente, avaliou-se o impacto da variação do intervalo entre os pacotes, do nível de imbricação e do rácio de otimização de rotas. Por fim, foi analisada a taxa perda de pacotes para os diversos paradigmas em estudo.

5.3.2.1 Variação do número de nós da rede móvel

Em primeiro lugar foi analisado o comportamento das várias soluções de mobilidade de redes em função do número de equipamentos móveis, que variou até um máximo de 50.

Os resultados apresentados na Figura 5.20 referem-se à média do *round trip time*, RTT, obtido para todas as combinações dos restantes parâmetros, nomeadamente, o tempo de intervalo entre pacotes, rácio de otimização de rotas e o nível de imbricação.



	5	10	15	20	25	30	35	40	45	50
nemo	3,557253364	3,666761919	3,718391246	3,837847651	3,655049683	3,667216883	4,11675811	3,905784208	3,704166524	3,797484596
miron	0,185123061	0,254966509	0,338681342	0,429105218	0,520558236	0,593610793	0,650644239	0,692827266	0,722706178	0,746940706
omen	0,136035954	0,145169969	0,157063138	0,168430687	0,17822	0,185891929	0,192343516	0,198433307	0,203608665	0,208238459

Figura 5.20 – Tempos médios de *round trip time* (RTT) como função do número de nós da rede móvel

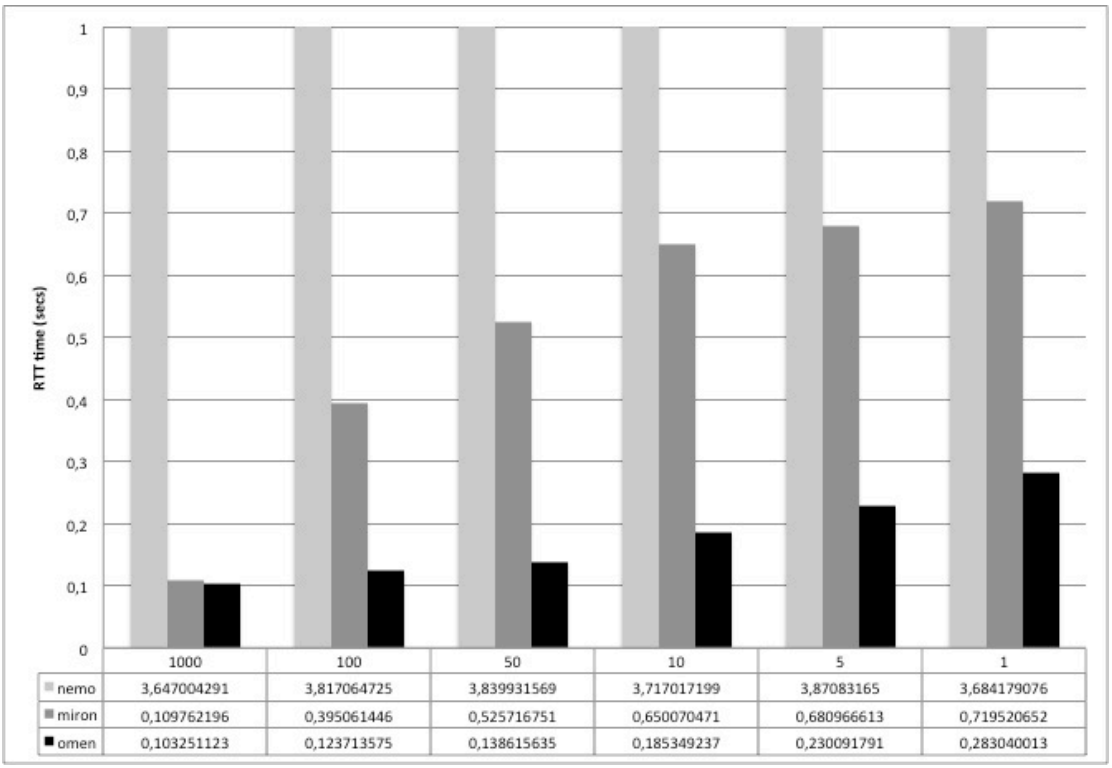
Pode-se verificar claramente nesta figura que, para a solução NEMO Basic Support Protocol, o desempenho piora substancialmente à medida que aumenta o número de nós da rede móvel, MNN, o mesmo se passando no caso do paradigma baseado na rede. Já no caso da solução de mobilidade baseada no cliente final sofre uma degradação muito leve.

É importante notar que, nesta como nas outras figuras, as barras estão limitadas a um máximo de 1 segundo, para garantir a legibilidade da figura.

5.3.2.2 Variação do intervalo entre pacotes

Também foi analisado o comportamento de cada uma das três soluções de mobilidade de rede em função da carga na rede. De modo a realizar este teste optou-se por variar o intervalo entre os pacotes de acordo com uma distribuição de Poisson.

A Figura 5.21 apresenta o tempo médio de *round trip time* (RTT) em função dos intervalos médios entre geração dos pacotes. Estes resultados contemplam todas as combinações dos restantes parâmetros em estudo.



	1000	100	50	10	5	1
nemo	3,647004291	3,817064725	3,839931569	3,717017199	3,87083165	3,684179076
miron	0,109762196	0,395061446	0,525716751	0,650070471	0,680966613	0,719520652
omen	0,103251123	0,123713575	0,138615635	0,185349237	0,230091791	0,283040013

Figura 5.21 –Tempo médio de *round trip time* (RTT) em função do intervalo entre pacotes

Os resultados mostram que tanto o NEMO Basic Support Protocol como o paradigma baseado na rede têm uma degradação acentuada do desempenho com o incremento da carga na rede. Uma vez mais, estes valores confirmam o impacto negativo que se verifica no *router* móvel devido às operações de mobilidade que se vêm obrigados a realizar em vez dos nós finais.

O impacto no desempenho é, por outro lado, muito menor no paradigma baseado no cliente final. Neste caso, o incremento do RTT médio é meramente devido ao aumento do tráfego existente na rede, não havendo influência dos mecanismos inerentes à mobilidade. De facto, no caso do OMEN o *router* móvel comporta-se como um mero encaminhador de

pacotes encontrando-se, portanto, sujeito à dinâmica esperada para este tipo de equipamentos.

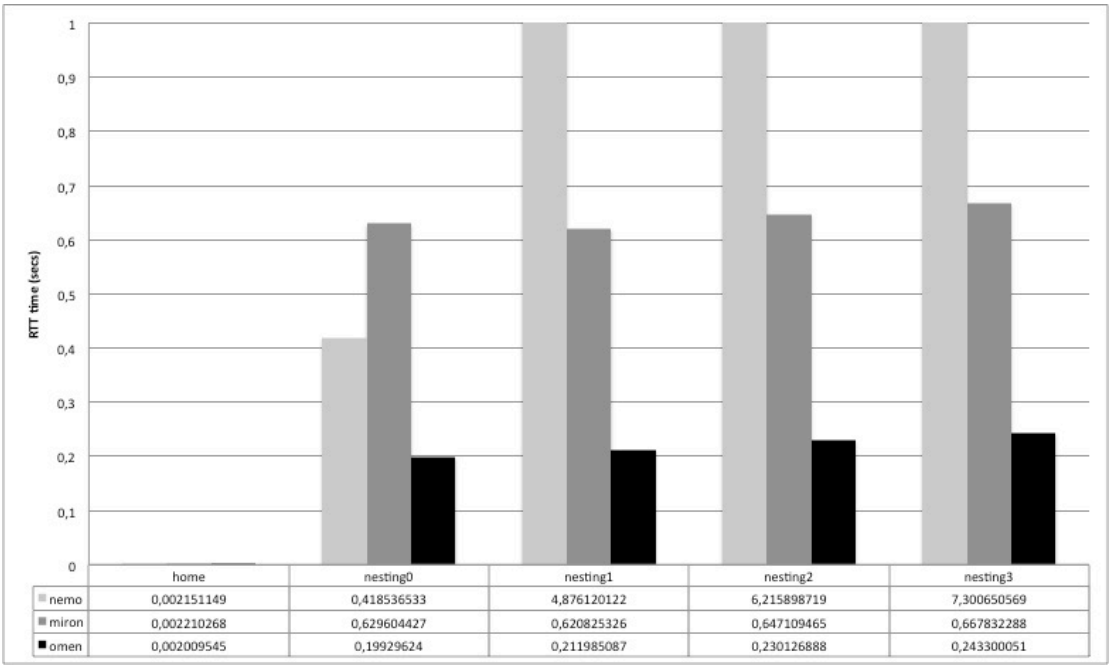
5.3.2.3 Variação do nível de imbricação

Um aspecto muito importante no estudo de soluções de mobilidade de redes é a análise de cenários com imbricação, já que estes colocam uma carga adicional nos *routers* móveis envolvidos no processo.

De modo a avaliar estas questões, contemplaram-se os seguintes cenários de ligação de uma rede móvel à rede externa: o *router* móvel estava na rede origem; o *router* móvel estava ligado a uma rede fixa, mas fora da rede origem (sem imbricação); l o *router* móvel estava ligado a outro *router* móvel (nível de imbricação 1); o *router* móvel estava ligado a um *router* móvel em imbricação (nível de imbricação 2); e, por fim, o *router* móvel estava ligado a um *router* móvel num nível de imbricação 2 (i.e., nível de imbricação 3).

A Figura 5.22 apresenta o *round trip time* (RTT) médio para cada uma das soluções de mobilidade em função ao nível de imbricação.

O primeiro aspeto que se nota na figura é que o comportamento das 3 soluções é muito bom quando considerado o cenário em que o *router* móvel se encontra na rede origem. Neste caso, o RTT médio não vai muito longe dos 2ms. De facto, neste caso não é utilizado nenhum mecanismo de mobilidade, já que o endereçamento está topologicamente correto.



	home	nesting0	nesting1	nesting2	nesting3
nemo	0,002151149	0,418536533	4,876120122	6,215898719	7,300650569
miron	0,002210268	0,629604427	0,620825326	0,647109465	0,667832288
omen	0,002009545	0,19929624	0,211985087	0,230126888	0,243300051

Figura 5.22 – Tempo médio de *round trip time* (RTT) em função do nível de imbricação

Por outro lado, nas outras situações os efeitos da mobilidade no desempenho são claramente visíveis.

Quando a rede móvel se desloca para uma rede estrangeira fixa – cenário sem imbricação, identificado por `nesting0` na figura –, o desempenho da solução NEMO Basic Support Protocol tem uma degradação acentuada devido ao facto de todos os pacotes terem que circular pelo túnel bidirecional MRHA, estando sujeitos ao efeito da triangulação.

No que parece à primeira vista uma surpresa, a solução baseada na rede tem um comportamento pior que a solução NEMO Basic Support Protocol, para este teste de mobilidade sem imbricação.

De facto, para todas as soluções de mobilidade, quando uma rede móvel se desloca para fora da sua rede origem, esta tem que realizar as tarefas relacionadas com a mobilidade de redes.

No caso da solução NEMO Basic Support Protocol esta operação consiste num procedimento de *binding update* entre o *router* móvel e o *home agent*, de modo a criarem o túnel bidirecional MRHA.

No caso das soluções baseadas em infraestrutura em geral – e no caso particular do MIRON –, torna-se necessária a realização do procedimento de *Return Routability* e de *binding update* para todos os fluxos dos diversos nós da rede móvel que estejam a circular no momento na rede móvel, independentemente de haver necessidade de otimizar ou não.

Por outro lado, no caso do OMEN, os procedimentos de *Return Routability* e de *binding update* são apenas realizados se forem necessários e, ainda por cima, são totalmente operados pelos nós da rede móvel.

O impacto negativo no desempenho para as soluções de mobilidade baseadas na rede é, assim, claramente visível. Neste ambiente de teste, em que a distância entre o *home agent* e o *mobile* router do nó correspondente é mínima, o esforço adicional de criar as rotas otimizadas não compensa a utilização do caminho mais longo e não otimizado do NEMO Basic Support Protocol.

Por outro lado, o *router* móvel OMEN age como um mero *router* e, deste modo, o seu desempenho é apenas afetado pela carga introduzida na rede, do mesmo modo que qualquer *router* fixo. Assim, para o caso do ambiente sem imbricação, o OMEN evidencia-se das restantes soluções.

Já nos casos da mobilidade imbricada, o desempenho da solução NEMO Basic Support Protocol é altamente penalizado pela amplificação da falta de otimização de rotas, enquanto que as soluções baseadas na rede e no cliente final não são significativamente afetadas.

Em todo o caso, a solução OMEN tem um comportamento melhor quando comparado com as restantes.

5.3.2.4 Variação do rácio de otimização de rotas

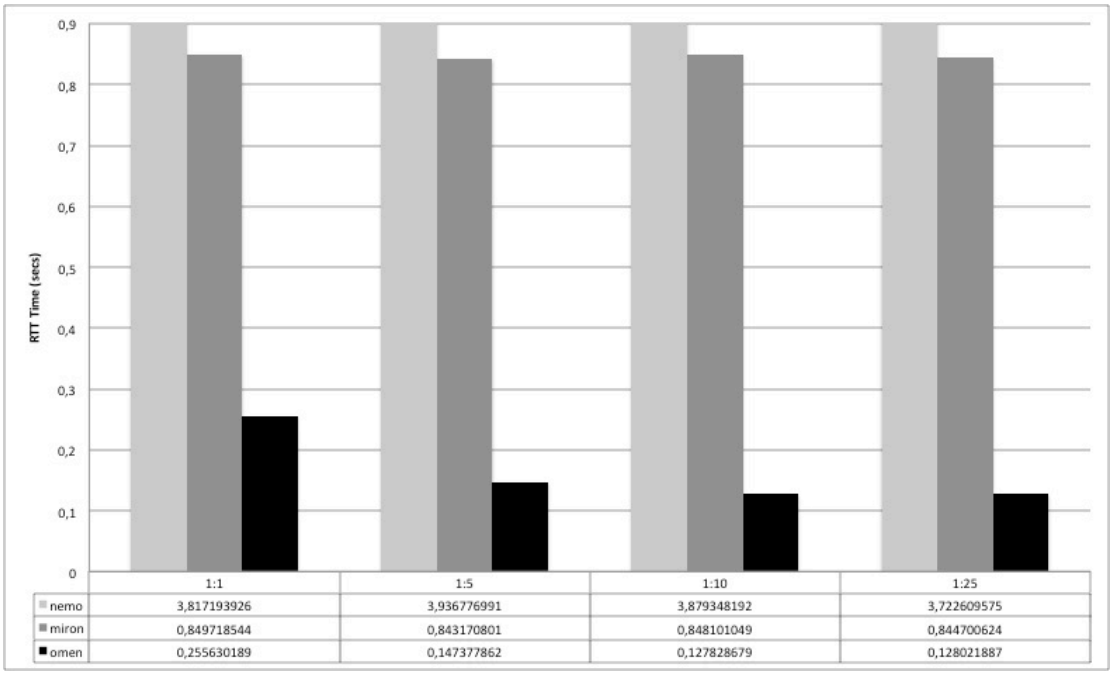
Quando é incrementada a percentagem de fluxos que utilizam otimização de rotas, o desempenho dos sistemas que executam as tarefas de otimização de rota será, naturalmente, afetado.

Se estes sistemas forem elementos de rede – em particular, *routers* – que lidam com grandes quantidades de fluxos, então o desempenho destes sistemas pode ser significativamente afetado com o incremento da carga na rede.

Por outro lado, se as operações de otimização de rotas forem da responsabilidade dos equipamentos finais, então os elementos da rede não serão afetados por essas operações, e os sistemas finais serão apenas sujeitos a um leve incremento na carga já que cada um lida com um número limitado de fluxos.

De modo a confirmar esta tese, foram realizadas testes com vários rácios de otimização de rotas, tal como explicado anteriormente. Os testes realizados envolveram os rácios de 1:1, 1:5, 1:10 e 1:25. Os resultados destes testes são apresentados na Figura 5.23.

É possível verificar que, para o caso do OMEN, o *round trip time* (RTT) médio diminui com o aumento da percentagem de nós da rede móvel que realizam otimização de rotas.



	1:1	1:5	1:10	1:25
nemo	3,817193926	3,936776991	3,879348192	3,722609575
miron	0,849718544	0,843170801	0,848101049	0,844700624
omen	0,255630189	0,147377862	0,127828679	0,128021887

Figura 5.23 – Tempo médio de *round trip time* (RTT) em função do rácio de otimização de rotas

Note-se que um rácio de 1:1 significa que apenas metade dos equipamentos estão a otimizar as rotas. Se o número de dispositivos que otimizam rotas incrementar, isto leva a um decréscimo no tempo médio de ida e volta e, consequentemente, a uma melhoria do desempenho da comunicação.

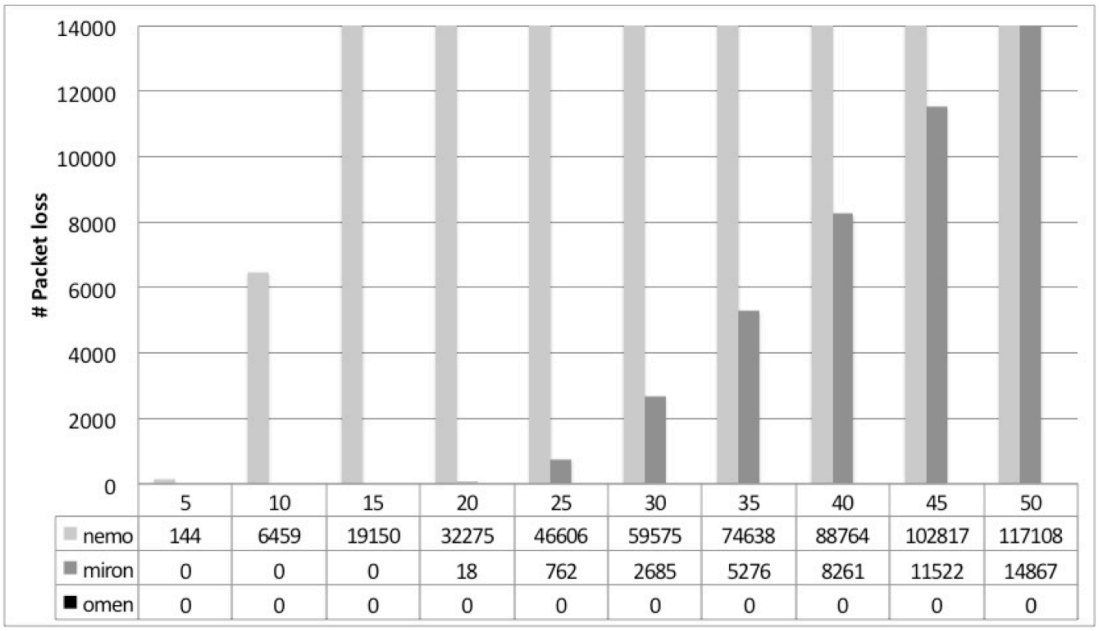
O rácio de otimização não influencia as restantes soluções em estudo. No caso da solução baseada na rede, a otimização de rotas é efetuada para todos os fluxos, quer seja necessário ou não. Por outro lado, o NEMO não realiza nenhuma otimização de rotas (*triangular routing*).

5.3.2.5 Perda de pacotes

A comparação do desempenho das três soluções de mobilidade de redes não estaria completa se não fosse analisada a perda de pacotes observada em ambientes de carga na rede.

O impacto no desempenho pode ser observado através do incremento do número de nós da rede móvel, da variação do nível de imbricação, ou da variação do intervalo entre os pacotes.

A Figura 5.24 apresenta a média de perda de pacotes em função ao número de nós da rede móvel. É importante notar que, para um determinado número de nós da rede móvel, a média de perda de pacotes foi obtida considerando todas as combinações dos restantes parâmetros em estudo.



	5	10	15	20	25	30	35	40	45	50
nemo	144	6459	19150	32275	46606	59575	74638	88764	102817	117108
miron	0	0	0	18	762	2685	5276	8261	11522	14867
omen	0	0	0	0	0	0	0	0	0	0

Figura 5.24 – Número médio de perda de pacotes em função do número de nós da rede móvel

Nesta figura é possível observar que a perda de pacotes cresce com o número de nós da rede móvel, ou seja, com a carga existente no sistema.

No caso do NEMO, a perda de pacotes começa com um número reduzido de equipamentos (para 5 nós da rede móvel, já havia perdas de 144 pacotes). Após uma análise mais detalhada dos resultados, foi possível constatar que estas perdas de pacotes ocorrem para as cargas mais elevadas (intervalo entre pacotes mais reduzidos) e para os níveis de imbricação mais elevados (nível 3).

Para o caso da solução de mobilidade de redes baseada na infraestrutura, as perdas de pacotes começam com 25 nós da rede móvel.

De forma notável, o OMEN nunca perde pacotes em nenhum dos cenários. Seria de esperar que, pelo menos, os nós da rede móvel que não realizam otimização de rotas introduzissem alguma perda de pacotes.

Contudo, é importante notar que na solução OMEN, independentemente do nível de imbricação, um nó da rede móvel que não use otimização de rotas comporta-se da mesma maneira que na solução NEMO Basic Support Protocol sem imbricação.

Tendo isto em consideração, foi realizada uma nova emulação para determinar qual o comportamento do NEMO Basic Support Protocol com imbricação zero tendo-se chegado à conclusão que começa a perder pacotes a partir de 45 nós de rede móvel.

Se for utilizado o ratio de 1:1 de otimização de rotas, em que metade dos equipamentos não realizam a operação de RO, isto significa que o OMEN deverá começar a perder pacotes aproximadamente a partir de 90 nós de rede móvel.

De modo a confirmar esta hipótese, foi realizada nova simulação com 100 nós da rede móvel, tendo sido obtidos os resultados apresentados na Figura 5.25.

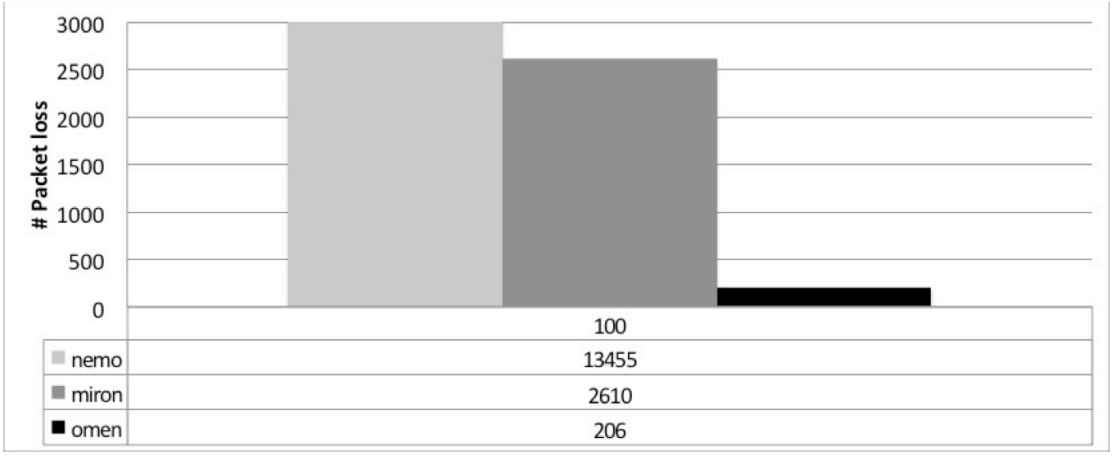


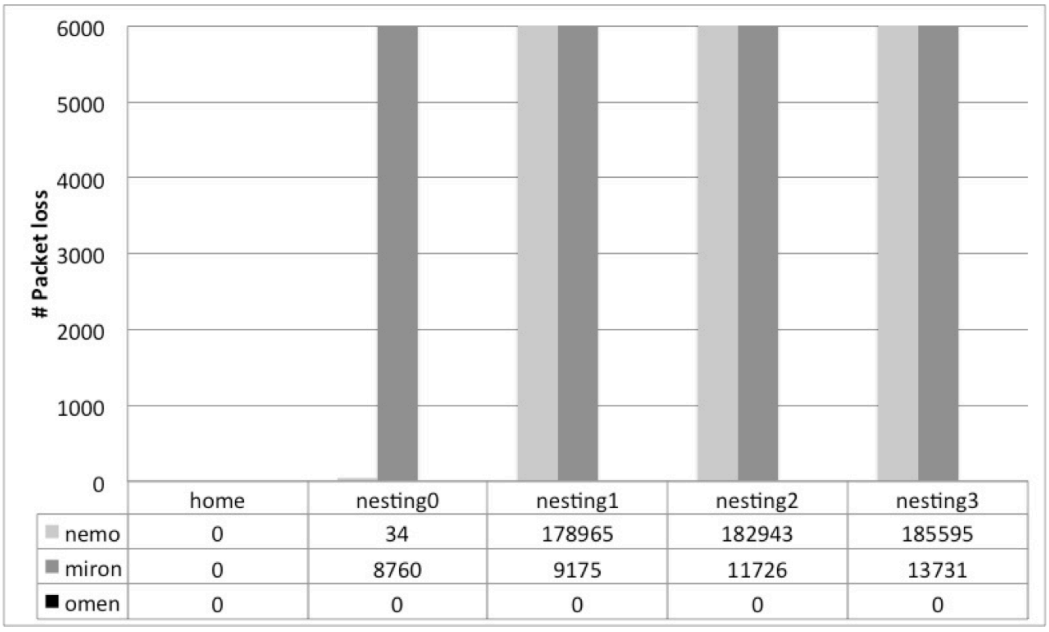
Figura 5.25 – Média de perda de pacotes para 100 nós da rede móvel, 1ms de intervalo entre pacotes, rácio de otimização de 1:1, sem imbricação

O gráfico confirma a hipótese, mostrando uma média de perda de 206 pacotes para a solução OMEN, quando são utilizados 100 nós da rede móvel, num ratio de 1:1 de otimização de rotas, com um intervalo entre pacotes de 1ms, e sem qualquer nível de imbricação.

Esta simulação foi realizada para as três soluções em estudo para verificar se o comportamento das restantes soluções se mantinha consistente com os resultados anteriores.

Ao mesmo tempo que confirma a hipótese, este gráfico mostra a clara vantagem da solução baseada no cliente final, arquitetura base da proposta OMEN, que conduz a uma infraestrutura mais leve, capaz de lidar com cargas na rede mais elevadas que as restantes soluções.

A Figura 5.26 mostra a perda de pacotes em função do nível de imbricação. É interessante e importante notar que a solução baseada na infraestrutura – MIRON – com um nível de imbricação zero (`nesting0`) tem uma perda de pacotes superior à solução NEMO Basic Support Protocol. À primeira vista parece surpreendente. Contudo, a explicação para este comportamento é muito simples.



	home	nesting0	nesting1	nesting2	nesting3
nemo	0	34	178965	182943	185595
miron	0	8760	9175	11726	13731
omen	0	0	0	0	0

Figura 5.26 – Média de perda de pacotes em função do nível de imbricação

Em todas as soluções, na passagem da rede origem (`home`) para o nível de imbricação zero (`nesting0`), têm que ser criadas novas ligações otimizadas para os fluxos em curso. Isto implica a necessidade de realizar todas as operações de mobilidade.

No caso do NEMO, esta operação consiste apenas no *binding update* e *binding acknowledgement*, necessário para criar o túnel MRHA. Todas as soluções de mobilidade dependem desta tarefa para poderem ter conectividade básica.

No entanto, o MIRON tem, ainda, que realizar as operações de *return routability* e, por fim, as operações de *binding update* e *binding acknowledgement* em vez de cada um dos nós da rede móvel. É esta carga repentina que, forçosamente, tem que ser realizada pelo *router* móvel, que leva a uma enorme carga num ponto central e, consequentemente, a uma maior perda de pacotes.

Quando o nível de imbricação aumenta, então o MIRON passa a ter um comportamento melhor que o NEMO devido aos mecanismos de otimização de rotas.

Por último, os resultados voltam, uma vez mais, a comprovar que o OMEN se comporta melhor quando comparado com as restantes soluções de mobilidade de redes, Conforme explicado anteriormente, o OMEN não tem perda de pacotes para um número inferior a 90 nós da rede móvel, neste cenário.

5.3.3. Conclusões da avaliação em cenários de carga elevada

O objetivo deste conjunto de testes visou a comparação do comportamento de três soluções de mobilidade de redes representativas quando sujeitas a situações de carga intensa. As soluções em estudo foram a NEMO Basic Support Protocol, o MIRON e o OMEN que representam, respetivamente, os paradigmas baseado no equipamento antigo, baseado na rede e baseado no cliente final.

Os testes mostraram que o esforço para minimizar o impacto ao nível dos sistemas finais e tornar a mobilidade tão transparente quanto possível leva a que o NEMO tenha uma degradação acentuada do desempenho, devido à falta de otimização de rotas.

Por outro lado, os resultados mostraram que as soluções baseadas na rede, nas quais o impacto de otimização de rotas é colocado nos elementos da rede, também levam a consideráveis degradações de desempenho em situações de média a alta carga na rede.

Devido às escolhas arquiteturais do OMEN, os *routers* móveis funcionam como *routers* normais, sem estarem sujeitos ao impacto das tarefas pesadas de mobilidade de redes, tais como otimização de rotas por fluxo. Este facto leva a um desempenho muito melhor do

OMEN quando em situações de carga média e alta, tal como foi demonstrado pelos resultados das simulações. O OMEN abre um caminho claro para a criação de *routers* mais leves, em termos de processamento, consumo de memória e energia, adiados aos benefícios óbvios de desempenho.

5.4. Avaliação em cenários com redes reais sem fios

Embora exista muito trabalho de investigação na área da mobilidade de nós e de redes, a realidade é que o comportamento das soluções existentes em cenários com redes reais é largamente desconhecido.

Por outro lado, apesar de haver implementações de mobilidade IP e de mobilidade de redes, de que são exemplos SHISA [Kame08] e NEPL [Nautilus08], entre outras [USAGI08] [Tahi08], estas são vocacionadas para as soluções comummente aceites (MIPv6 e NEMO Basic Support Protocol), e não permitem avaliar e comparar as diversas propostas existentes.

Neste contexto, foi realizado um conjunto de testes cujo objetivo foi o estudo dos paradigmas de mobilidade em causa em cenários contendo redes reais sem fios. Este estudo foi realizado com recurso ao emulador mobSim, que foi adaptado para utilizar ligações sem fios IEEE 802.11 reais na comunicação entre dispositivos móveis. Como vimos, o mobSim cria e envia pacotes reais entre dispositivos virtuais, utilizando redes reais. Nestes testes foram realizadas operações de mobilidade de nós e de rede, tendo sido aplicados diversos níveis de carga na rede – desde moderado até intenso.

Os dados relativos ao comportamento dos paradigmas permitiram tirar algumas conclusões sobre o potencial de cada solução em ambientes com redes reais, tais como os que são esperados na Internet do futuro, em que se prevê uma utilização generalizada da mobilidade.

5.4.1. Cenário de simulação

A Figura 5.27 apresenta a topologia base utilizada com o objetivo de comparar o desempenho dos três paradigmas em estudo. Neste cenário, todos os *routers* foram configurados com as rotas para todos os restantes *routers*, de modo a que a comunicação entre todas as redes fosse possível e o mais curta possível.

Existem vários tipos de redes no cenário apresentado. A rede móvel de teste é constituída pelo *home agent*, representado pelo *router* `RA` na figura, pelo *router* móvel `MRA` e por um

conjunto de nós da rede móvel que, no total, poderão gerar até 500 fluxos em simultâneo. Estes fluxos são representados na figura por Fl1 até Fl500. A comunicação entre o *router* RA e o *router* móvel MRA é feita através de uma ligação sem fios real (representado na figura pelo símbolo ⌊). Existem mais quatro redes móveis, constituídos pelos pares RB/MRB, RC/MRC, RD/MRD e RE/MRE, que não contêm nenhum nó da rede móvel e que servem apenas para criar níveis de imbricação de rede (ou seja, redes móveis dentro de outras redes móveis).

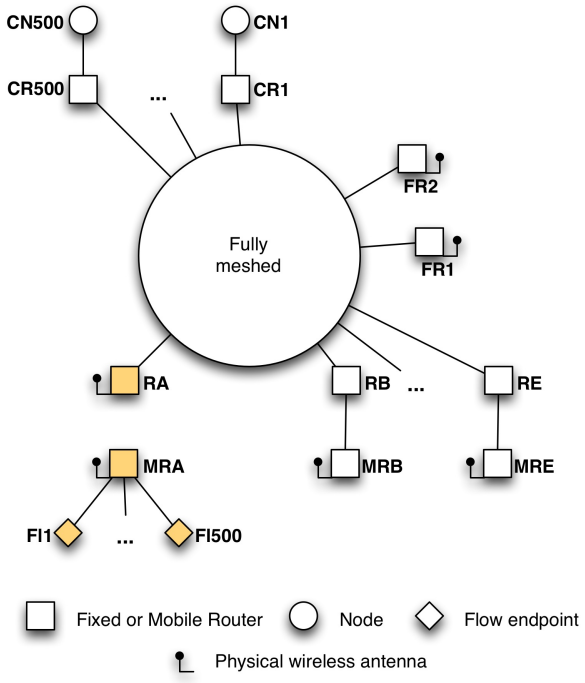


Figura 5.27 – Topologia base do cenário de simulação

No exemplo da Figura 5.28, é possível ver que a rede móvel servida pelo MRA está imbricada na rede do *router* MRE, e sucessivamente até ao *router* móvel MRB, fazendo com que a rede móvel A se encontre num nível de imbricação 4.

Os testes de mobilidade sem qualquer nível de imbricação são conseguidos com recurso ao *router* fixo FR1. Optou-se por utilizar uma rede separada para análise do comportamento da mobilidade sem imbricação para que não houvesse qualquer interferência com os restantes *routers* móveis.

Os cenários de mobilidade imbricada são construídos ao mover os *routers móveis* MRB, MRC, MRD e MRE para a rede FR2.

O movimento foi realizado da seguinte forma: o *router* móvel MRB move-se para a rede do *router* FR2, o MRC move-se para o MRB, o MRD move-se para o MRC, e o MRE move-se para o MRD.

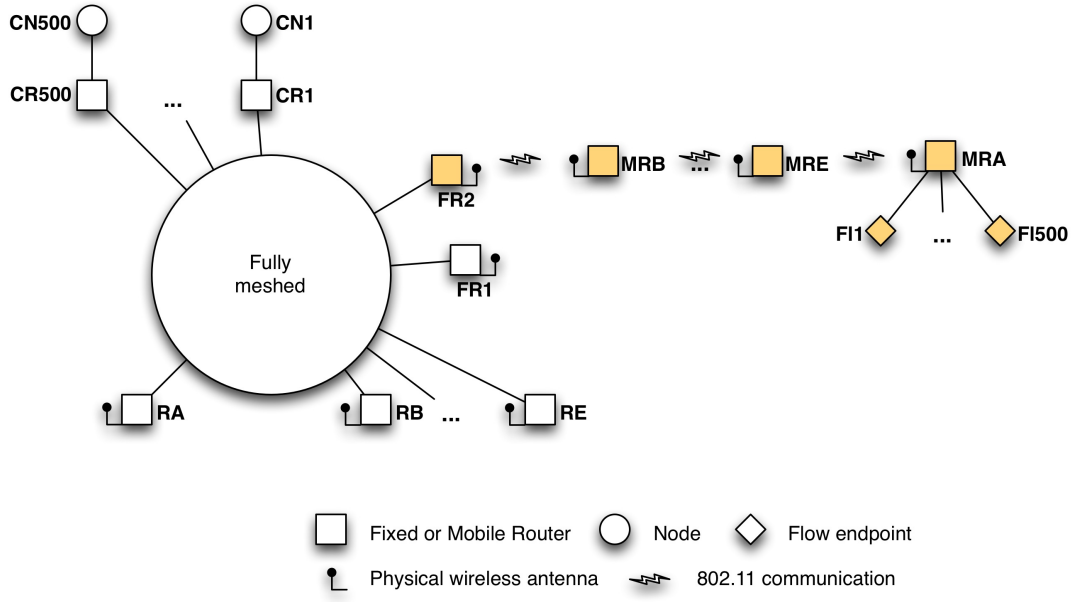


Figura 5.28 – Exemplo do cenário de emulação com 4 níveis de imbricação

O *router* móvel `MRA` vai juntar-se à rede `MRB` para obter o nível de imbricação 1. Posteriormente, junta-se à rede `MRC` para obter o nível imbricação 2, para a rede `MRD` para ficar com a imbricação 3 e, por fim, para o `MRE` para obter a imbricação 4.

A comunicação entre qualquer *router* móvel e a rede onde se encontra alojada é sempre realizada através da ligação sem fios (representada pelo símbolo).

As redes correspondentes são constituídas por um *router* correspondente (`CR1` até `CR500`) em que cada rede contém apenas um nó correspondente (`CN1` até `CN500`). De modo a evitar estrangulamentos na rede, cada nó correspondente apenas tem um fluxo proveniente da rede móvel em análise.

É importante notar que foi propositadamente removida da rede todo e qualquer tráfego externo à mobilidade, tal como tráfego de fundo, utilização de diferentes modelos de tráfego ou a imposição de pontos de estrangulamentos em determinadas partes da rede. Deste modo, os resultados apresentados derivam exclusivamente das opções arquiteturais de cada um dos paradigmas de mobilidade de rede em estudo.

Os testes que foram realizados cobrem todas as combinações dos seguintes valores dos parâmetros, para cada uma das soluções de mobilidade de redes em estudo:

- Tempo médio de intervalo entre pacotes – 50 milissegundos (ms), 100 ms, 250 ms e 500 ms; naturalmente, quanto menor for o intervalo entre pacotes maior é a carga na rede;
- Número de fluxos – 100, 200, 300, 400 e 500 fluxos;

- Rácio de otimização de rotas (RO) – 8:2 (i.e., 2 em cada 10 fluxos são otimizados), 1:1 e 2:8;
- Nível de imbricação – mobilidade sem imbricação, 1, 2, 3 e 4 níveis de imbricação.

Assim, foram realizados 300 tipos de testes distintos, para cada paradigma de mobilidade de redes. Como os testes foram realizados 3 vezes, para cada uma das soluções de mobilidade de redes, perfaz-se um total de 2 700 testes.

Para além dos valores apresentados, foram utilizados diversos parâmetros de atrasos em todas as emulações. Os valores escolhidos correspondem aos obtidos numa implementação de laboratório e que correspondem aos seguintes: tempo de DHCP – 300 ms; tempo de *return routability* – 200 ms; tempos de processamento dos pacotes HoTi, CoTi, HoT e CoT – 500 ms; tempo de estabelecimento do túnel MRHA – 10 ms; tempo de processamento das mensagens de *binding update* e *binding acknowledgement* – 10 ms.

5.4.2. Resultados de simulação

O objectivo dos vários testes, cujo resultados se apresentam aqui, foi o de estudar o comportamento dos diversos paradigmas em cenários contendo redes sem fios reais, complementando, desta forma, os resultados obtidos nas simulações apresentadas nas secções anteriores.

Foi analisada a capacidade de resposta perante a variação da carga na rede, da taxa de otimização de rotas, e do nível de imbricação.

Conforme indicado anteriormente, cada teste foi realizado 3 vezes. A escolha do melhor conjunto de testes foi feita com recurso ao Statgraphics tools[11], que fornece o teste HSD (Turkey's Honestly Significant Difference test), diagrama de dispersão e análise de variância (ANOVA).

As subsecções seguintes apresentam e discutem os resultados em detalhe.

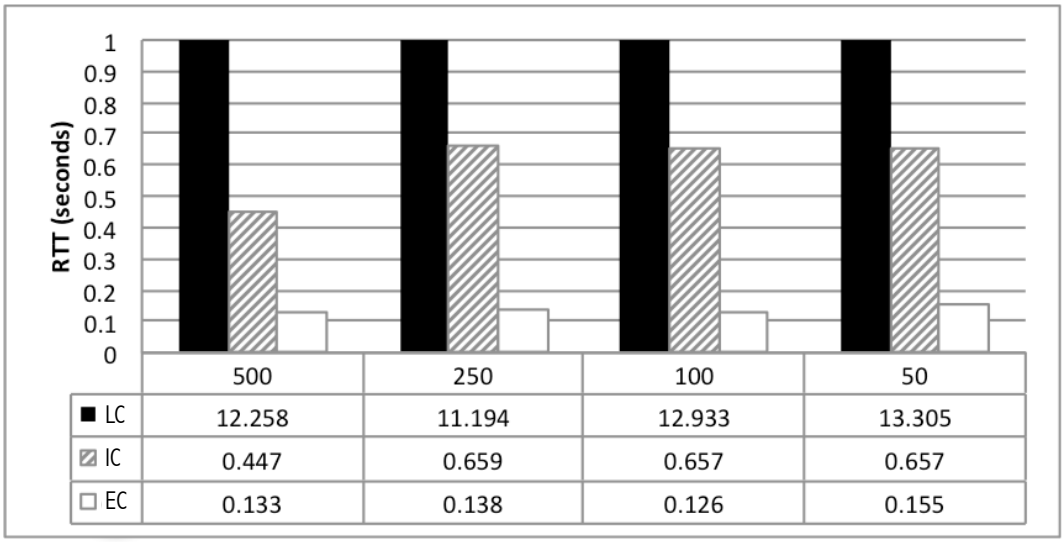
5.4.2.1 Variação de carga na rede

A variação da carga foi conseguida de duas formas distintas: com recurso à mudança do intervalo entre pacotes e, por outro lado, com a modificação do número de fluxos pacotes.

A Figura 5.29 apresenta o tempo médio de *round trip time* (RTT) para cada paradigma de mobilidade de redes em função do intervalo entre pacotes.

[11] http://www.statgraphics.com/

Note-se que o RTT médio para um determinado intervalo entre pacotes é calculado com recurso a todas as emulações realizadas para cada intervalo entre pacotes, independentemente dos restantes valores dos outros parâmetros (número de fluxos, rácio de otimização de rotas e nível de imbricação). O mesmo é aplicado aos restantes casos, para os quais, quando se analisa um determinado valor de um parâmetro, é realizada a média dos resultados de todas as emulações realizadas com o parâmetro com esse valor.



	500	250	100	50
LC	12.258	11.194	12.933	13.305
IC	0.447	0.659	0.657	0.657
EC	0.133	0.138	0.126	0.155

Figura 5.29 – Tempo médio de *round trip time* (RTT) em função do intervalo entre pacotes

É importante notar que, tal como nos casos anteriores, por motivos de legibilidade do gráfico os valores respeitantes à coluna do paradigma dos equipamentos antigos encontram-se limitados a um máximo de 1 segundo, em todas as figuras. O valor real da média RTT pode ser encontrado na secção numérica de cada figura.

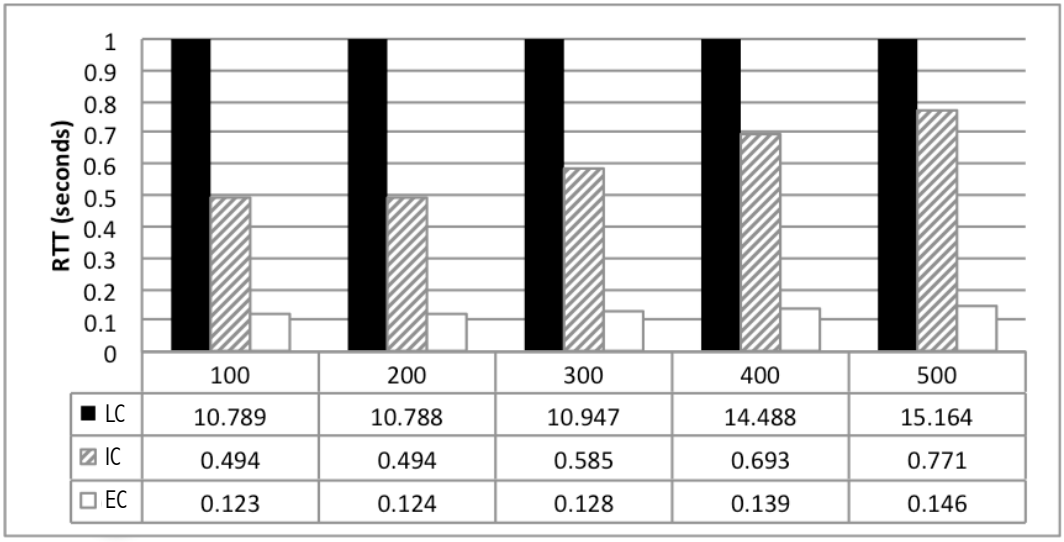
O paradigma baseado nos equipamentos antigos (LC) apresenta os piores valores de RTT, seguido do paradigma baseado na rede (IC). Por outro lado, o paradigma baseado no cliente final (EC) tem um comportamento bastante distanciado dos restantes.

No caso do paradigma baseado nos equipamentos antigos, não existe qualquer otimização de rotas pelo que, naturalmente, temos um fraco desempenho em termos de *round trip time* (RTT). Note-se que os valores muito elevados do RTT decorrem do facto de muitos cenários conterem vários níveis de imbricação. Este aspecto vai ser objeto de uma análise pormenorizada mais adiante.

Ao introduzir a otimização de rotas, o paradigma baseado na rede (IC) reduz de forma significativa a média do RTT. Conforme é incrementada a carga na rede, ou seja, conforme o

intervalo entre pacotes é decrementado, o desempenho do paradigma baseado na rede é ligeiramente afetado, mas rapidamente estabilizada devido ao factor de otimização de rotas.
Por outro lado, o desempenho de *round trip time* do paradigma baseado no cliente final (EC) é substancialmente melhor que o do paradigma baseado na rede, em todas as situações. A razão para este facto é que no caso do paradigma baseado na rede todos os fluxos são otimizados e quase toda a carga de sinalização é colocada no *routers* móveis que, deste modo, se transformam em pontos de estrangulamento.
Ao contrário, no caso do paradigma baseado no cliente final, os *routers* móveis comportam-se como simples encaminhadores de pacotes e, desta maneira, são muito menos afetados por estes níveis de carga na rede.

Outra perspectiva que, de igual modo, mostra a influência que a carga na rede pode introduzir nas soluções de mobilidade existentes é a que apresenta o *round trip time* em função ao número de fluxos introduzidos na rede, conforme se pode verificar na Figura 5.30.



	100	200	300	400	500
LC	10.789	10.788	10.947	14.488	15.164
IC	0.494	0.494	0.585	0.693	0.771
EC	0.123	0.124	0.128	0.139	0.146

Figura 5.30 – Tempo médio de *round trip time* (RTT) em função do número de pacotes

Uma vez mais, as diferenças entre os diversos paradigmas são bastante óbvias. É possível verificar a significativa degradação de desempenho do paradigma baseado na rede (IC) à medida que a carga na rede aumenta, confirmando a tese de que existe um ponto de estrangulamento.
Dado que não existe um ponto de estrangulamento na parte fixa da rede porque, conforme se explicou anteriormente, todos os fluxos vão para redes correspondentes distintas e usam caminhos separados, torna-se evidente que apenas resta o *router* móvel.

Acresce-se ainda que, utilizando a mesma topologia, o paradigma baseado no cliente final (EC) tem um comportamento significativamente superior em termos de qualidade e não apresenta sinais significativos de degradação de desempenho, confirmando deste modo que aliviando os *routers* móveis dos mecanismos de gestão da mobilidade e colocando-os no cliente final tem claras vantagens.

Como confirmação final da influência da carga na rede nos diversos paradigmas, a Figura 5.31 apresenta o número total de pacotes perdidos para os casos de 100, 300 e 500 fluxos.

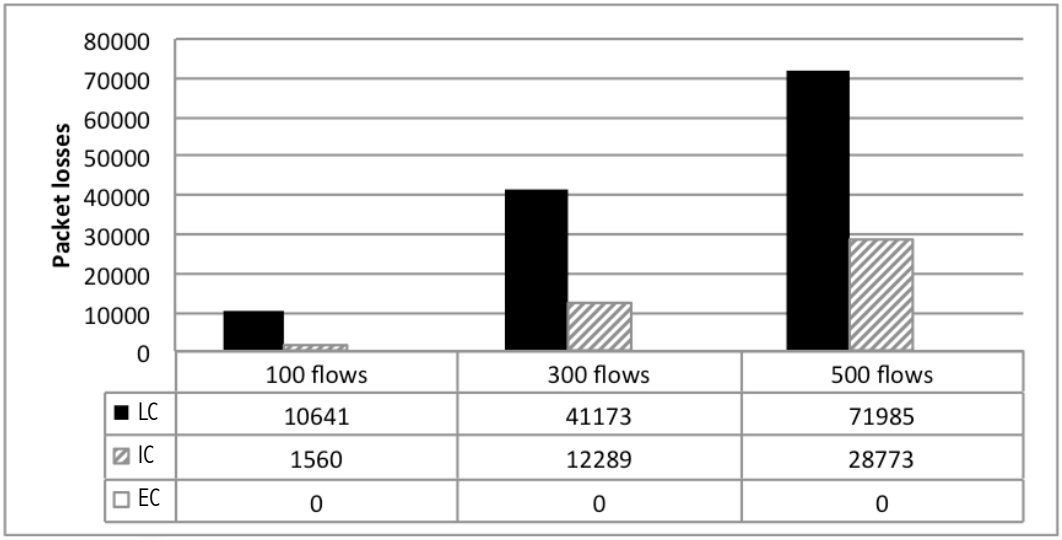


Figura 5.31 – Número total de pacotes perdidos em função do número de pacotes

Pode-se verificar um claro aumento da perda de pacotes com o aumento da carga na rede para os paradigmas baseados nos equipamentos antigos (LC) e baseado na rede (IC). De forma notável, para estes níveis de carga o paradigma baseado no cliente final (EC) não apresenta qualquer perda de pacotes.

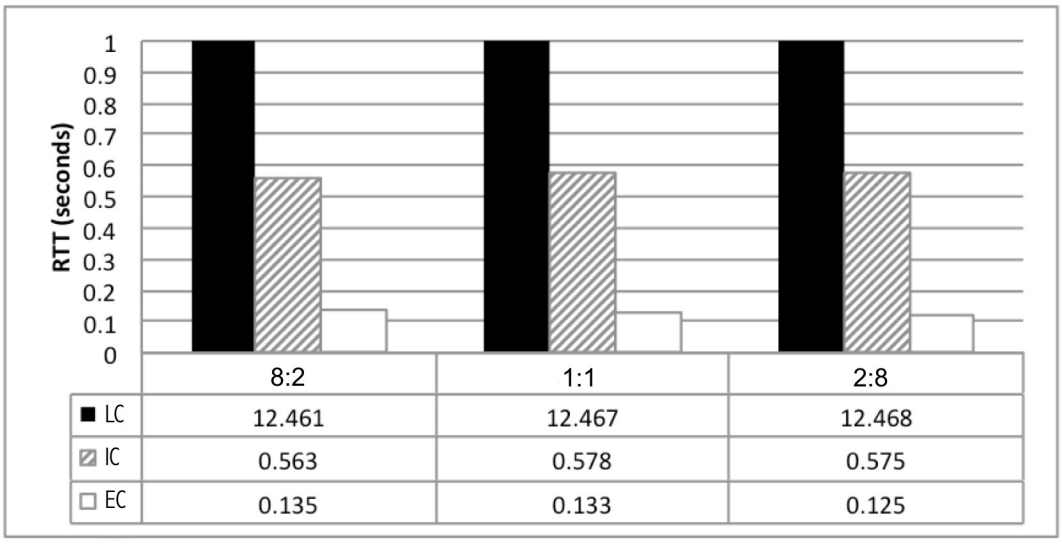
Do mesmo modo que na secção 5.3.2.5, foram realizadas emulações específicas para o paradigma baseado no cliente final de modo a determinar o número de fluxos a partir do qual se começam a verificar perda de pacotes, e foi determinado que seria a partir dos 900 fluxos.

5.4.2.2 Variação do rácio de otimização de rotas

Para a análise na perspectiva de otimização de rotas foram utilizados três rácios de otimização diferentes: `8:2`, que significa que para cada 10 fluxos, 2 utilizam uma rota otimizada; `1:1` e `2:8`.

É importante ter em conta que para os paradigmas baseados no equipamento antigo (`LC`) e baseado na rede (`IC`) os rácios de otimização de rotas não devem ter qualquer impacto, dado que não existe otimização de rotas no paradigma `LC` e, para o caso do paradigma `IC`, todos os fluxos são otimizados.

Os resultados da emulação apresentados na Figura 5.32 dizem respeito ao *round trip time* (RTT) em função do rácio de otimização de rotas.



	8:2	1:1	2:8
LC	12.461	12.467	12.468
IC	0.563	0.578	0.575
EC	0.135	0.133	0.125

Figura 5.32 – Média de *round trip time* (RTT) em função do rácio de otimização de rotas

Estes resultados confirmam que não existe uma variação significativa para os casos dos paradigmas `LC` e `IC`, conforme seria de esperar. Para além disso, como também era esperado, estes resultados confirmam que a otimização de rotas compensa, dado que o paradigma baseado na rede (`IC`) obtém tempos melhores de RTT do que o paradigma baseado nos equipamentos antigos (`LC`).

Por outro lado, os valores de *round trip time* médios para o paradigma de mobilidade de redes baseado no cliente final (`EC`) são significativamente inferiores aos do paradigma baseado na rede (`IC`).

De facto, à medida que o rácio de otimização de rotas aumenta, a média de RTT decrementa, dado que existe um maior número de fluxos que têm as suas rotas otimizadas.

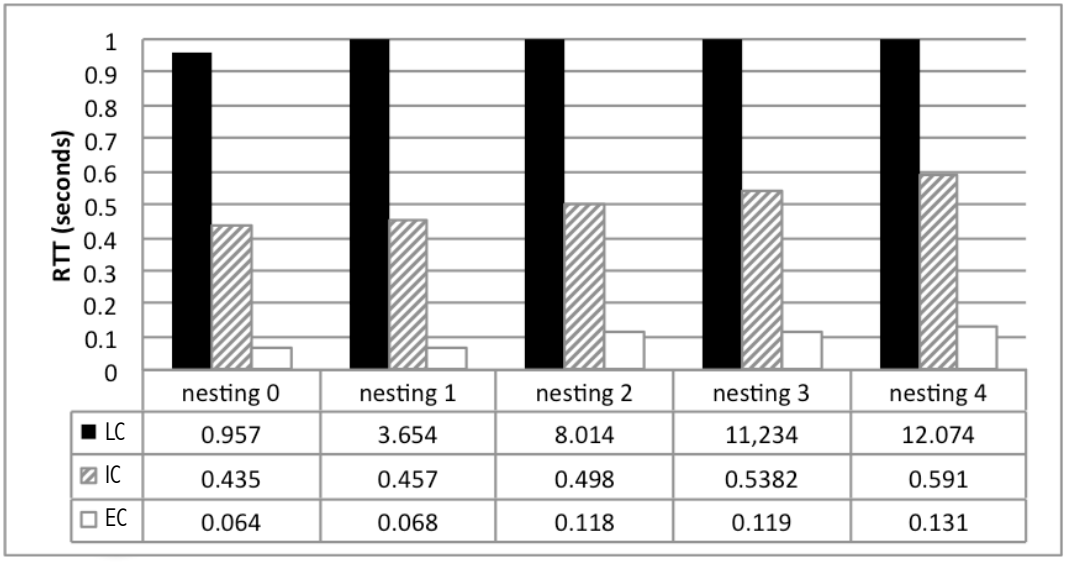
Contudo, é importante notar que o decréscimo não é muito acentuado porque, de facto, os valores de RTT já se encontram nos seus valores mínimos para o caso do paradigma baseado no cliente final (EC).

5.4.2.3 Variação do nível de imbricação

Uma análise final endereçou o comportamento dos vários paradigmas de mobilidade de redes em situações de imbricação.

No cenário de simulação, a rede móvel de referência, i.e., a rede que contém o *router* móvel MRA da Figura 5.27, tem a capacidade de se mover para uma rede fixa estrangeira ou imbricar noutra rede móvel. Quando o *router* móvel MRA se move para uma rede fixa estrangeira, então diz-se que está num nível de imbricação 0 (zero), na figura representado como nesting0. Se se mover para uma rede móvel que se encontre ligada a uma rede fixa estrangeira, então diz-se que o MRA se encontra no nível de imbricação 1 (um), representado por nesting1 na figura. O nível máximo de imbricação considerado neste estudo é 4 (quatro – nesting4).

A Figura 5.33 apresenta a média de *round trip time* (RTT) em função do nível de imbricação.



	nesting 0	nesting 1	nesting 2	nesting 3	nesting 4
LC	0.957	3.654	8.014	11,234	12.074
IC	0.435	0.457	0.498	0.5382	0.591
EC	0.064	0.068	0.118	0.119	0.131

Figura 5.33 – Média de *round trip time* (RTT) em função do nível de imbricação

O primeiro aspeto que sobressai é que existe um crescimento acentuado na média do *round trip time* (RTT) com o aumento do nível de imbricação para o caso do paradigma baseado nos equipamentos antigos. Este comportamento é, de facto, esperado e vem uma vez mais confirmar a consistência dos resultados da simulação, já que neste paradigma não existe

otimização de rotas. Se se aumentar o nível de imbricação aumenta-se o número de túneis dentro de túneis e, consequentemente, o número de redes que têm que ser atravessadas pelos pacotes. O resultado final é o incremento dramático do *round trip time*.

Outro aspecto a ter em consideração é que no caso do paradigma baseado na rede (IC) e no paradigma baseado no cliente final (EC), o nível de imbricação tem um impacto reduzido, embora perceptível, na média do *round trip time*. Ambos os paradigmas utilizam a otimização de rotas e, deste modo, assim que a otimização de rotas é estabelecida, o RTT permanece com valores baixos. Não obstante este facto, é possível notar um ligeiro incremento, que é superior para o caso do paradigma baseado na rede (IC).

A explicação para este incremento mais acentuado no caso do paradigma baseado na rede reside na quantidade de sinalização e no número de pacotes que têm que atravessar as redes sem fios. No caso do paradigma IC, todos os fluxos possuem otimização de rotas, o que se traduz num incremento de tráfego de sinalização relativa aos protocolos de mobilidade. Naturalmente, quanto maior for o nível de imbricação maior é a quantidade de redes móveis que os pacotes terão que atravessar e, deste modo, maior é o atraso. No caso do paradigma baseado no cliente final este incremento é inferior, dado que nem todas as rotas necessitam de ser otimizadas.

Como nota final, uma vez mais o paradigma de mobilidade de redes baseado no cliente final (EC) leva a resultados muito melhores quando comparado com o paradigma baseado na rede (IC), confirmando uma vez mais que, independentemente da perspectiva, o paradigma baseado no cliente final tem claras vantagens sobre o paradigma baseado na rede e, consequentemente, sobre o paradigma baseado no equipamento antigo.

5.4.3. Conclusões da avaliação em cenários com redes reais sem fios

A principal motivação para este conjunto de testes foi o estudo do comportamento de cada paradigma em cenários de mobilidade com redes reais sem fios.

De modo a comparar os paradigmas referidos, foi construído um cenário de média dimensão, com inclusão de ligações IEEE 802.11, com um número variável de fluxos. Este cenário foi implementado com recurso ao emulador mobSim e a redes sem fios reais.

Os resultados alcançados confirmam as ilações já tiradas nas sequências de testes anteriores, evidenciando o melhor desempenho da solução de mobilidade de redes proposta.

5.5. Conclusão

No presente capítulo, os paradigmas de mobilidade de rede foram sujeitos a uma extensa bateria de testes em vários tipos de cenários, recorrendo ao emulador mobSim. Foram estudados cenários de pequena dimensão, de muito grande dimensão, de carga elevada, e com redes reais sem fios. Em cada cenário, a análise foi efetuada de diversas perspetivas.

Os resultados obtidos permitiram confirmar as desvantagens já bem conhecidas do paradigma baseado nos equipamentos antigos, bastante limitado em termos de desempenho e escalabilidade. Este paradigma foi concebido a pensar nos equipamentos que não podem ser modificados mas que pretendam beneficiar da mobilidade de redes.

O paradigma baseado na rede adotou o mesmo requisito base do paradigma baseado nos equipamentos antigos: evitar toda e qualquer alteração nos nós finais. Como consequência, quase todas as funções de mobilidade são realizadas pelos elementos de rede – maioritariamente, pelo *router* móvel. Os resultados das simulações relativos a este paradigma mostram claramente que o desempenho dos elementos de rede é afetado por esta concentração de funcionalidades, traduzindo-se em efeitos como o aumento do *round trip time* resultante ou uma maior perda global de pacotes.

Os estudos efetuados demonstram que o paradigma baseado no cliente final tem um comportamento superior em relação aos outros dois paradigmas e que tem um bom potencial. Este comportamento mantém-se mesmo quando sujeito a uma elevada carga de tráfego. Como as tarefas relacionadas com a mobilidade estão a cargo dos nós da rede móvel, isto leva a que os *routers* móveis fiquem mais aliviados, o que se traduz numa melhoria global de desempenho.

6. Conclusão

Motivado pela crescente necessidade de um acesso à Internet ubíquo e ininterrupto, foram desenvolvidas diversas soluções de mobilidade de dispositivos e de redes. O protocolo de mobilidade de nós conta com uma solução completa e com boa aceitação por parte da comunidade científica. Contudo, o mesmo não se pode dizer em relação à mobilidade de redes, que apenas possui uma solução básica de mobilidade sem qualquer mecanismo de otimização de rotas.

Nesta tese foram apresentadas diversas limitações da solução para a mobilidade de redes desenvolvida pelo IETF, designada NEMO Basic Support Protocol, tal como foram apresentadas as soluções mais relevantes para resolver os problemas do RFC 3963. Não obstante a grande variedade de propostas existentes, nenhuma reúne consenso para uma utilização de âmbito global na Internet, havendo, desta forma, espaço para um estudo aprofundado com o objetivo de colmatar esta lacuna.

O presente capítulo apresenta as contribuições do trabalho realizado no âmbito da mobilidade de redes e identifica linhas para trabalho futuro.

6.1. Contribuições

A primeira contribuição nasce diretamente do estudo do estado da arte, no âmbito do qual foi necessário agrupar as diversas soluções relativas à mobilidade de redes de modo a tornar-se mais legível a sua possível comparação.

A partir desta necessidade, foram criados grupos baseados nos aspetos arquiteturais comuns a cada solução de mobilidade de redes, designados paradigmas. Foram identificados os seguintes: paradigma baseado nos equipamentos antigos, que possibilita o acesso imediato à mobilidade de redes sem necessidade de alterações nos nós finais; paradigma baseado na rede, de acordo com o qual o esforço da mobilidade de redes fica a cargo dos elementos da infraestrutura, libertando os nós da rede móvel destas tarefas; e, por último, o paradigma baseado no cliente, no qual os nós da rede móvel conhecem a sua condição de mobilidade e têm a seu cargo as funções daí decorrentes.

Com base neste conceito, tornou-se possível a realização de comparações entre os três paradigmas definidos, com base na comparação de propostas/soluções representativas de cada grupo.

O paradigma baseado no cliente foi proposto e desenvolvido no âmbito do presente trabalho. Embora se possam encontrar na literatura algumas menções à possibilidade de utilizar o cliente final como elemento principal na mobilidade de redes, nunca antes tinha sido apresentada nenhuma proposta nesse sentido digna de registo. De facto, a ideia subjacente à viabilidade da implementação da mobilidade de redes ao nível do cliente final, conjugada com o valor acrescido da utilização deste paradigma quando comparado com os restantes, constitui a contribuição central desta tese.

A seguir é sumarizado o trabalho realizado no que diz respeito à proposta OMEN, que está enquadrada no paradigma baseado no cliente.

6.1.1. OMEN

O ponto central da proposta OMEN reside na mudança da responsabilidade de execução dos procedimentos de gestão de mobilidade. Enquanto o paradigma baseado nos equipamentos antigos e o paradigma baseado na rede advogam a necessidade de não modificar os dispositivos finais a todo o custo, o paradigma baseado no cliente vai no sentido oposto, colocando o ónus da otimização de rotas nos dispositivos finais.

Partindo do pressuposto de que os nós da rede móvel devem estar conscientes da sua condição de mobilidade, o OMEN propõe que seja utilizado o *router* móvel como sistema que anuncia essa mobilidade. Assim que os clientes são informados de que ocorreu uma mudança do ponto de ligação à Internet da rede em que se encontram, podem realizar operações de otimização de rotas, se assim o entenderem, utilizando o *care-of address* do *router* móvel.

A proposta OMEN visa a não modificação dos protocolos existentes. Deste modo, foram encetados esforços no sentido de utilizar os protocolos MIPv6, *Neighbor Discovery* e NEMO Basic Support Protocol sem que fosse necessário proceder a qualquer modificação nos respetivos RFC. De facto, as modificações concentram-se apenas ao nível dos dispositivos finais e do *router* móvel, e consistem apenas na tomada de consciência da condição de mobilidade e consequente utilização desta nova funcionalidade utilizando os protocolos já existentes.

Esta mudança de paradigma torna possível a determinação da altura ideal para a realização do procedimento de otimização de rotas, dado que agora é o dispositivo final a tomar esta decisão. Como consequência, o número de ligações otimizadas pode diminuir drasticamente,

pois cada nó da rede móvel apenas otimizará a rota das ligações que realmente necessitem desta funcionalidade, minimizando o esforço que daí advém.

Por outro lado, a passagem do esforço de otimização de rotas para os dispositivos finais leva a que a infraestrutura se torne mais leve e, consequentemente, com melhor desempenho na comunicação. Também é importante notar que um elemento de rede mais leve passa necessariamente a consumir menos energia. Em ambientes de mobilidade as questões de consumo de energia assumem um papel preponderante.

Concomitantemente, o OMEN conduz a uma diminuição de perda de pacotes, pois diminui a sobrecarga dos elementos de rede, já que reduz o esforço relacionado com os procedimentos de otimização de rotas. Isto leva a um incremento na qualidade da comunicação e a um menor impacto introduzido na infraestrutura global.

Outro aspeto positivo da solução OMEN é o facto de ser complementar, podendo coexistir com outras soluções de mobilidade de redes. Esta possibilidade abre caminho a uma implementação gradual e sustentada, mediante a atualização dos dispositivos que venham a beneficiar desta nova funcionalidade. Assim, torna-se possível beneficiar de soluções intermédias que contemplem, por exemplo, a otimização de rotas implementadas ao nível da infraestrutura de rede.

O comportamento perante este novo paradigma vai depender do tipo de nó da rede móvel. Foram apresentadas as situações dos nó local fixo antigo, nó local fixo com suporte de MIPv6, nó local móvel, nó móvel visitante e NEMO imbricada. Para cada uma destas situações foram apresentados os procedimentos de implementação da mobilidade de redes.

As questões de segurança também foram analisadas, com particular enfoque nos problemas do envio do *Care-of address* através do RFC 4861, e no registo de um nó móvel visitante e de uma NEMO imbricada.

Foram realizados diversos testes que permitiram analisar a viabilidade da solução, assim como comparar as diversas soluções em estudo. Os testes foram realizados utilizando a ferramenta de emulação mobSim, desenvolvida no âmbito deste trabalho.

O primeiro conjunto de testes visou a validação funcional da proposta OMEN e a confirmação das suas principais características.

O segundo conjunto de testes visou uma comparação dos diversos paradigmas em ambientes de muita larga escala, tornando-se visível o impacto dos procedimentos de mobilidade de redes nos *routers* móveis. Este ambiente contou com 22.800 *routers*, 11.250 redes e mais de 27.000 dispositivos finais (fixos e móveis). Apesar da sua larga escala, não foram realizados testes de stress na rede, pelo que o impacto se deveu somente ao elevado número de elementos. Os resultados deste conjunto de testes podem ser resumidos no seguinte:

- O paradigma baseado no equipamento antigo tem diversas limitações de desempenho, não sendo adequado para implementação em larga escala;
- Já o paradigma baseado na rede é fortemente afetado, em termos de desempenho, pelas operações de otimização de rotas que, quando em grande número, levam a uma degradação acentuada da qualidade de comunicação entre os nós;
- Por fim, o paradigma baseado no cliente mostrou ter um potencial elevado devido ao impacto diminuto na infraestrutura de rede e, também, nas comunicações entre os equipamentos finais.

O conjunto seguinte de testes visou o estudo comparativo da proposta em ambientes de carga elevada. Nestes ambientes foram analisados os comportamentos dos diversos paradigmas quando sujeitos a vários níveis de stress. Os testes foram realizados de forma a não existirem pontos de congestionamento ou tráfego externo que pudessem influenciar os resultados. Este ambiente contou com um número reduzido de elementos, até um máximo de 50 nós da rede móvel. Os resultados permitiram concluir que:

- O NEMO Basic Support Protocol tem um comportamento muito negativo quando sujeito a ambientes de comunicação intensiva; de facto, o esforço para tornar a comunicação móvel transparente para os nós finais tem um preço demasiado elevado e proibitivo em ambientes de alguma carga de tráfego;
- O paradigma baseado na rede também é significativamente afetado pela carga na rede, traduzindo-se em consideráveis degradações de desempenho quando em situações de média e alta carga na rede; nestas situações é, também, patente uma elevada taxa de perda de pacotes;
- Já o paradigma baseado no cliente leva a que os *routers* se comportem como simples encaminhadores de pacotes e, deste modo, não estão sujeitos à carga adicional imposta pelas operações de mobilidade de redes; em situações de stress na rede, este paradigma tem um comportamento notável verificando-se, inclusivamente, que a perda de pacotes foi nula até um elevado número de nós da rede móvel.

O último conjunto de testes foi realizado em cenários de emulação que incluíam redes reais sem fios, IEEE 802.11, tornando possível a comparação dos paradigmas em estudo em situações tão próximas da realidade quando possível. Estas emulações visaram a análise em situações de stress moderado, para um número de fluxos não muito elevado – até um máximo de 500 fluxos a atravessar a rede móvel. Os resultados permitiram concluir que:

- Uma vez mais, o NEMO Basic Support Protocol tem um mau comportamento quando sujeito a situações de alguma carga;

- O paradigma baseado na rede é negativamente afetado pela concentração das operações de mobilidade de rede nos *routers* móveis e pela perda de pacotes, agravadas pelas comunicações sem fios;
- Por fim, o paradigma baseado no cliente tem um comportamento superior em relação aos restantes, mantendo-se com bom desempenho mesmo quando sujeito a ambientes de carga elevada; o facto de distribuir a carga das operações de mobilidade pelos nós da rede móvel leva a que haja um menor impacto na rede e, consequentemente, melhor desempenho da comunicação.

A realização dos testes referidos apenas foi possível com a utilização de uma ferramenta de emulação concebida para o efeito, no âmbito deste trabalho. Os aspetos principais dessa ferramenta são brevemente apresentados seguidamente.

6.1.2. mobSim

A falta de suporte nativo de soluções de mobilidade de redes e a falta de capacidade de escalabilidade por parte dos simuladores existentes ditou a necessidade de criação de um novo emulador. Os simuladores analisados em detalhe foram o ns-2, o ns-3, o OMNet++ e o OPNET Modeler.

O mobSim foi criado com vista a integrar, de forma nativa, o suporte de mobilidade de redes dos paradigmas baseado no equipamento antigo, baseado na rede e baseado no cliente. Por outro lado, este emulador contemplou o suporte de cenários que vão desde pequena escala até uma escala muito elevada.

Para a sua correta execução, o mobSim teve que ser construído de modo a respeitar os RFC 2460, RFC 6275, RFC 4861 e RFC 3963. Como requisitos de concepção, o mobSim teve que permitir acesso a detalhes de simulação muito finos, ser flexível em termos de definição de cenários – especificação das redes fixas e móveis, equipamentos finais fixos e móveis, topologias e comportamentos dinâmicos –, e permitir a utilização dos mesmos parâmetros e condições para as diversas soluções em análise.

Este novo emulador foi desenvolvido para tirar proveito dos ambientes paralelos dos *clusters*, beneficiando das ligações de alta velocidades entre os diversos nós físicos. Para cada nó físico atribuído à simulação é definido uma funcionalidade: ou nó principal (*master*), ou nó operacional (*slave*). Um nó físico é eleito o nó principal, sendo os restantes definidos como nós operacionais.

O nó principal é responsável pela construção dos equipamentos virtuais, ambientes e simulações a serem executados em todos os nós operacionais. Este nó também é

responsável por obter os resultados das simulações bem como terminar a execução das tarefas em todo o ambiente de teste.

Este emulador foi concebido para ser executado em *clusters*, em que as tarefas entram numa fila de processos ficando à espera de execução, normalmente em ambiente de processamento em batch. A sua execução não contempla, por isso, a possibilidade de intervenção humana. Deste modo, o nó principal tem o papel crucial de garante da boa execução de toda a simulação.

6.2. Trabalho futuro

O trabalho apresentado nesta tese abre um novo e abrangente campo de investigação futura. Naturalmente, novas soluções assentes no paradigma baseado no cliente deverão ser desenvolvidas. Nesta tese foi apresentada a proposta OMEN, mas muitas outras soluções de mobilidade baseada no cliente podem, e devem, ser desenvolvidas e analisadas. O estudo dos seus comportamentos em diferentes condições de rede e mobilidade reveste-se de particular iinteresse.

Por outro lado, mais que simulações ou mesmo emulações, devem ser desenvolvidas implementações, que deverão ser analisadas e implementadas em ambientes reais.

Para além disso, atualmente o mobSim está vocacionado para o estudo comparativo dos três paradigmas alvo da presente dissertação e exige muito trabalho de configuração de "baixo nível". Dado o interesse despertado por esta ferramenta na comunidade, seria muito interessante desenvolver este emulador por forma a torná-lo mais geral e abrangente, tendo em vista a sua disponibilização à comunidade científica. É um trabalho de grande envergadura mas, também, de grande interesse.

Numa ótica diferente, começa a surgir alguma apetência pelo estudo de soluções de mobilidade em camadas protocolares distintas da camada de rede, em particular na camada de transporte e, até, na camada de aplicação. O desenvolvimento de soluções deste tipo e a sua comparação com soluções na camada de rede são aspetos que merecem atenção.

Anexos

A. Cenário de larga escala

Os seguintes excertos apresentam a configuração do cenário de simulação utilizado na secção 5.2, com recurso à ferramenta mobSim. Esta configuração permite visualizar os *routers* de topo, os *routers* dos diversos níveis que foram criados e, também, os dispositivos finais utilizados. É importante referir que foram criados diversos cenários com variação de número de dispositivos finais e respetiva localização na rede virtual. Este excerto apenas mostra um dos cenários utilizado e serve para dar uma ideia do que foi realizado.

Dada a extensão da configuração, resolveu-se dividi-la em duas partes: uma dedicada à configuração dos *routers* de topo, apresentada na secção A.1., e outra dedicada apenas à configuração das redes abaixo do *router* de topo `router11`, apresentado na secção A.2.

A.1. *Routers* de topo

```
# Top Level Network
toprouter|routera
router|routera1;gateway|routera
router|routera2;gateway|routera
router|routera3;gateway|routera
router|routera4;gateway|routera
router|routera5;gateway|routera
router|routera11;gateway|routera1;extracode|HA.extracode
router|routera12;gateway|routera1;extracode|HA.extracode
router|routera13;gateway|routera1
router|routera21;gateway|routera2
router|routera22;gateway|routera2
router|routera23;gateway|routera2
router|routera31;gateway|routera3
router|routera32;gateway|routera3
router|routera33;gateway|routera3
router|routera41;gateway|routera4
router|routera42;gateway|routera4
router|routera43;gateway|routera4
router|routera51;gateway|routera5
router|routera52;gateway|routera5
router|routera53;gateway|routera5
```

A.2. Rede do *router11*

```
################################################################################
#######
router|routera111;gateway|routera11
router|routera112;gateway|routera11
mobilerouter|mra111;gateway|routera11;extracode|mr.extracode
multiplenode:1000|mmnna111;gateway|mra111;prog|mobsim_1port_udp.pl
multiplenode:1000|mmnnam111;gateway|mra111;prog|mobsim_1port_udp.pl;\
       extracode|mipv6.extracode
router|routera121;gateway|routera12
router|routera122;gateway|routera12
router|mra121;gateway|routera12
multiplenode:1|mmnna121;gateway|mra121;prog|mobsim_1port_udp.pl
router|routera131;gateway|routera13
router|routera132;gateway|routera13
router|mra131;gateway|routera13
multiplenode:1|mmnna131;gateway|mra131;prog|mobsim_1port_udp.pl

#''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
router|routera1111;gateway|routera111
router|routera1112;gateway|routera111
router|mra1111;gateway|routera111
multiplenode:1|mmnna1111;gateway|mra1111;prog|mobsim_1port_udp.pl
router|routera1121;gateway|routera112
router|routera1122;gateway|routera112
router|mra1121;gateway|routera112
multiplenode:1|mmnna1121;gateway|mra1121;prog|mobsim_1port_udp.pl

router|routera1211;gateway|routera121
router|routera1212;gateway|routera121
router|mra1211;gateway|routera121
multiplenode:1|mmnna1211;gateway|mra1211;prog|mobsim_1port_udp.pl
router|routera1221;gateway|routera122
router|routera1222;gateway|routera122
router|mra1221;gateway|routera122
multiplenode:1|mmnna1221;gateway|mra1221;prog|mobsim_1port_udp.pl

router|routera1311;gateway|routera131
router|routera1312;gateway|routera131
router|mra1311;gateway|routera131
multiplenode:1|mmnna1311;gateway|mra1311;prog|mobsim_1port_udp.pl
router|routera1321;gateway|routera132
router|routera1322;gateway|routera132
router|mra1321;gateway|routera132
multiplenode:1|mmnna1321;gateway|mra1321;prog|mobsim_1port_udp.pl

#''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
router|routera11111;gateway|routera1111
router|routera11112;gateway|routera1111
router|mra11111;gateway|routera1111
multiplenode:1|mmnna11111;gateway|mra11111;prog|mobsim_1port_udp.pl
router|routera11121;gateway|routera1112
router|routera11122;gateway|routera1112
router|mra11121;gateway|routera1112
multiplenode:1|mmnna11121;gateway|mra11121;prog|mobsim_1port_udp.pl

router|routera11211;gateway|routera1121
router|routera11212;gateway|routera1121
router|mra11211;gateway|routera1121
multiplenode:1|mmnna11211;gateway|mra11211;prog|mobsim_1port_udp.pl
router|routera11221;gateway|routera1122
router|routera11222;gateway|routera1122
router|mra11221;gateway|routera1122
multiplenode:1|mmnna11221;gateway|mra11221;prog|mobsim_1port_udp.pl

router|routera12111;gateway|routera1211
router|routera12112;gateway|routera1211
router|mra12111;gateway|routera1211
multiplenode:1|mmnna12111;gateway|mra12111;prog|mobsim_1port_udp.pl
router|routera12121;gateway|routera1212
router|routera12122;gateway|routera1212
router|mra12121;gateway|routera1212
multiplenode:1|mmnna12121;gateway|mra12121;prog|mobsim_1port_udp.pl
```

```
router|routera12211;gateway|routera1221
router|routera12212;gateway|routera1221
router|mra12211;gateway|routera1221
multiplenode:1|mmnna12211;gateway|mra12211;prog|mobsim_1port_udp.pl
router|routera12221;gateway|routera1222
router|routera12222;gateway|routera1222
router|mra12221;gateway|routera1222
multiplenode:1|mmnna12221;gateway|mra12221;prog|mobsim_1port_udp.pl

router|routera13111;gateway|routera1311
router|routera13112;gateway|routera1311
router|mra13111;gateway|routera1311
multiplenode:1|mmnna13111;gateway|mra13111;prog|mobsim_1port_udp.pl
router|routera13121;gateway|routera1312
router|routera13122;gateway|routera1312
router|mra13121;gateway|routera1312
multiplenode:1|mmnna13121;gateway|mra13121;prog|mobsim_1port_udp.pl

router|routera13211;gateway|routera1321
router|routera13212;gateway|routera1321
router|mra13211;gateway|routera1321
multiplenode:1|mmnna13211;gateway|mra13211;prog|mobsim_1port_udp.pl
router|routera13221;gateway|routera1322
router|routera13222;gateway|routera1322
router|mra13221;gateway|routera1322
multiplenode:1|mmnna13221;gateway|mra13221;prog|mobsim_1port_udp.pl

#$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
router|mra111111;gateway|routera11111
multiplenode:1|mmnnam111111;gateway|mra111111;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra111121;gateway|routera11112
multiplenode:1|mmnnam111121;gateway|mra111121;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra111211;gateway|routera11121
multiplenode:1|mmnnam111211;gateway|mra111211;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra111221;gateway|routera11122
multiplenode:1|mmnnam111221;gateway|mra111221;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra112111;gateway|routera11211
multiplenode:1|mmnnam112111;gateway|mra112111;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra112121;gateway|routera11212
multiplenode:1|mmnnam112121;gateway|mra112121;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra112211;gateway|routera11221
multiplenode:1|mmnnam112211;gateway|mra112211;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra112221;gateway|routera11222
multiplenode:1|mmnnam112221;gateway|mra112221;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode

router|mra121111;gateway|routera12111
multiplenode:1|mmnnam121111;gateway|mra121111;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra121121;gateway|routera12112
multiplenode:1|mmnnam121121;gateway|mra121121;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra121211;gateway|routera12121
multiplenode:1|mmnnam121211;gateway|mra121211;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra121221;gateway|routera12122
multiplenode:1|mmnnam121221;gateway|mra121221;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra122111;gateway|routera12211
multiplenode:1|mmnnam122111;gateway|mra122111;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra122121;gateway|routera12212
multiplenode:1|mmnnam122121;gateway|mra122121;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra122211;gateway|routera12221
multiplenode:1|mmnnam122211;gateway|mra122211;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra122221;gateway|routera12222
multiplenode:1|mmnnam122221;gateway|mra122221;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
```

```
router|mra131111;gateway|routera13111
multiplenode:1|mmnnam131111;gateway|mra131111;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra131121;gateway|routera13112
multiplenode:1|mmnnam131121;gateway|mra131121;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra131211;gateway|routera13121
multiplenode:1|mmnnam131211;gateway|mra131211;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra131221;gateway|routera13122
multiplenode:1|mmnnam131221;gateway|mra131221;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra132111;gateway|routera13211
multiplenode:1|mmnnam132111;gateway|mra132111;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra132121;gateway|routera13212
multiplenode:1|mmnnam132121;gateway|mra132121;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra132211;gateway|routera13221
multiplenode:1|mmnnam132211;gateway|mra132211;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
router|mra132221;gateway|routera13222
multiplenode:1|mmnnam132221;gateway|mra132221;prog|mobsim_1port_udp.pl;\
        extracode|mipv6.extracode
#$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

B. *Script* create_ipaddresseslst.pl

O *script* create_ipaddresseslst.pl consiste no componente essencial para a criação de dispositivos virtuais, associar endereços IP virtuais aos da rede física, e construir o *routing* de todo o cenário emulado.

O *script* começa por obter a lista de servidores disponíveis no *cluster*.

```
# Now we try to find which servers we are going to use
open AX, "servers.lst";
while (<AX>) {
        chop;
        $servers{$_}=1;
        $myipaddris=$_ unless ($myipaddris);
}
close AX;
```

De seguida, obtém o *template* que vai ser utilizado por todos os equipamentos virtuais.

```
# We will use this template for all virtual devices
open AX, "$template";
while (<AX>) {
        chop;
        split(/=/);
        $template{$_[0]}=$_[1];
}
close AX;
```

Posteriormente, vai buscar os ficheiros que estão disponíveis no cenário.

```
# Getting scenarios
print "Reading scenario: ";
open AX, "$f/$s/$file";
```

```
while (<AX>) {
        chop;
        s/#.*//;
        s/^\t//;
        next unless ($_);

        s/^([^\|]+)\|([^;]+);?//;
        $type=$1;
        $name=$2;
        if ($name=~/\_/) {
                print "ERROR: Node name \"$name\" at Line \"$_\" has a \
                        reserved word: \"_\"\n";
                print "Exiting!\n";
                exit;
        }

        foreach $k (keys %template) {
                $myinfo{$k}=$template{$k};
        }

        @entries=split(/;/);
        foreach $k (@entries) {
                @k=split(/[\|=]/,$k);
                $myinfo{$k[0]}=$k[1];
                delete($myinfo{$k[0]}) unless ($k[1]);
        }

        # Specific sections
        if ($type eq "toprouter") {
                $iptop++;
                $myinfo{"ipaddr2"}="TOP:$iptop";
                if ($ip{$myinfo{"ipaddr2"}}) {
                        print "Already exists this IP: $myinfo{ipaddr2}\n";
                        exit;
                }
                $ip{$myinfo{"ipaddr2"}}=1;

                $netname{$name}=$name;
                $myinfo{"ipaddr1"}="$netname{$name}:$name";
                if ($ip{$myinfo{"ipaddr1"}}) {
                        print "Already exists this IP: $myinfo{ipaddr1}\n";
                        exit;
                }
                $ip{$myinfo{"ipaddr1"}}=1;

                $myinfo{"delay_ha"}=200000 unless ($myinfo{"delay_ha"});
                $myinfo{"delay_mrha"}=100000 unless ($myinfo{"delay_mrha"});

                $toprouting{$name}="$netname{$name}::$myinfo{ipaddr2}";
                $toprouter{$name}=$name;
                $toprouterip{$name}="TOP:$iptop";
        }

        if ($type =~ m"^(mobile)?router$") {
                unless ($netname{$myinfo{"gateway"}}) {
                        print "Router network acquiring problem: unknown \
                                gateway: $myinfo{gateway}\n";
                        exit;
                }
                $myinfo{"ipaddr2"}="$netname{$myinfo{gateway}}:$name";
                if ($ip{$myinfo{"ipaddr2"}}) {
                        print "Already exists this IP: $myinfo{ipaddr2}\n";
                        exit;
                }
                $ip{$myinfo{"ipaddr2"}}=1;

                $netname{$name}=$name;
                $myinfo{"ipaddr1"}="$netname{$name}:$name";
                if ($ip{$myinfo{"ipaddr1"}}) {
                        print "Already exists this IP: $myinfo{ipaddr1}\n";
                        exit;
                }
                $ip{$myinfo{"ipaddr1"}}=1;

                $myinfo{"delay_ha"}=200000 unless ($myinfo{"delay_ha"});
                $myinfo{"delay_mrha"}=100000 unless ($myinfo{"delay_mrha"});

                unless ($toprouter{$myinfo{"gateway"}}) {
```

```
                print "Problem building the routing table... no top router \
                        for: $myinfo{gateway} at router $name\n";
                exit;
        }
        $toprouter{$name}=$toprouter{$myinfo{"gateway"}};
        $toprouting{$toprouter{$myinfo{"gateway"}}}.=";$netname{$name}::\
                $toprouterip{$toprouter{$myinfo{gateway}}}";
        $subrouting{$myinfo{"gateway"}}.=";$netname{$name}::\
                $netname{$myinfo{gateway}}:$name";
}

if ($type eq "mobilerouter") {
        $myinfo{"HA"}="$myinfo{gateway}:$myinfo{gateway}";
        $myinfo{"HoA"}=$myinfo{"ipaddr2"};
        $myinfo{"delay_handoff"}=500000;
        $myinfo{"delay_mrha"}=100000;
        $myinfo{"delay_bu"}=300000;

        $mobilerouter{$name}=$myinfo{"ipaddr2"};
}

if ($type eq "node" or $type=~/^multiplenode:/) {
        unless ($netname{$myinfo{"gateway"}}) {
                print "Node network acquiring problem: unknown \
                        gateway: $myinfo{gateway}\n";
                exit;
        }
        $myinfo{"ipaddr1"}="$netname{$myinfo{gateway}}:$name";
        if ($ip{$myinfo{"ipaddr1"}}) {
                print "Already exists this IP: $myinfo{ipaddr1}\n";
                exit;
        }
        $ip{$myinfo{"ipaddr1"}}=1;
        $node{$myinfo{"ipaddr1"}}=1;

        $myinfo{"extracode"}="cn.extracode" unless ($myinfo{"extracode"});
}

# Let get some important info for aquiring IPs
if ($toprouter{$myinfo{"gateway"}}) {
        $disttoprouter{$name}=$toprouter{$myinfo{"gateway"}};
} elsif ($toprouter{$name}) {
        $disttoprouter{$name}=$toprouter{$name};
} elsif ($name eq "00logs") {
        # Here is for logs server... we wont be processing this!
} else {
        print "Problem aquiring top router: $name\n";
        exit;
}
$disttoprouter{$myinfo{"ipaddr1"}}=$disttoprouter{$name} \
        if ($myinfo{"ipaddr1"});
$disttoprouter{$myinfo{"ipaddr2"}}=$disttoprouter{$name} \
        if ($myinfo{"ipaddr2"});

# Change IP of gateway to its correct IP...
$myinfo{"gateway"}.=":$myinfo{gateway}" if ($myinfo{"gateway"});

# Output results...
foreach $k (keys %myinfo) {
        if ($type=~/^multiplenode:(\d+)/) {
                my $nr=$1;
                if ($k=~/^ipaddr/) {
                        for($i=1;$i<=$nr;$i++) {
                                my $aux=$myinfo{$k}."-$i";
                                if ($ip{$aux}) {
                                        print "Already exists this IP: $aux\n";
                                        exit;
                                }
                                $ip{$aux}=1;
                                $node{$aux}=1;
                                ${$aux}=1;

                                $disttoprouter{$aux}=$disttoprouter{$name};

                                $final{$name."-$i"}.="$k=$aux\n";
                        }
                } else {
                        for($i=1;$i<=$nr;$i++) {
```

```
                                        $final{$name."-$i"}.="$k=$myinfo{$k}\n";
                                }
                        }
                } else {
                        $final{$name}.="$k=$myinfo{$k}\n";
                }
                delete($myinfo{$k});
        }
        $name=""; $type="";
}
close AX;
print "Done!\n";
```

A seguir é obtida a listagem dos movimentos dos dispositivos móveis.

```
print "Getting MR's moves: ";
open AX, "$scriptfile";
while (<AX>) {
        next unless (/^move;[^;]+;(.*)\n$/);
        $mrmove{$1}=1;
}
close AX;
print "Done!\n";
```

Na posse de toda a informação que necessita, o *script* procede à criação dos ficheiros de configuração dos dispositivos virtuais.

```
# Create files of final devices
print "Creating files: ";
foreach $k (keys %final) {
        open AX, ">$f/$paradigm/$k\.conf";
        print AX $final{$k};
        $routing="";

        if ($toprouting{$k}) {
                foreach $j (keys %toprouting) {
                        next if ($j eq $k);
                        $routing.="$toprouting{$j};";
                }
        }
        if ($subrouting{$k}) {
                $subrouting{$k}=~s/^;//;
                $routing.=$subrouting{$k};

                my @r=split(/;/,$subrouting{$k});
                foreach my $sr (@r) {
                        my @r1=split(/::/,$sr);
                        if ($subrouting{$r1[0]}) {
                                $routing.=addsub($sr,$r1[1]);
                        }
                }
        }

        if ($routing) {
                $routing=~s/;;/;/g;
                print AX "routing=$routing\n";
        }

        close AX;
}
print "Done!\n";
```

A criação da associação entre o endereço IP virtual e o endereço IP real é conseguido com o seguinte código.

```
# Build the ipaddress list
print "Building the IP address list: ";
open AX, ">$f/$paradigm/ipaddresses.lst";
print AX "# All basic IPs needed\n";
print AX "LOG:1        $myipaddris:1051\n";
foreach $k (keys %ip) {
        if ($spreadnodes and $node{$k}) {
```

```
                my $s="";
                foreach $j ( sort {$distnrnode{$b} <=> $distnrnode{$a}} \
                        keys %servers) {
                        next if ($j eq $myipaddris); # we don't allow nodes \
                                in main server to avoid collapse
                        $s=$j;
                }

                unless ($s) {
                        print  "ERROR:  Couldn't  find  an  available  server...  giving
up!\n";
                        exit;
                }

                $distnrnode{$s}++;
                $distport{$s}=$port unless ($distport{$s});
                print AX "$k   $s:$distport{$s} # node 1)\n";
                $distport{$s}++;
                next;
        } elsif (! $distip{$disttoprouter{$k}}) {
                foreach $j ( sort {$distnr{$b} <=> $distnr{$a}} keys %servers) {
                        next if ($distnr{$j}>=$nrnetpertoprouters);
                        next if ($distnr{$j}>=$nrnetatmyipaddris and \
                                $j eq $myipaddris);
                        $distip{$disttoprouter{$k}}=$j;
                }
                unless ($distip{$disttoprouter{$k}}) {
                        print "No server available... giving up!\n";
                        exit;
                }
                $distnr{$distip{$disttoprouter{$k}}}++;
                $distport{$distip{$disttoprouter{$k}}}=$port \
                        unless ($distport{$distip{$disttoprouter{$k}}});
        }
        print                                   AX                                      "$k
        $distip{$disttoprouter{$k}}:$distport{$distip{$disttoprouter{$k}}} # 1)\n";
        $distport{$distip{$disttoprouter{$k}}}++;
        if ($distport{$distip{$disttoprouter{$k}}} > $maxnrofport) {
                print "Maximum number of possible ports exceeded\niGiving up!\n";
                exit;
        }
}
print "Done!\n";

# Now, lets create an IP Address for each MR at each network
print "Creating an IP for each MR at each network\n";
$netname{"NoNet"}=1;
foreach $k (keys %mrmove) {
        @aux=split(/:/,$k);
        next unless ($netname{$aux[0]} and $mobilerouter{$aux[1]});
        if (!$ip{$k}) {
                print "$k      ".$distip{$disttoprouter{"$aux[1]:$aux[1]"}}.":".\
                        $distport{$distip{$disttoprouter{"$aux[1]:$aux[1]"}}}.\
                        " # 2)\n";
                print AX "$k   ".$distip{$disttoprouter{"$aux[1]:$aux[1]"}}.":".\
                        $distport{$distip{$disttoprouter{"$aux[1]:$aux[1]"}}}.\
                        " # 2)\n";
                $ip{$k}=1;
                $distport{$distip{$disttoprouter{"$aux[1]:$aux[1]"}}}++;
                 if ($distport{$distip{$disttoprouter{"$aux[1]:$aux[1]"}}} > \
                     $maxnrofport) {
                         print "Maximum number of possible ports \
                                exceeded\niGiving up!\n";
                         exit;
                 }
        }
}
```

C. *Script* mobsim_master.pl

O `mobsim_master.pl` é o responsável por construir os *scripts* dos dispositivos virtuais, informar o nó físico do *cluster*, instruir o início do processo, controlar a simulação e fazer a coleta dos dados.

O primeiro passo do *script* é a obtenção das informações dos paradigmas em estudo.

```
# We start getting all information about paradigms
for $f (@ARGV) {
        $aux=$f;
        $aux=~s/(\/|\\)$//;
        $aux=~s#.*(/|\\)([^/\\]+)$#$2#;
        print "-> $aux\n";

        # Getting list of IP Addresses
        $ipaddresses="";
        open AX, "$f/ipaddresses.lst";
        while ($aux1=<AX>) {
                chop($aux1);
                @aux1=split(/[\t ]+/,$aux1);
                @aux2=split(/\:/,$aux1[1]);
                @aux3=split(/\:/,$aux1[0]);
                $ipaddresses{$aux3[0]}.="\$ipaddress{\"$aux1[0]\"}=\"$aux2[0]\";\n";
                $ipaddresses{$aux3[0]}.="\$ipaddress_port{\"$aux1[0]\"}=\
                        \"$aux2[1]\";\n";
                $auxipaddr{$aux1[0]}=$aux2[0];
                $auxport{$aux1[0]}=$aux2[1];
        }
        close AX;
```

A informação dos tempos é outro passo importante.

```
        # Getting list of general delays
        $delays="";
        open AX, "$f/delays.lst";
        while ($aux1=<AX>) {
                chop($aux1);
                @aux1=split(/\t/,$aux1);
                $delays.="\$conf{\"$aux1[0]\"}=\"$aux1[1]\";\n";
        }
        close AX;
```

Seguidamente apresenta-se o código que vai buscar as configurações de cada dispositivo virtual.

```
        # Creating network programs
        open FIND, "find $f -name \"*.conf\" | sort |";
        while ($file=<FIND>) {
                chop($file);
                print " -> $file\n";
                delete @fconfs{keys %fconfs};
                $aux1=$file;
                $aux1=~s#.*(/|\\)([^/\\]+)\.conf$#$2#;
                open AX, "$file";
                while (<AX>) {
                        s/#.*//;
                        chop;
                        next unless ($_);
                        split(/=/);
                        $fconfs{$_[0]}=$_[1];
                }
                close AX;
```

Depois de várias verificações básicas, tal como verificar se está definido o programa a usar, ou se tem endereço IP e *gateway*, o programa continua para a inserção do código adicional. Após esta operação, o código é enviado para os nós operacionais do *cluster*.

```
@extracode="";
if ($fconfs{"extracode"}) {
        open AX, $f."/".$fconfs{"extracode"};
        @extracode=<AX>;
        close AX;
}

# Lets process this file...
$code=""; $extracode=0;
open AX, $conf{"bin_dir"}."/".$fconfs{"prog"};
while (<AX>) {
        if (/^#INSERT#CODE#HERE#/) {
                $code.=$_;
                $code.="\n# Confs inserted by MASTER\n";

                # General delays
                $code.=$delays;

                # All configs
                foreach $k (keys %fconfs) {
                        $code.="\$conf{\"$k\"}=\"$fconfs{$k}\";\n";
                }
                $code.="\$conf{\"myname\"}=\"$aux1\";\n";

                $code.="\n#  IP  addresses  automatically  inserted  by
MASTER\n";

                $code.="\$conf{\"NDHA\"}=\"NDHA:1\";\n";
                $code.="\n".$ipaddresses{"NDHA"};

                # Logging
                if ($fconfs{"logaddr"}) {
                        @aux=split(/\:/,$fconfs{'logaddr'});
                        $code.="\n".$ipaddresses{$aux[0]};
                }
                @aux=split(/\:/,$fconfs{'logaddr'});
                if ($ipaddresses{$aux[0]}) {
                        $code.="\n\$conf{\"logip\"}=\
                                \"$auxipaddr{$fconfs{'logaddr'}}\";\n";
                        $code.="\$conf{\"logport\"}=\
                                \"$auxport{$fconfs{'logaddr'}}\";\n";
                }

                # IP addresses
                @aux=split(/\:/,$fconfs{'ipaddr1'});
                $code.="\n\$conf{\"port1\"}=\
                        $auxport{$fconfs{'ipaddr1'}}\;\n";
                unless ($ipaddresses{$aux[0]}) {
                        print "No IP Addresses assigned for this \
                                host: $fconfs{'ipaddr1'}\n";
                        print "Error: exiting...\n";
                        exit;
                }
                $code.="\n".$ipaddresses{$aux[0]};
                if ($fconfs{'ipaddr2'}) {
                        @aux=split(/\:/,$fconfs{'ipaddr2'});
                        $code.="\n\$conf{\"port2\"}=\
                                $auxport{$fconfs{'ipaddr2'}}\;\n";
                        if ($ipaddresses{$aux[0]}) {
                                $code.="\n".$ipaddresses{$aux[0]};
                        }
                }

                # Routing tables
                if ($fconfs{'routing'}) {
                        $code.="\n# Routing tables:\n";
                        my @auxrr=split(/;/,$fconfs{'routing'});
                        foreach my $krr (@auxrr) {
                                my @auxrr1=split(/::/,$krr);
                                $code.="\$routing{\"$auxrr1[0]\"}=\
                                        \"$auxrr1[1]\";\n";
```

```
                        }
                }

                $extracode=1 if $fconfs{"extracode"};
                next;
        }

        if ($extracode) {
                if (/^#END#INSERT#CODE#HERE#/) {
                        if (@extracode) {
                                $code.="\n# Code automatically \
                                        inserted by MASTER\n";
                                $code.="@extracode\n";
                        }
                        $code.=$_;
                        $extracode=0;
                }
                next;
        }
        $code.=$_;
}
close AX;
```

Referências

[Abley03]	Abley, J., Black, B., and V. Gill, "Goals for IPv6 Site-Multihoming Architectures", RFC 3582, agosto 2003
[Adjie-Winoto00]	William Adjie-Winoto, Elliot Schwartz, Hari Balakrishnan, Jeremy Lilley, "The design and implementation of an intentional naming system", SIGOPS Oper. Syst. Rev. 34, 2000
[Ahmed10]	Waqas Ahmed, M. Jashim, "An SCTP Based Decentralized Mobility Framework", Masters Thesis, Department of Electrical Engineering, Blekinge Institue of Technlogy, Karlskrona, Sweden, julho 2010
[Arkko04]	Arkko, J., Devarapalli, V., and F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents", RFC 3776, junho 2004.
[Arkko09]	Arkko, J. and I. van Beijnum, "Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming", RFC 5534, junho 2009
[Aura02]	Aura, T. and J. Arkko, "MIPv6 BU Attacks and Defenses", Work in Progress, março 2002
[Aura02]	Aura, T. and J. Arkko, "MIPv6 BU Attacks and Defenses", Work in Progress, março 2002
[Ayaz09]	Serkan Ayaz, Christian Bauer, and Fabrice Arnal, "Minimizing end-to-end delay in global haha networks considering aeronautical scenarios", in Proceedings of the 7th ACM international symposium on Mobility management and wireless access (MobiWAC '09), New York, NY, USA, 42-49, 2009
[Bagnulo05]	Bagnulo, M., "Address selection in multihomed environments", Work in Progress, outubro 2005
[Bagnulo07]	Bagnulo, M. and J. Abley, "Applicability Statement for the Level 3 Multihoming Shim Protocol (Shim6)", Work in Progress, julho 2007
[Bagnulo08]	Bagnulo, M., "Default Locator-pair selection algorithm for the Shim6 protocol", Work in Progress, outubro 2008
[Bagnulo09]	Bagnulo, M., "Hash-Based Addresses (HBA)", RFC 5535, junho 2009

[Baldessari09]	R. Baldessari, T. Ernst, A. Festag, M. Lenardi, "Automotive Industry Requirements for NEMO Route Optimization", draft-ietf-mext-nemo-ro-automotive-req-02 (Work in progress), janeiro 2009
[Barré11]	S. Barré, J. Ronan, O. Bonaventure, "Implementation and evaluation of the Shim6 protocol in the Linux kernel", Computer Communications 34, pp. 1685-1695, março 2011
[Bernardos04]	Carlos J. Bernardos, Marcelo Bagnulo, María Calderón, "MIRON: MIPv6 Route Optimization for NEMO", 4th Workshop on Applications and Services in Wireless Networks (ASWN 2004), pp 189 a 197, agosto 2004
[Bernardos05a]	Carlos J. Bernardos et. al, Upgrade Vol IV, issue no. 2, "NEMO: Network Mobility in IPv6", abril 2005
[Bernardos07]	Carlos Bernardos et. al, "Mobile IPv6 Route Optimisation for Network Mobility (MIRON)", draft-bernardos-nemo-miron-01 (work in progress), julho 2007
[Bernardos11]	CJ. Bernardos, A. de la Oliva, F. Giust, "A IPv6 Distributed Client Mobility Management approach using existing mechanisms", in draft-bernardos-mext-dmm-cmip (Work in progress), março 2011
[Budzisz08]	L. Budzisz, R. Ferrus, A. Brunstrom, and F. Casadevall, "Towards transport-layer mobility: Evolution of SCTP multihoming," Computer Communication, vol. 31, pp. 980-998, dezembro 2008
[Chan11]	H. Chan (Ed.), "Problem statement for distributed and dynamic mobility management", in draft-chan-distributed-mobility-ps (Work in progress), outubro 2011
[Cheng05]	Wang-Cho Cheng, Chee-Wei Ang, Kim-Sing Wong, "A DNS-Based Host Portability Solution With Localized Location Updating", In Proceedings of the The 2005 Symposium on Applications and the Internet (SAINT '05), IEEE Computer Society, Washington, DC, USA, 2005
[Conta06]	A. Conta, S. Deering, M. Gupta, Ed.,"Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", IETF RFC 4443, março 2006
[Conta98]	Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, dezembro 1998
[Conti01]	Marco Conti, Enrico Gregori, Silvia Martelli, "DNS-based Architectures for an efficient Management of Mobile Users in Internet", In Proceedings of the 15th International Parallel & Distributed Processing Symposium (IPDPS '01), IEEE Computer Society, Washington, DC, USA, 2011
[Deering98]	Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, dezembro 1998
[Devarapalli05]	V. Devarapalli et. al, RFC 3963, "Network Mobility (NEMO) Basic Support Protocol", janeiro 2005.

[Devarapalli07]	Devarapalli, V. and F. Dupont, "Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture", RFC 4877, abril 2007
[Droms03]	R. Droms, et. al, RFC 3315, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", julho 2003
[Eastlake05]	Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, junho 2005
[Eastlake11]	Eastlake, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, maio 2011
[Eddy04]	W.M. Eddy, "At what layer does mobility belong?", IEEE Communications Magazine, pp. 155-159, outubro 2004
[Eddy09]	Eddy, W., Ivancic, W., and T. Davis, "Network Mobility Route Optimization Requirements for Operational Use in Aeronautics and Space Exploration Mobile Networks", RFC 5522, outubro 2009
[Ernst07]	Thierry Ernst et. al, RFC 4885, "Network Mobility Support Terminology", julho 2007
[Ernst07a]	T. Ernst, RFC 4886, "Network Mobility Support Goals and Requirements", julho 2007
[Farinacci11]	Farinacci, D. and V. Fuller, "LISP Map Server", draft-ietf-lisp-ms-12.txt (work in progress), 2011
[Farinacci11a]	Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format", draft-farinacci-lisp-lcaf-06.txt (work in progress), 2011
[Farinacci11b]	Farinacci, D., Fuller, V., Lewis, D., and D. Meyer, "LISP Mobility Architecture", draft-meyer-lisp-mn-06.txt (work in progress).
[Farinacci11c]	Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "LISP Alternative Topology (LISP-ALT)", draft-ietf-lisp-alt-09.txt (work in progress), setembro 2011
[Farinacci12]	D. Farinacci, V. Fuller, D. Meyer, D. Lewis, "Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-22 (work in progress), fevereiro 2012
[Ferguson00]	Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, maio 2000
[Gai07]	Silvano Gai, "Ipv6 The new protocol for internet and intranets", [Online] http://www.ip6.com/us/book/, dezembro 2007
[Gieben04]	Gieben, M., "DNSSEC: The Protocol, Deployment, and a Bit of Development," The Internet Protocol Journal, Volume 7, No. 2, junho 2004

[Gulbrandsen00]	A. Gulbrandsen, P. Vixie, and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)," Request for Comments 2782, Internet Engineering Task Force, fevereiro 2000
[Gundavelli08]	S. Gundavelli (Ed.), K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, "Proxy Mobile IPv6", RFC 5213, agosto 2008
[Gurtov09]	Andrei Gurtov, Miika Komu, Robert Moskowitz, "Host Identity Protocol: Identifier/Locater Split for host mobility and multihoming", IPJ, março 2009
[Handley99]	M. Handley, H. Schulzrinne, E. Schooler, and J. Rosen- berg, "SIP: session initiation protocol," Request for Comments 2543, Internet Engineering Task Force, março 1999
[Henderson03]	Henderson,T.,R.,Ahrenholz,J.,M.,Kim,J.,H., "Experience with the Host Identity Protocol for Secure Host Mobility and Multihoming", IEEE Wireless Communications and Networking, pp. 2120-2125, vol.3, março 2003
[Henderson03a]	Thomas R. Henderson, "Host Mobility for IP Networks: A Comparison", IEEE Network, Vol. 17, No. 6, pp. 18-26, novembro 2003
[Henderson08]	T. Henderson, P. Nikander, M. Komu, "Using the Host Identity Protocol with Legacy Applications", RFC 5338, setembro 2008
[Henderson08a]	Tom Henderson, "Manual for Network Simulator, version 2", [Online] http://www.isi.edu/nsnam/ns/doc/node575.html, acedido em 2008
[Iannone11]	Iannone, L., Saucez, D., and O. Bonaventure, "LISP Mapping Versioning", draft-ietf-lisp-map-versioning-05.txt (work in progress), 2011
[Imtiaz11]	Waqas A. Imtiaz, M. Afaq, Mohammad A.U. Babar, "mSCTP Based Decentralized Mobility Framework", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No.9, 2011
[Inria08]	Inria, "MobiWan: NS-2 extensions to study mobility in Wide-Area IPv6 Networks", [Online] http://www.inrialpes.fr/planete/mobiwan/, acedido em 2008
[Jakab11]	Jakab, L., Coras, F., Domingo-Pascual, J., and D. Lewis, "LISP Network Element Deployment Considerations", draft-ietf-lisp-deployment-02.txt (work in progress), novembro 2011
[Jayaraman08]	P. Jayaraman, R. Lopez, Y. Ohba, M. Parthasarathy, A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA) Framework", RFC 5193, maio 2008
[Jeong04]	Jaehoon Jeong, et. al, "Route optimization based on ND-proxy for mobile nodes in IPv6 mobile networks", in Vehicular Technology Conference 2004 (VTC 2004-Spring), IEEE 59th, pp. 2461-2465 Vol.5, 2004
[Jeong04a]	Jaehoon Paul Jeong, et. al, "ND-Proxy based Route and DNS Optimizations for Mobile Nodes in Mobile Network", in draft-jeong-nemo-ro-ndproxy-02.txt (Work in progress), fevereiro 2004

[Kame08]	The Kame project site, "SHISA; The new KAME Mobile IPv6 / NEMO stack", [Online] http://www.kame.net/newsletter/20041211/, acedido em 2008
[KCL08]	King's College London, "Academic Research and Teaching with OPNET software", URL: http://www.ctr.kcl.ac.uk/opnet/opnet.html, acedido em: 2008
[Kempf00]	J. Kempf and J. Rosenberg, "Finding a SIP server with SLP," Internet Draft, Internet Engineering Task Force, fevereiro 2000
[Kent05a]	Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, dezembro 2005
[Kent05b]	Kent, S., "IP Authentication Header", IETF RFC 4302, dezembro 2005
[Kent05c]	Kent, S., "IP Encapsulating Security Payload (ESP)", IETF RFC 4303, dezembro 2005
[Kent98]	Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, novembro 1998
[Koh04]	Seok Joo Koh, Moon Jeong Chang, Meejeong Lee, "mSCTP for soft handover in transport layer", Communications Letters, IEEE, março 2004
[Koh08]	Kim and S. Koh, "Analysis of handover latency for mobile IPv6 and mSCTP," IEEE International Conference on Communication Workshops, pp. 420-429, maio 2008
[Komu08]	Komu, M., Bagnulo, M., Slavov, K., and S. Sugimoto, "Socket Application Program Interface (API) for Multihoming Shim", Work in Progress, novembro 2008
[Laganier08]	Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", RFC 5204, abril 2008
[LCA08]	LCA site, "Bem vindos à página da Milipeia" [Online], [Online] http://www.lca.uc.pt/, acedido em 2008
[Lear96]	Lear, E., Katinsky, J., Coffin, J., and D. Tharp, "Renumbering: Threat or Menace?", Tenth USENIX System Administration Conference (LISA X), Chicago, IL, USA, setembro 1996
[Lee06]	C.H. Lee, J.R. Zheng, C.M. Huang, "SIP-based Network Mobility (SIP-NEMO) Route Optimization (RO)", draft-ming-nemo-sipnemo-01.txt, outubro 2006
[Madison98]	Madson, C. and R. Glenn, "The Use of HMAC-SHA-1-96 within ESP and AH", RFC 2404, novembro 1998.
[Maino11]	Maino, F., Ermagon, V., Cabellos, A., Sausez, D., and O. Bonaventure, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-00.txt (work in progress), julho 2011

[Manner04]	Manner, J. and M. Kojo, "Mobility Related Terminology", RFC 3753, junho 2004
[Marples00]	D. Marples, "Naming and accessing internet appliances using extensions to the session initiation protocol," in Proc. of SIP 2000 Conference and Exhibition, (Paris, France), maio 2000
[Mauchie10]	Fabian Mauchle, Sandra Frei, Andreas Rinke, "Simulating mobile IPv6 with ns-3", in SIMUTools '10 Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques, Bruxelas, Bélgica, 2010
[Meyer08]	David Meyer, "The Locator Identifier Separation Protocol (LISP)", Internet Protocol Journal - Volume 11, Number 1, março 2008
[Meyer09]	Meyer, D. and D. Lewis, "Architectural Implications of Locator/ID Separation", draft-meyer-loc-id-implications-02.txt (work in progress), janeiro 2009
[Mockapetris87a]	P. Mockapetris, "Domain names - concepts and facilities", RFC 1034, STD0013, novembro 1987
[Mockapetris87b]	P. Mockapetris, "Domain names - implementation and specification", RFC 1035, STD0013, novembro 1987
[Moskowitz06]	Moskowitz, R. and P. Nikander, "Host Identity Protocol (HIP) Architecture", RFC 4423, maio 2006
[Moskowitz08]	Moskowitz, R., Nikander, P., Jokela, P., Ed., and T. Henderson, "Host Identity Protocol", RFC 5201, abril 2008
[Na04]	Jongkeun Na et. al, "Route Optimization Scheme based on Path Control Header", draft-na-nemo-path-control-header-00 (work in progress), abril 2004.
[Na04a]	Jonkeun Na, "Supporting Route Optimization in Network Mobility (NEMO)", Technical Report INC2004-01, Seoul National University, dezembro 2004
[Na04b]	Jongkeun Na, Jaehyuk Choi, Seongho Cho, Chongkwon Kim, et al., "A Unified Route Optimization Scheme for Network Mobility," In Proceedings of 9th Intl. Conference on Personal Wireless Communications (LNCS 3260), Delft, The Netherlands, setembro 2004
[Narten07]	T. Narten et. al, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, Internet Engineering Task Force, setembro 2007
[Nautilus08]	Nautilus6, [Online] http://www.nautilus6.org/, acedido em 2008
[Ng07]	C. Ng, et. al, "Network Mobility Route Optimization Problem Statement", RFC 4888, julho 2007

[Ng07a]	C. Ng et. al, "Network Mobility Route Optimization Solution Space Analysis", RFC 4889, julho 2007
[Ng08]	C. Ng, J. Hirano, A. Petrescu, E. Paik, "Consumer Electronics Requirements for Network Mobility Route Optimization", draft-ng-nemo-ce-req-02 (work in progress), fevereiro 2008
[Nikander05]	Nikander, P., Arkko, J., Aura, T., Montenegro, G., and E. Nordmark, "Mobile IP Version 6 Route Optimization Security Design Background", RFC 4225, dezembro 2005
[Nikander08]	Nikander, P. and J. Laganier, "Host Identity Protocol (HIP) Domain Name System (DNS) Extensions", RFC 5205, abril 2008
[Nikander08a]	Nikander, P., Henderson, T., Vogt, C. and Arkko, J., "End-host Mobility and Multihoming with the Host Identity Protocol", RFC 5206, abril 2008
[Nordmark01]	Nordmark, E., "Securing MIPv6 BUs using return routability (BU3WAY)", Work in Progress, novembro 2001
[Nordmark05]	Nordmark, E., "Shim6-Application Referral Issues", Work in Progress, julho 2005
[Nordmark09]	Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, junho 2009
[ns-308]	ns-3, [Online] http://www.nsnam.org/, acedido em 2008
[ns-308a]	ns-3, "Wiki for ns-3", [Online] http://www.nsnam.org/wiki/index.php/Main_Page, acedido em 2008
[Ohnishi03]	H. Ohnishi, K. Sakitani, Y. Takagi, "HMIP based Route optimization method in a mobile network", in draft-ohnishi-nemo-ro-hmip-00 (Work in progress), outubro 2007
[OMNet09]	O. C. Site, "OMNet++ Community Site.", [Online] http://omnetpp.org/, acedido em 2009
[OPNET08]	OPNET site, "OPNET Application and Network Performance" [Online], [Online] http://www.opnet.com/, acedido em 2008
[OPNET08a]	OPNET site, "Scalable simulation", [Online] http://www.opnet.com/solutions/network_rd/scalable_simulation.html , acedido em: 2008
[OPNETSC06]	OPNET support center, "Models for the Vehicular Ad-Hoc Route Optimisation for NEMO (VARON) protocol", [Online] https://enterprise1.opnet.com/tsts/4dcgi/Models_SearchSubmit?QueryModels_what=FindAll&QueryRecordsPerPage=500, 2006

[Pandya95]	Pandya, R., "Emerging mobile and personal communication systems", IEEE Communications Magazine , Vol. 33, pp. 44--52, junho 1995
[Park07]	Jeonghoon Park, Tae-Jin Lee, and Hyunseung Choo, "Route Optimization with MAP-Based Enhancement in Mobile Networks", in Proceedings of the 7th international conference on Computational Science, ICCS 2007, Berlin, Heidelberg, 2007
[Patil11]	B. Patil, Ed., C. Williams, J. Korhonen, "Approaches to Distributed mobility management using Mobile IPv6 and its extensions", in draft-patil-mext-dmm-approaches (Work in progress), outubro 2011
[Perez-Costa03]	X. Perez-Costa and M. Torrent-Moreno, "A Performance Study of Hierarchical Mobile IPv6 from a System Perspective", 38th IEEE International Conference on Communications (ICC), Pages: 961 - 970, Anchorage, Alaska, maio 2003
[Perkins10]	C. Perkins, "IP Mobility Support for IPv4, Revised", RFC 5944, IETF, novembro 2010
[Perkins11]	C. Perkins et. al, RFC 6275, "Mobility Support in IPv6", julho 2011.
[Perkins96]	Perkins, C., "IP Encapsulation within IP", IETF RFC 2003, outubro 1996
[Perl08]	Perl site, "The Perl Programming Language", [Online] http://www.perl.org/, acedido em: 2008
[Prasad05]	Anand Prasad, Neeli Prasad, "802.11 WLANs and IP networking: security, QoS, and mobility", Artech House mobile communications library, ISBN 1-58053-789-8, 2005
[Ratola04]	Mika Ratola, "Which Layer for Mobility? - Comparing Mobile IPv6, HIP and SCTP", In HUT T-110.551 Seminar on Internetworking, 2004.
[Roe02]	Roe, M., "Authentication of Mobile IPv6 Binding Updates and Acknowledgments", Work in Progress, março 2002
[Roe02]	Roe, M., "Authentication of Mobile IPv6 Binding Updates and Acknowledgments", Work in Progress, março 2002.
[Rosenberg02]	Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, junho 2002
[Rosenberg02a]	J. Rosenberg et al., "Session Initiation Protocol (SIP) Extensions for Presence", Internet Draft draft-ietf-simple-presence-07.txt, IETF, junho 2002
[Rosenberg99]	J. Rosenberg and H. Schulzrinne, "The IETF internet telephony architecture and protocols," IEEE Network, Vol. 13, pp. 18–23, maio/junho 1999

[Saltzer93]	Saltzer J., "On The Naming and Binding of Network Destinations," IETF RFC 1498, agosto 1993
[Savola02]	Savola, P., "Security of IPv6 Routing Header and Home Address Options", Work in Progress, março 2002
[Schulzrinne00]	H. Schulzrinne and J. Rosenberg, "The session initiation protocol: Internet-centric signaling," IEEE Communica- tions Magazine, Vol. 38, outubro 2000
[Schulzrinne00a]	Schulzrinne, H., and E. Wedlund, "Application-layer mobility using SIP," ACM Mobile Computing and Commun. Rev., Vol. 4, No. 3, pp. 47–57, julho 2000
[Schulzrinne99]	H. Schulzrinne and J. Rosenberg, "Internet telephony: Architecture and protocols – an IETF perspective," Com- puter Networks and ISDN Systems, Vol. 31, pp. 237–255, fevereiro 1999
[Sekercioglu03]	Y. Ahmet Sekercioglu, Andras Varga and Gregory K. Egan, "Parallel Simulation Made Easy with OMNeT++". In Proceedings of the European Simulation Symposium (ESS 2003), Delft, The Netherlands, outubro 2003
[Shahriar07]	A. Z. M. Shahriar and M. Atiquzzaman, "Network Mobility in Satellite Networks", NASA Earth Science Technology Conference, maio 2007
[Snoeren01]	Alex C. Snoeren, Hari Balakrishnan, M. Frans Kaashoek, "Reconsidering Internet Mobility", Proc. of the 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII), maio 2001
[Soliman08]	H. Soliman, et. al, "Hierarchical Mobile IPv6 (HMIPv6) Mobility Management", IETF RFC 5380, outubro 2008
[Soto10]	I. Soto, C. J. Bernardos, M. Calderon, and T. Melia, "PMIPv6: A Network-based Localized Mobility Management solution", The Internet Protocol Journal, 13(3), setembro 2010
[Srisuresh01]	P. Srisuresh, K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, janeiro 2001
[Stallings05]	William Stallings, CISCO – The Internet Protocol Journal, "Mobile IP", 2005
[Stewart00]	Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, outubro 2000
[Stewart04]	R. Stewart et al., "Stream Control Transmission Protocol," in IETF RFC 2960, outubro 2000
[Stewart06]	Stewart, R., Arias-Rodriguez, I., Poon, K., Caro, A., and M. Tuexen, "Stream Control Transmission Protocol (SCTP) Specification Errata and Issues", RFC 4460, abril 2006

[Stewart07]	R. Stewart, Q. Xie, M. Tuexen, M. Maruyama, and M. Kozuka, "Internet-Draft SCTP Dynamic Address Reconfiguration". [Online]. http://tools.ietf.org/html/drafts/ietf-tsvwg-addip-sctp-22, junho 2007
[Stone02]	Stone, J., Stewart, R., and D. Otis, "Stream Control Transmission Protocol (SCTP) Checksum Change", RFC 3309, setembro 2002
[Tahi08]	TAHI Project, "Test and Verification for IPv6. Since 1998", [Online] http://www.tahi.org/, acedido em 2008
[Thomson07]	S. Thomson, T. Narten, T. Jinmei, RFC 4862, "IPv6 Stateless Address Autoconfiguration", setembro 2007
[Thubert06]	P. Thubert, et. al, "Global HA to HA protocol", in draft-thubert-nemo-global-haha-02 (Work in progress), setembro 2006
[Thubert09]	Thubert, P., R. Wakikawa, V. Devarapalli, "Global HA to HA Protocol", in draft-thubert-mext-global-haha-01 (Work in progress), julho 2009
[Tuexen05]	Tuexen, M., et al., "Authenticated Chunks for Stream Control Transmission Protocol (SCTP)", draft-ietf-tsvwg-sctp-auth-00.txt, junho 2005
[Tuexen07]	Tuexen, M., Stewart, R., Lei, P., and E. Rescorla, "Authenticated Chunks for Stream Control Transmission Protocol (SCTP)", RFC 4895, agosto 2007
[Tuexen11]	M. Tuexen,R. Stewart, "Stream Control Transmission Protocol (SCTP) Chunk Flags Registration", RFC 6096, janeiro 2011
[USAGI08]	USAGI Project, "Linux IPv6 Development Project", [Online] http://www.linux-ipv6.org/, acedido em 2008
[Vapi08]	Pedro Vale Pinheiro, Fernando Boavida, "OMEN – A New Paradigm for Optimal Network Mobility", 6th International Conference on Wired/Wireless Internet Communications (WWIC2008), Tampere, Finland, maio 2008
[Vapi09]	Pedro Vale Pinheiro, Fernando Boavida, "OMEN – Novo Paradigma de Mobilidade de Redes", 9ª Conferência sobre Redes de Computadores (CRC'2009), Oeiras, Portugal, outubro 2009
[Vapi10]	Pedro Vale Pinheiro, Shivam Jain, Fernando Boavida, "Comparação de soluções de mobilidade de redes num cenário de grande dimensão", 10ª Conconferência sobre Redes de Computadores (CRC'2010), Universidade do Minho, Braga, novembro 2010
[Vapi10a]	Pedro Vale Pinheiro, Fernando Boavida, "mobSim – Uma ferramenta para simulação de mobilidade de redes", 10ª Conferência sobre Redes de Computadores (CRC'2010), Universidade do Minho, Braga, novembro 2010
[Vapi11]	Pedro Vale Pinheiro, Fernando Boavida, "mobSim – A Network Mobility Simulation Tool for Very Large-Scale Scenarios", 4th IFIP International Conference on New Technologies, Mobility and Security (NTMS'2011), Paris, França, fevereiro 2011

[Vapi11a]	Pedro Vale Pinheiro, Shivam Jain, Fernando Boavida, "A Comparative Study of Network Mobility Paradigms", 9th International Conference on Wired/Wireless Internet Communications, Vilanova i la Geltrú, Barcelona, Espanha, junho 2011
[Vapi11b]	Pedro Vale Pinheiro, Fernando Boavida, "Testes e análise de stress de soluções para mobilidade de redes", in 11ª Conferência sobre Redes de Computadores (CRC'11), Coimbra, Portugal, novembro 2011
[Vapi12]	Pedro Vale Pinheiro, Fernando Boavida, "Some Results on Network Mobility Stress Testing", in 2nd Baltic Conference on Future Internet Communications (BCFIC 2012), Vilnius, Lithuania, abril 2012
[Vixie97]	P. Vixie, S. Thomson, Y. Rekhter, J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, abril 1997
[Wakikawa03]	Ryuji Wakikawa et. al, "ORC: Optimized Route Cache Management Protocol for Network Mobility", 10th International Conference on Telecommunications, vol 2, pp 1194-1200, fevereiro 2003.
[Wakikawa04]	Ryuji Wakikawa et. al, "Optimized Route Cache Protocol (ORC)", draft-wakikawa-nemo-orc-01 (work in progress), novembro 2004.
[Wakikawa06]	Ryuji Wakikawa, Pascal Thubert, Vijay Devarapalli, "Inter Home Agents Protocol Specification", in draft-wakikawa-mip6-nemo-haha-spec-01 (Work in progress), março 2006
[Wedlund99]	Elin Wedlund and Henning Schulzrinne, Mobility support using SIP. In Proceedings of the 2nd ACM international workshop on Wireless mobile multimedia (WOWMOM '99). ACM, New York, NY, USA, 1999
[Yousaf08]	Faqir Zarrar Yousaf and Christian Bauer and Christian Wietfeld, "An accurate and extensible mobile IPv6 (xMIPV6) simulation model for OMNeT++," in Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops. Marseille, France: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 1–8, 2008