

## Demonstrador de Redes Heterogêneas suportando Mobilidade

Diogo Gomes, Nuno Duarte, Nuno João Sénica, Rui L. Aguiar

**Resumo** – Este artigo apresenta um demonstrador de redes heterogênea desenvolvido tendo por base o projecto Europeu Moby Dick. O demonstrador suporta qualidade de serviço, mobilidade e autenticação de uma forma integrada. O artigo discute diversos aspectos da implementação do demonstrador, bem como alguns pacotes de software desenvolvidos. O artigo apresenta ainda testes realizados sobre o demonstrador.

**Abstract** – A demonstrator of heterogeneous networking, based on the European Project Moby Dick, is here presented. The demonstrator supports quality of service, mobility and integrated authentication. The paper discusses different aspects of the implementation of the demonstrator, as well as some packages of software developed. The paper further presents some tests performed on the demonstrator.

### I. INTRODUÇÃO

O desenvolvimento simultâneo de sistemas de comunicações de 3ª geração e de novos meios de acesso à Internet tem levado ao desenvolvimento de sistemas e conceitos que promovam a junção destas duas ideias chaves no mundo das telecomunicações actuais. Neste sentido, os aspectos de qualidade de serviço (*Quality of Service – QoS*) na Internet aparecem como uma das áreas de trabalho mais interessantes, dada a complexidade de que se reveste o planeamento e controlo de redes IP com QoS, quando associadas a ambientes móveis – a própria mobilidade dos utilizadores pode criar facilmente pontos de congestionamento na rede. Por sua vez, a múltipla complexidade de serviços possíveis de disponibilizar nestes ambientes torna o controlo da rede uma tarefa complexa.

Os comportamentos de uma sociedade orientada à informação criam um aumento de expectativas sem precedentes respeitante à sua infra-estrutura. A possibilidade de criar serviços que respondam a (virtualmente) qualquer requisito torna-se num assunto importante que irá determinar o sucesso das soluções futuras. Este problema complexo constitui uma fonte de interesse não só para comunidade científica como também para os operadores de telecomunicações. É neste cenário que se posiciona o projecto Moby Dick [1], um projecto Europeu que integra vários parceiros provenientes de diversos meios científicos e comerciais, com especial ênfase em operadores de telecomunicações. Este projecto pretende definir uma arquitectura capaz de suportar Qualidade de Serviço (QoS) num ambiente heterogêneo

onde coexistem Ethernet, Wifi (802.11b) e TD-CDMA (utilizado na camada de rádio do UMTS). Todo este ambiente suporta Segurança por meio do uso de IPSec e um sistema de Authorization, Authentication, Accounting and Charging (AAAC) baseado em DIAMETER. Segundo o conceito Moby Dick existe uma rede de comunicações onde coexistem diferentes tecnologias de acesso (Ethernet, WiFi, TD-CDMA) e onde o utilizador pode cruzar diversos domínios administrativos. Dentro da rede Moby Dick são possíveis múltiplos modelos de negócio (e.g. pré-pagos ou assinatura) estando a distribuição de recursos ou consumo de serviços de alguma forma documentados para auditoria e cobrança apropriadas.

### II. O PROJECTO MOBY DICK

O maior desafio da arquitectura Moby Dick é encontrar uma combinação adequada entre os elementos, os mecanismos e as funções de QoS, AAA e Mobilidade, que permitam um comportamento otimizado [1].

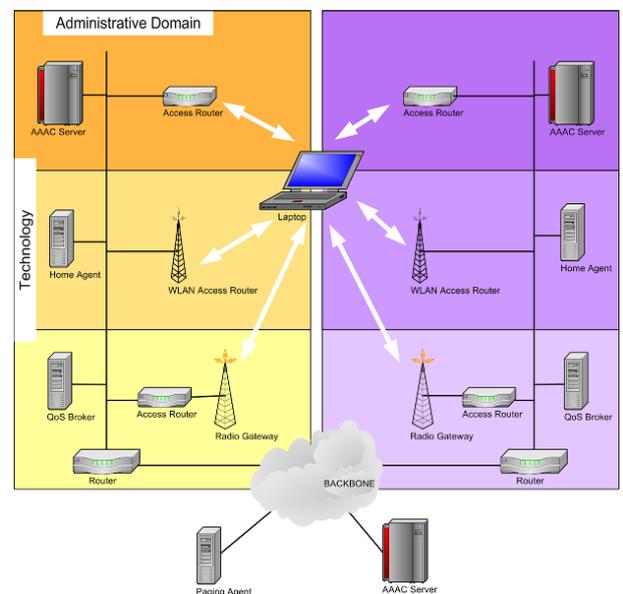


Figura 1 – O conceito de rede Moby Dick

No ambiente de referência Moby Dick, um utilizador tem um contrato com um fornecedor na sua rede de origem (*home domain*). Este fornecedor mantém uma entidade de AAA (AAA.h), responsável por validar e verificar as credenciais do utilizador, e contabilizar a utilização dos serviços. Os detalhes do contrato entre o utilizador e o fornecedor de acesso são acordados

baseados num SLA (*Service Level Agreement*) que é transformado num perfil de utilizador guardado na entidade AAA.h. O conteúdo deste perfil tem entre outras, informações de tarifa, permissões de *roaming* e parâmetros de QoS. Por exemplo, o fornecedor de acesso de uma rede estrangeira (estrangeira significa que o utilizador não possui qualquer contrato com a rede que pretende usar) disponibiliza ao utilizador recursos de rede de acordo com o conteúdo deste SLA, reflectindo o seu perfil da rede de origem (*home domain*). Pode dar-se o caso de não existir qualquer acordo de *roaming* entre o fornecedor da rede de origem e o fornecedor da rede estrangeira - e nesse caso não é permitido o uso de serviços comerciais por parte desse utilizador móvel.

### III. CARACTERÍSTICAS DO DEMONSTRADOR

Apesar de ter sido usado o sistema operativo Linux (Red Hat 7.2) como base no desenvolvimento do software dentro do projecto Moby Dick, optou-se por instalar uma outra distribuição (Suse 8.0), não só devido à familiarização com essa distribuição, mas também a incompatibilidades com a instalação do Red Hat 7.2 nas máquinas disponíveis.

Uma vez instalado o sistema operativo com a kernel 2.4.16 (kernel de base usada para o desenvolvimento de todo o software), e após configurada a rede em IPv6, o passo seguinte foi instalar um servidor DNS com suporte IPv6 (utilizou-se o BIND 9.2.1).

O demonstrador foi interligado ao 6bone, uma rede internacional baseada em IPv6. Este acesso foi implementado por intermédio de uma rede IPv6 já existente no IT. A conectividade IPv6 é feita através de um router CISCO 3600, que corre *ripngd* (tal como todos AR's da rede). Embora o router anunciasse à rede a *default gateway* IPv6, e em todos PCs estivesse presente essa *default gw*, os pacotes não eram encaminhados e apenas o CISCO e os PCs na mesma subnet é que tinham conectividade. Após alguma pesquisa, verificou-se que as kernels inferiores à versão 2.4.17 tinham problemas no *forward* de pacotes que utilizassem a *default gw*, e propunham como alternativa a definição de uma entrada de routing para a rede 2000::/3, único prefixo de rede IPv6 global.

Como é óbvio, a instalação da rede passou assim por diversas fases, tendo sido essencial utilizar uma estratégia gradativa. Essa estratégia passou também pelos diversos módulos de software, que foram sendo instalados individualmente, recorrendo-se a *dummies* para substituir as "outras" componentes. Foram assim possíveis (e essenciais para debug) pequenas demonstrações das diferentes componentes da rede, demonstrando isoladamente Fast Handover, QoS e também AAAC.

Com a correcção de alguns dos problemas existentes no software, foi possível passar a uma fase de integração com a quase totalidade dos componentes que deveriam existir na rede (com a excepção de IPsec). Como será

detalhado na próxima secção, foi possível a interligação entre: Registration, Fast Handover, Qos Manager, AAAC cliente, AAAC server, QoS Broker, Meter, Logger, Accounting. Ao atingir-se esta fase foi necessário fazer um planeamento cuidado das entidades funcionais suportadas por cada máquina (entrando em consideração com aspectos como capacidade de processamento, p.ex.), devido à necessidade de otimizar o equipamento do laboratório.

Ultrapassadas tais dificuldades obteve-se a estrutura representada na Fig. 2.

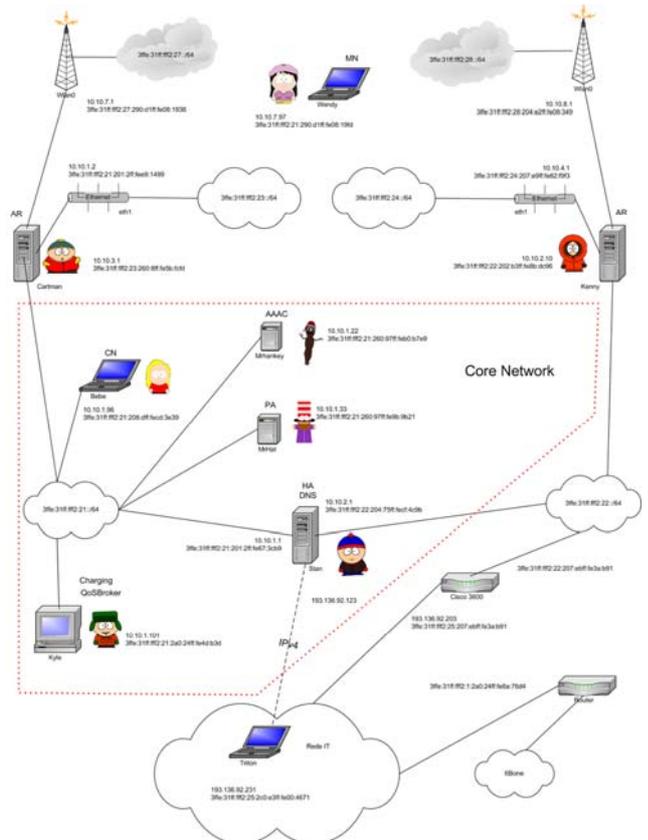


Figura 2 - Rede DMH – Diagrama estrutural e funcional da rede desenvolvida no âmbito do projecto.

Dado o seu carácter inovador, esta rede foi demonstrada publicamente na "IST Mobile & Wireless Communications Summit 2003".

### IV. SOFTWARE UTILIZADO

As entidades principais do demonstrador são:

- Mobile Terminal
- Access Router
- Home Agent
- Paging Agent
- AAAC Server
- QoS Broker [2]

O software usado para a rede de demonstração foi desenvolvido na sua maioria no âmbito do projecto Moby Dick, por diferentes parceiros do projecto.

Os vários softwares desenvolvidos podem por si só implementar entidades da arquitectura Moby Dick. No entanto existem outras entidades da arquitectura, que necessitam da integração de vários pacotes de software para poderem operar.

#### A. Mobile terminal

O terminal móvel (laptop) é constituído por diferentes componentes de software:

- NCP (Networking Control Panel) - uma interface gráfica para o utilizador poder definir e iniciar a execução de handover manuais.
- MTNM (Mobile Terminal Network Manager) - módulo que decide sobre a execução de handover
- Módulo de fast handover - tem o objectivo de despoletar os mecanismos de Fast Handover, após a recepção de uma mensagem de "START\_FHO" iniciada por parte do MTNM
- Módulo de registo – permite o registo perante o AAAC.
- Módulo de paging – responsável por manter o estado do Mobile Terminal actualizado (activo ou adormecido).
- Pilha IPv6 melhorada, baseada na pilha IPv6 da kernel 2.4.16 do Linux, mas adicionada de:
  1. MIPL 0.9.1, que proporciona funcionalidades a nível de mobilidade.
  2. SWAMP (Secure Wide-Area Mobility Package), que combina a implementação de IPSec com Mobile IPv6 para a kernel do Linux.
  3. DSCP marking software - atribui um código DSCP a todos os pacotes que saem da máquina, baseado num conjunto de regras que podem ser configuradas [3].
- Drivers para dispositivos de rede - também com a funcionalidade acrescida de filtragem (WLAN Router Advertisements)

#### B. Access Router

O Access router é constituído por diferentes componentes de software, este inclui:

- Um módulo de Fast Handover, que é responsável pela adequada transferência de informação na ocorrência de FHO.
- Um Paging Attendant que é responsável pelo paging na célula de acesso activa.
- Um ambiente AAAC, suportado por um módulo de metering e pelo AAAC Attendant
- Um QoS Attendant, responsável pela interface de suporte de QoS no Access Router.
- A stack IPv6 melhorada, baseada na stack IPv6 da kernel 2.4.16, mas adicionada de:
  1. SWAMP, que integra IPSec e Mobile-IPv6, necessário para garantir a segurança

na rede de acesso e a mobilidade de terminais..

2. Diffserv e filtragem de pacotes: para actuar sobre um pacote consoante o DSCP que este possui, isto é, poder atribuir uma determinada largura de banda a determinado fluxo caracterizado pelo seu DSCP.

- Drivers dos dispositivos de rede.
- Um módulo de Logger, que responsável pela auditoria das acções despoletadas no AR.

#### C. Bandwidth Broker (QoS Broker)

É um dos componentes mais importantes de toda a rede. O QoS Broker é a entidade que toma as decisões de controlo de admissão, e configura todos os componentes de acesso da rede de acordo com um conjunto de condições definidas administrativamente.

Esta entidade interage com o Servidor de AAAC, outros Brokers, e os routers da rede.

#### D. Servidor de AAAC

O servidor de AAAC é constituído por diferentes módulos dos quais se salientam os seguintes:

- User Profile - contém os dados de um utilizador específico (username, dados de autenticação, SLA, informação referente a charging, políticas, etc).
- Servidor Diameter - mantém a interface com outras entidades, usando o protocolo DIAMETER [4]
- Accounting BD - relacionado essencialmente com a contagem de recursos usados
- Módulo de Charging - calcula o preço pelo consumo de uma serviço prestado
- Interface com o Broker- usada para enviar ao Broker o NVUP (*Network View of the User Profile*). É também usada para enviar a descrição de NetServices
- Auditing - detecção de violação de Services Level Garantees (SLGs) entre o consumidor e o que fornece o serviço.

#### E. Paging Agent

O software de paging tem como função a geração e distribuição de mensagens de paging, bem como coordenação de todo o processo de paging. Recebe o tráfego destinado ao Mobile Nodes (MN) adormecido, que foi registado com o Paging Agent. Tem a função de registar o seu endereço como CoA alternativo no Home Agent, para os MN que pretendem passar ao estado adormecido.

### F. Home Agent

Home Agent existe devido ao suporte de mobilidade deste demonstrador. Este é fornecido pela stack Mobile IPv6 do MIPL. O Home Agent funciona como uma proxy para os Mobile Nodes que estão registados na sua Home Network mas que estão ligados através de outras redes. O Home Agent é o responsável pela entrega de pacote para os Mobile Nodes que estão registados perante ele. Por isso o Home Agent necessita de ser informado da localização actual do Mobile Node. O Home Agent tem que suportar a alteração do registo do CoA que é necessário para o paging.

### G. Software de demonstração

Para a demonstração do projecto foi necessário a utilização de software que permitisse a recepção e emissão de vídeo/som com suporte para IPV6. As aplicações instaladas foram o VideoLan e Rat (Robust Audio Tool).

## V. TESTES EFECTUADOS

Os testes efectuados tiveram como objectivo obter resultados para a apreciação da robustez e capacidade da rede, em ambientes móveis. De modo a analisar os tempos de comunicação, foi necessário proceder à instalação do *ntp* (network time protocol) para sincronizar todos os relógios dos computadores. Foi ainda possível comprovar (cf. [1-2]) que a especificação dos diversos protocolos estava correcta através dos diagramas de fluxo de mensagens, para cada teste.

### QoS Broker <=>Access Router

Para medir os tempos de resposta do QoS Broker às solicitações do AR foram feitas medidas com o QoS Broker a correr no *Taiqom* e usados os AR no *Kenny* e *Cartman*. A Tabela 1 mostra o resultado desses testes relativamente a diferentes mensagens.

Mensagem	Tempo de resposta ( $\mu$ s)
Client-Open	81
Pedido de Configuração	54320
Recusa de Pedido acesso	733
Aceitação de pedido de core	1537
Aceitação de pedido de acesso	3857
Keep-Alive	140
FHO	17746

Tabela 1 - Tempos de resposta às solicitações do AR

O tempo de aceitação de um novo cliente é extremamente rápido (81  $\mu$ s) o que era esperado pelo facto de ser uma acção muito simples, só necessitando o QoS Broker de criar uma entrada na tabela dos AR ligados antes de poder responder;

O tempo de resposta ao pedido de configuração é o mais elevado de todos os tempos que foram medidos, o que se deve ao facto de o QoS Broker necessitar de procurar qual o tipo de AR em questão e depois procurar na base de dados qual a configuração que lhe deve entregar. Este valor elevado não representa qualquer problema, uma vez que a configuração completa do AR só se faz quando este arranca;

A recusa de um pedido é mais rápida do que a aceitação porque quando o QoS Broker recusa um pedido não necessita reservar os recursos, uma vez que não há atribuição;

A resposta aos pedidos do core é mais rápida porque de acordo com a arquitectura da rede o core é uma zona "privilegiada" da rede, de onde não se verificam os perfis dos utilizadores;

O tempo de resposta a um pedido de *Fast Handover* é demasiado alto, podendo comprometer a resposta da rede ao movimento de um *Mobile Node*. Idealmente segundo os requisitos do projecto *Moby Dick* o tempo de resposta de toda a rede deveria ser da ordem do tempo que é actualmente gasto pelo QoS Broker. Existem dois motivos para que isto não aconteça. Em primeiro lugar a implementação existente usa um vector para armazenar os dados dos ARs pertencentes à rede e a lista de sockets de comunicação com os ARs. Esse tipo de implementação torna lento o processo de procura. Como no processo de FHO, se começa por procurar qual o AR apropriado (o que tem o endereço indicado no pedido) e depois se procura o socket que existe aberto para se poder enviar a resposta, a resposta do QoS Broker será algo lenta. Em segundo lugar, o QoS Broker imprime no interface do gestão dados referentes ao pedido recebido, aos serviços a transferir para o novo AR, bem como a composição do NVUP depois do processo de *Fast Handover* ter decorrido. A impressão destes dados é feita de modo a dar informação ao administrador dos serviços que são transferidos e qual o estado do processo de *Fast Handover*. Como esta informação pode ser muito extensa, vai adicionar muito tempo à resposta ao pedido de *Fast Handover*.

### QoS Broker <=>AAAC

Foram também medidos valores dos tempos de resposta às solicitações do servidor de AAAC. A Tabela 2 mostra os valores medidos para cada um desses pedidos.

Mensagem	Tempo de resposta ( $\mu$ s)
Client-Open	84
Definição dos perfis de QoS	384305
Autorização de NVUP	4007

Tabela 2 - Tempos de resposta às solicitações do servidor AAAC

Analisando os valores da tabela pode-se concluir que: a resposta à mensagem de estabelecimento de ligação tem um valor bastante baixo (84  $\mu$ s) de acordo com o que

acontecia entre o AR e QoS Broker. A mensagem de definição dos perfis de QoS contém muita informação que tem que ser lida, interpretada e armazenada pelo QoS Broker, o que justifica que tenha uma resposta tão demorada. Não se trata de um problema de implementação, uma vez que esta acção só é executada quando o servidor de AAAC se liga; o tempo de resposta do AAAC no envio de um NVUP é de 4 ms, o que parece perfeitamente aceitável.

#### **QoS Broker <=>Radio Gateway**

A Tabela 3 apresenta as mensagens entre QoS Broker e a Radio Gateway. Entre a RG e o QoS Broker existe sempre um AR que intervém no processo de pedido de recursos, que despoleta o pedido ao QoS Broker. Esse AR necessita ainda de uma resposta do QoS Broker de modo a que possa conduzir a informação que vem ou vai de ou para a rede da Radio Gateway. É essa a razão pela qual foram medidos dois tempos que aparecem na Tabela 3.

<i>Mensagem</i>	<i>Tempo de resposta em <math>\mu</math>s</i>
Contacto da RG	1079
Resposta ao AR	2388

Tabela 3 - Tempos de resposta a um pedido de um Mobile Node pertencente a uma rede de uma Radio Gateway

Verifica-se que o tempo de resposta à Radio Gateway é mais baixo do que ao AR, devido à forma como o QoS Broker está implementado. O QoS Broker executa a rotina que faz o controlo de admissão e que decide aceitar um fluxo. O QoS Broker também executa a rotina que procura se a origem ou destino do fluxo a ser admitido está numa rede de uma Radio Gateway. Caso isso aconteça ele envia a mensagem à Radio Gateway imediatamente e só depois responde ao AR.

Verificou-se ainda que o tempo de resposta ao AR é mais elevado na situação em que não intervém a Radio Gateway. Esta diferença deve-se ao facto de que neste caso o QoS Broker necessita de perder algum tempo a construir e a enviar a mensagem à Radio Gateway.

#### **QoS Manager**

##### *Atrasos ponto a ponto*

Para medir este tipo de atraso, efectuou-se o registo de um *Mobile Node*. Neste registo, o *Mobile Node* efectua o registo de Mobile IPv6 no Home Agent, este pacote é autorizado no QoS Manager, e o pacote de confirmação do Home Agent também é autorizado. Com a diferença de tempos entre a altura que o pacote parte do Mobile Node e chega a resposta pode-se calcular o atraso ponto a ponto do primeiro pacote (e resposta) enviado.

O atraso medido médio foi de 3,7 ms, o que é satisfatório.

##### *Atraso em pedidos de serviço / autorização*

Para a medição deste atraso pretendia-se medir o tempo entre a recepção do pacote que despoleta o pedido de serviço/autorização e o encaminhamento desse pacote. No entanto, esta medição não é de todo fiável devido à forma como o AR está implementado. Quando um pacote chega ao AR e necessita de ser autorizado, o AR descarta o pacote, envia o pedido de serviço/autorização ao QoS Broker e recebe a implementação do filtro, caso a resposta seja positiva. Entretanto o pacote descartado é substituído por um outro enviado pelo emissor do primeiro. Em consequência, este valor varia de acordo com a implementação do protocolo e da forma como o software usado foi feito.

De notar que este processo ocorre tanto para acessos de core como para acessos da rede de acesso.

##### *Atraso na implementação ou actualização do filtro*

Neste teste mediu-se o tempo entre a recepção do pacote COPS DEC que informa o AR que pode aceitar o fluxo em causa, e o envio do pacote COPS RPT que informa o QoS Broker que já configurou o filtro para esse fluxo.

Este atraso é de cerca de 420  $\mu$ s.

##### *Perda de pacotes*

Devido à forma como o Access Router está implementado existe sempre perda de pacotes. (O AR descarta o pacote que despoleta o pedido de serviço/autorização, como explicado anteriormente).

Desta forma é natural que haja uma perda de pacotes significativa caso os fluxos que os originam necessitem de ser autorizados. Caso haja uma demora na actualização/autorização de um fluxo os pacotes são perdidos pois não são retidos para que posteriormente possam ser encaminhados.

##### **Paging**

Nestes testes foram usados as versões *Dummy* de todo o *software* da rede *Moby Dick*, de forma a facilitar os testes. Para o cálculo de todos os atrasos, foi efectuado o mesmo teste 3 vezes, de modo a ter tempos fiáveis, e foram contabilizados os tempos de cada pacote. Os tempos apresentados são as médias dos 3 testes efectuados.

##### *Tempo total numa operação de paging*

Mediu-se o tempo que decorre desde que se faz um pedido ao Mobile Node até que ele responde. Foi usado o comando *ping* e foi medido o tempo que demora a chegar um pacote do tipo *echo reply* ao Correspondent Node. Este tempo de atraso inclui o armazenamento de pacotes iniciais no Paging Agent, o processo de *paging*, o tempo de re-registo do Mobile Node na rede e finalmente o envio da resposta. Actualmente este valor cobre não só o tempo adicional do *paging* mas todo o processo até à resposta. De modo a obter uma comparação, pode-se medir o tempo de resposta de um Mobile Node activo, embora possa implicar um caminho diferente. Pretende-se também medir se existe ou não perda de pacotes. A azul encontram-se o *Ping Request* (1) e o *Ping Reply* (12), a

laranja as mensagens do processo de *paging* (2,3,4,7,8,9,10,11) e a verde as mensagens de *Mobile IPv6* (5,6).

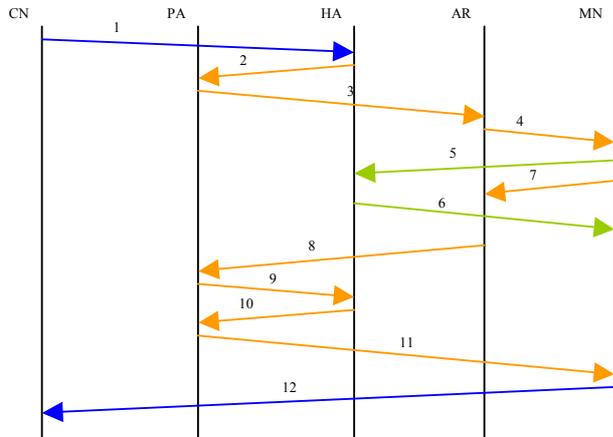


Diagrama 1 – Fluxo de mensagens trocadas numa operação paging

O Diagrama 1 mostra o fluxo de mensagens trocadas entre os diversos componentes de rede para o teste em causa. A forma como o teste foi efectuado permitiu que com um único teste fosse possível efectuar as medições necessárias para outros testes planeados.

Nº	Origem (µs)	Destino (µs)	Diferença (µs)
1	0	147	147
2	238	579	341
3	1592	2246	654
4	2534	3791	1257
5	3995	6004	2009
6	6178	7924	1746
7	4280	7086	2806
8	7398	7711	313
9	8514	8878	364
10	8930	9408	478
11	9556	10962	1406
12	11056	13354	2298

Tabela 4 - Tabela de tempos do fluxo de pacotes do Diagrama 1

Pela tabela de tempos (Tabela 4) pode-se verificar que o tempo total de uma operação de *paging* é cerca de 13,4 ms. Este valor mostra um excelente desempenho da rede, tendo em conta que grande parte do tempo é gasto na propagação dos pacotes, principalmente quando é utilizada tecnologia *wireless*. Quanto a perda de pacotes, não se registou em nenhum teste de *paging* qualquer perda de pacotes.

**Fast Handover**

O objectivo deste teste é verificar o desempenho de um *Fast Handover* numa situação real, em que todo o software é usado e existe algum tráfego na rede. Este processo tem em conta também as mensagens de AAAC

que não influenciam a decisão de existir ou não *Fast Handover*, pelo que é possível concluir que todo o processo de *Fast Handover* termina com a recepção da mensagem 9 ao *Mobile Node*. No Diagrama 2 os diferentes protocolos estão identificados a cores diferentes: a azul estão a mensagens de *Fast Handover* (1,2,4,5,8,9), a verde as mensagens de QoS (3,6,7) e a laranja as mensagens de AAAC (10,11).

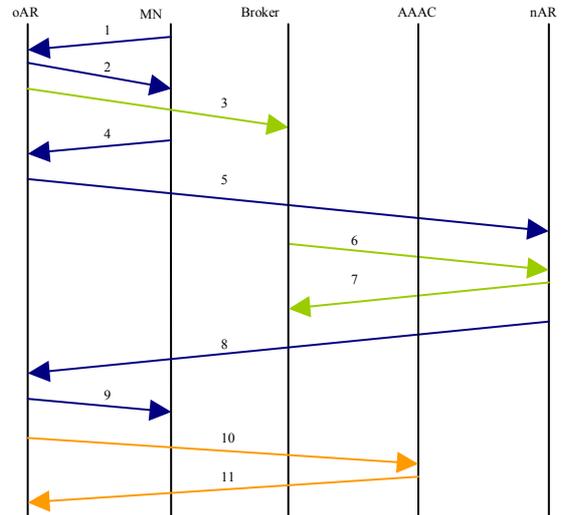


Diagrama 2 - Mensagens de Fast Handover com AAAC e QoS

Nº	Origem (µs)	Destino (µs)	Diferença (µs)
1	0	1657	1657
2	1938	3241	1303
3	2612	2844	232
4	3307	4749	1442
5	5677	6280	603
6	21366	21725	359
7	22898	23990	1092
8	23133	24294	1161
9	24568	25893	1325
10	122646	123208	562
11	139620	139966	346

Tabela 5 – Tempos medidos do fluxo do Diagrama 2

*Latência Geral Durante o Fast Handover*

Verificou-se que o processo de *Fast Handover* dura cerca de 140 ms. Tal como previsto o tempo necessário para efectuar o *Fast Handover* em si, sem ter em conta as mensagens de AAAC que não influenciam a decisão de *Fast Handover*. O tempo é o mesmo tendo todo o software integrado, ou usando o *dummy* de AAAC: cerca de 25 ms. Esta situação mostra a robustez de todo o processo.

O atraso principal encontra-se no tempo de espera entre as mensagens finais de *Fast Handover* e a primeira mensagens de AAAC a ser enviada.

*Pacotes perdidos durante o Fast Handover*

Nos testes de *Fast Handover* não foi registado nenhum pacote de sinalização perdido. No entanto, verificou-se que havia pacotes, neste caso ping, que eram perdidos

durante um *Fast Handover*. Este problema deve-se a alguns problemas ainda existentes nesta componente. P.ex., uma das razões é o facto de os pacotes de registo de *Mobile IPv6* não chegarem ao *Home Agent*, o que faz com que o *Mobile Node* não receba os pacotes que para ele são direccionados.

#### *Atraso específico entre o QoS Manager e QoS Broker*

As comunicações de QoS durante o processo de *Fast Handover* atrasam o referido processo em média 18 ms. Este tempo é justificável devido às verificações internas que o *QoS Broker* tem que efectuar de modo a garantir ao *Mobile Node* os serviços por ele contratados.

#### *Atraso específico entre AAAC Attendant e AAAC Server*

O atraso nas comunicações de AAAC em toda a rede durante o *Fast Handover* deve-se ao facto do oAR informar o AAAC Server dos valores contados durante o uso do *Mobile Node* nesse AR. Pelo facto de estas mensagens terem um carácter informativo e não decisivo para a concretização ou não do *Fast Handover*, pode-se admitir que não existe atraso do ponto de vista do *Fast Handover* devido as comunicações AAAC.

#### *Largura de banda de um fluxo TCP num FHO*

Foi feito um teste da variação da largura de banda dos fluxos TCP durante um FHO. Verificou-se que a largura de banda na altura do *Fast Handover* (seg, 99023) decresce ligeiramente (Fig. 3). Este decréscimo deve-se ao facto de as mensagens de *bicast* diminuírem ligeiramente a largura de banda útil durante um processo de *Fast Handover*.

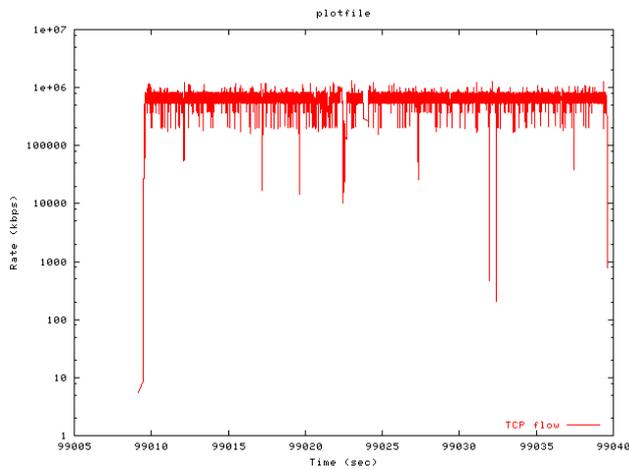


Figura 3 - Gráfico da largura de banda utilizada num FHO

## VI. SOFTWARE DESENVOLVIDO

Além de módulos de teste, foram ainda desenvolvidas duas componentes essenciais para o suporte de QoS.

### A. PTQoS

Tendo por base o QoSManager desenvolvido no âmbito do projecto Moby Dick e as suas limitações foi requerido

que se desenvolvesse um substituto capaz de substituir o QoSManager Moby Dick sem qualquer outra modificação ao restante software existente na rede Moby Dick. O suporte funcional deste software seria apenas QoS e não necessitaria de suportar FHO.

Tendo por base estes requisitos foi desenvolvido um novo QoS Manager, PTQoS, recorrendo às mesmas API's utilizadas pelo QoSManager do projecto Moby Dick.

A principal lacuna que o PTQoS combateu foi a flexibilização do numero de interfaces suportadas tanto a nível de core como a nível de acesso. Através de um desenvolvimento Orientado por Objectos em C++, criou-se uma abstracção que permite configurar várias interfaces de rede, tanto ligadas à rede de core como à rede de acesso. Outro aspecto importante foi a substituição dos vários processos existentes no QoSManager Moby Dick por um só processo com Threads. Esta substituição torna não só o QoSManager mais estanque como substituiu a necessidade de recorrer a memoria partilhada para troca de informação (processo lento para a aplicação em causa).

O uso das API's utilizadas pelo QoSManager Moby Dick restringiu as possíveis funcionalidades e capacidades do PTQoS desde muito cedo. Nomeadamente o facto da COPS API (componente essencial do software do Moby Dick) não ser "Thread safe".

Esta e outras limitações levaram a que se pensasse numa evolução de grau superior à do PTQoS que veio dar origem ao SPAAQE, o qual se descreve de seguida.

### B. SPAAQE

O SPAAQE é na sua essência um Edge Router de arquitectura DiffServ (DS). No entanto apresenta várias possibilidades que o tornam numa ferramenta poderosa de gestão e configuração de fluxos DS. O SPAAQE poderá ser utilizado em redes em que é necessário fazer admissão de fluxos vindos de diversos utilizadores com requisitos muito personalizados.

O SPAAQE tem pois por alvo Service Providers que desejem dar aos seus clientes níveis de qualidade de serviço altamente configuráveis, mantendo um elevado grau de gestão sobre a sua rede. O SPAAQE autentica, classifica e autoriza os fluxos por meio de informação trocada com uma rede QoS Broker's central [5]. Desta forma o SPAAQE torna-se ideal para redes que desejem ter vários Edge Routers com todas as funcionalidades do SPAAQE, e que necessitem constantemente de reconfigurações e/ou pequenas alterações.

Outra função importante do SPAAQE é a sua capacidade de processar os pacotes que o atravessam. O exemplo de um cenário onde esta função poderá ser útil é na criação de uma directoria de endereços para serviços. Concretizando no caso do DNS, o SPAAQE poderia fazer correspondência entre DNS server, configurado no utilizador e o endereço de DNS que conste numa directoria configurada no SPAAQE. Desta forma o

Service Provider poderia distribuir configurações aos seus utilizadores que nunca correriam risco de se tornar obsoletas, ou quando necessitasse de distribuir serviços por várias máquinas, não necessitaria de informar o utilizador do novo endereço do serviço (embora existam outras formas de realizar este exemplo específico).

A capacidade de sinalização de fluxos a terceiras entidades é uma opção flexível, que permite a expansão dinâmica de funcionalidades do SPAAQE. Desta forma e através de todas estas novas funcionalidades o SPAAQE apresenta-se como uma alternativa consistente e de valor acrescentado ao QoS Manager do Projecto Moby Dick.

### C. COPSpp

Apesar de existirem já diversas implementações do protocolo COPS tais como API da Intel e a API utilizada no Projecto Moby Dick, nenhuma destas responde aos requisitos estabelecidos para utilização no SPAAQE. A primeira por apenas disponibilizar source code para PEPs e o mesmo ser em C, a segunda por ser uma implementação demasiado simplista e também ela ser em C. Assim desenvolveu-se uma nova API Open Source em C++ que respeita o RFC 2748 (The COPS Protocol), é "thread-safe" e implementável tanto para uma interface para PEP's como para PDP's.

A COPSpp (leia-se COPS plus plus) é pois uma API C++ distribuída na forma de shared library que permite o desenvolvimento de aplicações com suporte a funcionalidades de comunicação via COPS.

## VII. CONCLUSÕES

Este artigo descreve um demonstrador implementado para suportar redes móveis heterogêneas. Esse demonstrador associa aspectos de mobilidade, QoS, AAAC, e paging numa mesma infra-estrutura. A utilização do protocolo IP como base neste tipo de infra-estrutura permite a interligação de diferentes redes de acesso. No entanto é claro que a rede global é bastante complexa e tem múltiplas interações, num número muito elevado de entidades funcionais.

No âmbito deste demonstrador foram feitas diversas medidas de desempenho, em termos de perdas, tempos de resposta, e larguras de banda. Foram identificadas as

operações mais longas, em termos temporais, e a sua relevância em termos de uma rede real suportando serviços ao utilizador.

Foram ainda desenvolvidas duas ferramentas de software, o PTQoS e o SPAAQE, que aumentaram as potencialidades da rede. Este desenvolvimento deu ainda uma visão mais completa da complexidade inerente ao desenvolvimento de uma plataforma tão flexível como esta.

A presença do projecto nas conferências científicas Wireless Summit 2003 e ConfTele 2003 permitiram ainda dar uma visão ao exterior do trabalho realizado, e realizar experiências reais com utilizadores não preparados. Foi uma experiência enriquecedora na medida que se pode expor o trabalho e interagir com a comunidade recebendo feedback do mesmo.

Por fim, e de uma forma mais subjectiva, é uma conclusão clara da implementação deste demonstrador que sistemas desta complexidade somente podem ser desenvolvidos por equipas multi-disciplinares, com intensa troca de experiências e conhecimentos.

## VIII. REFERÊNCIAS

- [1] Einsiedler, R.L. Aguiar, J. Jähnert, K. Jonas, M. Liebsch, R. Schmitz, P. Pacyna, J. Gozdecki, Z. Papir, J.I. Moreno, I. Soto, *The Moby Dick Project: a Mobile Heterogeneous All-IP Architecture*, Advanced Technologies, Applications and Market Strategies for 3G - ATAMS 2001, ISBN 3-88309-20-X, pp. 164-171, Kraków, 2001.
- [2] Víctor Marques, Rui L. Aguiar, Carlos Garcia, Jose Ignacio Moreno, Christophe Beaujean, Eric Melin, Marco Liebsch, *An IP-based QoS architecture for 4G operator scenarios*, IEEE Wireless Communications Magazine, June 2003.
- [3] Nichols, K., Blake, S., Baker, F. e D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.
- [4] P. Calhoun, et al., "Diameter Base Protocol"; Internet Draft, work in progress, draft-ietf-aaa-diameter-11.txt, June 2002.
- [5] Pedro Gonçalves, Diogo Gomes, Victor Marques, Rui L. Aguiar, QoS Control support for heterogeneous networks, CRC 2003, 2003.