Relatório de Estágio Formal

Implementação prática da rede Wireless do INESC no Projecto Europeu IST/MOICANE

Candidato: Pedro Miguel dos Santos Reis Vale Estrela

Mestrado em Engenharia Informática e de Computadores pelo IST

Membro Estagiário da Ordem dos Engenheiros Nº 9407,

Inscrito na Região Sul Nº 4353, do Colégio de Engenharia Informática

Patrono: Prof. Doutor Mário Serafim dos Santos Nunes

Doutorado em Engenharia Electrotécnica e de Computadores pelo IST

Membro Efectivo da Ordem dos Engenheiros Nº 29302,

Especialista em Telecomunicações,

Inscrito na Região Sul Nº 20295, do Colégio de Engenharia Electrotécnica

Entidade: INOV – INESC Inovação

Início do Estágio: Julho 2002

Fim do Estágio: Janeiro 2003

Sumário

O objecto do presente relatório é a descrição do trabalho realizado no âmbito do Programa de Estágio Formal do Colégio de Engenharia Informática da Ordem dos Engenheiros, que decorreu de Julho de 2002 a Janeiro 2003, com vista à passagem a Membro Efectivo do Membro Estagiário Pedro Vale Estrela, Mestrado em Engenharia Informática e de Computadores, tendo como Patrono o Eng. Mário Serafim Nunes, do Colégio de Engenharia Electrotécnica, Doutorado em Engenharia Electrotécnica e de Computadores.

O referido trabalho foi realizado na instituição INOV (INESC Inovação), tendo sido enquadrado no projecto Europeu Multiple Organisation Interconnection for Collaborative Advanced Network Experiments (MOICANE), financiado pelo programa Information Society Tecnology (IST). O trabalho centrou-se no trabalho de Engenharia necessário para a **implementação** prática da rede Wireless do INESC no referido Projecto Europeu, integrando o desenvolvimento e implementação de tecnologias de ponta numa solução completa, e incluíndo também aspectos de Investigação.

Índice

Lista de Apêndices

Lista de Figuras

Lista de Tabelas

1. Introdução

1.1 Objecto do relatório

O objecto do presente relatório é a descrição do trabalho realizado no âmbito do Programa de Estágio Formal do Colégio de Engenharia Informática da Ordem dos Engenheiros, que decorreu de Julho de 2002 a Janeiro 2003, com vista à passagem a Membro Efectivo do Membro Estagiário Pedro Vale Estrela, Mestrado em Engenharia Informática e de Computadores, tendo como Patrono o Eng. Mário Serafim Nunes, do Colégio de Engenharia Electrotécnica, Doutorado em Engenharia Electrotécnica e de Computadores.

O referido trabalho foi realizado na instituição INOV (INESC Inovação), tendo sido enquadrado no projecto Europeu Multiple Organisation Interconnection for Collaborative Advanced Network Experiments (MOICANE), financiado pelo programa Information Society Tecnology (IST). O trabalho centrou-se no trabalho de Engenharia necessário para a **implementação** prática da rede Wireless do INESC no referido Projecto Europeu, integrando o desenvolvimento e implementação de tecnologias de ponta numa solução completa, e incluíndo também aspectos de Investigação.

Ao longo deste relatório, serão detalhados tanto o enquadramento institucional da entidade onde se realizou o estágio, bem como a descrição do trabalho de um ponto de vista de Engenharia, nomeadamente focando as metodologias utilizadas na sua resolução, os problemas encontrados, e as conclusões resultantes do mesmo.

O relatório integra também dois pareceres adicionais, que fornecem dados complementares ao presente relatório. O primeiro, da entidade onde se realizou o estágio (INOV), certifica devidamente a sua realização no período temporal do mesmo. O segundo, do Patrono do Candidato, um especialista reconhecido da área, certifica a qualidade tanto do trabalho de Engenharia desenvolvido, como do presente relatório.

1.2 Enquadramento

1.2.1 Enquadramento institucional

O INOV surge no decurso de um amplo processo de reestruturação estratégica encetado pelo INESC em 1998, o qual tinha por orientação a criação de novas organizações especializadas

em diferentes áreas de actividade, por forma a obter uma superior capacidade de resposta mais consentânea com os actuais desafios do mercado.

Na sequência deste processo o INOV concentrou na sua estrutura uma parcela significativa da Área de Electrónica e Telecomunicações do INESC em Lisboa, herdando um reconhecimento pelo universo empresarial de valências tecnológicas ímpares em ambiente marcadamente profissional, como resultado de provas dadas na capacidade de transferência tecnológica para empresas já existentes ou emergentes.

Neste contexto, o INOV posiciona-se no mercado como a maior infra-estrutura tecnológica nacional do domínio das Tecnologias de Informação, Electrónica e Comunicações, porquanto herda um importante capital decorrente das inúmeras experiências e sinergias subjacentes ao seu ambiente de incubação - o INESC, dando início formal à sua actividade, no dia 1 Janeiro de 2001, enquanto associação privada sem fins lucrativos.

O INOV disponibiliza uma organização ágil e flexível, orientada à criação de competências tecnológicas e ao estabelecimento de laços de cooperação com os diferentes actores económicos (Universidades, Indústrias, Empresas, Operadores de Telecomunicações). Pretendendo para o efeito desenvolver uma actividade charneira entre a Universidade e as Empresas, alicerçada numa privilegiada cooperação com a Universidade, por forma a disponibilizar de modo sustentável, consistentes e inovadoras soluções face aos problemas e desafios enfrentados pelos seus parceiros.

Neste sentido, o INOV, enquanto parceiro tecnológico, poderá auxiliar as empresas na procura de novas oportunidades de negócio, mediante o desenvolvimento de soluções tecnológicas inovadoras, com elevado valor acrescentado, cruciais para a construção de vantagens comparativas que possibilitam alcançar elevados graus de competitividade e, nesse âmbito, permitem pro-activamente participar no processo de desenvolvimento sócio-económico nacional.

Para manter uma actuação de excelência técnica, o INOV aposta no conhecimento e aprendizagem, potenciando a realização de estágios, enquadrados em projectos de I&D.

No caso concreto deste estágio profissional, estes compromissos verificam-se na sua totalidade, dado se tratar de um trabalho de I&D que alia de forma perfeita a aprendizagem e o conhecimento nas vertentes complementares de Investigação e Desenvolvimento em tecnologia de ponta.

1.2.2 Motivação

O projecto Europeu MOICANE (www.moicane.com) tem por objectivo o estudo de mecanismos de suporte de Qualidade de Serviço (QoS – Quality of Service) extremo-a-extremo em redes de diferentes tecnologias.

O projecto conta com a participação de vários parceiros internacionais, dos países Portugal, Itália, Grécia, Roménia e França. Cada parceiro tem a seu cargo a criação de uma ilha experimental, constituída por várias redes de acesso e uma rede de *core*, onde vão ser implementadas e testadas as tecnologias mais recentes no suporte de Qualidade de Serviço IP no modelo *DiffServ* [10], e também de outras tecnologias IP emergentes, sendo estas aplicadas a várias tecnologias com e sem fios, como o 802.3, 802.11, xDSL e *bluetooth*.

A rede do MOICANE foi comprovada experimentalmente, através da utilização de aplicações com diferentes requisitos de qualidade e características de tráfego, nomeadamente do tipo Ensino à distância ("e-learning"), Laboratório Virtual, e outras aplicações multimedia remotas, como Voz sobre IP (VoIP), ou "video on demand" (VoD).

Em Portugal, o Instituto de Engenharia de Sistemas e Computadores (INESC) participa como parceiro durante todo o projecto, com uma ilha experimental que conta com as tecnologias de acesso *wireless* IEEE 802.11 e *Bluetooth*, além do acesso fixo Ethernet/802.3, estando o desenvolvimento do mesmo a cargo da sub-entidade INOV – Inesc Inovação.

Especificamente, a rede de acesso *wireless* IEEE 802.11 do INESC distingue-se de todas as outras de todos os parceiros por proporcionar um ambiente móvel sem fios no qual se vai integrar o suporte de qualidade de serviço IP com a mobilidade IP. Desta forma, no contexto de investigação do MOICANE, a rede de acesso 802.11 vai suportar os serviços de mobilidade e de qualidade de serviço estática.

Relativamente ao suporte de mobilidade, os terminais móveis vão ser "legacy" (sem suporte explicito de mobilidade), e vão-se movimentar no interior da rede sem limitações, uma vez que utilização tecnologia de acesso *wireless*. Do lado da rede, esta é constituída por elementos que são entidades IP ("routers"), o que obriga à implementação de uma solução de mobilidade IP adaptada a este cenário. Para este cenário, foi desenvolvida, implementada e testada uma nova proposta de micro-mobilidade IP – "Terminal Independent Mobility for IP" (TIMIP) [65]– o qual foi recentemente proposto como objecto de investigação no organismo próprio de normalização IP – o Internet Engineering Task Force (IETF).

Relativamente ao suporte de Qualidade de Serviço, esta foi implementada segundo o modelo Diffserv, com componentes adaptadas para o ambiente wireless 802.11. Este suporte estático é baseado na agregação de tráfego, incluindo as componentes de classificação, "metering", policiamento e "scheduling", de forma a possibilitar a criação e gestão de serviços diferenciados, desde a prioritização simples até à partilha de largura de banda entre diferentes classes de tráfego.

Assim, a motivação de este estágio integrado num contexto de I&D de um projecto europeu fornece uma oportunidade única de trabalhar e investigar naquilo que é o estado da arte nas redes móveis, em tecnologias de reconhecida ascensão – Mobilidade, Qualidade de Serviço e Redes sem Fios - no âmbito das Redes e Sistemas de Computadores

1.3 Objectivos e Avaliação do Trabalho

O presente estágio foca os *aspectos de Engenharia* utilizados na implementação da rede móvel protótipo, que incluem as componentes inovadoras necessárias no contexto do projecto de Investigação.

Neste sentido identificaram-se os seguintes objectivos do trabalho (conforme o programa de estágio):

- Criação, Desenho e Implementação da rede de acesso wireless 802.11 do INESC no MOICANE.
- Implementação prática, no sistema operativo Linux, de um protótipo do protocolo de micro-mobilidade TIMIP, que suporte a mobilidade optimizada de terminais "legacy" no interior da rede wireless.
- Implementação prática, no sistema operativo Linux, de um mecanismo de suporte de qualidade de serviço baseado no modelo DiffServ.

Complementar a qualquer identificação de Objectivos será a necessária Identificação e Avaliação dos resultados. No presente estágio, essa é constituída por duas componentes:

- Demonstração prática da rede de acesso wireless a Auditores Internacionais independentes, encarregues de avaliarem a qualidade global do projecto MOICANE, e a outros especialistas Nacionais e Internacionais;

- Elaboração de um relatório de estágio – o presente relatório – onde estão detalhadas as opções, metodologias e componentes especificas da solução final, considerando a visão prática da implementação a realizar.

1.4 Estrutura do relatório

Este Relatório está estruturada em 5 capítulos e complementada com 2 Pareceres e 11 anexos.

No primeiro capítulo foi efectuado o enquadramento e a motivação do estágio, enunciados os objectivos e avaliação do trabalho e definida a sua estrutura.

No capítulo 2 é descrito o protótipo desenvolvido no estágio - a rede de acesso *wireless* do MOICANE. Assim, vai ser descrito o demonstrador internacional do MOICANE, a ilha do INESC e a rede de acesso 802.11, esta última em relação à sua arquitectura, constituintes, e interacções (internas e externas). Este capítulo é complementado com a descrição sumária das tecnologias que foram utilizadas nesta rede de acesso, dividindo-se entre tecnologias de rede e de suporte de QoS.

O capítulo 3 foca em detalhe o trabalho de Engenharia necessário para a implementação desta rede, verificando-se as entidades, protocolos e mecanismos que foram desenvolvidos, e as suas relações entre si. Estas funcionalidades foram implementadas em exclusivo nos nós da rede, com base no sistema operativo Linux, que foi estendido com mecanismos adicionais programados em linguagem C. Por outro lado, certos mecanismos já existentes foram reaproveitados, sendo integrados ou costumizados para utilização em módulos das entidades da rede, nomeadamente as interacções com o *driver* 802.11, módulos de IP, encaminhamento, *forwarding*, NAT e *firewalling*.

O capítulo seguinte está detalhada a componente de avaliação do estágio relativo à demonstração prática da rede de acesso wireless a Auditores Internacionais independentes, e outros especialistas Nacionais e Internacionais, que verificaram a qualidade da solução desenvolvida, por via de testes de desempenho, com aplicações multimédia que evidenciam totalmente as características inovadoras, e com aplicações de teste com capacidade de efectuar medições rigorosas dos parâmetros da rede.

No último capítulo, apresenta-se as conclusões a respeito do trabalho de engenharia efectuada, dos resultados e das perspectivas de trabalho futuro.

Como complemento ao relatório, são apresentados dois pareceres adicionais do estagio (pareceres da entidade e do Patrono), e vários Apêndices que são relativos a outros pormenores da implementação não focados no texto principal.

O trabalho termina com uma lista de referências bibliográficas, e com um Glossário dos termos e abreviaturas utilizadas.

2. Descrição do Protótipo

Este capítulo descreve o protótipo desenvolvido no estágio - a rede de acesso *wireless* do MOICANE. Neste sentido, vai ser descrito o demonstrador internacional do MOICANE, a ilha do INESC e a rede de acesso 802.11, esta última em relação à sua arquitectura, constituintes, e interacções (internas e externas).

Numa segunda secção, são analisados genericamente a arquitectura dos elementos de redes, identificando os módulos existentes e as suas relações internas, e as operações efectuadas aos pacotes de dados no interior do encaminhador;

2.1 Descrição da rede de acesso

2.1.1 Enquadramento da rede de acesso no MOICANE

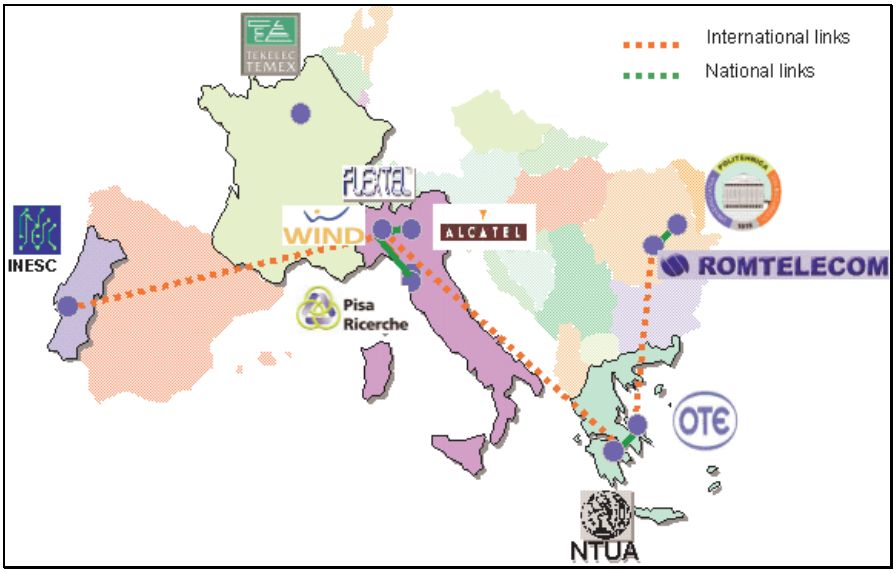


Figura 1: Arquitectura global do MOICANE

A rede internacional de demonstração criada para o projecto MOICANE está delineada na Figura 1, onde várias ilhas pertencentes a cada parceiro estão interligadas entre si por ligações internacionais. O objectivo principal deste projecto será o de analisar as novas tecnologias IP de suporte de qualidade de serviço, nomeadamente o Diffserv [10] e o Intserv [15], com a análise destas tecnologias verificada tanto num contexto local, onde se considera apenas o interior de cada ilha por si, mas também numa perspectiva global ao longo de todas as ilhas e ligações internacionais, por forma a existir durante todo o caminho dos fluxos de dados um suporte de qualidade de serviço permanente extremo-a-extremo.

Desta forma, cada participante no projecto tem a responsabilidade de criar uma ilha, constituída por um conjunto de redes de acesso e uma rede de suporte (*core*), sendo as primeiras mais vocacionadas para o teste e demonstração de novas tecnologias de rede, e as segundas para o estudo do suporte de QoS, usando mecanismos escaláveis e integrados ao longo dos vários domínios, recriando as necessidades das redes de core da Internet.

O demonstrador internacional final vai ser testado através da utilização de aplicações com diferentes requisitos de qualidade, como sejam aplicações de laboratório virtual e ensino à distância, o que leva a passar pela rede diferentes fluxos de dados prioritários, nomeadamente Voz, Áudio, Vídeo e Dados, que terão que coexistir com tráfego sem requisitos de QoS, como transferências de ficheiros ou simples tráfego de carga da rede.

Assim, será possível demonstrar o suporte de QoS extremo-a-extremo, tanto no interior como ao longo das várias ilhas, pelo que, tanto as ilhas como as ligações internacionais vão ter mecanismos de suporte de QoS, sendo as primeiras escaláveis, flexíveis e da responsabilidade de cada parceiro, e as segundas fixas e da responsabilidade dos vários provedores de serviço que oferecem as ligações ponto-a-ponto entre as ilhas[1].

De um modo global, o suporte do QoS foi realizado principalmente com base no modelo de Serviços Diferenciados, sendo este complementado ou substituído por outros mecanismos complementares caso-a-caso para algumas redes específicas, como a integração do modelo de serviços integrados no Diffserv por conversão.

Neste modelo, cada rede vai constituir um domínio DiffServ, formando a rede do MOICANE uma região DiffServ, e o suporte de QoS entre domínios assegurado através de Níveis de Serviço Contratados (SLAs - *Service Level Agreements*) que, para além dos aspectos contratuais, definem as características do serviço que a rede cliente espera receber e que a rede fornecedora se compromete a assegurar.

2.1.2 Ilha do INESC

Pela sua localização geográfica (sediada em Lisboa), a ilha do INESC está localizada num extremo do demonstrador internacional, estando ligada a este por via de uma ligação internacional para a ilha CPR (Pisa), com uma capacidade garantida de 1.5 Mbit/s

[1] No entanto, devido a dificuldades fora do controlo do projecto, nem todas as ligações internacionais tiveram efectivamente garantias de QoS.

(bidireccionais), e é disponibilizada pelas redes RCCN, GEANT, e GARR usando um túnel "IP dentro de IP" que encapsula os pacotes de dados que passam entre os dois extremos do túnel[2].

A ilha do INESC está resumida na Figura 2, e em grande detalhe no Anexo 1. Esta ilha, de igual forma das dos restantes parceiros, é dividida em duas componentes: várias redes de acesso, referentes às diversas tecnologias a demonstrar, e uma rede de *core* que as interliga, contendo o acesso internacional já referido.

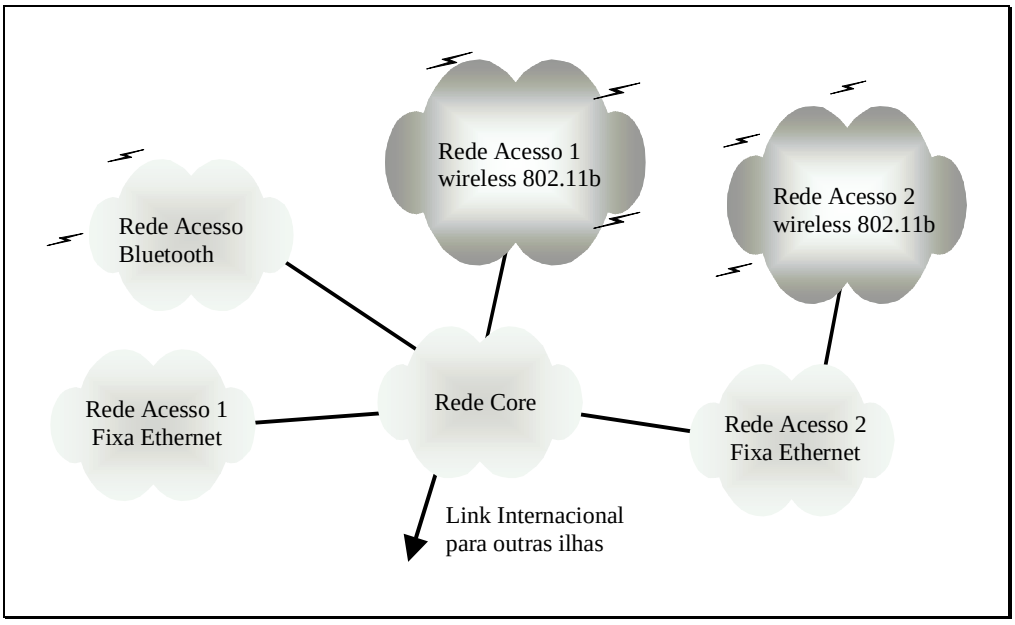


Figura 2: Arquitectura da ilha do INESC

Relativamente às primeiras, as redes de acesso são baseadas nas tecnologias *wireless* 802.11 [17], *wireless* bluetooth e *wired* em ethernet simples, de tal forma a que cada rede é especializada nos diferentes problemas e soluções de cada tecnologia diferente, nomeadamente nas diferenças de suporte de QoS e na mobilidade (este último só no caso das redes 802.11). A rede de core é baseada em tecnologia Ethernet com ligações de 10 Mbits entre os encaminhadores de core; este débito é suficientemente alto para suportar as aplicações de teste da rede, mas ainda é suficientemente baixo para ser possível saturar com tráfego, forçando a ocorrência da congestão necessária para o teste dos mecanismos de QoS.

[2] Este modelo vai ser o mais simples para os provedores dos serviços, por simplificar o encaminhamento nas suas redes de core por apenas terem que ligar entre si pontos específicos das suas redes, e possibilita a máxima flexibilidade às ilhas, que podem acordar entre si livremente as redes IP a usar no demonstrador.

Tem-se ainda que cada rede de acesso das presentes na ilha corresponde a uma subrede IP integrada na rede IP da ilha do INESC, utilizando o mecanismo CIDR[3], adequado para criar pequenas redes que comunicam entre si usando encaminhamento *estático*. Em particular, as redes de acesso *wireless* 802.11 são vistas do exterior como subredes IP, com um único router de acesso à sua rede de adjacente.

2.1.3 Rede de acesso wireless 802.11

Relativamente ao acesso *wireless* 802.11, existem duas redes de acesso independentes para demonstração dos mecanismos de micro-mobilidade, macro-mobilidade e QoS na tecnologia *wireless*. A primeira rede de acesso é a principal, onde serão efectuadas a maioria das experiências; a segunda é apenas uma cópia da anterior, para ser utilizada apenas nas experiências de macro-mobilidade, possibilitado por as duas redes corresponderem a dois domínios TIMIP distintos separados por várias redes IP.

Assim, a rede *wireless* 802.11 principal é a rede de acesso mais complexa da ilha INESC, tendo um nível de complexidade comparável à rede de core, e distinguindo-se de todas as outras no projecto MOICANE por apresentar um ambiente móvel sem fios, onde os terminais se podem mover livremente no interior da rede.

A rede está dividida em duas componentes distintas (ver Figura 3): uma *wireless* móvel que contém os terminais móveis, e uma componente fixa *wired* constituída por encaminhadores (fixos) que formam o *backbone* da rede. Nesta rede de suporte, alguns encaminhadores têm interfaces *wireless* 802.11 para os terminais se ligarem à rede durante o seu movimento, servindo assim de concentradores sem fios, e um destes detém a ligação à rede de core.

Internamente, os encaminhadores fixos estão ligados entre si por ligações dedicadas baseadas em ethernet, de forma que esta rede de suporte só é utilizada para encaminhar pacotes de/para os terminais móveis, restringidos ao interior da rede *wireless* para as comunicações locais entre terminais, e passando pela rede de core INESC no caso contrário.

[3] Classless Inter Domain Routing, um mecanismo que permite dividir redes IP em subredes de várias dimensões.

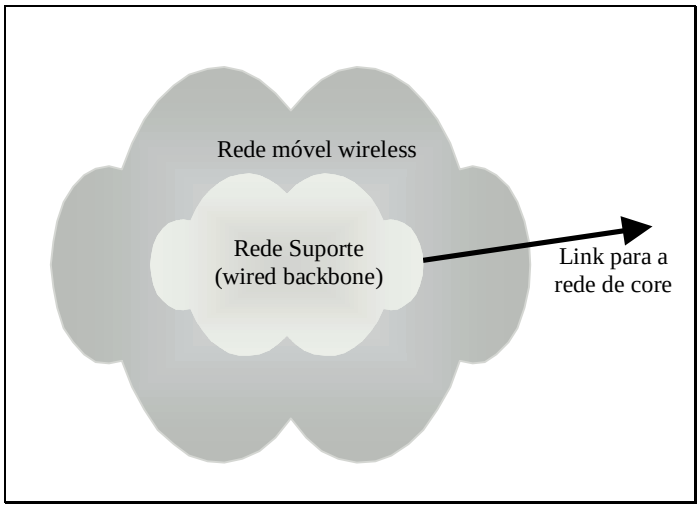


Figura 3: Componentes da rede de acesso *wireless* 802.1

Foi sobre esta rede de suporte que se centrou o trabalho desenvolvido, tendo sido as novas tecnologias IP de QoS e Mobilidade implementadas em exclusivo neste troço da rede, de forma transparente e de modo a que os terminais não participem nestes processos. Para tal, considerou-se que os terminais seriam legados, sem extensões ou alterações no seu *stack* IP, por forma a garantir a compatibilidade com o maior número possível de terminais.

Concretamente, a rede de demonstração está ilustrada na Figura 4, sendo constituída por 3 terminais móveis (C1, C2 e C3) e 3 encaminhadores fixos, dos quais 2 são pontos de acesso 802.11 (AP1 e AP2) e o restante é a GW desta rede.

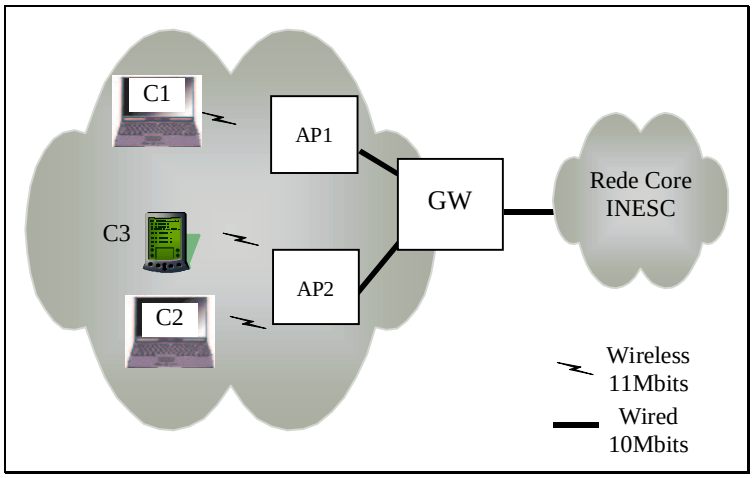


Figura 4: Rede *wireless* 802.11 de demonstração do inesc

Clientes da rede

Relativamente aos clientes da rede, procurou-se ter a maior variedade possível de terminais IP com interface 802.11 existentes, com diferentes opções de hardware e software, de forma a demonstrar o suporte de terminais legados da rede.

Desta forma, a rede conta com 2 PCs portáteis Toshiba com ambos os sistemas operativos Windows 2000 e Linux, e com interfaces PCMCIA 802.11 Lucent[4]; o terceiro cliente da rede é um PDA Siemens com Windows CE e interface 802.11 Linksys.

Estes clientes vão ser utilizados para testar a rede, o que permite verificar o suporte do QoS e da mobilidade que a rede oferece, sendo estas componentes testadas nos níveis *Funcional* e de *Desempenho*, pelo que os clientes vão usar aplicações com requisitos de QoS, como aplicações de "e-learning" e Laboratório Virtual, além de aplicações multimédia remotas, como Voz sobre IP (VoIP), ou "video a pedido" (Vídeo on demand - VoD).

Usando estas aplicações, os clientes podem estabelecer comunicações entre si, ou para o exterior da rede de acesso, onde se localizam os servidores e outros clientes fixos.

Além destas aplicações destinadas aos utilizadores finais, também são utilizadas aplicações avançadas de teste administrativo da rede, de forma a efectuar medições rigorosas da rede.

Rede de Suporte

Em relação aos encaminhadores que constituem a rede de suporte, estes já não têm a mesma variedade que caracteriza os clientes, sendo todos PCs Desktops fixos com o Sistema Operativo Linux, com extensões apropriadas para os requisitos concretos da rede.

Os encaminhadores estão dispostos numa árvore lógica em que a GW está localizada na raiz da árvore e os dois APs nas folhas, sendo todos ligados entre si por ligações Ethernet de 10 Mbit/s dedicadas.

Os dois APs têm interfaces *wireless* 802.11, as quais estão a operar no modo estruturado como Pontos de Acesso 802.11, de forma a que os terminais se possam associar explicitamente a estes. Para esta rede *wireless* é definido um *nome*[5] da rede, que a identifica e separa de outras redes 802.11 que possam existir, sendo assim comum aos APs e aos clientes desta rede.

Note-se que esta pequena estrutura é suficiente para testar as funcionalidades da rede e das aplicações, mas poderá crescer a qualquer altura, acrescentando mais APs ou nós na árvore

[4] No entanto, para teste de compatibilidade, a rede também foi testada com outros sistemas operativos, e outras interfaces *wireless* de outros fabricantes.

[5] parâmetro ESSID do 802.11

lógica existente. Os dois APs estão separados geograficamente o suficiente, para tornar possível a transição controlada dos terminais móveis entre os Pontos de Acesso, e, com vista a maximizar o débito e evitar as interferências, cada AP vai estar separado nas frequências, ocupando assim uma célula inteira, com um débito máximo de 11Mbit/s.

Ao contrário dos outros encaminhadores existentes na ilha do INESC com encaminhamento estático, nestes o encaminhamento é efectuado de uma forma totalmente automática pelo mecanismo TIMIP de suporte de mobilidade dos terminais móveis da rede, o qual apenas necessita saber a posição do encaminhador na árvore lógica da rede, de forma a automaticamente criar, manter e reconfigurar todo o encaminhamento necessário para suportar os terminais móveis no interior da rede durante os seus movimentos.

Por se usar um mecanismo de mobilidade ao nível IP, então a mudança dos terminais entre os APs incorre, além da latência inerente do 802.11, numa latência adicional necessária para a execução completa das alterações de encaminhamento para cada terminal particular; mas no entanto, este mecanismo foi desenhado explicitamente para assegurar transições de nível 3 muito rápidas, o que torna este peso adicional completamente despercebido, especialmente quando comparado com a latência inerente do nível 2.

Note-se que este encaminhamento de suporte de mobilidade só está presente no interior da rede, sendo transparente para fora desta, o que significa que a rede do exterior é apenas uma subrede IP, acessível pela sua GW, e lhe permite ser integrada no encaminhamento estático CIDR da rede de core da ilha do INESC sem qualquer dificuldade.

Por fim, a GW desta rede de acesso também inclui o suporte dos protocolos sMIP e MIP, que são utilizados para garantir o suporte de macro-mobilidade dos clientes da rede, de tal forma que estes podem transitar para outras redes com suporte MIP, como a segunda rede de acesso *wireless* da ilha do INESC.

Suporte de QoS

Por fim, para o suporte de QoS, a rede de suporte vai constituir um domínio Diffserv limitado, independente das outras redes da ilha do INESC. Cada encaminhador tem os mecanismos de suporte de QoS presentes em todas as interfaces de rede, sendo estas classificadas em interfaces *edge* (Fronteira) ou *core* (Núcleo).

As primeiras (edge) são aquelas por onde os pacotes de dados entram na rede de suporte, sendo aí classificados pela primeira e única vez neste domínio DiffServ; as segundas (core)

são as interfaces que trocam entre si pacotes de dados já marcados na classe agregada correcta, evitando a repetição da classificação já efectuada na interface *edge*.

Devido à topologia da rede em árvore, isto significa que são do tipo *edge* as interfaces *wireless* 802.11 dos APs, por estarem em contacto com os terminais móveis, e a interface Ethernet de ligação à rede de core, e todas as restantes interfaces que fazem as ligações dedicadas entre os encaminhadores são do tipo core (ver Figura 5).

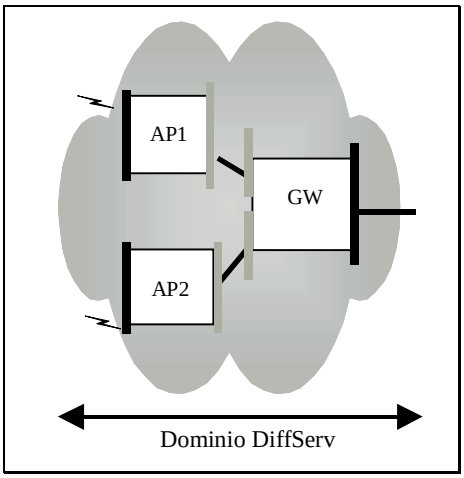


Figura 5: Rede de suporte wired como um domínio DiffServ

Assim, quando um pacote entra na rede de suporte, este é inicialmente classificado e policiado no primeiro encaminhador, sendo-lhe atribuída uma classe de serviço agregada, que é marcada no próprio pacote para futura referência[6]. Se o pacote for aceite pela decisão de policiamento, então vai ser colocado na interface de saída do encaminhador que o encaminhamento TIMIP indicar, para sofrer um tratamento de saída que depende da sua classe de serviço, e das condições actuais desta e das outras classes existentes (por exemplo, se o pacote pertencer a uma classe que excedeu o seu ritmo de saída, então o pacote será atrasado propositadamente até poder ser enviado). Estas operações de saída vão-se repetir ao longo dos encaminhadores seguintes, até que o pacote saia da rede de suporte.

No entanto, note-se que este mecanismo de QoS não garante o pretendido suporte de QoS extremo-a-extremo (embora esteja próximo); Concretamente, quando os pacotes chegam vindos da rede de core no sentido do *wireless* (*Downlink*), então vai existir QoS ao longo de toda a rede, porque o pacote é sempre emitido por interfaces pertencentes aos encaminhadores da rede, até que chegam à interface *wireless* e são processados com QoS.

Por outro lado, no sentido oposto (*UpLink*), só vai existir QoS *desde* o primeiro encaminhador do caminho - o AP, devendo-se isto, porque na emissão do pacote, o terminal vai usar o mecanismo distribuído DCF de acesso ao meio, que dá a todas as estações uma igual oportunidade de transmissão, não distinguindo assim as prioridades relativas dos vários pacotes dos terminais entre si, significando isto que os pacotes prioritários *uplink* podem ser

[6] concretamente no campo DSCP do cabeçalho IP

perdidos no *wireless*, por concorrerem em igualdade de circunstâncias com os pacotes não prioritários. No entanto, assim que o pacote chega ao AP, então passa automaticamente a ter QoS, qualquer que seja o seu destino[7].

Para resolver completamente este problema, seria necessária a utilização de um mecanismo de suporte de QoS de nível 2 específico do 802.11, comum a todas as estações e APs desta rede, que controlasse o acesso ao nível físico pelas estações. Um exemplo de um mecanismo com estas características seria o modo PCF do 802.11, embora sem suporte no hardware utilizado.

2.1.4 Rede secundária de acesso wireless 802.11

Além da rede de acesso *wireless* principal, a ilha do MOICANE conta com uma segunda rede *wireless* 802.11 de apoio, que com conjunção com a rede anterior, será utilizada para demonstrar a tecnologia desenvolvida de macro-mobilidade. Para tal, a segunda rede, representada na Figura 6, é constituída por um único AP, que acumula as funções de GW, AP, e Agente sMIP/MIP, sendo este o caso mínimo de um domínio TIMIP.

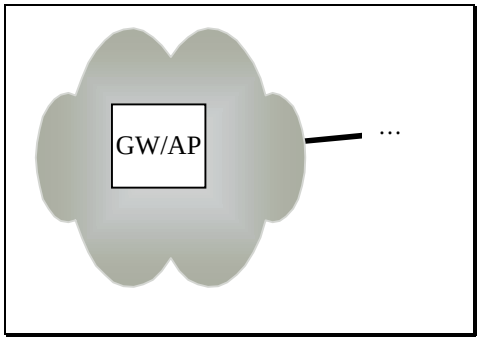


Figura 6: 2ª Rede wireless de demonstração do inesc

Relativamente ao AP da rede, a sua interface *wireless* está configurada de forma a que os terminais legados possam transitar com facilidade entre os dois domínios. Para isto, o AP vai estar configurado no modo estruturado do 802.11, e com um endereço de rede e frequência distintas da rede principal, o que separa totalmente as duas redes.

Esta configuração simples permite que os terminais transitem *controladamente* entre os dois domínios de uma forma simples, apenas escolhendo a rede de nível 2 a que se vão associar, que os protocolos de micro e macro mobilidade vão agir de forma automática para criar a mobilidade necessária para suportar os terminais.

[7] Esta característica aplica-se mesmo às comunicações entre estações associadas ao mesmo AP, dado que o TIMIP obriga a que todo o tráfego passe pelo AP dos terminais.

2.2 Arquitectura dos elementos de rede

Nesta secção analisa-se a arquitectura genérica dos elementos da rede de suporte, focando as características particulares que estes encaminhadores têm e que os distinguem dos encaminhadores genéricos IP, sendo descritos os módulos existentes nestes elementos, as interacções entre eles, e a vida do pacote desde a entrada até à saída do elemento de rede.

2.2.1 Módulos

Tal como já foi referido anteriormente, os elementos de rede são baseados num modelo base que detém as funções comuns a todos estes encaminhadores, e que pode ser especializado com funções e módulos adicionais para realizar as funções de APs e de GW da rede.

Para tal, a arquitectura genérica dos nós da rede está presente na próxima figura:

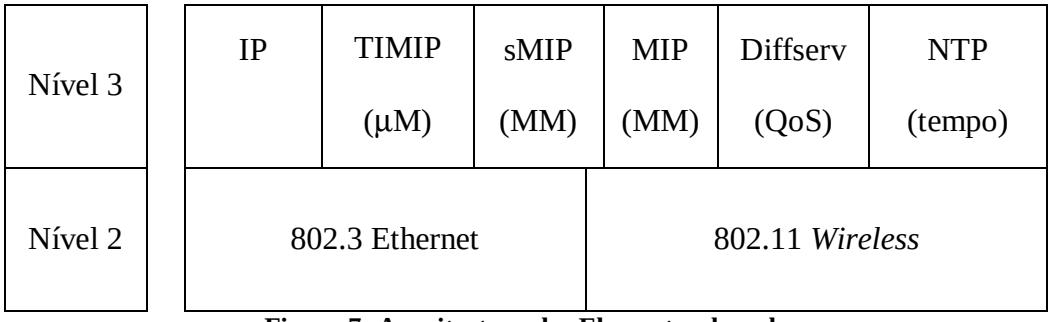


Figura 7: Arquitectura dos Elementos de rede

Os elementos de rede têm a maior parte das funcionalidades ao nível IP (nível 3), com funções que são independentes das várias tecnologias.

Neste nível, o módulo IP vai conter todas as funcionalidades dos encaminhadores genéricos que implementam o *stack* de protocolos TCP/IP versão 4, estando presentes neste módulo, para além de outras, funções como o *Forwarding*, resolução de endereços ARP, gestão de erros por ICMP, encapsulamento de pacotes em túneis IPIP, etc.

Estes mecanismos genéricos vão ser reaproveitados pelos novos módulos necessários nesta rede de acesso, sendo estes configurados dinamicamente pelos novos módulos.

Módulos de encaminhamento:

Associado ao *forwarding* está o módulo de encaminhamento, que em encaminhadores muito simples poderá ser estático, mas usualmente dinâmico para os encaminhadores mais avançados. Neste caso, o módulo de encaminhamento vai automaticamente aprender as localizações dos terminais móveis na rede de acesso, configurando a tabela de encaminhamento do sistema.

Quando existirem pacotes para encaminhar, o módulo de *forwarding* vai consultar esta tabela do sistema para encontrar o próximo nó e a interface de saída dos pacotes, e deste modo os módulos de encaminhamento vão configurar dinamicamente o módulo genérico de *forwarding*.

No caso desta rede de acesso *wireless*, todos os elementos de rede vão deter um módulo de encaminhamento dinâmico (TIMIP), responsável pelo suporte da micro-mobilidade dos terminais móveis no **interior** da rede, de forma a que tenha a noção da localização dos terminais móveis, o que é essencial para lhe dar a conectividade.

O conjunto dos encaminhadores da rede vão estar organizados numa árvore lógica, em que cada encaminhador tem um único nó ascendente, e o encaminhador localizado na raiz é denominado de GW e tem a ligação à rede de core.

Além da comunicação pela tabela de encaminhamento, o módulo TIMIP também vai comunicar com os seus correspondentes dos encaminhadores adjacentes por via de pacotes de controlo exclusivos deste protocolo. A última característica deste módulo é que deverá receber uma cópia do cabeçalho IP de todos os pacotes de dados que são recebidos pelas interfaces do encaminhador para inspecção.

Além do TIMIP, a GW da rede vai ter um segundo módulo de encaminhamento sMIP (*surrogate* MIP), complementar ao TIMIP para o suporte da macro-mobilidade dos terminais móveis, tornando a GW num *surrogate agent* sMIP com a capacidade de localizar e suportar a conectividade dos terminais móveis entre as redes, comunicando com o HA localizado na rede de origem do terminal legado como se fosse este.

Este módulo de encaminhamento tem interacções com TIMIP, para este último lhe induzir a fase de detecção dos terminais legados ao nível da macro-mobilidade, sendo estas criadas por via de uma sinalização interna definida entre os dois.

Note-se que estes dois tipos de encaminhamento são completamente independentes um do outro; nomeadamente, o TIMIP apenas encaminha os pacotes para terminais móveis que estejam no interior da rede de acesso, tendo o sMIP a responsabilidade complementar de encaminhar pacotes para terminais móveis entre as redes.

Além do sMIP, a GW tem também um módulo clássico de suporte do MIP na sua componente de servidor (papel de HA e FA), sendo a sua execução independente dos anteriores TIMIP e sMIP.

Módulos de QoS:

O módulo de DiffServ vai ser utilizado para adicionar à rede suporte de QoS, sendo os fluxos de dados agregados em classes de serviço e tratados com serviços diferenciados de acordo com a sua classe durante toda a rede.

Este módulo vai estar presente em todos os encaminhadores da rede de acesso e é ortogonal ao módulo do encaminhamento, detendo das funcionalidades de configuração, tratamento dos pacotes à entrada e tratamento dos pacotes à saída. A primeira parte vai interagir com o módulo IP para criar e alterar diferentes configurações estáticas de elementos de controlo de tráfego nas interfaces físicas do encaminhador, por forma a criar uma arquitectura adequada para executar as acções definidas na arquitectura DiffServ.

Quando a componente de configuração criar ou alterar uma dada configuração estática, cada interface do encaminhador vai deter de ambas as componentes genéricas DiffServ de Entrada e Saída, constituídas por elementos genéricos independentes das tecnologias, pertencentes ao módulo IP. Em cada interface a componente de Diffserv de entrada vai efectuar acções DiffServ assim que o pacote chega ao nível 3 do encaminhador.

Este módulo vai ser diferente quer se trate de uma interface fronteira ou core, uma vez que estas últimas **não** executam as acções de classificação e policiamento que são exclusivas das do tipo fronteira.

Além do Diffserv de entrada, cada interface do encaminhador também vai detém da componente de Diffserv de saída, com as funções de execução dos tratamentos agregados das varias classes (“*per hop behaviour”*) definidas nesta rede, por via de escalonadores de prioridades/“*weighted round robin”*.

A última acção que os pacotes sofrem antes de saírem do nível 3 do encaminhador é o de serem marcados com a sua classe de serviço agregado no campo apropriado do cabeçalho IP (campo DSCP).

Módulo de Tempo:

O último módulo que está ao nível IP presente em todos os encaminhadores, será o módulo de sincronização de tempo, que tem o objectivo de garantir que os relógios dos nós da rede de suporte estão o mais sincronizados possível de acordo com o relógio de referência da rede (relógio da GW), dado que este é um requisito do protocolo de encaminhamento TIMIP, e necessário para as medições a executar na rede.

Para isto, ter-se-á que usar um qualquer mecanismo que faça este acerto distribuído do relógio, como por exemplo o protocolo NTP (Network Time Protocol). Quando é usado este

protocolo, a GW vai ter um servidor NTP para responder com o valor do seu relógio actual aos clientes NTP, que estão presentes nos restantes nós da rede. Este protocolo, específico para esta função de acerto dos relógios, atinge margens de erro na sincronização bastantes baixas porque executa automaticamente acções adicionais de correcção e ajuste relativos à própria acção de propagação e tratamento das mensagens NTP.

Módulos do Nível 2:

As restantes componentes dos elementos da rede já são dependentes das tecnologias de rede, fazendo parte do nível 2 do encaminhador ("*link layer*"). Estas vão estar intimamente associadas às interfaces físicas e aos detalhes de cada tecnologia específica, existindo tantos componentes quanto as interfaces físicas presentes no encaminhador.

As interfaces físicas *wireless* 802.11 vão permitir que os terminais móveis se liguem à rede e se movimentem no interior desta, podendo qualquer elemento da rede de suporte ter uma interface deste tipo, o que torna num AP (Access Point, ponto de acesso da rede. Para tal, as interfaces 802.11 vão funcionar no modo estruturado do 802.11, de forma a ocuparem uma célula inteira, e participarem na rede 802.11 comum com o mesmo nome de rede que os outros APs.

A principal acção desta interface será o de receber e enviar pacotes de dados de/para os terminais móveis, pelo que, todos os pacotes emitidos pelos terminais que sejam recolhidos por esta interface vão ser entregues ao nível acima IP para encaminhamento (e também uma cópia do cabeçalho para o TIMIP).

O outro tipo de interface que os encaminhadores possuem é respeitante às interfaces Ethernet, que vão receber e enviar pacotes de dados de forma semelhante às interfaces *wireless*. Estas interfaces são utilizadas para ligar os encaminhadores entre si, e adicionalmente para ligar a rede de acesso à rede de core, no encaminhador na raiz da árvore.

Nesta situação, o encaminhador que tiver a ligação à rede de core será a GW desta rede, e o seu endereço IP desta interface o endereço identificador desta rede como um todo para os protocolos de encaminhamento).

2.2.2 Vida do pacote

No interior da rede de acesso vão existir apenas 2 tipos de fluxos diferentes de pacotes IP: fluxos de controle e dados IP.

Os primeiros são utilizados pelos componentes dos encaminhadores nas suas comunicações internas, nomeadamente a transferência de informações de mobilidade, configuração remota de QoS e sincronização de tempo, os quais são recebidos directamente pelo módulo correspondente no interior dos encaminhadores.

Todos os restantes fluxos vão ser considerados como fluxos de dados, com um tratamento genérico simples no interior do encaminhador, ilustrado na Figura 8, que descreve o processamento dos pacotes de dados no interior do núcleo do encaminhador, tanto no nível 2 como no 3. Para referência, a mesma figura com mais detalhe, incluindo os *deamons* de controle do encaminhadores da rede, está presente no Anexo 5.

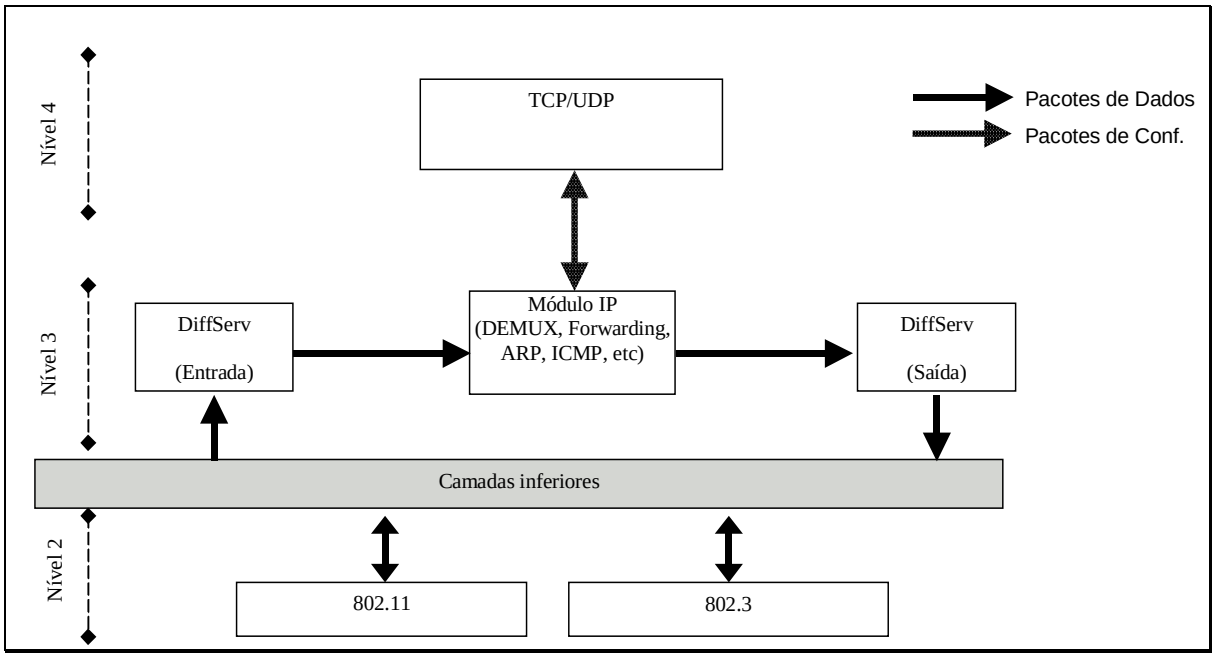


Figura 8: Vida dos pacotes de dados

Assim, quando um pacote de dados chega ao encaminhador por uma das interfaces deste, o pacote vai ser sempre entregue pelo nível 2 ao IP para processamento, e nesta operação, o pacote é entregue à componente de DiffServ de entrada da interface por onde chegou, e uma cópia do cabeçalho IP é entregue ao módulo TIMIP para análise.

Uma vez no Diffserv de entrada, o pacote vai sofrer diferentes tratamentos se esta interface for do tipo fronteira, ou core. No primeiro caso, o pacote vai ser classificado e policiado pela única vez nesta rede, sendo-lhe atribuído uma classe de serviço consoante as suas características (dependentes da configuração estática actual que a configuração Diffserv instanciou). No outro caso, o pacote já está previamente aceite e marcado numa classe por um encaminhador anterior, pelo que a sua classe é extraída do cabeçalho IP.

Seguidamente o pacote é entregue ao módulo genérico IP, que é igual aos encaminhadores normais IP. Neste módulo, a operação de *forwarding* vai considerar o destino do pacote, e escolher o próximo nó por onde este pacote de dados vai ser encaminhado, consultando para tal a tabela de encaminhamento do sistema, que os protocolos de encaminhamento vão manter consistente com o estado actual da rede de acordo com as localizações presentes dos terminais móveis.

O pacote é então entregue à interface de saída, sendo entregues transparentemente à componente Diffserv de Saída, onde o pacote vai ser considerado juntamente com os restantes da mesma classe, formando em conjunto um *Agregado*.

Nesta componente, os agregados são processados de acordo com as suas importâncias relativas e ocupação dos recursos de saída, por via de escalonadores que serializam os pacotes por várias métricas, que estão associados a filas de espera com diferentes limites ou tratamentos, como acções de perda antecipada (*early drop)*.

A última acção deste componente será o de marcar o pacote com a informação da sua classe, num campo específico do cabeçalho IP (DSCP) (com efeito apenas nos encaminhadores Fronteira da rede, já que nos *core* a marcação mantém-se inalterada).

Por fim, quando o pacote sair desta última componente, então é entregue à interface de saída do encaminhador por onde será transmitido para o próximo nó, ou para o seu destino final.

3. Implementação

Este Capítulo vai descrever em detalhe a implementação desta rede, verificando-se as entidades, protocolos e mecanismos que foram desenvolvidos, e as suas relações entre si. Esta implementação consistiu tanto na instanciação da tecnologia, como também no trabalho de engenharia necessário para a integração com as outras componentes da ilha do MOICANE.

Para tal, serão analisados os detalhes dos módulos existentes no encaminhador, com especial atenção aos que foram criados propositadamente para esta rede de acesso – de mobilidade e QoS.

3.1 Kernel Linux

Este módulo de *software* é a peça base onde todos os restantes módulos vão encaixar, estando bem documentado, como em [27], [28], [29] e [36]. Entre outros, o *kernel* inclui na sua componente de rede os protocolos base, *drivers*, controlo de tráfego, além de todas as funcionalidades necessárias para a utilização do hardware, como a gestão dos processos, memória, dispositivos, etc.

Como está descrito em grande detalhe em [68], nesta rede de acesso os nós usaram o *kernel* 2.4.18 (a versão estável mais recente, à data do início da implementação deste trabalho), com alterações específicas (*patches*) a diversas componentes, e com adições ao suporte de hardware utilizado, como as interfaces PCMCIA. Além disto, o *kernel* também foi configurado de uma forma especial, nomeadamente pela alteração do grão do sistema para um valor mais rápido, e pela medição do tempo de uma forma mais precisa.

3.2 Módulo IP

O módulo IP vai genericamente agrupar as funções genéricas dos encaminhadores Linux que implementam a base do protocolo IP clássico (nível 3), contendo as funções de *demultiplexing* (DEMUX), *forwarding*, resolução de endereços por ARP, gestão de erros (ICMP), etc. Este modulo, residente no Kernel do linux, e as suas utilidades associadas, estão bem documentados nas referências [28] a [39], cobrindo estas referencias tanto a configuração, utilização, programação e alteração das componentes de rede.

No caso dos encaminhadores da rede de acesso, este módulo vai ser configurado dinamicamente pelos módulos de Encaminhamento e Diffserv, de forma a que os pacotes de dados são sempre processados por mecanismos exclusivamente residentes no núcleo do sistema, o que aumenta substancialmente o desempenho (comparando com a sua manipulação por programas do espaço do utilizador).

Um aspecto chave existente no IP é a sua configuração pela tabela de encaminhamento por parte dos módulos de encaminhamento, que será utilizada pelo módulo de *forwarding* com as vantagens referidas no parágrafo anterior. No entanto, note-se que esta tem uma latência mínima de 2 segundos, por omissão, relativamente à efectivação das alterações, mas dado que esta latência está no caminho crítico do processo de *handover* do TIMIP, a tabela de encaminhamento foi alterada por forma a retirar esta limitação.

3.3 Driver 802.3

Este módulo consiste num *driver* de *software* de acesso às interfaces físicas Ethernet presentes nos elementos da rede. Relativamente ao hardware, esta carta *ethernet* é descrita em [68], sendo utilizado um *driver* já existente no *kernel linux*, que não necessitou de qualquer alteração. Este acrescenta uma interface ao sistema denominada de **eth**x, sendo automaticamente utilizada pelo módulo IP, e configurada pelo suporte de mobilidade e Diffserv (pelos comandos **ifconfig** e **tc**).

3.4 Driver 802.11

Este módulo também consiste num *driver* de *software* para utilização do hardware 802.11 que está presente nos APs da rede, descritos em [68]. Estas cartas têm uma interface PCMCIA, e utilizam o *chipset* prism2 da Intersil [55], o qual tem um modo especial, denominado de *hostAP* [57], no qual as funções básicas de estação são complementadas por *software*, de forma a realizar um AP completo compatível com qualquer cliente 802.11, tendo este *driver* o suporte deste modo especial.

Esta solução provou ser a mais flexível para os objectivos pretendidos neste trabalho, por permitir a alteração de grande parte das características do ponto de acesso 802.11, dado serem implementadas em *software*[8].

No início da implementação da rede de acesso, este *driver* estava ainda num estado inicial com suporte ao modo especial desejado, mas ainda numa fase inicial da sua depuração. Por esta razão, ao longo da duração deste trabalho, o módulo sofreu diversas alterações, sendo algumas genéricas, e outras específicas para esta rede de acesso.

Relativamente às primeiras, no contexto da rede foram descobertos e corrigidos diversos *bugs*, *race conditions*, ineficiências, etc., bem como foi adicionado suporte para mais funcionalidades, como certas *wireless extensions* e acções de gestão. Por serem genéricas, estas alterações foram propagadas para o *driver* original, para benefício da comunidade de utilizadores deste *software* [58].

Por outro lado, houve alterações específicas para esta rede de acesso, de que se destacam:

a) Adição da notificação assíncrona das acções de gestão do 802.11 para o TIMIP, utilizando um *socket netlink* de comunicação entre o núcleo e os *deamons* que correm no espaço do utilizador. Esta funcionalidade vai instanciar no 802.11 a detecção reactiva do protocolo TIMIP de uma forma extremamente eficiente.

b) Alteração da política de *queueing* interna do hardware, de forma a limitar o número máximo de pacotes de dados que podem estar pendentes para transmissão (à saída da interface *wireless*), para o valor de 10 pacotes (abaixo deste valor o desempenho da carta sofria, deixando se ser possível transmitir o máximo da tecnologia).

   Esta alteração deveu-se ao facto de o hardware ter memória interna (*buffer*) para transmissão de pacotes, utilizada para reduzir a ocupação do processador na acção de transmissão de pacotes em rajada. No entanto, esta optimização cria um atraso substancial nos pacotes prioritários quando a saída estiver saturada, dado que nesta fila não existe distinção das classes de pacotes (i.e., esta fila afasta demasiadamente a qualidade de serviço do nível 3 da transmissão).

[8] Esta solução foi comparada com outras possibilidades, nomeadamente pela utilização de *firmware* adicional para a funcionalidade de AP (*firmware* terciário [56]), mas que não permitia customizações. No entanto, foi a opção seguida por um trabalho paralelo a este, descrito em [60], utilizando outro driver [59].

De uma forma semelhante à da interface Ethernet, o *driver* vai criar uma interface ao sistema denominada de **wlan**x, sendo esta utilizada pelo módulo IP, e configurada tanto pelo suporte de mobilidade com pelo Diffserv (pelos comandos iwconfig, ifconfig e tc).

3.5 Módulo TIMIP

Para a implementação do TIMIP ([65] e [69]), cada nó da rede vai ter um *deamon* que instancia este protocolo. O processo completo da sua instalação, configuração e execução está detalhado em [68], sendo aqui apenas descrita a informação que não está nesta referência, nomeadamente a relativa à arquitectura interna, interacções e algoritmos/máquinas de estado.

Dada a sua natureza, este módulo foi implementado no espaço do utilizador, não necessitando de estar no núcleo para garantir um desempenho máximo[9]. Isto acontece porque este apenas vai, em certas ocasiões periódicas, interagir com o módulo de *forwarding* do IP através da configuração da tabela de encaminhamento do sistema.

Neste sentido, a componente crítica do *forwarding* dos pacotes de dados continua a ser efectuada no interior do núcleo, sendo apenas o seu controlo no exterior, o que tanto garante o seu desempenho, como mantém a facilidade de implementação do módulo. Assim, o controlo TIMIP é instanciado em "user space", sendo os dados IP em "kernel space".

Estas interacções com as componentes do IP são efectuadas principalmente por via de comandos externos do sistema, exceptuando-se as alterações à tabela de encaminhamento, que são efectuados de forma directa por *netlink*[10], dados que estão no caminho crítico do processo de *handover* do TIMIP. Por estas razões, o *deamon* terá que ser executado com os privilégios totais do sistema (utilizador *root*).

3.5.1 Arquitectura do módulo TIMIP

A figura seguinte – Figura 9 – vai descrever a estrutura interna do *deamon* TIMIP, que contém grupos de acções, estruturas de dados, e interfaces.

[9] Embora seja lançado com prioridade sobre os restantes programas e *deamons que se executam no sistema,* por via do comando *nice do Unix*.

[10] Sendo esta uma opção (na compilação do *deamon*), podendo-se utilizar comandos externos em todas as situações.

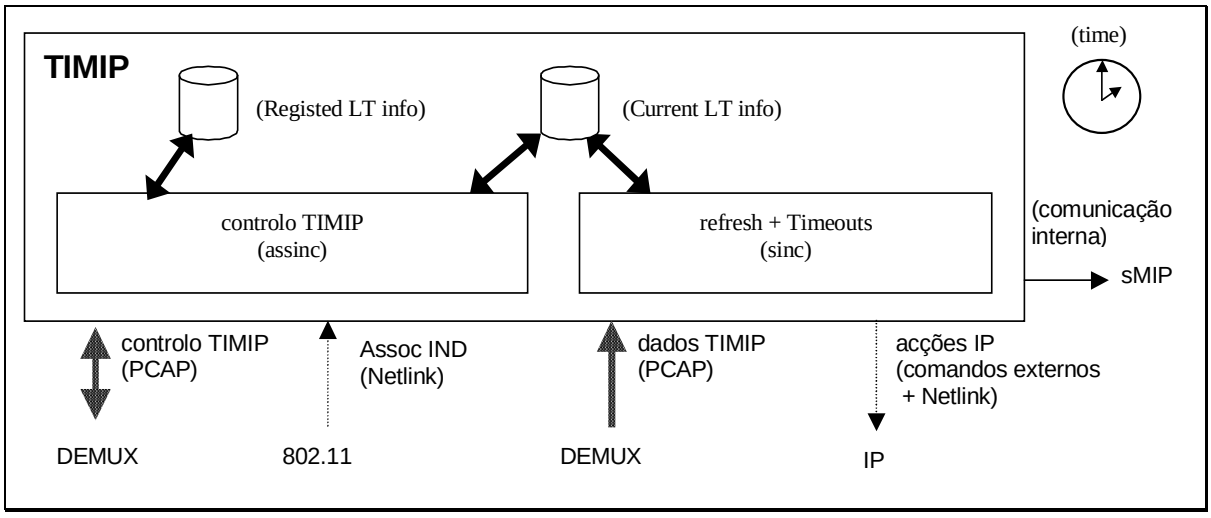


Figura 9: Interfaces do módulo TIMIP

3.5.1.1 Acções

O módulo TIMIP tem dois tipos de componentes de acções básicas, que vão estar relacionadas com as acções assíncronas e síncronas do *deamon*:

As primeiras estão relacionadas com o controlo TIMIP de suporte à criação e modificação da mobilidade dos terminais legados, tendo requisitos apertados de responsividade, pois entram no caminho crítico dos *handovers* TIMIP, devendo estes ser tão rápidos quanto possível para reduzir ao mínimo os períodos sem conectividade da responsabilidade da micro-mobilidade IP.

Entre estas, estão as acções despoletadas pela detecção dos terminais e pela recepção de mensagens de controlo TIMIP, utilizadas para a comunicação entre os nós da rede. Para tal, estas acções assíncronas têm o objectivo último de configurar a tabela de encaminhamento do nó, de acordo com o estado dos terminais móveis na rede.

O segundo módulo vai tratar dos algoritmos TIMIP que não têm este requisito de rapidez, nomeadamente a manutenção do estado dos terminais, envolvendo a recepção dos pacotes de dados IP, e as acções de *timeout* utilizadas para forçar os terminais a gerarem provas de vida, ou para cancelar os seus registos.

Normalmente, esta distinção entre acções prioritárias e não prioritárias leva à utilização de mecanismos de paralelismo de *software* com suporte de preempção, o que obrigaria à existência de sincronização interna, por via de mecanismos como semáforos.

Contudo, optou-se por desenhar este modelo com base num modelo totalmente síncrono, sem paralelismo, mas com um desempenho muito semelhante relativamente à responsividade das acções de mobilidade.

Este modelo alternativo tem a vantagem de não necessitar de sincronização, por ter apenas um único fio de execução, e de possibilitar o processamento das acções do protocolo sem requisitos tempo-real periodicamente em modo "batch", o que aumenta o desempenho do módulo ao reduzir o seu peso no sistema, e, em última instância, permitindo o suporte de mais clientes ao protocolo.

O modelo está ilustrado em forma de máquina de estados na Figura 10, e consiste no tratamento periódico das acções síncronas, em modo "batch", e apenas uma vez em cada segundo (valor configurável). Durante esta operação, no final de cada acção síncrona, é sempre verificada a existência de acções assíncronas, que se existirem vão interromper o tratamento síncrono (por serem mais prioritárias).

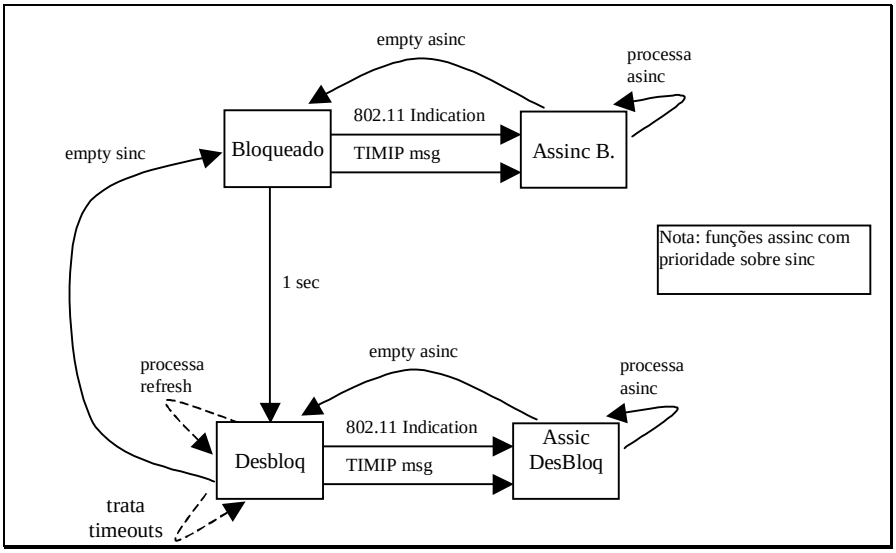


Figura 10: Máquina de estados para eventos assíncronos e síncronos

3.5.1.2 Estruturas de dados

As estruturas de dados vão conter as informações relativas aos terminais do ponto de vista deste nó da rede.

A primeira estrutura, BD_REGISTED, contém as informações dos terminais móveis que estão registados a operarem na rede, incluindo o seu endereço IP e MAC e outros dados. Para os nós da rede normais, esta tabela é usada somente em leitura, podendo o seu conteúdo ser alterado por ordem da GW por processos de gestão.

A segunda estrutura, BD_CURRENT, vai conter a relação dos terminais que estão activos localizados por descendentes deste nó. Esta informação é alterada de acordo com os algoritmos de registo TIMIP, descritos em [69], e podem incluir a detecção dos terminais nos APs, a chegada de mensagens TIMIP dos nós adjacentes, ou pelos *timeouts*.

Note-se que a tabela de encaminhamento do sistema é sempre alterada de acordo com as alterações desta tabela interna, por forma a instanciá-las no encaminhamento dos pacotes IP.

Por último, é também de referir que esta estrutura é concretizada por uma árvore binária, por forma a optimizar a pesquisa de informação que será indexada pelo endereço IP dos terminais.

3.5.1.3 Interfaces

O módulo vai comunicar com o exterior por via de interfaces bem definidas, do tipo uni ou bidireccionais, e que envolvem pacotes de informação ou apenas recursos internos do Linux, como a troca de mensagens.

Relativamente às interfaces que trocam pacotes, estas são efectuadas directamente de/para o módulo DEMUX do *kernel*, por onde todos os pacotes passam no nível 3, e que são entregues ao módulo interno do Linux para processamento (nomeadamente o TCP/UDP/ICMP/ FORWARDING). Para utilizar este tipo de interfaces, o *deamon* vai na sua inicialização registar-se directamente no DEMUX estabelecendo interfaces PCAP (Packet capture) [40].

Esta interface PCAP é uma API genérica multi-plataforma utilizada para a recepção e envio de pacotes directamente de/para as interfaces físicas do encaminhador, com uma utilização integrada nos *sockets* BSD. Esta API é largamente utilizada nas aplicações de inspecção/gestão/análise da rede, como o Ethereal e o Tcpdump.

Para isto, por cada interface física do encaminhador, vão existir 2 interfaces PCAP distintas para controlo e dados, onde se indica *que* pacotes desta interface o módulo pretende receber, por via de uma "string" com uma sintaxe e semântica simples. Em Linux, o PCAP tem uma implementação muito poderosa, dado que a componente de desmultiplexagem dos pacotes à entrada no IP é pré-compilada dinamicamente no DEMUX, efectuando-se a escolha em "kernel mode" de uma maneira muito eficiente. Os detalhes destas *strings* de captura utilizadas no *deamon* estão detalhadas no Anexo 2.

Esta separação entre controlo e dados, permite a divisão anterior entre acções síncronas e assíncronas, o que leva a uma grande responsividade do módulo em relação às operações críticas do *handover*.

Assim, o PCAP do controlo vai tratar exclusivamente dos pacotes de controlo TIMIP encapsulados em ICMP, e o PCAP dos dados vai receber apenas o cabeçalho IP dos dados encaminhados pelo encaminhador, para a inspecção do endereço de origem (apenas nos instantes síncronos bem determinados, como já foi referenciado).

Por outro lado, o *deamon* também vai, em situações bem definidas, gerar pacotes do tipo controlo por via de um dos PCAP da interface de saída, o que permite assemblar o pacote desejado especificando *todos* os campos de *todos* os níveis, sendo assim substancialmente mais poderoso que *sockets* clássicos INET. Entre estes, o *deamon* vai gerar pacotes TIMIP encapsulados em ICMP para as comunicações inter-nó das acções de update e acknowledge, pacotes ICMP *echo request* para a manutenção do estado dos terminais, e pacotes *gratuitous* ARP para a alteração em cada handover das caches ARP dos terminais com o novo valor da GW.

Relativamente à comunicação com as outras componentes do sistema, todas as acções de configuração das funções IP são efectuadas pela utilização dos comandos específicos externos, que têm a grande vantagem de se manterem válidos mesmo que o *kernel* linux sofra alterações internas. Entre estes, encontram-se o comando **route** (manipulação da tabela de encaminhamento), **arp** (manipulação da tabela de arp/proxy arp), **ip** (manipulação de túneis IPIP), **ifconfig** (gestão IP das interfaces físicas), **iwconfig** (gestão L2 das interfaces *wireless*).

A única excepção a este comportamento é nas alterações da tabela de encaminhamento do sistema, que terão de ser efectuadas de uma forma extremamente eficiente por estar no caminho crítico do *handover* dos terminais. Neste caso específico, a tabela é alterada usando uma interface *netlink* directa ao módulo FORWARD do *kernel* (que detém esta tabela).

A última interface existente é referente à detecção reactiva do TIMIP, originada no *driver* 802.11. Para isto, é também usada uma interface *netlink*, unidireccional, desde o *driver* directamente para o *deamon*.

Conceptualmente, também existiria uma interface entre o módulo e o módulo sMIP, necessária para a detecção dos movimentos dos terminais, mas no entanto, tal como vai ser descrito no módulo sMIP, tal interface é meramente interna, uma vez que o *deamon* inclui o suporte para os dois protocolos em simultâneo (mas que são todavia independentes entre si).

Por fim, é da responsabilidade do TIMIP o suporte do MIP, tal como está definido em [69], pelo que a GW TIMIP vai gerar pacotes de *beacons* MIP, que vão ser propagados pela rede por via dos nós TIMIP, que vão agir como um *relay agent* destes *beacons* MIP, passando-os desde os terminais até à GW e vice-versa.

3.5.2 Implementação dos algoritmos avançados TIMIP

Esta secção vai detalhar a implementação dos algoritmos mais importantes que foram definidos teoricamente no TIMIP (na referencia [69]).

3.5.2.1 Detecção da chegada dos terminais

Quando a primitiva assíncrona de chegada dos terminais é despoletada pelo mecanismo netlink, o *deamon* analisar imediatamente a informação que esta contém, que será o endereço MAC do terminal detectado.

Quando isto acontece, o *deamon* vai consultar a informação de registo (BD_REGISTED), para verificar se este terminal está registado na rede. Em caso afirmativo, então o seu endereço IP é extraído desta base de dados, sendo usado exclusivamente em todas as operações subsequentes.

Assim, quando recebe a informação da detecção, o AP vai:

a) Verificar o seu próprio relógio, que terá que estar sincronizado com os dos outros APs (por NTP, ou outro protocolo de sincronização de tempo), o valor temporal desta movimentação do terminal na rede, alterando (BD_CURRENT) com a informação que o terminal se encontra localizado na sua interface *wireless* e com o tempo da sua chegada.

b) Se anteriormente o terminal já estava presente neste AP, então significa que o nó não aprendeu nada de novo, pelo que o processo termina aqui.

c) No caso contrário, então o terminal vai colocar o antigo nó mais próximo (i.e. o nó por onde anteriormente o terminal era acessível) numa lista de nós que ainda não responderam com *acknowledge*, e cria um pacote de *update* relativo a este terminal com: endereço origem/destino apropriados, endereço do terminal móvel, *timestamp* actual. Este pacote de *update* é então enviado ao anterior próximo nó, utilizando a interface PCAP apropriada. (ver [69] relativamente ao formato dos pacotes de controlo TIMIP).

d) De seguida, o nó vai instanciar a alteração que ocorreu, alterando a tabela de encaminhamento do sistema com a informação mais actualizada que aprendeu. Para isto, o *deamon* vai comunicar por *netlink* directamente para o módulo *Forward* do *kernel* Linux, com a alteração desejada.

e) Por fim, o AP vai enviar um *gratuitous* ARP do valor da GW ao terminal (gerando um pacote **ARP reply** enviado por PCAP), e adiciona o seu MAC à tabela de ARP do sistema (pelo comando externo **ARP**).

3.5.2.2 Tratamento do update

O processamento dos *Updates* é a parte chave do funcionamento do *deamon* TIMIP, porque é por estas mensagens que a rede fica a saber a nova localização dos terminais.

Neste processo, quando um nó recebe uma mensagem de *update* vindo de um qualquer nó adjacente a si, então no caso normal este passa a ser o novo próximo nó do terminal, isto é, o nó por onde o terminal se localiza e por onde são encaminhados os pacotes que lhe são destinados (estas acções têm excepções/opções a descrever posteriormente).

Assim, quando recebe o *update*, o nó vai:

a) Comparar o *timestamp* da associação deste terminal com o seu actual (consultando BD_CURRENT), aceitando se for igual ou mais recente que o actual para este terminal.

b) Emitir um pacote de *acknowledge* ao emissor, gerando o pacote de resposta TIMIP, usando a interface correcta de PCAP para emitir o pacote. Note-se que se este pacote de *ack* não tem garantias de recepção; se se perder, o emissor original reenvia o *update* original, e que terá um comportamento idempotente neste nó.

c) Considera o novo *timestamp* do terminal (se for mais recente), alterando o estado do terminal (BD_Current).

d) Se o *update* manteve o próximo nó do terminal, então significa que o nó não aprendeu nada de novo, pelo que o processo termina aqui.[11]

e) No caso contrário, então o terminal vai colocar o antigo nó mais próximo (o nó por onde anteriormente o terminal era acessível) numa lista de nós que ainda não responderam com *acknowledge*, e recria o pacote de *update* com: endereço origem/destino apropriados, endereço do terminal móvel, *timestamp*. Este pacote de *update* é então enviado ao anterior próximo nó, por PCAP. (ver [69] relativamente ao formato dos pacotes de controlo TIMIP)

f) Excepções: tanto o AP anterior como a GW da rede vão terminar a propagação do *update*, quando o caminho anterior não apontar para um nó da árvore.[12]

[11] Embora, do seu ponto de vista, o nó não tenha aprendido nada de novo, o terminal pode ter-se movimentado num nível mais baixo na árvore de nós.

g) De seguida, o nó vai efectuar as acções locais necessárias para instanciar a alteração que ocorreu, bastando-lhe alterar a tabela de encaminhamento do sistema com a informação mais actualizada que aprendeu, comunicando por *netlink* directamente para o módulo *Forward* do *kernel linux*.

3.5.2.3 Resolução das inconsistências do estado na rede

Tal como foi descrito na descrição teórica do TIMIP, existem certas situações de inconsistência da rede que são resolvidas com o recurso ao *timestamp* presente em todas as mensagens de controlo TIMIP, e permite resolver os conflitos.

Neste sentido, quando um nó receber um *update* relativo a uma movimentação mais antiga que a actual (registada em BD_CURRENT), então é sinal que o nó que está a enviar a mensagem não tem a informação mais actualizada do terminal móvel, *embora pense que tem* (porque está a enviar o *update*).

Nestas condições, o nó não vai enviar o *acknowledge* como normalmente, pois isso indicaria a aceitação do *update*, mas sim uma resposta do tipo *update* com a informação mais actualizada do terminal (incluindo o novo *timestamp*), que quando for recebida pelo nó confuso vai obrigar à recorrecção do estado inconsistente da rede.

De uma forma complementar, quando um nó recebe um *acknowledge*, este vai verificar o *timestamp* a que se refere. Se for o esperado, então o nó que respondeu é retirado da lista dos nós que ainda têm que responder ao *ack*; se for mais antigo, então este nó está desactualizado, pelo que o *deamon* mantêm-no nesta lista, o que vai ter o efeito secundário de lhe transmitir o *update* mais recente (mais tarde, pelo mecanismo de *timeout*).

3.5.2.4 Acções síncronas

Esta secção vai descrever em detalhe as acções síncronas efectuadas pelo *deamon*, acções estas apenas processadas periodicamente (em "batch"), uma vez que as acções que despoletam não têm requisitos temporais apertados, ao contrário das mensagens de controlo e detecção dos movimentos. Basicamente, estas acções estão apenas relacionadas com a gestão dos *timers* do protocolo, que são descritos em [69].

---

[12] Note-se que a chegada da mensagem de update ao AP anterior do terminal pode ser utilizada para outras acções extra-mobilidade, como por exemplo a cópia do contexto anterior do terminal entre os dois APs (sendo esta funcionalidade útil para o suporte "*seamless*" de QoS na rede de acesso).

3.5.2.5 Controlo (Garantia de entrega)

O primeiro *timer*, configurado pela opção Timeout_ack do TIMIP, controla a retransmissão automática dos *updates* para os nós adjacentes da rede. Da forma como foi descrita anteriormente, cada nó tem uma lista dos nós que ainda não responderam com um *acknowledge*, o que pode ter acontecido porque houve perda de pacotes (tanto do *update*, como do *ack*), mas também se este não tiver sido aceite (ver acima). Assim, para cada nó nestas condições, depois de passar este tempo (Timeout_ack) sem resposta, o *update* é reemitido de novo para cada nó em questão.

3.5.2.6 Dados (Manutenção do estado)

O segundo *timer* controla a manutenção do estado do terminal móvel no nó, com o objectivo de verificar se o terminal ainda está no interior do domínio TIMIP.

No caso em que o terminal estiver silencioso, então este *timer* vai começar a ser disparado, com intervalos dinâmicos compreendidos entre as opções do TIMIP (Timeout_start) e (Timeout_min), com o valor inicial de Timeout_start, e com a optimização que o valor do *timeout* na GW é ligeiramente menor que os dos nós normais. De cada vez que o *timer* for disparado, o seu valor dinâmico é alterado da forma descrita em [69], e é gerado um pacote ICMP ECHO REQUEST destinado ao terminal para o forçar a responder, sendo este identificado como emitido pela GW da rede, e que tem o efeito secundário de refrescar todo o estado da rede de uma só vez.

De forma complementar, o *deamon* vai, de tempos a tempos, verificar “em batch” os cabeçalhos dos pacotes de dados recebidos por cada interface descendente (usando a interface o PCAP de dados), para verificar se algum pacote de dados emitido pelo terminal aparece da interface esperada (a interface onde o terminal está registado). Em caso afirmativo, então o terminal é refrescado, sendo o *timer* dinâmico alterado da forma descrita em [69].

Por fim, quando o *timer* é sucessivamente disparado sem resposta do terminal, então depois do tempo limite (Timeout_remove), o nó assume que o terminal saiu da rede ou foi desligado, sendo o seu estado removido do nó, incluindo as alterações efectuadas no módulo IP (encaminhamento, arp, etc.) (no entanto, note-se que este mecanismo só é utilizado para a remoção do terminal nesta situação; para as alterações do encaminhamento, é usada a sinalização explícita em todos os casos).

3.6 Módulo sMIP

O módulo sMIP vai complementar o TIMIP para o suporte de macro-mobilidade dos terminais legados, em que os agentes MIP executam as funções dos terminais em nome destes. Neste sentido, *embora se localize num agente MIP*, o sMIP é mais parecido ao cliente MIP do que ao Agente MIP.

Conceptualmente, o módulo sMIP é totalmente independente do TIMIP, com a excepção da que as acções de detecção, dado que são sempre despoletadas pelo TIMIP, dividindo-se entre a detecção de chegada e partida dos terminais. Deste modo, existe um módulo bem definido sMIP que implementa este protocolo, mas que coexise no interior do mesmo *deamon* TIMIP, o que simplifica a acção de execução do *software*, e comunicação entre os dois módulos.

Para implementar o sMIP, foi adaptado o *software* anteriormente realizado de implementação do cliente MIP, sendo a sua arquitectura, acções, algoritmos, etc., descritos em grande detalhe no trabalho [66].

Nesta implementação, cada GW sMIP vai executar vários clientes MIP em simultâneo, um para cada terminal legado com necessidade de macro-mobilidade, e executando apenas a fase 2 do Cliente MIP (registo). As outras fases do cliente MIP foram *desactivadas* na implementação do sMIP, e adaptadas da forma que seguidamnete se descreve:

3.6.1 Fase 1 – detecção

Quando o sMIP opera em conjunção com o TIMIP, este usa uma acção reactiva de detecção dos terminais substancialmente mais optimizada do que a acção clássica passiva. Neste sentido, as acções TIMIP são convertidas em acções de detecção do cliente MIP, de acordo com a seguinte tabela, o que leva cada cliente MIP no sMIP a alterar o seu estado de acordo com as acções originadas pelo TIMIP.

ACÇÃO TIMIP	==>	ACÇÃO sMIP
Power-up TIMIP de um terminal legado visitante com opção de sMIP		Detecção por parte do cliente MIP da chegada a uma rede visitada MIP com Fa (coincidente com a GW TIMIP)
Remoção de um terminal legado visitante		Detecção por parte do cliente MIP da chegada à sua rede de origem MIP, em que a GW é o seu HA

3.6.2 Fase 2 – registo

Para esta fase, o TIMIP vai apenas deixar que o sMIP se execute, pois este vai-se comportar automaticamente de acordo com o seu estado actual, induzido pelo TIMIP pela fase anterior. Desta forma, o cliente MIP vai registar-se automaticamente no seu HA quando o terminal chega ao domínio TIMIP pela primeira vez, manter este registo enquanto o terminal se mantiver no domínio, e desregistar-se no seu HA quando o terminal sair do domínio.

ACÇÃO TIMIP	==>	ACÇÃO sMIP
periodicamente, invocar o sMIP sem mudança de estado		criação/manutenção do processo de registo (fase 2) como está definido no MIP

3.6.3 Fase 3 – execução

Uma vez que o sMIP apenas terá que fazer chegar os pacotes ao domínio correcto, passando estes a serem encaminhados pela micro-mobilidade, a única acção do sMIP da fase de execução será a de aceitar os pacotes encapsulados enviados pelos diversos HAs, e para isto, na inicialização do *deamon*, este vai configurar o módulo IPIP com esta informação, utilizando para isto um comando externo apropriado.

3.7 Módulo MIP

De uma forma totalmente distinta do sMIP, o módulo MIP vai executar a componente de Agente MIP (componente servidor), sendo implementado como um *deamon* único a executar-se na GW da rede TIMIP, sendo configurado da forma apropriada para esta rede de acesso.

Tal como no caso anterior, este *software* está descrito em grande detalhe no trabalho [66], em relação à sua instalação, configuração e utilização. As únicas alterações necessária para a utilização do *deamon* MIP foram referente à desactivação tanto da fase 1 (localização) e 3 (registo), da forma semelhante ao sMIP, dado que estas funções são executadas pelo TIMIP.

3.8 Módulo DiffServ

3.8.1 Módulo DSR

Para o suporte de QoS na rede de acesso, foi utilizado o mesmo *software* desenhado para a rede de core da ilha do INESC, o DSR. Este *deamon* está detalhado no seu manual [46], e a sua instalação, configuração e utilização está descrita na referência [68].

Basicamente, DSR é um *deamon* que instancia a arquitectura Diffserv nos recursos de controle de tráfego existentes no Linux, que estão bem documentados nas referências [41] a [45], e em especial em [47], cobrindo todos os aspectos do controle de tráfego, relativamente aos filtros, filas, escalonadores, classes, etc., sendo estes elementos configuráveis no Linux por um comando externo denominado **tc**. Neste sentido, o *kernel* do Linux já tem as componentes necessárias para criar uma arquitectura Diffserv, embora não estejam integradas entre si, o que é o objectivo do DSR.

Assim, este *deamon* vai gerar a arquitectura Diffserv completa, com encaminhadores do tipo *edge* e *core*, apenas controlando os elementos do *kernel* por via da geração de comandos **tc**. Para tal, o DSR é configurável remotamente, ou por uma linha de comandos interna, o que lhe permite uma integração facilitada com outros componentes como os Bandwith Brokers. No caso concreto da rede de acesso *wireless*, o DSR é executado na sua configuração inicial (ver [68]), sendo depois customizado com filtros específicos para o tráfego que vai passar na rede, estando estes presentes em *scripts*.

Nestas circunstancias, embora a configuração dos filtros DSR seja manual, este já está preparado para receber externamente pedidos de QoS, o que exigiria apenas a implementação de um módulo controlador para tornar este aspecto do suporte de QoS totalmente automático[13].

Relativamente à arquitectura dos dois tipos de elementos de rede – *edge* e *core routers* – o DSR vai seguir a arquitectura definida pelo Diffserv, presente no Anexo 3.

Nestas, o encaminhador vai suportadas todas as classes de serviço actualmente normalizadas – EF, AF1 a 4 com 3 “drop precedences” e BE – escalonadas por um *scheduler* HTB

[13] O que poderá ser criado nomeadamente pela conversão Intserv=>DiffServ, com base na sinalização RSVP que descreve os fluxos de dados que passam na rede

(Hierarquical Token Bucket), que oferece uma precisão superior ao CBQ clássico [16], conjugado com um escalonador PRIO para o tráfego EF (limitado por um *token-bucket* TBF).

Assim, as interfaces a interfaces *edge* nos encaminhadores fronteira vão incluir todas as componentes de classificadores, policiadores e marcadores, enquanto que as do tipo *core* já não as vão ter, dado que só se aplicam ao interior do domínio Diffserv, onde os pacotes já terão que estar devidamente marcados e policiados.

Para criar esta arquitectura, o DSR vai usar os já referidos elementos de controlo de tráfego do linux, da forma detalhada pelas figuras presentes no Anexo 4.

3.8.2 Módulo DSR-Stats

O módulo DSR-Stats vai ser complementar ao módulo DSR descrito, por permitir a recolha periódica de estatísticas Diffserv do encaminhador, e com a possibilidade de as enviar remotamente para entidades centralizadoras da informação.

Uma destas será o sistema de monitorização desenvolvido para a análise da ilha do INESC no MOICANE, que permite a recolha, análise e comparação dos dados coleccionados dos diversos encaminhadores Diffserv da ilha do MOICANE, nomeadamente os presentes na rede de core, e na rede de acesso *wireless* principal.

Ambas estas componentes de monitorização estão descritas em grande detalhe em [46], tendo sido adaptadas para incluir o suporte da rede de acesso *wireless*.

4. Avaliação de resultados

Nesta capítulo vão ser descritos os testes efectuados para avaliar a implementação do protótipo desenvolvido, nomeadamente na solução de mobilidade proposta para terminais legados, e de outras características complementares presentes na rede de acesso. Estes testes foram distinguidos entre *funcionais* e de *desempenho*, de acordo com a sua natureza e complexidade de análise e demonstração.

Com base num subconjunto dos testes de desempenho, com aplicações multimédia e de medição rigorosa dos parâmetros da rede, foi preparado um guião de demonstração prática da rede aos Auditores Internacionais, e outros especialistas Nacionais e Internacionais, que atestaram a respeito da qualidade da solução desenvolvida.

Assim, este capítulo vai começar por descrever as aplicações utilizadas, e as condições e resultados dos testes funcionais e desempenho, sendo depois descrito o referido guião de demonstração.

4.1 Testes da Rede

4.1.1 Aplicações utilizadas nos Testes

Para realizar os testes da rede de acesso, utilizaram-se diversas aplicações de teste, divididas entre aplicações genéricas e aplicações de ensino à distância criadas no contexto do projecto MOICANE [54].

Assim, as primeiras incluem os programas clássicos de teste ao protocolo IP PING e TRACEROUTE (ver [31]), com os quais é possível avaliar de uma forma normalizada em todos os sistemas a conectividade simples entre dois nós da Internet, incluindo a latência da comunicação (ping) e o caminho que os pacotes utilizam para chegar ao destino (traceroute).

Por outro lado, foi utilizado um programa (MGEN/DREC) [50] para efectuar a geração de tráfego arbitrário de teste da rede, sendo utilizado para criar diversas situações que vão testar e medir as características especiais da rede. Este programa (ver Figura 11) permite a geração de vários fluxos de dados UDP entre nós IP, com suporte para *unicast* e *multicast*, possibilitando a emulação de padrões de tráfego de outras aplicações, ou apenas para carregar a rede.

No emissor, o tráfego pode ser gerado variando o número, tamanho, e distribuição dos pacotes; no receptor, o tráfego pode ser guardado em ficheiro, para posterior calculo de estatísticas relativamente ao débito, perdas, atraso, *jitter* da comunicação estabelecida, podendo estas serem bastante mais precisas que as possibilitadas pelo programa ping.

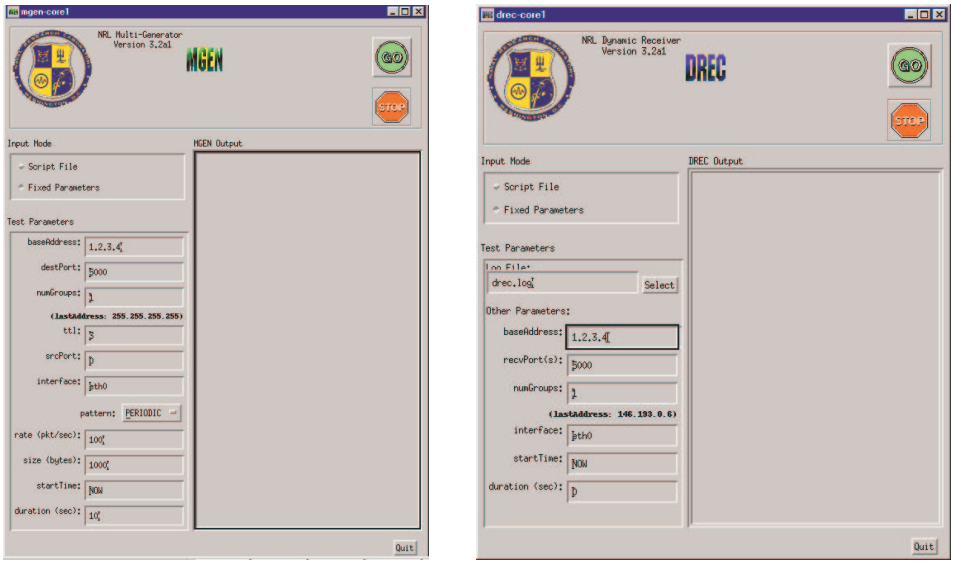


Figura 11: Gerador de tráfego MGEN/DREC

Este programa considera todos os seus valores ao nível UDP, de tal forma a medir as condições em que as aplicações dos utilizadores vão receber da rede, mas no entanto, em alguns testes a executar, pretende-se avaliar especificamente as características da tecnologia de rede 802.11, pelo que este programa foi alterado para efectuar as suas medições ao nível IP, de forma a que os tamanhos dos pacotes gerados sejam relativos aos entregues ao nível 2, e os débitos recebidos relativos à recepção pelo nível 3.

Além dos resultados extremo-a-extremo disponibilizados pelo MGEN/DREC, também foram efectuadas medições no interior da rede, sendo verificadas as estatísticas dos *deamons* dos *routers* (TIMIP/sMIP/DSR), e a utilização de *packet sniffers* nas interfaces de forma a verificar a estrutura dos pacotes IP gerados/recebidos, como também para efectuar medições de débitos em tempo real que são recebidos e enviados nestes elementos da rede. Para tal utilizou-se os programas Ethereal/TcpDump e o IPTraf para as funções descritas ([51], [52], [53]).

Por fim, a características da rede foram também testadas e demonstradas utilizando as aplicações dos clientes finais criadas para o ambiente Windows. Entre estas, foram utilizadas aplicações genéricas como o acesso a servidores WWW (Internet Explorer), transferência de

ficheiros (FTP), acesso remoto virtual a outros clientes (VNC), e aplicações multimedia como o programa de vídeo conferencia Netmeeting.

Além destas aplicações genéricas, foram utilizadas as aplicações desenvolvidas pelos parceiros Gregos no projecto MOICANE [54], e que criam um ambiente de *e-learning* necessário para um ensino à distância em que os intervenientes podem ser móveis. Estas são constituídas por diversas aplicações, entre o Chat (comunicação remota), VOD (Vídeo on Demand), VidConf (Audio e Vídeo Conferência remota), e Virtual Lab (ambiente de laboratório remoto que permite efectuar experiências e medições em instrumentos distantes), entre outras.

Das várias aplicações disponíveis, as mais interessantes para a demonstração da rede de acesso são as aplicações multimédia VOD e VidConf (ilustradas nas figuras seguintes), dado que têm requisitos específicos que deverão ser cumpridos por parte da rede.

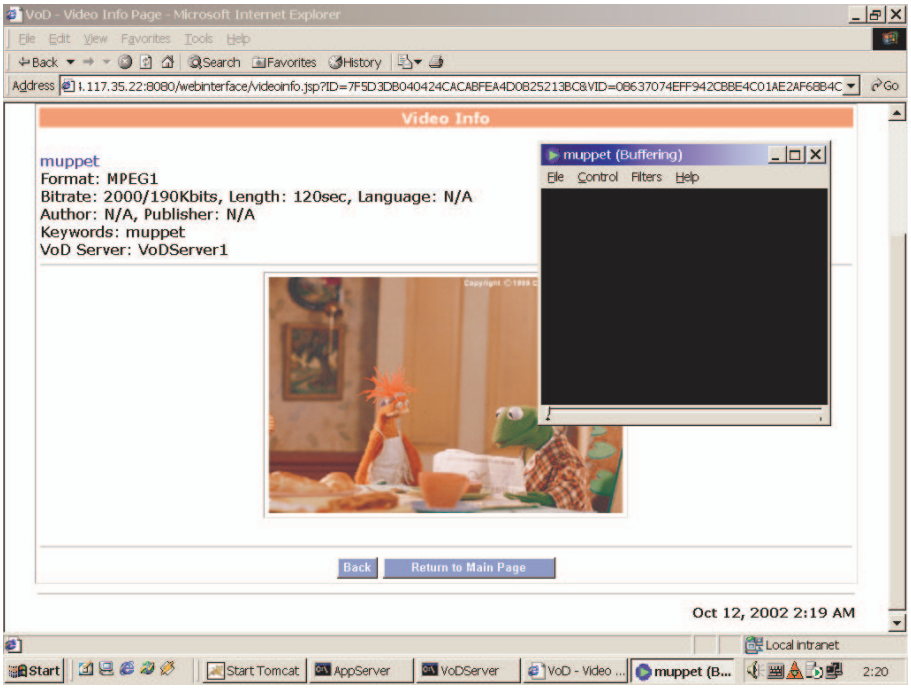


Figura 12: Cliente da Aplicação VOD desenvolvida no MOICANE

4.1.2 Testes Funcionais

Os testes funcionais vão focar os aspectos mais básicos da rede de acesso, que procuram verificar apenas se as características desejadas da rede de acesso estão presentes e correctas na rede implementada, como o TIMIP, o sMIP e o Diffserv, sem preocupações de desempenho, de tal forma a que estes testes sejam normalmente efectuados com ferramentas muito simples de teste da rede.

Dado que estes testes cobrem essencialmente casos particulares da cobertura dos testes de desempenho, neste relatório os testes que constituem esta classe apenas estão identificada na tabela seguinte.

TESTES FUNCIONAIS
Conectividade IP básica
Mobilidade TIMIP
Garantia de entrega das mensagens s de controlo TIMIP
Conectividade IP em mobilidade Global
Mobilidade IP em mobilidade Global
Filtragem DiffServ
Policiamento DiffServ
Marcação DiffServ
Testes Funcionais de Diffserv com Mobilidade TIMIP

4.1.3 Testes de Desempenho

Os testes de desempenho vão focar aspectos presentes na rede de acesso que os testes funcionais provam a sua correcta execução, mas que, por incluírem aspectos especiais na sua arquitectura e/ou optimizações concretas na sua implementação, este testes vão medir com rigor as operações despoletadas, de forma a avaliar a qualidade final da solução desenhada e implementada.

4.1.3.1 Velocidade do handover TIMIP

Este teste vai verificar qual é o peso da adição da micro-mobilidade TIMIP aos domínios IP, sendo este comparado com a utilização apenas da mobilidade proporcionada pelo nível 2.

Para isso, vai-se usar a mesma configuração da rede já testada nos testes funcionais, mas agora utilizando o gerador de tráfego MGEN ao invés do programa PING. Neste sentido, o gerador de tráfego emite um fluxo constante periódico de pacotes UDP numerados com o período de 1 milisegundo.

Estes pacotes são destinados a um terminal móvel, que recolhe e grava o fluxo para ficheiro, e que se movimenta alternadamente entre os dois APs; e por outro lado, estes verificam o tempo em que recebem o pacote de gestão de *associate* do 802.11, guardando esta informação em

ficheiro[14], sendo assim possível analisar o instante de tempo em que a ligação de nível 2 é reestabelecida pelo 802.11b.

Note-se que a rede de acesso já exige que os APs da rede estejam sincronizados entre si; adicionalmente, para este teste particular, também o relógio do terminal terá que estar sincronizado com o dos APs, de forma a possibilitar as comparações das medições efectuadas.

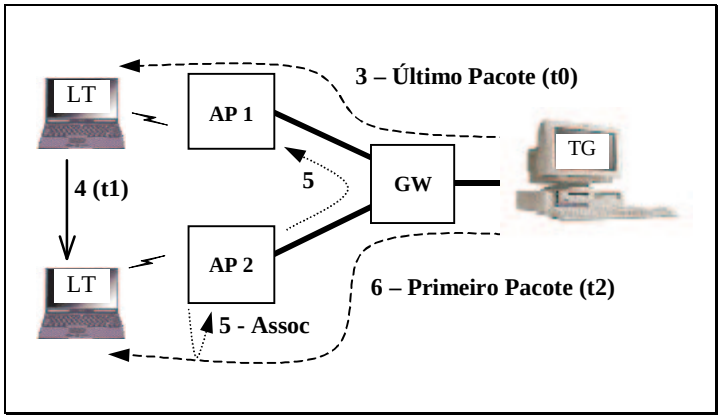


Figura 13: Detalhe do handover para teste da velocidade TIMIP

Neste sentido, a movimentação do terminal é decomposta da seguinte forma (ver Figura 13):

1. LT1 está associado ao AP1, e está a receber o tráfego gerado em TG.
2. LT1 movimenta-se de forma a aproximar-se da célula do AP2, ao mesmo tempo que se afasta do AP1
3. LT1 recebe o último pacote UDP pelo AP1, antes de perder a conectividade física – **Tempo T0**
4. LT1 executa o *handover* do nível 2, que termina assim que o pacote 802.11 de *associate* é recebido no AP2 **– Tempo T1**
5. O nível 2 do AP avisa o nível 3 da chegada do novo terminal móvel, que inicia a reconfiguração da rede, bastando que o AP2 e a GW alterem o seu encaminhamento para a nova localização do terminal.
6. LT1 recebe o primeiro pacote UDP entregue pelo AP2 **– Tempo T2**

Comparando os valores de **T0, T1** e **T2**, é possível extrair o tempo total do *handover* (T2 - T0), a componente relativa do TIMIP (T2 - T1), e a componente relativa do nível 2 (T1 - T0)

[14] Para máxima precisão, é marcado o tempo no *driver* assim que o *interrupt* do pacote de *associate* é disparado.

(ver Anexo 6Anexo 6). Teoricamente, a parte do TIMIP deverá ser muito menor que a parte do nível 2, dado que a acção de detecção é totalmente reactiva, e a fase de registo está optimizada para efectuar *handovers* locais ao terminal, e no pior caso nunca além do interior do domínio TIMIP.

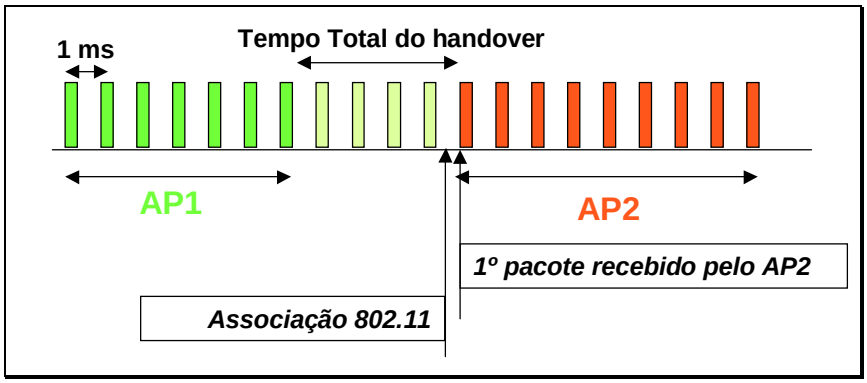


Figura 14: Detalhe do teste da velocidade do handover TIMIP

Executando o teste as vezes necessárias para um alto grau de confiança, verificou-se que o TIMIP demora em média **2ms** a transitar (0,224ms desvio padrão), enquanto que o 802.11 demora **101.06ms** a transitar (25,650ms desvio padrão). Os dados completos estão presentes no Anexo 6, incluindo um exemplo da forma de medição.

Analisando estes resultados, verifica-se então que em condições óptimas, a mobilidade TIMIP demorou em média 1ms por cada nível da árvore de nós necessário para concretizar o *handover*.

Uma vez que o TIMIP optimiza o registo (local), então os *handover*s são sempre limitados à parte da árvore dos APs envolvidos; uma vez que normalmente estes estarão localizados perto um do outro, então a parte envolvida da árvore será pequena, que se reflecte em poucos níveis da árvore (na maioria das movimentações).

Mesmo no pior caso, em que o registo tem que percorrer toda a árvore desde o AP até à GW, o peso do *handover* TIMIP será sempre (em condições óptimas) bastante menor que o tempo necessário na transição do 802.11, sendo assim totalmente despercebido em comparação com este.

Realce-se que a grande diferença é que o TIMIP permite a mobilidade optimizada ao longo de domínios inteiros, comparada com o interior de LANs no *handover* de nivel2 do 802.11, o que complementado com outras tecnologias IP mostra a sua abrangência superior.

4.1.3.2 Velocidade dos handovers em mobilidade Global

Este teste vai verificar qual é o peso da adição da micro-mobilidade TIMIP, conjugada agora com a macro-mobilidade sMIP aos domínios IP, o que permite a mobilidade global ao longo de toda a Internet. Neste sentido, este teste vai se basear no teste funcional efectuado de mobilidade global, mas medido da forma rigorosa especificada no teste de desempenho da micro-mobilidade (ver teste anterior). Neste teste, ilustrado na Figura 15, um terminal LT vai-se movimentar entre 3 APs que lhe estão acessíveis, de forma a efectuar movimentos entre o seu domínio de origem e um domínio visitado (movimentos 1 e 4), e no interior do domínio visitado entre os seus APs (movimentos 2 e 3).

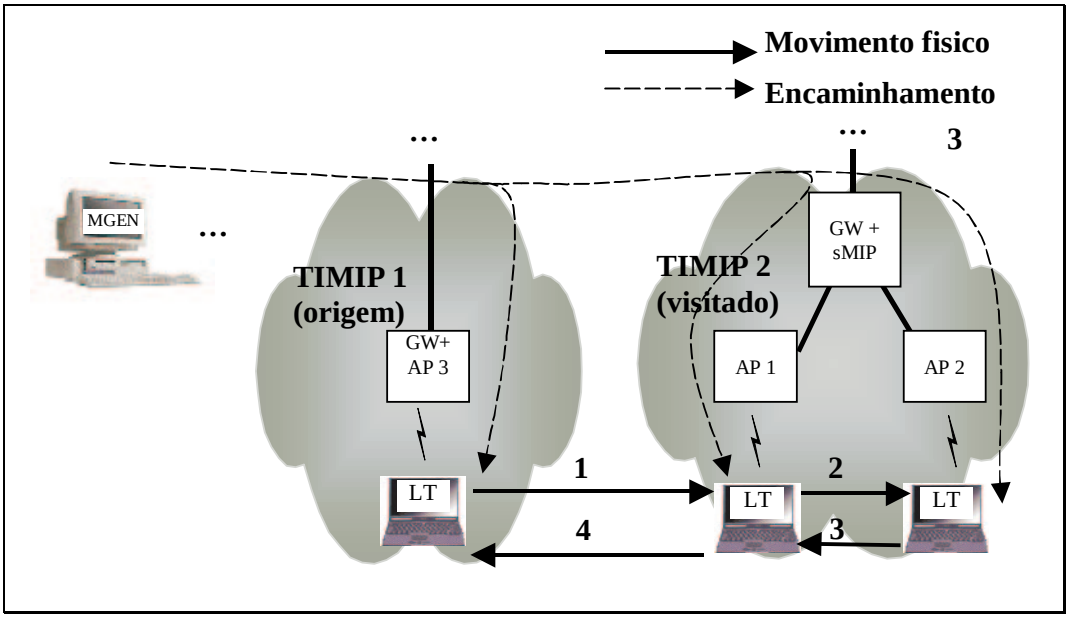


Figura 15: Teste de performance de mobilidade global TIMIP + sMIP

O ponto chave que este teste evidencia é a total independência dos dois tipos de mobilidade, dado que os movimentos 2 e 3 no domínio visitado vão ser instanciados apenas com o recurso à micro-mobilidade, independentemente da distância do terminal até ao domínio de origem, e assim com durações em tudo semelhantes às já encontradas no teste anterior, dado que as alterações a efectuar serem sempre locais limitadas ao domínio visitado.

Por outro lado, as movimentações 1 e 4 entre os domínios terão previsivelmente uma latência apreciável da ordem dos segundos, uma vez que (propositadamente) o processamento das acções de macro-mobilidade não têm qualquer optimização ou preocupação relativas ao seu desempenho (rapidez), conjugado com atrasos impostos pelo protocolo MIP.

Efectuando-se este teste, verificou-se os resultados sumariados na Tabela 1, obtidos com base em medições presentes no Anexo 7.

TIPO	de	para	Mobilidade:	Descrição	Latência
1	AP3	AP2	Macro+Micro	entrada num domínio visitado	3117 ms
2	AP2	AP1	Micro	movimentação em domínio visitado	3 ms
3	AP2	AP2	Micro	movimentação em domínio visitado	2 ms
4	AP2	AP3	Macro+Micro	retorno ao domínio de origem	5 ms

Tabela 1: Resultados teste sMIP+TIMIP

Nestes, é notória a diferença substancial entre o tempo necessário para restabelecer a conectividade entre a movimentação entre domínios, e no interior do domínio. Tal como foi previsto, as movimentações do tipo 1 só são finalizadas segundos depois de terem sido iniciadas, pois envolvem macro-mobilidade; por outro lado, as movimentações do tipo 2 e 3 são totalmente semelhantes às do teste anterior, com a diferença que operam num domínio visitado, sendo suficientes para provar a independência entre os dois tipos de mobilidade.

Por fim, as movimentações do tipo 4, relativas ao retorno dos terminais aos seus domínios de origem poderá parecer surpreendente, por embora envolver acções de macro-mobilidade, está na mesma ordem de grandeza dos movimentos de micro-mobilidade (embora ligeiramente superior). Este comportamento é explicado por optimizações explicitas na implementação do sMIP, dado que a acção de retorno dos terminais apenas terá que remover o túnel criado, não necessitando de qualquer troca de sinalização entre domínios. Outro factor que contribui para este valor é a optimização do sMIP possuir uma forma de detecção reactiva (despoletada pelo TIMIP que detém também de detecção reactiva despoletada pelo 802.11), que reduz esta fase a um valor marginal.

4.1.3.3 Débito oferecido pelo 802.11b

Este teste vai verificar qual é o débito máximo atingido em condições óptimas pelo 802.11b num fluxo simples *downlink*. Tal como foi analisado na descrição teórica, o 802.11b tem *overheads* muito pronunciados no nível 2 (como o *acknowledge*) e no nível 1 (*multi-rate*), que baixam significativamente o débito alcançado quando se utilizam pacotes pequenos no nível 3.

Para isto, vai ser testado com um fluxo constante de 11 Mbit/s ao nível IP, constituído por diversas combinações de tamanho do MTU do nível 2 *vs* número de pacotes por segundo emitidos, por forma a gerar o débito considerado. Estes fluxos são medidos durante largos períodos de tempo, desde o emissor MGEN localizado no AP, ao receptor DREC localizado no terminal, sendo no final analisadas as estatísticas da transmissão

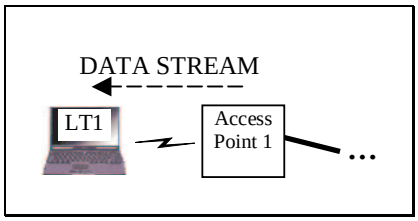


Figura 16: Teste de medição do débito máximo do 802.11b

Executando o teste, verificou-se que o débito atingido varia bastante com o tamanho dos pacotes, variando desde um máximo de **0.404 Mbit/s** para pacotes pequenos de **48 bytes IP**, e um máximo de **5.5 Mbit/s** para pacotes de **1500 bytes IP**. Os resultados estão resumidos no gráfico abaixo e os dados completos presentes no Anexo 8.

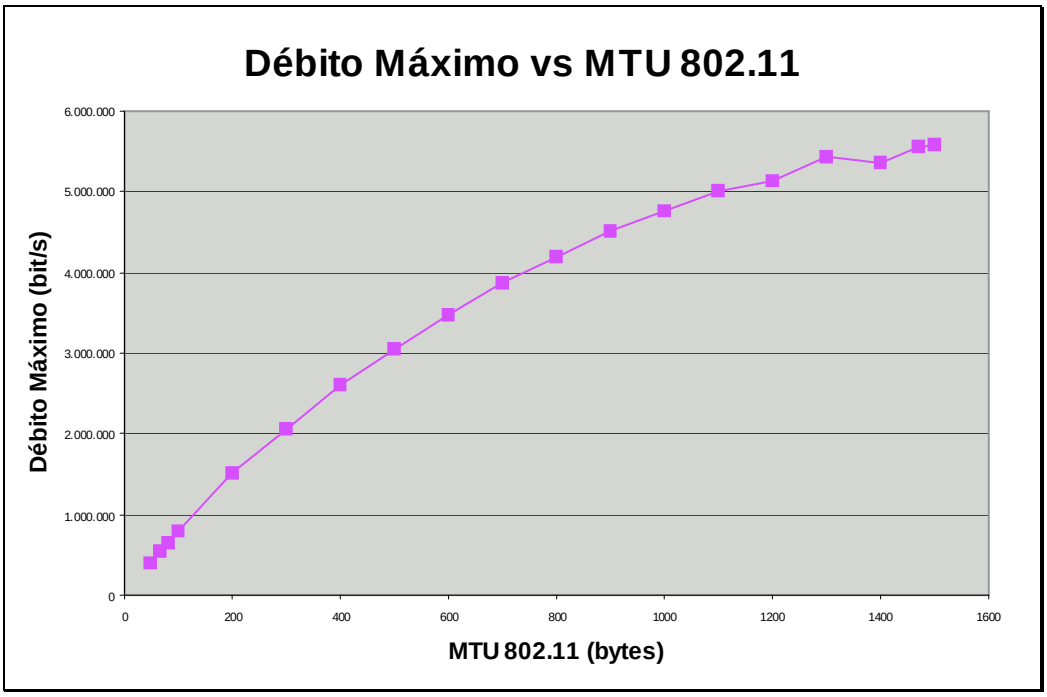


Figura 17: Resultados do Débito oferecido pelo 802.11b

Note-se que estes resultados estão em conformidade com estudos teóricos, e mostram que esta tecnologia oferece débitos bastante mais baixos que os anunciados pelos fabricantes (11 Mbit/s).

4.1.3.4 Teste de atraso da classe EF

Este teste vai avaliar o desempenho da classe EF na rede de acesso. Tal como está normalizada, esta classe define que terá que existir um limite máximo para o atraso dos pacotes respectivos, o que terá que ser respeitado mesmo quando esta classe coexistir com outras classes de tráfego menos prioritárias. Acresce ainda que a utilização de *software* para concretizar o suporte de QoS também vai introduzir um peso, que se pode reflectir neste requisito do EF.

Deste modo, vai ser verificado o valor do atraso máximo da classe EF quando esta compete com outras classes, e o valor do mesmo quando está sozinha, medindo assim o peso mínimo desta implementação do DiffServ. Para referência, ambos os valores são comparados com o valor do atraso nominal, em que não existem mecanismos de QoS no caminho dos pacotes IP na rede.

Para isto vai ser utilizada uma configuração (ver Figura 18), constituída por dois terminais fixos LT1 e LT2, localizados permanentemente no AP2, e dois PCs no exterior da rede que têm ligações dedicadas independentes à GW da rede, estando os relógios destes elementos sincronizados entre si.

Para este teste, o terminal móvel 1 vai receber um fluxo de teste emitido pelo PC1 e classificado na rede como tráfego prioritário EF, de forma a avaliar o atraso necessário no percurso dos pacotes EF nesta rede, e o segundo PC vai gerar um fluxo de carga BE, com as mesmas características dos anteriormente utilizadas, sendo recebido pelo terminal LT2. [15]

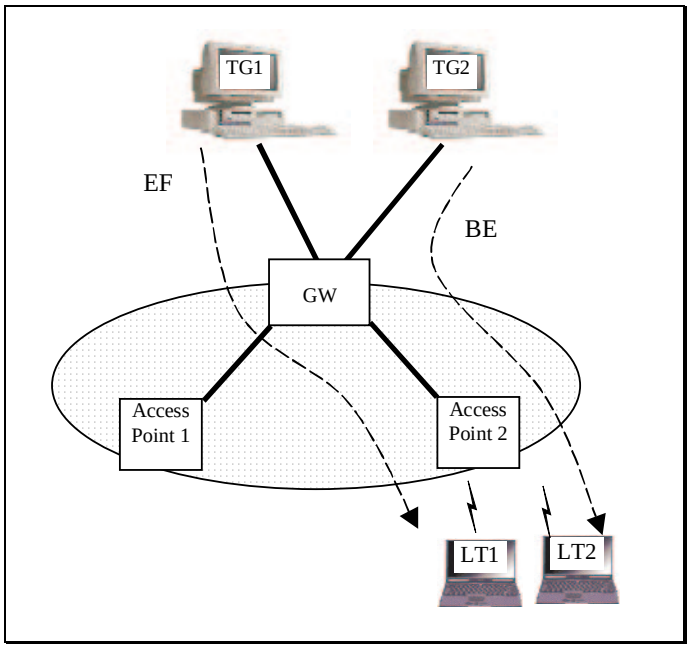


Figura 18: Testes de atraso do EF

Nestas condições vai ser medidos os seguintes atrasos na propagação dos pacotes:

[15] Outra forma de realizar esta experiência seria a de utilizar o programa **ping** para verificar o tempo de ida e volta entre os nós em questão, dado que aí a fonte de relógio seria a mesma (apenas o relógio do emissor). No entanto, esta experiência não pode ser testada de uma forma fiável, dado que os pacotes de respostas teriam que competir com o fluxo BE para aceder ao meio, (dado não existir suporte de QoS de nível 2 o 802.11).

a) fluxo EF sem carga BE, sem os mecanismos de DiffServ activos

b) fluxo EF sem carga BE, com os mecanismos de DiffServ activos

c) fluxo EF com carga BE, com os mecanismos de DiffServ activos

Estas medições estão resumidas na tabela seguinte:

teste	QoS Activo?	Carga BE?	Pacotes Gerados	Pacotes Recebidos	Latência Média	Latência Mínima	Latência Máxima
a)	Não	Não	1000	1000	1 ms	0 ms	10 ms
b)	Sim	Não	1000	1000	1 ms	0 ms	10 ms
c)	Sim	Sim	1000	1000	6 ms	1 ms	10 ms

Tabela 2: Teste de desempenho da classe EF

Destes resultados, pode-se verificar que todos os valores estão na mesma ordem de grandeza, sendo assim os diferentes cenários em teste interpretados como semelhantes para os utilizadores da rede de acesso.

Comparando os resultados a) e b), verifica-se que a adição do *software* de Diffserv à rede de acesso incorre um *overhead* mínimo em todas as situações, mas que é tão pequeno que não consegue ser detectado com o mecanismo criado, dado que o processo de sincronização dos relógios apenas garante uma margem de erro de 1ms, e por o MGEN ter uma precisão limitada no acesso ao relógio (também com um erro mínimo de 1ms).

Esta medição mostra contudo, que a estrutura de controlo de tráfego DiffServ presente no caminho crítico do *forwarding* dos pacotes de dados é escalável, principalmente por estar implementada em exclusivo no interior do *kernel* do Linux.

A segunda comparação entre os resultados b) e c) mostra que o tráfego EF sofre em média um aumento na sua latência de transmissão quando existem classes menos prioritárias a competir para a transmissão pela interface. Contudo, este aumento mantém o valor médio da latência na mesma ordem de grandeza, de tal forma a que os requisitos apertados dos SLAs associadas aos fluxos EF continuam a serem cumpridos, mesmo no pior caso.

4.2 Guião da Demonstração final

A avaliação dos objectivos definidos foi efectuada pela demonstração prática da rede de acesso wireless com as características acima descritas a Auditores Internacionais Independentes, encarregues de avaliarem a qualidade global do projecto MOICANE, tendo sido estruturada conforme se descreve em seguida.

A demonstração da rede de acesso 802.11b foi focada nos aspectos de suporte de micro-mobilidade TIMIP integrada com o suporte de QoS estático baseado em DiffServ. Para tal, foram usados dois PCs portáteis, executando o PC1 o cliente de VoD, e o PC2 sendo o um receptor de tráfego de enchimento. Este trafego é gerado na GW da rede de acesso, e é suficiente para saturar o meio físico no AP onde o portátil PC2 estiver localizado.

Durante a demonstração, apenas o portátil com o cliente de vídeo PC1 é movimentado entre os dois APs, forçando a ocorrência de transições (roaming), que são sinalizadas por um "beep" sonoro no AP envolvido. Esta topologia está ilustrada na Figura 19.

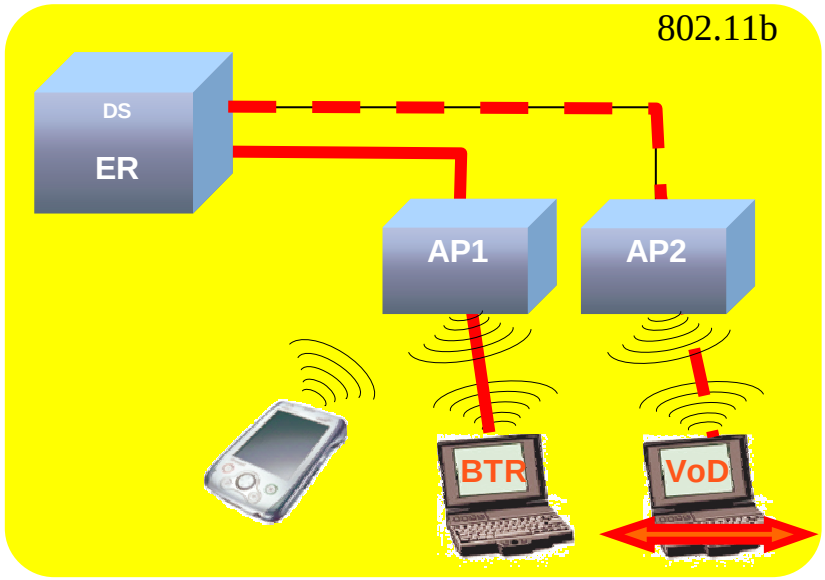


Figura 19: Arquitectura de TC criada pelo DSR nas Interfaces Egress

Na primeira experiência, o suporte de QoS encontrava-se desligado, pelo que quando o cliente de Video estava ligado à rede pelo AP1, o filme não tinha qualquer qualidade visual; por outro lado, quando estava ligado ao AP2, voltava a ter boa qualidade, dado esta parte da rede não estar carregada.

Na segunda experiência, o suporte de QoS foi aplicado à rede de acesso, o que resultou na manutenção da qualidade do filme permanentemente, com apenas uma pequena diminuição

no preciso instante da operação de transição (resultante da necessária mudança de frequências físicas).

Seguidamente, para uma análise mais profunda do suporte de mobilidade IP em 802.11b, foi executado o teste de desempenho que avalia o tempo do handover, considerando as componentes distintas dos níveis 2 e 3, sendo apenas estes último relacionado com o protocolo de micro-mobilidade TIMIP desenvolvido. Para tal, um conjunto de provas ("probes") é gerado periodicamente a cada milisegundo, desde a GW da rede até ao PC2, que é agora movido entre os dois APs. Depois de o handover se efectuar, é verificado o número de pacotes não-recebidos de *forma sequencial*, a que corresponde o tempo total do handover, e o tempo da recepção da associação de nível 2. Considerando estes dados, o tempo do nível 3 é apenas a diferença entre o tempo da recepção do primeiro pacote pelo novo AP e a recepção da mensagem associação.

Como evidenciado na Figura 20, foi demonstrado que o tempo total do handover foi de aproximadamente 59 msec (valor roxo), e que a contribuição para este valor pelo nível IP representa um pequeno peso de baixa magnitude, por ser apenas 4 ms (diferença entre os valores azuis).

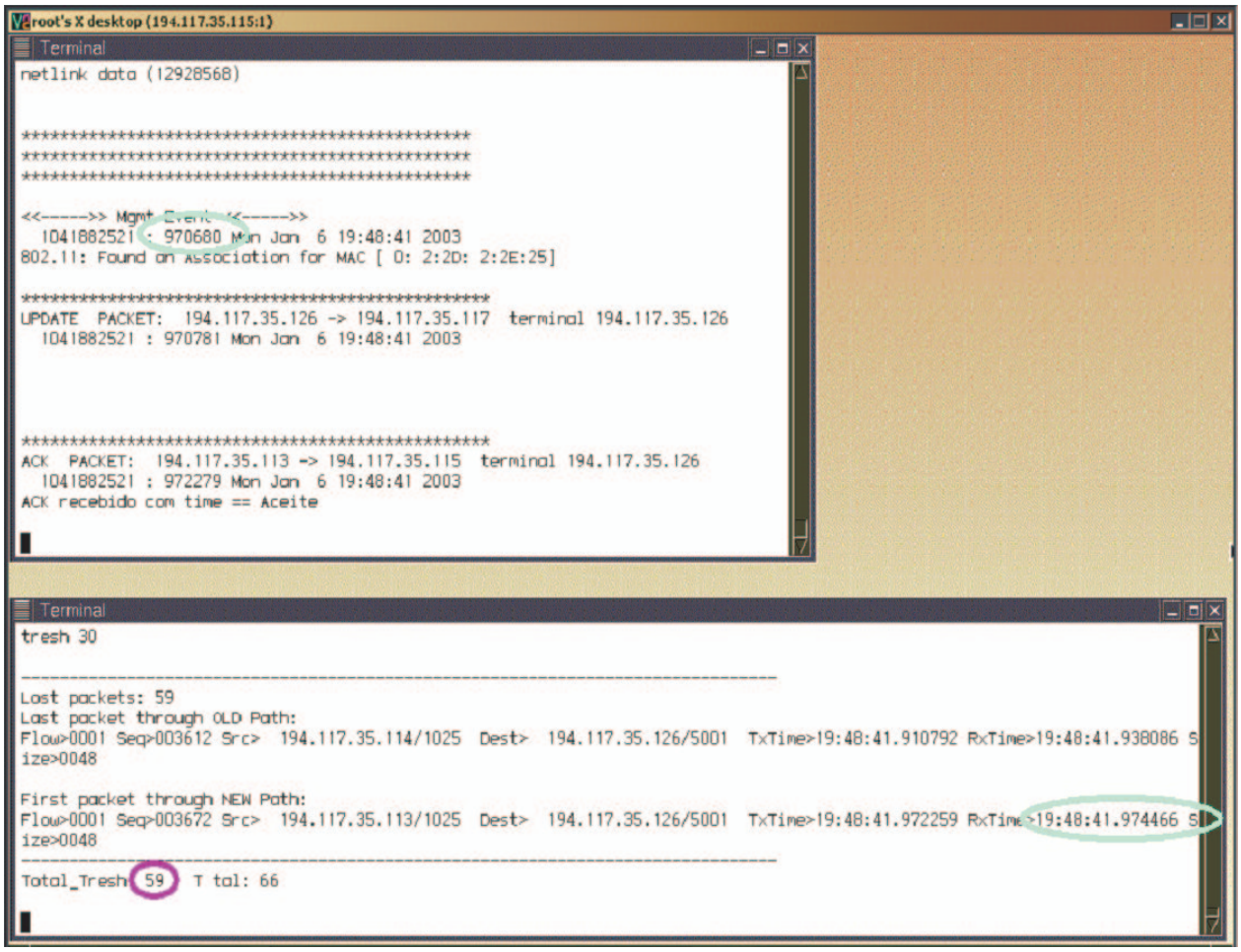


Figura 20: Exemplo de recolha de Dados TIMIP

Valor Roxo	Número de pacotes perdidos (Tempo Total do Handover)
Valores Azul-Claros	Tempo de recepção do primeiro pacote e tempo de recepção da mensagem de Associação (Tempo do TIMIP)

5. Conclusões

Este estágio formal propôs-se a desenvolver um trabalho de Engenharia *Completo*, sendo este uma condição necessária para a acreditação como membro efectivo do candidato à Ordem dos Engenheiros, nomedamente por ter sido devidamente orientado e acompanhado pelo respectivo Patrono (entre outros Engenheiros e profissionais altamente qualificados).

No caso particular, foram focados neste estágio os *aspectos de Engenharia* necessários no desenvolvimento e implementação de um protótipo de Investigação, no contexto de um projecto Europeu de I&D, que utiliza tecnologia de ponta e componentes inovadoras – a rede do INESC no MOICANE. Assim, foram objectivos deste estágio a criação, desenho e implementação da rede de acesso de demonstração neste projetco , mas também a concepção e desenvolvimento completo de componentes inovadoras que foram investigadas no contexto de uma Tese de Mestrado (decorrente em paralelo a este estágio).

Para o primeiro objectivo considera a execução de um trabalho de engenharia puro, em que foi necessário usar, integrar e alterar componentes de software e hardware já existentes, bem como a necessidade de trabalho em equipa; por outro lado, o segundo objectivo, mais vocacionado para Investigação, também teve componentes substanciais de engenharia, na concepção e desenvolvimento completos de componentes de software distribuídos reutilizáveis de tecnologia de ponta.

A qualidade das noções de Engenharia apreendidas e usadas no presente estágio é patente e reconhecida no presente relatório, descrevendo-se detalhadamente as mais importantes e releveantes opções, metodologias e componentes especifícas da solução final; por outro lado, a qualidade do trabalho em si é patente nos resultados medidos, e nas demonstrações efectuadas a Auditores Internacionais e outros especialistas, que atestaram e aprovaram a qualidade do prototipo desenvolvido.

Neste sentido, considera-se os objectivos do presente estágio, acima referenciados, como atingidos, sendo esta reflexão totalmente confirmada por pareceres sobre o estágio, apresentados em anexo a este relatório. Neste sentido, este estágio profissional foi de fulcral importância para o candidato, abrindo diversas hipóteses de trabalho futuro ao mesmo, e tendo constituido uma aprendizagem substancial de Engenharia.

Parecer da Entidade

Parecer do Patrono

Apêndices

Anexo 1 Detalhes da Ilha do INESC

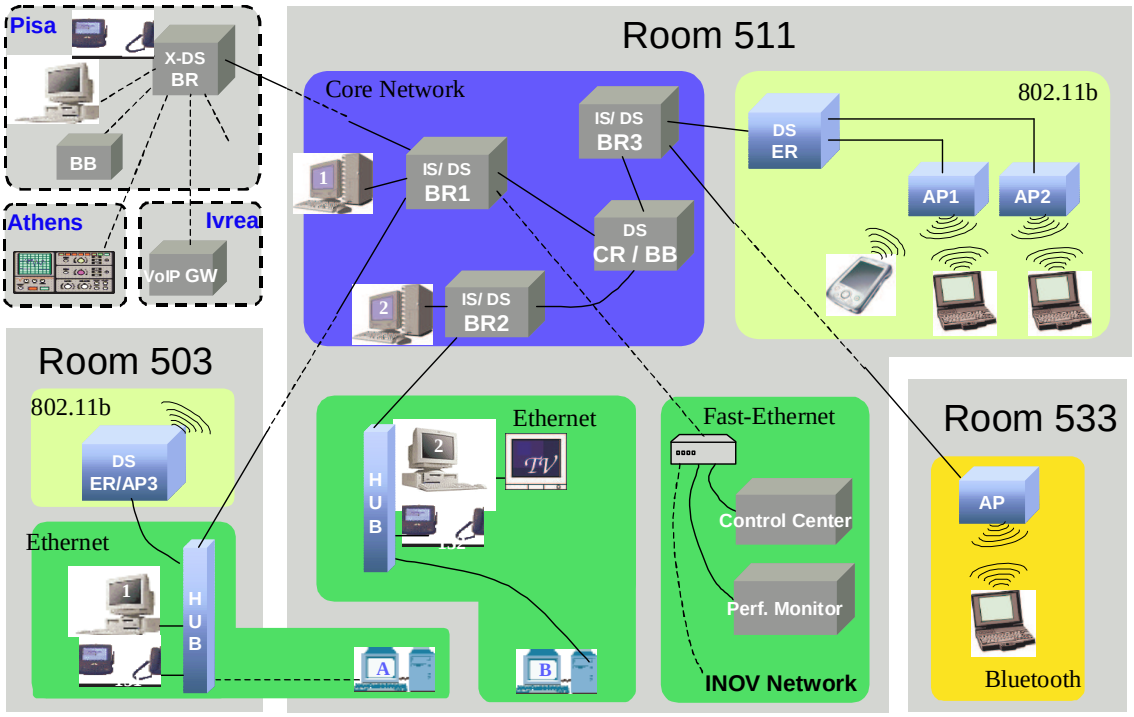


Figura 21: Esquema da ilha do INESC no demonstrador do MOICANE

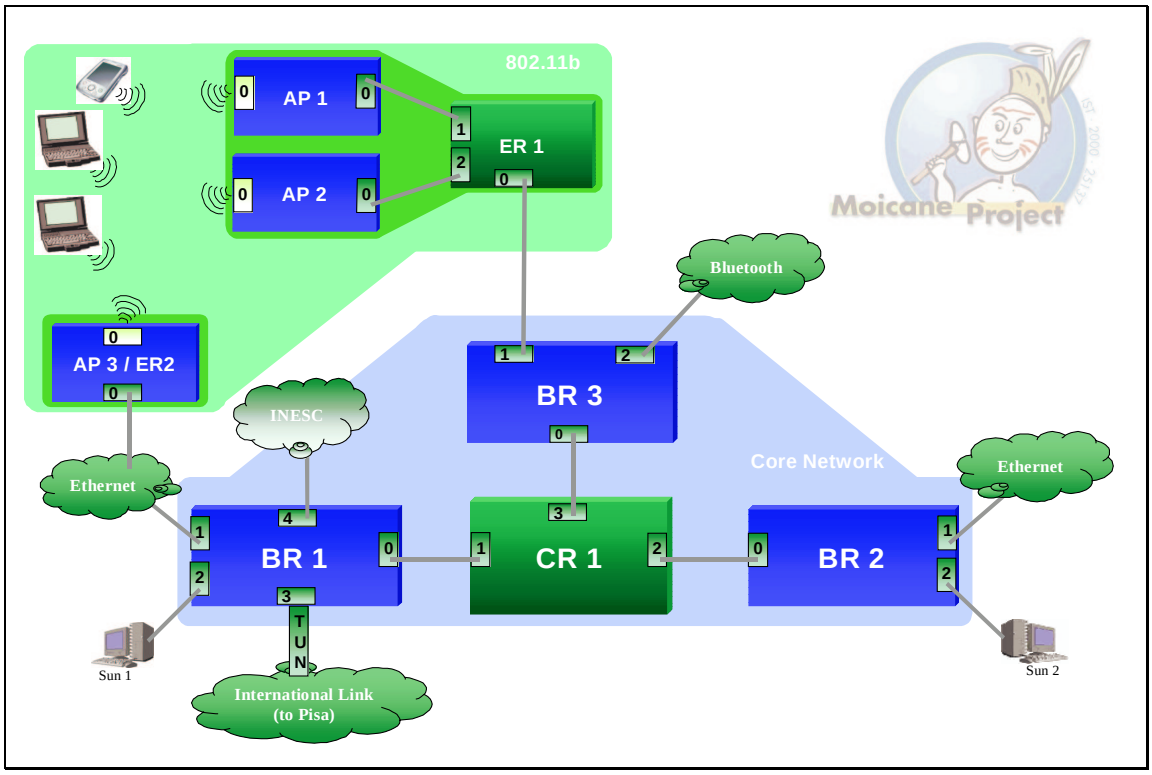


Figura 22: Ilha do INESC – Sistema de Monitorização

Anexo 2 Detalhes da interface PCAP

recolha de pacotes de CONTROLO:

```
match ip protocol 1 and icmp[0]=200
```

recolha de pacotes de DADOS:

caso 1) interface wireless 802.11 no AP

```
match ip dst 224.0.0.1 or ip dst 255.255.255.255 or ( not ether src
<THIS_ITF> and ether dst <THIS_ITF> )
```

caso 2) interface ethernet ligada a um nó descendente

```
match ether src <NODE_ITF> and ether dst <THIS_ITF>
```

caso 3) interface ethernet na GW ligada ao exterior

```
match none
```

Anexo 3 Arquitectura Diffserv

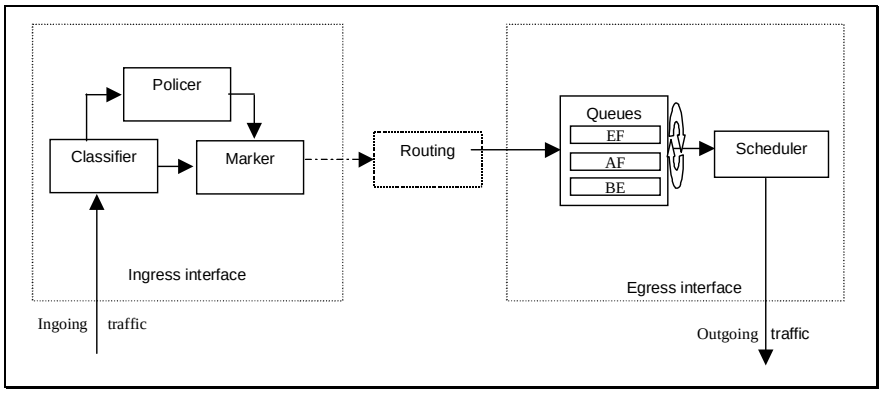


Figura 23: Interface edge Diffserv

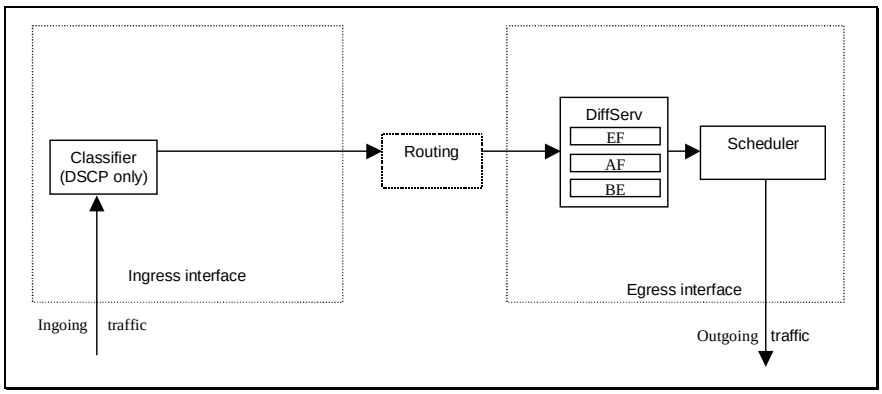


Figura 24: Interface core Diffserv

Anexo 4 Módulos de Controle de Tráfego em Linux

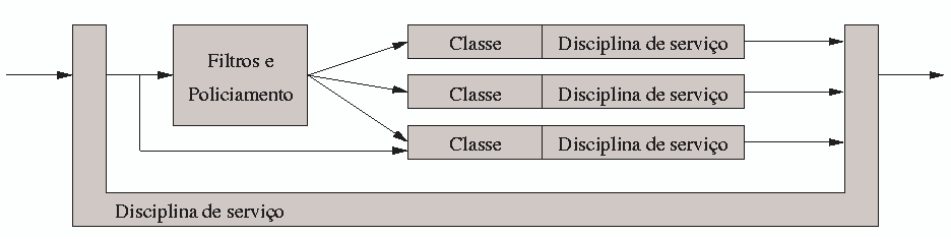


Figura 25: Elementos de controlo de tráfego no Linux

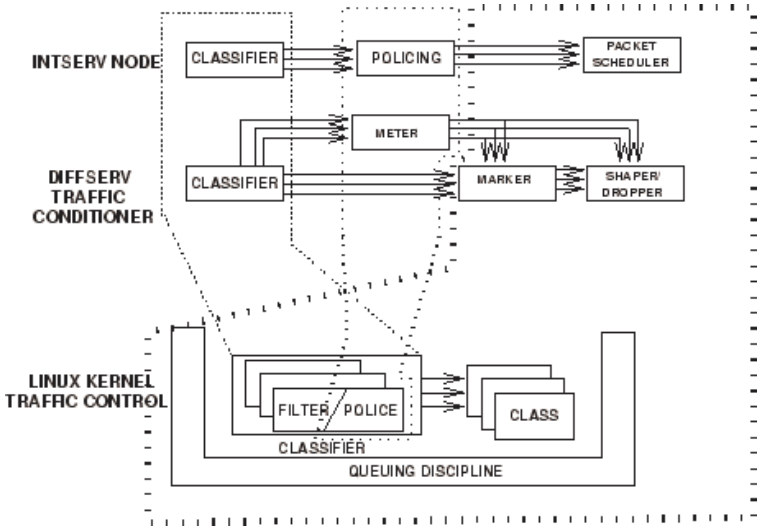


Figura 26: Mapeamento da Arquitectura Diffserv nos elementos de TC do Linux

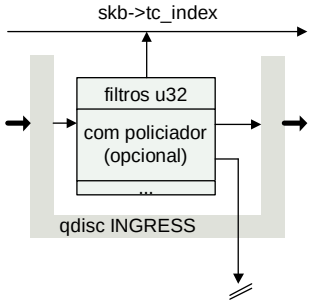


Figura 27: Arquitectura de TC criada pelo DSR nas Interfaces Ingress

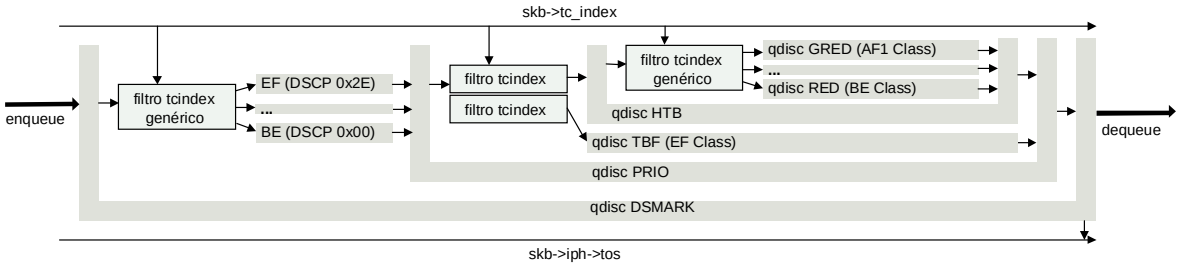


Figura 28: Arquitectura de TC criada pelo DSR nas Interfaces Egress

Anexo 5 Arquitectura dos nós da rede

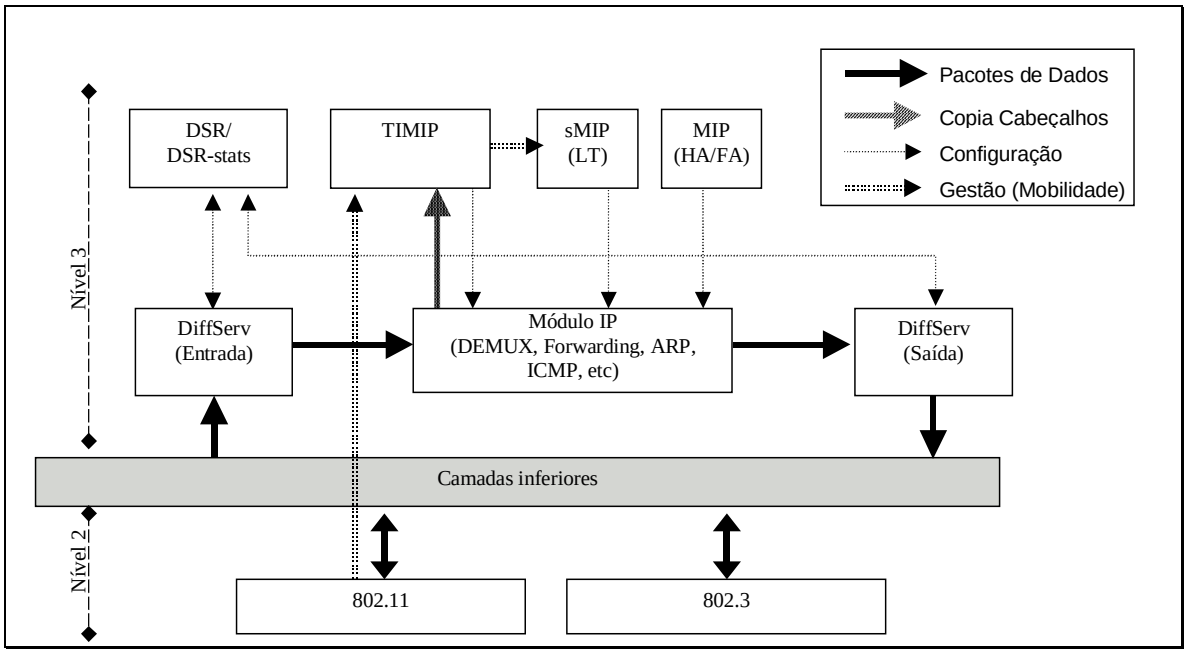


Figura 29: Detalhes da arquitectura dos nós da rede TIMIP

(inclui vida dos pacotes de dados e acções de gestão, e não inclui sinalização de controlo TIMIP/MIP)

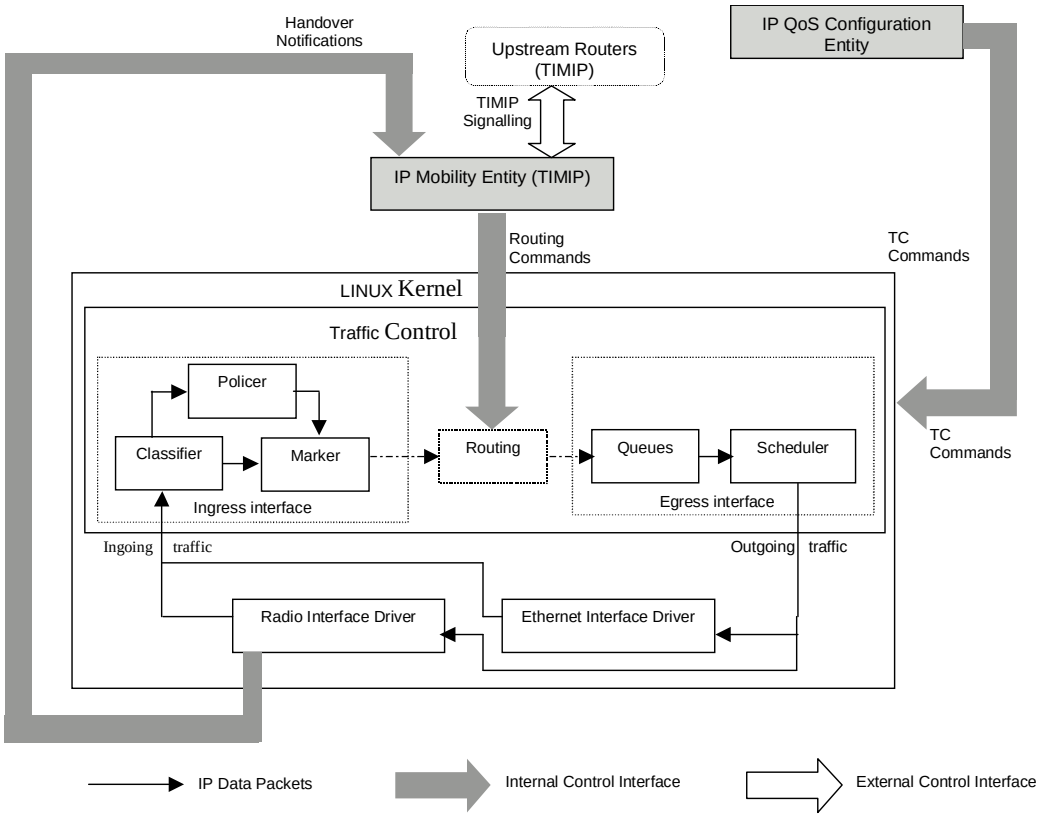


Figura 30: Detalhe da arquitectura dos nós da rede TIMIP (integrado com Diffserv)

Anexo 6 Medições da velocidade do TIMIP

	TESTE: VELOCIDADE DO HANDOVER TIMIP						
							TEMPO TOTAL
Last Packet	ASSOC	First Packet		TEMPO TIMIP		N2+N3	(ajustado)
t0	t1	t2		(First-Assoc)		(First-Last)	
0,728735	0,8202699	0,821986		1,716057716		93	93
0,382186	0,42097	0,423445		2,475036926		41	41
0,996395	0,16065	0,162491		1,840985123		-834	166
0,453759	0,53653	0,538345		1,814982147		84	84
0,7031	0,78423	0,786275		2,04500618		83	83
0,969334	0,04732	0,049759		2,439003643		-920	80
0,907012	0,035236	0,03696		1,723998985		-871	129
0,286822	0,36678	0,368845		2,064957352		82	82
0,399911	0,4887	0,49052		1,819967766		90	90
0,423115	0,50711	0,509695		2,585000343		86	86
0,101889	0,19643	0,198639		2,20903212		96	96
0,632414	0,71447	0,716567		2,096971123		84	84
0,885041	0,9644901	0,966532		2,041943962		81	81
0,343908	0,42329	0,425481		2,190985733		81	81
0,091151	0,17975	0,181783		2,033034332		90	90
0,124574	0,2283601	0,230217		1,856943123		105	105
0,023699	0,12473	0,126768		2,038009041		103	103
0,001434	0,12108	0,122678		1,597959068		121	121
0,610148	0,76709	0,769053		1,963037041		158	158
0,134105	0,25603	0,258638		2,608036507		124	124
0,483099	0,6700799	0,671748		1,668053482		188	188
		Média:		**2,039476272**	ms		103,0952381 ms
		Desvio padrão:		**0,224877254**	ms		25,65079365 ms

Valor máximo de desvio dos relógios de cada nó a uma referência central: 0.500 ms

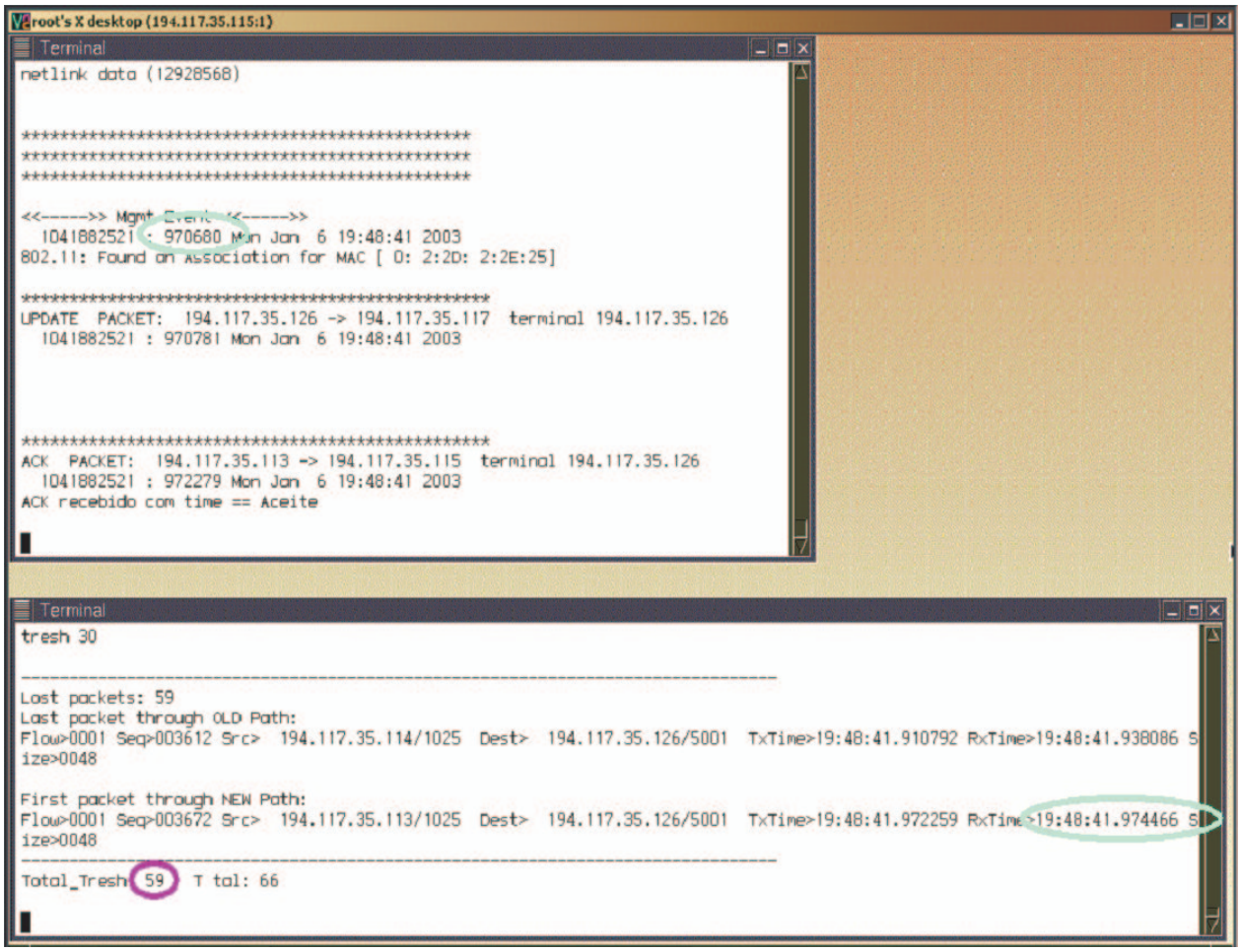


Figura 31: Exemplo de recolha de Dados TIMIP

Valor Roxo	Número de pacotes perdidos (Tempo Total do Handover)
Valores Azul-Claros	Tempo de recepção do primeiro pacote e tempo de recepção da mensagem de Associação (Tempo do TIMIP)

Anexo 7 Medições da velocidade do sMIP+TIMIP

TIPO	FROM	DEST	802.11 ASSOC	TIME		PAC_LOST	TIME_FIRST		assoc	first	timip	timip(ms)
	ap3	ap2	917588	21:41:24		3248	21:41:28	63916				
1	ap3	ap2	942615	21:32:41		3057	21:32:44	967625				
	ap3	ap2	581883	21:40:01		3046	21:40:04	584873				
				MÉDIA tipo 1		3117,00 ms						
	ap2	ap1	471783	21:29:18		43	21:29:18	474673	0,4718	0,474673	0,003	2,89
2	ap2	ap1	214438	21:29:37		37	21:29:37	218424	0,2144	0,218424	0,004	3,986
	ap2	ap1	910719	21:34:04		37	21:34:04	913612	0,9107	0,913612	0,003	2,893
				MÉDIA tipo 2		39,00 ms				MÉDIA TIMIP tipo1		3,26 ms
	ap1	ap2	983791	21:29:25		41	21:29:25	986640	0,9838	0,98664	0,003	2,849
3	ap1	ap2	82539	21:29:47		41	21:29:47	83941	0,0825	0,083941	0,001	1,402
	ap1	ap2	841724	21:31:57		42	21:31:57	843742	0,8417	0,843742	0,002	2,018
				MÉDIA tipo 3		41,33 ms				MÉDIA TIMIP tipo1		2,09 ms
	ap2	ap3	485595	21:30:14		35	21:30:14	492127	0,4856	0,492127	0,007	6,532
4	ap2	ap3	441298	21:32:16		38	21:32:16	445885	0,4413	0,445885	0,005	4,587
	ap2	ap3	198042	21:40:45		42	21:40:45	201427	0,198	0,201427	0,003	3,385
				MÉDIA tipo 4		38,33 ms				MÉDIA TIMIP tipo2		4,83 ms

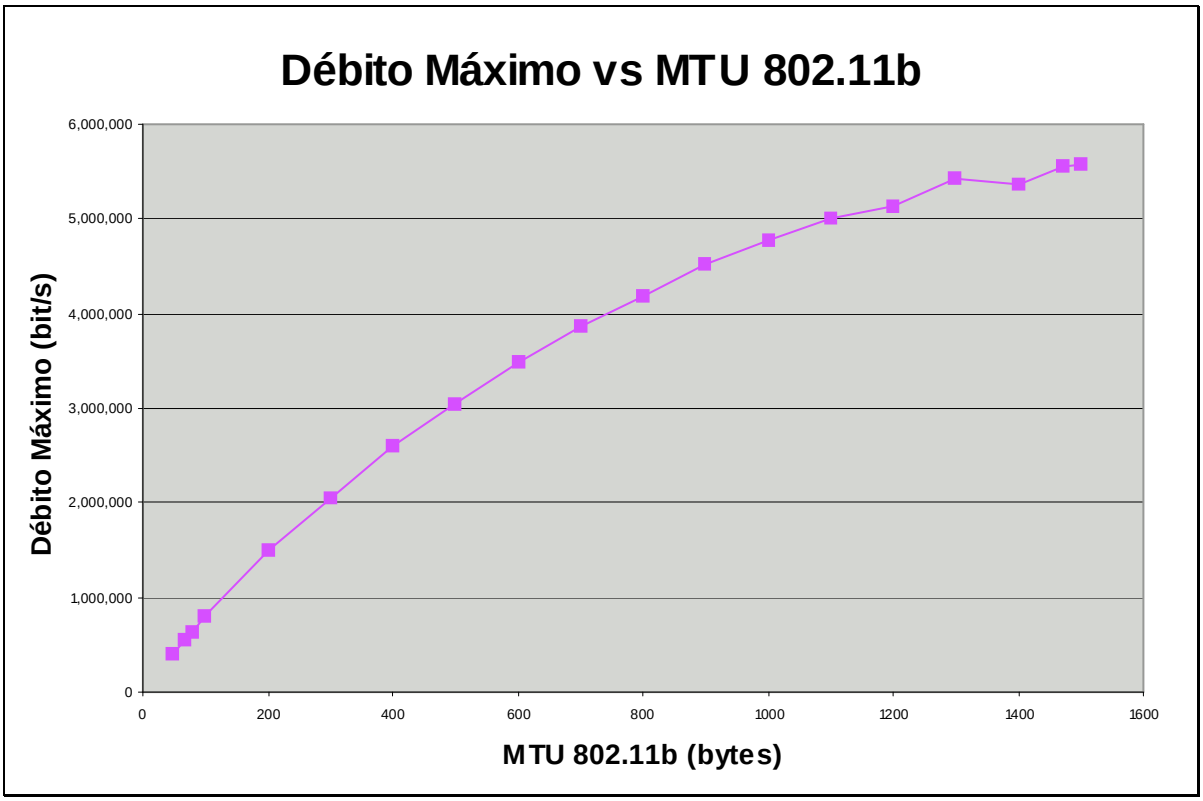
TIPO	Origem	destino	Mobilidade:	Descrição	Latência
1	AP3	AP2	Macro+Micro	entrada num domínio visitado	3117 ms
2	AP2	AP1	Micro	movimentação em domínio visitado	3 ms
3	AP2	AP2	Micro	movimentação em domínio visitado	2 ms
4	AP2	AP3	Macro+Micro	retorno ao domínio de origem	5 ms

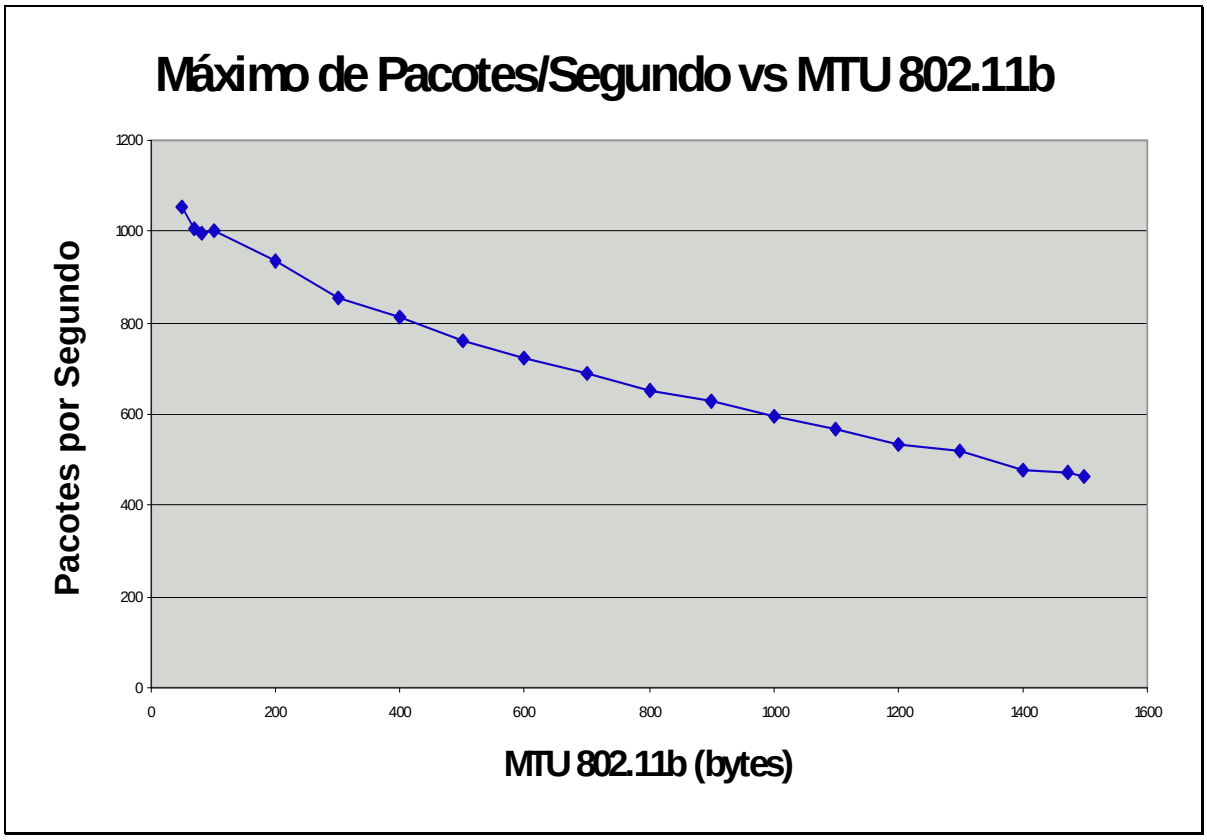
Valor máximo de desvio dos relógios de cada nó relativamente a uma referência central: 0.500 ms

Anexo 8 Medições Práticas do Débito Máximo/ MTU do 802.11b

Tamanho Pacote IP (bytes)	Débito Médio (bit/s)	Pacotes por Segundo
48	404.339	1052
68	548.758	1008
80	638.826	998
100	802.459	1003
200	1.500.410	937
300	2.051.732	854
400	2.603.112	813
500	3.050.142	762
600	3.478.919	724
700	3.864.118	690
800	4.182.089	653
900	4.520.987	627
1000	4.769.577	596
1100	5.008.725	569
1200	5.128.050	534
1300	5.435.140	522
1400	5.359.503	478
1472	5.562.734	472
1500	5.582.651	465

Tabela 3: Medições Práticas do débito máximo do 802.11b

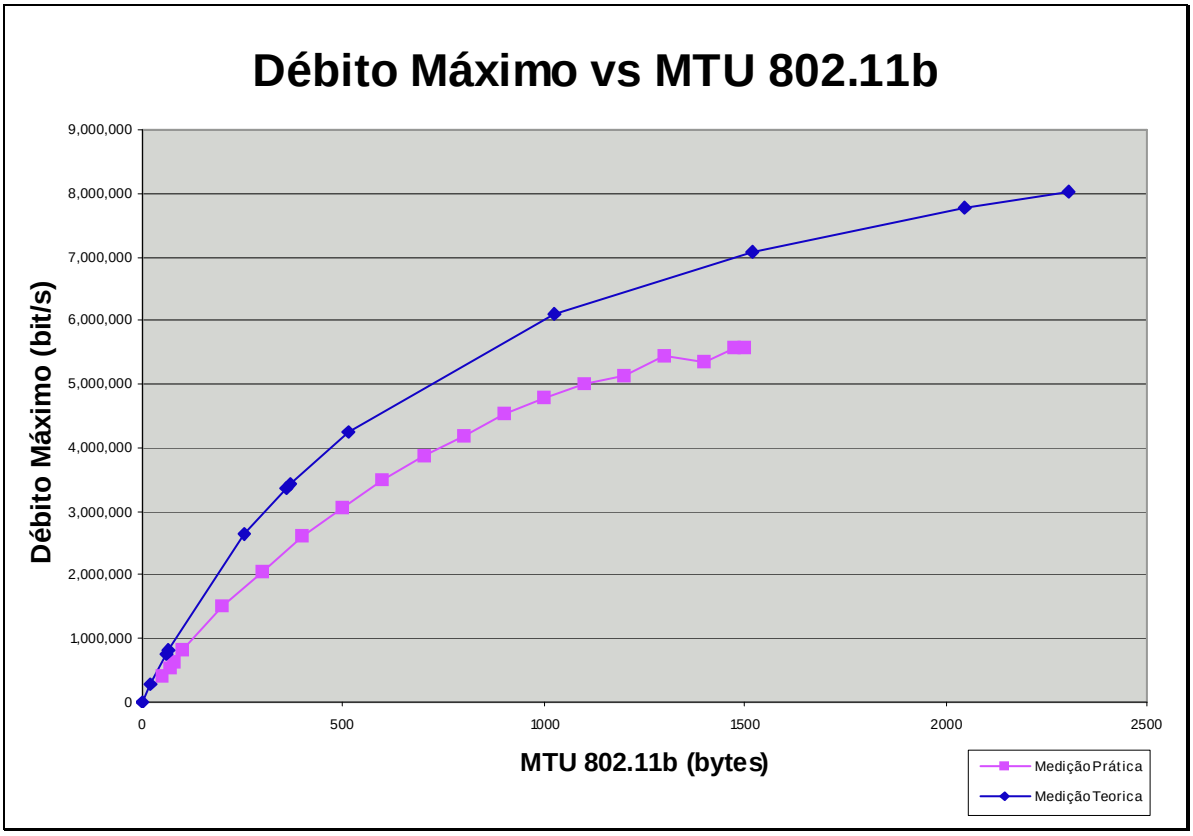

Débito Máximo vs MTU 802.11b
Débito Máximo (bit/s)
6,000,000
5,000,000
4,000,000
3,000,000
2,000,000
1,000,000
0
0
200
400
600
800
1000
1200
1400
1600
MTU 802.11b (bytes)


Máximo de Pacotes/Segundo vs MTU 802.11b
Pacotes por Segundo
1200
1000
800
600
400
200
0
0
200
400
600
800
1000
1200
1400
1600
MTU 802.11b (bytes)

Anexo 9 Comparação do Débito Máximo do 802.11

Tamanho Pacote IP (bytes)	Débito Médio (bit/s)
0	0
20	268000
60	765000
64	812000
256	2650000
357	3370000
368	3440000
512	4250000
1024	6090000
1518	7090000
2048	7780000
2304	8030000

Tabela 4: Medições Práticas do débito máximo do 802.11b



Anexo 10 Teste de Policiamento Diffserv

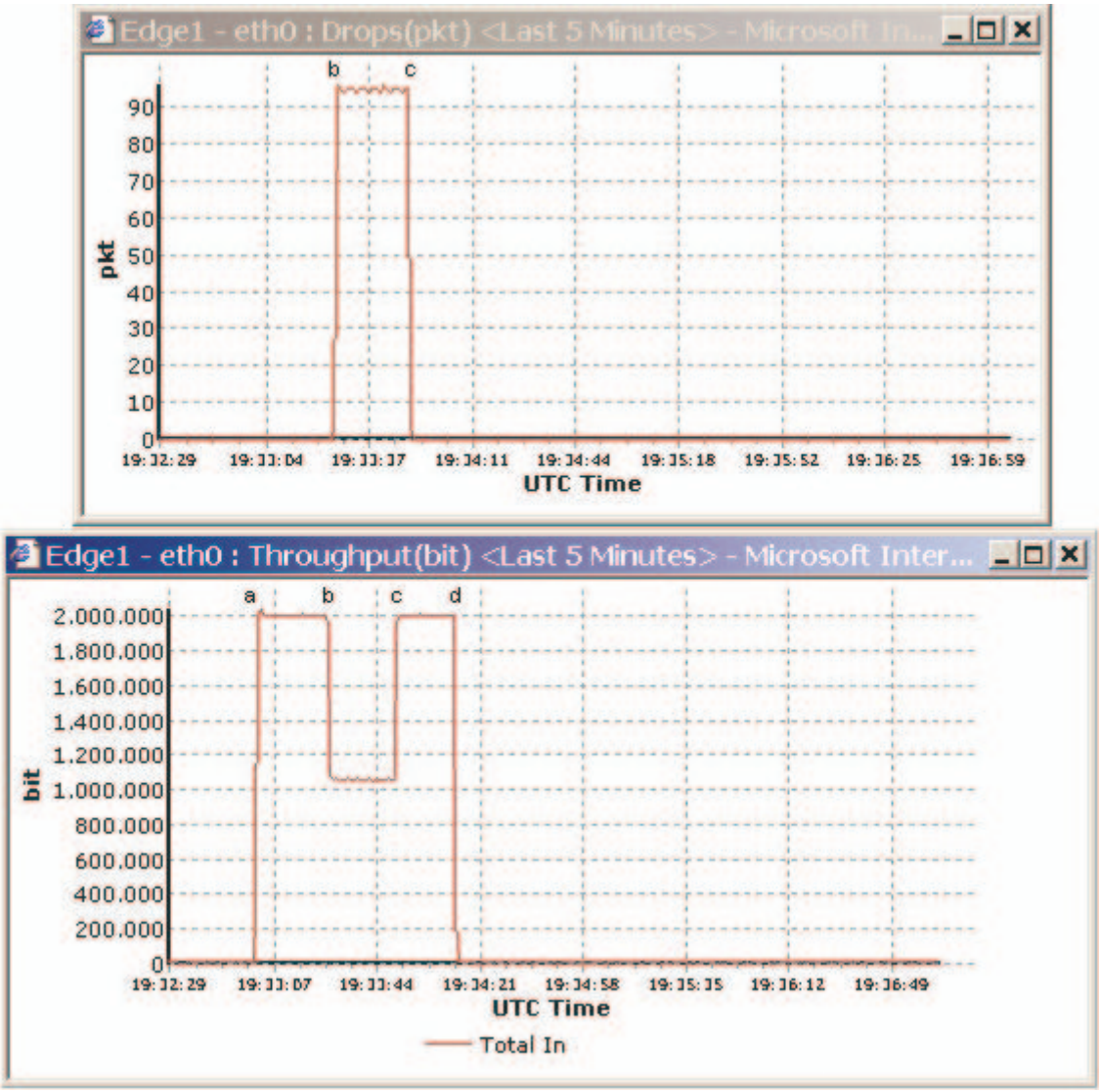


Figura 32: Entrada no domínio DiffServ: pacotes perdidos e débito transmitido

Passo	Descrição	Valor Recebido
a	Início do fluxo de teste de 2 Mbit/s	2 Mbits/s
b	Aplicação de um Policiador de 1 Mbit/s na entrada da rede	~ 1.05 Mbits/s
c	Remoção do Policiador	2 Mbits/s
d	Fim do fluxo de teste	0 Mbits/s

Tabela 5: Teste de Policiamento Diffserv – descrição dos passos

Anexo 11 Teste QoS e Mobilidade em Aplicações Multimédia

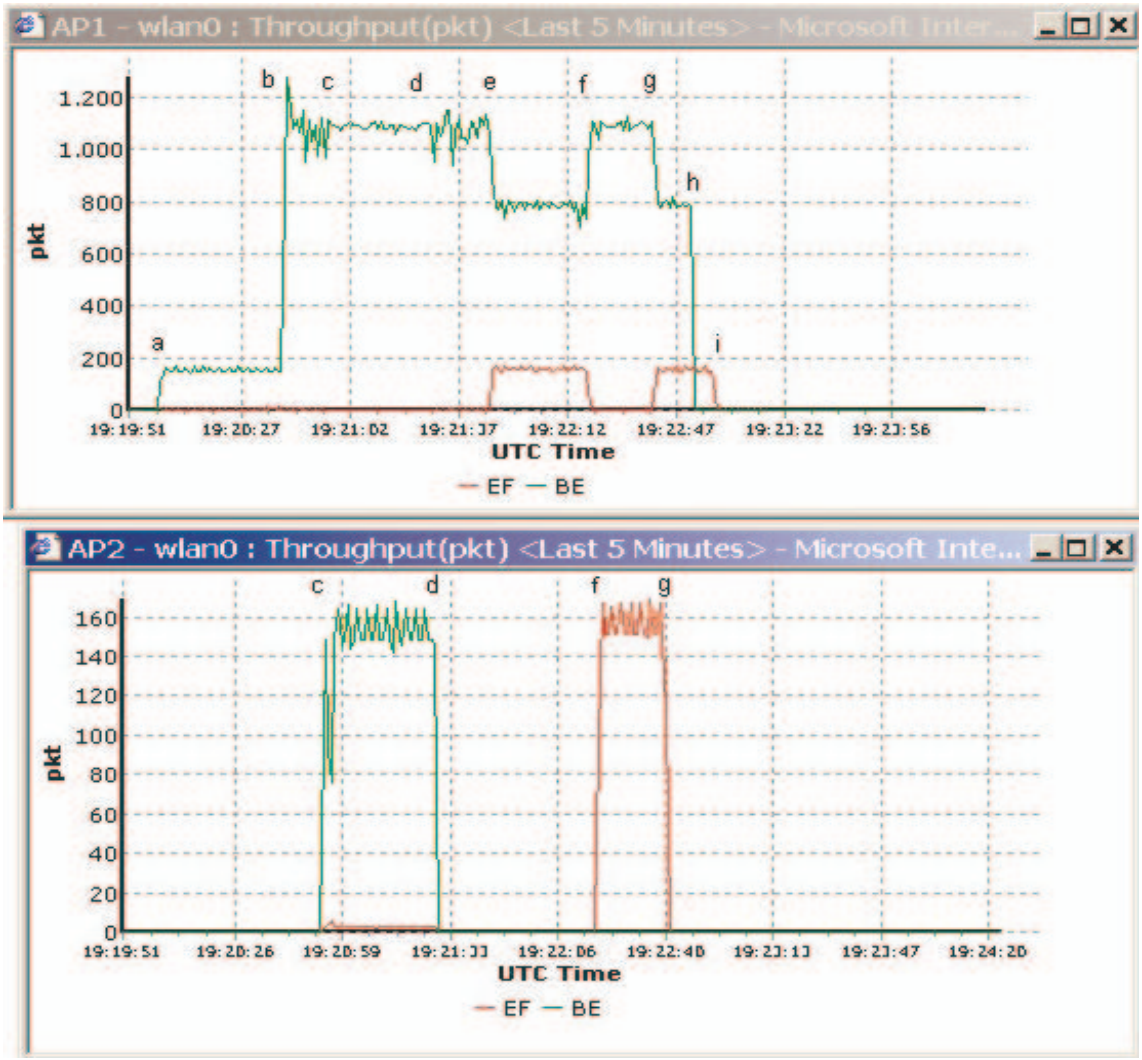


Figura 33: Interface *wireless* 802.11 do AP1 e AP2

Passo	Descrição	Qualidade do Vídeo
a	Início do *streaming* do Vídeo (sem QoS)	Boa
b	Início da Geração da Carga	Má
c	PC1 movimenta-se para o AP2	Boa
d	PC1 volta para o AP1	Má
e	Utilização dos filtros estáticos de QoS	Boa
f	PC1 movimenta-se para o AP2	Boa
g	PC1 volta para o AP1	Boa
h	Fim da Carga	Boa
I	Fim do Vídeo	Boa

Tabela 6: Teste de performance com aplicações multimédia – descrição dos passos

Bibliografia

Referências de organismos de Normalização

[1] Internet Engineering Task Force, www.ietf.org

[2] Institute for Electric and Electronics Engineering, www.ieee.org

[3] Mobile IP Charter, http://www.ietf.org/html.charters/mobileip-charter.html

[4] 802.11 Group, http://grouper.ieee.org/groups/802/11/index.html

Referências relativas a protocolos de mobilidade IP, e suporte IP

[5] C. Perkins, ed., "IP Mobility Support for IPv4" (Proposed Standard), IETF RFC 3320, Janeiro 2002, www.ietf.com

[6] J. Kristoff, “Mobile IP”, http://condor.depaul.edu/~jkristof/mobileip.html

[7] A. Campbell, et al, "Comparison of IP MicroMobility Protocols”, IEEE Wireless Communications, Fevereiro 2002.

[8] D. Plummer, “An Ethernet Address Resolution Protocol”, RFC 826, MIT-LCS, 1982

[9] Mills, D., "Network Time Protocol (Version 3) Specification, Implementation", RFC 1305, Março 1992.

Referências relativas a Suporte de Qualidade de Serviço

[10] S. Blake, ed., “An Architecture for Differentiated Services”, RFC 2475, Dec 1998.

[11] K. Nichols, ed., “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”, RFC 2474, Dezembro 1998

[12] B. Davie, ed., “An Expedited Forwarding PHB (Per-Hop Behavior)”, RFC 3246, Março 2002

[13] J. Heinanen, ed., “Assured Forwarding PHB group”, RFC 2597, June 1999

[14] B. Braden, ed., “Recommendations on Queue Management and Congestion Avoidance in the Internet.”, RFC 2309, Abril 1998.

[15] P Fergunson, D. Senie, “Integrated Services in the Internet Architecture: an Overview”, RFC 1633, Junho 1994.

[16] S. Floyd, V. Jacobson, “Link-sharing and Resource Management models for packet networks”, IEEE/ACM Transactions on Networking 3(4), 1995

Referências relativas ao 802.11

[17] IEEE, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, IEEE Std. 802.11, 1997. http://standards.ieee.org/getieee802/download/802.11-1999.pdf

[18] IEEE, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher Speed Physical Layer Extension in the 2.4 Ghz Band”, IEEE Std. 802.11b, 1999. http://standards.ieee.org/getieee802/download/802.11b-1999.pdf

[19] IEEE Std. 802.11f/D3, “Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation”, (Draft Supplement to IEEE Std 802.11, 1999 Edition), January 2002

[20] A. Grilo, M. Macedo, M. Nunes, “Service Disciplines for Support of Inteserv and Diffserv in IEEE 802.11 Access networks”, draft paper, 2001

Referências genéricas de redes e sistemas distribuidos

[21] S. A. Tanenbaum – “Computer Networks”, 3nd Edition, Prentice Hall International, 1996

[22] S. Keshav, S; “An Engineering Approach to Computer Networking, ATM Networks, The Internet and the Telephone Network”, AT&T Research, Addison-Wesley Publishing, 1997.

[23] Alves Marques, Paulo Guedes, "Tecnologia de Sistemas Distribuídos", FCA editora, 1998

Referências genéricas do Sistema Operativo Linux

[24] Matt Welsh, "Linux Installation and Getting Started" , Linux Documentation Project, 1996

[25] Matt Welsh, Lar Kaufman, "Running Linux", O'Reilly & Associates inc., 1995

[26] Lars Wirzenius, "Linux System Administrators' Guide 0.6", Linux Documentation Project, 1997

[27] L. Torvalds, “Linux kernel release 2.4.xx REAME”, incluído no kernel, Outubro 2001

Referências relativas à componente de Rede do Linux

[28] G. Dhandapani, A. Sundaresan, “Netlink Sockets – Overview”, Information and Telecommunications Technology Center, Department of Electrical Engineering & Computer Science, The University of Kansas, 1999, http://qos.ittc.ukans.edu/netlink/html/index.html

[29] G. Herrin, “Linux IP Networking - A Guide to the Implementation and Modification of the Linux Protocol Stack”, Maio 2000, http://kernelnewbies.org/documents/ipnetworking/linuxipnetworking.html

[30] Olaf Kirch, "The Network Administrators' Guide", Linux Documentation Project, 1996

[31] Terry Dawson, Linux NET-3-HOWTO (Linux Networking), Linux Documentation Project, 1998

[32] Paul Gortmaker, Linux Ethernet-Howto, Linux Documentation Project, 1998

[33] A. N. Kuznetsov, "IP Command Reference", IPRoute2 Documentation, 1999

[34] A. N. Kuznetsov, "Tunnels over IP in Linux-2.2", IPRoute2 Documentation, 1999

[35] Bob Edwards, “Proxy ARP Subnetting HOWTO”, Linux Documentation Project, 1997

[36] Alessandro Rubini, "Linux Device Drivers", O'Reilly & Associates inc., 1998

[37] Fred N. van Kempen, "ARP Man Page", Linux Programmer's Manual, net-tools Documentation, 1999

[38] Fred N. van Kempen, "Ifconfig Man Page", Linux Programmer's Manual, net-tools Documentation, 1997

[39] Phil Blundell, "Route Man page”, Linux Programmer's Manual, net-tools Documentation, 1997

[40] T. Carstens, “Programming with pcap”, http://www.tcpdump.org/pcap.htm, 2001

Referências relativas ao controlo de tráfego em linux

[41] W. Almesberger, “Linux Network Traffic Control - Implementation Overview (kernel 2.4)", Fevereiro 2001, ftp://icaftp.epfl.ch/pub/people/almesber/junk/tc-04FEB2001-0.ps.gz

[42] W. Almesberger, et al, "Differentiated Services on Linux", June 1999 (draft-almesberger-wajhak-diffserv-linux-01.txt) ftp://icaftp.epfl.ch/pub/linux/diffserv/misc/dsid-01.ps.gz, http://diffserv.sourceforge.net

[43] S. Radhakrishnan, “Linux - Advanced Networking Overview”, Information and Telecommunications Technology, Center Department of Electrical Engineering & Computer Science, The University of Kansas Lawrence, 1999, http://qos.ittc.ukans.edu/howto/index.html

[44] B. Hubert , et al, “Linux Advanced Routing & Traffic Control HOWTO”, Setembro 2000, http://lartc.org/howto/

[45] M. Devera, “HTB Linux queuing discipline manual”, http://luxik.cdi.cz/~devik/qos/htb/, Fevereiro 2002

[46] Pedro Catelas, José António Neves, “Monitorização de Qualidade de Serviço em redes IP com Serviços Diferenciados”, Relatório de Trabalho Final de Curso, Setembro 2002

[47] Rui Prior, “Qualidade de Serviço em Redes de Comutação de Pacotes”, Março 2001, http://telecom.inescn.pt/doc/msc/rprior2001_pt.html

[48] K. Wagner, “Short Evaluation of Linux's Token-Bucket-Filter (TBF) Queueing Discipline”, http://www.docum.org/stef.coene/qos/docs/other/tbf02_kw.ps, Maio 2001

Referências relativas aos Testes

[49] Rick Jones, "Network Performance Home Page" http://www.netperf.org/netperf/NetperfPage.html

[50] B. Adamson, Naval Research Laboratory (NRL): "Multi-Generator (MGEN) Toolset", Version 3.1, http://manimac.itd.nrl.navy.mil/MGEN/

[51] G. Java, "IPTraf User's Manual", Maio 2002, http://cebu.mozcom.com/riker/iptraf/2.7/manual.html

[52] Tcpdump group, Tcpdump Users Manual, http://www.tcpdump.org/tcpdump_man.html, 2002

[53] Richard Sharpe, Ed Warnicke, "Etheral User's Manual", http://www.ethereal.com/docs/user-guide/, 2002

[54] Moicane Deliverable D15, "Applications Documentation", D15/ICCS/WP4/V1.0, Março 2002.

Referências relativas ao hardware 802.11 (Chipset Prism)

[55] Intersil, "PRISM Driver Programmer's Manual v2.0". Intersil Restricted Distribution.

[56] Intersil, "Product Development Software Release Form – Cw10 Tertiary Firmware" Intersil Restricted Distribution.

[57] J. Malinen, "Host AP driver for Intersil Prism2", http://hostap.epitest.fi/, Setembro 2002

[58] J. Malinen, "Host AP ChangeLog", http://hostap.epitest.fi/cgi-bin/viewcvs.cgi/*checkout*/hostap/ChangeLog?rev=HEAD&content-type=text/plain, Setembro 2002

[59] AbsoluteValue Systems, "Linux-wlan Project", ftp://ftp.linux-wlan.org/pub/linux-wlan-ng/, Setembro 2002

[60] P. Miranda, R. Nunes, "Suporte de Qualidade de Serviço em Redes Sem Fios 802.11b", Relatório de Trabalho Final de Curso, Setembro 2000

[61] Lucent Technologies, "Roaming with WaveLAN/IEEE 802.11", WaveLAN Technical Bulletin 021/A, Decembro 1998

[62] D. Hinds, "Linux PCMCIA HOWTO", http://pcmcia-cs.sourceforge.net/

[63] J. Tourrilhes, "Linux Wireless LAN Howto", http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Linux.Wireless.html

[64] J. Tourrilhes, "Wireless Extensions for Linux", http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Linux.Wireless.Extensions.html

Referências relativas a Trabalhos nesta área do Autor

[65] P. Estrela, A. Grilo, T. Vazão e M. Nunes, "Terminal Independent Mobile IP (TIMIP)", Internet draft, draft-estrela-timip-00.txt, work in progress, Março 2002.

[66] P. Estrela, “Estudo e Implementação da Macro-Mobilidade na Internet”, Relatório de Trabalho Final de Curso (Regime Integrado), Setembro 2000

[67] P. Estrela, “Mobilidade em IP – “Estado da Arte ”, Trabalho Final da Cadeira de Tópicos Avançados em Conectividade e Sistemas Distribuídos, Abril 2002

[68] P. Estrela, “Rede de acesso do MOICANE”, Trabalho Final da Cadeira de Redes de Acesso Multi Serviço, Fevereiro 2002

[69] P. Estrela, “Protocolos de Mobilidade para Terminais IP”, Dissertação de Mestrado, Fevereiro 2003

Glossário

Estrangeirismo	Tradução
(tráfego) downlink/ downstream	(tráfego) no sentido descendente, *para* os terminais
(tráfego) uplink/ upstream	(tráfego) no sentido descendente, *dos* terminais
“early drop”	perda antecipada
access point	ponto de acesso
Acknowledge	mensagem de confirmação
Beacon	sonda
border router	encaminhador fronteira
Bridge	encaminhador de nível 2
Crossover	primeiro nó comum entre os dois caminhos envolvidos num handover, desde os terminais até ao topo da árvore de nós.
default router	encaminhador por omissão
Digest	resumo de dados calculado por processos de hashing
driver	controlador
edge router	encaminhador fronteira
gateway	encaminhador central em domínios TIMIP
handover	(processo de) transição de terminais
handshake	processo de estabelecimento de uma ligação
hashing	
host	terminal IP ou encaminhador IP
legacy	legado
management	gestão
mobile host	terminal móvel
overhead	ineficiência/peso de protocolos de comunicação
performance	desempenho
poll	autorizações explícitas de acesso ao meio em 802.11 por parte do AP
premium	(tratamento) distinguido
proxy	(entidade) que efectua qualquer acção em nome de outra, sendo

	esta pedida explicitamente
rate	ritmo/débito
roaming	transição
router	encaminhador de nível 3
routing	encaminhamento
scheduler	calendarizador/escalonador
soft state	característica do *estado* que é automaticamente eliminado se não for utilizado
stack	pilha (de protocolos)
standard	normalizado
surrogate	(entidade) que efectua qualquer acção em nome de outra, sem que tal seja pedida tanto implícita como explicitamente
switches	encaminhador de nível 2 entre tecnologias de rede iguais
timeout	terminação do limite de tempo
timestamp	marcação temporal
video on demand	vídeo a pedido

Abreviatura	Significado
802.11	Wireless LAN, WaveLan
802.3	IEEE 802.3 (Ethernet)
AF	Assured Forwarding
ANG	Access Network Gateway
AP	Access Point
API	Application Programming Interface
ARP	Address Resolution Protocol
BB	Bandwidth Broker
BE	Best Effort
BR	Border Router
BS	Base Station
CBQ	Class Based Queuing
CIDR	Classless Inter Domain Routing
CIP	Cellular IP
CL	Controlled Load
COPS	Common Open Policy Server
CRC	Cyclic Redundancy Check
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
DIFFSERV	Differentiated Services
DSCP	Differentiated Services Code Point
EF	Expedited Forwarding
ETSI	European Telecommunications Standards Institute
FA	Foreign Agent
FCS	Frame Check Sequence
FIFO	First in First out
FTP	File Transfer Protocol
GW	Gateway
HA	Home Agent
HAWAII	Handoff-Aware Wireless Access Internet Infrastructure
HDRR	Home Domain Root Router
ICMP	Internet Control Message Protocol
IEEE	Institute for Electric and Electronics Engineering

IETF	Internet Engineering Task Force
INESC	Instituto Engenharia de Sistemas e Computadores
INESC	Institute of Engineering of Systems and Computers
INOV	INESC Inovação
INTSERV	Integrated Services
IP	Internet Protocol
ISO	International Standards Organization
ISP	Internet Service Provider
Kbit/s	Kilobits por segundo
LAN	Local Area Network
LLC	Logical Link Control
LOS	Line of Sight
LT	Legacy Terminal
MAC	Medium Access Control
Mbit/s	Megabits por segundo
MD5	Message Digest n5
MIB	Management Information Base
MIP	Mobile IP
MOICANE	Multiple Organisation Interconnection for Collaborative Advanced Network Experiments
MT	Mobile Terminal
MTU	Maximum Transfer Unit
NE	Network Element
NT	Network Termination
NTP	Network Time Protocol
OSI	Open Standards Initiative
PC	Personnel Computer
PHB	Per Hop Behaviour
POTS	Plain Old Telephone Service
PSTN	Public Switched Telephone Network
QoS	Quality of Service
QoS	Quality of Service
RED	Random Early Detection
RF	Radio Frequency

RFC	Request for Comments
RSVP	Resource Reservation Protocol
RTCP	Real-Time Control Protocol
RTP	Real Time Transport Protocol
SLA	Service Level Agreement
SNR	Signal to Noise Ratio
TCP	Transmission Control Protocol
TIMIP	Terminal Independent Mobility for IP
UDP	User Datagram Protocol
VoD	Video on Demand
VoIP	Voice over Internet Protocol
WAN	Wide Area Network
WDS	Wireless Distribution System
WWW	World Wide Web