

## Aspectos de Medição de Qualidade de Serviço em Redes Móveis baseadas em IP

Nuno Inácio, Ricardo Azevedo, Rui L. Aguiar, Francisco Fontes

**Resumo** - Os aspectos de Qualidade de Serviço (Quality of Service – QoS) em redes avançadas aparecem como uma área de trabalho interessante, dada a complexidade de que se reveste o planeamento e controlo de redes IP com QoS, associadas a ambientes móveis.

Esta comunicação aborda problemas de medição em redes heterogéneas, em particular medidas relevantes para QoS, essenciais tanto para avaliação de desempenho, como para mecanismos de controlo. São discutidas duas ferramentas distintas: uma ferramenta de medição do estado de ligações TCP e um elemento de controlo de um sistema de medição baseado em pontas de prova (probes) IP. As ferramentas foram desenvolvidas e testadas com recurso à rede Moby Dick, obtendo-se bons resultados.

**Abstract** – Quality of Service (QoS) aspects are one of the most interesting areas of work in new advanced networks, due to the complexity of planning and controlling IP networks with QoS, especially when associated with mobile environments.

This paper discusses measuring problems in heterogeneous networks, particularly relevant measures for QoS. It covers two distinct tools: one for measurements of TCP connections; and the other a control element for a probe based system. The tools were developed and tested with the Moby Dick network with good results.

### I. INTRODUÇÃO

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

A implementação de QoS na Internet é uma área muito activa nos últimos anos e recentemente tem sido centrada em ambientes de mobilidade. Exemplo disso é o projecto Moby Dick [MOBYWEB, MOBY], que aborda, propondo

e demonstrando soluções, os temas de qualidade de serviço e autorização e autenticação de acesso. Esta rede é baseada no modelo de referência de serviços diferenciados, associado a mecanismos eficientes de mobilidade e a mecanismos de AAA (autorização, autenticação e contabilização), baseados no protocolo DIAMETER.

Este documento apresenta os resultados de um trabalho final de curso que aborda esta área. O documento começa por uma introdução a aspectos de medidas para o protocolo TCP na secção II e descreve uma ferramenta dedicada a análise de comportamentos deste protocolo durante *handovers* em redes móveis. A secção III aborda os conceitos de medidas baseadas em *probes* e por fim, a secção IV discute a implementação de um elemento de controlo para um sistema de medição baseado em *probes*. A secção V apresenta as principais conclusões retiradas do trabalho.

### II. MEDIDAS DE DESEMPENHO DO TCP EM AMBIENTES MÓVEIS

O TCP (*Transmission Control Protocol*) é um protocolo responsável pelo transporte de dados fim-a-fim. O TCP é um protocolo orientado à conexão, e tem mecanismos que protegem contra a perda de pacotes, a sua duplicação, a sua desordenação na entrega ou ainda a existência de dados corruptos. Este protocolo implementa ainda vários algoritmos para o controlo do tráfego na rede e para o reenvio de pacotes no caso destes serem perdidos em trânsito.

#### A Otenção de variáveis internas do TCP

Sendo o TCP um protocolo orientado à conexão, uma implementação do protocolo tem de guardar um grande conjunto de informação. Na implementação do protocolo em Linux acontece exactamente isso, sendo guardados vários valores por cada ligação TCP existente. A obtenção desses valores, geralmente internos ao *kernel*, pode ser de grande importância para se perceber o comportamento de determinada implementação num determinado cenário. A obtenção de valores para a janela de congestionamento poderá mostrar-nos como a performance do TCP se comporta durante determinados processos de rede – entre

os quais é de particular interesse o *handover*, a situação de mudança de ponto de ligação do terminal à rede.

Existem várias ferramentas para analisar o comportamento do TCP, sendo as mais interessantes o TBIT, o WEB100 e o Sockmon.

O TBIT [1] é uma ferramenta desenvolvida pelo “AT&T Center for Internet Research” que tem como objectivo caracterizar o comportamento do TCP num servidor remoto. Mais especificamente, esta ferramenta permite obter aproximações dos valores da *cwnd* (*Congestion Window*, um parâmetro importante para o desempenho do TCP), qual a versão do protocolo TCP utilizada, entre outras medidas. Com o uso desta ferramenta é ainda possível a detecção de erros de implementação do TCP no sistema operativo do servidor em questão.

WEB100 [2] é um projecto que tem como objectivo desenvolver uma plataforma onde possam ser executadas aplicações *web* que utilizem 100% da largura de banda disponível, seja ela qual for. De uma forma mais geral, o projecto tem como objectivo fazer com que qualquer máquina possa ter acesso a toda a largura de banda disponível sem a ajuda de peritos para as configurações. Com a utilização deste *software* o sistema operativo (no caso o Linux) conseguirá utilizar a *stack* TCP a tempo real, sem a intervenção por parte do utilizador, de forma a que a maior largura de banda possível seja utilizada na(s) ligação(ões). Neste momento existe apenas uma versão do projecto para Linux, mas é esperado que os seus promotores desenvolvam uma versão genérica que possa ser utilizada em qualquer sistema operativo. Como se trata de um projecto com uma duração de três anos, encontrando-se a meio na altura da realização do trabalho, estão ainda muitos dos objectivos por implementar.

O WEB100 é dividido em duas componentes distintas. A primeira componente é um *patch* ao *kernel* que é responsável por colocar instrumentos de medida, recolher todas as informações necessárias e colocá-las visíveis para posterior apresentação ou alteração. A segunda componente é uma biblioteca que tem como objectivo ler e escrever para as variáveis que o *patch* coloca visíveis. Com a utilização da referida biblioteca é possível a qualquer interessado desenvolver uma aplicação que leia e apresente as variáveis que o *patch* colocou visíveis. A comunicação entre o *kernel space* e *user space* é feita através do sistema de ficheiros */proc*. É através da escrita, por parte do *kernel*, e da leitura, por parte da biblioteca, no */proc* que ambos os componentes de *software* comunicam. Assim é possível a um utilizador alterar um determinado valor através da escrita no ficheiro certo do */proc*.

Os tipos de instrumentos disponíveis vão desde a medição de estatísticas sobre o tráfego que está a sair referente a uma determinada ligação TCP, o estado da ligação, informação e controle sobre os vários *buffers* de saída, RTT, congestionamento do emissor, etc.

Esta ferramenta mostra-se de grande interesse para a obtenção/observação e “afinamento” (*tuning*) das várias variáveis envolvidas no protocolo TCP. No entanto não foi possível utilizá-la na rede Moby Dick, a rede de teste disponível, pois esta baseia-se no *kernel* Linux 2.4.16, para o qual o *patch* WEB100 não funciona correctamente.

O Sockmon [3] é um módulo para *kernel* Linux 2.2.x e 2.4.x que permite a monitorização em tempo real de informação respeitante a todas as ligações TCP existentes, isto é, todas as ligações que se encontrem no estado ESTABLISHED. Também esta ferramenta faz uso do */proc* para mostrar as informações medidas no *kernel*. Uma vantagem deste método em relação ao utilizado no projecto Web100 tem a ver com o facto do Sockmon ser um módulo que é carregado e não um *patch* ao *kernel*. Para a utilização do Sockmon não é necessário a instalação que qualquer *patch*, sendo apenas necessário escolher, durante a configuração para a compilação do *kernel*, uma opção específica. Depois de recompilado o *kernel*, basta carregar o módulo e verificar em “*/proc/net/sockmon*” o valor das variáveis extraídas do *kernel*. Este módulo é, no entanto, incompatível com o *kernel* do Moby Dick (especificamente com o Swamp, que combina os *patches* de Mobile IPv6 e IPSec).

O facto do Sockmon não funcionar no *kernel* do Moby Dick levou-nos ao desenvolvimento de um *patch* específico para o *kernel* 2.4.16. Esse *patch* tem basicamente a mesma funcionalidade do Sockmon, permitindo reproduzir as suas facilidades dentro da rede de testes existentes no Instituto de Telecomunicações [8].

## B Medidas de handover na rede Moby Dick

O *patch* desenvolvido foi utilizado para executar medidas da *cwnd* na rede Moby Dick. Os testes consistiam na realização de *handovers* entre dois Access Routers (AR) da rede Moby Dick, um AR *wireless* e um AR *Ethernet*.

O primeiro teste realizado foi um *handover* de tecnologia *wireless* para *ethernet*. Foi iniciada uma ligação TCP com o *Mobile Node* (MN) a enviar os pacotes para um *router wireless* e posteriormente foi ligado um cabo *ethernet* ao MN passando este a enviar pacotes pela *ethernet*.

A imagem seguinte mostra um gráfico da janela de congestionamento da ligação TCP referida a cima.

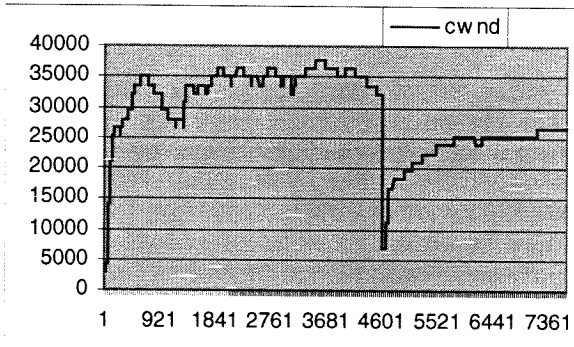


Figura 1 – janela de congestionamento num handover wlan->eth

Fig

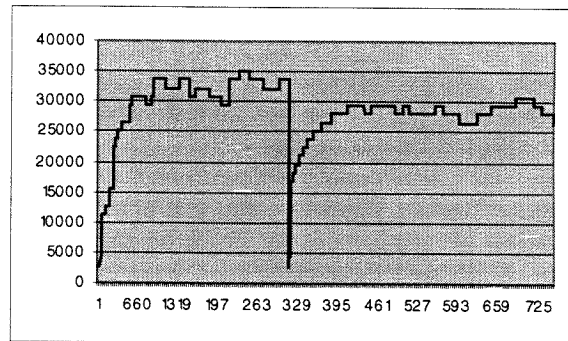


Figura 3 – janela de congestionamento num handover eth->wlan

Figura

Como se pode verificar existe uma quebra no valor da janela, reflectindo a diminuição da velocidade de ligação. A imagem seguinte mostra em pormenor a quebra na janela. É importante verificar que embora a ligação TCP sofra uma redução abrupta da janela, vendo com mais atenção reparamos que existem valores intermédios, isto é, o valor não cai para um valor mínimo instantaneamente.

É iniciada uma ligação TCP com o MN ligado à ethernet; durante a ligação é retirado o cabo da ethernet e o MN começa a enviar pacotes através da interface wireless. Como se pode verificar na imagem acima existe uma queda abrupta no valor da janela de congestionamento. Uma nova imagem, desta vez mais específica da queda abrupta do valor da janela, mostra-nos que ao contrário do que se passou no handover anterior, o valor da janela cai repentinamente para um valor mínimo, o que implica a existência de um *time-out*.

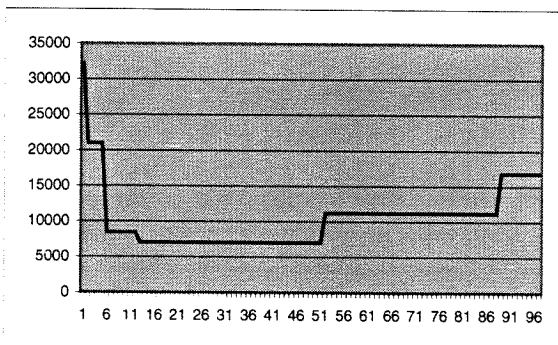


Figura 2 – pormenor da janela de congestionamento

Fig

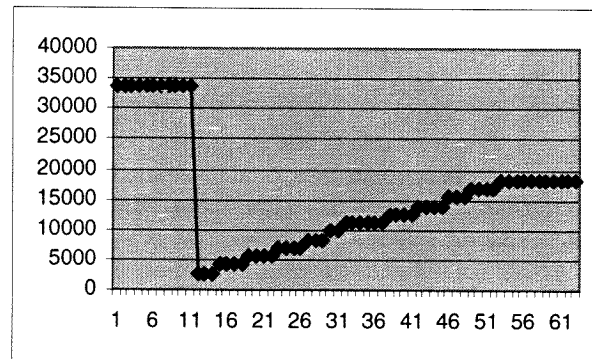


Figura 4 – pormenor da janela de congestionamento

Fi

Vimos a constatar que este comportamento é explicado pelo facto de o sistema operativo do MN começar a receber pacotes duplicados, uns vindos do AR *ethernet* outros vindos do AR *wireless*.

Num segundo teste realizou-se um *handover* da tecnologia *ethernet* para *wireless*. A Figura 3 apresenta um gráfico do valor da janela de congestionamento durante o *handover*.

Tal acontece pois o MN demora algum tempo a aperceber-se que o cabo *ethernet* foi retirado; esse intervalo de tempo é suficientemente grande para que aconteça um *time-out* na ligação TCP. De seguida, depois de se aperceber da inexistência de ligação física pela rede *ethernet*, o MN começa a utilizar a interface *wireless*.

### III MEDIDAS BASEADAS EM PROBES

O IETF tem desenvolvido nos últimos anos, através do IPPM WG (*IP Performance Metrics Working Group*) [6], vários documentos onde são definidas, conceptualmente, as várias métricas e metodologias de medição. A lista de métricas especificada pelo IPPM WG é algo extensa, e é provável que seja ainda alterada, visto alguns documentos não estarem ainda encerrados.

O IPPM WG divide as métricas em 3 tipos:

**singleton**, são medidas atómicas, as quais correspondem a valores de uma medição, por exemplo o atraso de um pacote;

**samples**, referem-se a métricas que derivam das *singleton*. São calculadas utilizando um certo número de instâncias das *singletons*;

**statistic**, são estatísticas feitas sobre as métricas *samples*.

As mais importantes a referir são portanto as *singleton*, que se dividem também elas em dois tipos: *One-way*, que são medidas unidireccionais entre dois pontos, isto é, as que se compõem da ida de um pacote (ou conjunto de pacotes), de uma origem até um destino; e *Round-trip* que são medidas bidireccionais, isto é, as que se compõem da ida de um pacote (ou conjunto de pacotes) de uma origem até um destino e volta.

É importante salientar que no caso da utilização de medidas *Round-trip* não existe qualquer certeza de que o caminho de ida é o mesmo que o caminho de volta; não há sequer garantias que exista um caminho de volta estável.

O principal parâmetro na definição de métricas é o tipo de pacote com o qual se realizam as medidas; por exemplo a verificação de conectividade IP pode ter vários resultados consoante os pacotes de teste utilizados, especialmente se estivermos na presença de *firewalls*.

Devido às possíveis diferenças obtidas entre testes com a mesma finalidade, o IPPM WG definiu o conceito abstracto de "*Type-P packets*" (Pacotes de Tipo-P) [7].

Assim, os pacotes utilizados para efectuar as medidas, sejam elas de que tipo forem são identificados como *Type-P packets* que, depois de especificados, identificam inequivocamente uma determinada medida.

As métricas *singleton* englobam a possibilidade da verificação de conectividade IP, o cálculo de latências, perdas e débito. Com a aplicação de métodos estatísticos às métricas *singleton* é possível obter métricas derivadas: médias, *jitter*, lista de distâncias, mínimos, máximos, etc..

A realização das medidas acima referidas, pode ser feita através de medidas passivas ou activas.

### A Monitorização Passiva

As medidas passivas utilizam o tráfego existente na rede. Como nenhum tráfego é artificialmente introduzido na rede, este tipo de monitorização só pode ser utilizado em situações onde os pacotes que se pretendem analisar já se encontram em trânsito. Um exemplo do uso deste tipo de aplicações é a negociação dos parâmetros de QoS na especificação de um serviço pedido por um cliente. Para a monitorização dos parâmetros de QoS é necessário a correlação de informação proveniente de diferentes pontos de medida (por exemplo para medidas *one-way*).

As medidas passivas não se remetem apenas à possibilidade de se capturarem pacotes, em tempo real, num determinado ponto de rede. A leitura efectuada a elementos de rede do operador e do cliente, colocando agentes nesses elementos (ou simplesmente efectuando leituras remotas da informação aí já existente, p.ex. com contadores em MIBs SNMP ou RMON) pode também ser considerado como uma medição passiva da rede. É de notar que dificilmente este tipo de leituras a objectos remotos nos dará uma visão da rede em tempo real. As informações assim obtidas são geralmente de âmbito mais alargado, e usadas para gestão de rede.

As medidas efectuadas de forma passiva devem ser aplicadas a dois tipos distintos de tráfego, rígido e elástico. Por tráfego rígido, entende-se aquele que é sensível a atrasos, onde não existe controlo de fluxo extremo a extremo. É tolerante a perdas e permite baixos valores de atrasos e de *jitter*. Geralmente trata-se de tráfego com requisitos de tempo real, seja *streaming* de vídeo, de áudio, ou mesmo telefonia *sobre* IP.

Pelo contrário, o tráfego elástico é pouco sensível a atrasos, permite ser adaptado às condições da rede mas é normalmente pouco tolerante a perdas. Como exemplo deste tipo de tráfego temos pacotes SNMP, HTTP, transferência de ficheiros, acesso a bases de dados, etc..

### B Monitorização Activa

O funcionamento da monitorização activa baseia-se na introdução de tráfego na rede, por parte de um dos elementos de medida (origem), que será posteriormente recolhido da rede por outro elemento de medida (destino). A grande desvantagem deste método é o facto de se estar a introduzir pacotes na rede, pacotes esses que não contêm *dados reais*, podendo vir a aumentar o congestionamento da rede. No entanto, através da especificação de determinados parâmetros, como, por exemplo, a periodicidade de pacotes enviados, a quantidade, entre

outros, é possível minorar os efeitos da introdução de pacotes extra na rede.

Neste caso, as medidas efectuadas servem essencialmente para fazer verificações de SLA, detecção de falhas, engenharia de tráfego, e monitorização geral da rede. É ainda possível utilizá-las para fornecer informação a um *QoS Broker* para auxílio a decisões de forma a maximizar a utilização dos recursos disponíveis.

As medidas efectuadas de forma activa deverão, tal como as medidas passivas, levar em conta os dois tipos de tráfego, elástico e rígido.

As medidas podem ser divididas em medidas de nível de rede e medidas de nível de serviço. O primeiro tipo, nível de rede, não leva em consideração o tipo de tráfego que está em trânsito (que se está a medir) e tem apenas como objectivo avaliar atrasos, *jitter*, perdas e *throughput* em termos de pacotes. É importante referir que embora estas medidas possam ser alteradas com o tipo de tráfego que se encontra em trânsito devido às políticas de QoS existentes na rede, ao nível de rede essa informação não é importante. Se, por exemplo, desejar saber qual o atraso de um *stream* de áudio, essa informação não está directamente relacionada com a qualidade do *stream*, aquando da sua chegada ao destino.

As medidas a nível de serviço pretendem dar alguma informação sobre qual o estado do serviço em causa. O tipo de métricas associado a este tipo de medidas serão erros, tempos de *download*, tempos de acesso ou ainda a qualidade de um determinado *stream* áudio ou vídeo, através de algoritmos bem conhecidos como PESQ [10], entre outros. Por exemplo, através do uso do algoritmo PESQ é possível medir o grau de dificuldade que um ouvinte terá em entender um determinado *stream* de áudio.

É importante que a plataforma de medição activa seja capaz de gerar tráfego e interpretar resultados que possam satisfazer os dois tipos de tráfego acima indicados.

### C *Arquitectura geral de uma plataforma de medidas*

Uma plataforma das medidas deve conter dois tipos de elementos: controlo e medida. Os elementos de controlo têm por objectivo definir qual o tipo de testes que se pretende fazer e enviar essa informação para os elementos de medida. É ainda da sua responsabilidade receber, guardar e retirar conclusões das medidas efectuadas pelos elementos de medida. Por exemplo, o elemento de controlo deve calcular dados estatísticos tendo por base os valores de medidas efectuadas pelos elementos de medida. Os elementos de medida têm por objectivo enviar e receber os pacotes de teste, com base nas informações de controlo recebidas do elemento de controlo. É ainda da

sua responsabilidade extrair métricas directas dos pacotes de teste, bem como enviar os resultados para o elemento de controlo.

Os dados enviados dos elementos de medida para o elemento de controlo devem ser feitos através do protocolo IPFIX, já que se trata de um *standard* projectado exactamente com esse objectivo [4].

Para que os resultados obtidos sejam correctos é necessário que os vários elementos envolvidos nas medidas estejam de alguma forma sincronizados. Para a sincronização podem-se utilizar antenas GPS ou pode-se utilizar o protocolo NTP, tendo este último menor precisão.

## IV SOFTWARE DESENVOLVIDO

### A *Descrição da plataforma IP Probes*

Como descrito na secção III-C, uma plataforma de medição tem geralmente dois tipos de elementos, o elemento de medida e o elemento de controlo. A plataforma IP-Probes segue esta definição e apresenta estes dois tipos de elementos. A designação para estes elementos foi de elemento de medição (EM) e elemento de operação (EO), respectivamente. Numa rede podem existir vários EMs e EOs, embora o mais normal seja que para um determinado número de EMs exista um EO de controlo. A figura 5 mostra de forma abstracta a localização dos diferentes elementos envolvidos na plataforma IP-Probes.

A definição de testes é feita tendo por base a definição de Sessão e de Sub-Sessão.

Uma sessão é um conjunto de  $N (N \geq 1)$  sub-sessões, que é identificada por um identificador único. As sub-sessões partilham as mesmas características da sessão no que diz respeito ao tipo de medição que se pretende. Uma sessão e as suas  $N$  sub-sessões partilham a seguinte informação:

- O EM deve ou não enviar pacotes;

- O EM deve reportar resultados;

- Em qual das sentidos o EM deve reportar resultados (no caso de *two-way*);

- Tipo de métrica, isto é, se a sessão é *one-way*, *round-trip* ou *forward*;

- Uma sub-sessão é identificada por um identificador único e por todas as características configuráveis que tiver. Essas características são as que existem para a construção de um pacote do tipo P.

A utilização de sub-sessões foi tomada em consideração pois pretendia-se que uma das funcionalidades da plataforma fosse executar sessões de teste de forma

recorrente (por exemplo, que todos os dias às X horas fosse executado um determinado teste). Assim, com a definição de sessão e sub-sessões associadas a uma sessão torna-se mais simples (organizada) a definição de uma sessão (ou sub-sessão) e o armazenamento dos resultados.

A imagem seguinte representa esquematicamente a plataforma IP Probe. Nela é visível a existência de dois elementos de medida e de um elemento de operação. Repare-se que o EO tem uma *interface* de comunicação, baseada em XML-RPC, que lhe dá a possibilidade de comunicar com entidades externas. É ainda visível na imagem a existência de antenas GPS, para a sincronização dos diversos elementos envolvidos nas medidas.

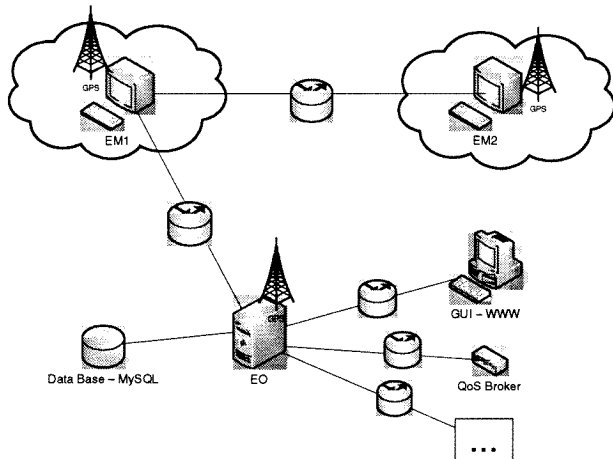


Figura 5 – Representação da plataforma IP Probe

**B** . Elemento de medida

O EM, tem como tarefa a realização das medidas e testes propriamente ditos, enviando e recebendo pacotes de teste (medição activa), com as características definidas, e guardando os resultados de uma forma não definitiva, isto é, não são escritos em disco. Outra responsabilidade do EM é o envio de resultados para o EO, periodicamente ou respondendo a pedidos explícitos.

O EM está dividido em dois blocos funcionais, o bloco de controlo e o bloco de medição.

O Bloco de Controlo deverá comunicar com o EO, recebendo os testes a realizar (sessões), e enviando resultados, quer totais, quer parciais, referentes à sessão que está a ser executada. Esta comunicação deverá ser feita através de uma *interface* de rede específica para esse efeito (a Interface de Gestão).

O Bloco de Medida deverá enviar e/ou receber pacotes de teste de/para outros EMs, fazendo de receptor, emissor ou de ambos. Um componente essencial nos EMs é o receptor GPS que faz com que todos os EMs estejam

sincronizados. Isto é fundamental para obter bons resultados nas medidas *One-way Delay*, em que os pacotes saem de um EM e chegam a outro: os resultados temporais enviados para o EO devem ser consistentes daí a necessidade de as diferentes unidades estarem sincronizadas temporalmente. O protocolo NTP pode também ser utilizado para tal, ainda que não tenha a mesma precisão que a conseguida pelo GPS e necessária para redes com débitos elevados. Usando GPS, os EM podem também funcionar como servidores NTP. A comunicação entre os EMs é efectuada utilizando-se uma *interface* de rede diferente da utilizada para a comunicação com o EO (a *interface* de Teste).

Os EMs extraem métricas directas dos pacotes de teste obtidos, independentemente das estatísticas finais a produzir; são produzidas e enviadas para o EO, medidas auxiliares de apoio, como por exemplo, quantos pacotes enviados, volume total de bytes enviados, tamanhos pacotes enviados, etc.

É possível aos EMs funcionarem como *proxy*, isto é, um EM pode efectuar *forward* de tráfego de teste para outro EM, podendo ou não o elemento que está a servir de *proxy* efectuar medidas.

**C** . Elemento de operação (EO)

O EO, tem por responsabilidade receber os resultados de uma sessão de teste e enviar a configuração de novas sessões de teste para os EMs.

O EO, é um sistema dividido em três blocos funcionais, EO Low, EO High e interface XML-RPC, como mostra a Figura 6.

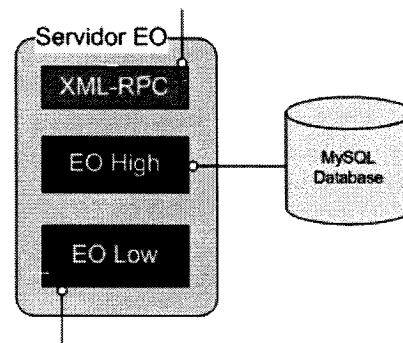


Figura 6 – Arquitectura do EO

O EO Low tem que lidar basicamente com duas acções: a configuração das Sessões e a obtenção dos resultados. O envio dos parâmetros relativo a uma sessão de teste entre o EO e os EMs é feito através do protocolo proprietário IP-Probe EO2EM Protocol (IPEP), assim como o envio de resultados dos EMs para o EO. Este bloco funcional é

responsável por toda a manipulação de rede entre os EMs e o EO.

O EO High é responsável pela comunicação com a base de dados e pela comunicação entre o EO LOW e a interface com o mundo exterior, através do bloco XML\_RPC. Na base de dados são guardadas as configurações relativas às sessões, bem como os resultados obtidos através dos testes efectuados. É ainda guardada na base de dados toda a informação referente aos EMs e EOs que possam co-existir na rede.

A interface XML-RPC é a parte responsável pela comunicação entre o EO High e o mundo exterior. Este bloco funcional disponibiliza um conjunto de *interfaces* com o exterior que lhe permitem interagir com diferentes entidades, permitindo uma fácil integração com outras ferramentas, como por exemplo um *QoS Broker*. Com esta funcionalidade é possível que o EO esteja localizado numa outra máquina que não a máquina onde o utilizador está a definir os parâmetros da sessão de teste.

Existe neste momento uma *interface web* capaz de comunicar com o bloco funcional XML-RPC, para a definição de sessões e obtenção de resultados. É ainda possível a manipulação de EMs e de EOs, é possível por exemplo criar-se um novo EM dizendo quais os seus IPs, nome e o local onde se encontra.

## V CONCLUSÕES

A utilização de uma plataforma de medição baseada em pontas de prova ou sondas é uma mais valia para os operadores de redes IP. A recolha de dados referentes às métricas normalizadas é a principal aplicação do sistema, permitindo que a definição de contratos de serviço prestados aos clientes ou a outros operadores sejam claramente definidos. A plataforma vem também permitir que os resultados sejam avaliados em tempo real permitindo que outras entidades, de controlo ou gestão, utilizem os resultados obtidos para se ajustar às condições da rede. Também o dimensionamento, o planeamento e a gestão podem ser facilitados, usando os dados obtidos na análise estatística. Em ambientes móveis esta plataforma tem especial interesse para analisar a variação do comportamento médio de desempenho do nós móveis, à medida que estes vão mudando de ponto de ligação à rede.

No entanto, este tipo de plataforma não está vocacionada para uma análise pormenorizada do comportamento do *handover*, algo que é importante para avaliar o comportamento protocolar da rede de comunicações. Neste sentido, a ferramenta TCP desenvolvida vem proporcionar um complemento interessante para um sistema de análise de desempenho para ambientes móveis da próxima geração.

O desenvolvimento do Elemento de Operação foi realizado com recurso à rede Moby Dick. Os resultados obtidos satisfizeram os requisitos iniciais do projecto, e em particular a análise feita ao comportamento do TCP permitiu clarificar alguns detalhes dos novos protocolos [8] desenvolvidos para esta rede.

## AGRADECIMENTOS

Os autores do trabalho agradecem a utilização da rede Moby Dick no IT, e a disponibilização do software original de IP-Probes por parte da Portugal Telecom Inovação, que serviu de base a parte deste trabalho.

## REFERÊNCIAS

- [1] TCP Behavior Inference Tool - <http://www.icir.org/tbit/>
- [2] Web100 project - <http://www.web100.org/>
- [3] Sockmon Socket Monitoring Utility - <http://sourceforge.net/projects/sockmon>
- [5] IP Flow Information Export (ipfix) - <http://www.ietf.org/html.charters/ipfix-charter.html>
- [6] IP Performance Metrics (ippm) - <http://www.ietf.org/html.charters/ippm-charter.html>
- [7] V. Paxson, G. Almes, J. Mahdavi, M. Mathis - Framework for IP Performance Metrics, <http://www.ietf.org/rfc/rfc2330.txt>
- [8] D. Gomes, N. Duarte, N. Sénica, "Demonstrador de Mobilidade Heterogénea", Revista do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro - Vol. 4 - Nº1, Setembro de 2003, ISSN: 1645-0493
- [9] Moby Dick Project - <http://www.ist-mobydick.org/>
- [10] Perceptual evaluation of speech quality - <http://www.itu.int/rec/recommendation.asp?type=folders&lang=e&parent=T-REC-P.862>