Universidade de Aveiro
2008
Departamento de Electrónica, Telecomunicações e Informática

Lidia Elena da Costa dos Reis

Medição e Caracterização de Tráfego Tempo-Real em Redes IP

Universidade de Aveiro
2008
Departamento de Electrónica, Telecomunicações e Informática

Lidia Elena da Costa dos Reis

Medição e Caracterização de Tráfego Tempo-Real em Redes IP

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Doutor Paulo Jorge Salvador Serra Ferreira, Professor Auxiliar Convidado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e co-orientação científica do Doutor Amaro Fernandes de Sousa, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Dedico este trabalho à minha família e amigos, especialmente aos meus pais, irmãos e namorado, que sempre me apoiaram e proporcionaram condições para que obtivesse sucesso ao longo da minha vida.

o júri

Presidente

Doutor João Nuno Pimentel da Silva Matos

Professor Associado da Universidade de Aveiro

Doutor Paulo Jorge Salvador Serra Ferreira (Orientador)

Professor Auxiliar da Universidade de Aveiro

Doutor Amaro Fernandes de Sousa (Co-Orientador)

Professor Auxiliar da Universidade de Aveiro

Doutor Joel José Puga Coelho Rodrigues

Professor Auxiliar do Departamento de Informática da Faculdade de Ciências da Engenharia da Universidade da Beira Interior

agradecimentos

Pretendo expressar os meus agradecimentos a todas as pessoas que têm contribuído para o desenvolvimento deste trabalho:

- Aos meus pais, Leonel e Lidia, pelo apoio, carinho e pelas condições favoráveis que proporcionaram para poder atingir os objectivos na minha vida pessoal e académica.
- Aos meus irmãos, Diana e Jesús, pelo carinho, apoio e por estarem sempre ao meu lado.
- Ao meu namorado, Vitor, pelo amor, apoio, compreensão e sobre tudo pela motivação que me deu ao longo da elaboração deste trabalho.
- Ao meu orientador, Doutor Paulo Salvador, pela colaboração e apoio e sugestões prestadas durante a realização do trabalho. Também queria agradecer ao Professor pela disponibilidade prestada.
- Ao meu co-orientador, Doutor Amaro Sousa, pela disponibilidade, colaboração e apoio prestados ao longo da realização deste trabalho.
- Ao Ricardo Rolhas, André Santos e Pedro Moreira, pela ajuda ao longo do trabalho, através dos seus conhecimentos e na integração ao laboratório do IT.
- Ao Hélder Veiga, pela ajuda fornecida com o funcionamento do DTMS-P2P.
- Aos meus amigos e colegas de curso, pelo apoio e companheirismo.

palavras-chave

Rede, VoIP, videoconferência, vídeo, pacotes, dados, sinalização, estatísticas, chamadas, fluxos, monitorização, protocolos, capturas, monitorização distribuida, *probes*, *super-probes.*

resumo

Nos dias que correm os recursos financeiros e de tempo são cada vez mais escassos, é por isto que as novas tecnologias procuram economizar estes recursos. Um exemplo disto são os serviços tempo-real prestados pelas redes IP, como é o caso do VoIP e da videoconferência.

Através de telefones VoIP é possível realizar chamadas telefónicas utilizando redes IP (por exemplo, *Internet*), desta forma aproveitam-se as redes de dados já existentes para transmitir voz. O VoIP oferece também mobilidade, pois um equipamento VoIP pode efectuar chamadas desde qualquer lugar, bastando que tenha acesso à *Internet*. Outra vantagem do VoIP é a interoperabilidade entre os diversos operadores (POTS e VoIP).

A videoconferência permite também economizar recursos de tempo e financeiros, visto que através deste serviço o utilizador pode participar em eventos sem necessidade de se deslocar fisicamente. Além disto os equipamentos de videoconferência facultam a emissão de vídeo através da *Internet*, permitindo aos utilizadores assistir a eventos em tempo-real ou visualizar vídeos de eventos previamente gravados.

Um dos objectivos desta dissertação é a medição e caracterização de tráfego com características de tempo-real em redes baseadas no protocolo IP, sendo um dos casos de estudos deste trabalho os serviços de VoIP e emissão de vídeo, prestados pela rede IP de investigação do IT de Aveiro.

Para efectuar a medição e caracterização de tráfego encontram-se disponíveis diversas ferramentas, como é o caso do *NTOP*, *TCPDump*, *TSTAT* e *TCPSTAT*. Estas ferramentas são muito úteis, mas não são adequadas para efectuar uma medição distribuída, pois ao utilizar qualquer uma destas ferramentas para monitorizar diversos pontos de uma rede torna-se necessário a configuração e execução dos testes computador a computador, tornando esta tarefa muito complexa.

De forma a tornar mais simples as medições distribuídas, a equipe de investigação do IT desenvolveu uma ferramenta de monitorização distribuída com uma arquitectura *peer-to-peer* hierárquica, trata-se do *DTMS-P2P*. Esta utiliza outras ferramentas existentes, por exemplo o *TCPDump*, mas acrescentando outras opções, que permitem uma medição distribuída sem necessidade de interagir nos diversos computadores cada vez que se deseja executar um teste ou consultar resultados. Esta ferramenta é também objecto de estudo desta dissertação, sendo contemplada no segundo caso de estudo.

keywords

Network, VoIP, videoconferencing, video, packages, data, signaling, statistics, calls, flows, monitoring, protocols, captures, distributed monitoring, probes, super-probes.

Abstract

Nowadays the financial resources and time are increasingly scarce, it is because of this that the new technologies try to save these resources. An example of this is the real-time services provided by IP networks, such as VoIP and videoconferencing.

With a VoIP phone is possible to make phone calls using IP networks (for example, Internet), by this way taking advantage of existing data networks for voice transmit. The VoIP also offers mobility, because the VoIP equipment can make calls from anywhere, having just an Internet access. Another advantage of VoIP is the interoperability between different operators (POTS and VoIP).

The videoconferencing also can save financial and time resources, because through this service the user can participate in events without needing to physically move. Besides this the videoconferencing equipment provides video delivery over the Internet, allowing users to watch events in real-time or view events previously recorded.

One of the objectives for this thesis is measurement and characterization of traffic with real-time characteristics in networks based on IP protocol, being one of the study cases of this work the VoIP services and video transmissions, provided by the investigation IP network on IT of Aveiro.

To make the measurement and characterization of traffic, there are many tools available, such as NTOP, TCPDump, TSTAT and TCPSTAT. These tools are very useful, but are not suitable to make a distributed measurement, because when using any of these tools to monitor various points of a network, becomes necessary to setup and execute computer to computer tests, making this task very complex.

In order to simplify the distributed measurements, the IT research team developed a tool for distributed monitoring with a peer-to-peer hierarchical architecture, it is the DTMS-P2P. This uses other tools available, such as TCPDump, but adding other options that allow a distributed measurement without needing to interact with the other computers each time you want to run a test or see results. This tool is also being studied in this thesis, being contemplated in the second study case.

Conteúdo:

Lista de figuras:

Lista de tabelas:

1 INTRODUÇÃO

Nos dias que correm, os recursos financeiros e de tempo são cada vez mais escassos e, por isso, as novas tecnologias procuram economizar estes recursos. Um exemplo disto são os serviços tempo-real prestados pelas redes IP, como é o caso do VoIP e da videoconferência.

Através de telefones VoIP é possível realizar chamadas telefónicas utilizando redes IP (por exemplo, *Internet*) e, desta forma, aproveitar as redes de dados já existentes para transmitir voz e/ou dados. As chamadas VoIP são de menor custo do que as chamadas na rede POTS, podendo chegar a ser gratuitas. O VoIP não só oferece uma redução de custos, como também oferece mobilidade, pois um equipamento VoIP pode efectuar chamadas de qualquer lugar, bastando para isso que tenha acesso à *Internet*, além disto permite a interoperabilidade entre os diversos operadores (POTS e VoIP).

A videoconferência também permite economizar recursos de tempo e financeiros, visto que através deste serviço o utilizador pode participar em eventos sem necessidade de se deslocar fisicamente. Além disto, os equipamentos de videoconferência IP facultam a emissão de vídeo através da *Internet*, permitindo aos utilizadores assistir a eventos em tempo-real ou visualizar vídeos de eventos previamente gravados.

É importante acompanhar o funcionamento destes serviços de forma a estudar o comportamento dos mesmos e poder detectar falhas, verificar as configurações, entre outras. Este estudo efectua-se recorrendo à monitorização e caracterização do tráfego.

Para poder efectuar a medição e caracterização do tráfego de serviços tempo-real, estão disponíveis inúmeras ferramentas, tais como o *NTOP*, *TCPDump*, *TSTAT* e *TCPSTAT*. Estas ferramentas são muito úteis, mas não são adequadas para efectuar medições distribuídas, pois ao utilizar qualquer uma destas ferramentas para monitorizar diversos pontos de uma rede torna-se necessário, além da instalação de computadores nos pontos de medição, a configuração e execução dos testes deve ser feita computador a computador, tornando esta tarefa muito complexa.

De forma a tornar mais simples as medições distribuídas, a equipe de investigação do IT desenvolveu uma ferramenta de monitorização distribuída com uma arquitectura *peer-to-peer* hierárquica, designada por *DTMS-P2P*. Esta ferramenta utiliza outras ferramentas existentes, por exemplo o *TCPDump*, mas acrescenta outras opções, como é o caso da configuração e execução de testes desde um ponto remoto, permitindo efectuar testes em

diversos pontos de uma rede, podendo ser configurados a partir de um único ponto central.

1.1 Objectivos

O objectivo principal desta dissertação é a **medição e caracterização de tráfego com características de tempo-real em redes baseadas no protocolo IP**. Esta caracterização deve ser feita tanto a nível dos pacotes como a nível da sessão. O plano de trabalho seguido para atingir este objectivo foi o seguinte:

- Familiarização com a operação da rede de telecomunicações e os respectivos protocolos de comunicações;
- Familiarização com as ferramentas de monitorização e medição de tráfego;
- Compreensão dos métodos de caracterização estatística de tráfego;
- Execução das medições de tráfego;
- Tratamento estatístico das medições e caracterização de tráfego medido.

Além disto, pretendeu-se **mostrar as funcionalidades de um sistema de monitorização distribuído** (DTMS-P2P). Para isto procedeu-se da seguinte forma:

- Compreensão dos sistemas *peer-to-peer* e familiarização com o sistema de medição de tráfego;
- Concepção de um cenário de monitorização distribuída;
- Execução de testes que mostrem as funcionalidades do DTMS-P2P.

1.2 Motivação da dissertação

O IT (Instituto de Telecomunicações) de Aveiro dispõe de uma rede IP de investigação que suporta além dos serviços típicos da *Internet*, os serviços de VoIP, videoconferência e emissão de vídeo. No caso do VoIP, encontram-se instalados um conjunto de telefones e um *CallManager* da Cisco. No caso da videoconferência e emissão de vídeo, os equipamentos utilizados são da TANDBERG (terminais de videoconferência e *Content Server*).

Pretende-se medir e caracterizar o tráfego gerado por estes serviços; tendo em conta que os utilizadores destes serviços são utilizadores reais, logo esta rede constitui um cenário adequado para a medição de tráfego real.

Pretende-se também mostrar as funcionalidades do DTMS-P2P, pois trata-se de uma ferramenta de monitorização distribuída, desenvolvida pela equipa de investigação do

Instituto de Telecomunicações de Aveiro. Esta ferramenta permite configurar diversos testes em diversos pontos da rede, assim como consultar os resultados obtidos através destes testes. Estas tarefas podem ser executadas no mesmo ponto onde se deseja efectuar a medição ou à distância.

1.3 Estrutura da dissertação

Esta dissertação encontra-se dividida em duas partes, uma primeira que contém o estado da arte e uma segunda parte que se refere aos casos de estudo.

O estado da arte é composto por três capítulos:

- Capítulo 2 – Protocolos utilizados nos serviços VoIP, videoconferência e emissão de vídeo: descreve alguns protocolos utilizados nos serviços VoIP, videoconferência e emissão de vídeo. Como é o caso do SCCP, MGCP, H.323, RTP, RTSP e MMS.
- Capítulo 3 – Sistema de monitorização distribuída – DTMS-P2P: descreve o funcionamento do DTMS-P2P.
- Capítulo 4 – Caracterização estatística do tráfego: explica o que é a caracterização estatística do tráfego e os métodos a seguir.

Na segunda parte da dissertação são descritos os procedimentos utilizados para efectuar os testes, e apresentados os resultados obtidos nos casos de estudo. Esta secção é composta por dois capítulos:

- Capítulo 5 – Rede IP de investigação do IT: refere-se ao estudo de dois serviços prestados pela rede IP de investigação do IT, trata-se do VoIP e da emissão de vídeo. O serviço de VoIP é fornecido pela rede VoIP contida nesta rede de investigação. No caso da emissão de vídeo, o serviço é fornecido pela secção da rede que presta o serviço de videoconferência (rede de videoconferência).
- Capítulo 6 – DTMS-P2P: são mostradas as funcionalidades do DTMS-P2P através de uma série de testes.

ESTADO DA ARTE

2. Protocolos utilizados nos serviços VoIP, videoconferência e emissão de vídeo

3. Sistema de monitorização distribuída - DTMS-P2P

4. Caracterização estatística do tráfego

2 PROTOCOLOS UTILIZADOS NOS SERVIÇOS VOIP, VIDEOCONFERÊNCIA E EMISSÃO DE VÍDEO

2.1 Introdução

Actualmente, além dos serviços típicos da *Internet*, as redes IP podem oferecer serviços de transmissão de voz e/ou dados (secção 2.1.1), assim como serviços de transmissão de áudio, vídeo e/ou dados (secção 2.1.2) desde que estejam devidamente preparadas, com os equipamentos e protocolos necessários.

Para que os equipamentos possam comunicar, é preciso estabelecer normas, de forma a que estes comuniquem utilizando a mesma "linguagem". Estas normas encontram-se definidas nos **protocolos**.

Os protocolos podem ser divididos em dois grupos:

- **Protocolos de sinalização** (secção 2.2): encarregam-se de estabelecer e finalizar as sessões; também servem para que os equipamentos indiquem a sua presença na rede, entre outras.
- **Protocolos de dados** (secção 2.3): estes são utilizados para transmitir os dados, por exemplo, a voz.

2.1.1 Voice over IP(VoIP)

O VoIP é uma tecnologia que permite estabelecer chamadas de voz através de uma rede de dados (por exemplo, *Internet*). Para este efeito, os sinais analógicos de voz devem ser convertidos em sinais digitais recorrendo a um *Digital Signal Processor* (DSP), sendo este processo efectuado em três fases: amostragem, quantificação e codificação (como se pode observar na seguinte figura). Após a conversão analógico-digital, os dados digitais são compactados e enviados sob a forma de pacotes com endereçamento IP através da *Internet*.

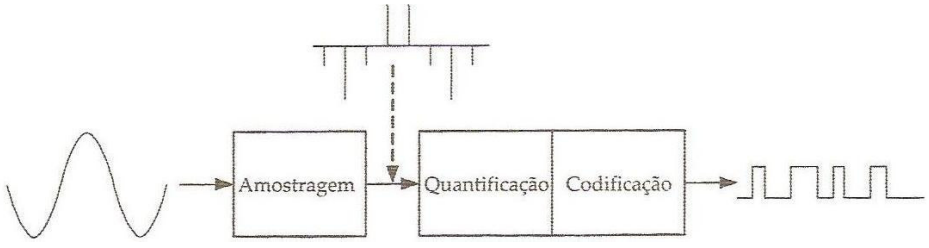

Figura 1 – Processo de digitalização [Sa07]

Esta codificação e transmissão de dados é feita com recurso a vários protocolos; é por isto que o VoIP é considerado, mais que um protocolo, um conjunto de protocolos.

O VoIP oferece as seguintes vantagens:

- Integração de um novo serviço sobre a rede IP, neste caso a voz.
- Interoperabilidade entre diversos provedores (POTS e VoIP).
- Utiliza redes de dados existentes.
- Independência da tecnologia de transporte.
- Redução de custo das chamadas.

2.1.2 Videoconferência e emissão de vídeo

A videoconferência é uma forma de comunicação, permitindo que pessoas que se encontram afastados fisicamente comuniquem como se estivessem no mesmo lugar, através da transmissão de vídeo, áudio e dados.

Uma sessão de videoconferência pode ser estabelecida entre dois ou mais participantes; quando é entre dois participantes diz-se que é uma **videoconferência ponto-a-ponto**; no caso de ter mais do que dois participantes diz-se que é uma **videoconferência multiponto** e, neste caso, é necessário ter uma *Multipoint Control Unit* (MCU)[1].

A videoconferência tem muitas vantagens, entre elas a economia do tempo e dos recursos, pois através da videoconferência evita-se o deslocamento físico das pessoas que participam num determinado evento, por exemplo uma reunião, uma apresentação. Uma outra vantagem é a possibilidade de gravar as conferências e posteriormente disponibilizar estes vídeos para outras pessoas poderem visualizar (emissão de vídeo).

2.2 Protocolos de sinalização

2.2.1 *Skinny Client Control Protocol* (SCCP)

O *Skinny Client Control Protocol* (SCCP) é um protocolo de sinalização proprietário da *Cisco*. Este protocolo é composto por um conjunto de mensagens, as quais são trocadas entre o telefone VoIP e o *CallManager* (Anexo I.1). Estas mensagens são enviadas sobre TCP e têm como função trocar a informação necessária para estabelecer e finalizar uma chamada, indicar ao *CallManager* a existência dos telefones na rede, entre outras.

Segundo a *Cisco,* as mensagens que compõem o SCCP podem ser agrupadas em três grupos diferentes: *registration and management messages, call control messages* e *media control*

[1] Segundo a *Fundação para a Computação Científica Nacional* (FCCN) a MCU "é responsável por criar uma sala virtual onde os participantes se reúnem" [FCCN04].

messages [Cisco0?b]. Na seguinte figura pode observar-se um esquema com a divisão dos tipos de mensagens conforme indica a Cisco.

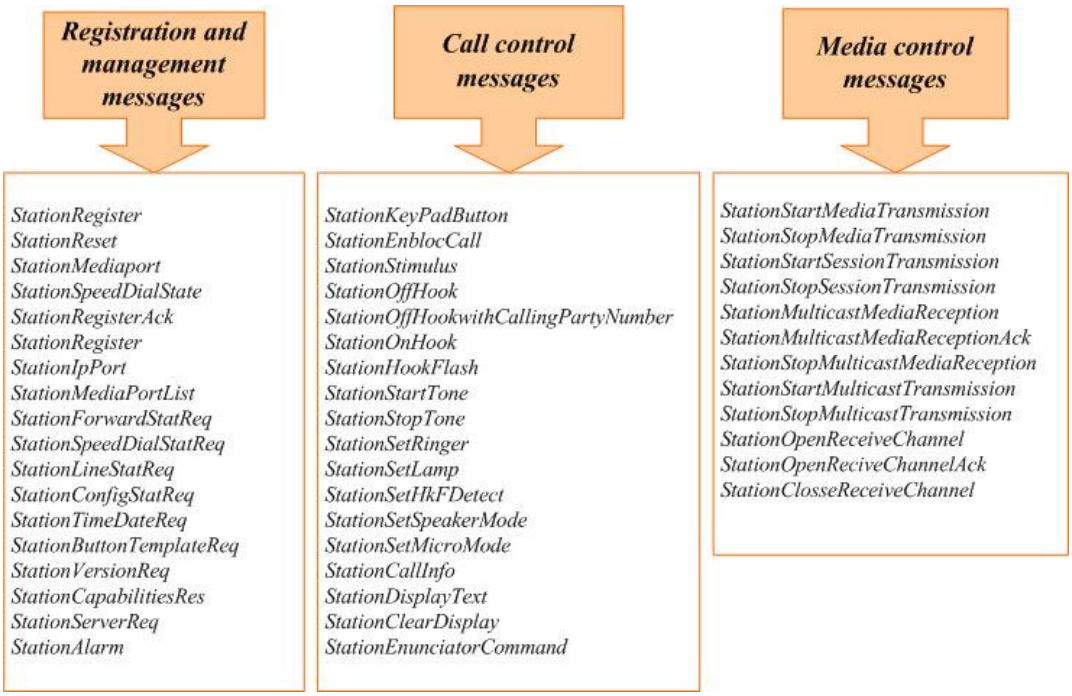

Figura 2 – Tipos de mensagens SCCP [Cisco0?b]

Algumas destas mensagens têm como função indicar ao *CallManager* o que está a acontecer no equipamento terminal, por exemplo as mensagens de *OnHookMessage* e *OffHookMessage*, as quais indicam se o auscultador está ou não no "gancho". Outras mensagens servem para indicar ao equipamento terminal alguma acção, um exemplo disso é a mensagem *SetRingerMessage* usada pelo *CallManager* para indicar quando o telefone começa a tocar (*on*) ou quando deve parar de tocar (*off*). Este tipo de mensagens são enviadas em situações específicas.

Além das mensagens que indicam uma acção num tempo específico, no protocolo SCCP também existem mensagens periódicas, este é o caso das mensagens *KeepAliveMessage*, as quais são enviadas, aproximadamente, de 30 em 30 segundos do telefone VoIP para o *CallManager*. Estas mensagens são utilizadas pelo telefone VoIP para informar o *CallManager* da sua existência na rede, este último responde com

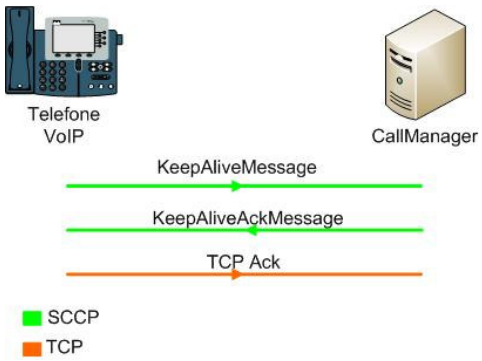

Figura 3 – Sequência de mensagens *KeepAlive*

uma mensagem do tipo *KeepAliveAckMessage*, a qual é novamente respondida pelo telefone com uma mensagem *TCP Ack*, como se mostra na figura 3.

2.2.2 *Media Gateway Control Protocol* (MGCP)

Este protocolo é utilizado para controlar as chamadas realizadas entre um elemento da rede VoIP e um elemento externo a esta rede, isto é, entre um telefone de uma rede VoIP e um telefone que pertence a uma central telefónica. O protocolo MGCP encontra-se definido na RFC 3435.

Num sistema MGCP existem dois tipos de elementos, o *Call Agent* e os *endpoints*. O *Call Agent* é o dispositivo (ou os dispositivos) que controla(m) as chamadas, e os *endpoints* são qualquer uma das portas de "voz" do *Gateway*. [Cisco06c]

O MGCP efectua o controlo das chamadas através de comandos e respostas, isto é, cada comando deve ter uma resposta obrigatória, estes comandos encontram-se na seguinte tabela:

Comandos	Origem/Destino	Descrição
EndpointConfiguration	*Call Agent/Gateway*	Utilizado para especificar a codificação dos sinais que serão recebidos pelo *endpoint*.
CreateConnection	*Call Agent/Gateway*	Cria uma conexão entre dois *endpoints*.
ModifyConnection	*Call Agent/Gateway*	Altera os parâmetros associados a uma ligação estabelecida.
DeleteConnection	*Call Agent/Gateway*	Termina a ligação actual.
	Gateway/Call Agent	Liberta uma ligação.
NotificationRequest	*Call Agent/Gateway*	Utilizado pata pedir ao gateway que envie notificações sobre a ocorrência de eventos em um determinado *endpoint*.
Notify	*Gateway/Call Agent*	Enviado quando ocorre algum evento.
AuditEndpoint	*Call Agent/Gateway*	Determina o estado de um determinado *endpoint*.
AuditConnection	*Call Agent/Gateway*	Devolve todos os parâmetros associados a uma ligação.
RestartInProgress	*Gateway/Call Agent*	Indica que o *endpoint* (ou grupo de *endpoints*) está a ser posto em serviço ou fora de serviço.

Tabela 1 – Comandos do protocolo MGCP [AF03 e Cisco06c]

Um exemplo de um de uma troca de mensagens, comando e resposta, encontra-se na figura 4. Trata-se do comando *Notify* (NTFY), enviado periodicamente de 15 em 15 segundos do *router* (*Gateway*) para o *CallManager* (*Call Agent*); este último responde (neste

exemplo, é usado o código 200 que indica que a operação foi executada com normalidade).

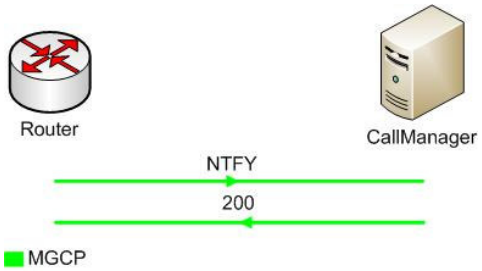

Figura 4 – Exemplo do comando *Notify* e resposta

Como já foi referido existem dois tipos de mensagens, os de comando e os de resposta, ambos compostos por um cabeçalho (cabeçalho do comando e cabeçalho da resposta, respectivamente) seguidos, opcionalmente, de uma descrição da sessão. [AF03]

O **cabeçalho da mensagem de comando** é composto por uma linha de comando (*command line*) e um conjunto de zero ou mais linhas de outros parâmetros. A linha de comando contém quatro elementos: *requested verb*, identificação da operação, nome do destino e versão do protocolo [AF03]. A seguinte tabela mostra os códigos utilizados para o *request verb*:

Verbo (Verb)	Código
AuditEndpoint	AUEP
AuditConnection	AUCX
CreateConnection	CRCX
DeleteConnection	DLCX
ModifyConnection	MDCX
NotificationRequest	RQNT
Notify	NTFY
RestartProgress	RSIP

Tabela 2 – Códigos utilizados para definir cada verbo (*requested verb*) [AF03 e Cisco06c]

O **cabeçalho da mensagem de resposta** é composto por uma linha de resposta, opcionalmente seguido de outros parâmetros. A linha de resposta contém um código de resposta (*Response Code* – Anexo II.1) e a identificação da operação (*Transaction ID*). [AF03]

2.2.3 H.323

O H.323 é uma norma ITU-T que define um conjunto de protocolos que permitem a transmissão de dados multimédia sobre redes de pacotes não orientadas à conexão sem garantia de qualidade de serviço [HR03].

Um sistema H.323 é constituído pelos seguintes componentes [HR03 e ITU-T06]:

- **Terminais** (*endpoints*)

São dispositivos que executam a norma H.323. Os terminais H.323 proporcionam capacidade de comunicação de áudio e opcionalmente de vídeo e de dados em conferências ponto a ponto ou multiponto.

Os terminais podem ser dispositivos específicos (*hardware*) ou computadores com programas que implementam a norma H.323 (*software*).

- ***Gateway***

  Proporciona conectividade entre uma rede H.323 e uma que não suporte este *standard*. Este componente faz a tradução dos formatos e procedimentos de transmissão.

- ***Gatekeeper***

  O *gatekeeper* não é obrigatório num sistema H.323. Este componente presta serviços de controlo de chamada, tradução de endereços, controlo de acessos aos recursos da rede por parte dos outros elementos da rede (terminais, *gateways* e MCUs), controlo de largura de banda, isto é, o *gatekeeper* gere uma determinada zona da rede.

- **MCU** (*Multipoint Control Unit*)

  Este componente proporciona as condições para poderem ser efectuadas sessões entre três ou mais terminais (sessão multiponto). Está formado por dois componentes lógicos:

  - MC (*Multipoint Controller*): encarregado da gestão do controlo das chamadas multiponto.
  - MP (*Multipoint Processor*): o objectivo desde componente é misturar os sinais de áudio, vídeo e dados procedentes dos diversos terminais.

Os *codecs* também fazem parte do sistema H.323. Existem três tipos de *codecs*: áudio, vídeo e dados. Um *codec*, quer seja de áudio, vídeo ou dados, codifica o sinal original num formato adequado à sua transmissão pela rede e faz a operação inversa no destino. Os *codecs* utilizados pelo H.323 para áudio são o G.711, G.722, G.723.1, G.728 e G.729, para o vídeo são o H.261 e o H.263 e para dados é o T.120 [HR03].

Como já foi referido anteriormente, a norma H.323 é composta por um conjunto de protocolos. Os três protocolos de controlo que formam parte do H.323 são os seguintes [HR03 e ITU-T06]:

- A sinalização da chamada é efectuada através do protocolo **H.225/Q.931**

A sinalização H.225 é feita através de TCP, pois esta deve ser feita sobre um canal fiável. O canal de sinalização é criado para transportar mensagens de controlo. A sinalização pode ser feita de duas formas: **directa** ou **indirecta**. No caso da sinalização ser **directa**, não há intervenção do *gatekeeper*, isto é, a sinalização é trocada directamente entre os terminais. Quando a sinalização é feita através do *gatekeeper*, designa-se sinalização **indirecta**.

- O estabelecimento da chamada desde a origem ao destino é feito através do **RAS H.225.0**

No caso da sinalização indirecta, o RAS (*Registration, Admission and Status*) é o protocolo que se estabelece entre equipamentos terminais e *gatekeepers* antes do estabelecimento de qualquer outro tipo de canal. As mensagens RAS viajam através de um canal UDP.

As funções deste protocolo são as seguintes:

- Descobrir o *gatekeeper*

  Utilizado pelos terminais (*endpoints*) para determinar em qual *gatekeeper* se devem registar. Este processo pode ser de dois tipos: **dinâmico** ou **estático**. No caso de ser **estático**, o terminal sabe *a priori* qual o endereço do seu *gatekeeper*. No caso da descoberta do *gatekeeper* ser efectuada de forma **dinâmica**, o terminal envia uma mensagem GRQ (*GatekeeperRequest*) em *multicast*, a qual é respondida afirmativamente (GCF – *GatekeeperConfirm*, caso (a) da seguinte figura) ou negativamente (GRJ – *GatekeeperReject*, caso (b) da seguinte figura) pelo(s) *gatekeeper*(s) do sistema.

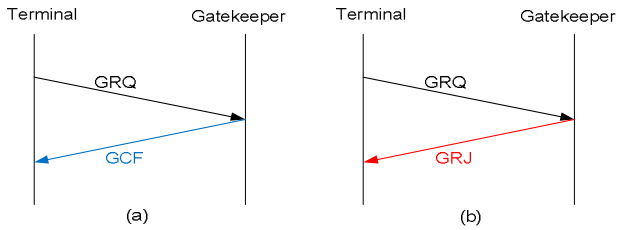

Figura 5 – Troca de mensagens para descobrir o *gatekeeper* dinamicamente

- Registar os terminais

  É através deste processo que o terminal comunica ao *gatekeeper* o seu endereço (endereço de transporte e *alias*). O registo deve ser efectuado antes de qualquer chamada, pois todos os terminais deverão registar-se no

gatekeeper no processo de descoberta. O pedido de registo é feito através da mensagem RRQ (*RegistrationRequest*), a qual é respondida afirmativamente (a) com *RegistrationConfirm* ou negativamente (b) com a mensagem *RegistrarionReject* (figura 6).

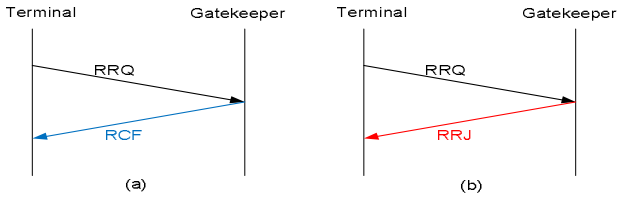

Figura 6 – Troca de mensagens do pedido de registo

Também pode ser efectuado um pedido de cancelamento de registo, este pode ser iniciado pelo terminal ou pelo *gatekeeper*. Este procedimento é efectuado através do envio da mensagem URQ (*UnsegisterRequest*). Na seguinte figura mostra-se o processo de cancelamento de registo iniciado pelo terminal, este pode ter resposta afirmativa ou negativa, *UnregisterConfirm* e *UnregisterReject* respectivamente.

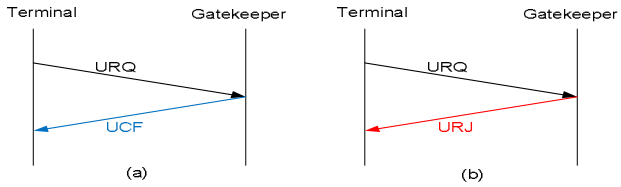

Figura 7 – Troca de mensagens do processo de cancelamento de registo iniciado pelo terminal

No caso do processo ser iniciado pelo *gatekeeper* a resposta do terminal é *UnregisterConfirm*.

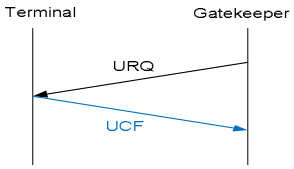

Figura 8 – Troca de mensagens do processo de cancelamento de registo iniciado pelo *gatekeeper*

- Localizar os terminais

Através do *alias* do terminal, pode obter informação de contacto do mesmo. Isto é feito através da mensagem LRQ (*LocationRequest*).

- Outras funções desempenhadas pelo RAS tais como admissões, alteração da largura de banda, informação do estado e abandono da sessão.

• A negociação dos fluxos de dados é realizada através do **H.245**

Estas mensagens de controlo têm como objectivo gerir o funcionamento do terminal H.323 e levar informações relacionadas com, por exemplo, capacidades, abertura e fecho de canais lógicos para o transporte de fluxos de dados, mensagens de controlo de fluxo, entre outros.

2.3 Protocolos de dados

2.3.1 *Real-Time Tranfer Protocol* (RTP)

O *Real-Time Tranfer Protocol* (RTP) é um protocolo utilizado em serviços *multicast* ou *unicast*, para transportar tráfego tempo-real. Este protocolo encontra-se definido nos RFCs 1889, 1890 e 3550.

Os pacotes do protocolo RTP são transportados sobre UDP. Não é usado o TCP, pois no caso do tráfego tempo-real é mais importante o tempo de chagada dos dados do que a fiabilidade dos mesmos. O protocolo UDP é mais indicado do que o TCP para tráfego tempo real, pois este não utiliza nenhum tipo de controlo de fluxo, erros, nem de congestionamento. Uma outra desvantagem do protocolo TCP relativamente ao UDP, é que o TCP é um protocolo orientado à conexão, o que não é indicado para tráfego *multicast.*

O RTP, através do seu cabeçalho, adiciona ao UDP as características necessárias para transportar este tipo de tráfego. Segundo a RFC 1889, o cabeçalho RTP é composto pelos seguintes campos (figura seguinte):

• **V** (*version*): 2 bits que indicam a versão do protocolo RTP.

• **P** (*padding*): um bit que quando este está activo (1) indica que existe um ou mais octetos adicionais de *padding*.

• **X** (*extension*): um bit que quando está activo (1) indica a existência de cabeçalho de extensão.

• **CC** (*CSRC count*): 4 bits que indicam o número de CSRCs.

• **M** (*marker*): um bit utilizado para permitir situações especiais, tais como marcar o início ou o fim de uma rajada de pacotes de vídeo ou áudio.

- **PT** (*payload type*): 7 bits que indicam o formato do *payload* utilizado (tipo de codificação). Os valores do PT estão definidos na RFC 1890, e também podem ser consultados no Anexo III.1.
- ***Sequence number***: um valor composto por 16 bits; o valor do *sequence number* do primeiro pacote é um valor aleatório, e depois é incrementado por cada pacote enviado, permitindo desta forma o receptor consiga ordenar os pacotes recebidos e identificar os pacotes não recebidos.
- ***Timestamp***: 32 bits que indicam o tempo de amostragem do primeiro octeto.
- **SSRC** (*synchronization source identifier*): 32 bits que identificam a fonte.
- **CSRC** (*contributing source identifier*): permite definir de 0 a 15 elementos, de 32 bits cada um, e serve para identificar as fontes que contribuem para o *payload* do pacote actual.

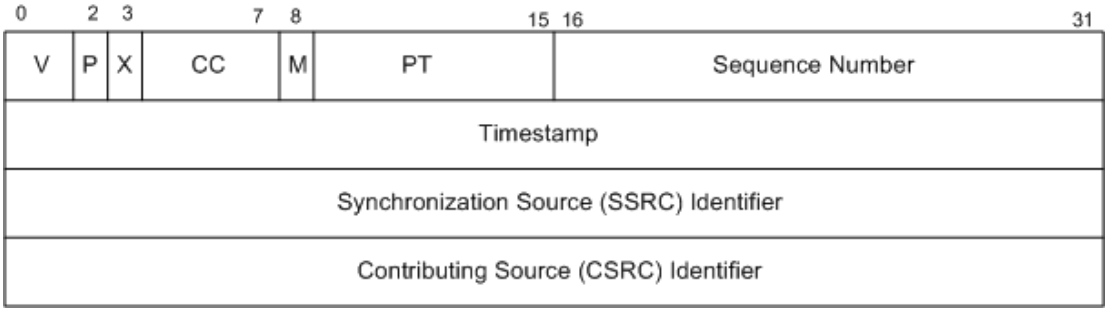

Figura 9 – Cabeçalho RTP

O RTP não garante qualidade de serviço nem realiza controlo de fluxo. Para garantir qualidade de serviço, o sistema pode recorrer às camadas inferiores que controlam a reserva de recursos, por exemplo, RSVP (*ReSerVation Protocol*). No caso do controlo de fluxo, o RTP proporciona informação de ordenação (*sequence number*) e de tempo (*timestamp*) que podem ser utilizadas pelo receptor para implementar o controlo de fluxo local. [HR03]

O protocolo RTP é complementado pelo protocolo RTCP (*Real Time Control Protocol*), pois é através deste que o RTP proporciona os serviços de controlo. O protocolo RTCP possui, entre outras, as seguintes funcionalidades [HR03]:

- É através deste protocolo que o receptor indica ao emissor a sua qualidade da recepção.
- É utilizado para sincronizar os fluxos (pois o áudio e o vídeo viajam em fluxos diferentes) ou sincronizar as fontes quando os fluxos têm diferentes origens.

- Identificação dos participantes.
- Controlo da sessão.

2.3.2 *Real Time Streaming Protocol* (RTSP)

O RTSP é um protocolo de nível da camada de aplicação, utilizado para a transmissão de fluxos de dados multimédia de forma controlada e em tempo real [HR03]. Este protocolo encontra-se definido na RFC 2326.

A sintaxe e o funcionamento deste protocolo são similares ao HTTP 1.1. O RTSP foi concebido desta forma para aproveitar a tecnologia desenvolvida para o HTTP (por exemplo, segurança). [HR03 e SRL98]

No caso do RTSP, o controlo é efectuado por um canal independente; por exemplo, o controlo RTSP pode ser feito através de uma conexão TCP, enquanto que a transmissão de dados pode ser feita através do protocolo UDP. [SRL98]

Segundo a RFC 2326 existem **mensagens de pedido** e de **resposta** (estas mensagens utilizam, por defeito, o porto 554). As **mensagens de pedido** incluem no seu cabeçalho o comando apropriado segundo a situação. Por exemplo, quando o cliente deseja indicar ao servidor que pode começar a enviar os dados, envia uma mensagem de pedido com o comando PLAY (estes comandos podem ser consultados na tabela do Anexo IV.1). As **mensagens de resposta** indicam o estado em que se encontra o pedido, este estado é indicado através de um dos códigos de estado da tabela que se encontra no Anexo IV.2.

O servidor RTSP mantém a informação de estado de cada cliente conectado a ele. Existem quatro estados, *Init*, *Ready*, *Playing* e *Recording*, os quais aplicam-se tanto para o cliente como para o servidor [HR03 e SRL98]. No Anexo IV.3 encontram-se os diagramas de estado do cliente e do servidor.

Algumas das vantagens deste protocolo são [HR03]:

- Interoperabilidade

  Permite operar entre as aplicações cliente-servidor de diferentes provedores.

- Portabilidade

  Pode ser implementado sobre qualquer plataforma. Esta propriedade e a anterior tornam o RTSP um protocolo flexível.

- Fiabilidade

Este protocolo foi elaborado sobre técnicas existentes e suficientemente provadas fazendo deste um protocolo fiável e robusto. Alguns dos protocolos considerados na elaboração do RTSP foram o RTP, HTTP, TCP e UDP.

2.3.3 Microsoft Media Server (MMS)

O protocolo MMS é utilizado para transferência de dados multimédia em tempo real. Este protocolo utiliza uma conexão TCP para controlo da sessão (as mensagens de controlo encontram-se definidas no Anexo V.1), enquanto que os dados multimédia podem ser transmitidos através desta mesma conexão TCP ou através de UDP. No caso de os dados serem transmitidos via UDP, como este não garante a entrega, o cliente pode enviar uma mensagem do protocolo MMS pedindo ao servidor que reenvie um pacote UDP, caso este não tenha sido recebido. [Microsoft08]

As mensagens do protocolo MMS utilizam o porto 1755 tanto para TCP como para UDP. [Microsoft08]

3 SISTEMA DE MONITORIZAÇÃO DISTRIBUÍDA - DTMS-P2P

3.1 Introdução

A **monitorização** é uma operação fundamental na gestão de redes que consiste em acompanhar o seu funcionamento, de forma a estudar o seu comportamento, detectar falhas, verificar configurações, entre outras.

Existem diversas ferramentas de monitorização que se podem agrupar em dois grupos genéricos:

- **Ferramentas de monitorização activa:** são aquelas que inserem tráfego na rede em questão para efectuar a análise (por exemplo, o *ping*). No caso geral, estas ferramentas destinam-se a fornecer estatísticas de desempenho da rede entre dois pontos (por exemplo, o rácio de pacotes perdidos). Estas estatísticas podem ser de sentido único ou de ida e volta (no caso de serem de *ida e volta*, as estatísticas referem-se ao tráfego que flui em ambos os sentidos). [Nog08]
- **Ferramentas de monitorização passiva:** analisam o tráfego que flui na rede em questão sem introduzir tráfego próprio (por exemplo, o *TCPDump*). Este tipo de ferramentas são utilizadas normalmente para identificar os protocolos e medir características do tráfego (por exemplo, tamanho médio dos pacotes). [Nog08]

Actualmente existem diversas ferramentas de monitorização e gestão de redes, que aplicam técnicas de monitorização activa ou passiva. Estas surgem devido à necessidade de acompanhar o funcionamento das redes. Alguns exemplos deste tipo de ferramentas são os seguintes:

- NTOP: ferramenta de monitorização e gestão de redes que pode ser utilizada em plataformas *Unix* e *Win32*. Desenvolvida inicialmente por Luca Deri e Stefano Suin na Universidade de Pisa (Itália), para detecção e resolução de problemas na rede do campus universitário, tem como funções principais a medição e monitorização de tráfego, detecção de violações de segurança de redes e optimização e planeamento de redes. [AR00]
- TCPDump: permite capturar os pacotes que passam através da interface do computador que a executa. Os pacotes podem ser visualizados em tempo real ou guardados num ficheiro para posterior análise. Funciona nos sistemas operativos *Unix* e faz uso da livraria *libpcap*. Existe uma versão para *Windows*, chamada de *WinDump* que faz uso da livraria *winpcap*.

- TSTAT: ferramenta de análise estatística de tráfego. Permite capturar e analisar o tráfego em tempo real, ou então simplesmente realizar uma análise a uma captura realizada previamente. [TNG08]
- TCPSTAT: permite realizar capturas e analisar as mesmas, permitindo também a análise de capturas previamente realizadas. O TCPSTAT produz relatórios com estatísticas geradas das capturas. É possível escolher quais as estatísticas a mostrar no relatório devolvido, tais como: bytes por segundo, pacotes por segundos, número bytes num determinado intervalo, número de pacotes num determinado intervalo de tempo, número de pacotes TCP, número de pacotes UDP, entre outros. Também é possível configurar o intervalo de tempo entre análises.

Todas estas ferramentas permitem realizar medições e testes num ponto específico da rede. Caso se deseje capturar e/ou analizar o tráfego em pontos distintos de uma rede, além de instalar computadores nestes pontos, seria necessário configurar computador a computador os testes a efectuar, pelo que uma medição distribuída torna-se muito complexa. Uma solução para este problema é o DTMS-P2P (*Distributed Traffic Measurement System with Peer-to-Peer Architecture*). Trata-se de uma ferramenta de monitorização distribuída com uma arquitectura *peer-to-peer* hierárquica, desenvolvida pela equipa de investigação do IT (Instituto de Telecomunicações) de Aveiro. O DTMS-P2P possibilita a execução remota de acções de monitorização, permite também guardar e recuperar resultados de monitorização de forma distribuída.

3.2 DTMS-P2P

O DTMS-P2P (*Distributed Traffic Measurement System with Peer-to-Peer Architecture*) é uma ferramenta de monitorização de redes que pode ser utilizada remotamente. Esta ferramenta permite a execução de diferentes módulos de monitorização em qualquer um dos nós, sendo possível observar os resultados obtidos a partir do nó remoto. O DTMS-P2P segue uma arquitectura do tipo P2P (*peer-to-peer*), garantindo assim uma maior tolerância a falhas e armazenamento distribuído da informação medida. [Vei07]

3.2.1 Requisitos

Esta ferramenta pode ser utilizada tanto no *Windows* como no *Linux*. O único requisito para a execução do DTMS-P2P é a instalação do *J2SE Java Runtime Environment (JRE) 5.0* ou superior. [Vei07]

3.2.2 Funcionamento

O DTMS-P2P tem duas entidades principais o **cliente** e o **nó**. O **cliente** é a interface entre o sistema de monitorização e o utilizador e é nesta entidade onde são realizadas as configurações das medições e obtenção de resultados. O **nó** é o elemento que realiza as medições e armazena os resultados. Este exige maior processamento e capacidade de armazenamento do que o cliente. Um ou mais nós formam um grupo e cada grupo é responsável pela monitorização de uma secção da rede, identificado com um *Group ID*, permitindo uma maior escalabilidade. [Vei07]

Um nó pode operar em dois modos diferentes, *probe* ou *super-probe*, podendo mudar dinamicamente entre um modo e o outro, de forma a se ajustar às diferentes condições e recursos da rede. [Vei07]

Um nó no modo *super-probe* é responsável pelo controlo dos outros nós conectados a ele (*probes*) e também pela comunicação entre grupos. Num grupo podem existir um ou mais nós *super-probe*, todos eles inter-ligados entre si. [Vei07]

Os clientes e as *probes* ligam-se a um nó *super-probe*. Desta forma, a inserção de novos elementos na rede é transparente para o resto da rede. Um nó operando no modo *super-probe* pode estar ligado a mais do que um cliente e mais do que um nó. É por isto que os nós que operam no modo *super-probe* necesitam de mais recursos de processamento e capacidade de armazenamento, pois estes executam mais funções do que os nós no modo *probe*. [Vei07]

Seguidamente mostra-se um exemplo que contém três grupos de medição:

- O grupo 0 tem uma *probe* (*Probe 0.1*) e um cliente (*Client 0.1*) ligados a uma *super-probe* (*Super-probe* 0.1)
- O grupo 1 contém unicamente uma *super-probe*.
- O grupo 2 tem duas super-probes, *Super-probe 2.1* e *Super-probe 2.2*; a primeira tem duas *probes* ligadas a ela (*Probe 2.1* e *Probe 2.2*) e a segunda não contém nenhuma ligação a outras *probes* ou clientes.

Como é possível observar, os nós *super-probe* encontram-se ligadas entre si.

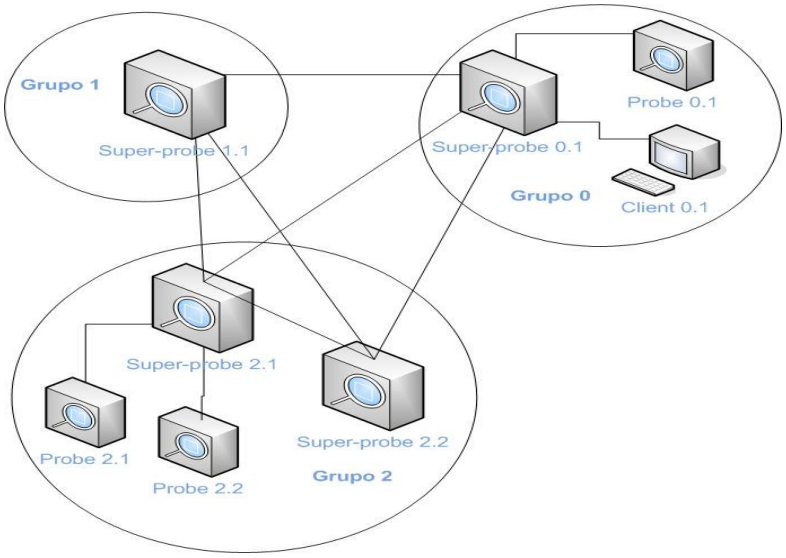

Figura 10 – Ligações entre elementos

De forma a saber quais os nós que pertencem à rede, o DTMS-P2P contém duas listas com estas informações: uma encontra-se no disco (*File of Known Nodes* (FKN)) e a outra na memória (*Cache of Known Nodes* (CKN)). Ambas contêm informações sobre o grupo de medição, o modo de operação do nó (*probe* ou *super-probe*), o endereço IP e o porto. A CKN contém um campo adicional: a média do *Round Trip Time* (RTT).

As informações contidas na CKN são utilizadas pelos novos nós que estão a ligar-se à rede, isto porque a estrutura desta lista permite identificar quais os nós que se encontram geograficamente próximos.

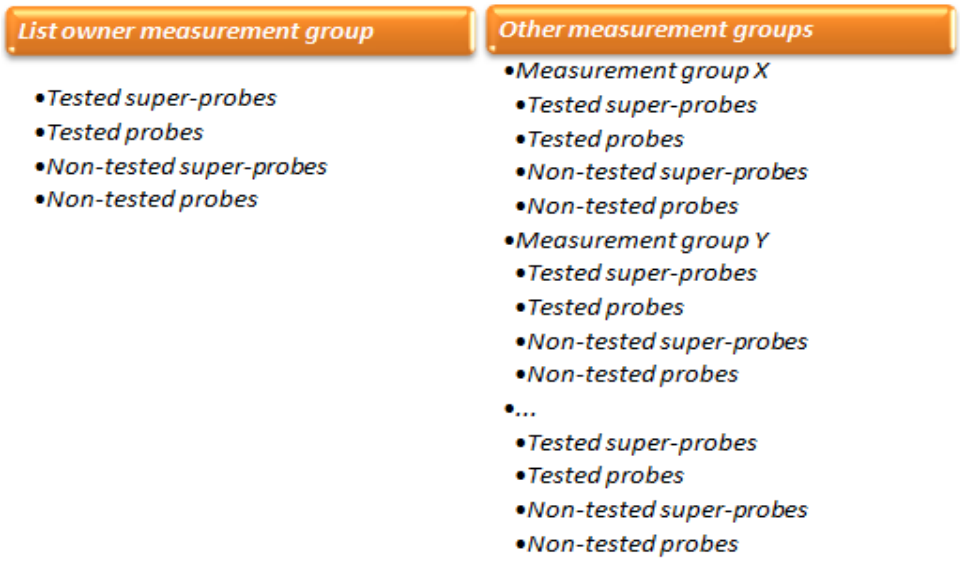

Figura 11 – Estrutura da lista *Cache of Known Nodes* (CKN) [Vei07]

A CKN está estruturada segundo três elementos: grupo de medição, modo (*probe* ou *super-probe*) e estado (testado ou não testado). Como se observa na figura 11, primeiro são divididos em dois grupos: os nós que pertencem ao grupo em questão (*List owner measurement group - ownerMG*) e os que pertencem a outros grupos (*Other measurement groups - otherMGs*). Dentro do primeiro grupo são classificados como: *super-probes* testados (*Tested super-probes*), *probes* testados (*Tested probes*), *super-probes* não testados (*Non-tested super-probes*) e *probes* não testados (*Non-tested probes*). Quanto ao segundo grupo (*otherMGs*) segue a mesma estrutura, mas primeiro são divididos segundo o grupo de medição ao qual pertencem (*Measurement group X, Measurement group Y, …*). Em cada subgrupo de *probes* e *super-probes* testados e não testados os nós são organizados segundo o RTT, deixando no fim da lista os elementos onde não é possível calcular o RTT. Desta forma, os elementos que se encontram geograficamente mais perto apareceram no início da lista, aumentando assim a probabilidade de que os elementos se liguem a outros elementos geograficamente próximos. [Vei07]

Além dos ficheiros que contêm informação sobre os nós, o DTMS-P2P tem mais dois tipos de ficheiros que guardam os dados das medições: um designado por *light data*, que contém os parâmetros de monitorização do sistema e estatísticas do tráfego relacionado com cada grupo, e outro é designado por *heavy data*, este armazena os resultados de todas as medições programadas.

3.2.3 Configuração e execução dos elementos

Após ter instalado o *J2SE Java Runtime Environment (JRE) 5.0* ou uma versão superior, é possível executar qualquer um dos elementos (nó ou cliente) do DTMS-P2P.

Antes de iniciar cada nó, deve-se actualizar o ficheiro *FileOfKnownNodes.xml* do *FilesOfKnownNodes*, isto é, escrever os dados correspondentes ao(s) nó(s) que opera(m) no modo *super-probe*. Por exemplo, considerando a figura 10, antes de iniciar qualquer nó do grupo 0, deve-se preencher o ficheiro *FileOfKnownNodes.xml* com os dados da *Super-probe 0.1,* conforme a figura 12, onde o endereço IP da *super-probe* deste grupo é o 192.168.139.1 e 22368 corresponde ao número do porto.

Depois de iniciar o nó, o *FileOfKnownNodes.xml* é actualizado com os restantes elementos do grupo. Isto é possível observar na figura 13 em que este ficheiro foi preenchido com os dados da *Probe 0.1*.

```
- <FileOfKnownNodes>
  - <GroupID id="00000000000000000000000000000000">
    - <super-probes>
      - <node>
          <IPVN>4</IPVN>
          <IP>192.168.136.1</IP>
          <port>22368</port>
        </node>
      </super-probes>
      <probes />
    </GroupID>
  </FileOfKnownNodes>
```

Figura 12 – Ficheiro *FileOfKnownNodes.xml* antes de iniciar os nós

```
- <FileOfKnownNodes>
  - <GroupID id="00000000000000000000000000000000">
    - <super-probes>
      - <node>
          <IPVN>4</IPVN>
          <IP>192.168.136.1</IP>
          <port>22368</port>
        </node>
      </super-probes>
    - <probes>
      - <node>
          <IPVN>4</IPVN>
          <IP>192.168.136.1</IP>
          <port>30929</port>
        </node>
      </probes>
    </GroupID>
  </FileOfKnownNodes>
```

Figura 13 – Ficheiro *FileOfKnownNodes.xml* após iniciar os nós

O mesmo deve ser feito para o cliente, mas nesse caso deve ser actualizado o *FileOfKnownNodes.xml* do *clientFilesOfKnownNodes*.

Seguidamente mostra-se como configurar os nós e os clientes.

3.2.3.1 Nós

A configuração do nó pode ser feita através da interface gráfica (GUI – *Graphical User Interface*) ou via consola. Atrevés da consola o comando utilizado é o seguinte:

java dtms_p2p.DTMS_P2P_Node [options]

Este comando deve ser executado no directório do DTMS-P2P. No campo *options* devem ser especificadas as opções desejadas (aquelas que não forem especificadas tomarão o valor definido por defeito). Os valores a configurar no campo *options* encontram-se definidos no manual do DTMS-P2P (versão 1.0) e também podem ser consultados através dos comandos:

java dtms_p2p.DTMS_P2P_Node -? ou *java dtms_p2p.DTMS_P2P_Node -help*

No caso da configuração ser feita através da interface gráfica (figura 14), se o sistema operativo utilizado é o *Linux*, basta executar o seguinte comando no directório do DTMS-P2P: *java dtms_p2p.DTMS_P2P_NodeGUI*

Caso o sistema operativo seja *Windows*, poderá ser executado da mesma forma ou através do executável, *DTM_P2P_NodeGUI.jar*.

Para alterar as opções de configuração desta ferramenta através da interface gráfica, deve-se recorrer ao submenu *Options…*, que se encontra no menu *Tools*. Neste submenu são apresentadas quatro colunas (figura 15):

- *Option:* são apresentadas todas as opções configuráveis.
- *Value:* é nesta coluna que podemos alterar o valor conforme o desejado.
- *Default:* mostra o valor por defeito.
- *Description*: contém uma pequena descrição de cada opção.

Após definidas as configurações, o nó pode ser iniciado através da opção *Connect* do menu *File.*

Antes de iniciar o nó, é importante ter em atenção que os módulos de medição utilizados pelo *Windows* e pelo *Linux* não são os mesmos. Assim, na configuração dos nós deve ser especificado qual o ficheiro a utilizar, pois no directório *SupportedMonitoringModules* existem dois ficheiros em XML que indicam quais os módulos de medição suportados, um para *Windows* (*SupportedMonitoringModules*) e outro para *Linux* (*SupportedMonitoringModulesLinux*). Isto pode ser configurado de duas formas:

- No ambiente gráfico: alterando o campo indicado com *–scf* do menu *Options.*
- Através das linhas de comando: incluindo no campo *[options]* a opção «*–scf (Nome_do_SupportedMonitoringModulesFile)*».

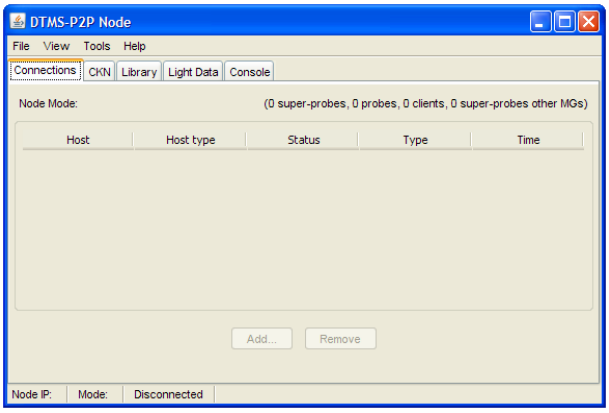

Figura 14 – DTMS-P2P Nó versão gráfica

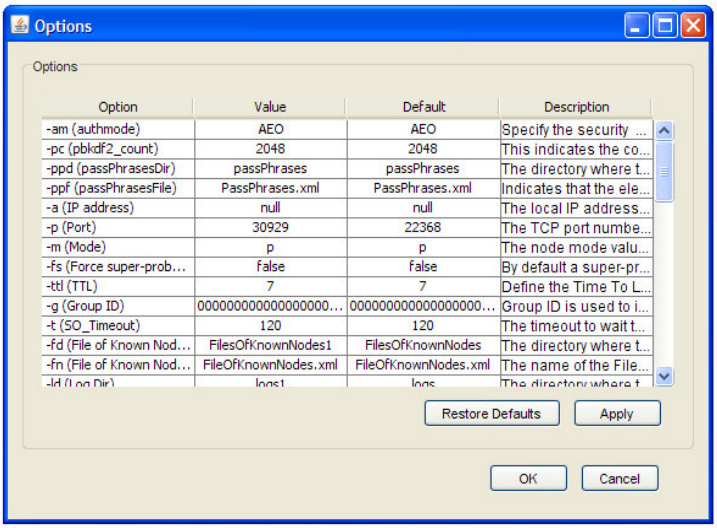

Figura 15 – Opções

3.2.3.2 Clientes

Para iniciar o cliente basta executar o seguinte comando, tanto no *Windows* como no *Linux*: *java dtms_p2p.DTMS_P2P_Client [options]*

Os diferentes valores a atribuir no campo *[options]* encontram-se especificados no manual do DTMS-P2P (versão 1.0). Tal como no caso dos nós, também é possível consultar estes valores através da consola, executando um dos seguintes comandos:

java dtms_p2p.DTMS_P2P_Client -?

ou

java dtms_p2p.DTMS_P2P_Client -help

Ao executar o cliente, é apresentado um menu (figura 16), que está dividido em três grupos: *Active Measurement* (medição activa), *Passive Measurement* (medição passiva) e *Other Options* (outras opções).

Através do primeiro submenu, *Active Measurement*, é possível configurar módulos de monitorização activa. Por defeito, o DTMS-P2P permite a execução do: *Ping*, *Trace route* e o *J-OWAMP*.

No caso do *Passive Measurement* (segundo submenu), tal como no caso anterior, é possível configurar módulos de monitorização, tais como o *TCPDump*. Note-se que em ambos os tipos de medição, o DTMS-P2P permite executar outros testes de monitorização (activa ou passiva), desde que estes sejam previamente instalados.

Por último, é apresentado um terceiro submenu, que contém diversas opções que tornam esta ferramenta muito útil. Por exemplo, é possível descarregar resultados desde qualquer um dos nós para a pasta de *Download* do cliente, é possível consultar quais os nós activos em cada grupo ou quais as módulos de monitorização instalados, etc…

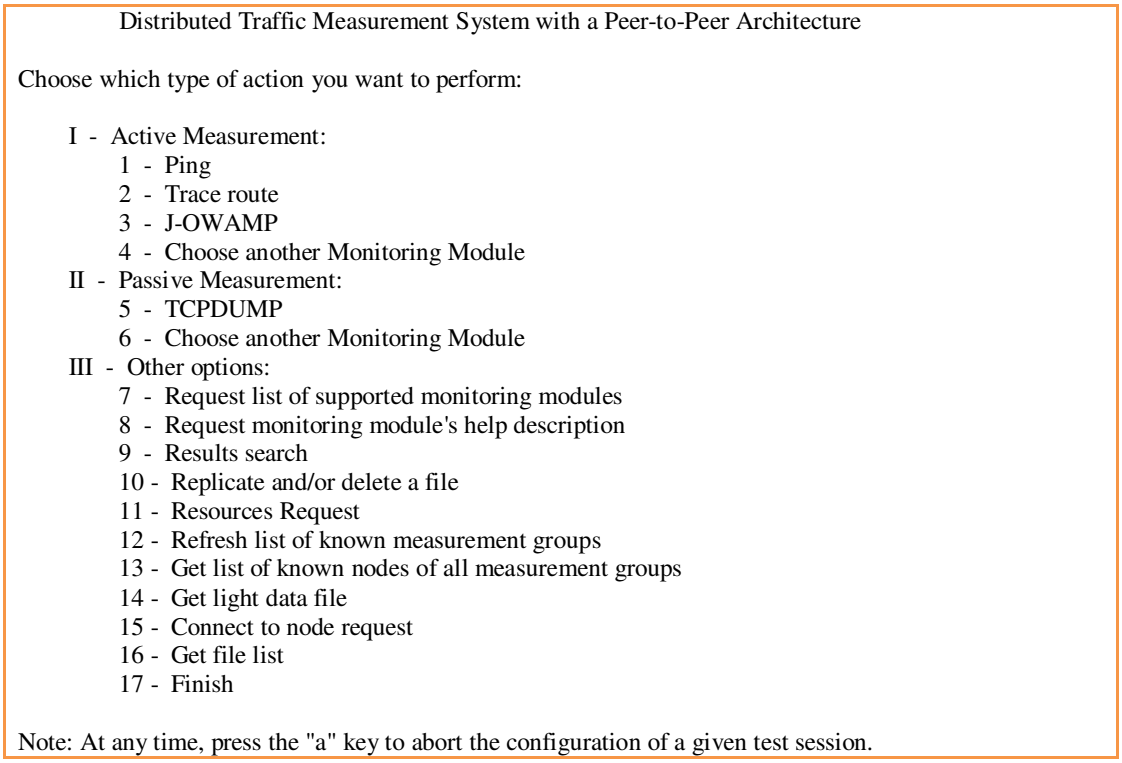

Figura 16 – Menu principal do cliente

4 CARACTERIZAÇÃO ESTATÍSTICA DO TRÁFEGO

4.1 Introdução

Após efectuadas as capturas e concluído o processo de monitorização, é possível realizar a caracterização estatística do tráfego, para poder descrever através de valores estatísticos o comportamento do mesmo.

É muito importante realizar análises deste tipo a redes com características de tempo-real (por exemplo, VoIP e videoconferência IP) para poder constatar se a configuração utilizada garante ou não qualidade de serviço.

A caracterização do tráfego pode ser realizada tanto ao **nível dos pacotes** como ao **nível da sessão** e para ambos os casos recorre-se à **estatística descritiva** (secção 4.2) e à **função de probabilidade** (secção 4.3).

4.2 Estatística descritiva

A estatística descritiva procura "sintetizar e representar de uma forma compreensível a informação contida num conjunto de dados" [GC97]. No caso de dados quantitativos[2] as técnicas de análise estatística podem dividir-se em três grupos:

- Formas de representação tabular ou gráfica de dados.
- Estatísticas.
- Representação gráfica de estatísticas.

A **representação tabular ou gráfica de dados** consiste em expor os resultados obtidos através de tabelas ou gráficos. Este tipo de representação ajuda a visualização dos dados, o que facilita a análise.

As **estatísticas** são calculadas com base nos dados obtidos possibilitando a análise e caracterização dos dados. "Com o cálculo de estatísticas, pretende-se traduzir em números aquilo que se apreende da observação de uma tabela de frequências ou de um histograma" [GC97] (uma tabela de frequências indica com que frequência se repete um determinado valor dos dados).

Entre as estatísticas existentes, para a caracterização de tráfego, recorre-se às seguintes:

- **Valor mínimo**: corresponde ao menor valor registado nos dados obtidos.

[2] Num estudo estatístico deste tipo, as variáveis são consideradas **quantitativas discretas**, pois estas são, por exemplo, a quantidade de pacotes por intervalo de tempo ou o número de bytes transmitidos num intervalo de tempo**.**

- **Média** (ou **média amostral**): "para uma amostra constituída por N dados $x_n (n=1,2,...,N)$, a média amostral é definida pela expressão:

$$\bar{x} = \frac{1}{N} \cdot \sum_{n=1}^{N} x_n \text{ "} \quad \text{[GC97]} \qquad \text{(equação 1)}$$

- **Valor máximo**: corresponde ao valor mais elevado dos dados obtidos.
- **Mediana** (ou **mediana amostral**): "considere-se que os dados que integram uma amostra são colocados por ordem crescente ou decrescente dos seus valores, formando um vector $(x_1^*, x_2^*, ..., x_N^*)$. A mediana amostral é definida nos seguintes termos:
  - se o número (N) de dados que constituem a amostra for ímpar, a mediana toma o valor do dado que, naquele vector, ocupa a posição central (isto é, $Med = x^*_{(N+1)/2}$) (equação 2)
  - se o número de dados for par, a mediana toma o valor médio dos dois termos cujas localizações no vector mais se aproximam da posição central (isto é, $Med = (x^*_{N/2} + x^*_{N/(2+1)})/2$)." [GC97] (equação 3)
- **Moda**: "trata-se de uma medida que indica o valor ou a gama de valores nos quais a concentração dos dados amostrais é máxima". [GC97]
- **Variância** (ou **variância amostral**): utilizada para descrever a dispersão de uma população de grandes dimensões da qual só temos uma amostra aleatória limitada. [GC97]

$$s^2 = \frac{1}{N-1} \cdot \sum_{n=1}^{N} \left(x_n - \bar{x}\right)^2 \quad \text{[GC97]} \qquad \text{(equação 4)}$$

- **Desvio padrão** (ou **desvio padrão amostral**): corresponde à raiz quadrada da variância amostral.

Por último, a **representação gráfica de estatísticas** consiste em representar através de gráficos os valores obtidos nas estatísticas. A diferença entre este tipo de gráficos e os gráficos referidos na representação gráfica de dados, é que na representação gráfica de estatística utilizam-se valores estatísticos obtidos a partir dos dados, e na representação gráfica de dados utilizam-se puramente os dados.

4.3 Função de probabilidade

Neste caso fala-se de **função de probabilidade** e não de **função densidade de probabilidade**, pois na caracterização de tráfego as variáveis são discretas, e só se fala em **função densidade de probabilidad**e quando a variável é contínua.

Desta forma o conceito de **função de probabilidade** é o seguinte:

"Seja Y uma variável aleatória discreta. Designa-se por **função de probabilidade** da variável Y, e denota-se por $p(y)$, à função que associa a cada valor particular que a variável pode tomar, y, a probabilidade de Y ser igual a y. Em notação simbólica, a função de probabilidade de Y é definida pela expressão

$$p(y) = \text{Probabilidade}(Y = y) = P(Y = y)\text{"} \quad \text{[GC97]} \qquad \text{(equação 5)}$$

CASOS DE ESTUDO

5. Rede IP de investigação do IT

6. DTMS-P2P

5 REDE IP DE INVESTIGAÇÃO DO IT

5.1 Introdução

O Instituto de Telecomunicações (IT) de Aveiro dispõe de uma rede IP de investigação que suporta os serviços de dados típicos da *Internet* e também os serviços de VoIP, videoconferência e emissão de vídeo. No âmbito desta dissertação será efectuada a monitorização e caracterização do tráfego gerado pelos serviços de VoIP e emissão de vídeo. Não foi possível realizar o mesmo estudo para a videoconferência pois não existiu nenhum evento que o permitisse em tempo útil.

O esquema de implementação desta rede é o seguinte:

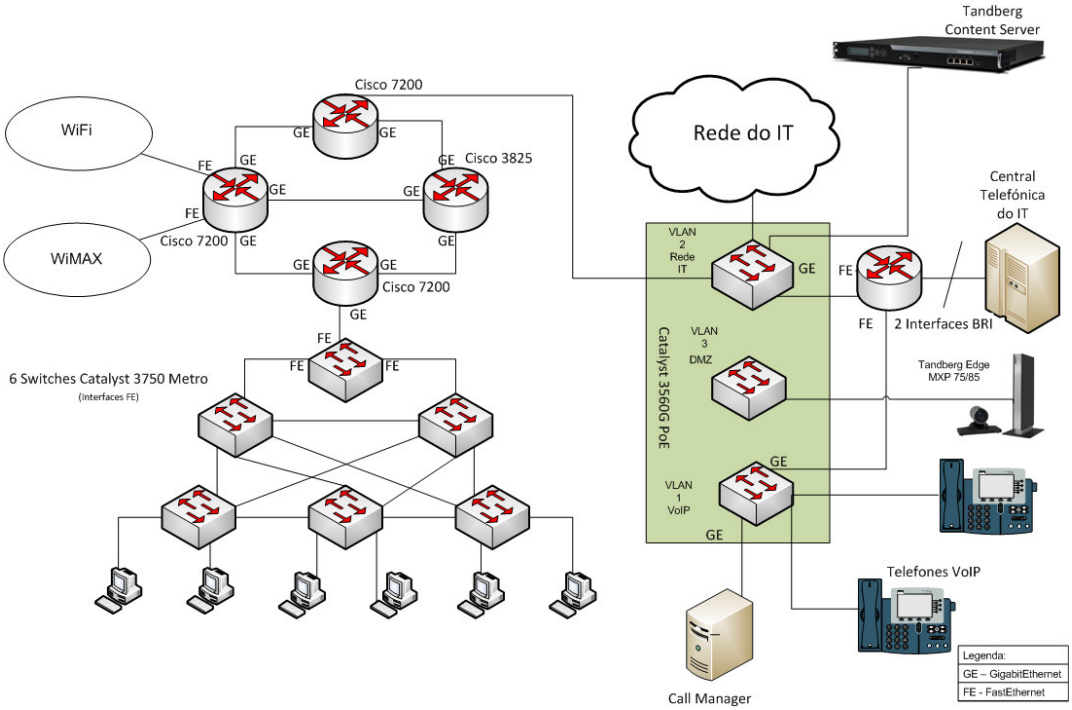

Figura 17 – Rede IP de investigação do IT [SR0?]

Seguidamente será explicado como está formada a rede que presta o serviço VoIP (secção 5.2) e a rede que presta o serviço de videoconferência e emissão de vídeo (secção 5.4).

5.2 Rede VoIP

Como se observa na figura 18, esta rede é formada, basicamente, por cinco telefones VoIP e o *CallManager*. Existe também uma central telefónica, pois é através desta que são efectuadas as chamadas para fora da rede VoIP.

Os endereços IP utilizados por cada um destes equipamentos são os seguintes:

Equipamento	Endereço IP	Máscara de rede
Telefone VoIP 1	10.0.0.86	255.0.0.0
Telefone VoIP 2	10.0.0.89	
Telefone VoIP 3	10.0.0.96	
Telefone VoIP 4	10.0.0.98	
CallManager	10.0.0.254	
Router (interface Fe0/0)	10.0.0.1	
Router (interface Fe0/1)	193.136.92.43	255.255.254.0
Telefone VoIP 5	(*)	

(*) O endereço é atribuído dinamicamente

Tabela 3 – Endereços IP dos equipamentos utilizados na rede VoIP

Como é possível observar, a rede actualmente contém cinco telefones, quatro deles estão instalados na VLAN 1 e o outro encontra-se na rede do IT (VLAN 2). Como se observa na tabela anterior, os telefones que se encontram na VLAN 1 têm endereço IP fixo, enquanto que o telefone da VLAN 2 não tem nenhum endereço atribuído porque está configurado de forma a obter endereço dinamicamente. Os telefones utilizados são de dois tipos: *Cisco IP Phone 7961 Series* e *Cisco IP Phone 7941 Series*, figuras 19 e 20 respectivamente.

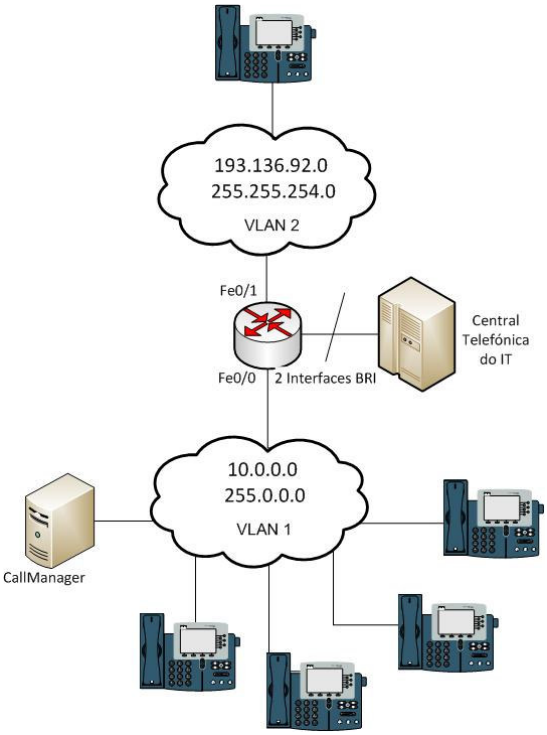

Figura 18 – Disposição dos telefones e do *CallManager* na rede

Figura 19 – *Cisco Unified IP Phone 7961G* [Cisco08b]

Figura 20 – *Cisco Unified IP Phone 7941G* [Cisco08a]

Estes telefones VoIP permitem a ligação de computadores à rede pois, como se mostra na figura seguinte, os computadores pendurados aos telefones VoIP ficam na rede do IT (VLAN 2), independentemente do telefone VoIP se encontrar na VLAN 1 ou na VLAN 2.

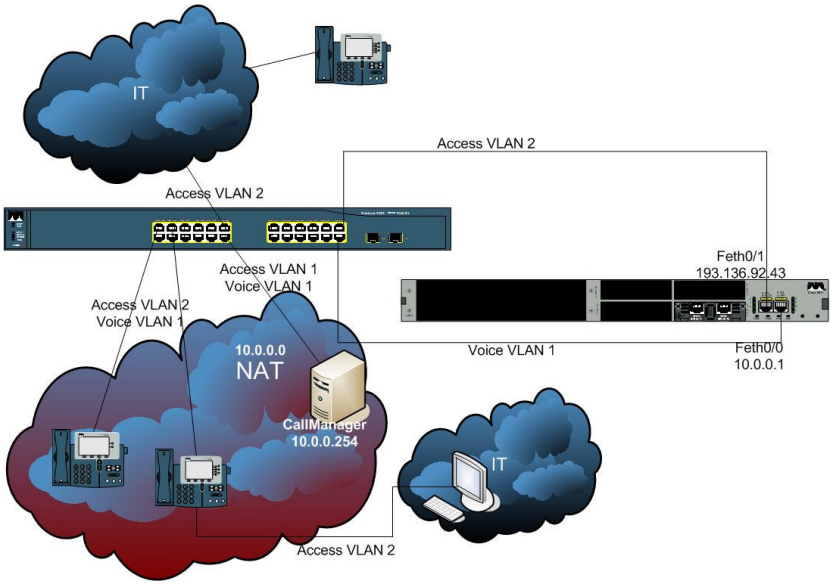

Figura 21 – Implementação do VoIP [SR0?]

Num sistema deste tipo é fundamental, além dos telefones, a existência de um *CallManager* (figura 18). O *CallManager*, neste caso *Cisco Unified CallManager* versão *5,* é o equipamento que processa as chamadas. O *CallManager,* além de intervir no estabelecimento e finalização das chamadas, também troca sinalização continuamente com os telefones de forma a saber se estes continuam pertencendo à rede e disponibiliza uma série de serviços como por exemplo *Abbreviated dialing* (permitindo a marcação rápida, isto é associando um número telefónico a um número entre 1 e 99), *Auto Answer* (após o telefone ter tocado 1 ou 2 vezes a chamada é atendida), *Hold* (permite suspender e retomar a chamada), *Redial* (marcar números recentes), entre outros. [Cisco06b]

Para poder funcionar como um sistema, o *CallManager* e os telefones têm de utilizar os mesmos protocolos, isto é, "devem falar o mesmo idioma". Os manuais dos equipamentos indicam quais os protocolos suportados, neste caso, encontram-se definidos no *Cisco Unified IP Phone Administration Guide for Cisco Unified CallManager 5.1 (SCCP)* (Anexo VI.1). Dentro do conjunto de protocolos suportados, esta rede utiliza o SCCP (*Skinny Client Control Protocol* – secção 2.2.1) e o MGCP (*Media Gateway Control Protocol* – secção 2.2.2) para sinalização, e o RTP (*Real-Time Transfer Protocol* – secção 2.3.1) para o transporte de dados. Os portos definidos para a troca de pacotes em cada um destes protocolos são os seguintes:

- Protocolo RTP: os pacotes deste protocolo são trocados pelos telefones através dos portos UDP da gama 16384 – 32767. [Cisco06a]

- Protocolo SCCP: o porto utilizado pelo *CallManager* para enviar e receber estas mensagens é o TCP 2000.
- Protocolo MGCP: este protocolo corre sobre UDP e utiliza o porto 2427 para comunicar entre o Cisco *CallManager (Call Agent)* e o *Gateway*.

No caso dos pacotes de sinalização, a troca de pacotes é feita de forma diferente dependendo do protocolo. No caso do MGCP, as mensagens são trocadas entre o *CallManager* e o *Gateway*, e no caso do protocolo SCCP os pacotes de sinalização são trocados entre os telefones e o *CallManager*. Na figura 22 é possível observar entre quais elementos da rede é efectuada a troca de pacotes de sinalização segundo o protocolo.

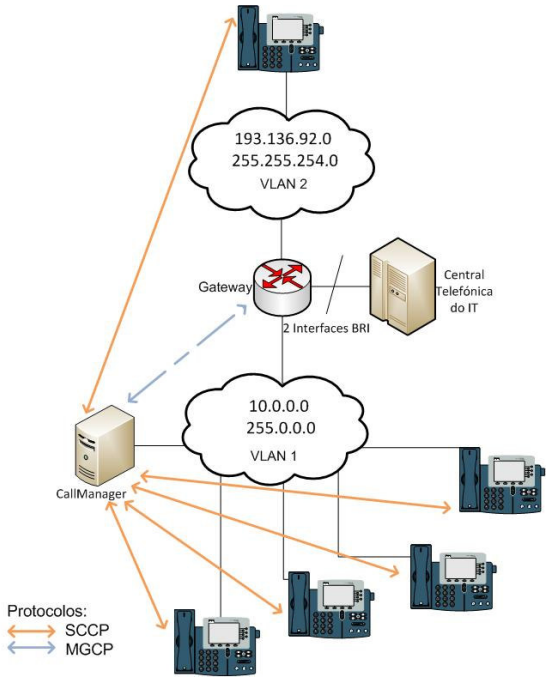

Figura 22 – Troca de pacotes de sinalização

Os pacotes de dados (protocolo RTP) são trocados directamente entre os dois telefones VoIP em questão ou entre o telefone VoIP e o *Gateway* (esta última situação acontece quando a chamada é efectuada entre um telefone da rede VoIP e um telefone da central telefónica). O *Payload Type* utilizado neste sistema pelo protocolo RTP é o PCMU (*Pulse Code Modulation* μ *-law* – Anexo III.1) o qual corresponde ao PT igual a zero. O PCMU encontra-se definido na recomendação G.711 do *ITU Telecommunication Standardization Sector* (ITU-T) e é utilizado nos sistemas telefónicos para comprimir a voz, pois o áudio é codificado em 8 *bits* por amostra e são transmitidas 8000 amostras por segundo[3], o que resulta em 64*kbps*. [Sch96 e SC03]

[3] São transmitidas 8000 amostras por segundo ($f_a = 8000_{Hz}$) de forma a ter uma margem de segurança e poder cumprir o *Teorema de Nyquist* ($f_a > 2 \times f_m$, onde f_a é a frequência de amostragem e f_m é a frequência máxima do sinal que se pretende amostrar), isto é, sendo que a frequência máxima dos canais telefónicos é de 3400*Hz* (f_m), pois esta varia entre 300 e 3400*Hz*, a frequência de amostragem deve ser no mínimo 6800*Hz* (f_a). [Sa07]

5.3 Testes efectuados à rede VoIP

5.3.1 Metodologia

Na rede VoIP foram realizados dois tipos de testes:

- Um teste para a caracterização do tráfego ao **nível do pacote** (Teste 1 – secção 5.3.1.1).
- Um conjunto de testes para a caracterização do tráfego ao **nível da sessão** (Teste 2 – secção 5.3.1.2).

Para a realização destes testes utilizaram-se as seguintes ferramentas de monitorização e gestão de redes:

- O *TCPDump* para realizar as capturas.
- O NTOP para ajudar a visualizar o funcionamento da rede, isto é, quais os *hosts* activos, protocolos utilizados, portos utilizados, tráfego gerado, entre outros.
- O *Wireshark* para visualização e análise das capturas.
- Além do *Wireshark,* elaboraram-se s*cripts* que permitissem filtrar das capturas informações importantes (por exemplo, pacotes por unidade de tempo, bytes por unidade de tempo, pacotes perdidos, entre outros), de forma a analisar as capturas.

Nos testes efectuados à rede VoIP, as capturas foram realizadas na VLAN 1 (figura 18), recorrendo a uma porta de *mirroring*[4].

5.3.1.1 Teste 1 – Caracterização do tráfego ao nível do pacote

Para efectuar a caracterização do tráfego ao nível do pacote, realizou-se uma captura "longa". Esta foi iniciada numa segunda-feira à tarde, capturando continuamente até sexta-feira à tarde (a captura iniciou-se às 14:59:13 da segunda-feira 10 de Março de 2008 e concluiu às 17:22:29 da sexta-feira 14 de Março de 2008, o que resulta num total de 354.196,03 segundos).

5.3.1.2 Teste 2 – Caracterização do tráfego ao nível da sessão

No caso da caracterização ao nível da sessão, realizaram-se vários testes de forma a comparar os diferentes tipos de telefonemas, em diferentes horas do dia. Os testes realizados foram os seguintes:

[4] Na porta de *mirroring* é possível capturar todos os pacotes que passam em determinadas portas, isto é, nesta porta são "espelhados" os pacotes das portas indicadas nas configurações.

- Chamadas entre dois telefones VoIP na mesma VLAN; neste caso, os dois na VLAN 1 (10.0.0.0/8).
- Chamadas entre dois telefones VoIP em VLANs diferentes; neste caso, um na VLAN 1 e outro na VLAN 2 (193.136.92.0/23).
- Chamadas entre um telefone VoIP (VLAN 1) e um telefóne ligado à central telefónica.

Foram efectuadas três repetições, de cada um dos tipos de chamadas, em três períodos diferentes do dia (às 11, 14 e às 17horas). Cada chamada teve uma duração aproximada de 2 minutos.

5.3.2 Resultados e conclusões

Como consequência da utilização da porta de *mirroring*, as capturas apresentaram réplicas de todos os pacotes capturados, pois esta foi configurada para espelhar todos os pacotes da VLAN 1, e como os pacotes capturados têm origem e destino portas que pertencem a equipamentos da VLAN 1, estes foram capturados duas vezes, primeiro quando estavam a sair da sua origem e uma segunda vez quando estavam a chegar à porta destino.

Esta réplica de pacotes não implicou dificuldade nenhuma no caso dos pacotes TCP, pois estes foram facilmente detectados como réplicas no *Wireshark*, permitindo assim eliminar os pacotes através da aplicação de filtros.

O mesmo não acontece para pacotes sobre UDP, pois o *Wireshark* só permitiu eliminar os pacotes duplicados em alguns casos. No caso do protocolo MGCP, foi possível eliminar os pacotes duplicados utilizando os filtros. No caso dos restantes pacotes UDP (RTP), mesmo quando estes apresentam elementos que permitem detectar pacotes duplicados (como é o caso do *sequence number*), não foi possível eliminar pacotes deste tipo recorrendo ao *Wireshark*. Recorreu-se então à realização de *scripts* que permitiram eliminar os pacotes duplicados nestas situações.

Para eliminar estes pacotes duplicados, procedeu-se da seguinte maneira:

1. Filtrou-se a captura de forma a obter unicamente pacotes dos protocolos VoIP e eliminaram-se os pacotes repetidos que o *Wireshark* permitiu eliminar.
2. Acrescentou-se à janela do *Wireshark* os seguintes elementos:
   - Número do pacote (*No.*)
   - Tempo em segundos desde o primeiro pacote recebido (*Time*)

- Endereço IP origem (*Source*)
- Porta origem (*Src Port*)
- Endereço IP destino (*Destination*)
- Porta destino (*Dest Port*)
- Protocolo (*Protocol*)
- Tamanho do pacote (*Pkt Length*)
- Informação do pacote (*Info*)

Estes campos foram acrescentados de forma a tornar mais visível às informações sobre os pacotes (na seguinte figura é possível observar a disposição dos dados de cada pacote linha a linha).

No.	Time	Source	Src port	Destination	Dest port	Protocol	Pkt Length	Info
1	0.000000	10.0.0.89	49670	10.0.0.254	2000	SKINNY	66	KeepAliveMessage
2	0.000336	10.0.0.254	2000	10.0.0.89	49670	SKINNY	66	KeepAliveAckMessage
3	0.016131	10.0.0.89	49670	10.0.0.254	2000	TCP	64	49670 > cisco-sccp [ACK] Seq=13 Ack=13 Win=8192 Len=0
4	5.980212	10.0.0.96	49445	10.0.0.254	2000	SKINNY	66	KeepAliveMessage
5	5.981637	10.0.0.254	2000	10.0.0.96	49445	SKINNY	66	KeepAliveAckMessage
6	5.996426	10.0.0.96	49445	10.0.0.254	2000	TCP	64	49445 > cisco-sccp [ACK] Seq=13 Ack=13 Win=8192 Len=0
7	6.046718	10.0.0.86	49502	10.0.0.254	2000	SKINNY	66	KeepAliveMessage
8	6.046729	10.0.0.254	2000	10.0.0.86	49502	SKINNY	66	KeepAliveAckMessage
9	6.063719	10.0.0.86	49502	10.0.0.254	2000	TCP	64	49502 > cisco-sccp [ACK] Seq=13 Ack=13 Win=8192 Len=0
10	6.496899	193.136.93.104	52807	10.0.0.254	2000	SKINNY	66	KeepAliveMessage

Figura 23 – Janela do *Wireshark*

3. Utilizando o *Wireshark,* converteu-se o ficheiro das capturas (*.cap*) num ficheiro do tipo CSV (*Comma Separated Values summary*), isto é, um ficheiro em que cada elemento do pacote (Número, endereço IP, entre outros) é separado por vírgulas.
4. Desenvolveu-se um *script* que permite eliminar os pacotes duplicados (*eliminarDupVoIP.sh* – Apêndice A.1), neste caso os pacotes de dados.

Desenvolveram-se ainda outros *scripts* que permitem determinar informações importantes das capturas, para o estudo do tráfego da rede:

- O *script encontraProtocolo.sh* (Apêndice A.2) cria ficheiros com os pacotes dos diferentes protocolos utilizados (os de sinalização e o de dados).
- O *script editcap.sh* (Apêndice A.3) cria um directório com diversos ficheiros, onde cada um contém informação referente aos pacotes, isto é, o número do pacote, tempo, endereço IP origem, porto origem, endereço IP destino, porto destino, protocolo, tamanho do pacote e informação do mesmo.
- Com os ficheiros obtidos através do *script editcap.sh,* é possível calcular os pacotes transmitidos em *X* segundos, isto através do *script contadorPacVoip.sh* (Apêndice A.4).
- Utilizando o ficheiro que contém o número de pacotes transmitidos por intervalos de tempo (*PacINTtempo.txt*) e o ficheiro que contém o tamanho de cada pacote

(*pktLength.txt*), é possível calcular o número de bytes transmitidos nesse intervalo de tempo, através do *script contadorBytesVoip.sh* (Apêndice A.5).

- Adicionalmente, desenvolveram-se *scripts* para o cálculo do *jitter* (*CalculaJitter.sh* – Apêndice A.7) e dos pacotes perdidos (*contaPacPerdidos.sh* – Apêndice A.8). Estes recorrem às informações contidas nos pacotes RTP (*info*), as quais devem ser previamente editadas recorrendo ao *script editRTPinfo.sh* (Apêndice A.6).

5.3.2.1 Teste 1 – Caracterização do tráfego ao nível do pacote

Antes de efectuar a caracterização do tráfego, recorreu-se aos *scripts* enunciados anteriormente, de forma obter informações sobre a captura. Além destes *scripts*, neste caso, utilizaram-se os seguintes dois *scripts* para analisar as chamadas:

- *procura_Mark.sh* (Apêndice A.9): permite detectar o inicio de cada chamada, permitindo também a contabilização das chamadas efectuadas ao longo da captura.
- *editchamadas.sh* (Apêndice A.10): determina algumas informações sobre as chamadas (por exemplo, os endereços IP origem e destino).

Através do *Wireshark*, foi possível visualizar os endereços IP dos elementos intervenientes na captura. Na tabela seguinte (que apresenta os endereços detectados) é possível verificar que os endereços do *CallManager* e dos telefones VoIP 1 a 4 coincidem com os endereços da tabela 3 (secção 5.2). Quanto ao telefone VoIP 5, observa-se que este ao longo da captura utilizou três endereços IP diferentes, pois como foi referido anteriormente, este telefone obtém o endereço IP dinamicamente.

Elemento	Endereço IP
CallManager	10.0.0.254
Telefone VoIP 1	10.0.0.86
Telefone VoIP 2	10.0.0.89
Telefone VoIP 3	10.0.0.96
Telefone VoIP 4	10.0.0.98
Telefone VoIP 5 (telefone VoIP da rede do IT)	193.136.93.104
	193.136.93.161
	193.136.93.200

Tabela 4 – Tabela de endereços IP dos telefones e do *CallManager*

Além dos endereços IP, verificou-se também quais os protocolos intervenientes nesta captura. Como era de esperar, foram detectados os protocolos RTP, SCCP e MGCP que coincidem com os protocolos referidos na secção 5.2.

Além da caracterização do tráfego, efectuou-se uma análise às chamadas efectuadas, que se encontra subdividida nas duas partes seguintes (secções 5.3.2.1.1 e 5.3.2.1.2).

5.3.2.1.1 Parte I - Caracterização do tráfego

Na figura seguinte é possível observar o comportamento do tráfego VoIP, no caso dos pacotes (vermelho) e dos bytes (azul) transmitidos minuto a minuto ao longo da captura. Ao observar o comportamento de ambos os gráficos verifica-se que existe uma relação directa entre ambos, esta é o tamanho médio dos pacotes durante esse minuto.

Fala-se em tamanho médio dos pacotes durante um minuto em específico, pois o tamanho dos pacotes não é constante. Por exemplo, na seguinte figura:

- No minuto 311: verifica-se que ao longo deste minuto são transmitidos 38*pacotes*, o que corresponde neste caso a 2584*bytes*; assim, neste minuto o tamanho médio dos pacotes transmitidos é de 68*bytes*.

  No minuto 1541: ao longo deste minuto são transmitidos 6037*pacotes*, o que corresponde neste caso a $1{,}287\times10^{6}$ *bytes*. Logo, ao longo deste minuto o tamanho médio dos pacotes transmitidos é de 213,1854*bytes*. Como era de esperar, nesta situação o tamanho médio dos pacotes é superior ao caso anterior, pois neste caso, estamos na presença de uma chamada em que a maior parte dos pacotes existentes são de dados (que na maior parte dos casos são de maior tamanho que os de sinalização).

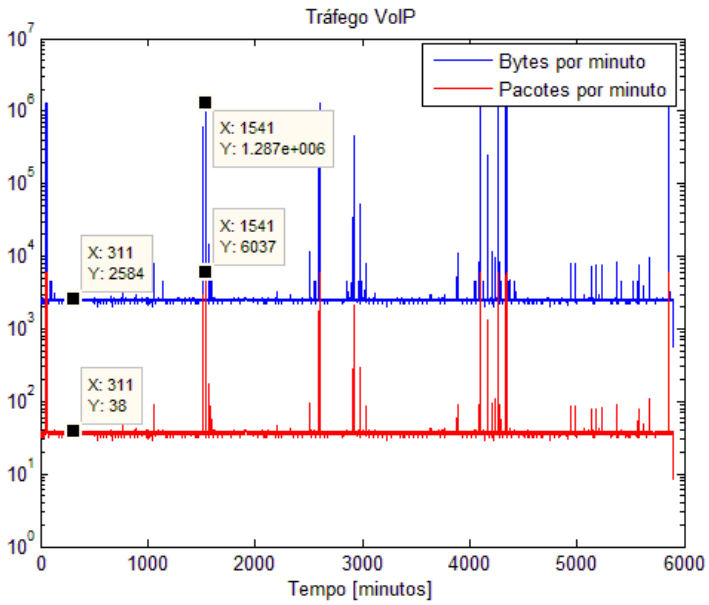

Figura 24 – Gráfico do tráfego VoIP

Como é possível constatar neste gráfico, o tráfego VoIP na maior parte do tempo apresenta um comportamento aproximadamente constante em torno aos 38*ppm* (2584*Bpm*), existindo também alguns picos. Estes instantes onde aumenta a intensidade do tráfego acontecem quando há uma chamada.

Os 38 pacotes por minuto devem-se ao facto de existirem pacotes de sinalização periódicos. A figura 25 ilustra o que acontece com os pacotes sinalização periódicos ao longo de um minuto. Como já foi referido anteriormente (secção 2.2.1), cada um dos cinco telefones envia de 30 em 30 segundos pacotes *KeepAlive* do protocolo SCCP, o que implica que cada telefone envia, durante um minuto, dois pacotes deste tipo. Visto que a troca de mensagens *KeepAlive* é composta por três pacotes (figura 3, secção 2.2.1), durante um minuto cada telefone dá origem a seis pacotes de sinalização e assim os cinco telefones dão origem a 30 pacotes SCCP. Os restantes oito pacotes são do protocolo MGCP: de 15 em 15 segundos é enviado um pacote de notificação do *router* para o *CallManager* e este envia uma mensagem de resposta (figura 4, secção 2.2.2).

Os gráficos da função probabilidade dos pacotes e dos bytes (figuras 26 e 27 respectivamente) confirmam a existência dos pacotes de sinalização, pois pelo facto de serem periódicos aparecem com frequência ao longo da captura e, logo, apresentam uma elevada probabilidade de acontecer. É por isto que o valor que apresenta maior probabilidade é 38*ppm*, que corresponde aos 38*pacotes* de sinalização da figura 25. No caso dos bytes este valor é representado por 2584*bytes*.

Os 35*ppm* (2338*Bpm*) também representam pacotes de sinalização periódicos. Estes 35 *pacotes* acontecem em situações em que, por exemplo, um dos telefones não conseguiu enviar os dois *keepAlives* ao longo do mesmo minuto e assim o telefone envia menos três pacotes que os outros, resultando num total de 35*pacotes*.

Como é possível observar nestes gráficos (figuras 26 e 27) a probabilidade de serem enviados 38 e 35*ppm* é de 69,008% e 23,274%, respectivamente. Os restantes 7,718% encontram-se distribuídos ao longo do eixo dos *xx*, como se pode observar no *zoom* efectuado ao eixo dos *yy* (figura 26). O mesmo acontece no caso dos bytes.

Além disto, através dos gráficos da função de probabilidade, é possível concluir que podem ser enviados ao longo de um minuto até 6107*pacotes,* o que em bytes corresponde a aproximadamente 1,3*MBpm*. Estes valores máximos acontecem ao longo das chamadas, os quais correspondem aos picos atingidos no gráfico da figura 24.

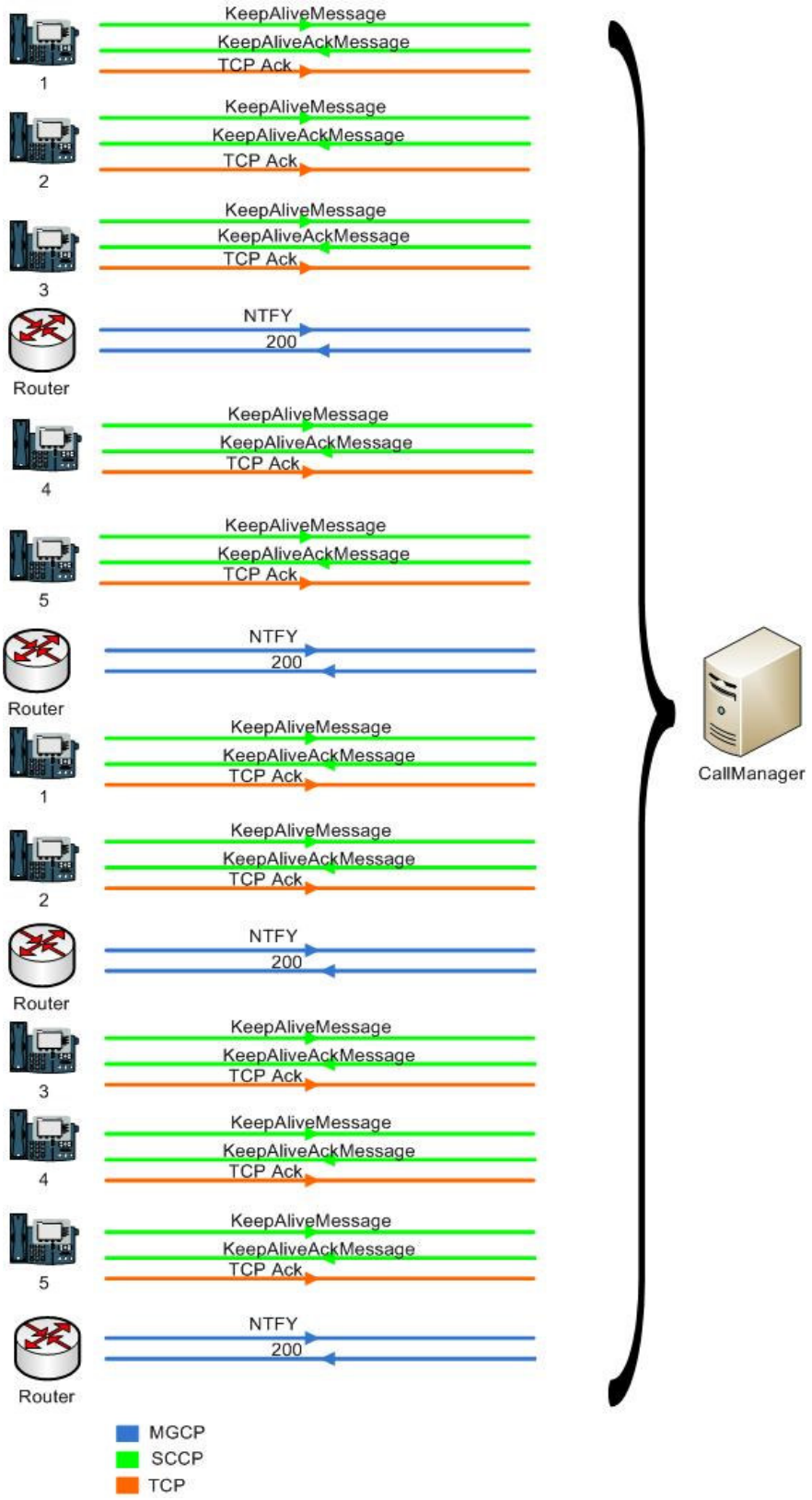

Figura 25 – Exemplo dos pacotes de sinalização enviados ao longo de um minuto

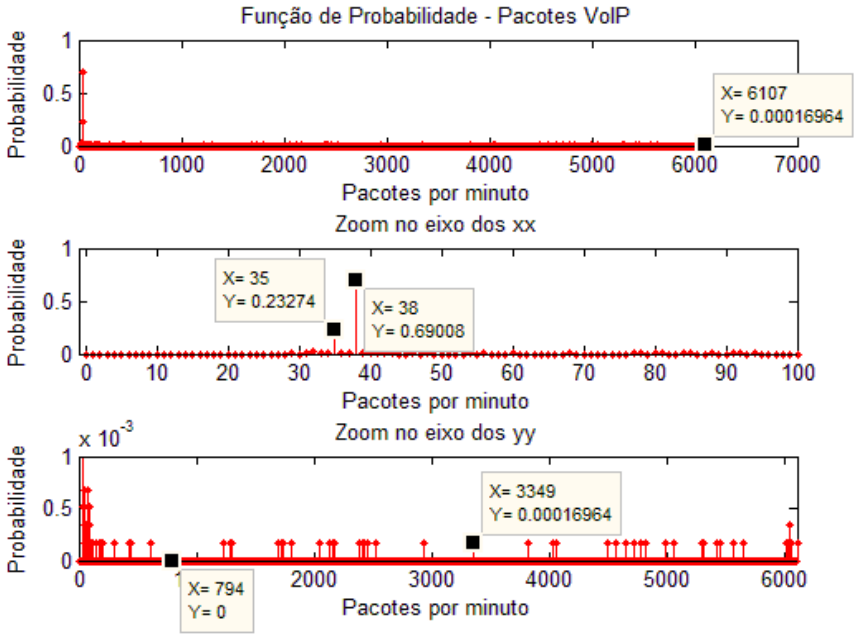

Figura 26 – Função de probabilidade do tráfego VoIP (pacotes por minuto)

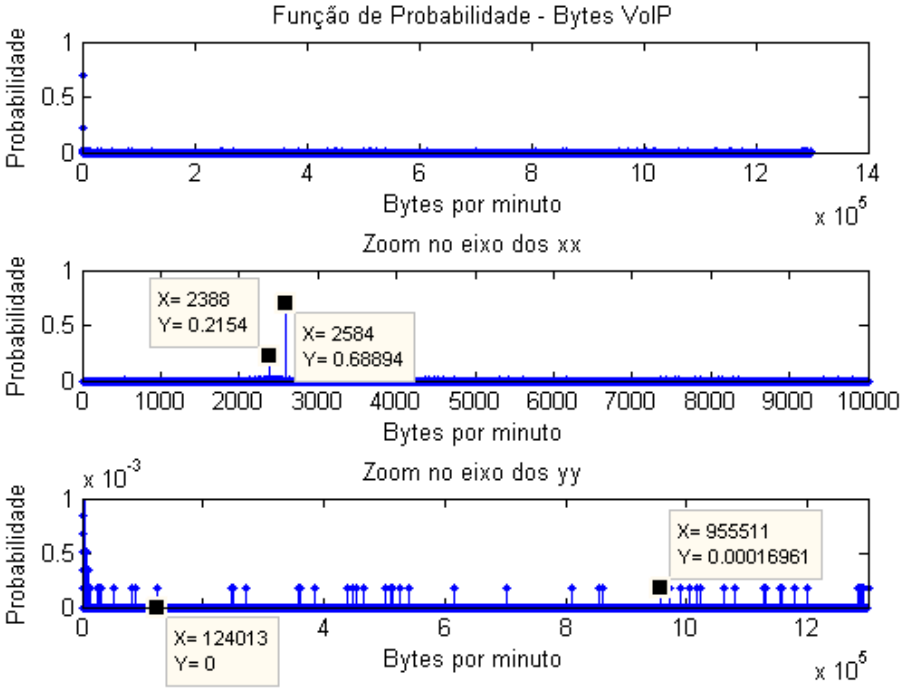

Figura 27 – Função de probabilidade do tráfego VoIP (bytes por minuto)

Na seguinte tabela é possível verificar que a mediana e a moda têm o mesmo valor. Isto deve-se ao facto de a moda repetir-se muito mais do que os outros valores (isto pode verificar-se no gráfico das figura 26 e 27), e ao organizar os valores obtidos por ordem

crescente (ou decrescente) o valor central, ou mediana, corresponde ao mesmo valor que mais se repete (moda).

	Tráfego VoIP						
	Valor Máximo	Média	Valor Mínimo	Mediana	Moda	Variância	Desvio Padrão
Ppm	6107	82,23	29	38	38	223008,08	472,24
Bpm	1299743	12144,71	1996	2584	2584	10195371210	100972,13

Tabela 5 – Estatísticas do tráfego VoIP em pacotes e bytes por minuto

Na captura realizada, verificou-se que as percentagens dos pacotes de **dados** e **sinalização** foram muito próximas, sendo que os dados representaram 54,36% dos pacotes VoIP e a sinalização 45,64%. Neste caso a percentagem de pacotes de dados ultrapassou a percentagem de pacotes de sinalização, mesmo quando os pacotes de sinalização tenham pacotes periódicos e os de dados aparecem unicamente durante as chamadas. Isto acontece porque os pacotes de dados ao longo de uma chamada aparecem com maior intensidade do que os pacotes de sinalização.

No gráfico seguinte é possível observar que existe uma diferença mais notável quando comparados os bytes de dados (78,74%) e os de sinalização (21,26%), pois os pacotes de dados, além de existirem em maior quantidade, geralmente são de maior tamanho do que os de sinalização.

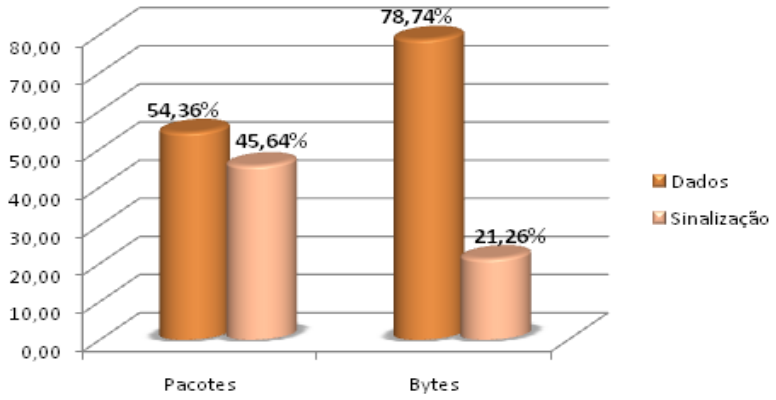

Figura 28 – Percentagem de tráfego de dados e de sinalização VoIP

Dentro dos pacotes de sinalização, neste caso, verificou-se que existe maior percentagem de pacotes SCCP (78,62%). Isto acontece porque estes pacotes são trocados sempre no estabelecimento e finalização das chamadas, e além disso existem mensagens periódicas (*KeepAlive*) que são trocadas entre cada um dos telefones e o *CallManager*. No caso do MGCP, os pacotes são enviados no estabelecimento e finalização de chamadas entre um telefone VoIP e um telefone da central telefónica. Existem também pacotes

periódicos (*NTFY*), mas, neste caso, mesmo sendo mais frequentes do que os pacotes periódicos do SCCP (as mensagens *KeepAlives* são enviadas de 30 em 30 segundos e as *NTFY* de 15 em 15 segundos), estes são trocados unicamente entre o *router* e o *CallManager*. Ao observar os valores dos bytes de cada protocolo, a situação mantém-se.

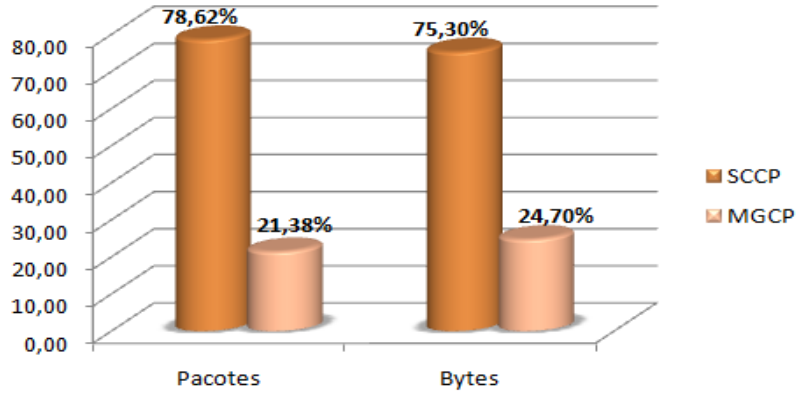

Figura 29 – Percentagem dos pacotes e bytes dos protocolos de sinalização

Caracterização do tráfego - Dados:

Através dos gráficos do tráfego de dados que se encontram no Apêndice B.1, e dos gráficos da função de probabilidade (figuras 30 e 31), verificou-se que os pacotes de dados aparecem em grande quantidade, mas em espaços de tempo curtos (surgem unicamente ao longo das chamadas).

Como se observa nos gráficos que contêm o comportamento do tráfego ao longo da chamada, no Apêndice B.1, na maior parte do tempo não existem pacotes de dados. Isto pode-se confirmar através dos gráficos da função de probabilidade, pois tanto no caso dos pacotes transmitidos por minuto como no caso dos bytes, o zero representa o valor com maior probabilidade de acontecer, sendo em ambos os casos de 98,846%. Além disto, na tabela 10, verifica-se que o valor que mais se repete (moda) é 0*ppm* e 0*Bpm*.

Como se observa na tabela 6, no máximo são enviados 6059*ppm* e no caso dos bytes aproximadamente 1,3*MBpm*. Nos gráficos seguintes é possível verificar que dentro da quantidade de pacotes de dados transmitidos por minuto, os que apresentam maior probabilidade são os que se encontram em torno aos 6000*ppm*, o mesmo acontece nos bytes, pois os de maior probabilidade encontram-se à volta dos 1,3*MBpm*.

A explicação para a existência de uma maior probabilidade na transmissão de 6000*ppm*, é a utilização do *codec* G.711 o qual proporciona um fluxo de dados de 64*kbps*. Os 6000 pacotes por minuto correspondem a 100*pps* para ambos os fluxos, o que resulta em 50*pps*

para um dos fluxos, visto que os pacotes de dados são de 214*bytes* dos quais 54*bytes* são cabeçalho (isto é, os dados são os restantes 160*bytes*), durante um segundo são enviados 64*kbits* ($50pps \times 160bytes \times 8\frac{bits}{bytes} = 64\,kbps$).

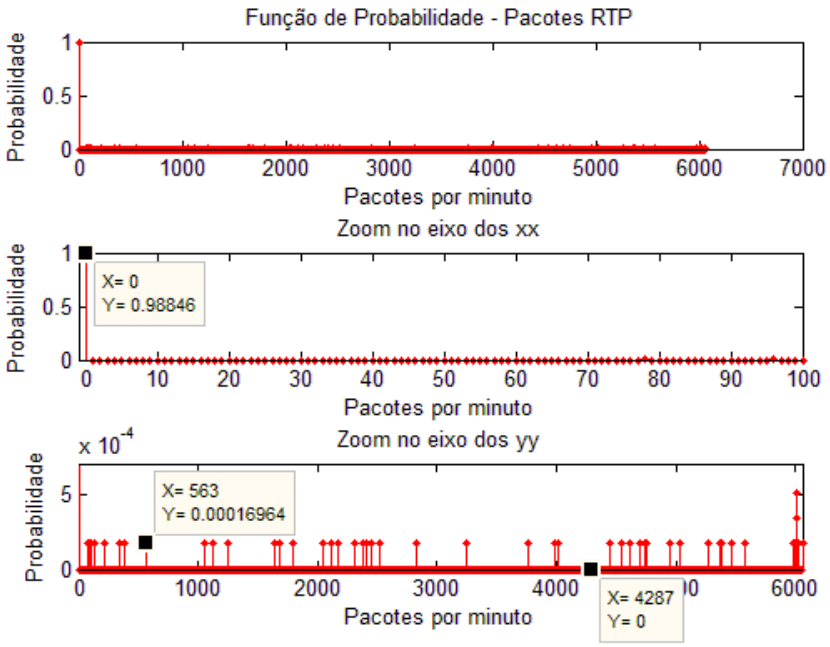

Figura 30 – Função de probabilidade do tráfego de dados (pacotes por minuto)

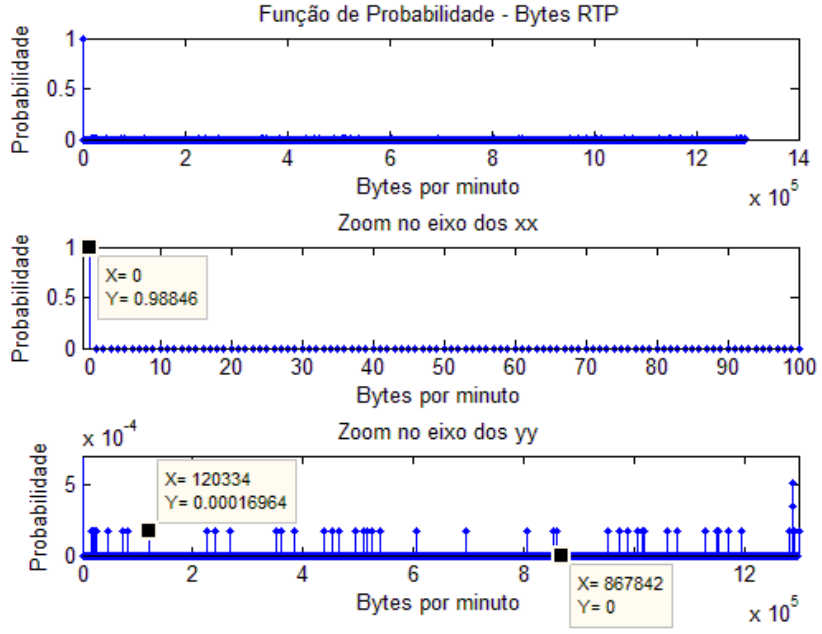

Figura 31 – Função de probabilidade do tráfego de dados (bytes por minuto)

	Tráfego VoIP – Dados						
	Valor Máximo	Média	Valor Mínimo	Mediana	Moda	Variância	Desvio Padrão
ppm	6059	44,83	0	0	0	223095,64	472,33
Bpm	1296038	9589,38	0	0	0	10208204078	101035,66

Tabela 6 – Estatísticas do tráfego de dados VoIP em pacotes e bytes por minuto

Caracterização do tráfego - Sinalização:

No caso da sinalização, contrariamente aos dados, ao longo da captura há sempre pacotes de sinalização presentes (figuras 32 e 33), pois a probabilidade de 0*ppm* e 0*Bpm* é nula. Este facto é também comprovado pela tabela 7, visto que os valores mínimos para os pacotes e bytes transmitidos são diferentes de zero.

Através dos gráficos confirma-se, também, a existência de pacotes de sinalização periódicos, o que se traduz num comportamento constante em torno aos 38*ppm* (aproximadamente 2584*Bpm*) no gráfico do tráfego de sinalização (figura 111, Apêndice B.2). A presença de pacotes de sinalização periódicos também pode confirmar-se através dos gráficos da função probabilidade (figuras 32 e 33), pois o valor com maior probabilidade de acontecer é 38*ppm*.

Além dos pacotes de sinalização periódicos, existem outros pacotes de sinalização que aparecem em situações específicas (por exemplo, no estabelecimento de uma chamada). É por isto que se observa um aumento na taxa de pacotes (bytes) de sinalização transmitidos em determinados instantes de tempo (Apêndice B.2). Este comportamento também se traduz na possibilidade de existirem mais de 38*pacotes* (2584*bytes*) ao longo de um minuto, como se observa no *zoom* efectuado no eixo dos *yy* dos gráficos da função de probabilidade dos pacotes e dos bytes.

No caso da sinalização, os valores máximos são inferiores aos valores máximos obtidos para os dados (tabelas 6 e 7) pois, como já foi referido anteriormente, ao longo de uma chamada a intensidade do fluxo aumenta, e durante a mesma o tráfego é essencialmente composto por pacotes de dados.

	Tráfego VoIP – Sinalização						
	Valor Máximo	Média	Valor Mínimo	Mediana	Moda	Variância	Desvio Padrão
ppm	239	37,52	3	38	38	34,02	5,83
Bpm	21343	2581,49	218	2584	2584	365961,87	604,95

Tabela 7 – Estatísticas do tráfego de sinalização VoIP em pacotes e bytes por minuto

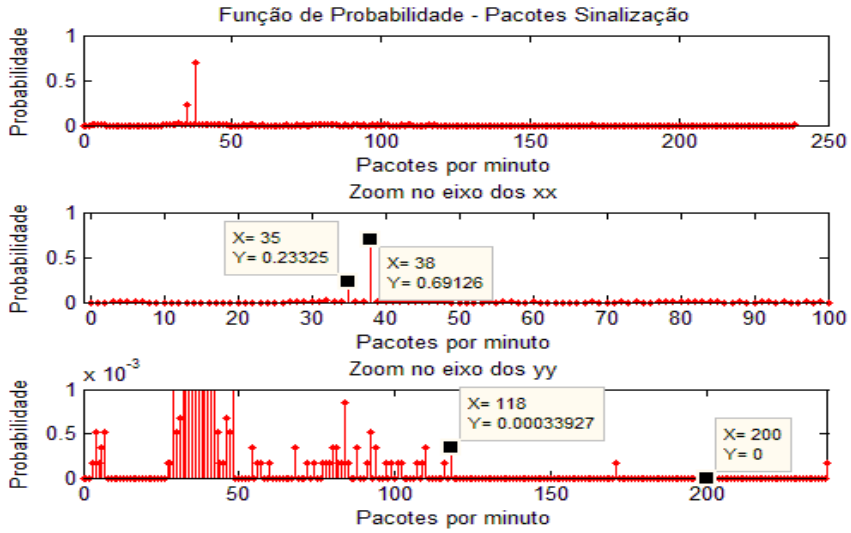

Figura 32 – Função de probabilidade do tráfego de sinalização (pacotes por minuto)

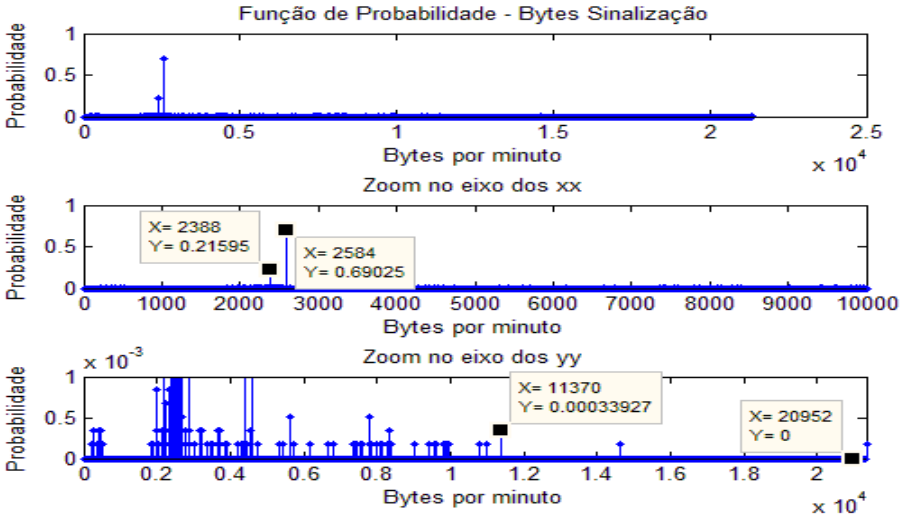

Figura 33 – Função de probabilidade do tráfego de sinalização (bytes por minuto)

Como já foi referido anteriormente, nesta rede são utilizados dois tipos de protocolos de sinalização: **SCCP** e **MGCP**.

No caso do protocolo **MGCP**, verifica-se através dos gráficos da função de probabilidade a existência de pacotes MGCP periódicos, pois o valor com maior probabilidade de acontecer corresponde a 8*ppm* (632*Bpm*). Estes oito pacotes enviados ao longo de um minuto correspondem aos pacotes *NTFY* trocados entre o *router* e o *CallManager* (figura 25).

Como se pode verificar na figura 35 (gráfico da função de probabilidade dos pacotes MGCP), existe uma probabilidade de 98,898% de serem enviados 8 pacotes por minuto, e, devido a esta probabilidade ser muito elevada, os valores da moda e a mediana correspondem também a 8 pacotes por minuto (tabela 8). No caso dos bytes por minuto, a moda e a mediana também coincidem, pois a probabilidade de serem enviados 632*Bpm* é elevada (98,898%) quando comparada com as restantes probabilidades.

Existem também situações em que são enviados mais do que 8*ppm* (Apêndice B.2), tais como por exemplo no estabelecimento ou finalização de uma chamada entre um telefone da rede VoIP e um telefone da central telefónica. Os valores variam entre 2 e 48 pacotes por minuto (158 e 5973 bytes por minuto). O valor mínimo acontece devido há existência de perdas de pacotes *NTFY*, pois era suposto ao longo de um minuto serem trocados à volta de 8 pacotes.

Na seguinte figura é possível observar uma situação de perda destes pacotes. Como é possível observar, o *Transaction ID* (identifica o número da transacção, neste caso encontra-se a amarelo) é acrescentado de um em um conforme são enviadas as mensagens *NTFY* e verifica-se que entre o 156847929 e o 15684734 perderam-se quatro pacotes.

No. .	Time	Source	Src port	Destination	Dest port	Protocol	Pkt Length	Info
40731	2779.359895	10.0.0.1	2427	10.0.0.254	2427	MGCP	98	NTFY 156847928 *@Cisco2811GW.av.it.pt MGCP 0.1
40732	2779.361449	10.0.0.254	2427	10.0.0.1	2427	MGCP	60	200 156847928
43748	2794.361257	10.0.0.1	2427	10.0.0.254	2427	MGCP	98	NTFY 156847929 *@Cisco2811GW.av.it.pt MGCP 0.1
43749	2794.361266	10.0.0.254	2427	10.0.0.1	2427	MGCP	60	200 156847929
48868	2869.365580	10.0.0.1	2427	10.0.0.254	2427	MGCP	98	NTFY 156847934 *@Cisco2811GW.av.it.pt MGCP 0.1
48869	2869.366025	10.0.0.254	2427	10.0.0.1	2427	MGCP	60	200 156847934
51879	2884.366938	10.0.0.1	2427	10.0.0.254	2427	MGCP	98	NTFY 156847935 *@Cisco2811GW.av.it.pt MGCP 0.1
51880	2884.366947	10.0.0.254	2427	10.0.0.1	2427	MGCP	60	200 156847935

Figura 34 – Exemplo de perda de pacotes NTFY

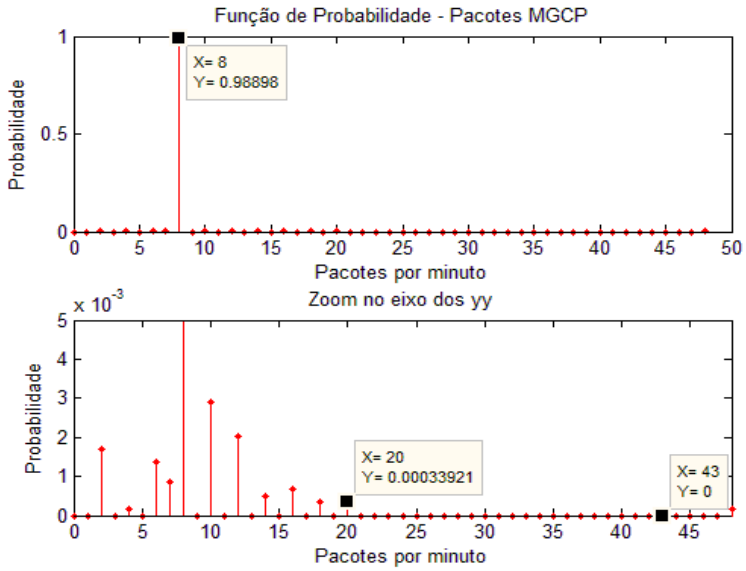

Figura 35 – Função de probabilidade dos pacotes por minuto do protocolo MGCP

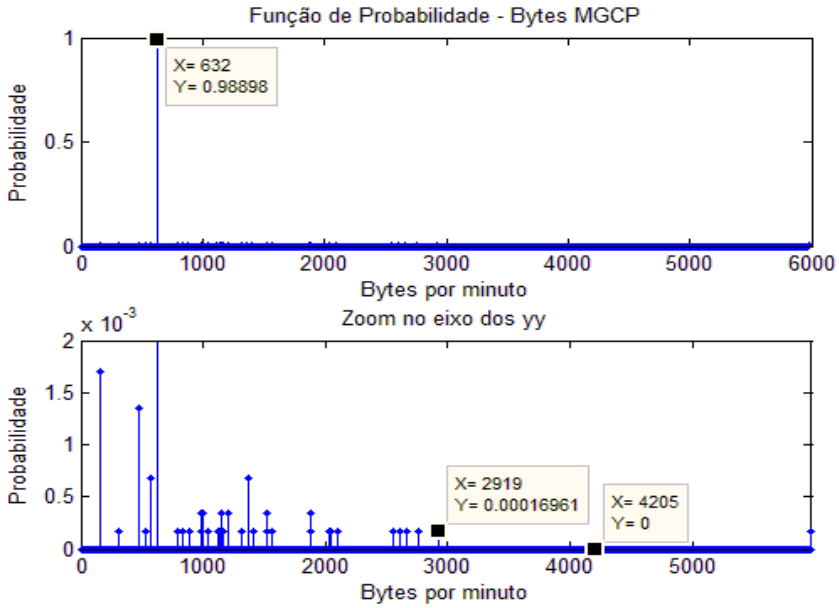

Figura 36 – Função de probabilidade dos bytes por minuto do protocolo MGCP

	Tráfego VoIP – Sinalização: Protocolo MGCP						
	Valor Máximo	Média	Valor Mínimo	Mediana	Moda	Variância	Desvio Padrão
ppm	48	8,02	2	8	8	0,52	0,72
Bpm	5973	637,69	158	632	632	12401,20	111,36

Tabela 8 – Estatísticas do tráfego do protocolo MGCP

No caso dos pacotes do protocolo SCCP, assim como acontece com o MGCP, através dos gráficos do tráfego SCCP (Apêndice B.2) e função de probabilidade (figuras 37 e 38) é possível mostrar a existência de pacotes periódicos. Neste caso, e como era de esperar, o valor que mais se repete é 30*ppm* (1952*Bpm*). Estes 30 pacotes correspondem aos *keepAlive* enviados pelos cinco telefones ao longo de um minuto (figura 25).

Existem situações em que os cinco telefones não conseguem enviar os pacotes *KeepAlive* as duas vezes durante o mesmo minuto. Por exemplo, se um telefone enviar os *KeepAlive* pela primeira vez durante um minuto X após os primeiros trinta segundos, o segundo grupo de pacotes *KeepAlive* não poderá ser enviado dentro do minuto X mas sim no minuto X+1 e assim no minuto X são enviados 27 pacotes $(4\times 6+3)$. Isto acontece com uma probabilidade de 23,389%.

Além disto, verifica-se que ao longo da captura há sempre pacotes do protocolo SCCP, pois o valor mínimo é diferente de zero pacotes por minuto (ou zero bytes por minuto).

Assim como nos casos anteriores a moda (30*ppm* e 1952*Bpm*) repete-se com uma probabilidade elevada (aproximadamente 69%) fazendo com que a mediana coincida com os mesmos valores da moda (tabela 9).

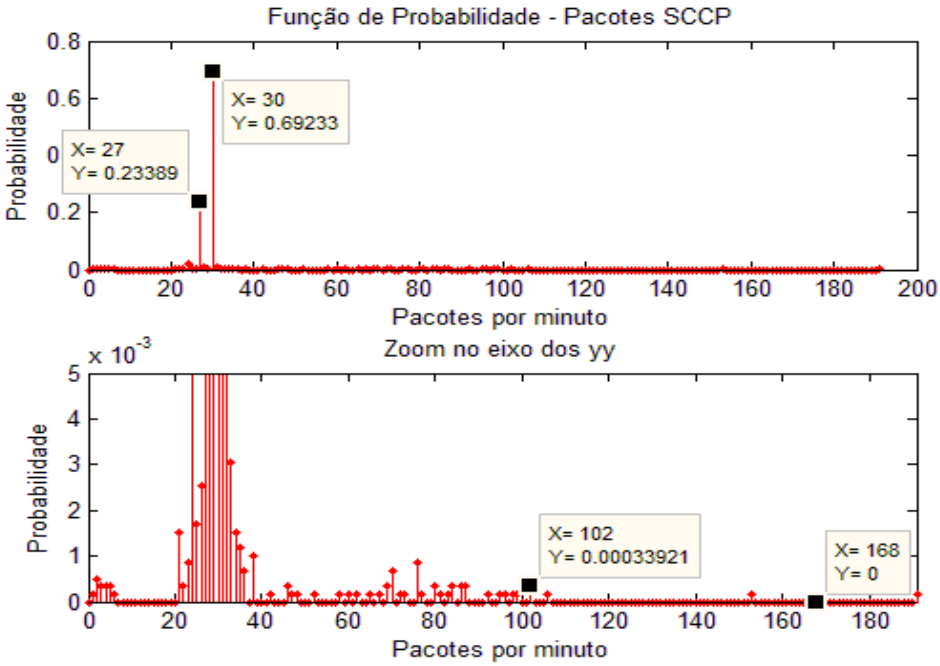

Figura 37 – Função de probabilidade dos pacotes por minuto do protocolo SCCP

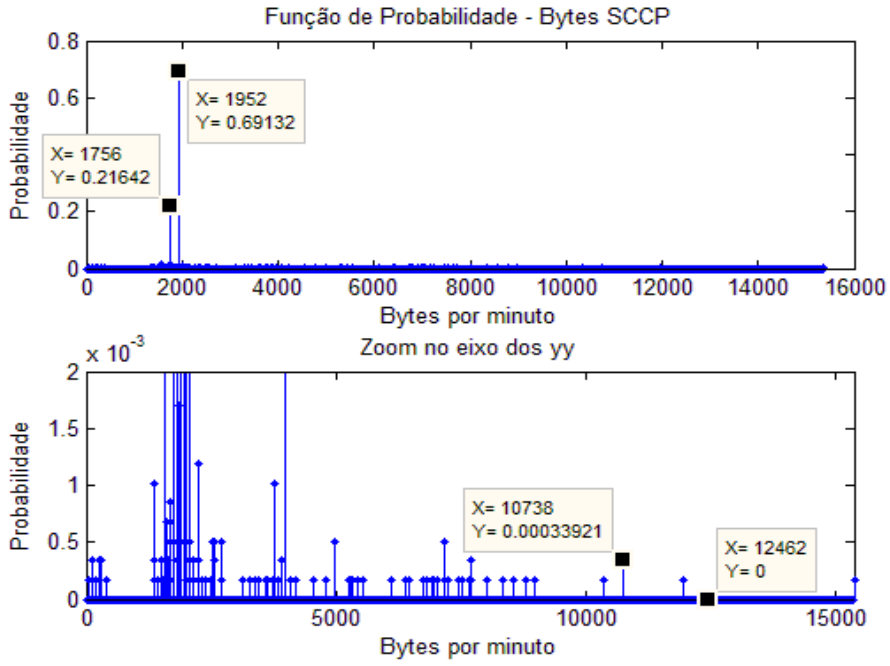

Figura 38 – Função de probabilidade dos bytes por minuto do protocolo SCCP

	Tráfego VoIP – Sinalização: Protocolo SCCP						
	Valor Máximo	Média	Valor Mínimo	Mediana	Moda	Variância	Desvio Padrão
ppm	191	29,50	1	30	30	28,27	5,32
Bpm	15370	1943,67	60	1952	1952	280164,63	529,31

Tabela 9 – Estatísticas do tráfego do protocolo SCCP

5.3.2.1.2 Parte II - Análise das chamadas

Através da captura, verificou-se que ao longo da semana foram efectuadas 23 chamadas. Recorrendo aos pacotes *OpenReceiveChannel*, *OpenReceiveChannelAck*, *StartMediaTransmission*, *CloseReceiveChannel* e *StopMediaTransmission*, obtiveram-se alguns dados importantes sobre as chamadas.

É importante referir que as chamadas possuem um identificador, de forma a poderem ser distinguidas umas das outras. Neste caso, os *Conference ID* que as chamadas utilizaram foram os seguintes:

Nº	*Conference ID*	Nº	*Conference ID*	Nº	*Conference ID*	Nº	*Conference ID*
1	27928862	7	27928887	13	27928903	19	27928920
2	27928867	8	27928889	14	27928905	20	27928928
3	27928869	9	27928891	15	27928907	21	27928930
4	27928871	10	27928893	16	27928912	22	27928932
5	27928873	11	27928897	17	27928914	23	27928938
6	27928885	12	27928901	18	27928918		

Tabela 10 – *Conference ID* utilizados

Nas tabelas seguintes (tabelas 11 e 12) são apresentados os endereços IP e portos utilizados para a troca de sinalização e de dados em cada chamada:

Nº	Endereço SCCP 1 [*IP:porto*]	Endereço SCCP 2 [*IP:porto*]	Endereço MCP 1 [*IP:porto*]	Endereço MGCP 2 [*IP:porto*]
1	10.0.0.254:2000	10.0.0.86:49502	10.0.0.254:2427	10.0.0.1:2427
2	10.0.0.254:2000	10.0.0.86:49502	10.0.0.254:2427	10.0.0.1:2427
3	10.0.0.254:2000	10.0.0.86:49502	10.0.0.254:2427	10.0.0.1:2427
4	10.0.0.254:2000	10.0.0.86:49502	10.0.0.254:2427	10.0.0.1:2427
5	10.0.0.254:2000	10.0.0.86:49502	10.0.0.254:2427	10.0.0.1:2427
6	10.0.0.254:2000	193.136.93.104:52079	10.0.0.254:2427	10.0.0.1:2427
7	10.0.0.254:2000	10.0.0.89:52020	10.0.0.254:2427	10.0.0.1:2427
8	10.0.0.254:2000	10.0.0.89:52020	10.0.0.254:2427	10.0.0.1:2427
9	10.0.0.254:2000	10.0.0.96:49445	10.0.0.254:2427	10.0.0.1:2427
10	10.0.0.254:2000	10.0.0.96:49445	10.0.0.254:2427	10.0.0.1:2427
11	10.0.0.254:2000	10.0.0.89:52020	10.0.0.254:2427	10.0.0.1:2427
12	10.0.0.254:2000	10.0.0.86:52078	10.0.0.254:2427	10.0.0.1:2427

13	10.0.0.254:2000	10.0.0.86:52078	10.0.0.254:2427	10.0.0.1:2427
14	10.0.0.254:2000	10.0.0.89:52020	10.0.0.254:2427	10.0.0.1:2427
15	10.0.0.254:2000	193.136.93.104:49257	Não utiliza	Não utiliza
16	10.0.0.254:2000	193.136.93.104:49257	10.0.0.254:2427	10.0.0.1:2427
17	10.0.0.254:2000	193.136.93.104:49257	10.0.0.254:2427	10.0.0.1:2427
18	10.0.0.254:2000	10.0.0.89:52020	10.0.0.254:2427	10.0.0.1:2427
19	10.0.0.254:2000	10.0.0.89:52020	10.0.0.254:2427	10.0.0.1:2427
20	10.0.0.254:2000	10.0.0.86:52078	10.0.0.254:2427	10.0.0.1:2427
21	10.0.0.254:2000	10.0.0.86:52078	10.0.0.254:2427	10.0.0.1:2427
22	10.0.0.254:2000	10.0.0.86:52078	10.0.0.254:2427	10.0.0.1:2427
23	10.0.0.254:2000	10.0.0.86:51816	10.0.0.254:2427	10.0.0.1:2427

Tabela 11 – Endereços IP e portos utilizados na troca de sinalização em cada chamada

Nº	Endereço RTP 1 [*IP:porto*]	Endereço RTP 2 [*IP:porto*]	Nº	Endereço RTP 1 [*IP:porto*]	Endereço RTP 2 [*IP:porto*]
1	10.0.0.1:16920	10.0.0.86:26276	13	10.0.0.1:17486	10.0.0.86:31742
2	10.0.0.1:17086	10.0.0.86:31846	14	10.0.0.1:19088	10.0.0.89:17126
3	10.0.0.1:16732	10.0.0.86:18324	15	10.0.0.254:24698	193.136.93.104:29408
4	10.0.0.1:16718	10.0.0.86:19724	16	10.0.0.1:16604	193.136.93.104:17090
5	10.0.0.1:18318	10.0.0.86:20186	17	10.0.0.1:16672	193.136.93.104:22556
6	10.0.0.1:19530	193.136.93.104:21562	18	10.0.0.1:18618	10.0.0.89:25024
7	10.0.0.1:17614	10.0.0.89:16778	19	10.0.0.1:19456	10.0.0.89:26654
8	10.0.0.1:18404	10.0.0.89:25772	20	10.0.0.1:18294	10.0.0.86:25966
9	10.0.0.1:19306	10.0.0.96:16970	21	10.0.0.1:18752	10.0.0.86:16678
10	10.0.0.1:17042	10.0.0.96:20616	22	10.0.0.1:17282	10.0.0.86:32110
11	10.0.0.1:16592	10.0.0.89:30660	23	10.0.0.1:18292	10.0.0.86:28002
12	10.0.0.1:18846	10.0.0.86:27888			

Tabela 12 – Endereços IP e portos utilizados na troca de dados em cada chamada

Através da tabela 12 verifica-se que todas as chamadas, exceptuando a 15 (a verde), foram efectuadas entre um telefone da rede VoIP e um da central telefónica, pois os dados são trocados entre um telefone da rede VoIP e o *router* (10.0.0.1, endereço RTP 1). Por esta razão, é possível verificar através da tabela 11 que estas mesmas chamadas utilizam o protocolo de sinalização MGCP além do SCCP.

A chamada 15 tem um comportamento diferente, pois trata-se de uma chamada que foi posta em espera. É por isto que o *endereço RTP 1* não é nem de um telefone da rede VoIP nem do *router*, mas sim do *CallManager*. Além disto, nesta chamada verifica-se que a sinalização utiliza unicamente o protocolo SCCP, pois neste caso só intervêm o *CallManager* e um telefone da rede VoIP.

O tráfego desta chamada, como era de esperar, tem um único sentido, do *CallManager* para o telefone VoIP (figura 39), pois trata-se de um sinal sonoro (por exemplo, música) para indicar a quem efectuou a chamada que a mesma se encontra em espera.

```
No.,Time,Source,Src Port,Destination,Dst Port,Protocol,Pkt Length,Info
536324,250312.295055,10.0.0.254,24698,193.136.93.104,29408,RTP,214,PT=ITU-T G.711 PCMU  SSRC=0x759  Seq=1885  Time=800
536326,250312.314016,10.0.0.254,24698,193.136.93.104,29408,RTP,214,PT=ITU-T G.711 PCMU  SSRC=0x759  Seq=1886  Time=960
536328,250312.335060,10.0.0.254,24698,193.136.93.104,29408,RTP,214,PT=ITU-T G.711 PCMU  SSRC=0x759  Seq=1887  Time=1120
536330,250312.354019,10.0.0.254,24698,193.136.93.104,29408,RTP,214,PT=ITU-T G.711 PCMU  SSRC=0x759  Seq=1888  Time=1280
536332,250312.375062,10.0.0.254,24698,193.136.93.104,29408,RTP,214,PT=ITU-T G.711 PCMU  SSRC=0x759  Seq=1889  Time=1440
536334,250312.394017,10.0.0.254,24698,193.136.93.104,29408,RTP,214,PT=ITU-T G.711 PCMU  SSRC=0x759  Seq=1890  Time=1600
536336,250312.415063,10.0.0.254,24698,193.136.93.104,29408,RTP,214,PT=ITU-T G.711 PCMU  SSRC=0x759  Seq=1891  Time=1760
536338,250312.434022,10.0.0.254,24698,193.136.93.104,29408,RTP,214,PT=ITU-T G.711 PCMU  SSRC=0x759  Seq=1892  Time=1920
536340,250312.455072,10.0.0.254,24698,193.136.93.104,29408,RTP,214,PT=ITU-T G.711 PCMU  SSRC=0x759  Seq=1893  Time=2080
536342,250312.474022,10.0.0.254,24698,193.136.93.104,29408,RTP,214,PT=ITU-T G.711 PCMU  SSRC=0x759  Seq=1894  Time=2240
536344,250312.495066,10.0.0.254,24698,193.136.93.104,29408,RTP,214,PT=ITU-T G.711 PCMU  SSRC=0x759  Seq=1895  Time=2400
```

Figura 39 – Pacotes de dados da chamada 15

Verifica-se também nas tabelas anteriores (tabela 11 e 12), que os portos utilizados encontram-se dentro do esperado segundo as configurações especificadas na secção 5.2:

- O *CallManager* (10.0.0.254) troca os pacotes SCCP através do porto TCP 2000 e os MGCP através do UDP 2427.
- Os portos utilizados para a troca dos pacotes de dados (RTP) estão dentro da gama definida na secção 5.2, isto é, entre 16384 e 32767.

Através dos pacotes referidos anteriormente, calculou-se a duração de cada uma das chamadas. Como é possível observar no gráfico da figura 40, 12 das 23 chamadas (52,17% das chamadas) tiveram duração inferior a um minuto.

Nº	Duração [*min*]	Nº	Duração [*min*]	Nº	Duração [*min*]	Nº	Duração [*min*]
1	20,6690	7	0,2741	13	2,8303	19	0,0759
2	0,1670	8	1,8810	14	0,1391	20	1,4795
3	1,5573	9	0,0464	15	0,1566	21	5,2121
4	0,0293	10	0,3184	16	0,8648	22	3,2556
5	7,0079	11	0,0359	17	0,5872	23	4,4538
6	1,2062	12	3,3264	18	0,1006		

Tabela 13 – Duração das chamadas

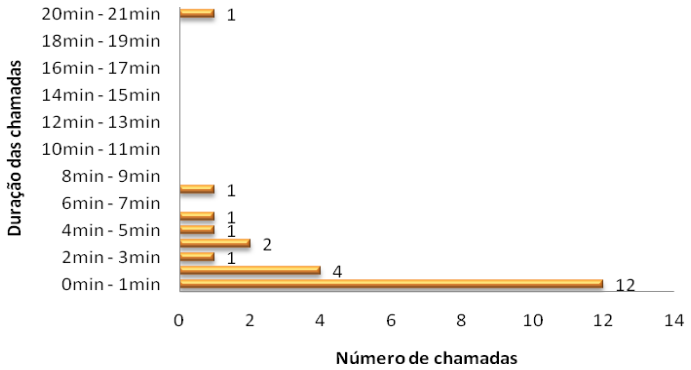

Figura 40 – Gráfico do número de chamadas segundo a duração

5.3.2.2 Teste 2 – Caracterização do tráfego ao nível da sessão

Para analisar as capturas deste teste além dos *scripts* descritos no inicio da secção 5.3.2, utilizaram-se também outros dois *scripts*, para situações específicas deste tipo de capturas:

- No caso das chamadas feitas entre um telefone VoIP e um da central telefónica, os dados contêm pacotes RTP e RTCP. Por forma a poder estudar o comportamento dos pacotes de dados, separaram-se estes dois tipos de pacotes, pois os RTCP são pacotes de controlo, e os RTP são os que transportam os dados. Para poder separar estes pacotes recorreu-se ao *script VLAN_CT_DadosEdit.sh* (Apêndice A.11).
- Para poder calcular o *jitter* e o número de pacotes perdidos, separaram-se os fluxos recorrendo ao *script chamadaA_B.sh* (Apêndice A.12).

Como já foi referido na secção 5.3.1.2 no teste 2 foram efectuados diversos tipos de chamadas durante vários períodos do dia. As chamadas efectuadas foram:

- Entre dois telefones VoIP na mesma VLAN, neste caso os dois na VLAN 1.
- Entre dois telefones VoIP em VLANs diferentes, neste caso um na VLAN 1 e outro na VLAN 2.
- Entre um telefone VoIP (VLAN 1) e um na central telefónica (CT).

Através destas chamadas, verificou-se que não está activo o modo de supressão de silêncio, pois a terceira repetição de cada chamada foi feita com os auscultadores desconectados dos telefones de forma a não ser captado nenhum som, e na mesma estas chamadas apresentaram pacotes de dados.

Seguidamente é apresentada uma análise de cada um dos tipos das chamadas efectuadas.

5.3.2.2.1 Chamadas entre dois telefones VoIP na mesma VLAN

Durante o estabelecimento e finalização de uma chamada efectuada entre dois telefones da VLAN 1, a sinalização é trocada entre cada um dos telefones em questão e o *CallManager*, através do porto TCP 2000 do *CallManager*. Neste tipo de chamada, como ambos os telefones são VoIP, o único protocolo de sinalização utilizado é o SCCP.

Os pacotes de dados são trocados directamente entre os telefones que estão a efectuar a chamada, utilizando portos dentro da gama referida na secção 5.2 (UDP 16384-32767). Um exemplo destas trocas de mensagens encontra-se no Apêndice C.1.1.

Seguidamente, mostram-se os gráficos do tráfego de dados e de sinalização de cada uma das chamadas efectuadas. Como se pode observar, todas as chamadas têm um comportamento similar, tanto no caso dos dados como no caso da sinalização (figuras 41, 42 e 43).

No caso dos **dados**, são enviados no máximo entre 100 e 101*pps* por ambos os fluxos (tabela 28, Apêndice C.2.1). Isto corresponde a aproximadamente 50*pps* por fluxo, o que leva a uma largura de banda consumida de 64*kbps*[5], como é definido pelo *codec* G.711 (o fluxo A corresponde aos pacotes que têm como origem o telefone VoIP cujo endereço IP é o 10.0.0.96 e o fluxo B é o que vai no sentido oposto). Isto verifica-se na figura 44.

Quanto à **sinalização**, verifica-se através dos gráficos, que os picos máximos acontecem no estabelecimento e finalização das chamadas, pois nestes casos os telefones trocam diversos tipos de pacotes SCCP, enquanto que durante uma chamada os únicos pacotes trocados por eles são os *KeepAlive*.

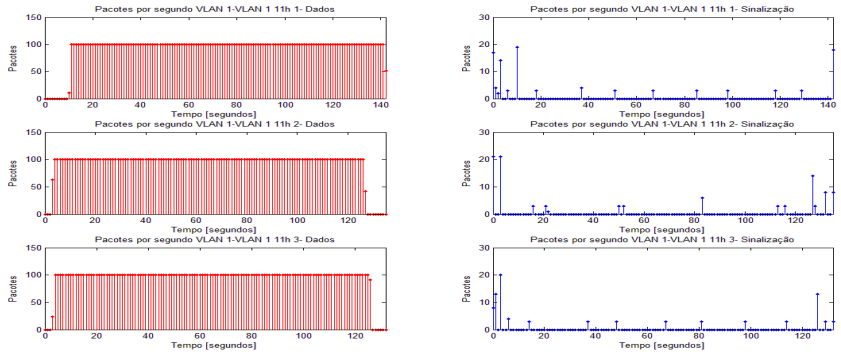

Figura 41 – Gráficos do tráfego das chamadas feitas às 11 horas entre dois telefones da VLAN 1

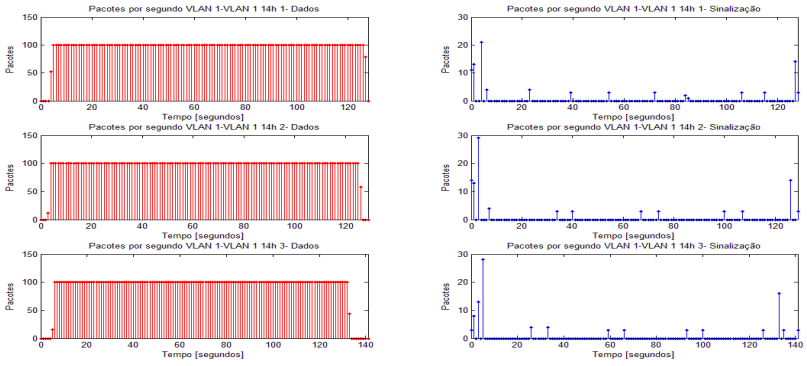

Figura 42 – Gráficos do tráfego das chamadas feitas às 14 horas entre dois telefones da VLAN 1

[5] Os pacotes de dados são de 214*bytes*, dos quais 54*bytes* são de cabeçalho e os restantes 160*bytes* são de dados.

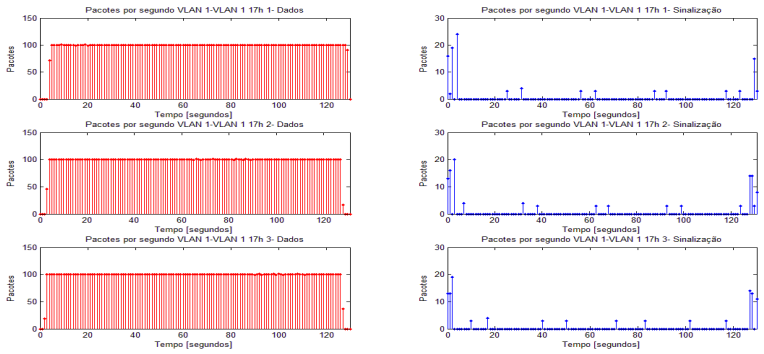

Figura 43 – Gráficos do tráfego das chamadas feitas às 17 horas entre dois telefones da VLAN 1

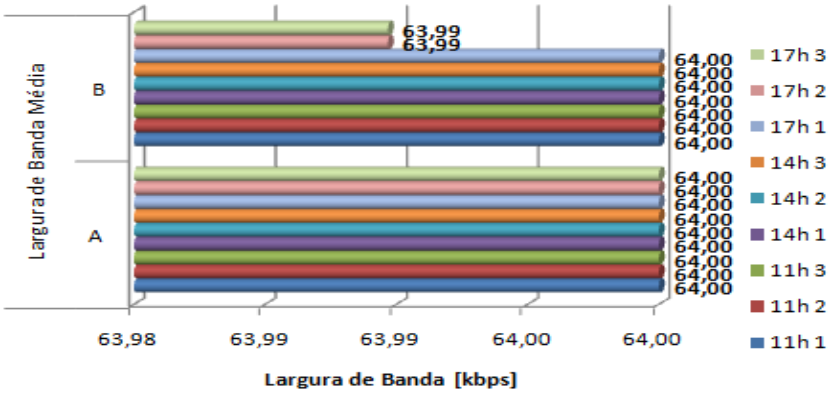

Figura 44 – Largura de banda média das chamadas efectuadas entre telefones VoIP da mesma VLAN

O *jitter* médio variou entre 0,023428 *ms* e 0,050721 *ms* (tabela 29, Apêndice C.2.1), exceptuando na primeira chamada efectuada às 11 horas, pois neste caso os valores atingidos pelo *jitter* são superiores, para o fluxo A o *jitter* é de 0,1657 *ms* e para o fluxo B é 0,1432 *ms*. Este aumento da variação do atraso pode ser consequência da existência de uma maior quantidade de pacotes (de outras aplicações) nas filas de espera dos *routers*, pois ao existirem muitos pacotes nestas filas pode fazer com que alguns pacotes de voz esperem mais do que outros.

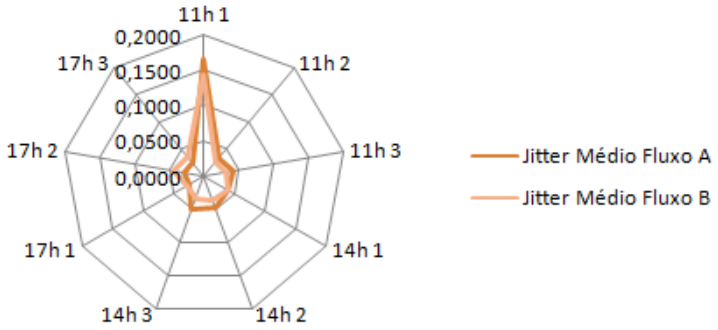

Figura 45 – ***Jitter*** médio (chamadas entre dois telefones VoIP da VLAN 1)

5.3.2.2.2 Chamadas entre dois telefones VoIP em VLANs diferentes

Neste caso, as chamadas têm um comportamento parecido ao caso anterior. Isto verifica-se através do exemplo no Apêndice C.1.2, no qual é possível observar que o estabelecimento e finalização da chamada são efectuados recorrendo unicamente ao protocolo de sinalização SCCP, utilizando o porto TCP 2000. A troca de pacotes de sinalização é efectuada entre o telefone em questão e o *CallManager*. No caso dos dados, também são trocados directamente entre os dois telefones, através do protocolo RTP utilizando os portos UDP 18220 e 26600, os quais se encontram dentro da gama indicada na secção 5.2.

Os gráficos seguintes mostram o comportamento do tráfego de dados e sinalização deste tipo de chamadas (dois telefones VoIP, um na VLAN 1 e outro na VLAN 2). Neste caso também se verifica que no estabelecimento e finalização das chamadas a quantidade de pacotes de **sinalização** enviados é muito superior à quantidade enviada ao longo das chamadas, pois durante as chamadas os únicos pacotes de sinalização existentes são os pacotes *KeepAlive* de ambos os telefones, e no estabelecimento e finalização das chamadas existem diversos tipos de pacotes SCCP. No caso dos **dados**, o comportamento é aproximadamente constate em torno dos 100*pps*.

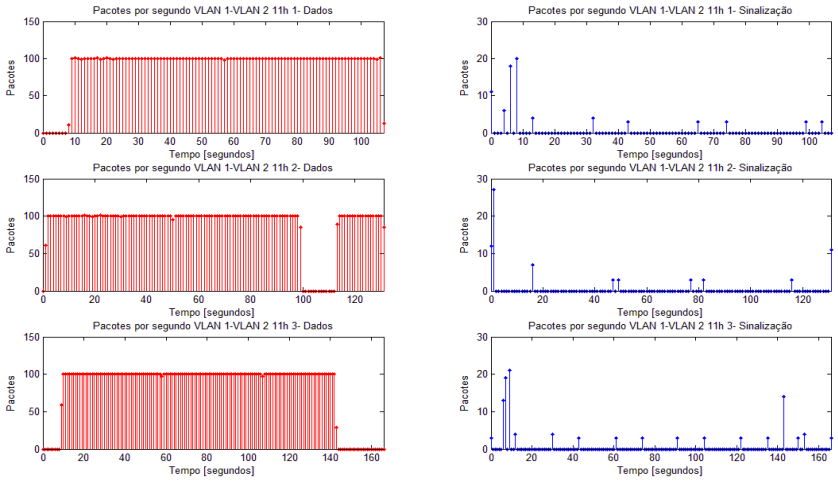

Figura 46 – Gráficos do tráfego das chamadas feitas às 11 horas entre um telefone VoIP da VLAN 1 e outro telefone VoIP da VLAN 2[6]

[6] Na segunda chamada efectuada às 11 horas verificou-se uma interrupção na captura de 12 segundos, entre o segundo 100 e o 112.

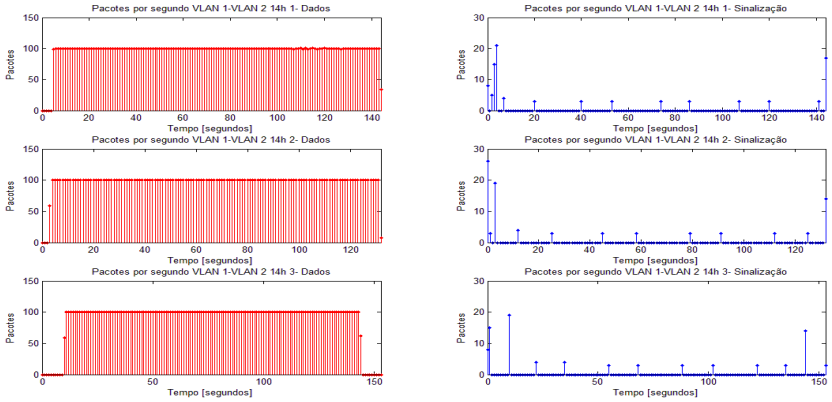

Figura 47 – Gráficos do tráfego das chamadas feitas às 14 horas entre um telefone VoIP da VLAN 1 e outro telefone VoIP da VLAN 2

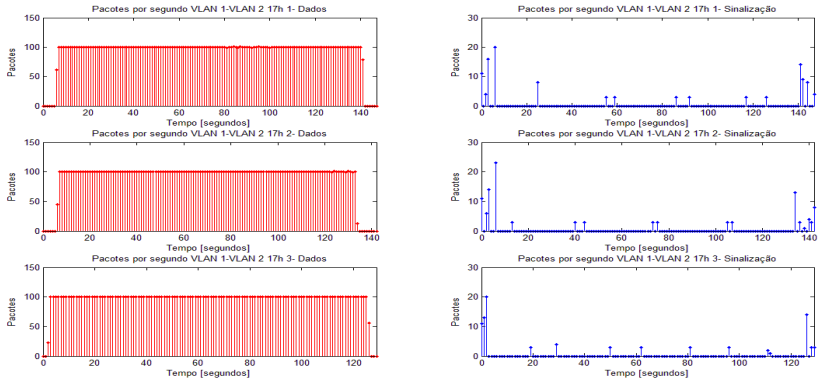

Figura 48 – Gráficos do tráfego das chamadas feitas às 17 horas entre um telefone VoIP da VLAN 1 e outro telefone VoIP da VLAN 2

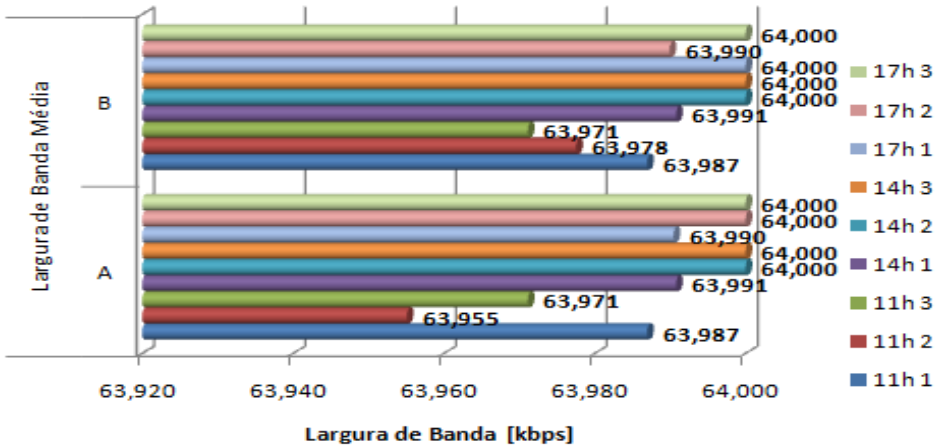

Figura 49 – Largura de banda média das chamadas efectuadas entre telefones VoIP de VLANs diferentes

As chamadas efectuadas às 11 horas, como se verifica na tabela 31 do Apêndice C.2.2, apresentaram perdas de pacotes. Na primeira chamada foram perdidos 0,04% dos pacotes

de cada um dos fluxos, na segunda chamada perderam-se 11,41% do fluxo A e 11,39% do fluxo B, e na terceira chamada perderam-se 0,08% e 0,07% dos pacotes do fluxo A e B, respectivamente (o fluxo A corresponde aos pacotes que vão desde o telefonte da VLAN 1 para o telefone da VLAN 2, e o fluxo B é o que vai no sentido oposto).

Quanto à variação do atraso é possível verificar, através da figura 50, que na maior parte dos casos o *jitter* do fluxo A é inferior ao *jitter* do fluxo B, excepto na segunda e terceira chamada efectuada às 14 horas em que ambos os fluxos apresentaram iguais valores de *jitter*. Em geral, os valores de *jitter* não ultrapassaram os 0,0485*ms* e foram sempre superiores a 0,0242*ms* (consultar tabelas do Apêndice C.2.2). A diferença entre o *jitter* apresentado pelo fluxo A e o fluxo B pode ser consequência das configurações de QoS (*Quality of Service*), pois nesta rede o tráfego procedente da VLAN 1 tem prioridade sobre o restante tráfego.

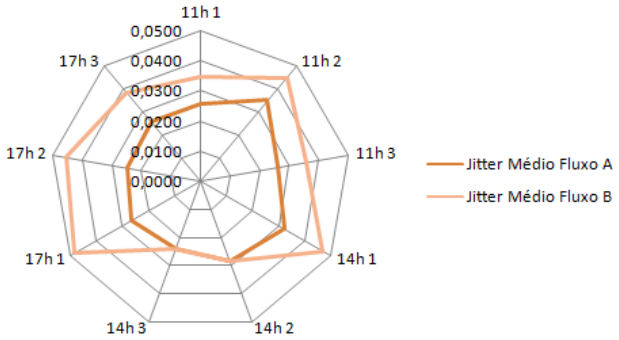

Figura 50 –*Jitter* médio (chamadas entre dois telefones VoIP de VLANs diferentes)

5.3.2.2.3 Chamadas entre um telefone VoIP (VLAN 1) e um na central telefónica (CT)

Neste caso durante o estabelecimento e finalização das chamadas há dois tipos de protocolos de sinalização, SCCP e MGCP. As mensagens do protocolo SCCP são trocadas entre o *CallManager* e o telefone da VLAN 1 através do porto TCP 2000 do *CallManager*, e as mensagens do protocolo MGCP são trocadas entre o *CallManager* e o *router* através do porto UDP 2427.

A transmissão de dados é feita entre o telefone e o *router*, através do protocolo RTP utilizando os portos da gama 16384 e 32767 (secção 5.2). Neste caso existe também um outro tipo de pacotes de dados, trata-se de pacotes RTCP, utilizados para controlo.

Os pacotes RTCP utilizam os portos dentro da mesma gama referida anteriormente. Estes utilizam o porto X+1 considerando que os RTP utilizam o porto X, e vão unicamente desde o *router* para o telefone VoIP. Isto pode verificar-se no gráfico seguinte, pois a

percentagem de pacotes RTCP no fluxo A (desde o telefone VoIP para o *router*) é nula. Além disto, como se observa na figura seguinte, este tipo de pacotes não ultrapassaram os 0,43% do total dos pacotes de dados e os restantes pacotes representam os RTP (figuras do Apêndice C.1.3 e tabela 34 do Apêndice C.2.3).

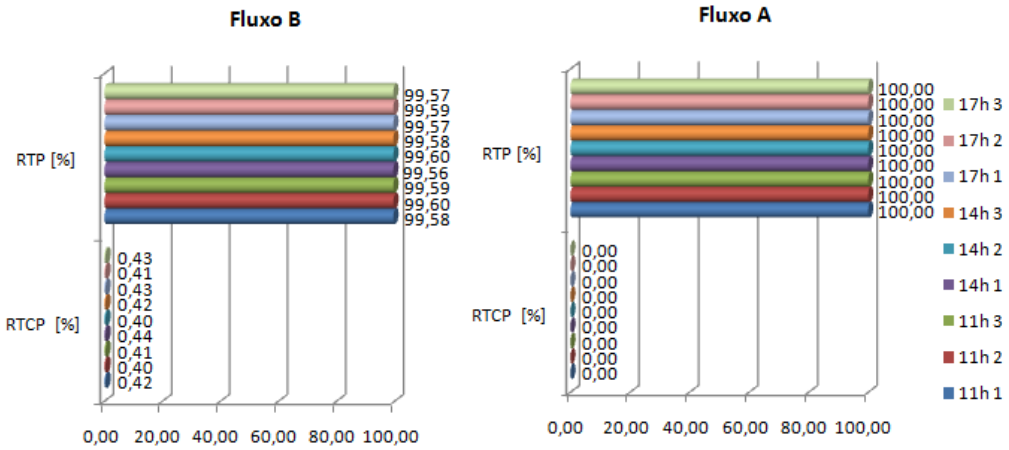

Figura 51 – Percentagens de pacotes RTCP

No Apêndice C.1.3 encontram-se exemplos dos pacotes trocados no estabelecimento, finalização e durante uma chamada deste tipo.

Uma vez mais e como se pode observar nos gráficos seguintes, os pacotes de **sinalização** apresentam picos de maior intensidade durante o estabelecimento e finalização das chamadas.

No caso dos **dados**, observa-se um comportamento constante em torno dos 100*pps* (tabela 32, Apêndice C.2.3).

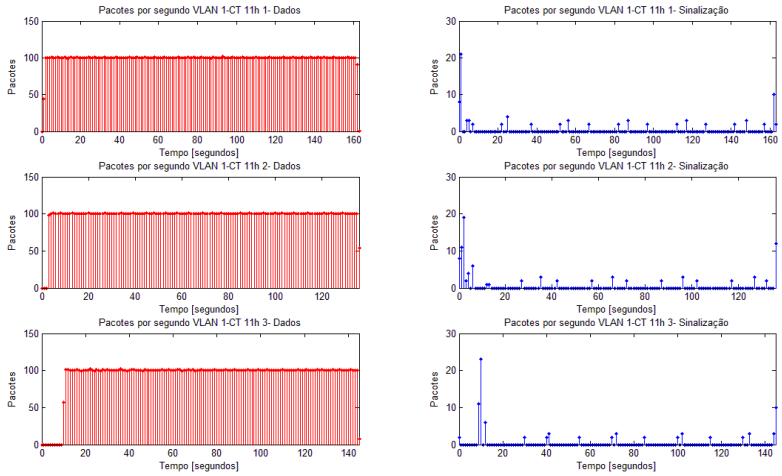

Figura 52 – Gráficos do tráfego das chamadas feitas às 11 horas entre um telefone VoIP da VLAN 1 e um telefone da central telefónica

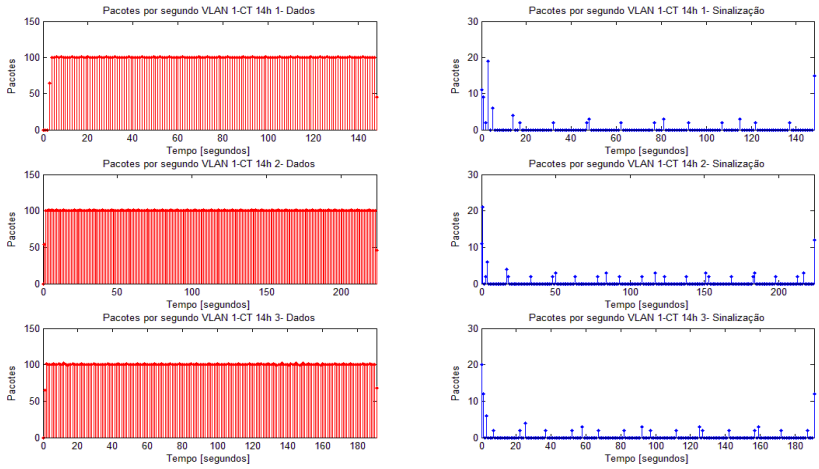

Figura 53 – Gráficos do tráfego das chamadas feitas às 14 horas entre um telefone VoIP da VLAN 1 e um telefone da central telefónica

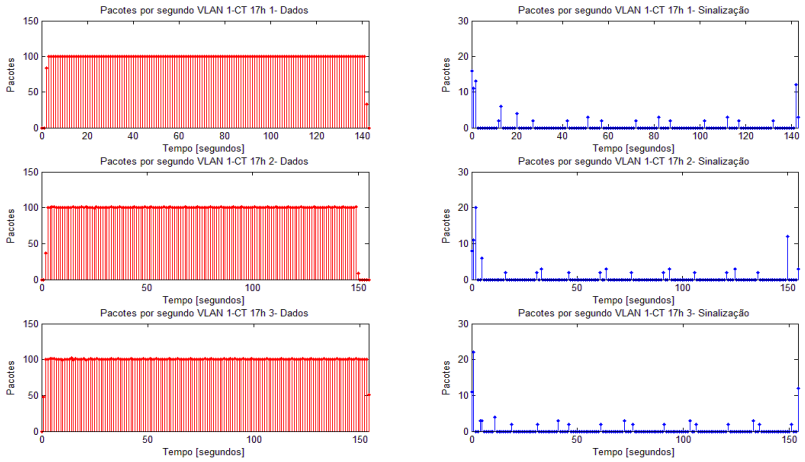

Figura 54 – Gráficos do tráfego das chamadas feitas às 17 horas entre um telefone VoIP da VLAN 1 e um telefone da central telefónica

No gráfico correspondente à largura de banda (figuras 55), verifica-se que cada fluxo consumiu 64*kbps* ou 63,99*kbps*, o que corresponde aos 50*pps* enviados (o fluxo A representa os pacotes que têm como origem o telefone VoIP da VLAN 1 e destino o telefone da central telefónica, e o fluxo B representa os pacotes que vão no sentido oposto).

O fluxo B apresenta valores de *jitter* superiores aos apresentados pelo fluxo A, pois o *jitter* do fluxo A varia entre 0,0288 e 0,3379*ms*, e no caso do fluxo B varia entre 1,5725 e 1,6654*ms* (figura 56). Esta diferença nos valores de *jitter* de ambos os fluxos, pode ser consequência da conversão dos formatos, pois esta pode ser mais demorada num sentido do que no outro.

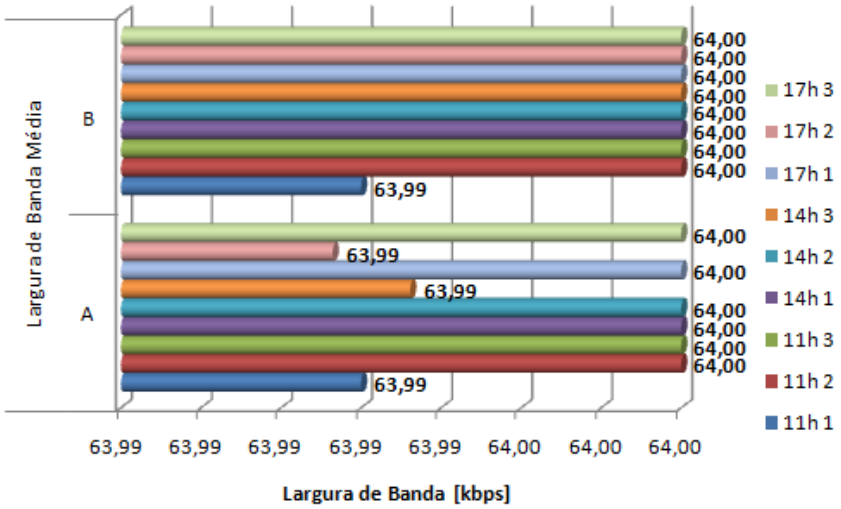

Figura 55 – Valores máximos e mínimos atingidos pela largura de banda

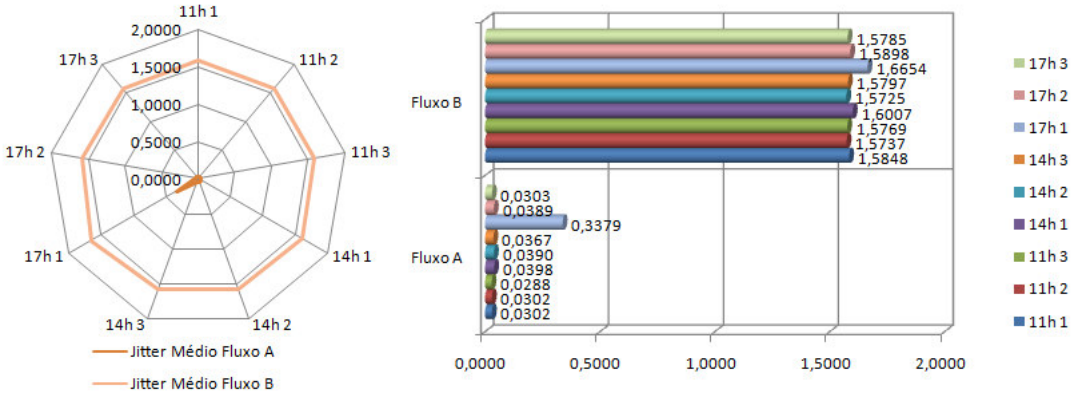

Figura 56 – *Jitter* médio (chamadas entre em telefone VoIP e um telefone da central telefónica)

5.3.2.2.4 Comparação entre os diferentes tipos de chamadas

De **forma geral**, tanto no caso dos dados como na sinalização, todas as chamadas, em condições normais, têm um comportamento muito similar. As diferenças entre os tipos de chamadas basicamente encontram-se na quantidade de pacotes de sinalização e dados trocados ao longo da chamada, e nos protocolos utilizados.

Através da análise das capturas realizadas, verificou-se que numa chamada entre dois telefones VoIP da rede instalada no IT, isto é, ambos na VLAN 1 ou um na VLAN 1 e o outro na VLAN 2, a percentagem de pacotes de sinalização é superior ao caso em que a chamada é efectuada entre um telefone VoIP (neste caso na VLAN1) e um na central telefónica (CT). Isto verifica-se no gráfico da figura 57.

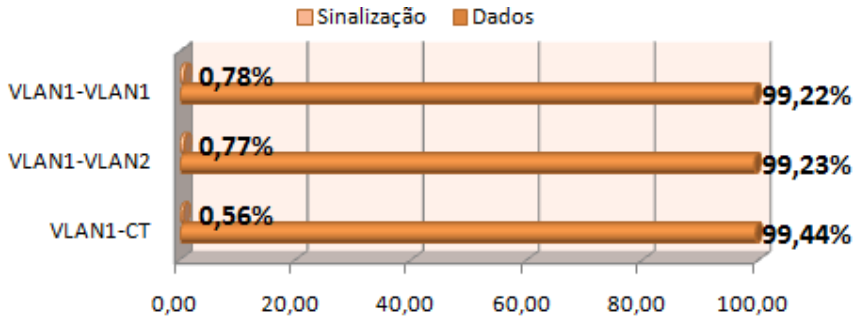

Figura 57 – Gráfico das percentagens dos pacotes de dados e de sinalização trocados nos diferentes tipos de chamadas

Quanto aos protocolos utilizados, verificou-se que no caso da chamada ser feita entre dois telefones VoIP (ambos telefones na VLAN 1 ou um na VLAN 1 e outro na VLAN 2) os únicos protocolos utilizados são o RTP para os dados e o SCCP para a sinalização. No caso das chamadas entre um telefone VoIP e um telefone da rede POTS, na sinalização são utilizados dois protocolos, SCCP e MGCP, e no caso dos dados utiliza-se o RTP, mas neste caso o RTP vem acompanhado do RTCP.

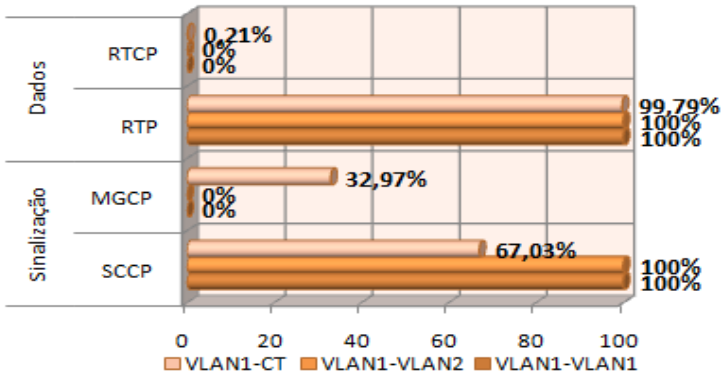

Figura 58 – Gráfico das percentagens dos diferentes tipos de protocolos

Como se verifica no gráfico seguinte, o protocolo MGCP envia menos mensagens do que o protocolo SCCP. É por isso que as chamadas em que ambos os telefones recorrem ao protocolo SCCP (VLAN 1 – VLAN 1 e VLAN 1 – VLAN 2) a percentagem de sinalização é superior ao caso em que a chamada é feita entre um telefone VoIP e um da central telefónica (isto, é através do *router* – VLAN 1 – CT), pois neste caso o telefone VoIP utiliza SCCP, mas o *router* utiliza o protocolo MGCP.

Como foi possível observar nos gráficos apresentados nas secções anteriores (5.3.2.2.1, 5.3.2.2.2 e 5.3.2.2.3) todas as chamadas tiveram um comportamento similar:

- No caso dos **dados**, ao longo da chamada foram enviados aproximadamente 100*pps* por ambos os fluxos.

- No caso da **sinalização**, os picos acontecem no estabelecimento e finalização da chamada, ao longo da chamada são enviados alguns pacotes de sinalização de carácter periódico.

No caso dos pacotes de **sinalização,** é de notar que em todas as chamadas há uma elevada probabilidade de não existirem pacotes deste tipo (0*pps*). Como já foi referido anteriormente ao longo de uma chamada, os únicos pacotes de sinalização existentes são os de índole periódica, o que origina muitos períodos de tempo em que não existem pacotes de sinalização, pois os pacotes *keepAlive* (protocolo SCCP) aparecem periodicamente a cada trinta segundos e os do protocolo MGCP (*NTFY*) aparecem com uma frequência de aproximadamente quinze segundos.

Nos gráficos da função de probabilidade dos pacotes de sinalização também se verifica que, contrariamente aos dados, os valores máximos atingidos não correspondem às probabilidades mais elevadas, pois os únicos instantes de tempo onde são transmitidos os valores máximos de pacotes de sinalização são durante o estabelecimento e finalização das chamadas. Nestes casos podem, no máximo, ser enviados entre 16 e 29 pacotes por segundo.

Também é possível observar nos seguintes gráficos, que as chamadas feitas entre dois telefones VoIP, isto é, os dois na VLAN 1 ou um na VLAN 1 e outro na VLAN 2, o valor que se repete mais vezes seguidamente a zero pacotes por segundos é três pacotes por segundo. Isto acontece devido à existência de pacotes de sinalização SCCP periódicos, como é o caso dos *KeepAlive* (cada pacote *KeepAlive* é respondido com um *KeepAliveAck* e este último tem como resposta um *Ack*, o qual origina um conjunto de três pacotes de sinalização).

No caso das chamadas feitas entre um telefone da VLAN 1 e um da central telefónica, o segundo valor com mais probabilidade de acontecer é dois pacotes por segundo. Isto acontece pois além dos pacotes *KeepAlive* do protocolo SCCP, existem também os NTFY que são os pacotes de sinalização periódicos do protocolo MGCP. Os pacotes NTFY estão sempre acompanhados de uma mensagem de resposta, dando origem a dois pacotes (figura 4). Visto que estes acontecem aproximadamente quatro vezes ao longo de um minuto e os *KeepAlives* são enviados duas vezes durante um minuto, a probabilidade de serem enviados 2*pps* é maior do que a probabilidade de serem enviados 3*pps*.

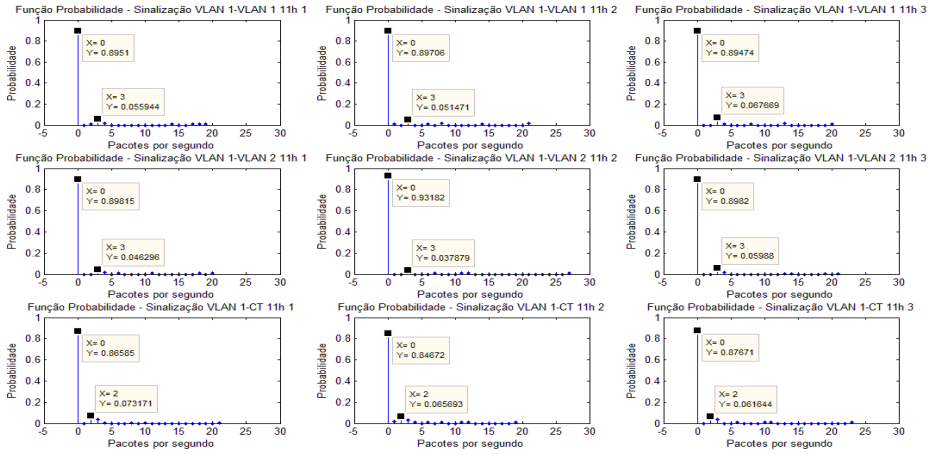

Figura 59 – Função probabilidade dos pacotes de sinalização das chamadas efectuadas às 11 horas

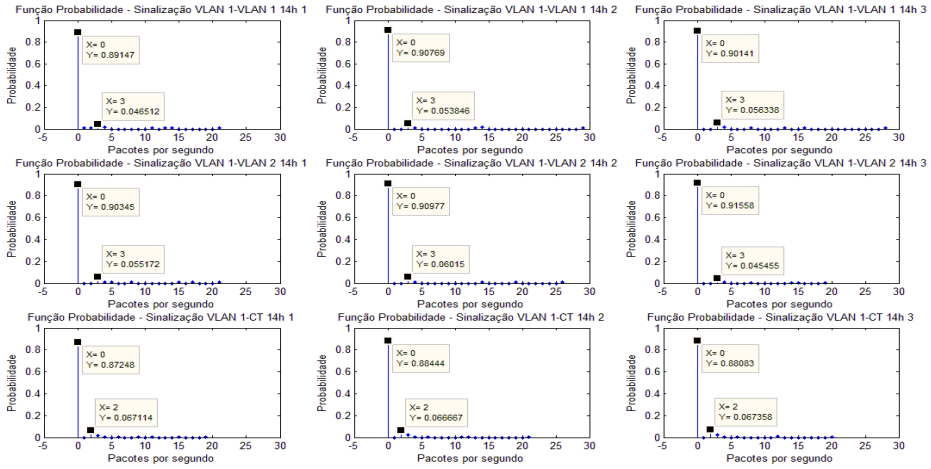

Figura 60 – Função probabilidade dos pacotes de sinalização das chamadas efectuadas às 14 horas

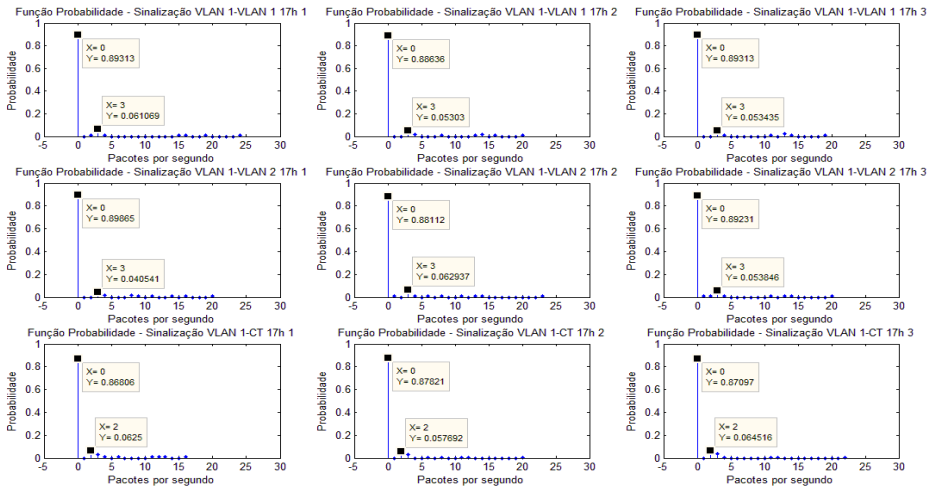

Figura 61 – Função probabilidade dos pacotes de sinalização das chamadas efectuadas às 17 horas

No caso dos **dados**, verificou-se que em todas as chamadas existe a probabilidade de que a quantidade de pacotes enviados ao longo de um segundo seja nula. Isto acontece devido à existência de pacotes de sinalização periódicos, pois durante o segundo em que são enviados estes pacotes de sinalização, não estão a ser enviados pacotes de dados.

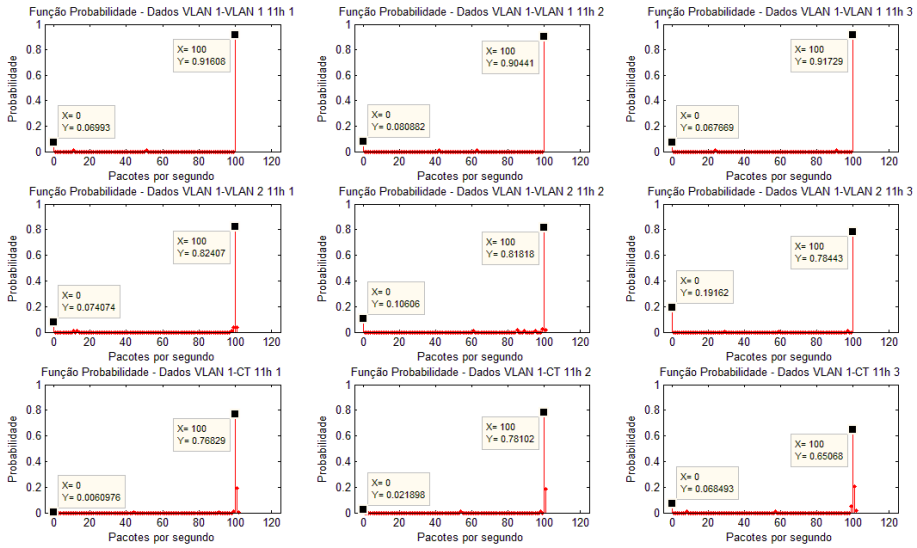

Figura 62 – Função probabilidade dos pacotes de dados das chamadas efectuadas às 11 horas

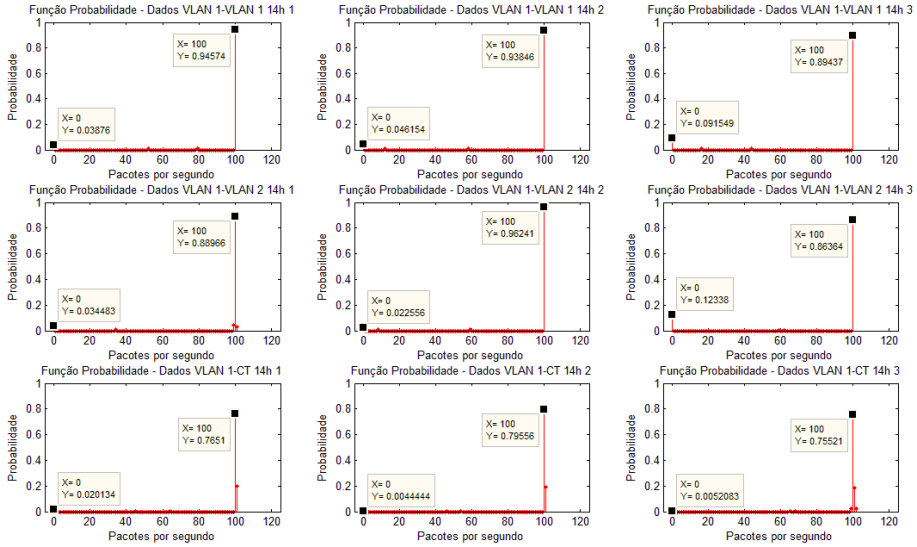

Figura 63 – Função probabilidade dos pacotes de dados das chamadas efectuadas às 14 horas

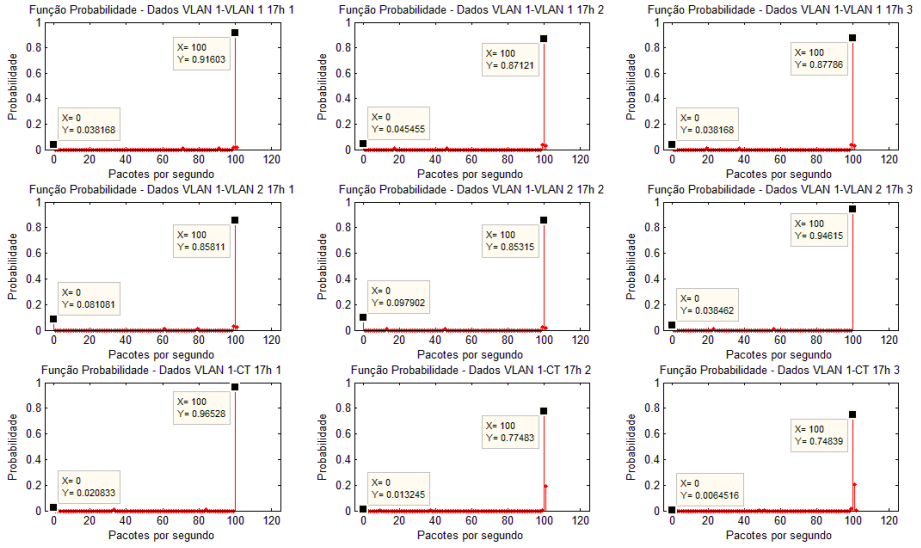

Figura 64 – Função probabilidade dos pacotes de dados das chamadas efectuadas às 17 horas

Além disso verificou-se que o valor que aparece com maior probabilidade é 100*pps*, o que era de esperar devido a utilização do codec G.711. Como já foi referido anteriormente, estes 100*pps* correspondem a ambos os fluxos, logo cada fluxo envia aproximadamente 50*pps* o que leva a uma largura de banda consumida de 64*kbps* (figura 65), pois cada pacote de dados é de 214*bytes*, dos quais 160 são de dados e 54 é cabeçalho:

$$50\ pps \times 160\ bytes\ \times 8\frac{bits}{bytes} = 64\ kbps$$

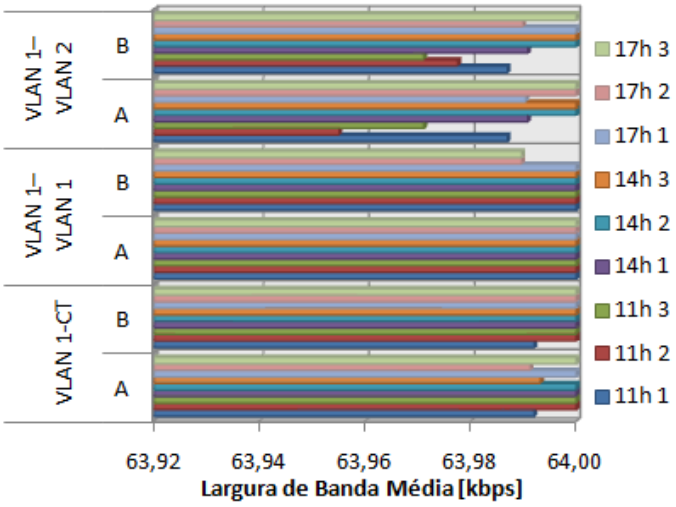

Figura 65 – Largura de banda média consumida por cada chamada

Verificou-se também que as únicas chamadas que apresentaram pacotes perdidos, foram as efectuadas às 11 horas entre um telefone da VLAN 1 e um da VLAN 2, sendo

estas as chamadas que apresentaram valores de largura de banda mais baixos, como se observa na figura 65.

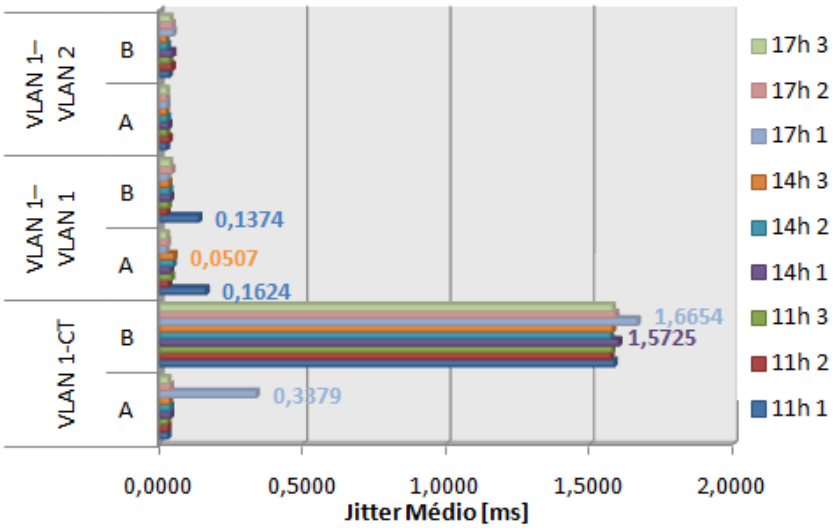

Figura 66 – *Jitter* médio obtido em cada chamada

Uma divergência entre os diferentes tipos de chamadas encontra-se nos valores do *jitter*. No gráfico da figura 66, verifica-se que o *jitter* do fluxo B das chamadas efectudas entre a VLAN 1 e a central telefónica, além de ser superior ao *jitter* do fluxo A da mesma chamada (figura 56), é também superior a todos os outros casos. Esta diferença nos valores do *jitter*, como já foi referido na secção 5.3.2.2.3, pode ser consequência da velocidade de conversão dos formatos.

5.4 Videoconferência e emissão de vídeo

Este sistema é composto por um equipamento de vídeo (*Tandberg Edge 95/85/75 MXP*) e um *Content Server* (*Tandberg Content Server*). Como é possível observar na figura 17 (secção 5.1), o *Tandberg Content Server* encontra-se na rede do IT, e o equipamento de vídeo encontra-se dentro de uma *DeMilitarized Zone* (DMZ)[7] na rede de Estúdios[8] da *Fundação para a Computação Científica Nacional* (FCCN). Os endereços IP utilizados por estes equipamentos são os seguintes:

Equipamento	Endereço IP
Equipamento de vídeo	193.136.252.137
Content Server	193.136.92.40

Tabela 14 – Endereços IP dos equipamentos da rede de videoconferência e emissão de vídeo

[7] Uma DMZ ou "zona desmilitarizada" é utilizada entre uma rede de confiança e uma rede que não seja de confiança, de forma a proteger a nossa rede de possíveis ataques.

[8] A rede de Estúdios da FCCN é uma rede que contém todos os equipamentos de videoconferência da rede RCTS (a RCTS é uma rede de investigação e ensino gerida pela FCCN – http://www.fccn.pt/).

O equipamento de vídeo, *Tandberg Edge 95/85/75 MXP*, é composto por uma câmara *High-Definition* (HD), um microfone, um controlo remoto e uma unidade HD. A unidade HD é o equipamento central e é neste que se liga o microfone, monitor e a câmara.

Figura 67 – *Tandberg* Edge 95/85/75 MXP
[Tandberg07b]

Este equipamento utiliza os protocolos H.323 (secção 2.2.3) e RTP (secção 2.3.1) para comunicar com o *Content Server:* o H.323 é utilizado para estabelecer e finalizar a sessão entre eles, e o RTP para a troca de pacotes de dados.

Como já foi referido na secção 2.3.1, o protocolo RTP utiliza pacotes separados para a transmissão de áudio e vídeo. É através do campo PT do cabeçalho que é especificado o *codec* utilizado: neste caso o *Payload* utilizado para o áudio é o G.722[9], e para o vídeo é o H.263[10].

Além dos protocolos, encontram-se definidos uma série de portos através dos quais os equipamentos devem trocar pacotes, segundo o protocolo em questão. No manual *Tandberg MXP: Administrator's Guide* estão definidos os portos utilizados por este sistema para uma chamada H.323. Neste caso, o porto utilizado depende da configuração escolhida, visto que existem dois tipos de configuração possíveis, **dinâmica** e **estática** [Tandberg07c]:

- **Dinâmica**: os portos são atribuídos aleatoriamente entre 2048 e 65535; no entanto, os portos H.323 dinâmicos são atribuídos aleatoriamente entre 11000 e 65535.
- **Estática**: irá utilizar os portos listados nas tabelas da figura seguinte.

[9] O *codec* G.722 corresponde ao *Payload Type* (PT) número 9, isto pode verificar-se no Anexo III.1. Este pertence à família dos *codecs* de áudio, é considerado um algoritmo de codificação e descodificação de áudio de alta qualidade [Tandberg07c]. O G.722 permite codificar a voz a 7*kHz* em três velocidades diferentes, 64, 56 e 48*kbps*. Sendo que neste caso a velocidade utilizada é de 64*kbps*.

[10] O H.263 é um algoritmo de codificação e descodificação de vídeo, desenhado para transmissões de baixa velocidade sem perda de qualidade [HR03].

Point-to-Point + DuoVideo		
Function	**Port**	**Type**
Gatekeeper Discovery (RAS)	1719	UDP
Q.931 Call Setup	1720	TCP
H.245	Range 5555-5574	TCP
Video	Range 2326-2385	UDP
Audio	Range 2326-2385	UDP
Data/FECC	Range 2326-2385	UDP

MultiSite + DuoVideo		
Function	**Port**	**Type**
Gatekeeper Discovery (RAS)	1719	UDP
Q.931 Call Setup	1720	TCP
H.245*	Range 5555-5574	TCP
Video	Range 2326-2485	UDP
Audio	Range 2326-2485	UDP
Data/FECC	Range 2326-2485	UDP

* While using MultiSite, if a site is disconnected and reconnected without terminating the entire conference, the next site to be connected will have a H.245 port outside of the specified range. If this functionality is required through a firewall, the range of TCP ports can be extended past 5564. However, if a site is disconnected and reconnected, without ending the conference enough times one can quickly end up outside of this range again.

Figura 68 – Portos utilizados pelo *Tandberg* MXP [Tandberg07c]

O vídeo captado pelo equipamento de vídeo é enviado para o *Tandberg Content Server*. Este equipamento serve de intermediário entre o equipamento de vídeo do IT e outros equipamentos de vídeo (caso se trate de uma videoconferência) ou entre o equipamento de vídeo do IT e utilizadores que desejem assistir ao vídeo via *Web* (caso seja emissão de vídeo), pois o *Content Server* disponibiliza os vídeos através da *Internet* (http://www.av.it.pt/farol/map-tele/), permitindo que os utilizadores assistam ao vídeo em tempo real, e também possibilita a opção de visualizar vídeos previamente gravados.

Como já foi referido anteriormente o *Content Server* comunica com o equipamento de vídeo através do protocolo de sinalização H.323 e do protocolo de dados RTP, mas para comunicar com os computadores que se ligam a ele para assistir ao vídeo utiliza protocolo MMS (secção 2.3.3) ou o protocolo RTSP (secção 2.3.2), conforme as definições do computador.

Figura 69 – *Tandberg* Content Server [Tandberg07a]

Segundo o manual do equipamento, o *Content Server* utiliza os portos da gama 2326-2365 para enviar vídeo para o equipamento de vídeo recorrendo ao protocolo RTP. Os outros portos utilizados pelo *Content Server* encontram-se na tabela da figura seguinte (por exemplo, utiliza o porto 554 para comunicar através do protocolo RTSP e utiliza o porto 1755 no caso do protocolo MMS).

Port	Protocols	Used by
80	TCP	The TANDBERG Content Server web application
443	TCP	The TANDBERG Content Server web application on SSL
554	TCP, UDP	WMS RTSP Server Control Protocol
1718	UDP	Gatekeeper discovery
1719	UDP	RAS
1720		Listen Port
1755	TCP, UDP	WMS MMS Server Control Protocol
2090	TCP	The TANDBERG Content Server database
3389	TCP	Remote Desktop Protocol
8008	TCP	The TANDBERG Content Server application
8080	TCP	WMS HTTP Server Control Protocol
8096	TCP	Windows Media Administration Site on SSL
8098	TCP	Windows Web Administration on SSL

Figura 70 – Tabela com os portos utilizados pelo TCS [Tandberg07a]

5.5 Testes efectuados ao sistema de emissão de vídeo

5.5.1 Metodologia

Efectuaram-se três tipos de testes, de forma a poder estudar os pacotes trocados no estabelecimento e finalização da sessão (Teste 3 – secção 5.5.1.1), e também observar as diferenças entre a transmissão de vídeo em tempo real (Teste 4 – secção 5.5.1.2) e a transmissão de um vídeo previamente gravado (Teste 5 – secção 5.5.1.3).

Para a realização destes testes, recorreu-se também às ferramentas de monitorização e gestão de redes referidas na secção 5.3.1.

5.5.1.1 Teste 3 – Estabelecimento e finalização da sessão

Foram realizadas capturas na VLAN 3 (ver figura 17), de forma a poder ver quais os pacotes trocados entre o *Content Server* e o equipamento de vídeo, durante o estabelecimento e finalização da sessão. Para este efeito o equipamento de vídeo foi ligado e desligado.

5.5.1.2 Teste 4 – Tráfego no *Content Server* durante a transmissão de um seminário

Realizaram-se capturas durante a transmissão de um dos seminários do programa de *Doutoramento MAP-Tele*. Os seminário deste programa foram transmitidos em tempo real através da página http://www.av.it.pt/farol/map-tele/. Isto é, as pessoas que desejavam assistir remotamente a estas conferências podiam ligar-se ao *Content Server* através da interface *Web,* e assim visualizar em directo o vídeo pois o vídeo gerado pelo

equipamento terminal de vídeo (*TANDBERG Edge 95/85/75 MXP*) é enviado para o *Content Server*, e é através deste que os vídeos são disponibilizados ao resto das pessoas através de uma interface *Web*.

Para poder estudar o tráfego que entra e sai do *Content Server*, estas capturas foram realizadas na porta do *switch* que liga ao *Content Server* (VLAN 2). Para este efeito recorreu-se a uma porta de *mirroring*.

5.5.1.3 Teste 5 – Tráfego no *Content Server* quando se descarrega um vídeo

Neste teste, realizaram-se capturas no mesmo ponto que no Teste 4, mas, neste caso, as capturas foram realizadas num dia em que não havia transmissão em directo de nenhum evento, pois a intenção do teste é perceber o que acontece quando são descarregados os vídeos de eventos pré-gravados (disponíveis em http://www.av.it.pt/farol/map-tele/).

Este teste foi efectuado utilizando três computadores com sistemas operativos e *browsers* diferentes, de forma a estudar o comportamento em cada caso. Os computadores utilizados foram os seguintes:

PC	Endereço IP	Sistema Operativo	*Browser*
1	193.136.93.84	*Windows XP*	*Internet Explorer 7*
2	172.16.41.75	*Windows Vista*	*Internet Explorer 7*
3	193.136.93.210	*Linux*	*Mozilla Firefox 2*

Tabela 15 – Informações dos PCs utilizados

5.5.2 Resultados e conclusões

Para análise das capturas efectuadas durante a transmissão de vídeo recorreu-se aos seguintes *scripts*:

- *editcap.sh* (Apêndice A.3) / *editcapRTP.sh* (Apêndice A.13): utilizados para editar as capturas; geram ficheiros com as distintas informações dos pacotes.
- *contadorPac.sh*: este é igual ao utilizado no VoIP para contar os pacotes (*contadorPacVoIP.sh* - Apêndice A.4).
- *contadorBytes.sh*: utilizado para contar o número de bytes transmitidos num determinado instante de tempo, igual ao utilizado no VoIP (*contadorBytesVoIP.sh* - Apêndice A.5).
- *editRTPinfo.sh*: utilizado para editar o campo *info* do RTP e seguidamente calcular o *jitter* e os pacotes perdidos (Apêndice A.6).
- *calcularJitter.sh*: devolve os valores do *jitter* (Apêndice A.7).
- *contaPacPerdidos.sh*: contabiliza o número de pacotes perdidos (Apêndice A.8).

• *OrganizarPacotesDados.sh*: no caso do vídeo verificou-se a existência de pacotes fora de ordem, isto não permitiu calcular correctamente o número de pacotes perdidos; para poder calcular correctamente o número de pacotes perdidos deve-se organizar os pacotes recorrendo a este *script* (Apêndice A.14).

• *Portos_src_dst.sh:* determina os portos utilizados em determinada chamada (consultar Apêndice A.15).

• *tamanhosPacVideo.sh*: contabiliza o número de pacotes segundo os tamanhos (Apêndice A.16).

5.5.2.1 Teste 3 – Estabelecimento e finalização da sessão

Estabelecimento da sessão:

O terminal A, neste caso o 193.136.252.137 (equipamento de vídeo), comunica com o *gatekeeper* (193.136.252.46) utilizando o protocolo RAS (H.225.0) através da mensagem *admissionRequest.* O *gatekeeper* responde a esta mensagem, indicando ao equipamento de vídeo se a sessão pode ou não ser efectuada. Caso a chamada possa ser efectuada utiliza o pacote *admissionConfirm,* caso contrário a mensagem enviada é *admissionReject.* Neste caso, foi respondida com um *admissionConfirm* (figura 71).

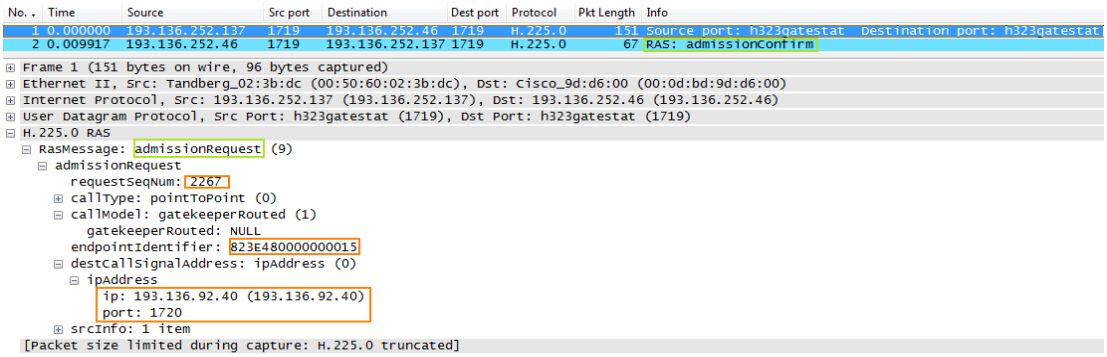

Figura 71 – Mensagens do protocolo RAS (H.225.0) no estabelecimento da sessão

A mensagem *admissionConfirm* utiliza o *requestSeqNum* para indicar a quem pertence esta resposta, e anuncia quais as características da chamada. Estas mensagens são trocadas através do porto UDP 1719 do equipamento de vídeo (*Gatekeeper Discovery* – RAS), conforme o especificado no manual do equipamento (figura 68).

O pacote *admissionRequest* contém informação sobre a sessão, como por exemplo, o número de pedido, o tipo de chamada, informação do destino, largura de banda, ID da conferência e muitas outras informações, as quais não são possíveis visualizar na captura pois esta foi feita limitando os tamanhos dos pacotes. É através desta mensagem que o

terminal A indica qual o endereço IP e o número do porto do terminal B, como é possível observar na figura 71 o porto indicado como porto do terminal B é o 1720, este é o porto de escuta (consultar a tabela da figura 70).

Após ter comunicado com o *gatekeeper*, o terminal A utiliza o protocolo Q.931 para estabelecer a chamada com o terminal B, através do porto 1720 (Q.931 *Call Setup*). Este porto foi previamente estabelecido através das mensagens de admissão do protocolo RAS H.225.0. Como é possível observar, o porto utilizado pelo terminal A (equipamento de vídeo) é o TCP 11002, o qual indica que os portos H.323 estão a ser atribuídos dinamicamente (secção 5.4).

No.	Time	Source	Src port	Destination	Dest port	Protocol	Pkt Length	Info
3	0.109977	193.136.252.137	11002	193.136.92.40	1720	TCP	74	11002 > h323hostcall [SYN] Seq=0 Win=16384 Len=0 MSS=1460 WS=0 TSV=65
4	0.109984	193.136.92.40	1720	193.136.252.137	11002	TCP	60	h323hostcall > 11002 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
5	0.111258	193.136.252.137	11002	193.136.92.40	1720	TCP	60	11002 > h323hostcall [ACK] Seq=1 Ack=1 Win=17520 Len=0
6	0.130619	193.136.252.137	11002	193.136.92.40	1720	Q.931	1283	SETUP[Packet size limited during capture]
7	0.130626	193.136.92.40	1720	193.136.252.137	11002	TCP	60	h323hostcall > 11002 [ACK] Seq=1 Ack=1230 Win=6963 Len=0
8	0.166826	193.136.92.40	1720	193.136.252.137	11002	TCP	60	[TCP segment of a reassembled PDU]
9	0.166833	193.136.92.40	1720	193.136.252.137	11002	TCP	137	h323hostcall > 11002 [PSH, ACK] Seq=5 Ack=1230 Win=8192 Len=83
10	0.166836	193.136.252.137	11002	193.136.92.40	1720	TCP	60	11002 > h323hostcall [ACK] Seq=1230 Ack=88 Win=17433 Len=0
11	0.188529	193.136.92.40	1720	193.136.252.137	11002	TCP	60	[TCP segment of a reassembled PDU]
12	0.188535	193.136.92.40	1720	193.136.252.137	11002	TCP	169	h323hostcall > 11002 [PSH, ACK] Seq=92 Ack=1230 Win=8192 Len=115
13	0.188539	193.136.252.137	11002	193.136.92.40	1720	TCP	60	11002 > h323hostcall [ACK] Seq=1230 Ack=207 Win=17401 Len=0

Figura 72 – Mensagens do protocolo Q.931 no estabelecimento da sessão

Durante a sessão:

Os terminais, neste caso o equipamento de vídeo (193.136.252.137 – terminal A) e o *Content Server* (193.136.92.40 – terminal B), trocam os pacotes de áudio e vídeo utilizando a gama de portos definida nos manuais dos equipamentos (figura 68). São utilizados quatro portos para cada um dos terminais (figura seguinte): um para cada tipo de dados, áudio e vídeo, e um para controlo de cada tipo de dados.

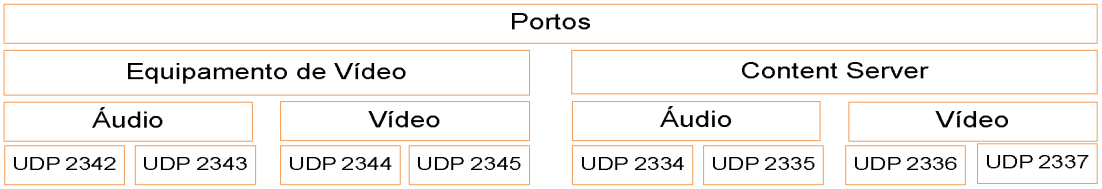

Figura 73 – Portos utilizados para a troca de dados

No caso do equipamento de vídeo (193.136.252.137), o porto usado para transportar o áudio é o 2342 e o vídeo é o 2344 (isto verifica-se por observação dos tamanhos dos pacotes, pois no caso do áudio os pacotes são todos de 214 *bytes* e no caso do vídeo os pacotes variam entre 73 e 1456 *bytes*). Os pacotes de controlo de áudio utilizam o porto 2343 (2342+1) e os pacotes de controlo de vídeo utilizam o porto 2345 (2344+1).

O *Content Server* utiliza para transporte e controlo de áudio o 2334 e 2335, respectivamente, e no caso do vídeo, utiliza para o transporte o porto 2336 e para controlo o 2337.

Os portos utilizados encontram-se dentro da gama definida nos manuais dos equipamentos (secção 5.4). Quanto aos *codecs* utilizados verifica-se que no áudio utiliza-se o G.722 e no vídeo o H.263.

No.	Time	Source	Src port	Destination	Dest port	Protocol	Pkt Length	Info
138	7.370636	193.136.92.40	2334	193.136.252.137	2342	RTP	214	PT=ITU-T G.722, SSRC=0x864D2BDD, Seq=6965, Time=56320
139	7.373960	193.136.252.137	2344	193.136.92.40	2336	H.263	1343	PT=ITU-T H.263, SSRC=0x6B548D9B, Seq=54000, Time=3393189460 MODE A H26
140	7.376073	193.136.92.40	2337	193.136.252.137	2345	RTCP	146	Sender Report
141	7.381649	193.136.252.137	2342	193.136.92.40	2334	RTP	214	PT=ITU-T G.722, SSRC=0x86871E84, Seq=37778, Time=234053482
142	7.382702	193.136.252.137	2344	193.136.92.40	2336	H.263	171	PT=ITU-T H.263, SSRC=0x6B548D9B, Seq=54001, Time=3393189460, Mark MODE
143	7.391746	193.136.92.40	2334	193.136.252.137	2342	RTP	214	PT=ITU-T G.722, SSRC=0x864D2BDD, Seq=6966, Time=56480
144	7.400578	193.136.252.137	2342	193.136.92.40	2334	RTP	214	PT=ITU-T G.722, SSRC=0x86871E84, Seq=37779, Time=234053642
145	7.412250	193.136.92.40	2334	193.136.252.137	2342	RTP	214	PT=ITU-T G.722, SSRC=0x864D2BDD, Seq=6967, Time=56640
146	7.421723	193.136.252.137	2342	193.136.92.40	2334	RTP	214	PT=ITU-T G.722, SSRC=0x86871E84, Seq=37780, Time=234053802
147	7.423996	193.136.252.137	2344	193.136.92.40	2336	H.263	1252	PT=ITU-T H.263, SSRC=0x6B548D9B, Seq=54002, Time=3393195400 MODE A H26
148	7.432182	193.136.252.137	2343	193.136.92.40	2335	RTCP	134	Sender Report
149	7.432190	193.136.252.137	2345	193.136.92.40	2337	RTCP	134	Sender Report

Figura 74 – Pacotes de dados (áudio e vídeo)

Como é possível observar na figura anterior, são transmitidos pacotes de dados em ambos sentidos, isto é do equipamento de vídeo (terminal A – 193.136.252.137) para o *Content Server* (terminal B – 193.136.92.40), e vice-versa. Tratando-se de uma transmissão de vídeo era suposto observar tráfego num único sentido. Neste caso, existe tráfego no sentido contrário porque o *Content Server* envia para o equipamento de vídeo uma imagem a dizer "*Recording*" que indica que a sessão está activa. O tráfego gerado por esta imagem representa 20% do tráfego total.

Figura 75 – Imagem que o *Content Server* envia para o equipamento de vídeo

Finalização da sessão:

Na figura seguinte observam-se as mensagens trocadas no processo de finalização da chamada. Na parte final, o terminal A envia ao *gatekeeper* uma mensagem a indicar que a sessão foi concluída, *disengageRequest*, e este aceita o pedido respondendo com *disengageConfirm*.

No.	Time	Source	Src port	Destination	Dest port	Protocol	Pkt Length	Info
1180	7.551266	193.136.252.137	11003	193.136.92.40	3232	TCP	60	11003 > mdtp [ACK] Seq=1017 Ack=450 Win=17520 Len=0
1181	7.559672	193.136.252.137	11002	193.136.92.40	1720	H.225.0	104	[Packet size limited during capture]
1182	7.559678	193.136.92.40	1720	193.136.252.137	11002	TCP	60	h323hostcall > 11002 [ACK] Seq=207 Ack=1280 Win=8146 Len=0
1183	7.570025	193.136.252.137	11002	193.136.92.40	1720	TCP	60	11002 > h323hostcall [FIN, ACK] Seq=1280 Ack=207 Win=17520 Len=0
1184	7.570031	193.136.92.40	1720	193.136.252.137	11002	TCP	60	h323hostcall > 11002 [FIN, ACK] Seq=207 Ack=1281 Win=8192 Len=0
1185	7.570034	193.136.252.137	11002	193.136.92.40	1720	TCP	60	11002 > h323hostcall [ACK] Seq=1281 Ack=208 Win=17520 Len=0
1186	7.580360	193.136.252.137	1719	193.136.252.46	1719	H.225.0	119	Source port: h323gatestat Destination port: h323gatestat[Packet
1187	7.592915	193.136.252.46	1719	193.136.252.137	1719	H.225.0	60	RAS: disengageConfirm

Figura 76 – Mensagens trocadas durante a finalização da sessão

5.5.2.2 Teste 4 – Tráfego no *Content Server* durante a transmissão de um seminário

Capturou-se o tráfego gerado pelo equipamento de vídeo, o *Content Server* e os equipamentos terminais que assistiram ao seminário através do *site* http://www.av.it.pt/farol/map-tele/. Este seminário efectuou-se no dia 7 de Março de 2008 e teve uma duração de 1 hora e 50 minutos aproximadamente.

Ao longo da captura, verificou-se que nem todas as pessoas assistiram ao seminário do inicio ao fim. No gráfico seguinte observa-se que no máximo existiram dez pessoas, em simultâneo e no total, assistiram ao seminário 17 pessoas.

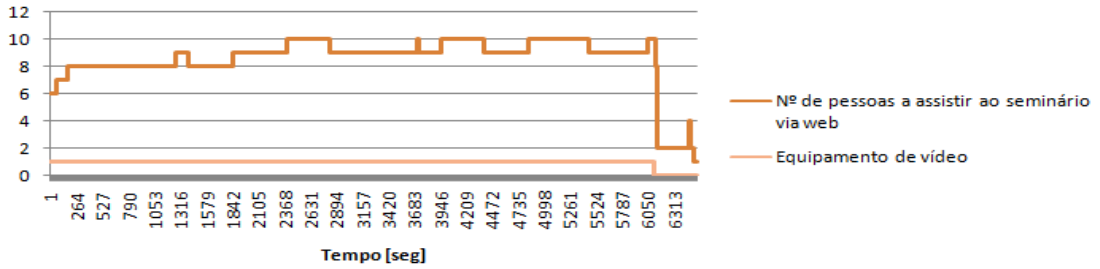

Figura 77 – Número de pessoas a assistir ao seminário através do *site*

Em média, os utilizadores mantiveram sessões de aproximadamente 57 minutos e como se pode observar no gráfico anterior, na maior parte do tempo existiram pelo menos nove pessoas a assistir a este seminário através da *Web*.

Como tem sido referido ao longo deste trabalho, o vídeo captado pelo equipamento de videoconferência é enviado para o *Content Server* e este distribui o vídeo aos utilizadores através do *site* do IT (estes utilizadores são referidos como equipamentos terminais). Assim, o tráfego pode ser dividido em duas partes: uma primeira parte que se refere ao tráfego entre o equipamento de vídeo e o *Content Server*, e uma segunda parte que corresponde aos pacotes trocados entre o *Content Server* e os equipamentos terminais.

Tráfego entre o *Content Server* e o equipamento de vídeo:

Entre o *Content Server* e o equipamento de vídeo, além do tráfego de sinalização que há no estabelecimento e finalização, como já foi mostrado no teste 3 (secção 5.5.2.1), existe também tráfego de dados, o qual se divide em dois: **vídeo** e **áudio**.

A quantidade de pacotes de **áudio** transmitidos ao longo de um segundo é aproximadamente constante durante a transmissão do seminário e a quantidade de pacotes de **vídeo** oscila. Observa-se também (figura 78) que na maior parte do tempo existem mais pacotes de áudio do que de vídeo. Isto verifica-se olhando para o valor médio de pacotes transmitidos ao longo de cada segundo no fluxo de áudio e de vídeo, pois em média são transmitidos menos pacotes de vídeo do que de áudio (tabela 16).

No caso da quantidade de bytes por segundo do fluxo de áudio e de vídeo (figura 79), a situação inverte-se pois como se pode observar existem mais bytes de vídeo do que de áudio. Isto deve-se ao facto dos pacotes de áudio serem de tamanho constante, 214*bytes* e os de vídeo variam entre 73 e 1456*bytes* e como se pode observar no gráfico da figura 80,

82% dos pacotes de vídeo têm tamanho superior a 256*bytes* (logo a maior parte dos pacotes de vídeo são maiores que os pacotes de áudio).

	Áudio [*pps*]	Vídeo [*pps*]		Áudio [*pps*]	Vídeo [*pps*]
Valor Máximo	103	154	Moda	0	0
Média	26,92	16,34	Variância	1889,31	813,67
Valor Mínimo	0	0	Desvio Padrão	43,47	28,52
Mediana	0	0			

Tabela 16 – Valores estatísticos do fluxo de áudio e vídeo

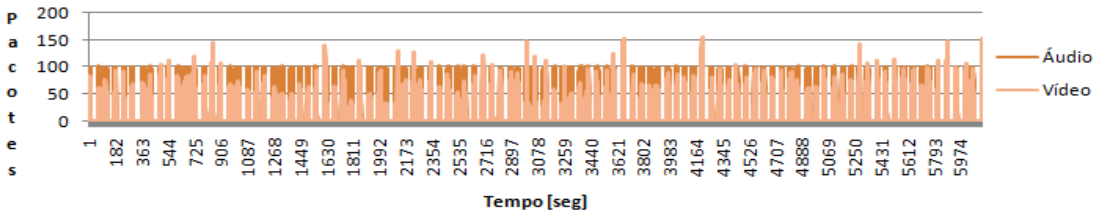

Figura 78 – Gráfico do tráfego de áudio e vídeo em pacotes por segundo

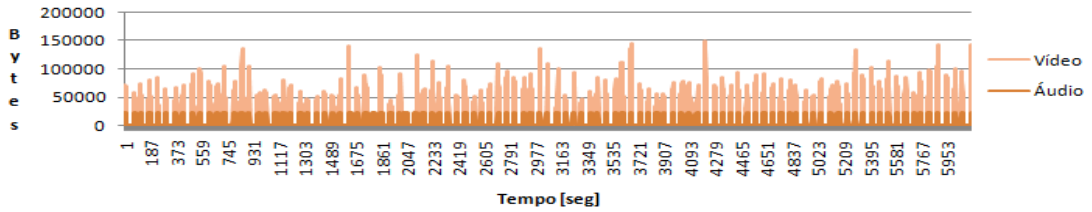

Figura 79 – Gráfico do tráfego de áudio e vídeo em bytes por segundo

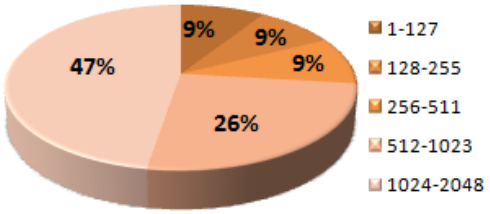

Figura 80 – Gráfico dos tamanhos dos pacotes de vídeo em *Bytes*

Como foi referido no teste 3 (secção 5.5.2.1), o fluxo de dados vai desde o equipamento de vídeo para o *Content Server* (fluxo A) e do *Content Server* para o equipamento de vídeo (fluxo B). Além disso, os dados dividem-se em dois tipos de pacotes, áudio e vídeo. No caso do áudio, da totalidade dos bytes transmitidos, 50,63% são do fluxo A e 49,37% do fluxo B. Quanto ao vídeo, o fluxo A transmite 91,92% dos bytes, e os restantes 8,08% correspondem ao fluxo B.

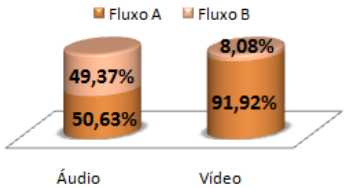

Figura 81 – Gráfico da percentagem de bytes de áudio e vídeo transmitidos por cada tipo de fluxo

Como é possível observar nos gráficos seguintes (figuras 82 e 83), a largura de banda ocupada no tráfego de áudio é igual tanto para o fluxo A como para o fluxo B, enquanto que no caso do vídeo verifica-se que a largura de banda ocupada pelo fluxo A é superior à ocupada pelo fluxo B.

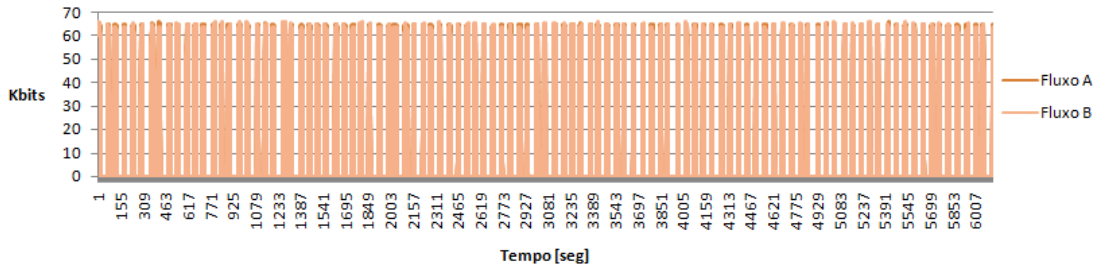

Figura 82 – Gráfico da largura de banda dos fluxos de áudio

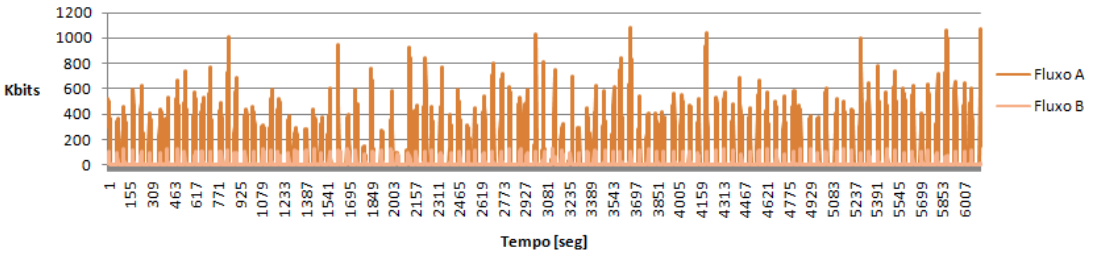

Figura 83 – Gráfico da largura de banda dos fluxos de vídeo

No caso do **áudio,** as larguras de banda utilizadas pelo fluxo A e fluxo B são muito semelhantes dado que os equipamentos não têm activos nenhum mecanismo de supressão de silêncio (é por isto que mesmo quando o que o fluxo B está a transportar é uma imagem, há sempre presença de pacotes de áudio). Além disto, no gráfico da largura de banda (figura 82), verifica-se que a largura de banda consumida por ambos os fluxos é de 64*kbps*, o que era de esperar, visto que o *codec* utilizado para a transmissão de áudio é o G.722 a 64*kbps*.

No caso do **vídeo**, o *codec* utilizado é o H.263. Nesta situação, há uma diferença notável quanto à largura de banda utilizada, visto que o fluxo B atingiu, no máximo, os 131,05*kbps* enquanto que o fluxo A atingiu 1095,92*kbps*. Esta diferença deve-se ao facto do fluxo A transmitir o vídeo do seminário e o fluxo B está a transmitir unicamente uma imagem (figura 75). Além disto e como se pode observar no gráfico seguinte, o fluxo A é maioritariamente composto por pacotes grandes (maiores que 1024*bytes*) enquanto que no fluxo B só 8,78% dos pacotes estão incluídos nessa gama, pois neste caso a maior parte dos pacotes pequenos (menores que 127*bytes*).

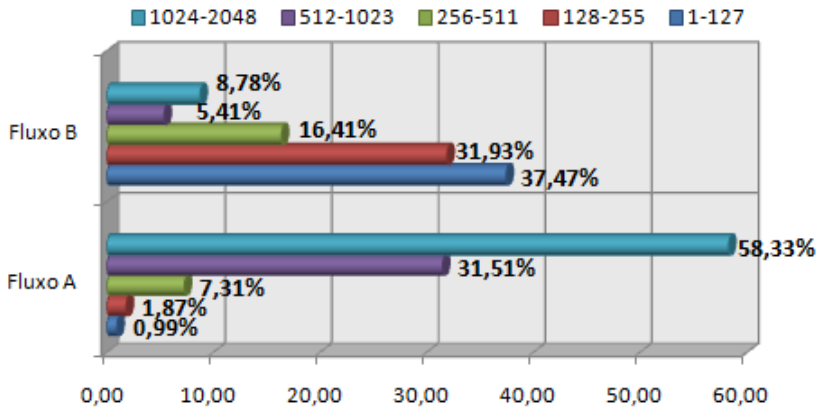

Figura 84 – Gráfico das percentagens de pacotes de cada fluxo segundo os tamanhos [bytes]

Tráfego entre o *Content Server* e os equipamentos terminais:

Como já foi referido anteriormente, o *Content Server* disponibiliza o vídeo do seminário em directo através do *site* do IT. Entre o *Content Server* e cada um dos equipamentos terminais, há uma troca de pacotes de dados e de controlo, os dados vão desde o *Content Server* para o equipamento terminal, e os pacotes de controlo podem ir em qualquer um dos sentidos. Cada equipamento terminal utiliza um protocolo para comunicar com o *Content Server*, pode ser o MMS ou o RTSP, como se verifica na tabela do Apêndice D.

Como é possível observar nos seguintes gráficos (figuras 85 e 86), a quantidade de pacotes que entra no *Content Server* (CS) é inferior à quantidade de pacotes que sai do mesmo, a diferença é ainda mais notável quando se olha para os *bytes*, pois os pacotes que vão dos equipamentos terminais para *Content Server* são de controlo e no geral são de menor tamanho do que os de dados.

No gráfico da figura 87 verifica-se que os pacotes que vão do *Content Server* para os equipamentos terminais (isto é, tráfego que sai do CS) são de maior tamanho do que os que vão no sentido oposto. Pois a maioria dos pacotes que entram no *Content Server* têm entre 1 e 127*bytes*, e no caso dos pacotes que vão desde o *Content Server* para os equipamentos terminais, a maior parte destes têm entre 512 e 2048*bytes*.

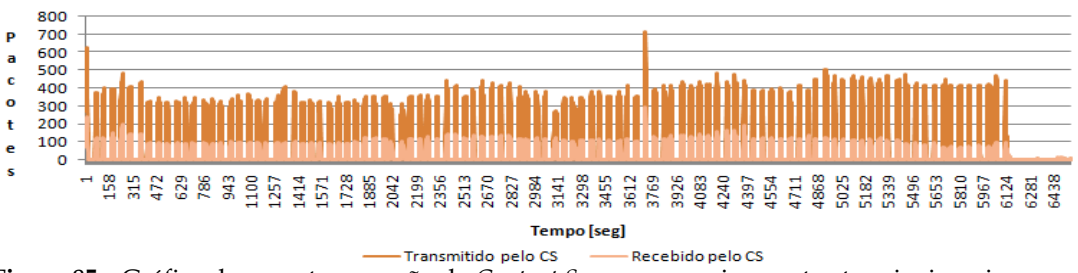

Figura 85 – Gráfico dos pacotes que vão do *Content Server* aos equipamentos terminais e vice-versa

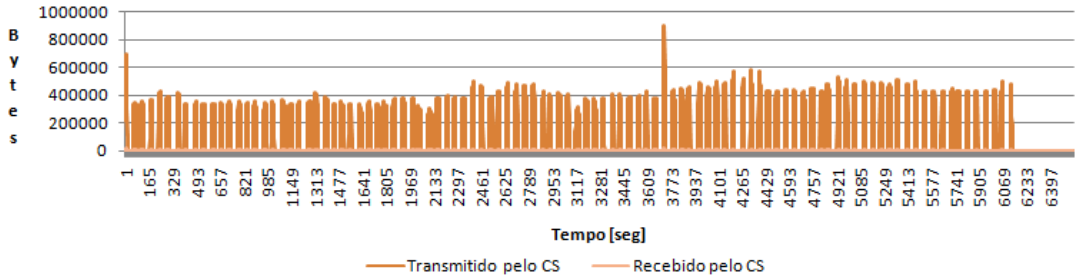

Figura 86 – Gráfico dos bytes que vão do *Content Server* aos equipamentos terminais e vice-versa

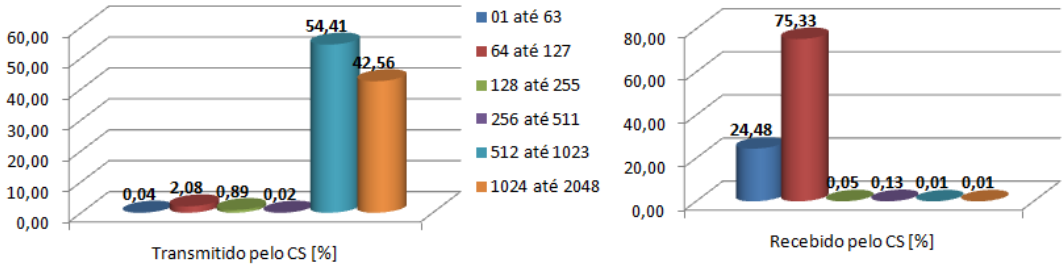

Figura 87 – Gráfico das percentagens dos tamanhos dos pacotes do fluxo que sai do *Content Server* para os equipamentos terminais e vice-versa

5.5.2.3 Teste 5 – Tráfego no *Content Server* quando se descarrega um vídeo

Neste caso efectuaram-se três tipos de *downloads*, recorrendo a computadores com sistemas operativos diferentes, como se mostra na seguinte tabela. Tanto no caso do PC com *Windows XP* como no *Windows Vista* verificou-se que ambos recorreram ao protocolo, RTSP (secção 2.3.2), para o *download* do vídeo, enquanto que o PC com sistema operativo *Linux* utilizou o protocolo MMS (secção 2.3.3).

PC	Endereço IP	Sistema Operativo	*Browser*
1	193.136.93.84	*Windows XP*	*Internet Explorer 7*
2	172.16.41.75	*Windows Vista*	*Internet Explorer 7*
3	193.136.93.210	*Linux*	*Mozilla Firefox 2*

Tabela 17 – Informações dos PCs utilizados

Como é possível observar nas figuras 88 e 89, os computadores com *Internet Explorer 7* (isto é, o PC 1 e PC 2) utilizaram o protocolo RTSP recorrendo à porta TCP 554 do *Content Server* (193.136.92.40). Esta porta corresponde à porta definida no manual do equipamento para o protocolo RTSP (figura 70).

No caso do PC 3 (figura 90), o protocolo utilizado para o transporte dos dados, foi o MMS (este protocolo no *Wireshark* aparece com o nome MSMMS), e como era de esperar, no caso deste protocolo, o porto utilizado é o TCP 1755 do *Content Server* (figura 70).

No.	Time	Source	Src port	Destination	Dest port	Protocol	Pkt Length	Info
74	7.492242	193.136.92.40	554	193.136.93.84	3884	RTSP	1514	Continuation
75	7.492245	193.136.92.40	554	193.136.93.84	3884	RTSP	1514	Continuation
76	7.492248	193.136.93.84	3884	193.136.92.40	554	TCP	60	softrack-meter > rtsp [ACK] Seq=3498 Ack=35368 Win=65535 Len=0
77	7.492252	193.136.92.40	554	193.136.93.84	3884	RTSP	1514	Continuation
78	7.492857	193.136.92.40	554	193.136.93.84	3884	RTSP	1514	Continuation
79	7.492864	193.136.92.40	554	193.136.93.84	3884	RTSP	1514	Continuation
80	7.492866	193.136.93.84	3884	193.136.92.40	554	TCP	60	softrack-meter > rtsp [ACK] Seq=3498 Ack=38288 Win=65535 Len=0
81	7.492870	193.136.92.40	554	193.136.93.84	3884	RTSP	1514	Continuation
82	7.492873	193.136.92.40	554	193.136.93.84	3884	RTSP	1514	Continuation
83	7.492876	193.136.92.40	554	193.136.93.84	3884	RTSP	1514	Continuation
84	7.492879	193.136.93.84	3884	193.136.92.40	554	TCP	60	softrack-meter > rtsp [ACK] Seq=3498 Ack=41208 Win=65535 Len=0
85	7.493362	193.136.92.40	554	193.136.93.84	3884	RTSP	1514	Continuation
86	7.493369	193.136.92.40	554	193.136.93.84	3884	RTSP	1514	Continuation

Figura 88 – Pacotes transmitidos pelo PC 1

No.	Time	Source	Src port	Destination	Dest port	Protocol	Pkt Length	Info
140	0.310789	193.136.92.40	554	172.16.41.75	54153	RTSP	1514	Continuation
141	0.315538	172.16.41.75	54153	193.136.92.40	554	TCP	60	54153 > rtsp [ACK] Seq=1 Ack=132988 Win=68 Len=0
142	0.315545	193.136.92.40	554	172.16.41.75	54153	RTSP	1514	Continuation
143	0.315548	193.136.92.40	554	172.16.41.75	54153	RTSP	1514	Continuation
144	0.315551	193.136.92.40	554	172.16.41.75	54153	RTSP	1514	Continuation
145	0.317456	172.16.41.75	54153	193.136.92.40	554	TCP	60	54153 > rtsp [ACK] Seq=1 Ack=137368 Win=68 Len=0
146	0.317462	193.136.92.40	554	172.16.41.75	54153	RTSP	1514	Continuation
147	0.317466	193.136.92.40	554	172.16.41.75	54153	RTSP	1514	Continuation
148	0.317468	193.136.92.40	554	172.16.41.75	54153	RTSP	1514	Continuation
149	0.318761	172.16.41.75	54153	193.136.92.40	554	TCP	60	54153 > rtsp [ACK] Seq=1 Ack=143208 Win=68 Len=0
150	0.318769	193.136.92.40	554	172.16.41.75	54153	RTSP	1514	Continuation
151	0.318772	193.136.92.40	554	172.16.41.75	54153	RTSP	1514	Continuation[Packet size limited during capture]
152	0.318775	193.136.92.40	554	172.16.41.75	54153	RTSP	1514	Continuation
153	0.319939	193.136.92.40	554	172.16.41.75	54153	RTSP	1514	Continuation

Figura 89 – Pacotes transmitidos pelo PC 2

No.	Time	Source	Src port	Destination	Dest port	Protocol	Pkt Length	Info
57	0.954655	193.136.92.40	1755	193.136.93.210	48705	MSMMS	1514	Data: seq=2254806884, len=00580
58	0.955124	193.136.92.40	1755	193.136.93.210	48705	TCP	1514	ms-streaming > 48705 [ACK] Seq=44386 Ack=761 Win=64775 Len=1448 TSV=1955685 TSER=1
59	0.955132	193.136.92.40	1755	193.136.93.210	48705	TCP	1514	ms-streaming > 48705 [ACK] Seq=45834 Ack=761 Win=64775 Len=1448 TSV=1955685 TSER=1
60	0.955138	193.136.92.40	1755	193.136.93.210	48705	MSMMS	1514	Data: seq=2565519066, len=00881
61	0.955143	193.136.92.40	1755	193.136.93.210	48705	TCP	1514	ms-streaming > 48705 [ACK] Seq=48730 Ack=761 Win=64775 Len=1448 TSV=1955685 TSER=1
62	0.955744	193.136.92.40	1755	193.136.93.210	48705	TCP	1514	ms-streaming > 48705 [ACK] Seq=50178 Ack=761 Win=64775 Len=1448 TSV=1955685 TSER=1
63	0.955755	193.136.92.40	1755	193.136.93.210	48705	TCP	1514	ms-streaming > 48705 [ACK] Seq=51626 Ack=761 Win=64775 Len=1448 TSV=1955685 TSER=1
64	0.955760	193.136.92.40	1755	193.136.93.210	48705	TCP	1514	ms-streaming > 48705 [ACK] Seq=53074 Ack=761 Win=64775 Len=1448 TSV=1955685 TSER=1
65	0.955765	193.136.92.40	1755	193.136.93.210	48705	TCP	1514	ms-streaming > 48705 [ACK] Seq=54522 Ack=761 Win=64775 Len=1448 TSV=1955685 TSER=1
66	0.955770	193.136.92.40	1755	193.136.93.210	48705	TCP	1514	ms-streaming > 48705 [ACK] Seq=55970 Ack=761 Win=64775 Len=1448 TSV=1955685 TSER=1
67	0.956934	193.136.92.40	1755	193.136.93.210	48705	MSMMS	1514	Data: seq=3057682028, len=00068
68	0.995824	193.136.93.210	48705	193.136.92.40	1755	TCP	66	48705 > ms-streaming [ACK] Seq=761 Ack=58866 Win=10688 Len=0 TSV=1573581 TSER=1955
69	0.995855	193.136.92.40	1755	193.136.93.210	48705	TCP	1514	ms-streaming > 48705 [ACK] Seq=58866 Ack=761 Win=64775 Len=1448 TSV=1955685 TSER=1

Figura 90 – Pacotes transmitidos pelo PC 3

Além disto verificou-se que, o comportamento do tráfego ao longo do *download* do vídeo é similar ao comportamento do tráfego entre o *Content Server* e os equipamentos terminais do teste anterior (teste 4). Como se observa na figura seguinte, a quantidade de *bytes* transmitida pelo *Content Server* é muito superior à recebida.

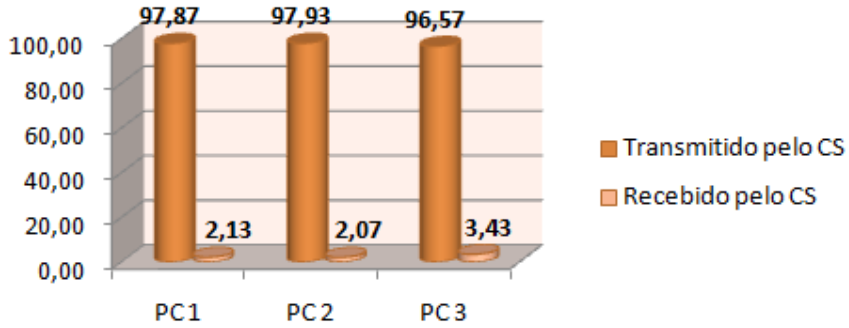

Figura 91 – Percentagem de *bytes* transmitidos e recebidos pelo CS em cada um dos casos

6 DTMS-P2P

6.1 Testes efectuados

6.1.1 Metodologia

Foram efectuados alguns testes de forma a mostrar a funcionalidade desta ferramenta. Para isto utilizaram-se dois grupos de medição (grupo 0 e grupo 1), formados por três pontos de prova (*probes*) e um cliente, como se pode observar na figura seguinte.

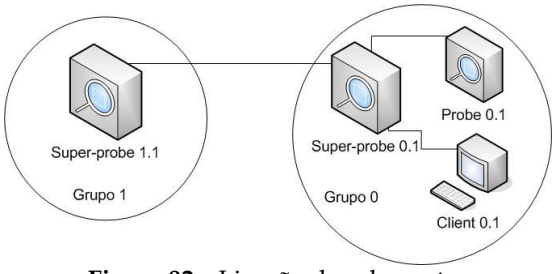

Figura 92 – Ligação dos elementos

De forma a obter o sistema da figura anterior, instalaram-se os computadores nos lugares indicados na figura seguinte. Nesta figura, também são apresentados os valores da máscara de rede e *gateway* atribuídos por defeito a cada um dos PCs.

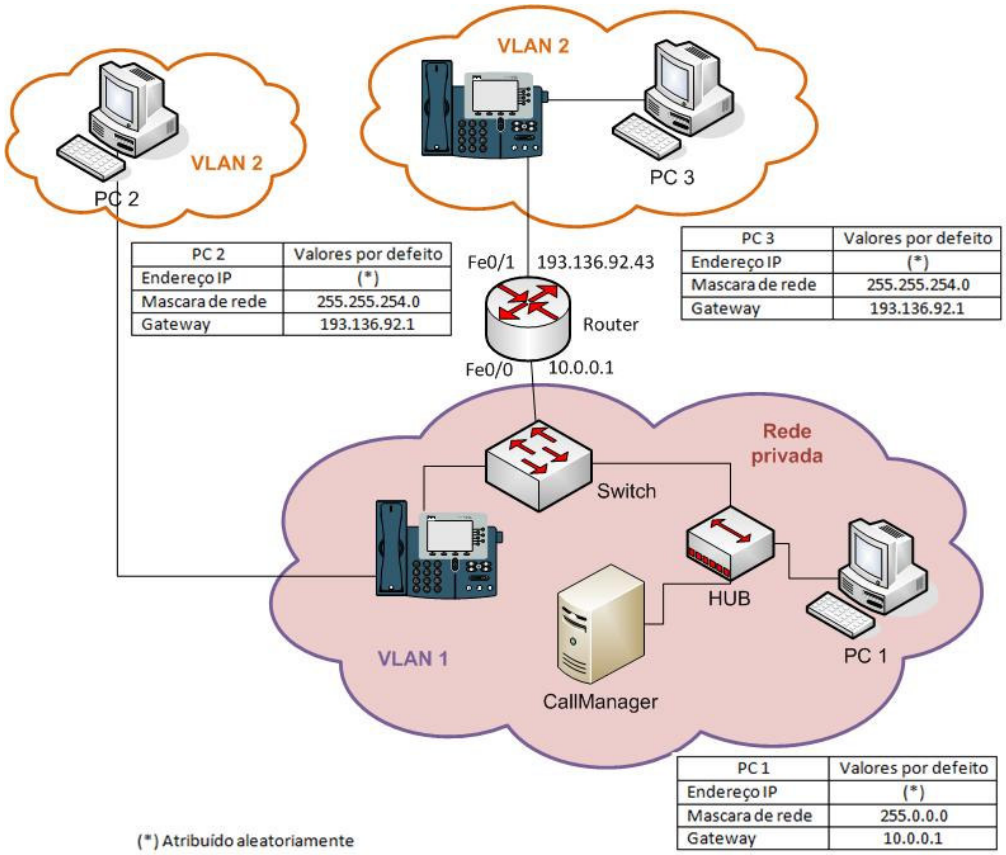

Figura 93 – Montagem utilizada para os testes efectuados com o DTMS-P2P

A **Super-probe 1.1** corresponde ao ponto de medição colocado junto ao *CallManager* na rede 10.0.0.0/8 (PC 1 da figura anterior). Este é o único elemento do grupo 1.

O grupo 0 contém dois pontos de medição: uma *super-probe* (**Super-probe 0.1)** que corre num PC ligado no telefone VoIP da rede 10.0.0.0/8 (PC 2), e uma *probe* (**Probe 0.1)** que corre num outro PC, este último está ligado num telefone VoIP da rede do IT (PC 3). Neste grupo encontra-se também o cliente (**Client 0.1**).

6.1.1.1 Teste 6 – Utilizando os módulos de monitorização do *Linux* instalados por defeito no DTMS-P2P

Configuraram-se os três computadores com os seguintes endereços:

PC	Observações	Informações do computador		
		Endereço IP	Mascara de rede	*Gateway*
PC 1	Ligado ao HUB para medir tráfego do *CallManager*	10.0.0.73	255.0.0.0	10.0.0.1
PC 2	Ligado ao telefone VoIP da rede VoIP (10.0.0.96)	193.136.93.103	255.255.254.0	193.136.92.43
PC 3	Ligado ao telefone VoIP da rede do IT (193.136.93.109)	193.136.93.158	255.255.254.0	193.136.92.43

Tabela 18 – Configurações dos computadores

Como é possível observar, o *gateway* dos PCs 2 e 3 não é o 193.136.92.1, *gateway* utilizado por defeito na VLAN 2 (rede do IT), mas sim o 193.136.92.43, este corresponde ao endereço IP da interface *Fe0/1* do *router*. Esta alteração no *gateway* dos computadores que se encontravam na VLAN 2, deve-se ao facto do *CallManager* estar dentro de uma rede privada[11].

Isto foi necessário, pois o DTMS-P2P envia periodicamente *ping* a todos os nós para assim poder calcular o RTT e poder dar origem ao *Light Data File* (contém os parâmetros do sistema de monitorização e o RTT, com este ficheiro torna-se possível fazer um desenho da rede).

Caso o *gateway* dos PCs 2 e 3 não tivessem sido alterados não seria possível efectuar *ping* para dentro da rede 10.0.0.0/8, e o DTMS-P2P não conseguiria calcular o RTT, como consequência não iria gerar o *Light Data File*.

[11] O *CallManager* e todos os elementos da VLAN 1 se encontram detrás de NAT, pois esta é uma rede privada, isto é possível verificar através da figura 21 na secção 5.2 e na figura 93 desta secção.

Antes de iniciar qualquer nó, alterou-se o ficheiro *FileOfKnownNodes.xml* do directório *FilesOfKnownNodes* em todos os nós, de forma a indicar quais são as *super-probes* de cada grupo. Como mostra a figura seguinte, basta inserir os dados das *super-probes*, pois este ficheiro actualiza-se automaticamente ao iniciar cada um dos nós, armazenando a informação dos nós ligados em cada grupo.

```
- <FileOfKnownNodes>
  - <GroupID id="00000000000000000000000000000001">
    - <super-probes>
      - <node>
          <IPVN>4</IPVN>
          <IP>10.0.0.73</IP>
          <port>22368</port>
        </node>
      </super-probes>
      <probes />
    </GroupID>
  - <GroupID id="00000000000000000000000000000000">
    - <super-probes>
      - <node>
          <IPVN>4</IPVN>
          <IP>193.136.93.103</IP>
          <port>22368</port>
        </node>
      </super-probes>
      <probes />
    </GroupID>
</FileOfKnownNodes>
```

Figura 94 – Alteração do ficheiro *FileOfKnownNodes.xml* do directório *FilesOfKnownNodes*

Após ter configurado o *FileOfKnownNodes.xml* em todos os nós, iniciou-se cada um dos nós utilizando os comandos indicados na tabela seguinte. Utilizou-se a versão não gráfica, pois o modo gráfico bloqueia ao ser executado no *Linux*, isto não acontece quando é executado o modo gráfico no *Windows*. Suspeita-se de algum problema ou incompatibilidade com o *Java* do *Linux*.

PC	Comando utilizado
1	*java dtms_p2p.DTMS_P2P_Node –g 01 –scf SupportedMonitoringModulesLinux.xml –m sp*
2	*java dtms_p2p.DTMS_P2P_Node –scf SupportedMonitoringModulesLinux.xml –m sp*
3	*java dtms_p2p.DTMS_P2P_Node –scf SupportedMonitoringModulesLinux.xml*

Tabela 19 – Comandos utilizados para iniciar cada um dos nós

Da tabela anterior pode retirar-se a seguinte informação:

- No caso do PC 1 indicou-se o grupo (*-g 01*) pois por defeito eles iniciam no grupo 0.
- Só foi indicado o modo (*-m*) no caso dos PCs 1 e 2, pois por defeito eles iniciam no modo *probe* (*p*) e neste caso desejava-se que estes iniciassem no modo *super-probe* (*sp*).

- Visto que os testes foram realizado em *Linux*, e por defeito os nós utilizam os módulos de monitorização do *Windows* (*SupportedMonitoringModules.xml*), neste caso foi necessário indicar em todos os nós que os módulos de monitorização a utilizar eram os do *Linux* (*SupportedMonitoringModulesLinux.xml*) através do *–scf*.

Para poder iniciar o *Client* no PC 1 procedeu-se à alteração do ficheiro *FileOfKnownNodes.xml* do directório *clientFilesOfKnownNodes*, para isso bastou inserir neste ficheiro as *super-probes*, como mostra a figura 94. Seguidamente iniciou-se o cliente, utilizando o seguinte comando: *java dtms_p2p.DTMS_P2P_Client*

Desta forma o cliente inicia considerando as opções por defeito, isto é liga-se ao grupo 0 e utiliza a porta 21164.

Os nós ficaram configurados com os seguintes endereços IP e portos (isto pode ser verificado consultado os ficheiros do *lightData* das *Super-probes)*:

Nó	PC	Informações do Nó	
		Endereço IP	Número de porto
Super-probe 0.1	PC 2	193.136.93.103	22368
Probe 0.1	PC 3	193.136.93.158	22368
Client 0.1	PC 1	10.0.0.73	21164
Super-probe 1.1			22368

Tabela 20 – Informações dos Nós

Após ter iniciado os pontos de prova e o cliente realizaram-se os testes.

6.1.1.1.1 Teste 6.1 – Configuração de um módulo de monitorização activa – ping

Neste caso o teste configurado foi o p*ing*, que corresponde à opção 1 do menu. Para isto efectuaram-se os seguintes passos:

- Escolheu-se o nó origem.
- Escolheu-se o nó destino.
- Definiu-se o número de repetições (*-c*), neste caso utilizou-se 5 repetições.
- Seguidamente podia-se escolher a opção de efectuar o teste imediatamente ou definir uma data para efectuar o mesmo, neste caso optou-se por efectuar o teste imediatamente.
- Após configurar o teste, podia-se optar por esperar a resposta do *ping* ou não. Caso a opção escolhida seja esperar pela resposta, após finalizado o teste é indicado se foi ou não realizado com sucesso, e também indica onde foi guardado o resultado, dando a opção de descarregar o resultado para o PC onde está activo o cliente. Caso

a opção escolhida seja não esperar pela resposta, continua-se na utilização normal do cliente sem nenhuma indicação de finalização do teste efectuado. Neste caso optou-se por esperar pela resposta.

Configuraram-se então testes para os diferentes nós (como mostra a tabela seguinte), de forma a testar a conectividade entre eles.

Nó origem	Nó destino
10.0.0.73	193.136.93.158
	193.136.93.103
193.136.93.103	193.136.93.158
	10.0.0.73
193.136.93.158	193.136.93.103
	10.0.0.73

Tabela 21 – Nós origem e destino dos testes configurados

Um exemplo de configuração de um destes testes encontra-se no Apêndice E.1.

6.1.1.1.2 Teste 6.2 – Configuração de um módulo de monitorização passiva – *TCPDump*

No caso da monitorização passiva o teste utilizado foi o *TCPDump*, este corresponde à opção 5. Configurou-se a execução deste teste em cada um dos nós de forma que os três iniciassem a captura simultaneamente, para isto foi necessário ter em conta a hora e data de cada computador, pois o teste é efectuado conforme aos valores de hora e data do PC onde está a correr o nó que executará o teste.

Durante a captura realizou-se uma chamada do telefone VoIP da VLAN 2 para o telefone VoIP da VLAN 1, para gerar mais tráfego.

Um exemplo de configuração deste teste encontra-se no Apêndice E.2. Este exemplo mostra como configurar o *TCPDump* na *super-probe* do grupo 1 de forma a executar o teste da seguinte maneira: *-c 1000 -i eth1 -w cap4_pc1.cap*

A opção *-c* é obrigatória, esta indica o número de pacotes a capturar. As restantes opções foram introduzidas para personalizar a captura:

- Como o computador tinha duas placas *Ethernet* foi necessário indicar através de qual seria feita a captura, utilizando o comando *-i eth1* (pois neste caso a captura efectuou-se através da *eth1*). Caso não seja dada nenhuma indicação a captura será realizada na opção por defeito do *TCPDump*, que corresponde à *eth0*.
- Utilizou-se também a opção *-w cap4_pc1.cap* para que a captura fosse realizada no formato «*.cap*» e assim poder ser visualizada no *Wireshark*, caso esta opção seja

omitida a captura é feita em formato «*.txt*». O nome indicado, *cap4_pc1,* é ignorado pelo DTMS-P2P pois ele atribui os nomes sempre seguindo o mesmo formato para facilitar a procura de resultados entre os nós.

6.1.1.1.3 Teste 6.3 – Configuração de outras opções

Neste caso executaram-se as seguintes acções, de forma a mostrar outras funcionalidades do DTMS-P2P:

- Listar os nós de todos os grupos (opção 13, *Get list of known nodes of all measurement groups*).
- Descarregar resultados obtidos em alguns testes (opção 9, *Results search*).
- Saber que tipos de testes são suportados por um determinado nó (opção 7, *Request list of supported monitoring modules*).
- Obter informações dos diferentes testes (opção 8, *Request monitoring module's help description*).
- Descarregar o *Light Data File* (opção 14, *Get light data*).
- Descarregar o *File List* (opção 16, *Get file list*).
- Obter informação de um nó (opção 11, *Resources Request*).

No Apêndice E.3 encontram-se exemplos de como utilizar estas opções.

6.1.1.2 Teste 7 – Configuração de um novo módulo de monitorização

O DTMS-P2P permite acrescentar novos módulos de monitorização, nesta secção mostra-se como proceder para configurar uma nova ferramenta de monitorização. O programa acrescentado ao DTMS-P2P foi o *oping*[12].

Antes de iniciar os nós adaptou-se o ficheiro que contém os módulos de monitorização suportados, neste caso *SupportedMonitoringModulesLinux.xml,* de forma a integrar este novo módulo de monitorização.

Preencheram-se os requisitos necessários, neste caso o *monitoringModule id*, *name*, *commandToGetHelpDescription* e *mustUse (*como mostra a figura seguinte):

- O *monitoringModule id* e o *name,* neste caso correspondem ao próprio nome da ferramenta *oping*.

[12] Utiliza ICMP para testar a acessibilidade entre *hosts,* suporta múltiplos *pings* em paralelo utilizando IPv4 ou IPv6.

- Para obter informação sobre o *oping* basta fazer na linha de comandos *oping*, é por isto que na opção *commandToGetHelpDescription* preencheu-se simplesmente com a palavra *oping*.
- No *mustUse* incluiu-se unicamente a opção *–c,* para limitar o número de repetições do *ping*.

```
- <monitoringModule id="oping">
    <name>oping</name>
    <commandToGetHelpDescription>oping</commandToGetHelpDescription>
    <listOfOptionsToSaveToFile />
  - <restrictions>
      <mustUse>-c</mustUse>
      <doNotUse />
    </restrictions>
  </monitoringModule>
```

Figura 95 – Secção de código acrescentado no *SupportedMonitoringModulesLinux.xml*

Após acrescentar a ferramenta ao ficheiro *SupportedMonitoringModulesLinux.xml*, configuraram-se alguns testes recorrendo a esta ferramenta. Para isto recorreu-se à mesma configuração da figura 92, mas neste caso os três computadores encontravam-se configurados da seguinte forma:

PC	Observações	Informações do computador		
		Endereço IP	Mascara de rede	*Gateway*
PC 1	Ligado ao HUB para medir tráfego do *CallManager*	10.0.0.73	255.0.0.0	10.0.0.1
PC 2	Ligado ao telefone VoIP da rede VoIP (10.0.0.96)	193.136.93.103	255.255.254.0	193.136.92.43
PC 3	Ligado ao telefone VoIP da rede do IT (193.136.93.158)	193.136.93.161	255.255.254.0	193.136.92.43

Tabela 22 – Configurações dos computadores

Adaptou-se o ficheiro *FileOfKnownNodes.xml* dos directórios *FilesOfKnownNodes* e *clientFilesOfKnownNodes* com os dados das *super-probes* mostrados na tabela seguinte, procedendo da mesma forma que foi indicado na figura 94 (secção 6.1.1.1) e seguidamente iniciaram-se os nós e o cliente da mesma forma como se fez no teste 6 (tabela 19).

Nó	PC	Informações do Nó	
		Endereço IP	Número de porto
Probe 0.1	PC 3	193.136.93.161	22368
Super-probe 0.1	PC 2	193.136.93.103	22368
Client 0.1			21164

Super-probe 1.1	PC 1	10.0.0.73	22368

Tabela 23 – Informações dos Nós

Como se observa na tabela anterior o cliente, neste caso, foi executado no PC2, e obtém-se o mesmo esquema da figura 92.

Para conferir se os três nós se encontravam correctamente ligados utilizou-se a opção 13 (*Get list of known nodes of all measurement groups*), pois esta mostra uma lista dos nós que pertencem a cada grupo.

Seguidamente verificou-se se o novo módulo tinha sido correctamente acrescentado ao DTMS-P2P, executando as opções 7 e 8, *Request list of supported monitoring modules* e *Request monitoring module's help description,* respectivamente.

Após confirmar que o *oping* ficou bem instalado, configuraram-se então os testes com esta ferramenta da seguinte forma:

- Escolheu-se a opção 6, a qual permite escolher outro módulo de monitorização passiva.
- Introduziu-se o nome do teste, neste caso *oping*.
- Escolheu-se o nó que vai executar o teste.
- Introduziu-se as opções de configuração do teste, neste caso:

  -c número_repetições endereços_ip_dos_destinos

- Após ter configurado o teste, indicou-se que o mesmo fosse efectuado imediatamente e sem esperar resposta.

Configurou-se o *oping* em ambas as *super-probes*, para cada uma delas realizar *ping* aos outros nós da rede com cinco repetições (*-c 5*). Um exemplo de configuração deste teste encontra-se no Apêndice E.4.

6.1.2 Resultados e conclusões

Com a realização dos testes no DTMS-P2P, verificou-se que esta ferramenta permite a execução de diversos módulos de monitorização activa e passiva, além disto permite executar um conjunto de opções que fornecem informações sobre os nós e os módulos de monitorização.

Inicialmente esta ferramenta exibe um menu dividido em três secções, a primeira secção mostra quais os testes de monitorização activa que esta ferramenta fornece, a segunda exibe uma listagem dos módulos de monitorização passiva e por último aparece

um conjunto de opções que permitem consultar diversas informações (figura 16, secção 3.3.4.2).

Além disto, verificou-se que após a execução dos testes, o DTMS-P2P guarda os resultados obtidos nas pastas de *results* dos respectivos nós. Utilizando para estes ficheiros nomes que seguem sempre o mesmo formato, de forma a facilitar a procura dos resultados numa rede constituída por diferentes *probes*. O formato utilizado para os nomes dos ficheiros é o seguinte:

GrupoDoCliente_IPCliente.PortoCliente_GrupoDoNó_IPDoNó.PortoDoNó_Tempo_ConfiguraçãoDoTeste.res

Onde o *GrupoDoCliente*, *IPCliente* e *PortoCliente* correspondem aos dados do cliente, grupo, endereço IP e número do porto, respectivamente. Seguidamente aparecem os dados referentes ao nó onde é executado o teste (*GrupoDoNó*, *IPDoNó* e *PortoDoNó*). Por último, aparece o instante em que iniciou o teste (*Tempo*) e a configuração utilizada para a execução do teste (*ConfiguraçãoDoTeste*). Por exemplo:

00000000000000000000000000000000_193.136.93.103.21164_00000000000000000000000000000000_193.136.93.103.22368_1209053271240_oping -c 5 193.136.93.161 10.0.0.73.res

Do nome do ficheiro do exemplo anterior é possível extrair as seguintes informações:

- O teste executado foi o *oping*, este é uma ferramenta que permite a execução de múltiplos *pings*. O *oping* não vem instalado por defeito no DTMS-P2P, nas secções 6.1.1.2 e 6.1.2.2 mostra-se como realizar a instalação e configuração de novas ferramentas de monitorização no DTMS-P2P. Neste caso o *ping* foi executado para o 193.136.93.161 e o 10.0.0.73, sendo repetido 5 vezes.
- O cliente é o 193.136.93.103:21164.
- O nó onde está a ser executado o teste é o 193.136.93.103:22368.
- Neste caso tanto o Cliente como o nó pertencem ao grupo 0.
- O teste foi iniciado às 16:07:51 do dia 24 de Abril de 2008 (*1209053271240*[13]).

Quando não é possível realizar um teste de monitorização, isto é, o teste falha, os resultados são guardados numa pasta chamada *error*, dentro do directório *results*. O nome dos ficheiros de erro seguem a mesma lógica do nome do ficheiro de resposta, mas neste caso utilizam a extensão «*.err*» e não a «*.res*».

[13] Este valor representa a quantidade de milissegundos desde o dia 1 de Janeiro de 1970, é neste dia que se inicia a Era UNIX.

Além disto verificaram-se outros aspectos importantes quanto à utilização desta ferramenta, os quais são apresentados nas seguintes secções junto com os resultados obtidos.

6.1.2.1 Teste 6 – Utilizando os módulos de monitorização do *Linux* instalados por defeito no DTMS-P2P

6.1.2.1.1 Teste 6.1 – Configuração de um módulo de monitorização activa – *ping*

Como já foi referido na secção 6.1.1.1.1 e como mostra o Apêndice E.1, estes testes foram configurados de forma a serem executados imediatamente e optou-se por esperar pela resposta.

Quando é escolhida a opção de esperar pela resposta, ao concluir a execução do teste, o DTMS-P2P devolve uma mensagem a indicar se este teste foi ou não executado com existo. No exemplo mostrado no Apêndice E.1 a mensagem devolvida nesta situação foi a seguinte:

```
Waiting for the response to the command request. Press any key to
continue.

The command was successfully processed and the results were saved to
00000000000000000000000000000000_10.0.0.73.21164_0000000000000000000000000
00000000_193.136.93.158.22368_1208513961074_ping 10.0.0.73 -c 5.res file.
Press any key to continue.
```

Figura 96 – Mensagem que indica se o teste foi ou não efectuado com êxito e onde ficaram guardados os resultados

A mensagem contida na figura anterior indica que o teste foi efectuado com êxito e que os resultados foram guardados num ficheiro cujo nome é:

00000000000000000000000000000000_10.0.0.73.21164_0000000000000000000000000000000000_193.136.93.158.22368_1208513961074_ping 10.0.0.73 -c 5.res

Devido às alterações efectuadas nas configurações de rede dos computadores (ver tabela 18, secção 6.1.1.1), o *ping* entre os diferentes pontos de prova foi realizado com êxito e os resultados foram guardados nas respectivas pastas *results*. Na tabela seguinte mostram-se os valores mínimos, médios e máximos obtidos no *ping*.

Nó	Teste	RTT [*ms*]		
		Mínimo	Média	Máximo
Super-probe 0.1 (193.136.93.103)	*ping 10.0.0.73 -c 5*	0.777	1.650	1.985
	ping 193.136.93.158 -c 5	0.410	0.459	0.496
Probe 0.1 (193.136.93.158)	*ping 10.0.0.73 -c 5*	0.949	1.452	2.071
	ping 193.136.93.103 -c 5	0.402	0.457	0.508

Super-probe 1.1 (10.0.0.73)	*ping 193.136.93.103 -c 5*	0.807	1.684	2.072
	ping 193.136.93.158 -c 5	0.917	1.646	2.199

Tabela 24 – Resultados dos *pings* realizado entre os diferentes nós

Com este teste, foi possível mostrar como configurar um teste de monitorização activa, e além disto, foi possível verificar as vantagens da utilização de uma ferramenta de medição distribuída, pois esta permite-nos configurar uma série de testes (neste caso, diversos *pings* em diversos computadores) através de um único computador.

6.1.2.1.2 Teste 6.2 – Configuração de um módulo de monitorização passiva – TCPDump

Como foi indicado na secção 6.1.1.1.2 os três pontos de medição iniciaram as capturas simultaneamente. Para isto é fundamental conhecer a configuração horária de cada um dos nós intervenientes, pois como já foi referido anteriormente, o teste é executado conforme a hora do nó que o executa e não conforme o Cliente.

Conhecendo a hora e data de cada computador configurou-se o teste indicando o instante de tempo em que deveria ser efectuado, introduzindo o ano, mês, dia, hora, minuto e segundo. A figura seguinte mostra como devem ser introduzidos os dados referentes ao instante de tempo de execução do teste, este exemplo foi extraído do Apêndice E.2.

```
Do you want the remote node to immediately process the command (y/n)
(Default y): n

Enter the time you want the command to be processed:
Introduce the year: 2008
Introduce the month: 04
Introduce the date: 18
Introduce the hour: 12
Introduce the minutes: 25
Introduce the seconds: 00
```

Figura 97 – Exemplo de como configurar um teste para ser executado em determinado instante de tempo

Após iniciada a captura realizou-se uma chamada desde o telefone VoIP da VLAN 2 para o telefone VoIP da VLAN 1, de forma a gerar tráfego.

Obtiveram-se três capturas, cada uma delas foi guardada na respectiva pasta *results*. Devido à disposição dos pontos de medição, com as capturas obtidas é possível medir:

- Através da *Super-probe* 1.1:
  - ✓ O tráfego do *CallManager*.
- Através da *Super-probe* 0.1:
  - ✓ O tráfego gerado pelo PC 2 (193.136.93.103).

- ✓ O tráfego que o telefone VoIP 10.0.0.96 envia para o PC 2, isto foi possível pois activou-se neste telefone a opção *span to pc port*.

- Através da *Probe* 0.1 :
  - ✓ O tráfego gerado pelo PC 3 (193.136.93.158).
  - ✓ O tráfego que o telefone VoIP 103.136.93.109 envia para o PC 3, da mesma forma que no caso anterior neste telefone activou-se a opção *span to pc port*.

Desta forma confirma-se, uma vez mais, a importância do sistema de medição distribuída, pois através de um único ponto, neste caso o Cliente, foi possível programar diversos pontos de prova para executarem a captura simultaneamente, e assim poder obter uma visão do tráfego que flui em distintos pontos da rede numa determinada situação.

6.1.2.1.3 Teste 6.3 – Configuração de outras opções

Opção 13 - *Get list of known nodes of all measurement groups*:

Ao executar a opção 13 obteve-se uma listagem das *probes* e *super-probes* de cada grupo, como mostra a figura seguinte, este exemplo foi extraído do Apêndice E.3.1.

```
Nodes in the Measurement Group 00000000000000000000000000000001:
          1 - Super-probe  10.0.0.73:22368

Nodes in the Measurement Group 00000000000000000000000000000000:
          1 - Super-probe  193.136.93.103:22368
          2 - Probe        193.136.93.158:22368
```

Figura 98 – Resultado da opção 13

O resultado obtido após a execução desta opção confirma a configuração dos nós conforme a figura 92 (secção 6.1.1). Pois existem dois grupos, o grupo 1 é formado unicamente por uma *super-probe* e o grupo 0 contém uma *super-probe* e uma *probe*.

A listagem dos nós mostra também qual o endereço IP e o porto utilizado por cada nó. É possível verificar que estas informações coincidem com as informações da tabela 18 (secção 6.1.1.1).

Opção 9 – *Results search*:

No caso da opção 9 (Apêndice E.3.2), ao indicar qual o grupo onde se deseja realizar a pesquisa e o critério de pesquisa, aparece uma listagem numerada dos ficheiros que contêm o critério de pesquisa, possibilitando o *download* dos ficheiros, para isto basta escolher o número correspondente ao ficheiro que se deseja descarregar.

A listagem pode apresentar réplicas, um exemplo disto encontra-se na figura seguinte, esta mostra uma secção da listagem obtida após a pesquisa dos ficheiros que continham o critério «*ping*» no grupo 0, recorrendo à opção 9 (Apêndice E.3.2). Verifica-se então que o ficheiro 1, neste caso, encontra-se guardado na *probe* (193.136.93.103:22368) e na *super-probe* (193.136.93.158:22368) do grupo 0.

```
Choose the file you want to download:

        Option  -  File Name    File Size (in bytes)    # Nodes
Node IP         Node Port       Node Type       Speed   FlagUploadSpeed
FlagHaveUploaded        FlagBusy    FlagPush

          1     -
00000000000000000000000000000000_10.0.0.73.21164_00000000000000000000000000000000_193.136.93.158.22368_1208513961074_ping 10.0.0.73 -c 5.res
5082

193.136.93.103      22368   super-probe     0       0       1       0
0

193.136.93.158      22368   probe   0       0       1       0       0
```

Figura 99 – Exemplo de uma secção da lista obtida como resultado da opção 9

No Apêndice E.3.2 verifica-se que ambos nós do grupo 0 contêm os ficheiros procurados, isto acontece pois por defeito os resultados são guardados nos nós que executam o teste e replicados em 10 nós (este valor é configurável, *-nr*). Estas réplicas são criadas de forma a garantir o acesso aos resultados, mesmo quando o nó que efectuou o teste já não se encontre disponível. As réplicas também ajudam a acelerar o processo de *download*, pois desta forma os resultados de um teste podem ser descarregados de diferentes fontes ao mesmo tempo.

Opção 7 – *Request list of supported monitoring modules*:

Os resultados obtidos na opção 7 foram iguais para os três nós, pois todos foram configurados com o mesmo ficheiro que contém os módulos de monitorização suportados. Neste caso o ficheiro utilizado foi o *SupportedMonitoringModulesLinux.xml*, pois os nós foram executados em computadores cujo sistema operativo era o *Linux*.

Em cada um dos nós obteve-se uma listagem dos módulos de monitorização que estes suportavam e quais as restrições de cada um deles. A figura seguinte mostra a listagem obtida, este exemplo foi extraído do Apêndice E.3.3.

```
        Monitoring Module                       Restrictions
OWAMP_ControlClient             <mustUse>senderPort 4181; receiverPort
22368; -P 21164-21174</mustUse> <doNotUse></doNotUse>
```

```
owping          <mustUse>senderPort 4181; receiverPort 22368; -P 21164-
21174</mustUse> <doNotUse></doNotUse>

TCPDump         <mustUse>-c</mustUse> <doNotUse>-F; -l; -m; -
r</doNotUse>

tracepath               <mustUse></mustUse> <doNotUse></doNotUse>
traceroute              <mustUse></mustUse> <doNotUse></doNotUse>

ping            <mustUse>-c</mustUse> <doNotUse>-t</doNotUse>
```

Figura 100 – Resultado da opção 7

Opção 8 – *Request monitoring module's help description*:

No caso da opção 8 solicitou-se informação sobre o *TCPDump* a partir da *Super-probe 1.1* (Apêndice E.3.4), e obteve-se uma pequena descrição do funcionamento desta ferramenta. A figura seguinte mostra a informação obtida através desta opção.

```
Help description received from node 10.0.0.73:22368 for the tcpdump
monitoring module:
tcpdump version 3.9.7
libpcap version 0.9.7
Usage: tcpdump [-aAdDeflLnNOpqRStuUvxX] [-c count] [ -C file_size ]
                [ -E algo:secret ] [ -F file ] [ -i interface ] [ -M
secret ]
                [ -r file ] [ -s snaplen ] [ -T type ] [ -w file ]
                [ -W filecount ] [ -y datalinktype ] [ -Z user ]
                [ expression ]
```

Figura 101 – Resultado da opção 8

Opção 14 e 16:

Nos casos das opções 14 (*Get light data* – consultar Apêndice E.3.5) e 16 (*Get file list* – consultar Apêndice E.3.6), descarregou-se para a pasta de *downloads* do cliente os ficheiros *light data* e *file list* dos nós indicados. Como é possível observar nas figuras seguintes, é apresentada uma mensagem que indica quando o *download* é finalizado e o nome do ficheiro que contém as informações solicitadas.

```
      The LightData.xml download completed in 00:00:00 seconds and 204
milliseconds at a rate of 3.541 Kb/sec.
The compiled light data file of the super-probe 10.0.0.73:22368 was
succesfully downloaded and it was stored at the downloads directory as
10.0.0.73.22368_LightData_1208515274287.xml.
```

Figura 102 – Exemplo da mensagem que indica que o *download* do *LightData.xml* foi bem sucedido (opção 14)

```
      The FileList.xml download completed in 00:00:00 seconds and 191
milliseconds at a rate of 6.42 Kb/sec.
The file list of the node 193.136.93.103:22368 was succesfully
downloaded and it was stored at the downloads directory as
193.136.93.103.22368_FileList.xml.
```

Figura 103 – Exemplo da mensagem que indica que o *download* do *FileList.xml* foi bem sucedido (opção 16)

Estes ficheiros permitem obter informações sobre os nós, no caso do *LightData.xml* obtém-se uma descrição das ligações, e o *FileList.xml* fornece uma listagem dos ficheiros guardados na pasta *results* do respectivo nó.

Opção 11 – *Resources request*:

Esta opção mostra os recursos de um determinado nó, por exemplo o número máximo e mínimo de ligações permitidas, o número de ligações existentes, entre outros. Neste caso pediu-se informação sobre a *Probe* (193.136.93.158:22368) do grupo 0, e o resultado obtido foi o seguinte (Apêndice E.3.7):

```
The Resources message with the requested information was received from
193.136.93.158:22368. Received information:

Maximum number of connections allowed, if the node is in super-probe
mode: 65535
Maximum number of connections to probes allowed, if the node is in
super-probe mode: 65535
Current number of connections to super-probes of its measurement group:
1
Current number of connections to probes: 0
Current number of connections to clients: 0
Current number of connections to super-probes of other measurement
groups: 0

Number of Resources Information received: 1

Resources Information 1
Flag Upload Speed: 0
Flag Download Speed: 0
Flag Push: 0
Authentication Mode: 7
Node Addresses:
IPVN: 4
Port: 22368
IP Address: /193.136.93.158
Free Memory: 62 Mbytes
Occupied Memory: 1 Mbytes
Free storage space: 49229 Mbytes
Average available bandwidth upstream: 0 kbps
Average available bandwidth downstream: 0 kbps
```

Figura 104 – Resultado da opção 11

6.1.2.2 Teste 7 – Configuração de um novo módulo de monitorização

Após ter configurado e iniciado os nós e o cliente como indicado na secção 6.1.1.2, executou-se a opção 13 (*Get list of known nodes of all measurement groups*), de forma a conferir que os nós tinham sido configurado correctamente segundo a figura 92 (secção 6.1.1). Obteve-se a informação contida na figura seguinte, a qual está coerente segundo o configurado na secção 6.1.1.2 (tabela 22).

```
Nodes in the Measurement Group 00000000000000000000000000000001:
        1 - Super-probe  10.0.0.73:22368

Nodes in the Measurement Group 00000000000000000000000000000000:
        1 - Super-probe  193.136.93.103:22368
        2 - Probe        193.136.93.161:22368
```

Figura 105 – Resultado da opção 13

Seguidamente confirmou-se através da opção 7 (*Request list of supported monitoring*) se os módulos de monitorização incluíam o *oping*. Verificou-se esta informação em cada um dos nós e como era de esperar, neste caso, os três nós apresentaram nesta lista os mesmos módulos de monitorização, pois todos eles foram configurados com o mesmo ficheiro (*SupportedMonitoringModulesLinux.xml*). A figura seguinte mostra o resultado obtido quando executada a opção 7 na *Super-probe* do grupo 1 (*Super-probe 1.1* – figura 92), e como se pode observar o *oping* faz parte desta listagem.

```
 The remote node 10.0.0.73:22368 supports the given monitoring modules:

    Monitoring Module                Restrictions

OWAMP_ControlClient                  <mustUse>senderPort 4181; receiverPort 22368; -P 21164-21174</mustUse> <doNotUse></doNotUse>

oping          <mustUse>-c</mustUse> <doNotUse></doNotUse>

owping                   <mustUse>senderPort 4181; receiverPort 22368; -P 21164-21174</mustUse> <doNotUse></doNotUse>

TCPDump        <mustUse>-c</mustUse> <doNotUse>-F; -l; -m; -r</doNotUse>

tracepath            <mustUse></mustUse> <doNotUse></doNotUse>

traceroute           <mustUse></mustUse> <doNotUse></doNotUse>

ping          <mustUse>-c</mustUse> <doNotUse>-t</doNotUse
```

Figura 106 – Resultado da opção 7

Utilizou-se a opção 8 (*Request monitoring module's help description*), para obter informação sobre este novo módulo de monitorização, *oping*. Assim como no caso da opção 7, o resultado obtido foi igual para todos os nós.

Na figura seguinte mostra-se o resultado obtido quando escolhida a opção *Request monitoring module's help description* na *Super-probe* do grupo 1.

```
Help description received from node 10.0.0.73:22368 for the oping monitoring module:

Usage: oping [-46] [-c count] [-i interval] host [host [host ...]]
```

Figura 107 – Resultado da opção 8

Assim como no caso dos módulos de monitorização previamente definidos no DTMS-P2P, os resultados obtidos nos dois testes efectuados recorrendo ao *oping* ficaram guardados nas respectivas pastas de *results*, e como se observa na seguinte figura, utilizam o mesmo formato nos nomes dos ficheiros.

Super-probe 0.1
00000000000000000000000000000000_193.136.93.103.21164_00000000000000000000000000000000_193.136.93.103.22368_1209053271240_oping -c 5 193.136.93.161 10.0.0.73.res
Super-probe 1.1
00000000000000000000000000000000_193.136.93.103.21164_00000000000000000000000000000001_10.0.0.73.22368_1209053578536_oping -c 5 193.136.93.161 193.136.93.103.res

Tabela 25 – Nomes dos ficheiros com resultados do *oping*

Os resultados obtidos em ambos os testes foram os seguintes:

Nó	Teste		rtt [ms]		
			min	avg	Max
Super-probe 0.1 (193.136.93.103)	*oping -c 193.136.93.161 10.0.0.73*	193.136.93.161	0.571	0.615	0.648
		10.0.0.73	0.880	1.349	1.956
Super-probe 1.1 (10.0.0.73)	*oping -c 193.136.93.161 193.136.93.103*	193.136.93.161	1.233	1.940	2.128
		193.136.93.103	0.894	1.194	2.336

Tabela 26 – Resultados obtidos no *oping*

Com este teste verifica-se que o DTMS-P2P é uma ferramenta flexível, pois esta permite a instalação de outros módulos de monitorização. Além disto, verificou-se também que a instalação de novas ferramentas não implica grandes dificuldades, pois basta inserir os dados da nova ferramenta no ficheiro que contém os módulos de monitorização suportados, *SupportedMonitoringModulesLinux.xml* no caso do *Linux* ou *SupportedMonitoringModules.xml* no caso do *Windows*.

7 CONCLUSÕES

7.1 Rede IP de investigação do IT

A rede IP de investigação do IT fornece os serviços de VoIP, videoconferência e emissão de vídeo. Nesta dissertação, como já foi referido anteriormente, efectuou-se uma medição e caracterização do tráfego gerado pelos serviços: VoIP e emissão de vídeo. Concluindo-se o seguinte:

Rede VoIP:

No caso do VoIP, a rede contém equipamentos da Cisco, nomeadamente, telefones VoIP e *CallManager*. Estes recorrem aos protocolos de sinalização SCCP e MGCP. O protocolo SCCP é utilizado pelos telefones VoIP para comunicar com o *CallManager*, enquanto que o MGCP é utilizado pelo *CallManager* para comunicar com o router que serve de interface entre o *CallManager* e a central telefónica. Além destes protocolos, os equipamentos utilizam o protocolo RTP para transportar dados.

Do estudo efectuado a esta rede, verificou-se que o tráfego que flui pela rede perante a ausência de chamadas é unicamente composto por pacotes de sinalização periódicos, tratando-se de 38*pps* aproximadamente, mas este valor vai depender da quantidade de telefones existentes na rede. Existem situações em que a quantidade de pacotes de sinalização aumenta, isto pode acontecer, por exemplo, no estabelecimento e na finalização de uma chamada. Quanto aos pacotes de dados, verificou-se que, estes só existem durante as chamadas, e aparecem em maior quantidade do que os pacotes de sinalização. Neste caso, como não está activo nenhum método de supressão de silêncio, é enviada sempre a mesma quantidade de pacotes, logo, cada fluxo de pacotes envia 50*pps* o que corresponde a 64*kbps*, isto é consequência do codec utilizado, pois é o G.711.

Verificou-se também que o comportamento do tráfego de dados e de sinalização nos diferentes tipos de chamadas (VLAN 1 – VLAN 1, VLAN 1 – VLAN 2 e VLAN 1 – CT) é similar. Em todos os casos existiu uma maior quantidade de pacotes de sinalização durante o estabelecimento e finalização da sessão, pois ao longo da chamada os únicos pacotes de sinalização que existem são os pacotes de índole periódica, além destes durante as chamadas existem também pacotes de dados. As diferenças entre os tipos de chamadas encontram-se basicamente nos protocolos utilizados, pois no caso da chamada ser efectuada entre telefones VoIP, utiliza-se o SCCP para sinalização e o RTP para dados, e quando a chamada é efectuada entre um telefone VoIP e um da central telefónica,

utilizam-se dois protocolos de sinalização diferentes: SCCP e MGCP, no caso dos dados utiliza também o RTP, mas neste caso este é acompanhado do RTCP.

Ao longo de uma sessão, considerando também o estabelecimento e finalização da mesma, a quantidade de pacotes de sinalização é muito inferior à quantidade de pacotes de dados. No caso da chamada ser efectuada entre dois telefones VoIP (VLAN 1 – VLAN 2 e VLAN 1 – VLAN 1) a sinalização representa aproximadamente 0,775% do total dos pacotes, no entanto nas chamadas entre um telefone VoIP e um da central telefónica (VLAN 1 – CT) os pacotes de sinalização representam 0,56%. Esta diferença nas percentagens de pacotes de sinalização em ambos os casos, deve-se ao facto do protocolo MGCP enviar menos mensagens de sinalização do que o SCCP.

Além disto verificou-se algumas diferenças nos valores do *jitter* dependendo do tipo de chamada. Nas chamadas entre telefones VoIP, os pacotes que vão desde o telefone VoIP da VLAN 2 para o telefone VoIP da VLAN 1 apresentaram valores de *jitter* superiores aos que vão no sentido oposto, e também são superiores aos valores apresentados pelos pacotes que são trocados pelos dois telefones VoIP da VLAN 1, isto pode ser consequência das configurações de QoS (*Quality of Service*), pois nesta rede o tráfego procedente da VLAN 1 tem prioridade sobre o restante tráfego. Além disto verificou-se que os pacotes que vão desde a central telefónica para o telefone VoIP da VLAN 1 apresentaram valores de *jitter* superiores a qualquer um dos casos anteriores, esta maior variação no atraso dos pacotes deste fluxo pode ser consequência da conversão dos formatos, pois esta pode ser mais demorada num sentido do que no outro.

Emissão de vídeo:

Para poder fornecer o serviço de videoconferência e emissão de vídeo, a rede IP de investigação do IT utiliza equipamentos da *Tandberg*, neste caso, a rede é composta por um equipamento de vídeo e um *Content Server*. O protocolo de sinalização utilizado por estes equipamentos é o H.323, enquanto que o protocolo de dados é o RTP.

O *Content Server*, além de proporcionar a possibilidade de efectuar videoconferência, também permite a emissão de vídeo, pois este equipamento disponibiliza vídeos em tempo-real ou vídeos previamente gravados, através da Internet (http://www.av.it.pt/farol/map-tele/). Para visualizar estes vídeos, os equipamentos terminais (computadores que se ligam ao *Content Server* através da *Internet*) recorrem ao protocolo MMS ou ao RTSP.

Através dos testes efectuados, verificou-se que o tráfego gerado pelo serviço de emissão de vídeo pode ser dividido em dois:

- Tráfego entre o *Content Server* e o equipamento vídeo:

  Neste caso, verificou-se que além do tráfego que vai desde o equipamento de vídeo para o *Content Server*, existe também tráfego no sentido oposto, pois o Content Server envia para o equipamento de vídeo uma imagem de forma a indicar que a sessão se encontra activa. Este tráfego só representa 20% do total.

  Além disto verificou-se que o áudio e o vídeo viajam em pacotes separados, utilizando portos diferentes. Concluiu-se também que estes sistemas não têm activo nenhum método de supressão de silêncio, pois o tráfego de áudio é sempre constante, e são enviados aproximadamente 64 *kbps* (o codec utilizado pelo áudio é o G.722 a 64 *kbps*). No caso do vídeo o codec utilizado é o H.263; para este tipo de tráfego a largura de banda consumida é muito superior, chegando a atingir os 1095,92*kbps*.

- Tráfego entre o *Content Server* e os equipamentos terminais:

  Tanto no caso do *download* de um vídeo em directo, como no caso de um vídeo previamente gravado, verificou-se que o tráfego transmitido pelo *Content Server* é muito superior ao tráfego redebido pelo mesmo. Pois o tráfego que entra no *Content Server* é unicamente de controlo, enquanto que o tráfego que sai, além de conter pacotes de controlo também transporta os dados (vídeo e áudio).

  Além disto, através dos testes efectuados, verificou-se que os computadores que se ligam ao *Content Server* para assistir ao vídeo, podem utiliziar o protocolo MMS ou o RTSP.

7.2 DTMS-P2P

O DTMS-P2P permite criar um sistema de monitorização, através da configuração de um conjunto de nós (*probe* e *super-probe*), os quais são associados em grupos. É através destes nós e de um cliente (*Client*) que se torna possível a monitorização de uma rede de forma distribuída.

Através da configuração dos testes de monitorização activa e passiva no DTMS-P2P, verificou-se a importância da utilização de uma ferramenta de medição distribuída. Pois através de um único computador, neste caso o cliente, é possível configurar um conjunto de testes nos outros computadores, onde correm os nós, sem necessidade de deslocação

física computador a computador, o qual se torna muito vantajoso sobretudo em redes de grandes dimensões.

Este sistema de monitorização distribuída, além de permitir a configuração de testes de forma remota, permite também consultar resultados dos testes efectuados em qualquer um dos nós. Isto é possível porque o DTMS-P2P guarda os resultados dos testes em ficheiros de texto, seguindo sempre o mesmo formato na atribuição do nome. Os ficheiros que contêm resultados ficam guardados nos respectivos nós numa pasta chamada results, além disto, estes ficheiros são replicados nos outros nós do grupo, de forma a garantir o acesso aos resultados, mesmo quando o nó que efectuou o teste já não se encontre disponível. As réplicas também ajudam a acelerar o processo de *download*, pois desta forma os resultados de um teste podem ser descarregados de diferentes fontes ao mesmo tempo.

Também se verificou que o DTMS-P2P permite adicionar novos módulos de monitorização, para isto basta actualizar o ficheiro que contém os módulos de monitorização suportados. A instalação de novas ferramentas não implica dificuldade, isto faz do DTMS-P2P uma ferramenta flexível.

O DTMS-P2P também oferece um conjunto de opções que permitem:

- Obter informações sobre o sistema de monitorização;
- Consultar quais os módulos de monitorização suportados pelo sistema de monitorização;
- Solicitar informação sobre algum módulo de monitorização em específico;
- Entre outras.

De forma geral, pode-se concluir que esta ferramenta é muito útil quando se deseja efectuar a monitorização de uma rede em diferentes pontos. A única dificuldade encontrada na utilização do DTMS-P2P foi o facto da versão gráfica não funcionar correctamente no *Linux*, pois esta bloqueava, suspeitando-se de algum problema ou incompatibilidade com o Java do *Linux*.

SIGLAS E ACRÓNIMOS

ASF	*Advanced Systems Format*
BootP	*Bootstrap Protocol*
Bpm	*Bytes por minute*
bps	*Bits por Segundo*
CDF	*Comulative Distribution Function*
CKN	*Cache of Known Nodes*
CSV	*Comma Separated Values summary*
DMZ	*DeMilitarized Zone*
DNS	*Domain Name System*
DSP	*Digital Signal Processor*
DTMS-P2P	*Distributed Traffic Measurement System with a Peer-to-Peer Architecture*
FCCN	*Fundação para a Computação Científica Nacional*
FKN	*File of Known Nodes*
GUI	*Graphical User Interface*
HD	*High-Definition*
HTTP	*Hypertext Transfer Protocol*
INTOP	*Interactive NTOP*
IP	*Internet Protocol*
ISSO	*International Organization for Standardization*
IT	*Instituto de Telecomunicações*
ITU	*International Telecommunication Union*
ITU-T	*ITU Telecommunication Standardization Sector*
MC	*Multipoint Controller*
MCU	*Multipoint Control Unit*
MGCP	*Media Gateway Control Protocol*
MMS	*Microsoft Media Server*
MP	*Multipoint Processor*
NTOP	*Network Traffic Probe*
NTP	*Network Time Protocol*
P2P	*Peer-to-Peer*
POST	*Plain Old Telephone Service*
ppm	*Pacotes por minute*
pps	*Pacotes por Segundo*
RAS	*Registration, Admission and Status*
RRD	*Round-Robin Database*
RTCP	*Real-Time Transport Control Protocol*
RTP	*Real-Time Transport Protocol*
RTT	*Round Trip Time*
SCCP	*Skinny Client Control Protocol*
TCP	*Transmission Control Protocol*
TCS	*TANDBERG Content Server*
TFTP	*Trivial File Transfer Protocol*
TSTAT	*TCP STatistics and Analysis Tool*

UDP	*User Datagram Protocol*
URI	*Uniform Resource Identifier*
URL	*Uniform Resource Locator*
VLAN	*Virtual Local Área Networks*
VoIP	*Voice over Internet Protocol*

REFERÊNCIAS

[AF03] Andreasen, F. & Foster, B. (2003). ***Media Gateway Control Protocol (MGCP) Version 1.0.*** Acedido em 10 de Junho de 2008, em http://www.ietf.org/rfc/rfc3435.txt

[AR00] Almeida, J. & Ramlie, Y. (2000). ***NTOP – Network TOP*: An overview**. Enschede: University of Twente. Acedido em 23 de Janeiro de 2008, em http://www.ntop.org/ntop-overview.pdf

[Cisco0?a] Cisco Systems, Inc (200-?). ***Cisco Unified IP Phone Administration Guide for Cisco Unified CallManager 5.1 (SCCP):* Cisco Unified IP Phone 7961G/7961G-GE and 7941G/7941G-GE**. Acedido em 26 de Março de 2008, em http://www.cisco.com/en/US/docs/voice_ip_comm/cuipph/7961g_7961g-ge_7941g_7941g-ge/5_1/english/administration_sccp/guide/61SCCP51.pdf

[Cisco0?b] Cisco Systems, Inc (200-?). ***SCCP Call Flows***. Acedido em 24 de Março de 2008, em http://www.cisco.com/univercd/cc/td/doc/product/voice/ata/ataadmn/sccp/sccpaaph.pdf

[Cisco05] Cisco Systems, Inc (2005). ***MGCP 1.0 and Additional MGCP Packages***. Acedido em 12 de Junho de 2008, em http://www.cisco.com/en/US/docs/voice_ip_comm/pgw/9/feature/module/9.5_1_/FMSegov.pdf

[Cisco06a] Cisco Systems, Inc (2006). ***Cisco Unified CallManager 5.1 TCP and UDP port usage***. Acedido em 30 de Julho de 2008, em http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/port/5_1/51plrev1.pdf

[Cisco06b] Cisco Systems, Inc (2006). ***Cisco Unified CallManager Features and Services Guide:* Release 5.0(4)**. Acedido em 26 de Março de 2008, em http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/admin/5_0_4/ccmfeat/ccmfeat.html

[Cisco06c] Cisco Systems, Inc (2006). ***Understanding MGCP Interactions with Cisco CallManager***. Acedido em 12 de Junho de 2008, em http://www.cisco.com/en/US/tech/tk1077/technologies_tech_note09186a00801da84e.shtml

[Cisco08a] Cisco Systems, Inc (1992-2008). ***Cisco Unified IP Phone 7941G***. Acedido em 25 de Março de 2008, em http://www.cisco.com/en/US/products/ps6513/prod_view_selector.html

[Cisco08b] Cisco Systems, Inc (1992-2008). ***Cisco Unified IP Phone 7961G***. Acedido em 25 de Março de 2008, em http://www.cisco.com/en/US/products/ps5945/prod_view_selector.html

[FCCN04] FCCN (2004). ***MCU***. Acedido em 31 de Março de 2008, em *http://www.fccn.pt/index.php?module=pagemaster&PAGE_user_op=view_page&PAGE_id=432&947cda2253a1dc58fe23dc95ac31cbed=00a6557d16e6a323d0af678fb43a11b6*

[GC97] Guimarães, R. & Cabral, J. (1997). ***Estatística*** (2ª ed). Lisboa: McGraw-Hill.

[HR03] Huidobro, J. & Roldán, D. (2003). ***Integración de voz y datos:*** **Call Centers: Tecnología y aplicaciones**. Madrid: McGraw-Hill.

[ITU-T06] ITU-T (2006). ***Serie H: Sistemas audiovisuales y multimedia:*** **Sistemas de comunicación multimédia baseados em paquetes**. Recomendación UIT-T H.323. Acedido em 20 de Julho de 2008, em http://www.itu.int/rec/T-REC-H.323-200606-I/en

[Jav0?] Javvin Technologies (200-?). ***Cisco SCCP: Skinny Client Control Protocol***. Acedido em 27 de Março de 2008, em http://www.javvin.com/protocolSCCP.html

[Microsoft08] Microsoft Corporation (2008). ***[MS-MMSP]*****: Microsoft Media Server (MMS) Protocol Specification**. Acedido em 30 de Julho de 2008, em http://download.microsoft.com/download/9/5/E/95EF66AF-9026-4BB0-A41D-A4F81802D92C/%5BMS-MMSP%5D.pdf

[Nog08] Nogueira, A. (?). ***Medições de Tráfego***. Acedido em 05 de Julho de 2008, em http://www.av.it.pt/nogueira/investigacao.html

[NTOP07] NTOP (2007). ***Download NTOP***. Acedido em 23 de Janeiro de 2008, em http://www.ntop.org/download.html.

[Sa07] Sá, R. (2007). ***Sistemas e Redes de Telecomunicações***. Lisboa: FCA.

[SC03] Schulzrinne, H & Casner, S. (2003). ***RTP Profile for Audio and Video Conferences with Minimal Control***. RFC 3551. Acedido em 26 de Março de 2008, em http://www.ietf.org/rfc/rfc3551.txt

[Sch96] Schulzrinne, H. (1996). ***RTP Profile for Audio and Video Conferences with Minimal Control***. RFC 1890. Acedido em 26 de Março de 2008, em http://www.ietf.org/rfc/rfc1890.txt

[SR0?] Santos, A. & Rolhas, R. (200-?). ***Plataforma FAROL*****: Relatório Intermédio 1**. Aveiro: Instituto de Telecomunicações Aveiro

[SRL98] Schulzrinne, H., Rao, A. & Lanphier, R. (1998). ***Real Time Streaming Protocol (RTSP)***. RFC 2326. Acedido em 12 de Julho de 2008, em http://www.ietf.org/rfc/rfc2326.txt

[Tandberg07a] TANDBERG (2007). ***TANDBERG Content Server:*** **Administrator Guide**. Acedido em 31 de Março de 2008, em http://www.tandberg.com/collateral/documentation/User_Manuals/TANDBERG%20Content%20Server%20Administrators%20Guide%20(S3).pdf

[Tandberg07b] TANDBERG (2007). ***TANDBERG Edge 95/85/75 MXP***. Acedido em 31 de Março de 2008, em http://www.tandberg.com/collateral/product_brochures/TANDBERG_95_85_75MXP.pdf

[Tandberg07c] TANDBERG (2007). ***TANDBERG MXP:*** **Administrator's Guide**. Acedido em 31 de Março de 2008, em http://www.tandberg.com/collateral/documentation/User_Manuals/TANDBERG%20MXP%20Administrators%20Guide%20(F6).pdf

[TNG08] Telecommunication Networks Group (2008). ***TSTAT :*** **Howto**. Acedido em 30 de Janeiro de 2008, em http://tstat.tlc.polito.it/HOWTO.shtml.

[Vei07] Veiga, H. (2007). ***DTMS-P2P:*** **User's Manual (version 1.0)**. Aveiro: Universidade de Aveiro / Instituto de Telecomunicações Aveiro.

BIBLIOGRAFIA

Andreasen, F. & Foster, B. (2003). ***Media Gateway Control Protocol (MGCP):*** **Version 1.0**. RFC 3435. Acedido em 10 de Junho de 2008, em http://www.ietf.org/rfc/rfc3435.txt

Carlson, A., Crilly, P. & Rutledge, J. (2002). ***Communication Systems:*** **An Introduction to Signals and Noise in Electrical Communication** (4ª ed.). Boston: McGraw-Hill.

FCCN (2004). ***H.323***. Acedido em 24 de Outubro de 2007, em http://www.fccn.pt/index.php?module=pagemaster&PAGE_user_op=view_page&PAGE_id=405&947cda2253a1dc58fe23dc95ac31cbed=3bd28d08a629e296aa06127db96698d0

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., *et al* (1999). ***Hypertext Transfer Protocol – HTTP/1.1***. RFC 2616. Acedido em 29 de Março de 2008, em http://www.ietf.org/rfc/rfc2616.txt

Minoli, D. & Minoli, E. (1998). ***Delivering Voive over IP Networks***. New York: Wiley Computer Publishing.

NTOP (2005). ***Manpage of NTOP***. Acedido em 23 de Janeiro de 2008, em http://www.ntop.org/ntop-man.html

Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V. (1996). ***RTP: A Transport Protocol for Real-Time Applications.*** RFC 1889. Acedido em 26 de Março de 2008, em http://www.ietf.org/rfc/rfc1889.txt

Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V. (2003). ***RTP: A Transport Protocol for Real-Time Applications.*** RFC 3550. Acedido em 26 de Março de 2008, em http://www.ietf.org/rfc/rfc3550.txt

APÊNDICES

Apêndice A - *Scripts*

A.1 - *Script* para eliminar pacotes de dados VoIP duplicados (*eliminarDupVoIP.sh*)

Este ficheiro elimina os pacotes de dados duplicados, visto que o *Wireshark* não permite eliminar estes pacotes mesmo quando eram de fácil identificação através do *sequence number*. Como é possível observar na seguinte figura, os *sequence numbers* (*seq*) dos pacotes repetidos são iguais.

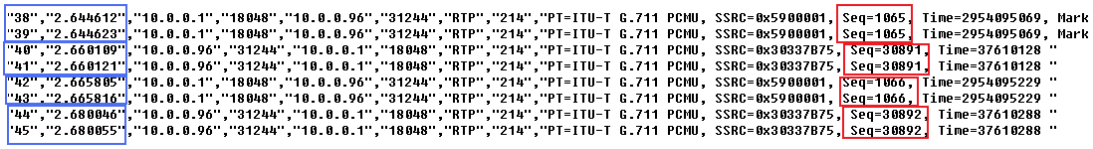

Figura 108 – Pacotes de dados duplicados

Neste caso ignoraram-se, para a comparação, os dois primeiros campos, isto é o número de pacote e o tempo (os quais encontram-se num quadro azul na figura anterior), pois estes eram os únicos campos que diferiam entre os pacotes repetidos, os restantes elementos eram idênticos.

Seguidamente apresenta-se o código utilizado. Como é possível constatar, recorreu-se ao comando «*uniq –f **n***», pois este permite eliminar linhas iguais ignorando os ***n*** primeiros campos.

```
#!/bin/bash

####
# Elimina os pacotes duplicados
###

echo "Nome do ficheiro:"
read fich

# Cria um ficheiro (fichn) com todos os pacotes UDP e RTP a partir do ficheiro original (fich)
fichn="UDP_RTP.txt"
sed -n '/RTP\|UDP/p' $fich > $fichn

# Troca "," por espaços, desta forma cada campo fica separado por um espaços
nomeficheiro="ficheiroteste.txt"
sed 's/,/ /g' $fichn > $nomeficheiro

# Apaga as linhas repetidas ignorando os n primeiros campos "uniq -f n ficheiro_origem ficheiro_destino"
fich2="semDup1.txt"
uniq -f 2 $nomeficheiro $fich2

# Volta a colocar as  ","
fich3="semDup.txt"
```

```
sed 's/" "/","/g'  $fich2 > $fich3

# Apaga tudo o que é UDP e RTP
fich4="sem_UDP_RTP.txt"
fich5="semDup2.txt"
sed '/RTP\|UDP/D' $fich > $fich4

# Junta os ficheiros, neste caso o que não tem UDP e RTP com o ficheiro que contem os pacotes UDP e RTP sem duplicados
cat sem_UDP_RTP.txt semDup.txt > $fich5
cat $fich4 $fich3 > $fich5

fich6="semDup_semAspas_$fich.txt"
sed 's/"//g' $fich5 > $fich6

fich7="semDup_ordenado_$fich.txt"
sort -n -t "," -k 1 -o $fich7 $fich6

rm $fichn
rm $fich2
rm $fich3
rm $fich4
rm $fich5
rm $fich6
rm $nomeficheiro
echo "****"

# O ficheiro semDup_ordenado_$fich.txt contém a captura completa mas sem os pacotes duplicados

exit

# MIEET
# Lidia Elena da Costa dos Reis
# Medição e Caracterização de Tráfego Tempo-Real em Redes IP
```

Este *script* aplica-se aos ficheiros *csv* que contêm as capturas completas, previamente filtradas pelo *Wireshark,* isto é, só deve apresentar réplicas de pacotes de dados, por exemplo se este ficheiro chama-se **voip_csv**, o resultado deste *script* é um ficheiro cujo nome é: **semDup_ordenado_voip_csv.txt.**

A.2 - *Script* para separar os pacotes de dados dos pacotes de sinalização, segundo os protocolos (*encontraProtocolo.sh*)

Este *script* aplica-se um ficheiro que contém os pacotes capturados (exemplo, o ficheiro de texto que contém a captura sem pacotes repetidos) e cria dois directórios, um com os ficheiros que contêm pacotes de sinalização (neste caso SCCP e MGCP) e outro com ficheiros que contêm pacotes de dados (neste caso RTP).

```
#!/bin/bash

####
# Procura no Ficheiro nomefich os pacotes dos protocolos SKINNY, TCP, MGCP e RTP, e cria um ficheiro para cada tipo de protocolo
```

```
####

echo "Nome do ficheiro:"
read nomefich

# Protocolos
# Dados
protocolDados="RTP\|UDP"
# Sinalização
protocolSCCP="SKINNY\|TCP"                  # SCCP
protocolMGCP="MGCP"                         # MGCP
protocolSinalizacao="SKINNY\|TCP\|MGCP"     # Todos sinalização

# Nomes dos ficheiros
nomefich1="linhas_Dados.txt"
nomefich2="linhas_SCCP.txt"
nomefich3="linhas_MGCP.txt"
nomefich4="linhas_Sinalizacao.txt"

# Cria um ficheiro com as linhas (pacotes) que contêm o nome de determinado protocolo
# Se quiser saber o número da linha basta acrescentar -n --> grep -n "\<$protocol1\>" $nomefich >
$nomefich2
grep  "\<$protocolDados\>" $nomefich > $nomefich1
grep  "\<$protocolSCCP\>" $nomefich > $nomefich2
grep  "\<$protocolMGCP\>" $nomefich > $nomefich3
grep  "\<$protocolSinalizacao\>" $nomefich > $nomefich4

# Directórios
direc1="Dados"
mkdir $direc1
chmod 777 $direc1
cp $nomefich1 $direc1
direc2="Sinalizacao"
mkdir $direc2
chmod 777 $direc2
cp $nomefich2 $nomefich3 $nomefich4 $direc2

rm $nomefich1
rm $nomefich2
rm $nomefich3
rm $nomefich4

exit

# MIEET
# Lidia Elena da Costa dos Reis
# Medição e Caracterização de Tráfego Tempo-Real em Redes IP
```

A.3 - *Script* utilizado para editar a captura (*editcap.sh*)

Este *script* aplica-se um ficheiro que contém os pacotes capturados (exemplo, o ficheiro de texto que contém a captura sem pacotes repetidos) e cria um directório com os seguintes ficheiros:

Nome do ficheiro	Conteúdo do ficheiro
numero.txt	Números dos pacotes.
time.txt	Tempo de cada pacote.
srcIP.txt	Endereço IP origem de cada pacote.
srcPort.txt	Número de porto origem dos pacotes.
dstIP.txt	Endereço IP destino dos pacotes.
dstPort.txt	Número de porto destino de cada pacote.
protocol.txt	Protocolo utilizado por cada pacote
pktLength.txt	Tamanho dos pacotes.
Info.txt	Informação relativa a cada pacote.

Tabela 27 – Ficheiros gerados pelo *editcap.sh*

O código correspondente a este *script* é:

```
#!/bin/bash

####
# Cria um ficheiro com cada coluna da captura, isto é um ficheiro por elemento da captura
####

echo "Nome do ficheiro:"
read nomefich

# Elimina as aspas
nomeficheiro="sem_aspas.txt" # elimina as aspas
sed 's/"//g' $nomefich > $nomeficheiro

# Número do pacote
file1="numero.txt"
cat $nomeficheiro | cut -f 1 -d "," > $file1
echo $file1

# Time
file2="time.txt"
cat $nomeficheiro | cut -f 2 -d "," > $file2
echo $file2

# Source IP
file3="srcIP.txt"
cat $nomeficheiro | cut -f 3 -d "," > $file3
echo $file3

# Source Port
file4="srcPort.txt"
cat $nomeficheiro | cut -f 4 -d "," > $file4
echo $file4

# Destination IP
file5="dstIP.txt"
cat $nomeficheiro | cut -f 5 -d "," > $file5
echo $file5

# Destination Port
file6="dstPort.txt"
cat $nomeficheiro | cut -f 6 -d "," > $file6
```

```
echo $file6

# Protocol
file7="protocol.txt"
cat $nomeficheiro | cut -f 7 -d "," > $file7
echo $file7

# Pkt Length
file8="pktLength.txt"
cat $nomeficheiro | cut -f 8 -d "," > $file8
echo $file8

# Info
file9="Info.txt"
cat $nomeficheiro | cut -f 9 -d "," > $file9
echo $file9

directorio="cap_$nomefich"
mkdir $directorio
chmod 777 $directorio
cp $file1 $file2 $file3 $file4 $file5 $file6 $file7 $file8 $file9 $directorio

rm $file1
rm $file2
rm $file3
rm $file4
rm $file5
rm $file6
rm $file7
rm $file8
rm $file9
rm $nomeficheiro

echo "***"
exit

# MIEET
# Lidia Elena da Costa dos Reis
# Medição e Caracterização de Tráfego Tempo-Real em Redes IP
```

A.4 - *Script* utilizado para contabilizar os pacotes transmitidos em X segundos (*contadorPacVoIP.sh*)

Este *scritp* utiliza o ficheiro *time.txt* para calcular o número de pacotes transmitidos em *X* segundos (este intervalo é definido pelo utilizador através de *tempoi*), para isto calcula o próximo valor do tempo e compara com cada linha do ficheiro *time.txt*, por exemplo, se deseja saber quantos pacotes foram enviados durante um minuto, inicialmente conta quantas linhas há até chegar ao valor 60 (visto que este ficheiro se encontra em segundos), calcula o próximo valor de tempo, neste caso 120 e conta entre 60 e 120 segundos, e assim até atingir o valor final.

Como resultado obtêm-se dois ficheiros, um que contêm simplesmente o número de pacotes transmitidos de *X* em *X* segundos (*PacINTtempo.txt*) e outro mais detalhado

(*pacotesPorIntTempo.txt*), neste último indica linha a linha o tempo e o número de pacotes transmitidos, por exemplo: *até 60 seg existem 38 pacotes*.

Seguidamente encontra-se o código deste *script*:

```
#!/bin/bash

####
# Conta o número de pacotes de x em x segundos
####

# Nome do ficheiro
echo "Nome do ficheiro:"
read fich3

# Intervalo de tempo
echo "Intervalo de tempo em segundos:"
read tempoi

fich="fichtemp.txt"
fich2="temp1.txt"
fich1="temp2.txt"

# Formatação do ficheiro de forma a poder manipular o mesmo
cat $fich3 | cut -f 1 -d "." > $fich2
sed 's/Time//g' $fich2 > $fich1
sed '/^$/d' $fich1 > $fich

# Número de linhas
wc -l $fich > numlinhastxt.txt
numlinhas=`cat numlinhastxt.txt | cut -f 1 -d " "`
rm numlinhastxt.txt

# Iniciar contador de linhas
linha=0

# Tempo inicial e final
primeiroval=`sed -n '1 p' $fich`
ultimoval=`sed -n '$ p' $fich`

resPac="pacotesPorIntTempo.txt"
tempo=$tempoi              # Tempo inicial
resultado="PacINTtempo.txt"
rm $resultado

echo "Tempo inicial = $primeiroval seg " > $resPac
echo "Tempo final = $ultimoval seg " >> $resPac
echo "Intervalos de $tempo seg" >> $resPac
echo "****************************************" >> $resPac

# Cálculo do m e do primeiro valor de tempo comparando com o primeiro valor obtido para não perder informação
m=1                        # Multiplicador
cont=0
primeiroval=`sed -n '1 p' $fich`
rm $resPac
if [ $primeiroval -ge $tempoi ]
```

```
then
        tempo=$tempoi
        while [ $primeiroval -ge $tempo ]
        do
                echo "até $tempo seg existem $cont pacotes" >> $resPac       # Imprime num ficheiro
                echo $cont >> $resultado
                let m++ #incrementa m
                tempo=$[$tempoi*$m]
        done
elif [ $primeiroval -lt $tempoi ]
then
        tempo=$tempoi
        m=1
fi

cat $fich | while read i
                do
                        let linha++   #incrementa linha
                        echo $linha
                        if [ "$i" -lt $tempo ]; then  #menor
                          let cont++
                      if [ $linha -eq $numlinhas ]; then
                                echo "até $tempo seg existem $cont pacotes" >> $resPac       # Imprime
num ficheiro
                                echo $cont >> $resultado
                          fi
                        elif [ "$i" -ge $tempo ]; then #maior igual
                                echo "até$tempo seg existem $cont pacotes" >> $resPac
                                echo $cont >> $resultado

                      while [ $i -ge $tempo ]
                                do
                                        let m++
                                tempo=$[$tempoi*$m]
                                        if [ $i -ge $tempo  ]
                                        then
                                                cont=0
                                                echo "até $tempo seg existem $cont pacotes" >>
$resPac  # Imprime num ficheiro
                                                echo $cont >> $resultado
                                        else
                                                cont=1
                                        fi
                                done
                                # para não perder o ultimo valor
                                if [ $linha -eq $numlinhas ]; then
                                        # Imprime num ficheiro
                                        echo "até $tempo seg existem $cont pacotes" >> $resPac
                                        echo $cont >> $resultado
                                fi
                      fi
                done

rm $fich
rm $fich1
rm $fich2

echo "***"
```

```
exit

# MIEET
# Lidia Elena da Costa dos Reis
# Medição e Caracterização de Tráfego Tempo-Real em Redes IP
```

A.5 - *Script* utilizado para contabilizar os bytes transmitidos em X segundos (*contadorBytesVoip.sh*)

Este *script* olha para o ficheiro que tem o número de pacotes transmitidos em *X* segundos (*PacINTtempo.txt*), este valor corresponde ao número de linhas a somar no ficheiro que contém os tamanhos dos pacotes (*pktLenght.txt*), por exemplo, se em determinado intervalo de tempo foram transmitidos 10 pacotes, são somados os tamanhos dos 10 pacotes que correspondem a esse intervalo de tempo.

O resultado deste *script* é um ficheiro que contém o número de bytes transmitidos por intervalo de tempo, cujo nome é *bytesPorIntTempo.txt*.

Seguidamente encontra-se o código correspondente a este *script*:

```
#!/bin/bash

####
# Calcula o número de bytes transmitidos em determinado intervalo de tempo
####

# Ficheiro que contém o número de pacotes transmitidos de x em x segundos
fichPAC="PacINTtempo.txt"

# Ficheiro que contém os tamanhos dos pacotes
fich2="pktLength.txt"

# Intervalo de tempo
echo "Intervalo de tempo em segundos:"
read tempo

fich="fichtemp.txt"
fich1="temp2.txt"

# Formatação do ficheiro de forma a poder manipular o mesmo
sed 's/Pkt Length//g' $fich2 > $fich1
sed '/^$/d' $fich1 > $fich

# Número de linhas
wc -l $fich > numlinhastxt.txt
numlinhas=`cat numlinhastxt.txt | cut -f 1 -d " "`
rm numlinhastxt.txt
linha=0                    # Iniciar contador de linhas do fichPAC

resBytes="bytesPorIntTempo.txt"

cont=0
j=0
```

```
fichbytes=(`cat $fich | while read ii
do
echo $ii
done;`)

echo "Bytes por $tempo seg" > $resBytes
cat $fichPAC | while read i;                    # Procura linha-a-linha no ficheiro que contém o número de pacotes por intervalo de tempo
                do
                        let linha++             # Incrementa linha
                        echo "Linha $linha"

                        linhapacotes=$[$i+$cont] # O número de pacotes por intervalo de tempo corresponde ao número de linhas a ler no fich ($i)
                        j=$[1+$cont]                    # Inicia o contador de linhas do fich
                        val=0                           # Valor inicial

                        # Se não há pacotes nesse intervalo de tempo, não calcula o tamanho, pois este é zero
                        if [ $i -eq 0 ]
                        then
                                echo "$val" >> $resBytes
                                echo $val
                        fi

                        while   [ $j -le $linhapacotes ]
                        do
                                n=$[$j-1]               # Posição de leitura do array
                                val=$[$val+${fichbytes[$n]}]
                                # Se j é igual a linhapacotes, isto é, se chegou ao número x de pacotes transmitidos nesse intervalo de tempo, para de somar e imprime o valor
                                if [ $j -eq $linhapacotes ]
                                then
                                        echo "$val" >> $resBytes
                                        echo $val
                                fi
                                let cont++
                                let j++
                        done
done

rm $fich
rm $fich1

echo "***"
exit

# MIEET
# Lidia Elena da Costa dos Reis
# Medição e Caracterização de Tráfego Tempo-Real em Redes IP
```

A.6 - *Script* utilizado editar a informação obtida no campo *Info* dos pacotes RTP (*editRTPinfo.sh*)

Utiliza o ficheiro *Info.txt* dos pacotes de dados, para obter informações importantes sobre estes pacotes, como o são o *sequence number* (utilizado para contar pacotes perdidos – ver Apêndice A.8) e o tempo (utilizado para o cálculo do *Jitter* – ver Apêndice A.7).

```
#!/bin/bash

####
# Cria um ficheiro com cada coluna da captura
####

nomefich="Info.txt"

# Sequence Number
file1="seqRTPT.txt"
cat $nomefich | cut -f 4 -d "=" > $file1
file1f="seqRTP.txt"
cat $file1 | cut -f 1 -d " " > $file1f

# Time
file2="timeRTPT.txt"
cat $nomefich | cut -f 5 -d "=" > $file2
file2f="timeRTP.txt"
cat $file2 | cut -f 1 -d " " > $file2f

rm $file1
rm $file2

echo "***"

exit

# MIEET
# Lidia Elena da Costa dos Reis
# Medição e Caracterização de Tráfego Tempo-Real em Redes IP
```

A.7 - *Script* utilizado para cálculo do *Jitter* (*calculaJitter.sh*)

Este *script* calcula o *Jitter* utilizando os ficheiros que contêm o tempo em que foi enviado o pacote (*time.txt*) e o tempo considerado pelo protocolo RTP (*timeRTP.txt* – consultar Apêndice A.6). O *Jitter* foi calculado seguindo o indicado na RFC 3550.

O código utilizado é o seguinte:

```
#!/bin/bash

####
# Calcula o jitter segundo a fórmula do RFC 3550
####

nometimestamp="timeRTP.txt"
nometime="time.txt"
```

```
# Número de linhas
wc -l $nometime > numlinhastxt.txt
numlinhas=`cat numlinhastxt.txt | cut -f 1 -d " "`
rm numlinhastxt.txt

varR=(`cat $nometime | while read i
do
echo $i
done;`)

varS=(`cat $nometimestamp | while read ii
do
echo $ii
done;`)

nomefichsaida="jitter.txt"
echo "Jitter" > $nomefichsaida

# D(a,b)=(Rb-Ra)-(Sb-Sa)*cte
# J(b)=J(a)+(|D(a,b)|-J(a))/16

cte=0.000125 # 1/8000, G.711 8000Hz

Ja=0
jj=0 # a
numlinhasx=$[$numlinhas-1]
while [ $jj -lt $numlinhasx ]
do
        k=$[$jj+1] # b
        #echo $jj
        echo $k

        Rb=${varR[$k]}
        Ra=${varR[$jj]}
        Sb=${varS[$k]}
        Sa=${varS[$jj]}

        R=`echo "$Rb-$Ra"|bc -l`
        Si=`echo "$Sb-$Sa"|bc -l`
        S=`echo "$Si*$cte"|bc -l`

        D=`echo "$R-$S"|bc -l`

        Dx=`echo "$D*100000000"|bc -l`
        echo $Dx > vard.txt
        dteste=`cat vard.txt | cut -b 1-2`

        rm vard.txt

        if [ $dteste -lt 0 ]
        then
                D=`echo "-1*$D"|bc -l`
        fi

        x=`echo "$D-$Ja"|bc -l`
        y=`echo "$x/16"|bc -l`
        Jb=`echo "$Ja+$y"|bc -l`
        echo $Jb >> $nomefichsaida
```

```
        echo "Jb=$Jb"
        let jj++
        Ja=$Jb # Próximo Ja é o actual Jb
done

echo "***"
exit

# MIEET
# Lidia Elena da Costa dos Reis
# Medição e Caracterização de Tráfego Tempo-Real em Redes IP
```

A.8 - *Script* utilizado para contar pacotes perdidos (*contaPacPerdidos.sh*)

Este *script* conta o número de pacotes perdidos utilizando o ficheiro que contém o *sequence number* (*seqRTP.txt* – consultar Apêndice A.6). O código utilizado é o seguinte:

```
#!/bin/bash

####
# Conta os pacotes de dados perdidos
####

nomefich1="seqRTP.txt"

# Número de linhas
wc -l $nomefich1 > numlinhastxt.txt
numlinhas=`cat numlinhastxt.txt | cut -f 1 -d " "`
rm numlinhastxt.txt

sequenceN=(`cat $nomefich1 | while read i
do
echo $i
done;`)

nomefichsaida="info_chamadas.txt"
echo "Tempo entre um pacote e o seguinte" > $nomefichsaida

perdidos=0          # Contador de pacotes perdidos
retransmitidos=0  # Contador de pacotes retransmitidos
foradeordem=0
j=0
while [ $j -lt $numlinhas ]
do
        k=$[$j+1]
        if [ $k -lt $numlinhas ]
        then
                # Cálculo de pacotes perdidos
                if [ ${sequenceN[$k]} -eq 0 ]
                then
                        vark=65536
                else
                        vark=${sequenceN[$k]}
                fi

                varj=${sequenceN[$j]}
                teste=$[$vark-$varj]
                if [ $teste -gt 1 ]
```

```
                then
                        perdidos=$[$perdidos+$teste]
                        echo "j=$j k=$k val1=$varj val2=$vark perdidos=$perdidos teste=$teste"
                elif [ $teste -eq 0 ]
                then
                        let retransmitidos++
                        echo    "j=$j    k=$k    val1=$varj    val2=$vark    retransmitidos=$retransmitidos
teste=$teste"

                elif [ $teste -lt 0 ]
                then
                        let foradeordem++
                        echo    "j=$j    k=$k    val1=$varj    val2=$vark    fora_de_ordem=$foradeordem
teste=$teste"
                fi
                #echo "j=$j k=$k val1=$varj val2=$vark perdidos=$perdidos teste=$teste"
        else
                echo "j=$j val1=$varj perdidos=$perdidos "
        fi
        let j++
done

echo "Numero de pacotes perdidos = $perdidos" > $nomefichsaida
echo "Numero de pacotes retransmitidos = $retransmitidos" >> $nomefichsaida
echo "Numero de pacotes fora de ordem = $foradeordem" >> $nomefichsaida
echo "***"

exit

# MIEET
# Lidia Elena da Costa dos Reis
# Medição e Caracterização de Tráfego Tempo-Real em Redes IP
```

A.9 - *Script* utilizado para localizar os pacotes RTP que contêm o *Mark* (*procura_Mark.sh*)

O primeiro pacote RTP transmitido numa chamada contém uma marca: *Mark*. Este *script* cria um ficheiro de texto (*RTP_mark.txt*), que contém todos os pacotes RTP que contêm a palavra *Mark*. Desta forma é possível localizar o início dos pacotes de dados das chamadas realizadas.

O código utilizado é o seguinte:

```
#!/bin/bash

####
# Cria um ficheiro com todos os pacotes que contêm o MARK
####

fichDados="linhas_Dados.txt"

# Procura o Mark (este sinaliza o inicio de uma chamada)
fichRTP_mark="RTP_mark.txt"
sed -n '/Mark/p' $fichDados > $fichRTP_mark

echo "***"
```

```
exit

# MIEET
# Lidia Elena da Costa dos Reis
# Medição e Caracterização de Tráfego Tempo-Real em Redes IP
```

A.10 - *Script* utilizado para obter informações importantes das chamadas realizadas ao longo da captura efectuada durante uma semana (*editchamadas.sh*)

Este *script* utiliza dois ficheiros obtidos através do *Wireshark, chamadastxt.txt* e *chamadascsv.txt,* os quais contêm informações sobre os pacotes *OpenReceiveChannel* e *CloseReceiveChannel,* em formatos diferentes, sendo mais conveniente obter algumas informações de um ficheiro e outras do outro.

As informações obtidas através deste *script* são: *Conference ID, IPDestino, IPOrigem, PortoDestino, PortoOrigem, Tempo e TipoPac,* as quais são guardadas em ficheiros diferentes com os respectivos nomes dentro do directório *Info*.

O código utilizado no *script* é o seguinte:

```
#!/bin/bash

####
# Obtém informações importantes das chamadas
####

# Obter conference ID
nomefich="chamadastxt.txt"
# Este ficheiro de texto é obtido através do Wireshark, filtrando a captura de forma a obter só os pacotes OpenReceiveChannel e CloseReceiveChannele guardar a mesma em formato txt

nomefich1="conferenceIDteste.txt"
grep  "\<Conference ID\>" $nomefich > $nomefich1

nomefich2="conferenceIDteste2.txt"
cat $nomefich1 | cut -f 2 -d ":" > $nomefich2

nomefich3="conferenceID.txt"
sed 's/ //g' $nomefich2 > $nomefich3

rm $nomefich1
rm $nomefich2

# Obter tempos
nomeficheiro="chamadascsv.txt"
# Este ficheiro de texto é obtido através do Wireshark, filtrando a captura de forma a obter só os pacotes OpenReceiveChannel e CloseReceiveChannele guardar a mesma em formato csv

nomeficheiro1="sem_aspas.txt" # Elimina as aspas
sed 's/"//g' $nomeficheiro > $nomeficheiro1

nomeficheiro2="Tempo.txt"
cat $nomeficheiro1 | cut -f 2 -d "," > $nomeficheiro2
```

```
nomeficheiro3="PortoOrigem.txt"
cat $nomeficheiro1 | cut -f 4 -d "," > $nomeficheiro3

nomeficheiro4="IPOrigem.txt"
cat $nomeficheiro1 | cut -f 3 -d "," > $nomeficheiro4

nomeficheiro5="PortoDestino.txt"
cat $nomeficheiro1 | cut -f 6 -d "," > $nomeficheiro5

nomeficheiro6="IPDestino.txt"
cat $nomeficheiro1 | cut -f 5 -d "," > $nomeficheiro6

nomeficheiro7="TipoPac.txt"
cat $nomeficheiro1 | cut -f 9 -d "," > $nomeficheiro7

rm $nomeficheiro1

directorio="Info"
mkdir $directorio
chmod 777 $directorio
cp $nomefich3 $nomeficheiro2 $nomeficheiro3 $nomeficheiro4 $nomeficheiro5 $nomeficheiro6
$nomeficheiro7 $directorio

rm $nomefich3
rm $nomeficheiro2
rm $nomeficheiro3
rm $nomeficheiro4
rm $nomeficheiro5
rm $nomeficheiro6
rm $nomeficheiro7

echo "***"

exit

# MIEET
# Lidia Elena da Costa dos Reis
# Medição e Caracterização de Tráfego Tempo-Real em Redes IP
```

A.11 - *Script* utilizado para separar os pacotes RTP dos RTCP (*VLAN_CT_DadosEdit.sh*)

Este *script* é utilizado para separar os pacotes do protocolo RTP dos pacotes pertencentes ao RTCP, pois nas chamadas entre um telefone VoIP (VLAN 1) e um telefone da central telefónica, além de existirem pacotes de dados RTP existem também pacotes de controlo RTCP.

O código referente a este *script* é o seguinte:

```
#!/bin/bash

####
# Separa os pacotes do protocolo RTCP dos pacotes do protocolo RTP
####
```

```
echo "Nome do ficheiro"
read nomefich

protocolUDP="UDP"
protocolRTP="RTP"

nomefich1="linhas_UDP.txt"
nomefich2="linhas_RTP.txt"

grep  "\<$protocolUDP\>" $nomefich > $nomefich1
grep  "\<$protocolRTP\>" $nomefich > $nomefich2

echo "***"

exit

# MIEET
# Lidia Elena da Costa dos Reis
# Medição e Caracterização de Tráfego Tempo-Real em Redes IP
```

A.12 - *Script* utilizado para separar o fluxo de dados da chamada em dois ficheiros (*chamadaA_B.sh*)

Este é utilizado para separar o fluxo de dados de uma chamada em dois ficheiros, contendo cada um deles pacotes que viajam num único sentido. Este *script* é aplicado a cada uma das chamadas efectuadas no teste 2 feito na rede VoIP.

Para isto é preciso introduzir o endereço IP origem de cada um dos fluxos (A e B), seguidamente é procurado na coluna correspondente ao *srcIP* (*source* IP), do ficheiro que contém todos os pacotes de dados, e separam-se os pacotes em dois ficheiros segundo o endereço IP origem, um para cada fluxo.

Este *script* origina dois ficheiros, um com o nome *chamada_a.txt* e outro *chamada_b.txt*. No primeiro ficheiro encontram-se todos os pacotes de dados cujo endereço IP origem coincide com o primeiro endereço IP introduzido. E o *chamada_b.txt* contém os pacotes de dados do fluxo que tem como origem o segundo endereço IP introduzido.

O código deste *script* é o seguinte:

```
#!/bin/bash

####
# Procura os IP e cria dois ficheiros, um com os dados que vão num sentido e outro com os que vão no sentido oposto
####

# Separa os fluxos considerando o IP origem
fichsrcIP="srcIP.txt"
cat $fichsrcIP | head -n 6
```

```
# Visto que no ficheiro original linhas_Dados.txt não mostra diferença entre o IP origem e o destino a não ser a sua posição, alterou-se este ficheiro de forma a esta ser mais notável e poder utilizar o grep
fichsrc="src.txt"
x=`cat $fichsrcIP | wc -l`
i=2

# Cria ficheiro linhas_Dados_2.txt, de forma a ser mais fácil localizar os srcIP
echo "srcIP" > $fichsrc
while [ $i -le $x ];
do
        echo "srcIP" >> $fichsrc
        let i++
done

fichsrcIP3="srcIP_novo.txt"
paste $fichsrc $fichsrcIP -d '=' > $fichsrcIP3

paste numero.txt time.txt $fichsrcIP3 srcPort.txt dstIP.txt dstPort.txt protocol.txt pktLength.txt Info.txt -d ','
> linhas_Dados_2.txt

rm $fichsrc
rm $fichsrcIP3

# Procura os srcIP e cria cada ficheiro
echo "Insira o endereço IP 1:"
read IP1
echo "Insira o endereço IP 2:"
read IP2
echo "Separando fluxos"
fich1a="chamada_a$ch.txt"
fich1b="chamada_b$ch.txt"
grep "\<srcIP=$IP1\>" linhas_Dados_2.txt > $fich1a
echo $IP1
grep "\<srcIP=$IP2\>" linhas_Dados_2.txt > $fich1b
echo $IP2
rm linhas_Dados_2.txt

echo "***"

exit

# MIEET
# Lidia Elena da Costa dos Reis
# Medição e Caracterização de Tráfego Tempo-Real em Redes IP
```

A.13 - *Script* utilizado para editar as capturas de vídeo que contêm unicamente pacotes RTP (*editcapRTP.sh*)

Este *script* é similar ao *editcap.sh* (Apêndice A.3), pois dá origem a um directório com vários ficheiros que contêm informações sobre a captura (consultar tabela 27, Apêndice A.3), neste caso o *script* é aplicado a um ficheiro *csv* que contém unicamente pacotes RTP das capturas feitas na rede de emissão de vídeo.

O código utilizado é o seguinte:

```
#!/bin/bash

####
# Cria um ficheiro com cada coluna da captura, isto é um ficheiro por elemento da captura
####

echo "Nome do ficheiro:"
read nomefich1

temp1="temp1.txt"
nomefich="RTP_$nomefich1.txt"
nomefich_temp="temp2.txt"
nomefich_temp1="temp3.txt"

sed 's/"No.","Time","Source","Src Port","Destination","Dst Port","Protocol","Pkt Length","Info"//g' $nomefich1 > $nomefich_temp
sed '/^$/d' $nomefich_temp > $nomefich_temp1


# Troca "," por espaços, desta forma cada campo fica separado por um espaço
nomeficheiro="ficheiroteste.txt"
sed 's/,/ /g' $nomefich_temp1 > $temp1

# Volta a colocar as  ","
fich3="semDup.txt"
sed 's/" "/","/g'  $temp1 > $nomefich

# Elimina as aspas
nomeficheiro="sem_aspas.txt" # elimina as aspas
sed 's/"//g' $nomefich > $nomeficheiro

# Número do pacote
file1="numero.txt"
cat $nomeficheiro | cut -f 1 -d "," > $file1
echo $file1

# Time
file2="time.txt"
cat $nomeficheiro | cut -f 2 -d "," > $file2
echo $file2

# Source IP
file3="srcIP.txt"
cat $nomeficheiro | cut -f 3 -d "," > $file3
echo $file3

# Source Port
file4="srcPort.txt"
cat $nomeficheiro | cut -f 4 -d "," > $file4
echo $file4

# Destination IP
file5="dstIP.txt"
cat $nomeficheiro | cut -f 5 -d "," > $file5
echo $file5

# Destination Port
file6="dstPort.txt"
```

```
cat $nomeficheiro | cut -f 6 -d "," > $file6
echo $file6

# Protocol
file7="protocol.txt"
cat $nomeficheiro | cut -f 7 -d "," > $file7
echo $file7

# Pkt Length
file8="pktLength.txt"
cat $nomeficheiro | cut -f 8 -d "," > $file8
echo $file8

# Info
file9="Info.txt"
cat $nomeficheiro | cut -f 9 -d "," > $file9
echo $file9

directorio="cap_$nomefich"
mkdir $directorio
chmod 777 $directorio
cp $file1 $file2 $file3 $file4 $file5 $file6 $file7 $file8 $file9 $directorio

rm $file1
rm $file2
rm $file3
rm $file4
rm $file5
rm $file6
rm $file7
rm $file8
rm $file9
rm $nomeficheiro
rm $temp1
rm $nomefich
rm $nomefich_temp
rm $nomefich_temp1
echo "***"

exit

# MIEET
# Lidia Elena da Costa dos Reis
# Medição e Caracterização de Tráfego Tempo-Real em Redes IP
```

A.14 - *Script* utilizado para organizar os pacotes de dados (*organizarPacotesDados.sh*)

Perante a existência de pacotes fora de ordem não é possível contabilizar correctamente o número de pacotes perdidos recorrendo ao *contaPacPerdidos.sh* (pois este *script* só considera a situação em que os pacotes estão organizados), para poder obter valores correctos é necessário organizar os pacotes utilizando o *organizarPacotesDados.sh* e posteriormente aplicar o *script* que contabiliza os pacotes perdidos.

O código utilizado neste *scritp* é o seguinte:

```
#!/bin/bash

####
# Quando é detectada a existência de pacotes fora de ordem aplica-se este scritp de forma organizar os pacotes
####

fich="seqRTP.txt"
fichOrg="seqRTPorg.txt"
sort -n -o $fichOrg $fich

exit

# MIEET
# Lidia Elena da Costa dos Reis
# Medição e Caracterização de Tráfego Tempo-Real em Redes IP
```

A.15 - *Script* que devolve os portos utilizados (*Portos_src_dst.sh*)

Este *script* devolve num ficheiro os números de portos utilizados numa determinada captura, para isto recorre-se ao seguinte código:

```
#!/bin/bash

# Este script cria um ficheiro com os portos utilizados

portosrc="srcPort.txt" #ficheiro que contem os portos origem de cada pacote
portodst="dstPort.txt" #ficheiro que contem os portos destino de cada pacote


fich="Port_src_dst.txt"
fichtemp="Port_src_dst_temp.txt"

portosrc1="srcPorttemp.txt"
portodst1="dstPorttemp.txt"

sed 's/Src Port//g' $portosrc > $portosrc1
sed 's/Dst Port//g' $portodst > $portodst1

# cria ficheiro com os portos utilizados
echo "Src Port" > $fichtemp
cat $portosrc1 | sort | uniq >> $fichtemp
echo "Dst Port" >> $fichtemp
cat $portodst1 | sort | uniq >> $fichtemp

sed '/^$/d' $fichtemp > $fich

# eliminar ficheiros temporários
rm $fichtemp
rm $portosrc1
rm $portodst1

exit

# MIEET
# Lidia Elena da Costa dos Reis
# Medição e Caracterização de Tráfego Tempo-Real em Redes IP
```

A.16 - *Script* utilizado para contabilizar os pacotes segundo o tamanho (*tamanhosPacVoIP.sh*)

Este *script* contabiliza os pacotes segundo o tamanho, neste caso dividiu-se os pacotes em diferentes grupos segundo o tamanho:

- Entre 1 e 63*bytes*
- Entre 64 e 127*bytes*
- Entre 128 e 255*bytes*
- Entre 256 e 511*bytes*
- Entre 512 e 1023*bytes*
- Entre 1024 e 2048*bytes*

```
#!/bin/bash
#####
# Contabiliza o número de pacotes segundo os tamanhos
#####
pktLength="pktLength.txt" # Ficheiro que contem os tamanhos dos pacotes
pktLength1="pktLengthtemp.txt"
pktLength2="pktLengthtemp2.txt"

sed 's/Pkt Length//g' $pktLength > $pktLength1
sed '/^$/d' $pktLength1 > $pktLength2

pktOrg="pktOrg.txt"

cat $pktLength2 | sort -g > $pktOrg

wc -l $pktOrg > numlinhastxt.txt
numlinhas=`cat numlinhastxt.txt | cut -f 1 -d " "`
rm numlinhastxt.txt

resultado="tamanhosPac.txt"

# Estes valores variam entre 73 e 1456bytes, os quais vao ser contabilizados nos seguintes grupos: menor que 64, entre 64 e 128, entre 128 e 256, entre 256 e 512, entre 512 e 1024, entre 1024 e 2048
contador=0
linha=0

echo "Numero de linhas $numlinhas"
echo "Entre 0 e 63 bytes" > $resultado
cat $pktOrg | while read i
        do
                let linha++
                valor1=64
                if [ "$i" -lt $valor1 ]; then
                        let contador++
                fi
                if [ $linha -eq $numlinhas ]; then
                        echo $contador >> $resultado
                        echo "Entre 0 e 127 bytes --> $contador"
                fi
        done
```

```
# Reinicia o contador
contador=0
linha=0

echo "Entre 64 e 127 bytes" >> $resultado
cat $pktOrg | while read i
	do
		let linha++
		valor1=128
		valor2=64
		if [ "$i" -lt $valor1 ]; then  # menor
			if [ "$i" -ge $valor2 ]; then # maior igual
				let contador++
			fi
		fi
		if [ $linha -eq $numlinhas ]; then
			echo $contador >> $resultado
			echo "Entre 128 e 255 bytes --> $contador"
		fi
	done

# Reinicia o contador
contador=0
linha=0

echo "Entre 128 e 255 bytes" >> $resultado
cat $pktOrg | while read i
	do
		let linha++
		valor1=256
		valor2=128
		if [ "$i" -lt $valor1 ]; then  # menor
			if [ "$i" -ge $valor2 ]; then # maior igual
				let contador++
			fi
		fi
		if [ $linha -eq $numlinhas ]; then
			echo $contador >> $resultado
			echo "Entre 128 e 255 bytes --> $contador"
		fi
	done

# Reinicia o contador
contador=0
linha=0

echo "Entre 256 e 511 bytes" >> $resultado
cat $pktOrg | while read i
	do
		let linha++
		valor1=512
		valor2=256
		if [ "$i" -lt $valor1 ]; then  # menor
			if [ "$i" -ge $valor2 ]; then # maior igual
				let contador++
			fi
		fi
```

```
                        if [ $linha -eq $numlinhas ]; then
                                echo $contador >> $resultado
                                echo "Entre 256 e 511 bytes --> $contador"
                        fi
                done

# Reinicia o contador
contador=0
linha=0

echo "Entre 512 e 1023 bytes" >> $resultado
cat $pktOrg | while read i
                do
                        let linha++
                        valor1=1024
                        valor2=512
                        if [ "$i" -lt $valor1 ]; then  # menor
                                if [ "$i" -ge $valor2 ]; then # maior igual
                                        let contador++
                                fi
                        fi
                        if [ $linha -eq $numlinhas ]; then
                                echo $contador >> $resultado
                                echo "Entre 512 e 1023 bytes --> $contador"
                        fi
                done

# Reinicia o contador
contador=0
linha=0

echo "Entre 1024 e 2047 bytes" >> $resultado
cat $pktOrg | while read i
                do
                        let linha++
                        valor1=2048
                        valor2=1024
                        if [ "$i" -lt $valor1 ]; then  # menor
                                if [ "$i" -ge $valor2 ]; then # maior igual
                                        let contador++
                                fi
                        fi
                        if [ $linha -eq $numlinhas ]; then
                                echo $contador >> $resultado
                                echo "Entre 1024 e 2047 bytes --> $contador"
                        fi
                done

rm $pktLength1
rm $pktLength2
rm $pktOrg
exit
# MIEET
# Lidia Elena da Costa dos Reis
# Medição e Caracterização de Tráfego Tempo-Real em Redes IP
```

Apêndice B - Gráficos do tráfego da captura do teste 1

B.1 - Tráfego de dados

Estes gráficos mostram o comportamento dos pacotes e bytes de dados transmitidos ao longo da captura.

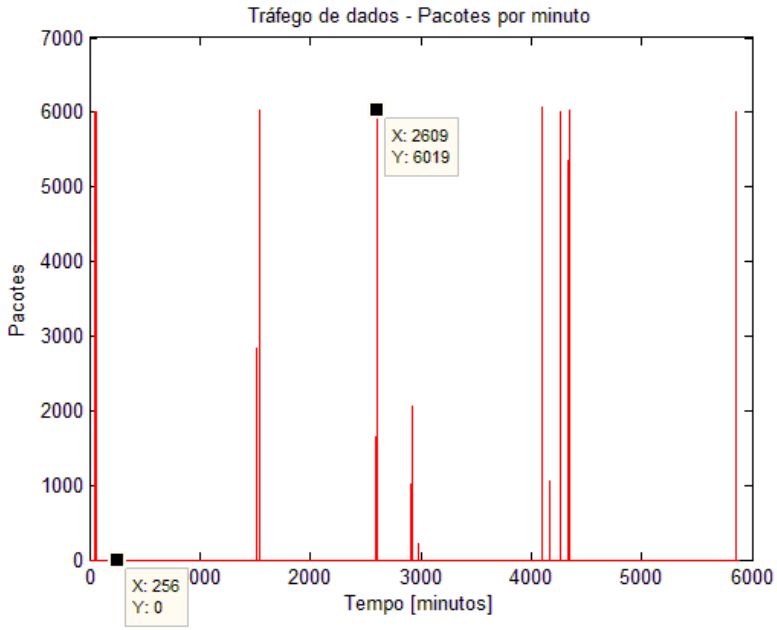

Figura 109 – Tráfego de dados em pacotes por minuto

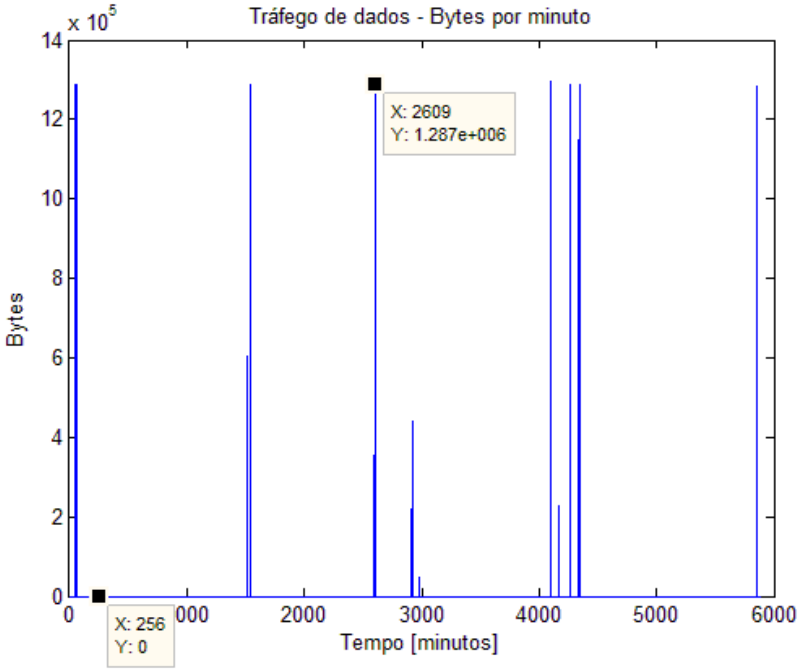

Figura 110 – Tráfego de dados em bytes por minuto

B.2 - Tráfego de sinalização

O seguinte gráfico mostra o comportamento dos pacotes e bytes de sinalização trocados ao longo da captura:

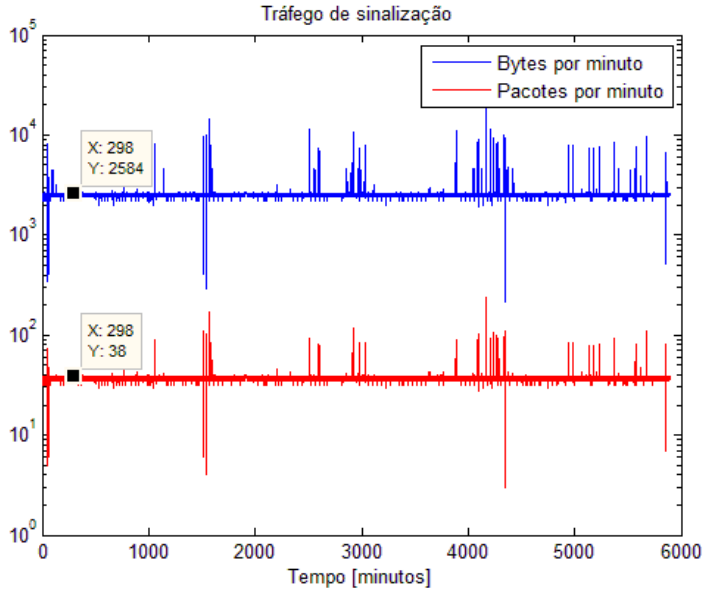

Figura 111 – Tráfego de sinalização em bytes e pacotes por minuto

Na rede VoIP utilizam-se dois tipos de protocolos de sinalização, nos seguintes gráficos encontra-se representado o comportamento do tráfego do protocolo MGCP e do SCCP.

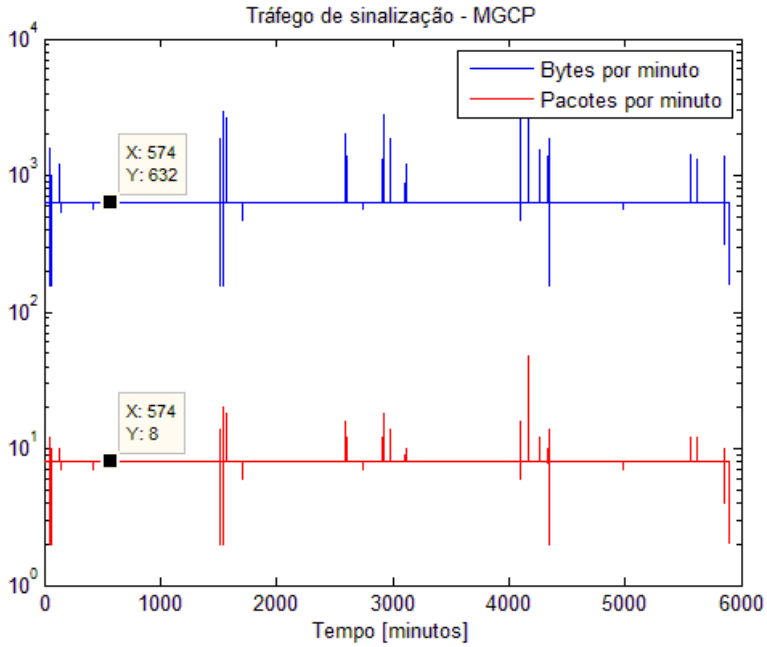

Figura 112 – Protocolo MGCP

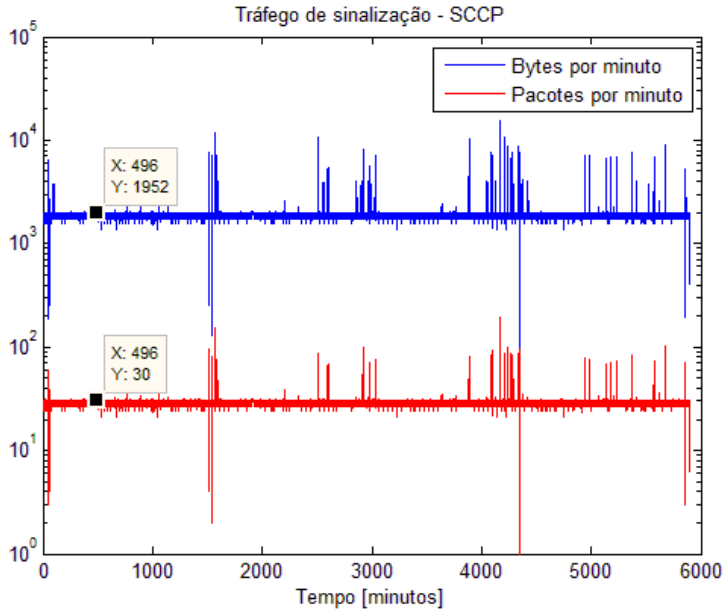

Figura 113 – Protocolo SCCP

Apêndice C - Chamadas do teste 2

C.1 - Exemplos de chamadas

C.1.1 Exemplo de pacotes trocados numa chamada entre dois telefones da VLAN 1

A seguinte figura mostra um exemplo de estabelecimento de uma chamada entre dois telefones da VLAN 1, extraído da terceira chamada realizada às 17 horas.

Figura 114 – Exemplo de estabelecimento de uma chamada entre dois telefones da VLAN 1

A seguinte figura mostra a troca de mensagens de sinalização durante a finalização da mesma chamada.

Figura 115 – Exemplo de finalização de uma chamada entre dois telefones da VLAN 1

Seguidamente mostra-se uma imagem que mostra como é feita a troca de pacotes de dados durante uma chamada entre dois telefones da VLAN 1.

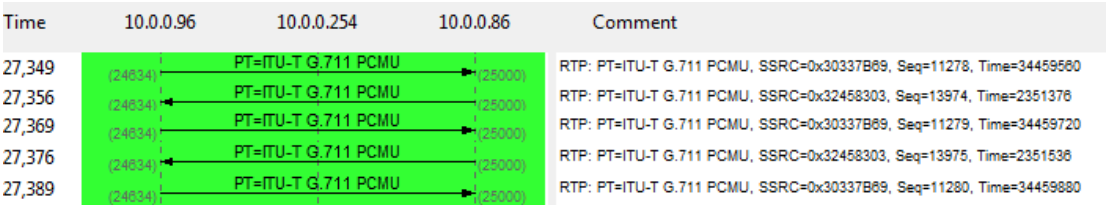

Figura 116 – Exemplo dos pacotes de dados trocados ao longo de uma chamada entre dois telefones da VLAN 1

C.1.2 Exemplo de pacotes trocados numa chamada entre um telefone da VLAN 1 e um da VLAN 2

A seguinte figura mostra uma secção do fluxo de dados trocado durante uma chamada entre dois telefones VoIP, um na VLAN 1 e outro na VLAN 2. Este exemplo foi extraído da primeira chamada realizada às 14 horas.

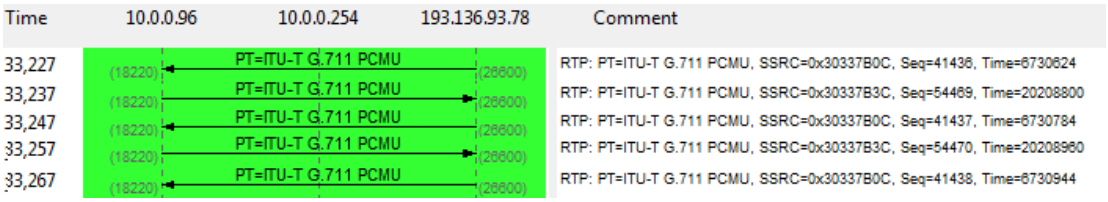

Figura 117 – Exemplo dos pacotes de dados trocados ao longo de uma chamada entre dois telefones VoIP, um na VLAN 1 e outro na VLAN 2

As seguintes figuras mostram a troca de pacotes durante o estabelecimento e finalização da chamada.

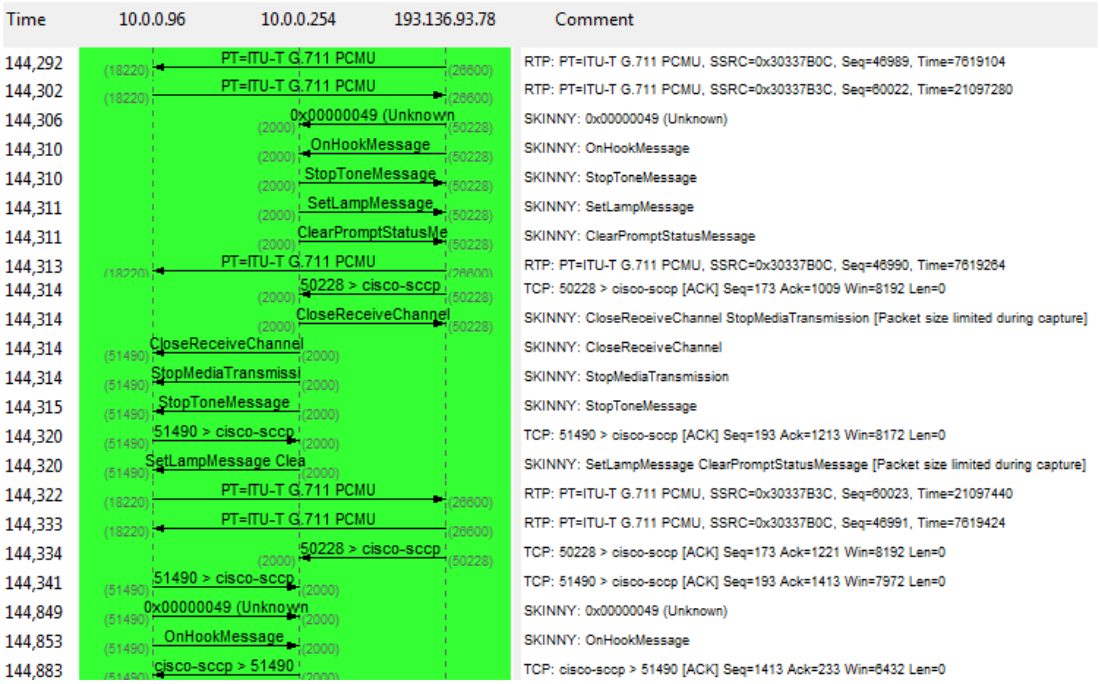

Figura 118 – Exemplo dos pacotes trocados durante a finalização de uma chamada entre um telefone da VLAN 1 e um da VLAN 2

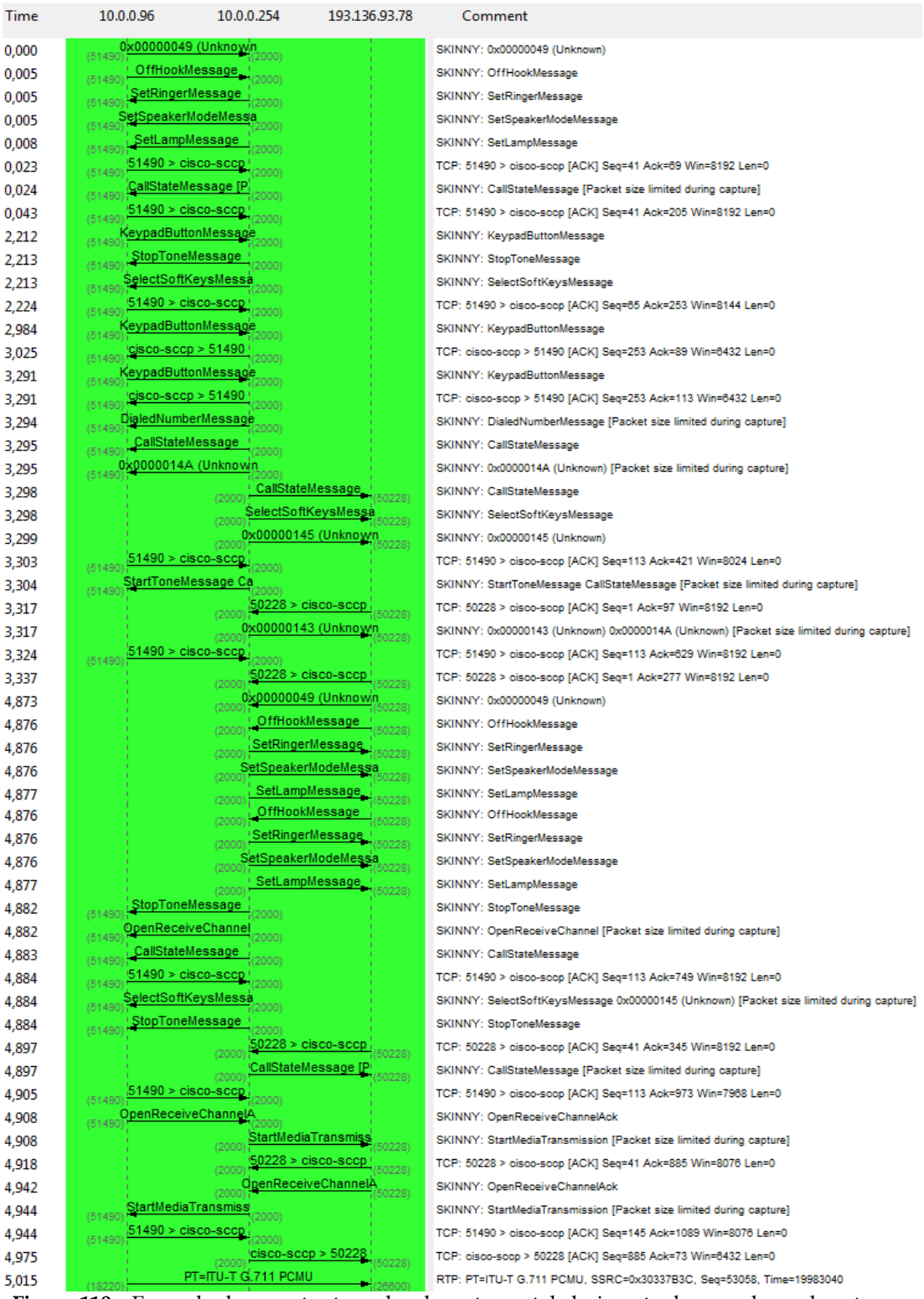

Figura 119 – Exemplo dos pacotes trocados durante o estabelecimento de uma chamada entre um telefone da VLAN 1 e um na VLAN 2

C.1.3 Exemplo de pacotes trocados numa chamada entre um telefone da VLAN 1 e um da central telefónica

A seguinte figura mostra um exemplo de estabelecimento de uma chamada entre um telefone da VLAN 1 e um da central telefónica, extraído da primeira chamada realizada às 11horas.

Figura 120 – Exemplos dos pacotes trocados durante o estabelecimento de uma chamada entre um telefone na VLAN 1 e um na central telefónica

Neste caso as mensagens de dados são trocadas entre o telefone VoIP e o *router*, como mostra a seguinte figura.

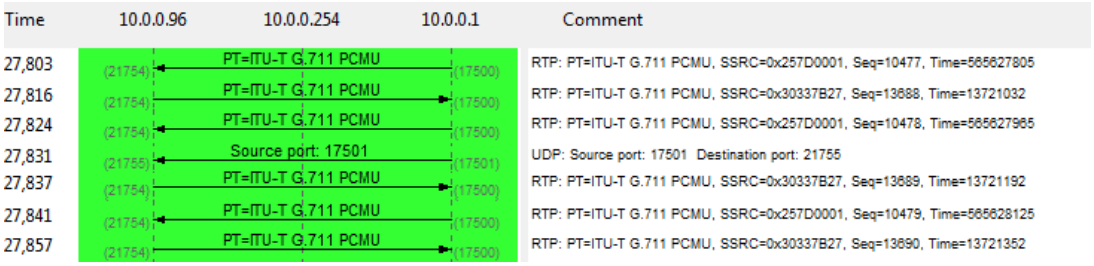

Figura 121 – Exemplos dos pacotes trocados durante a chamada entre um telefone na VLAN 1 e um na central telefónica

Os pacotes trocados na finalização da sessão da mesma chamada encontram-se na seguinte figura.

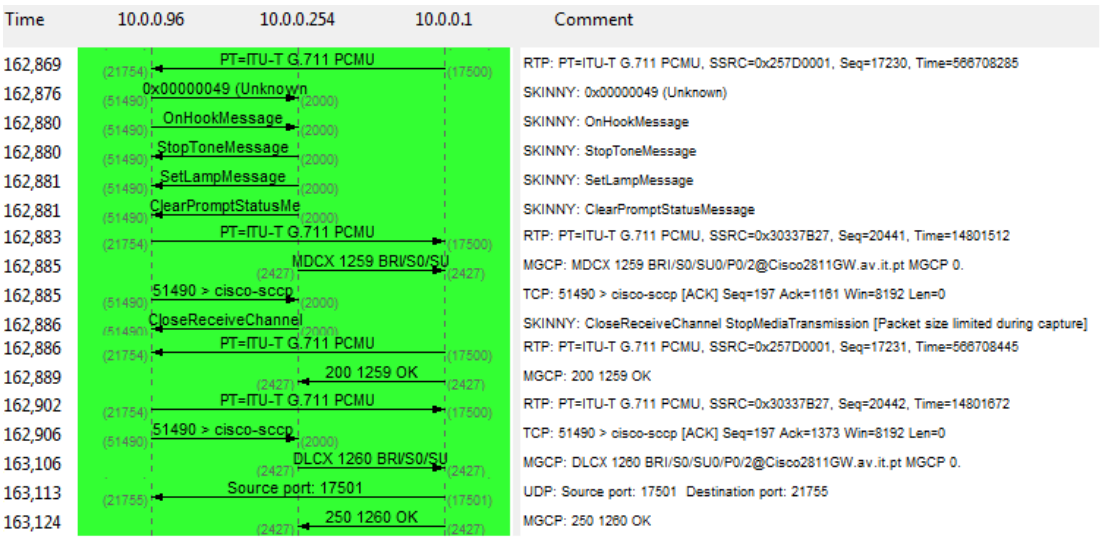

Figura 122 – Exemplos dos pacotes trocados durante a finalização de uma chamada entre um telefone na VLAN 1 e um na central telefónica

C.2 - Estatísticas

C.2.1 Tabelas com valores estatísticos das chamadas realizadas entre dois telefones da VLAN 1

A seguinte tabela mostra os valores estatísticos dos pacotes por segundos de dados e sinalização em cada chamada realizada entre dois telefones VoIP da VLAN 1:

			Dados [*pps*]			Sinalização [*pps*]		
			11h	14h	17h	11h	14h	17h
VLAN 1 – VLAN 1	1	Valor Máximo	100,0	100,0	101,0	19,0	21,0	24,0
		Média	92,0	95,6	95,9	0,7	0,7	0,8
		Valor Mínimo	0,0	0,0	0,0	0,0	0,0	0,0
		Mediana	100,0	100,0	100,0	0,0	0,0	0,0
		Moda	100,0	100,0	100,0	0,0	0,0	0,0
		Variância	713,1	392,5	374,7	8,5	7,5	11,0
		Desvio Padrão	26,7	19,8	19,4	2,9	2,7	3,3
	2	Valor Máximo	100,0	100,0	101,0	21,0	29,0	20,0
		Média	91,2	94,4	94,4	0,7	0,7	0,9
		Valor Mínimo	0,0	0,0	0,0	0,0	0,0	0,0
		Mediana	100,0	100,0	100,0	0,0	0,0	0,0
		Moda	100,0	100,0	100,0	0,0	0,0	0,0
		Variância	772,1	507,0	501,4	9,1	10,9	9,8
		Desvio Padrão	27,8	22,5	22,4	3,0	3,3	3,1
	3	Valor Máximo	100,0	100,0	101,0	20,0	28,0	19,0
		Média	92,6	89,9	97,3	0,6	0,7	0,8

		Valor Mínimo	0,0	0,0	0,0	0,0	0,0	0,0
		Mediana	100,0	100,0	100,0	0,0	0,0	0,0
		Moda	100,0	100,0	100,0	0,0	0,0	0,0
		Variância	670,9	890,7	233,1	6,4	9,3	9,0
		Desvio Padrão	25,9	29,8	15,3	2,5	3,0	3,0

Tabela 28 – Valores estatísticos das chamadas entre dois telefones da VLAN 1

A seguinte tabela contém as percentagens de pacotes perdidos e os valores do *jitter* médio (é o valor médio dos valores de *jitter* obtidos ao longo da chamadas), para cada sentido A e B, onde A representa o tráfego que vai do 10.0.0.96 para o 10.0.0.89 (ou 10.0.0.86, alguns testes foram feitos num telefone e outros no outro, devido à disponibilidade dos mesmos), e B representa o fluxo no sentido oposto.

Hora	Fluxo	VLAN 1 – VLAN 1	
		Pacotes perdidos [%]	*Jitter* [*ms*]
11h 1	A	0	0,1624
	B	0	0,1374
11h 2	A	0	0,0331
	B	0	0,0276
11h 3	A	0	0,0415
	B	0	0,0322
14h 1	A	0	0,0394
	B	0	0,0394
14h 2	A	0	0,0486
	B	0	0,0361
14h 3	A	0	0,0507
	B	0	0,0336
17h 1	A	0	0,0234
	B	0	0,0283
17h 2	A	0	0,0283
	B	0	0,0439
17h 3	A	0	0,0248
	B	0	0,0368

Tabela 29 – Fluxo A e B das chamadas entre os dois telefones da VLAN 1

Os seguintes gráficos mostram o comportamento da largura de banda ao longo das chamadas:

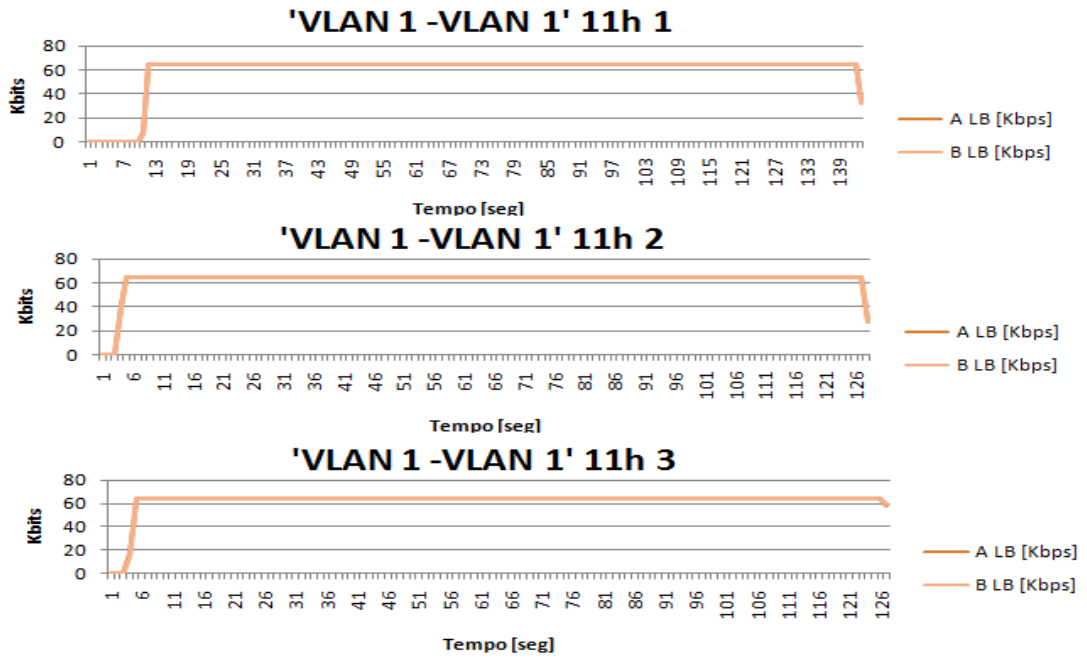

Figura 123 – Largura de banda ocupada nas chamadas efectuadas às 11 horas entre dois telefones da VLAN 1

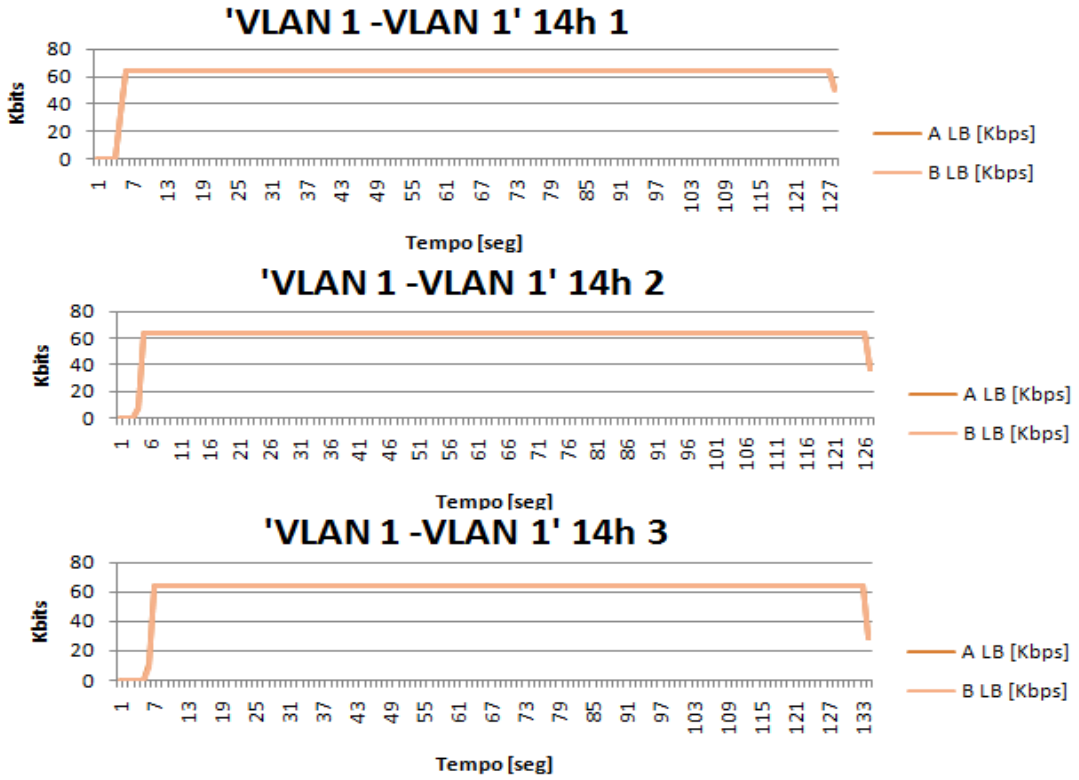

Figura 124 – Largura de banda ocupada nas chamadas efectuadas às 14 horas entre dois telefones da VLAN 1

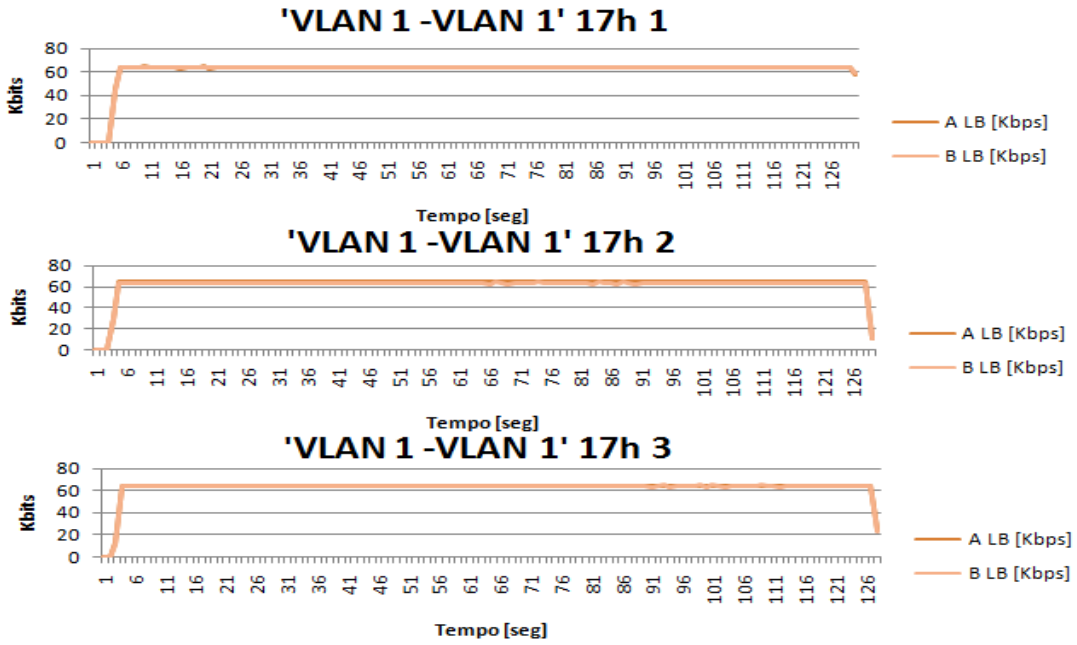

Figura 125 – Largura de banda ocupada nas chamadas efectuadas às 17 horas entre dois telefones da VLAN 1

C.2.2 Tabelas com valores estatísticos das chamadas realizadas entre um telefone da VLAN 1 e um da VLAN 2

Os valores estatísticos referentes aos pacotes de dados e sinalização transmitidos por segundo ao longo das chamadas efectuadas entre um telefone VoIP da VLAN 1 e um telefone VoIP da VLAN 2, encontram-se especificados na seguinte tabela:

			Dados [*pps*]			Sinalização [*pps*]		
			11h	14h	17h	11h	14h	17h
VLAN 1 – VLAN 2	1	Valor Máximo	101,0	101,0	101,0	20,0	21,0	20,0
		Média	90,9	96,1	91,5	0,7	0,6	0,8
		Valor Mínimo	0,0	0,0	0,0	0,0	0,0	0,0
		Mediana	100,0	100,0	100,0	0,0	0,0	0,0
		Moda	100,0	100,0	100,0	0,0	0,0	0,0
		Variância	809,8	362,1	756,6	8,4	7,4	8,0
		Desvio Padrão	28,5	19,0	27,5	2,9	2,7	2,8
	2	Valor Máximo	101,0	101,0	100,0	27,0	27,0	26,0
		Média	70,2	88,7	96,7	0,7	0,5	0,7
		Valor Mínimo	0,0	0,0	0,0	0,0	0,0	0,0
		Mediana	100,0	100,0	100,0	0,0	0,0	0,0
		Moda	100,0	100,0	100,0	0,0	0,0	0,0
		Variância	2056,5	957,2	293,4	10,6	8,0	9,6
		Desvio Padrão	45,3	30,9	17,1	3,3	2,8	3,1
	3	Valor Máximo	101,0	100,0	100,0	23,0	21,0	19,0
		Média	89,2	80,1	87,1	0,7	0,7	0,6
		Valor Mínimo	0,0	0,0	0,0	0,0	0,0	0,0

		Mediana	100,0	100,0	100,0	0,0	0,0	0,0
		Moda	100,0	100,0	100,0	0,0	0,0	0,0
		Variância	943,3	1571,2	1096,0	8,0	7,4	5,8
		Desvio Padrão	30,7	39,6	33,1	2,8	2,7	2,4

Tabela 30 – Valores estatísticos das chamadas entre dois telefones VoIP, um na VLAN 1 e outro na VLAN 2

A seguinte tabela contém informações sobre a percentagem de pacotes perdidos e a média do *jitter*, para ambos sentidos do fluxo, onde A representa o fluxo de pacotes desde o telefone 10.0.0.96 para o 193.136.93.78, e B representa o sentido oposto do fluxo.

Hora	Fluxo	VLAN 1 – VLAN 2	
		Pacotes perdidos [%]	*Jitter* [*ms*]
11h 1	A	0,04	0,0254
	B	0,04	0,0346
11h 2	A	11,41	0,0347
	B	11,39	0,0448
11h 3	A	0,08	0,0260
	B	0,07	0,0357
14h 1	A	0	0,0324
	B	0	0,0468
14h 2	A	0	0,0285
	B	0	0,0285
14h 3	A	0	0,0242
	B	0	0,0242
17h 1	A	0	0,0265
	B	0	0,0485
17h 2	A	0	0,0249
	B	0	0,0452
17h 3	A	0	0,0256
	B	0	0,0381

Tabela 31 – Fluxo A e B das chamadas entre os dois telefones VoIP, um na VLAN 1 e outro na VLAN 2

Seguidamente encontram-se os gráficos da largura de banda ocupada ao longo de cada chamada:

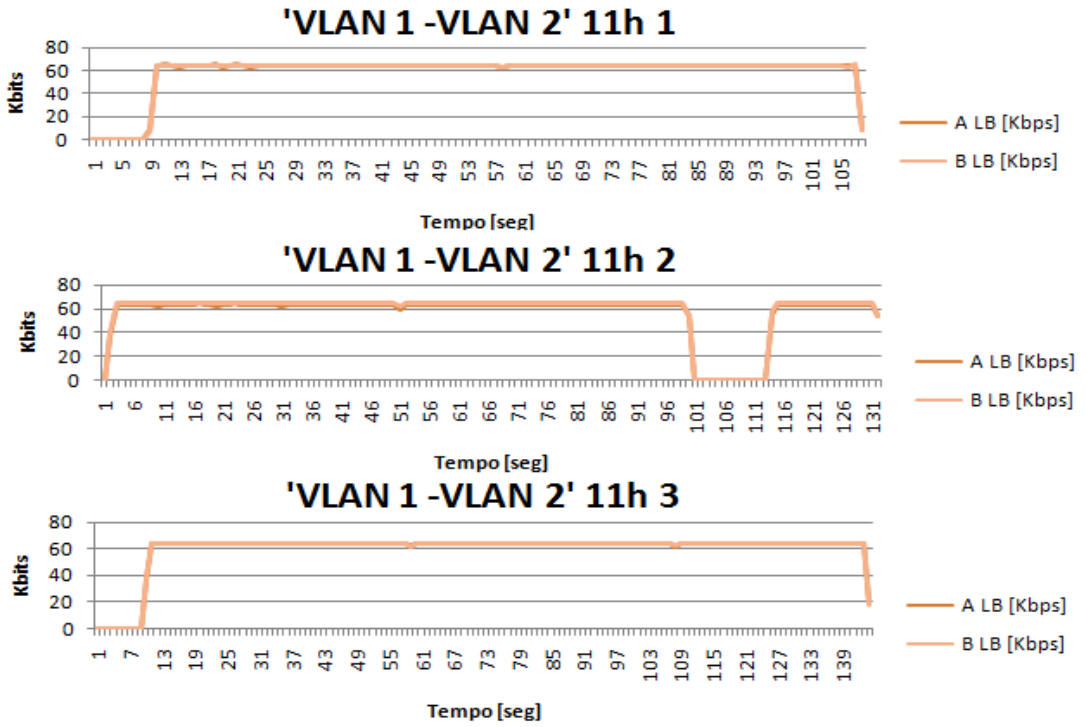

Figura 126 – Largura de banda ocupada nas chamadas efectuadas às 11 horas entre um telefone da VLAN 1 e um da VLAN 2

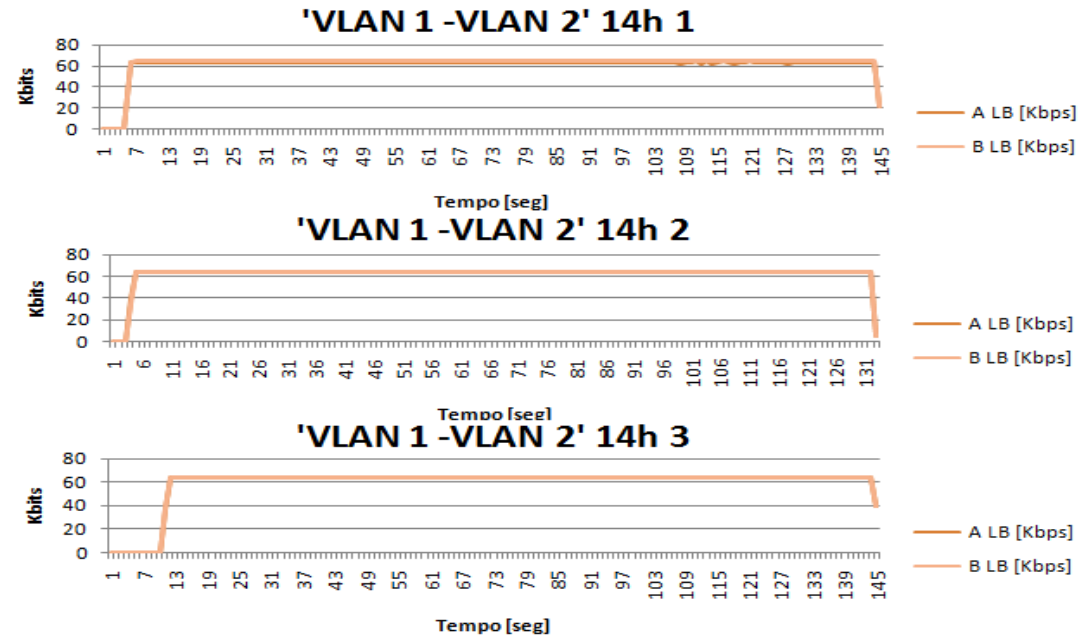

Figura 127 – Largura de banda ocupada nas chamadas efectuadas às 14 horas entre um telefone da VLAN 1 e um da VLAN 2

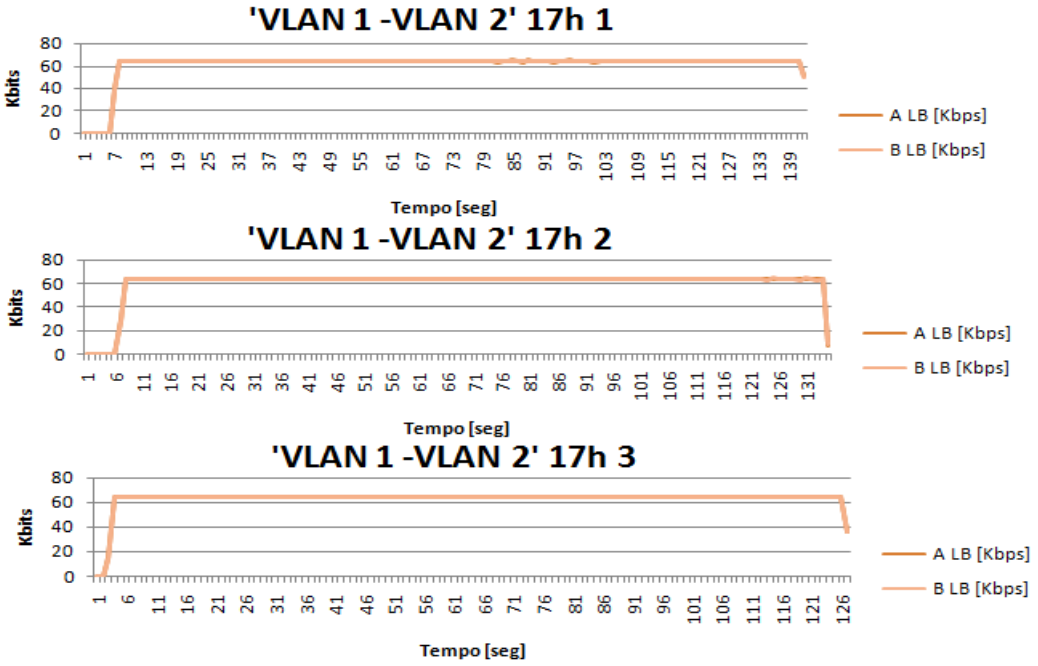

Figura 128 – Largura de banda ocupada nas chamadas efectuadas às 17 horas entre um telefone da VLAN 1 e um da VLAN 2

C.2.3 Tabelas com valores estatísticos das chamadas realizadas entre um telefone da VLAN 1 e um da central telefónica

As seguintes tabelas contêm os valores estatísticos dos pacotes de dados e sinalização, das chamadas entre um telefone VoIP da VLAN 1 e um da central telefónica (CT). Também contêm os valores do *jitter* médio e percentagens de pacotes perdidos (o fluxo A representa os pacotes que têm como origem o telefone 10.0.0.96 e destino o *router* (10.0.0.1), enquanto que o fluxo B representa os pacotes que vão no sentido oposto).

			Dados [*pps*]			Sinalização [*pps*]		
			11h	14h	17h	11h	14h	17h
VLAN 1 – CT	1	Valor Máximo	102,0	101,0	105,0	21,0	19,0	16,0
		Média	98,6	97,6	98,4	0,5	0,6	0,6
		Valor Mínimo	0,0	0,0	0,0	0,0	0,0	0,0
		Mediana	100,0	100,0	100,0	0,0	0,0	0,0
		Moda	100,0	100,0	100,0	0,0	0,0	0,0
		Variância	139,4	226,2	171,8	4,2	5,7	5,3
		Desvio Padrão	11,8	15,0	13,1	2,0	2,4	2,3
	2	Valor Máximo	101,0	101,0	101,0	19,0	21,0	20,0
		Média	97,6	99,3	97,8	0,7	0,5	0,6
		Valor Mínimo	0,0	0,0	0,0	0,0	0,0	0,0
		Mediana	100,0	100,0	100,0	0,0	0,0	0,0
		Moda	100,0	100,0	100,0	0,0	0,0	0,0
		Variância	230,8	66,8	210,5	5,5	3,7	5,1

		Desvio Padrão	15,2	8,2	14,5	2,4	1,9	2,3
	3	Valor Máximo	102,0	102,0	102,0	23,0	20,0	22,0
		Média	92,4	99,3	98,9	0,6	0,5	0,6
		Valor Mínimo	0,0	0,0	0,0	0,0	0,0	0,0
		Mediana	100,0	100,0	100,0	0,0	0,0	0,0
		Moda	100,0	100,0	100,0	0,0	0,0	0,0
		Variância	703,1	64,0	97,1	5,7	4,1	5,3
		Desvio Padrão	26,5	8,0	9,9	2,4	2,0	2,3

Tabela 32 – Valores estatísticos das chamadas entre um telefone da VLAN 1 e um da central telefónica

Hora	Fluxo	VLAN 1 – CT	
		Pacotes perdidos [%]	*Jitter* [*ms*]
11h 1	A	0	0,0302
	B	0	1,5849
11h 2	A	0	0,0302
	B	0	1,5737
11h 3	A	0	0,0288
	B	0	1,5769
14h 1	A	0	0,0398
	B	0	1,6007
14h 2	A	0	0,0390
	B	0	1,5725
14h 3	A	0	0,0367
	B	0	1,5797
17h 1	A	0	0,3379
	B	0	1,6654
17h 2	A	0	0,0389
	B	0	1,5896
17h 3	A	0	0,0303
	B	0	1,5785

Tabela 33 – Fluxo A e B das chamadas entre um telefone da VLAN 1 e um da central telefónica

A seguinte tabela mostra as percentagens de pacotes de dados e a percentagem de pacotes de controlo, pois numa chamada feita a um telefone na central telefónica, ao longo da transmissão de dados o *router* envia ao telefone pacotes RTCP através dos portos X+1, quando o porto X é o porto utilizado pelo RTP para a transmissão de dados.

Hora	Fluxo	RTP	
		Pacotes de dados - RTP [%]	Pacotes de controlo – RTCP [%]
11h 1	A	100	0
	B	99,58	0,42
11h 2	A	100	0
	B	99,60	0,40
11h 3	A	100	0
	B	99,59	0,41
14h 1	A	100	0
	B	99,56	0,44
14h 2	A	100	0
	B	99,60	0,40
14h 3	A	100	0
	B	99,58	0,42
17h 1	A	100	0
	B	99,57	0,43
17h 2	A	100	0
	B	99,59	0,41
17h 3	A	100	0
	B	99,57	0,43

Tabela 34 – Pacotes de dados e de controlo do protocolo RTP

Um exemplo destes pacotes RTCP é o seguinte, o qual foi extraído da primeira chamada efectuada às 11*horas*. Como se pode observar os pacotes RTP utilizam o porto UDP 17500 no caso do *router* (10.0.0.1) e o porto UDP 21754 no caso do telefone VoIP da VLAN 1 (10.0.0.96), e os pacotes de controlo utilizam os portos UDP 17501 e 21755, respectivamente.

No. .	Time	Source	Src port	Destination	Dest port	Protocol	Pkt Length	Info
557	4.200603	10.0.0.1	17500	10.0.0.96	21754	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x257D0001, Seq=9297, Time=565439005
559	4.216341	10.0.0.96	21754	10.0.0.1	17500	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x30337B27, Seq=12508, Time=13532232
561	4.221829	10.0.0.1	17500	10.0.0.96	21754	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x257D0001, Seq=9298, Time=565439165
563	4.227595	10.0.0.1	17501	10.0.0.96	21755	RTCP	170	Sender Report
565	4.235314	10.0.0.96	21754	10.0.0.1	17500	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x30337B27, Seq=12509, Time=13532392
566	4.241032	10.0.0.1	17500	10.0.0.96	21754	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x257D0001, Seq=9299, Time=565439325

Figura 129 – Exemplo de pacotes de controlo do protocolo RTP trocados durante uma chamada entre um telefone da VLAN 1 e um da central telefónica

Devido à existência destes pacotes de controlo (RTCP) foi preciso recorrer a um outro *script*, para separar os RTP dos RTCP e poder assim calcular o *jitter* e os pacotes perdidos, este *scritp* é o *VLAN_CT_DadosEdit.sh* (Apêndice A.11)

Os seguintes gráficos mostram o comportamento da largura de banda ao longo de uma chamada entre um telefone da VLAN 1 e um da central telefónica:

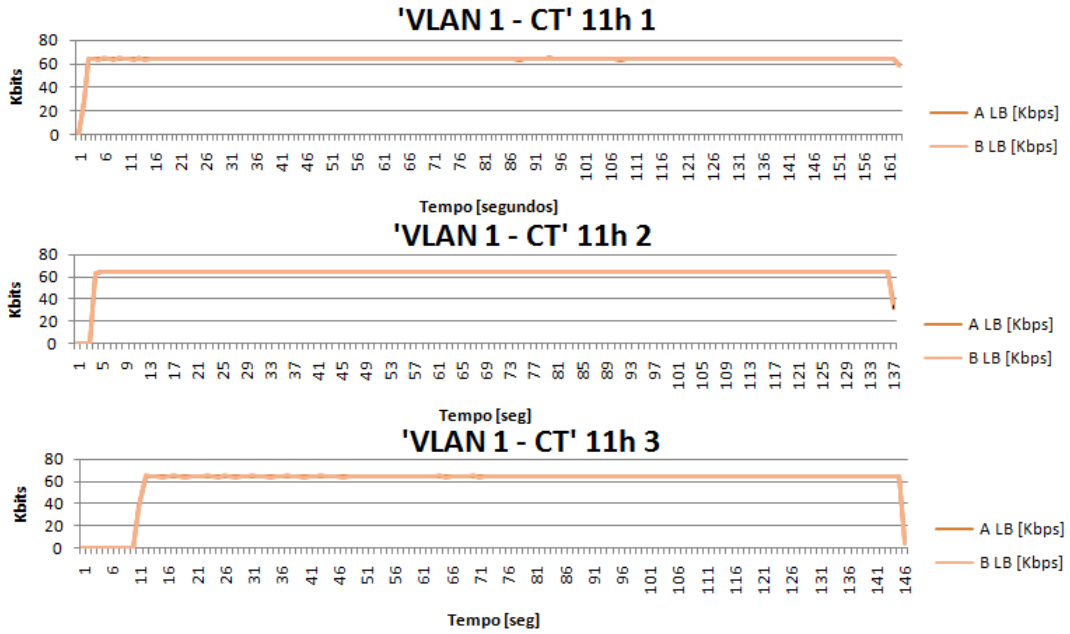

Figura 130 – Largura de banda ocupada nas chamadas efectuadas às 11 horas entre um telefone da VLAN 1 e um da central telefónica

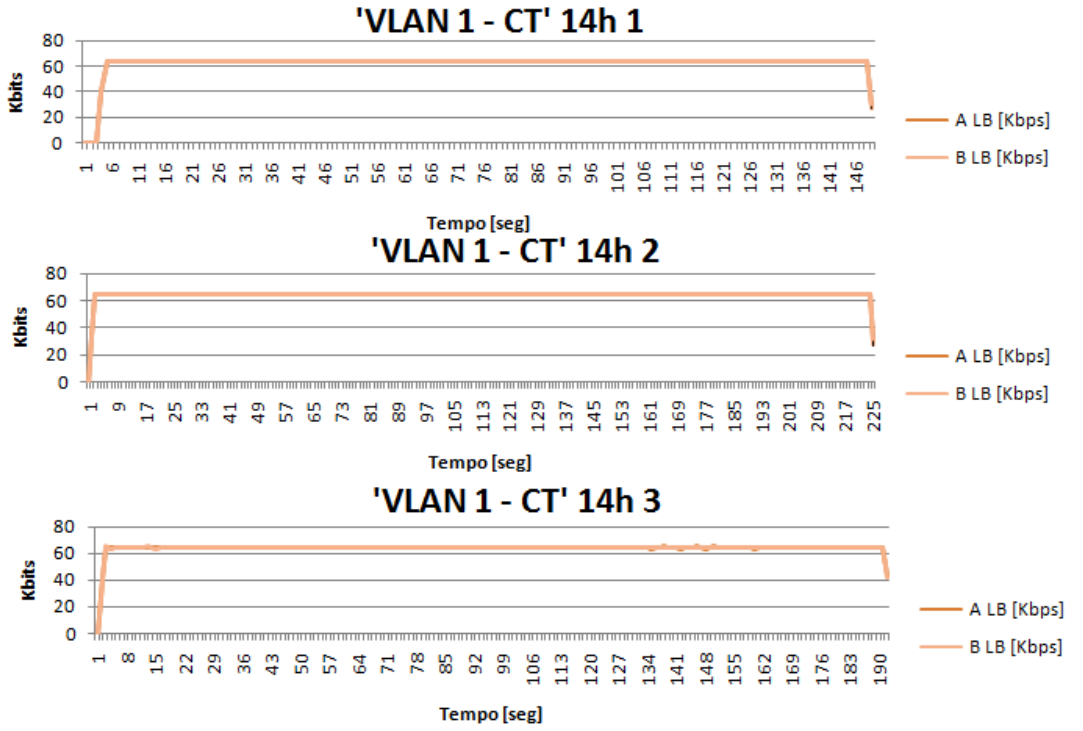

Figura 131 – Largura de banda ocupada nas chamadas efectuadas às 14 horas entre um telefone da VLAN 1 e um da central telefónica

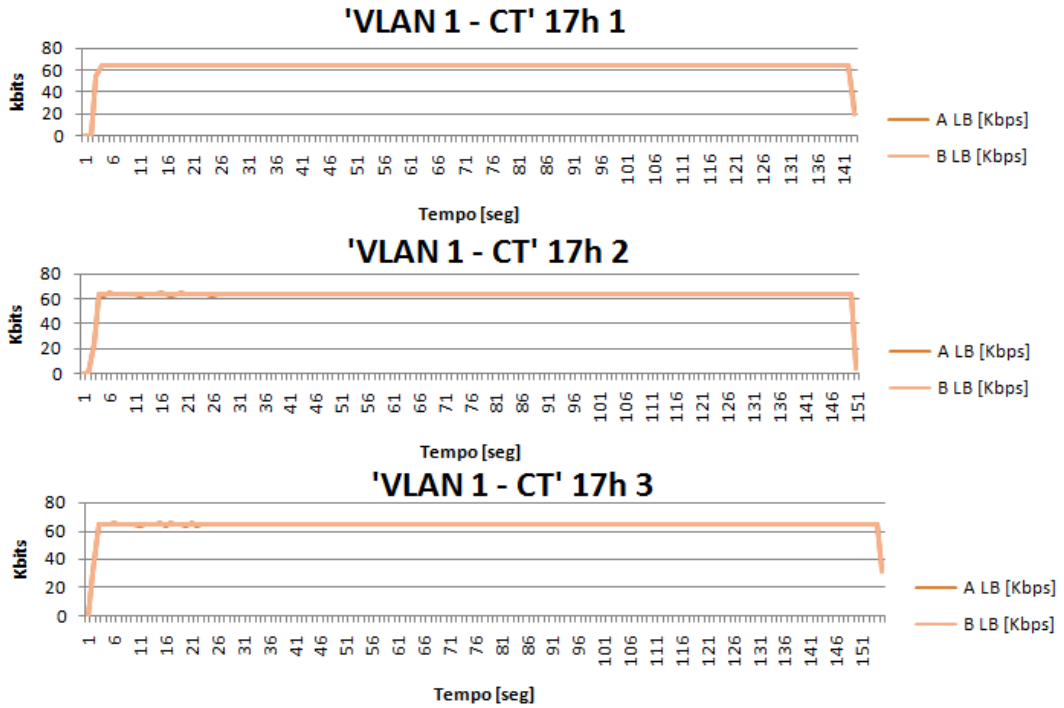

Figura 132 – Largura de banda ocupada nas chamadas efectuadas às 17 horas entre um telefone da VLAN 1 e um da central telefónica

Apêndice D - Emissão de vídeo

D.1 - Protocolos utilizados pelos terminais para comunicar com o *Content Server*

Na seguinte tabela, encontram-se os protocolos utilizados pelos diferentes equipamentos terminais para comunicar com o *Content Server* e assistir ao seminário em tempo real:

	Endereço IP do equipamento terminal	Protocolo utilizado para comunicar com o *Content Server*	Observações sobre o tipo se sessão utilizado para	
			Controlo	Dados
1	193.136.93.139	RTSP	TCP	TCP
2	193.136.87.34	MMS	TCP	TCP
3	193.136.93.78	RTSP	TCP	UDP
4	84.90.24.18	MMS	TCP	TCP
5	193.136.93.182	RTSP	TCP	UDP
6	193.136.92.42	RTSP	TCP	UDP
7	193.136.93.150	MMS	TCP	TCP
8	193.136.93.242	MMS	TCP	TCP
9	193.136.93.241	RTSP	TCP	UDP
10	193.136.92.144	MMS	TCP	UDP
11	213.58.227.228	RTSP	TCP	TCP
12	193.136.33.147	RTSP	TCP	UDP
13	193.137.154.38	RTSP	TCP	TCP
14	139.136.93.217	RTSP	TCP	UDP
15	193.136.93.250	MMS	TCP	TCP
16	193.136.93.225	RTSP	TCP	TCP
17	193.136.93.221	MMS	TCP	TCP

Tabela 35 – Protocolos utilizados pelos equipamentos terminais para asistir ao vídeo em tempo real

Apêndice E - DTMS-P2P

E.1 - Exemplo de configuração de um módulo de monitorização activa no DTMS-P2P

Neste caso o teste configurado é um *ping,* através do DTMS-P2P desde o 193.136.93.158 para o 10.0.0.73 com 5 repetições (*-c*).

Para facilitar a leitura, as opções escolhidas em cada secção estão a **negrito** e cor-de-laranja, por exemplo: **2**.

```
Distributed Traffic Measurement System with a Peer-to Peer Architecture

Choose which type of action you want to perform:

        I  -  Active Measurement:
                1  -  Ping
                2  -  Trace route
                3  -  J-OWAMP
                4  -  Choose another Monitoring Module

        II  -  Passive Measurement:
                5  -  TCPDUMP
                6  -  Choose another Monitoring Module

        III  -  Other options:
                7  -  Request list of supported monitoring modules
                8  -  Request monitoring module's help description
                9  -  Results search
                10 -  Replicate and/or delete a file
                11 -  Resources Request
                12 -  Refresh list of known measurement groups
                13 -  Get list of known nodes of all measurement groups
                14 -  Get light data file
                15 -  Connect to node request
                16 -  Get file list
                17 -  Finish

Note: At any time, press the "a" key to abort the configuration of a
given test session.
1

                Configuration of the node where the test measurement
should be processed

Within the available measurement groups, choose the measurement group
where you want to configure a given node:

        Option              Measurement Group ID         Number of Super-
probes
          1     -   00000000000000000000000000000001            1
          2     -   00000000000000000000000000000000            1
2

        Option              Configuration Type
          1     -   Choose the node you want to configure among all the
nodes of the 00000000000000000000000000000000 measurement group
```

```
(Default).
          2     -   Introduce a node address
1

Choose the node you want to configure:
          1 - Super-probe  193.136.93.103:22368
          2 - Probe        193.136.93.158:22368
2

                Configuration of the node to where the test measurement
should be processed

Within the available measurement groups, choose the measurement group
where you want to configure a given node:

        Option             Measurement Group ID         Number of Super-
probes
          1     -   00000000000000000000000000000001            1
          2     -   00000000000000000000000000000000            1
          3     -   Introduce the address of a machine not connected to
the DTMS-P2P network
1

        Option             Configuration Type
          1     -   Choose the node you want to configure among all the
nodes of the 00000000000000000000000000000001 measurement group
(Default).
          2     -   Introduce a node address
1

Choose the node you want to configure:
          1 - Super-probe  10.0.0.73:22368
1

Introduce the options you want to use for the ping 10.0.0.73 command to
be processed. Press Enter for no options.

The following restrictions must be respected:

Measurement Group                       Node Address
Monitoring Module              Restrictions

00000000000000000000000000000000    193.136.93.158:22368
ping            <mustUse>-c</mustUse><doNotUse>-t</doNotUse>
-c 5
The following command will be processed: ping 10.0.0.73 -c 5

Press any key to continue or "r" to re-enter the command to be processed:

Do you want the remote node to immediately process the command (y/n)
(Default y):

Do you want to wait for the response to the command request (y/n)
(Default y):

Waiting for the response to the command request. Press any key to
continue.

The command was successfully processed and the results were saved to
```

```
00000000000000000000000000000000_10.0.0.73.21164_00000000000000000000000000000000_193.136.93.158.22368_1208513961074_ping 10.0.0.73 -c 5.res file.
Press any key to continue.
```

E.2 - Exemplo de configuração de um módulo de monitorização passiva no DTMS-P2P

Neste exemplo mostra-se como proceder para configurar a execução do *TCPDump* na *super-probe* do grupo1 (10.0.0.73). Utilizando as opções:

-c 1000 -i eth1 -w cap4_pc1.cap

Para facilitar a leitura, as opções escolhidas em cada secção estão a **negrito** e cor-de-laranja, por exemplo: **2**.

```
Distributed Traffic Measurement System with a Peer-to-Peer Architecture

Choose which type of action you want to perform:

        I  -  Active Measurement:
                1  -  Ping
                2  -  Trace route
                3  -  J-OWAMP
                4  -  Choose another Monitoring Module

        II  -  Passive Measurement:
                5  -  TCPDUMP
                6  -  Choose another Monitoring Module

        III  -  Other options:
                7  -  Request list of supported monitoring modules
                8  -  Request monitoring module's help description
                9  -  Results search
                10 -  Replicate and/or delete a file
                11 -  Resources Request
                12 -  Refresh list of known measurement groups
                13 -  Get list of known nodes of all measurement groups
                14 -  Get light data file
                15 -  Connect to node request
                16 -  Get file list
                17 -  Finish

Note: At any time, press the "a" key to abort the configuration of a
given test session.
5

Choose the type of TCPDUMP command you want to perform:
        1  -  winDump (Usually Windows)
        2  -  tcpDump (Usually Linux)

Note: At any time, press the "a" key to abort the configuration of a
given test session.
2

Configuration of the node where you want to process the measurement.

Within the available measurement groups, choose the measurement group
```

```
where you want to configure a given node:

        Option              Measurement Group ID         Number of Super-
probes
          1     -   00000000000000000000000000000001            1
          2     -   00000000000000000000000000000000            1
1

        Option              Configuration Type
          1     -   Choose the node you want to configure among all the
nodes of the 00000000000000000000000000000001 measurement group
(Default).
          2     -   Introduce a node address
1

Choose the node you want to configure:
          1 - Super-probe  10.0.0.73:22368
1

Introduce the options you want to use for the tcpdump command to be
processed. Press Enter for no options.

The following restrictions must be respected:

Measurement Group                        Node Address
Monitoring Module               Restrictions

00000000000000000000000000000001    10.0.0.73:22368             tcpdump
<mustUse>-c</mustUse><doNotUse>-F; -l; -m; -r</doNotUse>
-c 1000 -i eth1 -w cap4_pc1.cap
The following command will be processed: tcpdump -c 1000 -i eth1 -w
cap4_pc1.cap

Press any key to continue or "r" to re-enter the command to be
processed:

Do you want the remote node to immediately process the command (y/n)
(Default y): n

Enter the time you want the command to be processed:
Introduce the year: 2008
Introduce the month: 04
Introduce the date: 18
Introduce the hour: 12
Introduce the minutes: 25
Introduce the seconds: 00


   Do you want to wait for the response to the command request (y/n)
(Default y): n
```

E.3 - Outras opções do DTMS-P2P

Neste apêndice mostra-se o funcionamento de algumas funções do DTMS-P2P.

Para facilitar a leitura, as opções escolhidas em cada secção estão a **negrito** e cor-de-laranja, por exemplo: **2**.

E.3.1 Opção 13 - *Get list of known nodes of all measurement groups*

```
Distributed Traffic Measurement System with a Peer-to-Peer Architecture

Choose which type of action you want to perform:

        I  -  Active Measurement:
                1  -  Ping
                2  -  Trace route
                3  -  J-OWAMP
                4  -  Choose another Monitoring Module

        II  -  Passive Measurement:
                5  -  TCPDUMP
                6  -  Choose another Monitoring Module

        III  -  Other options:
                7  -  Request list of supported monitoring modules
                8  -  Request monitoring module's help description
                9  -  Results search
                10 -  Replicate and/or delete a file
                11 -  Resources Request
                12 -  Refresh list of known measurement groups
                13 -  Get list of known nodes of all measurement groups
                14 -  Get light data file
                15 -  Connect to node request
                16 -  Get file list
                17 -  Finish

Note: At any time, press the "a" key to abort the configuration of a
given test session.
13

Nodes in the Measurement Group 00000000000000000000000000000001:

          1 - Super-probe  10.0.0.73:22368

Nodes in the Measurement Group 00000000000000000000000000000000:

          1 - Super-probe  193.136.93.103:22368
          2 - Probe        193.136.93.158:22368

Note: At any time, press the "a" key to abort the current action or any
other key to refresh the list of known nodes.
```

E.3.2 Opção 9 - *Results search*

Neste caso pesquisou-se no grupo 0 os resultados dos *pings* e descarregou-se um dos resultados apresentados.

```
Distributed Traffic Measurement System with a Peer-to-Peer Architecture

Choose which type of action you want to perform:

        I  -  Active Measurement:
                1  -  Ping
                2  -  Trace route
                3  -  J-OWAMP
                4  -  Choose another Monitoring Module
```

```
        II  -  Passive Measurement:
                5  -  TCPDUMP
                6  -  Choose another Monitoring Module

        III  -  Other options:
                7  -  Request list of supported monitoring modules
                8  -  Request monitoring module's help description
                9  -  Results search
                10 -  Replicate and/or delete a file
                11 -  Resources Request
                12 -  Refresh list of known measurement groups
                13 -  Get list of known nodes of all measurement groups
                14 -  Get light data file
                15 -  Connect to node request
                16 -  Get file list
                17 -  Finish

Note: At any time, press the "a" key to abort the configuration of a
given test session.
9

Within the available measurement groups, choose the measurement group
where you want to process the results search:

        Option             Measurement Group ID         Number of Super-
probes
          1     -   00000000000000000000000000000001            1
          2     -   00000000000000000000000000000000            1
          3     -   Global Search
2

Introduce the search criteria: ping

Choose the file you want to download:

        Option  -  File Name    File Size (in bytes)    # Nodes
Node IP         Node Port       Node Type       Speed   FlagUploadSpeed
FlagHaveUploaded        FlagBusy    FlagPush

          1     -
00000000000000000000000000000000_10.0.0.73.21164_0000000000000000000000000
00000000_193.136.93.158.22368_1208513961074_ping 10.0.0.73 -c 5.res
5081

193.136.93.103      22368   super-probe     0       0       1       0
0
          2     -
00000000000000000000000000000000_10.0.0.73.21164_0000000000000000000000000
00000000_193.136.93.103.22368_1208513845788_ping 193.136.93.158 -c 5.res
5321

193.136.93.103      22368   super-probe     0       0       1       0
0
          3     -
00000000000000000000000000000000_10.0.0.73.21164_0000000000000000000000000
00000000_193.136.93.158.22368_1208513979583_ping 193.136.93.103 -c 5.res
5321
```

```
193.136.93.103      22368   super-probe     0       0       1       0
0
          4     -
00000000000000000000000000000000_10.0.0.73.21164_00000000000000000000000000
00000000_193.136.93.103.22368_1208513781637_ping 10.0.0.73 -c 5.res
5081

193.136.93.103      22368   super-probe     0       0       1       0
0

Note: At any time, press the "a" key to return to the Main Menu.

Choose the file you want to download:

        Option  -  File Name    File Size (in bytes)    # Nodes
Node IP         Node Port       Node Type       Speed   FlagUploadSpeed
FlagHaveUploaded        FlagBusy    FlagPush

          1     -
00000000000000000000000000000000_10.0.0.73.21164_00000000000000000000000000
00000000_193.136.93.158.22368_1208513961074_ping 10.0.0.73 -c 5.res
5082

193.136.93.103      22368   super-probe     0       0       1       0
0

193.136.93.158      22368   probe   0       0       1       0       0
          2     -
00000000000000000000000000000000_10.0.0.73.21164_00000000000000000000000000
00000000_193.136.93.103.22368_1208513845788_ping 193.136.93.158 -c 5.res
5322

193.136.93.103      22368   super-probe     0       0       1       0
0

193.136.93.158      22368   probe   0       0       1       0       0
          3     -
00000000000000000000000000000000_10.0.0.73.21164_00000000000000000000000000
00000000_193.136.93.158.22368_1208513979583_ping 193.136.93.103 -c 5.res
5322

193.136.93.103      22368   super-probe     0       0       1       0
0

193.136.93.158      22368   probe   0       0       1       0       0
          4     -
00000000000000000000000000000000_10.0.0.73.21164_00000000000000000000000000
00000000_193.136.93.103.22368_1208513781637_ping 10.0.0.73 -c 5.res
5082

193.136.93.103      22368   super-probe     0       0       1       0
0

193.136.93.158      22368   probe   0       0       1       0       0

Note: At any time, press the "a" key to return to the Main Menu.
1

The client will download the
```

```
00000000000000000000000000000000_10.0.0.73.21164_00000000000000000000000000000000_193.136.93.158.22368_1208513961074_ping 10.0.0.73 -c 5.res file.

Choose the file you want to download:

        Option  -  File Name    File Size (in bytes)    # Nodes
Node IP         Node Port       Node Type       Speed   FlagUploadSpeed
FlagHaveUploaded        FlagBusy    FlagPush

          1     -
00000000000000000000000000000000_10.0.0.73.21164_00000000000000000000000000000000_193.136.93.158.22368_1208513961074_ping 10.0.0.73 -c 5.res
5082

193.136.93.103      22368   super-probe     0       0       1       0
0

193.136.93.158      22368   probe   0       0       1       0       0
          2     -
00000000000000000000000000000000_10.0.0.73.21164_00000000000000000000000000000000_193.136.93.103.22368_1208513845788_ping 193.136.93.158 -c 5.res
5322

193.136.93.103      22368   super-probe     0       0       1       0
0

193.136.93.158      22368   probe   0       0       1       0       0
          3     -
00000000000000000000000000000000_10.0.0.73.21164_00000000000000000000000000000000_193.136.93.158.22368_1208513979583_ping 193.136.93.103 -c 5.res
5322

193.136.93.103      22368   super-probe     0       0       1       0
0

193.136.93.158      22368   probe   0       0       1       0       0
          4     -
00000000000000000000000000000000_10.0.0.73.21164_00000000000000000000000000000000_193.136.93.103.22368_1208513781637_ping 10.0.0.73 -c 5.res
5082

193.136.93.103      22368   super-probe     0       0       1       0
0

193.136.93.158      22368   probe   0       0       1       0       0

Note: At any time, press the "a" key to return to the Main Menu.

The
00000000000000000000000000000000_10.0.0.73.21164_00000000000000000000000000000000_193.136.93.158.22368_1208513961074_ping 10.0.0.73 -c 5.res
download completed in 00:00:00 seconds and 231 milliseconds at a rate of
2.145 Kb/sec.
```

E.3.3 Opção 7 - *Request list of supported monitoring modules*

Neste caso consultaram-se quais os módulos suportados pela *super-probe* do grupo 1.

```
Distributed Traffic Measurement System with a Peer-to-Peer Architecture
```

```
Choose which type of action you want to perform:

        I  -  Active Measurement:
                1  -  Ping
                2  -  Trace route
                3  -  J-OWAMP
                4  -  Choose another Monitoring Module

        II  -  Passive Measurement:
                5  -  TCPDUMP
                6  -  Choose another Monitoring Module

        III  -  Other options:
                7  -  Request list of supported monitoring modules
                8  -  Request monitoring module's help description
                9  -  Results search
                10 -  Replicate and/or delete a file
                11 -  Resources Request
                12 -  Refresh list of known measurement groups
                13 -  Get list of known nodes of all measurement groups
                14 -  Get light data file
                15 -  Connect to node request
                16 -  Get file list
                17 -  Finish

Note: At any time, press the "a" key to abort the configuration of a
given test session.
7

Within the available measurement groups, choose the measurement group
where you want to configure a given node:

        Option             Measurement Group ID         Number of Super-
probes
          1     -   00000000000000000000000000000001            1
          2     -   00000000000000000000000000000000            1
1

        Option             Configuration Type
          1     -   Choose the node you want to configure among all the
nodes of the 00000000000000000000000000000001 measurement group
(Default).
          2     -   Introduce a node address
1

Choose the node you want to configure:
          1 - Super-probe  10.0.0.73:22368
1

The remote node 10.0.0.73:22368 supports the given monitoring modules:

        Monitoring Module                       Restrictions
OWAMP_ControlClient             <mustUse>senderPort 4181; receiverPort
22368; -P 21164-21174</mustUse> <doNotUse></doNotUse>

owping          <mustUse>senderPort 4181; receiverPort 22368; -P 21164-
21174</mustUse> <doNotUse></doNotUse>
```

```
TCPDump          <mustUse>-c</mustUse> <doNotUse>-F; -l; -m; -
r</doNotUse>

tracepath               <mustUse></mustUse> <doNotUse></doNotUse>

traceroute              <mustUse></mustUse> <doNotUse></doNotUse>

ping            <mustUse>-c</mustUse> <doNotUse>-t</doNotUse>

Press any key to continue
```

E.3.4 Opção 8 - *Request monitoring module's help description*

Recorrendo à opção 8, solicitou-se descrição do módulo de monitorização *TCPDump* da *super-probe* do grupo 1.

```
Distributed Traffic Measurement System with a Peer-to-Peer Architecture

Choose which type of action you want to perform:

        I  -  Active Measurement:
                1  -  Ping
                2  -  Trace route
                3  -  J-OWAMP
                4  -  Choose another Monitoring Module

        II  -  Passive Measurement:
                5  -  TCPDUMP
                6  -  Choose another Monitoring Module

        III  -  Other options:
                7  -  Request list of supported monitoring modules
                8  -  Request monitoring module's help description
                9  -  Results search
                10 -  Replicate and/or delete a file
                11 -  Resources Request
                12 -  Refresh list of known measurement groups
                13 -  Get list of known nodes of all measurement groups
                14 -  Get light data file
                15 -  Connect to node request
                16 -  Get file list
                17 -  Finish

Note: At any time, press the "a" key to abort the configuration of a
given test session.
8

Introduce the name of the monitoring module you want to use in this test
session: tcpdump

Within the available measurement groups, choose the measurement group
where you want to configure a given node:

        Option              Measurement Group ID         Number of Super-
probes
          1     -   00000000000000000000000000000001            1
          2     -   00000000000000000000000000000000            1
```

```
1

        Option              Configuration Type
          1     -   Choose the node you want to configure among all the
nodes of the 00000000000000000000000000000001 measurement group
(Default).
          2     -   Introduce a node address
1

Choose the node you want to configure:
          1 - Super-probe  10.0.0.73:22368
1

Help description received from node 10.0.0.73:22368 for the tcpdump
monitoring module:

tcpdump version 3.9.7
libpcap version 0.9.7
Usage: tcpdump [-aAdDeflLnNOpqRStuUvxX] [-c count] [ -C file_size ]
                [ -E algo:secret ] [ -F file ] [ -i interface ] [ -M
secret ]
                [ -r file ] [ -s snaplen ] [ -T type ] [ -w file ]
                [ -W filecount ] [ -y datalinktype ] [ -Z user ]
                [ expression ]

Press any key to continue
```

E.3.5 Opção 14 - *Get light data*

Mostra-se como descarregar para a pasta de *downloads* do cliente o *light data* dum nó, neste caso da *super-probe* do grupo 1.

```
Distributed Traffic Measurement System with a Peer-to-Peer Architecture

Choose which type of action you want to perform:

        I  -  Active Measurement:
                1  -  Ping
                2  -  Trace route
                3  -  J-OWAMP
                4  -  Choose another Monitoring Module

        II  -  Passive Measurement:
                5  -  TCPDUMP
                6  -  Choose another Monitoring Module

        III  -  Other options:
                7  -  Request list of supported monitoring modules
                8  -  Request monitoring module's help description
                9  -  Results search
                10 -  Replicate and/or delete a file
                11 -  Resources Request
                12 -  Refresh list of known measurement groups
                13 -  Get list of known nodes of all measurement groups
                14 -  Get light data file
                15 -  Connect to node request
                16 -  Get file list
```

```
                17 -  Finish

Note: At any time, press the "a" key to abort the configuration of a
given test session.
14

Within the available measurement groups, choose the measurement group
where you want to configure a given node:

        Option             Measurement Group ID         Number of Super-
probes
          1     -   00000000000000000000000000000001            1
          2     -   00000000000000000000000000000000            1
1

        Option             Configuration Type
          1     -   Choose the node you want to configure among all the
nodes of the 00000000000000000000000000000001 measurement group
(Default).
          2     -   Introduce a node address
1

Choose the node you want to configure:
          1 - Super-probe  10.0.0.73:22368
1

The LightData.xml download completed in 00:00:00 seconds and 204
milliseconds at a rate of 3.541 Kb/sec.
The compiled light data file of the super-probe 10.0.0.73:22368 was
succesfully downloaded and it was stored at the downloads directory as
10.0.0.73.22368_LightData_1208515274287.xml.

Press any key to continue.
```

E.3.6 Opção 16 - *Get file list*

Mostra-se como descarregar para a pasta de *downloads* do cliente o *file list* dum nó, neste caso da *super-probe* do grupo 0.

```
Distributed Traffic Measurement System with a Peer-to-Peer Architecture

Choose which type of action you want to perform:

        I  -  Active Measurement:
                1  -  Ping
                2  -  Trace route
                3  -  J-OWAMP
                4  -  Choose another Monitoring Module

        II  -  Passive Measurement:
                5  -  TCPDUMP
                6  -  Choose another Monitoring Module

        III  -  Other options:
                7  -  Request list of supported monitoring modules
                8  -  Request monitoring module's help description
                9  -  Results search
                10 -  Replicate and/or delete a file
```

```
                11 -  Resources Request
                12 -  Refresh list of known measurement groups
                13 -  Get list of known nodes of all measurement groups
                14 -  Get light data file
                15 -  Connect to node request
                16 -  Get file list
                17 -  Finish

Note: At any time, press the "a" key to abort the configuration of a
given test session.
16

Within the available measurement groups, choose the measurement group
where you want to configure a given node:

        Option             Measurement Group ID         Number of Super-
probes
          1     -   00000000000000000000000000000001            1
          2     -   00000000000000000000000000000000            1
2

        Option             Configuration Type
          1     -   Choose the node you want to configure among all the
nodes of the 00000000000000000000000000000000 measurement group
(Default).
          2     -   Introduce a node address
1

Choose the node you want to configure:
          1 - Super-probe  193.136.93.103:22368
          2 - Probe        193.136.93.158:22368
1

The FileList.xml download completed in 00:00:00 seconds and 191
milliseconds at a rate of 6.42 Kb/sec.
The file list of the node 193.136.93.103:22368 was succesfully
downloaded and it was stored at the downloads directory as
193.136.93.103.22368_FileList.xml.


Press any key to continue.
```

E.3.7 Opção 11 - *Resources Request*

Mostra os recursos de um determinado nó, neste caso a informação requerida é da *probe* do grupo 0.

```
Distributed Traffic Measurement System with a Peer-to-Peer Architecture

Choose which type of action you want to perform:

        I  -  Active Measurement:
                1  -  Ping
                2  -  Trace route
                3  -  J-OWAMP
                4  -  Choose another Monitoring Module

        II  -  Passive Measurement:
```

```
                5  -  TCPDUMP
                6  -  Choose another Monitoring Module

        III  -  Other options:
                7  -  Request list of supported monitoring modules
                8  -  Request monitoring module's help description
                9  -  Results search
                10 -  Replicate and/or delete a file
                11 -  Resources Request
                12 -  Refresh list of known measurement groups
                13 -  Get list of known nodes of all measurement groups
                14 -  Get light data file
                15 -  Connect to node request
                16 -  Get file list
                17 -  Finish

Note: At any time, press the "a" key to abort the configuration of a
given test session.
11

Configuration of the node you want to request for the available
resources.

Within the available measurement groups, choose the measurement group
where you want to configure a given node:

        Option             Measurement Group ID         Number of Super-
probes
          1     -   00000000000000000000000000000001            1
          2     -   00000000000000000000000000000000            1
2

        Option             Configuration Type
          1     -   Choose the node you want to configure among all the
nodes of the 00000000000000000000000000000000 measurement group
(Default).
          2     -   Introduce a node address
1

Choose the node you want to configure:
          1 - Super-probe  193.136.93.103:22368
          2 - Probe        193.136.93.158:22368
2

If the node is a super-probe, do you want to receive the information
about the available resources at the probes under its control (y/n)
(Default y):

Waiting for the response to the resources request. Press any key to
continue.

The Resources message with the requested information was received from
193.136.93.158:22368. Received information:

Maximum number of connections allowed, if the node is in super-probe
mode: 65535
Maximum number of connections to probes allowed, if the node is in
super-probe mode: 65535
Current number of connections to super-probes of its measurement group:
```

```
1
Current number of connections to probes: 0
Current number of connections to clients: 0
Current number of connections to super-probes of other measurement
groups: 0

Number of Resources Information received: 1

Resources Information 1
Flag Upload Speed: 0
Flag Download Speed: 0
Flag Push: 0
Authentication Mode: 7
Node Addresses:
IPVN: 4
Port: 22368
IP Address: /193.136.93.158
Free Memory: 62 Mbytes
Occupied Memory: 1 Mbytes
Free storage space: 49229 Mbytes
Average available bandwidth upstream: 0 kbps
Average available bandwidth downstream: 0 kbps
Press any key to continue.
```

E.4 - Configurando o *oping* no DTMS-P2P

Neste exemplo configurou-se o *oping* no PC 2 (193.136.93.103), para este executar *ping* para os outros dois computadores (193.136.93.161 e 10.0.0.73), com cinco repetições (-c 5)

Para facilitar a leitura, as opções escolhidas em cada secção estão a **negrito** e cor-de-laranja, por exemplo: **2**.

```
Distributed Traffic Measurement System with a Peer-to-Peer Architecture

Choose which type of action you want to perform:

        I  -  Active Measurement:
                1  -  Ping
                2  -  Trace route
                3  -  J-OWAMP
                4  -  Choose another Monitoring Module

        II  -  Passive Measurement:
                5  -  TCPDUMP
                6  -  Choose another Monitoring Module

        III  -  Other options:
                7  -  Request list of supported monitoring modules
                8  -  Request monitoring module's help description
                9  -  Results search
                10 -  Replicate and/or delete a file
                11 -  Resources Request
                12 -  Refresh list of known measurement groups
                13 -  Get list of known nodes of all measurement groups
                14 -  Get light data file
                15 -  Connect to node request
                16 -  Get file list
                17 -  Finish
```

```
Note: At any time, press the "a" key to abort the configuration of a
given test session.
6

Configuration of the node where you want to process the measurement.

Introduce the name of the monitoring module you want to use in this test
session: oping

Within the available measurement groups, choose the measurement group
where you want to configure a given node:

        Option             Measurement Group ID         Number of Super-
probes
          1     -   00000000000000000000000000000001            1
          2     -   00000000000000000000000000000000            1
2

        Option             Configuration Type
          1     -   Choose the node you want to configure among all the
nodes of the 00000000000000000000000000000000 measurement group
(Default).
          2     -   Introduce a node address
1

Choose the node you want to configure:
          1 - Super-probe  193.136.93.103:22368
          2 - Probe        193.136.93.161:22368
1

Introduce the options you want to use for the oping command to be
processed. Press Enter for no options.

The following restrictions must be respected:

Measurement Group                       Node Address
Monitoring Module              Restrictions

00000000000000000000000000000000    193.136.93.103:22368
oping            <mustUse>-c</mustUse> <doNotUse></doNotUse>

-c 5 193.136.93.161 10.0.0.73
The following command will be processed: oping -c 5 193.136.93.161
10.0.0.73

Press any key to continue or "r" to re-enter the command to be
processed:

Do you want the remote node to immediately process the command (y/n)
(Default y):


Do you want to wait for the response to the command request (y/n)
(Default y): n
```

ANEXOS

Anexo I - SCCP

I.1 - Tipos de mensagens

Na seguinte tabela encontram-se os tipos de mensagens trocadas no protocolo SCCP e os respectivos códigos.

Code	*Station Message ID Message*
0x0000	*Keep Alive Message*
0x0001	*Station Register Message*
0x0002	*Station IP Port Message*
0x0003	*Station Key Pad Button Message*
0x0004	*Station Enbloc Call Message*
0x0005	*Station Stimulus Message*
0x0006	*Station Off Hook Message*
0x0007	*Station On Hook Message*
0x0008	*Station Hook Flash Message*
0x0009	*Station Forward Status Request Message*
0x11	*Station Media Port List Message*
0x000A	*Station Speed Dial Status Request Message*
0x000B	*Station Line Status Request Message*
0x000C	*Station Configuration Status Request Message*
0x000D	*Station Time Date Request Message*
0x000E	*Station Button Template Request Message*
0x000F	*Station Version Request Message*
0x0010	*Station Capabilities Response Message*
0x0012	*Station Server Request Message*
0x0020	*Station Alarm Message*
0x0021	*Station Multicast Media Reception Ack Message*
0x0024	*Station Off Hook With Calling Party Number Message*
0x22	*Station Open Receive Channel Ack Message*
0x23	*Station Connection Statistics Response Message*
0x25	*Station Soft Key Template Request Message*
0x26	*Station Soft Key Set Request Message*
0x27	*Station Soft Key Event Message*
0x28	*Station Unregister Message*
0x0081	*Station Keep Alive Message*
0x0082	*Station Start Tone Message*
0x0083	*Station Stop Tone Message*
0x0085	*Station Set Ringer Message*
0x0086	*Station Set Lamp Message*
0x0087	*Station Set Hook Flash Detect Message*
0x0088	*Station Set Speaker Mode Message*
0x0089	*Station Set Microphone Mode Message*
0x008A	*Station Start Media Transmission*

0x008B	*Station Stop Media Transmission*
0x008F	*Station Call Information Message*
0x009D	*Station Register Reject Message*
0x009F	*Station Reset Message*
0x0090	*Station Forward Status Message*
0x0091	*Station Speed Dial Status Message*
0x0092	*Station Line Status Message*
0x0093	*Station Configuration Status Message*
0x0094	*Station Define Time & Date Message*
0x0095	*Station Start Session Transmission Message*
0x0096	*Station Stop Session Transmission Message*
0x0097	*Station Button Template Message*
0x0098	*Station Version Message*
0x0099	*Station Display Text Message*
0x009A	*Station Clear Display Message*
0x009B	*Station Capabilities Request Message*
0x009C	*Station Enunciator Command Message*
0x009E	*Station Server Respond Message*
0x0101	*Station Start Multicast Media Reception Message*
0x0102	*Station Start Multicast Media Transmission Message*
0x0103	*Station Stop Multicast Media Reception Message*
0x0104	*Station Stop Multicast Media Transmission Message*
0x105	*Station Open Receive Channel Message*
0x0106	*Station Close Receive Channel Message*
0x107	*Station Connection Statistics Request Message*
0x0108	*Station Soft Key Template Respond Message*
0x109	*Station Soft Key Set Respond Message*
0x0110	*Station Select Soft Keys Message*
0x0111	*Station Call State Message*
0x0112	*Station Display Prompt Message*
0x0113	*Station Clear Prompt Message*
0x0114	*Station Display Notify Message*
0x0115	*Station Clear Notify Message*
0x0116	*Station Activate Call Plane Message*
0x0117	*Station Deactivate Call Plane Message*
0x0118	*Station Unregister Ack Message*

Tabela 36 – Tipos de mensagens do protocolo SCCP [Jav0?]

Anexo II - MGCP

II.1 - Códigos de resposta do protocolo MGCP versão 0.1 e 1.0

A seguinte tabela contém os códigos de resposta do protocolo MGCP das versões 0.1 e 1.0, no caso de estudo desta dissertação a versão utilizada é a 0.1.

Table 1 MGCP Return Codes and Descriptions

1.0 Return Code	0.1 Return Code	Description
000	NA	Response acknowledgement.
100	NA	Transaction is being executed. Completion response will follow.
101	NA	Transaction has been queued.
200	200	Transaction was executed normally.
250	250	Connection was already deleted.
400	400	Transaction not executed, transient error.
401	401	Phone is already off hook.
402	402	Phone is already on-hook.
403	400	Endpoint does not have sufficient resources.
404	400	Insufficient Bandwidth.
405	400	Endpoint is restarting.
406	400	Transaction timeout.
407	400	Transaction aborted.
409	400	Internal overload.
410	400	Endpoint not available.
500	500	Endpoint unknown.
501	501	Endpoint is not ready.
502	502	Endpoint does not have sufficient resources.
503	502	All of wildcard is too complicated.
504	510	Unknown or unsupported command.
505	510	Unknown remote connection descriptor.
506	510	Unable to satisfy both local connection option and remote connection descriptor.
507	510	Unsupported functionality.
508	510	Unknown quarantine handling.
509	510	SDP Error.
510	510	Protocol error.
511	511	Unrecognized extension.
512	512	Gateway not equipped to detect events.
513	513	Gateway not equipped to generate signal.
514	514	Transaction could not be executed because the gateway cannot send the specified announcement.
515	515	Invalid connection ID.
516	516	Unknown Call ID.
517	517	Unsupported/Invalid mode.

Figura 133 – Códigos de resposta do protocolo MGCP (Parte I) [Cisco05]

Table 1 MGCP Return Codes and Descriptions (continued)

1.0 Return Code	0.1 Return Code	Description
518	518	Unsupported/Invalid package.
519	519	Endpoint does not have a digit map.
520	520	Endpoint restarting.
521	NA	Endpoint redirected to another call agent.
522	510	No such signal or event.
523	510	Unknown action or illegal combination of actions.
524	510	Internal inconsistency in LocalConnectionOptions (LCO).
525	510	Unknown extension in LCO.
526	502	Insufficient bandwidth.
527	510	Missing RemoteConnectionDescriptor.
528	510	Incompatible protocol version.
529	501	Hardware failure.
530	501	CAS signaling protocol error.
531	501	Failure of a grouping of trunks (facility error).
532	510	Unsupported values in LCO.
533	502	Insufficient bandwidth. Response too large.
534	502	Codec negotiation failure.
535	510	Packetization period not supported.
536	510	Unsupported RestartMethod.
537	510	Unknown or unsupported digit map extension, since the gateway does not have the digit map.
538	512 or 513	Event/Signal parameter error.
540	515	Per endpoint connection limit was exceeded.
596	596	VISM-specific return code indicating VCC failure or VCC could not be set up.
598	598	Media connection failure.
599	599	VISM-specific return code indicating media connection loss.

Figura 134 – Códigos de resposta do protocolo MGCP (Parte II) [Cisco05]

Anexo III - RTP

III.1 - Tipos de *payload*

Os tipos de *payload* utilizados pelo RTP são os seguintes:

PT	*Encoding name*	Áudio [A] / Vídeo [V]	*Clock rate* [Hz]
0	PCMU	A	8000
1	1016	A	8000
2	G721	A	8000
3	GSM	A	8000
4	*Unassigned*	A	8000
5	DVI4	A	8000
6	DVI4	A	16000
7	LPC	A	8000
8	PCMA	A	8000
9	G722	A	8000
10	L16	A	44100
11	L16	A	44100
12	*Unassigned*	A	
13	*Unassigned*	A	
14	MPA	A	90000
15	G728	A	8000
16 – 23	*Unassigned*	A	
24	*Unassigned*	V	
25	CelB	V	90000
26	JPEG	V	90000
27	*Unassigned*	V	
28	Nv	V	90000
29	*Unassigned*	V	
30	*Unassigned*	V	
31	H261	V	90000
32	MPV	V	90000
33	MP2T	AV	90000
34 – 71	*Unassigned*	?	
72 – 76	*Reserved*	N/A	N/A
77 – 95	*Unassigned*	?	
96 – 127	*Dynamic*	?	

Tabela 37 – Tipos de *payload* [Sch96]

Anexo IV - RTSP

IV.1 - Comandos utilizados nas mensagens de pedido

A seguinte tabela contém os comandos utilizados nas mensagens de pedido do protocolo RTSP:

Comandos	Origem/Destino	Requisitos	Descrição
DESCRIBE	Cliente/Servidor	Recomendando	Este método recupera a descrição de uma apresentação ou objecto multimédia identificado através de um URL pedido a um servidor. O servidor responde com uma descrição do recurso solicitado.
ANNOUNCE	Cliente/Servidor	Opcional	Mostra a descrição de uma apresentação ou um objecto multimédia identificado através de um URL pedido a um servidor.
	Servidor/Cliente		Anunciam actualizações da descrição da sessão em tempo real.
GET_PARAMETER	Cliente/Servidor	Opcional	Recupera o valor de um parâmetro de uma apresentação ou fluxo especificado num URI.
	Servidor/Cliente		
OPTIONS	Cliente/Servidor	Necessário	Pode ser emitido em qualquer momento. Não influencia o estado actual.
	Servidor/Cliente	Opcional	
PAUSE	Cliente/Servidor	Recomendado	Faz com que a transmissão do fluxo seja interrompida temporariamente.
PLAY	Cliente/Servidor	Necessário	Indica ao servidor que pode começar a enviar os dados segundo os métodos definidos no *SETUP*.
RECORD	Cliente/Servidor	Opcional	Este comando inicia a gravação de um conjunto de dados multimédia de acordo com a descrição da apresentação.
REDIRECT	Servidor/Cliente	Opcional	Indica ao cliente que deve conectar-se a outra localização do servidor.

SETUP	Cliente/Servidor	Necessário	Especifica como será transportado o fluxo de dados.
SET_PARAMETER	Cliente/Servidor	Opcional	Utilizado para definir o valor de um parâmetro de uma apresentação ou fluxo especificado num URI.
	Servidor/Cliente		
TEARDOWN	Cliente/Servidor	Necessário	Pára a transmissão dos dados de um determinado URI e liberta os recursos associados à esta transmissão.

Tabela 38 – Comandos do RTSP [SRL98]

IV.2 - Códigos de estado

Na seguinte tabela encontram-se os códigos de estado utilizados nas mensagens de resposta do RTSP:

Código	Descrição	Código	Descrição
100	*Continue*	**414**	*Request-URI Too Long*
200	*OK*	**415**	*Unsupported Media Type*
201	*Created*	**451**	*Invalid parameter*
250	*Low on Storage Space*	**452**	*Illegal Conference Identifier*
300	*Multiple Choices*	**453**	*Not Enough Bandwidth*
301	*Moved Permanently*	**454**	*Session Not Found*
302	*Moved Temporarily*	**455**	*Method Not Valid In This State*
303	*See Other*	**456**	*Header Field Not Valid*
305	*Use Proxy*	**457**	*Invalid Range*
400	*Bad Request*	**458**	*Parameter Is Read-Only*
401	*Unauthorized*	**459**	*Aggregate Operation Not Allowed*
402	*Payment Required*	**460**	*Only Aggregate Operation Allowed*
403	*Forbidden*	**461**	*Unsupported Transport*
404	*Not Found*	**462**	*Destination Unreachable*
405	*Method Not Allowed*	**500**	*Internal Server Error*
406	*Not Acceptable*	**501**	*Not Implemented*
407	*Proxy Authentication Required*	**502**	*Bad Gateway*
408	*Request Timeout*	**503**	*Service Unavailable*
410	*Gone*	**504**	*Gateway Timeout*
411	*Length Required*	**505**	*RTSP Version Not Supported*
412	*Precondition Failed*	**551**	*Option not support*
413	*Request Entity Too Large*		

Tabela 39 – Códigos de estado utilizados nas mensagens de resposta do protocolo RTSP [SRL98]

IV.3 - Diagramas de estado do cliente e do servidor do protocolo RTSP

Os seguintes diagramas de estado foram elaborados recorrendo à informação da RFC 2326.

Diagrama de estados do cliente:

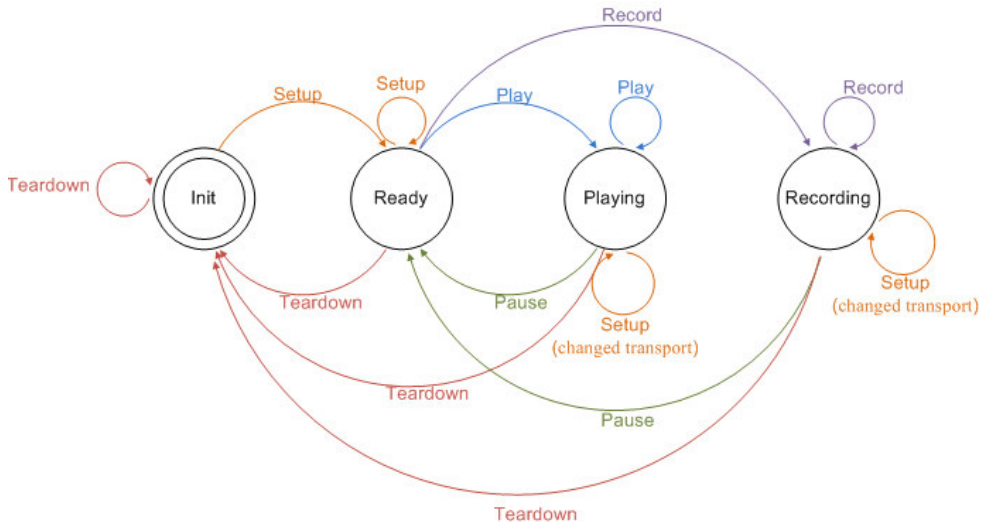

Figura 135 – Diagrama de estados do cliente do protocolo RTSP [SRL98]

Diagrama de estados do servidor:

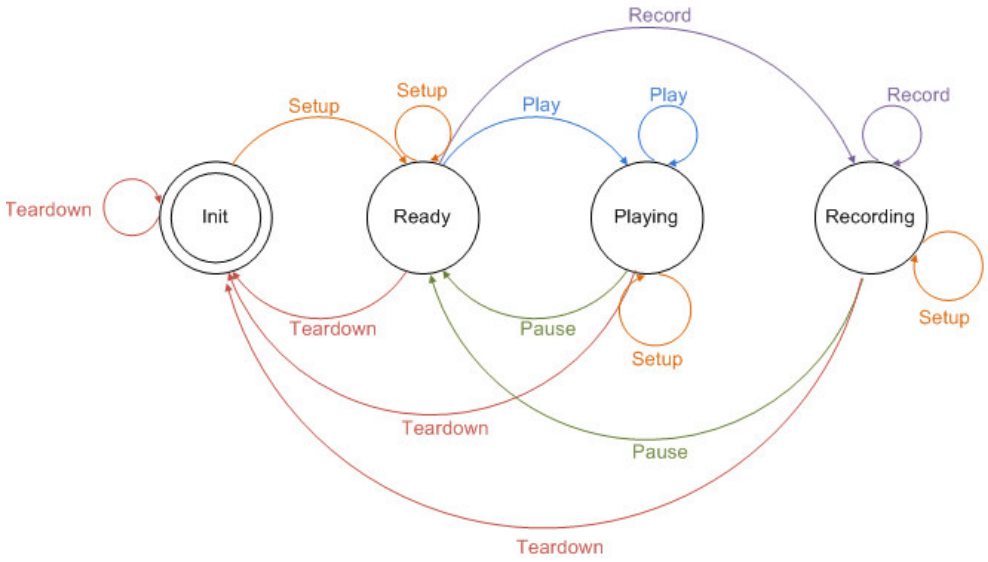

Figura 136 – Diagrama de estados do servidor do protocolo RTSP [SRL98]

Anexo V - MMS

V.1 - Tipos de mensagens do protocolo MMS

Na seguinte tabela encontram-se os tipos de mensagens utilizadas pelo MMS, assim como o sentido em que são enviadas e uma pequena descrição das mesmas.

Tipos de mensagens	Origem/ Destino	Descrição
LinkMacToViewerPing	Servidor/ Cliente	Enviado para verificar se um determinado cliente está activo.
LinkMacToViewerReportConnectedEX	Servidor/ Cliente	Enviada em resposta à mensagem *LinkViewerToMacConnect*.
LinkMacToViewerReportConnectedFunnel	Servidor/ Cliente	Enviada em resposta à mensagem *LinkViewerToMacConnectFunnel*
LinkMacToViewerReportDisconnectedFunnel	Servidor/ Cliente	Enviada em resposta à mensagem *LinkViewerToMacConnectFunnel*, quando ocorre um erro no processamento do pedido.
LinkMacToViewerReportEndOfStream	Servidor/ Cliente	Através desta mensagem o servidor informa ao cliente que foi atingido o fim da lista (*playlist*).
LinkMacToViewerReportFunnelInfo	Servidor/ Cliente	Enviada em resposta à mensagem *LinkViewerToMacFunnelInfo*, através desta o servidor atribui um identificador único ao cliente. Este identificador é utilizado pelo cliente nos pacotes *RequestPacketListResend*.
LinkMacToViewerReportOpenFile	Servidor/ Cliente	Enviada em resposta à mensagem *LinkViewerToMacOpenFile*.
LinkMacToViewerReportReadBlock	Servidor/ Cliente	Enviada em resposta à mensagem *LinkViewerToMacReadBlock*.
LinkMacToViewerReportRedirect	Servidor/ Cliente	Resposta da mensagem *LinkViewerToMacOpenFile*. É utilizada para redireccionar o cliente para uma nova localização.
LinkMacToViewerReportStartedPlaying	Servidor/ Cliente	Enviada em resposta à mensagem *LinkViewerToMacStartPlaying*.
LinkMacToViewerReportStartStriding	Servidor/ Cliente	Enviada em resposta à mensagem *LinkViewerToMacStartStriding*.
LinkMacToViewerReportStreamChange	Servidor/ Cliente	Notifica o cliente que está a iniciar-se o fluxo da próxima entrada na *playlist* do servidor.
LinkMacToViewerReportStreamSwitch	Servidor/ Cliente	Enviada em resposta à mensagem *LinkViewerToMacStreamSwitch*.
LinkMacToViewerSecurityChallenge	Servidor/ Cliente	Solicita a autenticação do cliente.
LinkViewerToMacCancelReadBlock	Cliente/ Servidor	Enviada para solicitar ao servidor o cancelamento do envido de cabeçalho do ficheiro ASF, o qual

		tinha sido solicitado anteriormente pelo cliente, através da mensagem *LinkViewerToMacReadBlock*.
LinkViewerToMacCloseFile	Cliente/ Servidor	Informa ao servidor que a sessão multimédia finalizou.
LinkViewerToMacConnect	Cliente/ Servidor	Através desta mensagem o cliente solicita uma ligação ao servidor.
LinkViewerToMacConnectFunnel	Cliente/ Servidor	Solicita que os pacotes de informação sejam enviados pelo servidor utilizando um protocolo específico e para uma porta específica.
LinkViewerToMacFunnelInfo	Cliente/ Servidor	Solicita um identificador par utilizar no *RequestPacketResend packets*.
LinkViewerToMacLogging	Cliente/ Servidor	O cliente utiliza esta mensagem para enviar ao servidor estatísticas sobre o fluxo actual.
LinkViewerToMacOpenFile	Cliente/ Servidor	Especifica o nome do recurso no servidor, a ser enviado.
LinkViewerToMacPong	Cliente/ Servidor	Esta mensagem é enviada em resposta à *LinkMacToViewerPing*. É através desta que o cliente informa ao servidor que ainda se encontra activo.
LinkViewerToMacReadBlock	Cliente/ Servidor	O cliente envia esta mensagem para solicitar o cabeçalho do ficheiro ASF. O cabeçalho do ficheiro ASF contém todos os parâmetros de iniciação do *codec*, necessários para o cliente descodificar a informação multimédia.
LinkViewerToMacSecurityResponse	Cliente/ Servidor	Esta mensagem é enviada pelo cliente em resposta da autenticação requerida pelo servidor.
LinkViewerToMacStartPlaying	Cliente/ Servidor	Enviada pelo cliente para solicitar ao servidor que inicie o fluxo de conteúdo para *playback* a uma taxa normal (tempo-real).
LinkViewerToMacStartStriding	Cliente/ Servidor	Enviada para pedir ao servidor que inicie o fluxo de contudo para *playback* em velocidade acelerada, podendo esta ser para a frente ou para trás.
LinkViewerToMacStopPlaying	Cliente/ Servidor	Esta mensagem é enviada pelo cliente para pedir ao servidor que pare o fluxo de pacotes de dados.
LinkViewerToMacStreamSwitch	Cliente/ Servidor	Enviada para seleccionar ou desseleccionar, ou substituir o fluxo individual que está a ser recebido actualmente.

Tabela 40 – Tipos de mensagens do protocolo MMS [Microsoft08]

Anexo VI - Rede VoIP

VI.1 - Protocolos suportados pelo Cisco Unified IP Phone 7961G/7961G-GE e 7941G/7941G-GE

Esta tabela mostra todos os protocolos suportados por estes equipamentos, corresponde à tabela 1-1 (*Table 1-1 Supported Networking Protocols on the Cisco Unified IP Phone*) do manual *Cisco Unified IP Phone Administration Guide for Cisco Unified CallManager 5.1 (SCCP).*

Networking Protocol	Purpose	Usage Notes
Bootstrap Protocol (BootP)	BootP enables a network device such as the Cisco Unified IP Phone to discover certain startup information, such as its IP address.	If you are using BootP to assign IP addresses to the Cisco Unified IP Phone, the BOOTP Server option shows "Yes" in the network configuration settings on the phone.
Cisco Discovery Protocol (CDP)	CDP is a device-discovery protocol that runs on all Cisco-manufactured equipment. Using CDP, a device can advertise its existence to other devices and receive information about other devices in the network.	The Cisco Unified IP Phone uses CDP to communicate information such as auxiliary VLAN ID, per port power management details, and Quality of Service (QoS) configuration information with the Cisco Catalyst switch.
Dynamic Host Configuration Protocol (DHCP)	DHCP dynamically allocates and assigns an IP address to network devices. DHCP enables you to connect an IP phone into the network and have the phone become operational without you needing to manually assign an IP address or to configure additional network parameters.	DHCP is enabled by default. If disabled, you must manually configure the IP address, subnet mask, gateway, and a TFTP server on each phone locally. Cisco recommends that you use DHCP custom option 150. With this method, you configure the TFTP server IP address as the option value. For additional supported DCHP configurations, refer to *Cisco Unified CallManager System Guide.*

Figura 137 – Protocolos suportados pelos telefones (Parte I) [Cisco0?a]

Networking Protocol	Purpose	Usage Notes
Hypertext Transfer Protocol (HTTP)	HTTP is the standard way of transferring information and moving documents across the Internet and the web.	Cisco Unified IP Phones use HTTP for the XML services and for troubleshooting purposes.
IEEE 802.1X	The IEEE 802.1X standard defines a client-server-based access control and authentication protocol that restricts unauthorized clients from connecting to a LAN through publicly accessible ports. Until the client is authenticated, 802.1X access control allows only Extensible Authentication Protocol over LAN (EAPOL) traffic through the port to which the client is connected. After authentication is successful, normal traffic can pass through the port.	The Cisco Unified IP Phone implements the IEEE 802.1X standard by providing support for the EAP-MD5 option for 802.1X authentication. When 802.1X authentication is enabled on the phone, you should disable the PC port and voice VLAN. Refer to the "Supporting 802.1X Authentication on Cisco Unified IP Phones" section on page 1-18 for additional information.
Internet Protocol (IP)	IP is a messaging protocol that addresses and sends packets across the network.	To communicate using IP, network devices must have an assigned IP address, subnet, and gateway. IP addresses, subnets, and gateways identifications are automatically assigned if you are using the Cisco Unified IP Phone with Dynamic Host Configuration Protocol (DHCP). If you are not using DHCP, you must manually assign these properties to each phone locally.
Real-Time Transport Protocol (RTP)	RTP is a standard protocol for transporting real-time data, such as interactive voice and video, over data networks.	Cisco Unified IP Phones use the RTP protocol to send and receive real-time voice traffic from other phones and gateways.

Figura 138 – Protocolos suportados pelos telefones (Parte II) [Cisco0?a]

Networking Protocol	Purpose	Usage Notes
Real-Time Control Protocol (RTCP)	RTCP works in conjunction with RTP to provide QoS data (such as jitter, latency, and round trip delay) on RTP streams.	RTCP is disabled by default, but you can enable it on a per phone basis using Cisco Unified CallManager. For more information, see the "Network Configuration Menu" section on page 4-27.
Skinny Client Control Protocol (SCCP)	SCCP includes a messaging set that allows communications between call control servers and endpoint clients such as IP Phones. SCCP is proprietary to Cisco Systems.	Cisco Unified IP Phones use SCCP for call control. You can configure the Cisco Unified IP Phone to use either SCCP or Session Initiation Protocol (SIP).
Session Initiation Protocol (SIP)	SIP is an emerging standard for setting up telephone calls, multimedia conferencing, and other types of communications on the Internet.	Cisco Unified IP Phones use SCCP for call control. You can configure the Cisco Unified IP Phone to use either SCCP or SIP.
Transmission Control Protocol (TCP)	TCP is a connection-oriented transport protocol.	Cisco Unified IP Phones use TCP to connect to Cisco Unified CallManager and to access XML services.
Transport Layer Security (TLS)	TLS is a standard protocol for securing and authenticating communications.	When security is implemented, Cisco Unified IP Phones use the TLS protocol when securely registering with Cisco Unified CallManager.
Trivial File Transfer Protocol (TFTP)	TFTP allows you to transfer files over the network. On the Cisco Unified IP Phone, TFTP enables you to obtain a configuration file specific to the phone type.	TFTP requires a TFTP server in your network, which can be automatically identified from the DHCP server. If you want a phone to use a TFTP server other than the one specified by the DHCP server, you must manually assign the IP address of the TFTP server using the Network Configuration menu on the phone.

Figura 139 – Protocolos suportados pelos telefones (Parte III) [Cisco0?a]

Networking Protocol	Purpose	Usage Notes
User Datagram Protocol (UDP)	UDP is a connectionless messaging protocol for delivery of data packets.	Cisco Unified IP Phones transmit and receive RTP streams, which utilize UDP.

Figura 140 – Protocolos suportados pelos telefones (Parte IV) [Cisco0?a]

Anexo VII - Videoconferência e emissão de vídeo

VII.1 - Protocolos suportados pelo *TANDBERG MXP*

A seguinte figura mostra quais os protocolos suportados pelo TANDBERG MXP. Esta figura corresponde a um dos apêndices do manual *TANDBERG MXP*.

PROTOCOLS SUPPORTED

TANDBERG supports a variety of protocols:

TCP/IP - TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL

A set of networking protocols that provides connectivity over LAN/WAN to any network computer.

HTTP - HYPERTEXT TRANSFER PROTOCOL

A protocol used to transfer information on the internet. A web-browser interface is used to access the management computer. (Max number of simultaneous connections is unlimited, although only one is processed at a time).

HTTPS - HYPERTEXT TRANSFER PROTOCOL SECURE SOCKETS

If a Web server supports the SSL protocol (establish a secure communications channel to prevent the interception of critical information), the Internet address for the server will begin with https:// instead of http://

FTP - FILE TRANSFER PROTOCOL

A member of the TCP/IP suite of protocols used to copy files between two computers on the Internet. FTP provides standard method for remote software upgrades. (Max. number of simultaneous connection = 1).

TELNET

Telnet provides access to management functions by using a standard command-line interface. (Max. number of simultaneous connection = 8, in addition to the RS232 connection).

TELNET CHALLENGE

TMS (Tandberg Management Suite) uses MD5-Challenge Response algorithm (RFC-1321) Telnet access for encryption of password over the IP network.

SNMP - SIMPLE NETWORK MANAGEMENT PROTOCOL

A standard network protocol for management and surveillance of TCP/IP networks (RFC 1157 SNMP v1, RFC 1213 MIB-II).

DHCP - DYNAMIC HOST CONFIGURATION PROTOCOL

DHCP is a TCP/IP protocol that offers dynamic IP addresses and other configuration parameters (subnet mask) to network clients. It provides safe reliable and simple network configuration, prevents address conflicts, and helps conserve the use of client IP addresses.

DNS - DOMAIN NAME SYSTEM

A hierarchical distributed database that contains mappings of DNS domain names to various types of data, such as IP addresses. DNS enables the location of computers and services by user-friendly names, and it also enables the discovery of other information stored in the database.

NETWORK PROTOCOL

Network Protocols are a set of rules and conventions for sending information over communication networks. These rules govern the content, format, timing, sequencing, and error control of messages exchanged among network devices.

LAN – LOCAL AREA NETWORK

A communications network connecting a group of computers, printers, and other devices located within a relatively limited area (for example, a building).

WAN – WIDE AREA NETWORK

A communications network connecting geographically separated computers, printers, and other devices.

SSH – SECURE SHELL

SSH provides the same functionality as Telnet, but the session is encrypted.

SSL - SECURE SOCKETS LAYER

SSL is a proposed open standard for establishing a secure communications channel to prevent the interception of critical information.

Figura 141 – Protocolos suportados pelo equipamento de vídeo [GC97]