

# Ritmo Musical Adaptável à Latência de Rede no Sistema *Public Sound Objects*

Álvaro Barbosa <sup>†‡</sup>

<sup>†</sup> MTG - Music Technology Group  
Universidad Pompeu Fabra  
Calle Ocata 1  
08003 Barcelona, España  
+34 93 542 21 04  
abarbosa@iua.upf.es

Jorge Cardoso

<sup>‡</sup> CITAR - Centro de Investigação Ciências e  
Tecnologias das Artes  
Universidade Católica Portuguesa  
Rua Diogo Botelho 1327  
4169-005 Porto, Portugal  
+351 22 616 62 91  
jccardoso@porto.ucp.pt

**ABSTRACT** — Quando estabelecidos sobre redes informáticas de longa distância os sistemas de criação musical colaborativos comportam um problema crítico: a latência da rede, que é um impedimento à comunicação musical em tempo real. Num estudo recente, foi estabelecida a relação entre tolerância à latência e o ritmo musical. Este resultado emergiu de uma experiência levada a cabo pelos autores deste artigo, na qual foram aplicadas condições simuladas de latência de rede à performance de diferentes músicos enquanto tocavam temas clássicos de Jazz. O projecto *Public Sound Objects* – PSOs é um espaço musical partilhado na web, que tem constituído um paradigma experimental para implementação e teste de diferentes abordagens para a comunicação musical on-line. Este artigo descreve a implementação da última versão do sistema PSOs, incluindo uma componente para instrumentos de música em rede que incorpora a latência como uma funcionalidade de software, que adapta dinamicamente o seu ritmo à latência de comunicação medida em tempo real.

**INDEX TERMS** — Instrumentos de Música em Rede; Latência na Performance em Tempo Real; Instrumentos Musicais Electrónicos com Interface Desacoplados; Composição Algorítmica; Performance Musical à Distância; Colaboração Musical; Música Pública; Objectos Sonoros;

## I. INTRODUÇÃO

Nos últimos anos a Música em Rede tem captado progressivamente a atenção da Comunidade de Criadores de Música feita por Computador. Com o advento da comunicação via Internet, a dispersão geográfica entre os participantes de uma performance musical mediada por computador alcançou uma extensão mundial.

Recentemente a publicação de estudos na área da criação musical em rede, por Álvaro Barbosa em 2003 [1], Gill Weinberg em 2002 [2] e Dante Tanzi em 2001 [3], possibilitou a categorização de diferentes sistemas baseados em diversas arquitecturas e topologias. Nestes sistemas está bem patente que este novo meio de comunicação possui características específicas, com as quais se têm que confrontar. Sobretudo a latência de rede e o *jitter* representam

diferenças significativas em relação a paradigmas de colaboração musical presenciais, atendendo a que a performance musical está tradicionalmente ligada à noção de sincronismo em tempo real entre instrumentos e músicos.

É possível demonstrar que, ao nível da geografia global, existem limitações físicas na actual tecnologia de rede que irá sempre introduzir uma latência mais elevada do que os valores mínimos aceitáveis para uma colaboração acústica em tempo real.

Para o ouvido humano perceber dois sons simultâneos, estes não devem estar separados no tempo mais de 20 ms [4], o que significa que, para existir uma consciência mútua numa performance bilateral este limiar será à volta de 40 ms (o período de tempo que demoraria a um artista a apreender a reacção dos outros artistas à sua acção). Note-se que a percepção de dois sons diferentes simultâneos depende em grande medida das características do som (timbre, tonalidade ou volume sonoro), do estilo de música e de outros tipos de feedback, como por exemplo, estímulos visuais ou físicos. Contudo, o limiar de 40 ms é razoavelmente elevado para se enquadrar no pior cenário possível. Este valor reforça os resultados da pesquisa levada a cabo por Natham Shuett [5], em 2002 na Universidade de Stanford, que definiu experimentalmente o Ensemble Performance Threshold (EPT) para música rítmica entre 20-30 ms, bem como o resultado da pesquisa efectuada por Nelson Lago em 2004 [6] na Universidade de São Paulo, que obteve resultados semelhantes.

Se considerarmos a mais pequena distância de ligação peer-to-peer entre dois pontos opostos no planeta, obtemos uma distância aproximada de 20.004,5 km, correspondente a metade da distância do perímetro aproximado da terra (40.009 km). Mesmo com transferência de dados à velocidade da luz (aprox. 300 mil km/s) e dispondo de largura de banda ilimitada, a latência bi-direccional seria de aproximadamente 133,4 ms, o que excede em muito o limiar mínimo aceitável. Além disso, a

latência em redes informáticas é por natureza altamente variável e imprevisível, criando erros de base de tempo, perda de sequência e até perda parcial de conteúdo, resultando em condições impeditivas para uma performance controlada.

Uma abordagem possível para enfrentar este cenário, é aceitar a latência de rede como um elemento natural na criação de música na internet. A ideia que a latência de rede constitui a acústica própria da internet, e que os compositores deveriam criar música assimilando este facto, é claramente expressa no artigo on-line de Golo-Föllmer “Soft Music” que cita o ponto de vista pessoal do artista experimental Atau Tanaka sobre este tópico: “*Considero bastante interessante o atraso de tempo da internet e penso nisso como um tipo de acústica único deste meio (...) em vez de tocar a música existente nesta nova base de tempo, o que é interessante para mim é tentar encontrar uma linguagem musical que funcione neste eixo temporal (...)*” (excerto de uma entrevista em vídeo disponível em [7]).

O conceito de um espaço acústico na Internet foi também reforçado por Chris Chafe do grupo SoundWIRE, no Centro para Pesquisa em Música e Acústica da Universidade de Stanford (CCRMA), que publicou vários estudos acerca da influência das condições da rede na comunicação acústica [8] [9].

## II. RITMO E LATÊNCIA

Diversas experiências foram realizadas com o objectivo de determinar a quantidade máxima de latência na comunicação que pode ser tolerada entre músicos para acompanhar uma performance sincrónica.

Alguns dos resultados mais significativos relativamente aos efeitos da latência temporal na precisão do ensemble foram obtidos por Chris Chafe e Michael Gurevish [10]. Da experiência realizada no CCRMA conclui-se claramente que, ao aumentar a latência de comunicação entre pares a tentar sincronizar um ritmo estável de batimentos de palmas, os sujeitos têm tendência para abrandar o ritmo.

Igualmente, uma experiência levada a cabo pelos autores deste artigo em Junho de 2004 no Departamento de Som e Imagem da Universidade Católica Portuguesa visou, entre outros objectivos, estudar a relação entre Ritmo e Latência. Na experiência, foram aplicadas condições de latência de rede simuladas, à performance de quatro músicos diferentes, enquanto tocavam temas clássicos de Jazz com quatro instrumentos diferentes (Baixo, Percussão, Piano e Guitarra). Num sistema experimental instalado em estúdio, os músicos recebiam, através de *headphones*, o *feedback* acústico dos seus próprios instrumentos com latência aplicada. A sua performance era sincronizada com um metrónomo, ao longo de várias sessões, com tempos diferentes (Batidas Por Minuto – BPMs). Em cada sessão aumentou-se a latência até o músico não ser capaz de manter uma performance sincrónica.

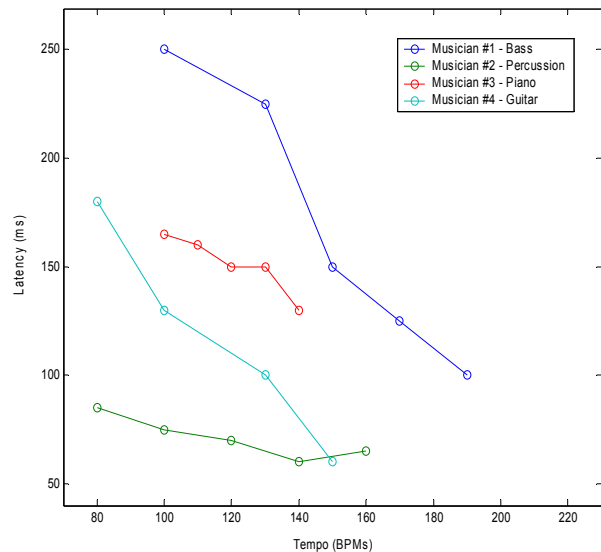


Figura 1. Auto-Teste de tolerância individual à latência na performance de 4 músicos diferentes

O Gráfico da Figura 1 mostra que, independentemente das capacidades instrumentais ou do instrumento musical, todos os músicos foram capazes de tolerar mais latência no *feedback* individual para ritmos mais lentos. A única excepção a esta tendência ocorre na curva da percussão quando aumentado para 160 BPMs, o que está relacionado com uma sobreposição da estrutura rítmica da música, juntamente com o facto de ser bastante difícil de isolar totalmente o artista do som directo do instrumento em instrumentos de percussão. Nestes resultados é claro que há uma relação inversa entre Ritmo e tolerância à Latência.

Após obterem estes resultados, os autores procederam à implementação deste conceito no sistema de *Public Sound Objects* (PSOs), visando alcançar um instrumento musical de rede que incorpore a latência como uma função de software, que adapta dinamicamente o seu Ritmo à latência de comunicação medido em tempo real.

## III. O PROJECTO PUBLIC SOUND OBJECTS

O projecto de *Public Sound Objects* (PSOs) consiste num Ambiente Virtual de Colaboração via web com ênfase na performance musical. Este projecto está ser desenvolvido no Grupo de Tecnologia Musical da Universidade de Pompeu Fabra em Barcelona, conjuntamente com o Centro de Investigação em Ciência e Tecnologia das Artes (CITAR) da Universidade Católica Portuguesa no Porto. O PSOs tem sido um modelo experimental para implementação e teste de diferentes abordagens na comunicação musical on-line. Em Dezembro de 2002 foi

publicada em [11], uma especificação preliminar do sistema, e implementado o primeiro protótipo<sup>1</sup>.

Conceptualmente, o PSOs explora a noção de um espaço web de partilha para a comunidade de criação musical e de uma instalação artística que une o espaço físico e a presença virtual na Internet. O Sistema visa permitir interacção síncrona estabelecendo a base para a improvisação sonora conjunta entre utilizadores da web.

A arquitectura do sistema foi projectada seguindo os seguintes pontos-chave: (a) É baseada numa Topologia Centralizada com Servidor suportando múltiplos utilizadores ligados simultaneamente e comunicando entre eles através do som; (b) É um evento público permanente com características especiais que apelam, tanto ao público do “mundo real” como a um público virtual on-line; (c) A contribuição dos participantes on-line é limitada adequadamente para que a coerência estética total da peça possa ser assegurada; (d) O sistema é modular e escalável permitindo a futura expansão e diferentes configurações.

Neste sistema as matérias-primas, disponibilizadas aos utilizadores para manipulação durante uma performance, são Objectos de Sonoros. A definição de um Objecto de Sonoro como um elemento relevante do processo de criação de música remonta ao início dos anos 60 [12]. Segundo Pierre Schaeffer, um Objecto de Sonoro é definido como “qualquer fenómeno de som ou acontecimento percebido como um todo coerente (...) independentemente da sua fonte ou significado” [13]. De um ponto de vista psicoacústico e perceptual, a definição de Schaeffer é extremamente útil, visto que proporciona um paradigma muito eficaz para esculpir o valor simbólico expresso por uma peça sonora.

No sistema PSOs um sintetizador de som tempo real localizado no servidor activa um Objecto de Sonoro segundo a acção do utilizador. Uma vez que o *Feedback* da performance de outros utilizadores é estritamente acústico, a distinção de Objectos Sonoros na paisagem sonora global é o elemento chave que permite a consciência da acção individual de cada utilizador.

#### IV. A ARQUITECTURA DO PSOS

O sistema PSOs é composto pelo Servidor PSOs e por múltiplos Clientes PSOs. Os clientes manuseiam uma interface visual interactiva e o servidor lida com toda a computação relativamente à síntese e transformação de som. É uma aplicação com o Interface desacoplado levada ao cenário mais extremo, no qual o motor de síntese é separado da interface do utilizador numa Rede Global [14].

A implementação actual segue a metáfora de uma bola que se movimenta infinitamente entre as paredes de uma sala vazia. Quando a bola bate numa das paredes é enviada uma mensagem de rede para o servidor central

onde o Objecto de Som correspondente é accionado, tocado através de um altifalante e simultaneamente transmitido de volta para o utilizador numa mistura estéreo de todos os sons que estão presentes naquele momento.

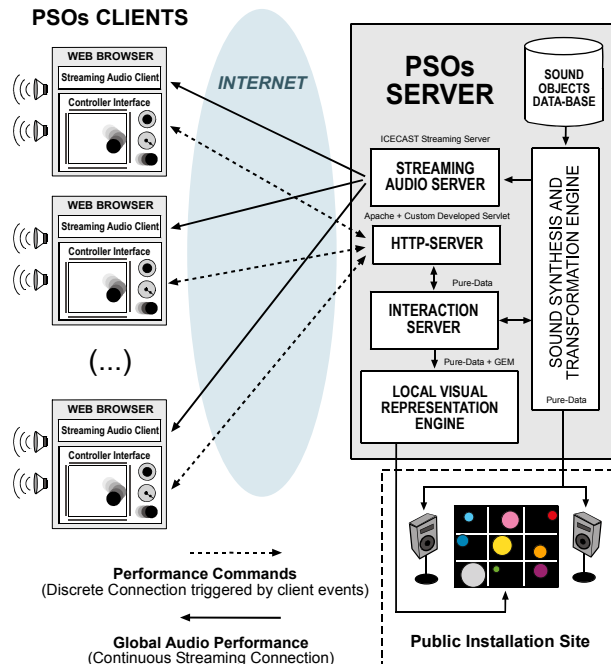


Figura 2. A Arquitectura do Sistema PSOs

Os clientes comunicam com o servidor através do HTTP enviando e recebendo pacotes de informação. Há vários tipos de pacotes de informação que os clientes podem enviar, mas os mais importantes são o *ImpactPacket* – que informa o servidor que uma bola em movimento embateu numa das paredes; o *ControlPacket* – que informa o servidor que o utilizador alterou o valor de um dos controlos do interface; e o *PingPacket* – que se usa para medir a latência de rede entre o cliente e o servidor.

Os pacotes são recebidos pelo servidor através de uma aplicação Web que as reencaminha para o Servidor de Interação – um módulo do Servidor PSOs que gere clientes, instrumentos e eventos produzidos pelo Cliente PSOs. Em função dos pacotes de dados recebidos, um som é gerado pelo Motor de Síntese e Transformação e enviado de volta para o cliente pelo Servidor *Streaming*, sendo a representação visual do cliente actualizada no local de instalação pelo Motor Local de Representação Visual.

O Servidor e os Clientes são compostos por diferentes módulos, descritos nas secções seguintes.

<sup>1</sup> Uma versão experimental do Sistema PSOs está disponível no endereço: <http://www.iaa.upf.es/~abarbosa/>

#### IV.1 O Servidor HTTP

Os Clientes ligam-se ao Servidor PSOs através de ligações HTTP standard. Embora a opção inicial fosse implementar comunicações baseadas em UDP – mais rápidas do que um protocolo TCP como o HTTP – a ideia teve de ser abandonada por duas razões principais:

- A maior parte dos *firewalls* bloqueia todo o tráfego UDP, o que significa que um grande número de utilizadores não conseguiriam aceder ao nosso servidor. Também iria aumentar a dificuldade na disponibilização do Servidor PSOs pelas mesmas razões: o *firewall* teria de permitir o tráfego UDP numa porta específico.

- Algumas políticas de segurança dos browsers para *applets* de Java só lhes permitem fazer ligações usando o protocolo HTTP.

Para ultrapassar estas restrições foi concebido um sistema de comunicação usando um protocolo que todas as *firewalls* autorizam: HTTP. Para isto foi implementada uma aplicação de servidor, usando a tecnologia Java *Servlet*, que funciona como um proxy entre a *applet* do Cliente do PSOs e o Servidor de Interação. Basicamente esta *servlet* só passa dados recebidos do Cliente PSOs para o Servidor de Interação e vice-versa.

#### IV.2 O Servidor de Interação

O Servidor de Interação é a peça central no Servidor PSOs. É um módulo de Pure Data (PD) que recebe pacotes de dados dos clientes sob a forma de datagramas (através do Servidor HTTP) e actua em função do tipo de pacote recebido.

Um objecto PD teve de ser implementado para a recepção dos datagramas UDP – que se designou como Extended Netreceive [xnetreceive] – uma vez que os objectos existentes para este propósito não permitem que os PD adquiram o endereço IP e o número da porta do cliente que iniciou a comunicação.

Os tipos de pacotes definidos até este momento são os seguintes:

**AvailableInstruments:** Quando o Servidor de Interação recebe este tipo de pacote envia como resposta os números dos instrumentos que estão disponíveis. Os instrumentos são numerados de 1 a 9.

**LockInstrument:** Este tipo de pacote é enviado para o servidor quando o utilizador escolhe um instrumento para tocar. O número do instrumento é especificado no pacote. O Servidor de Interação irá verificar se o instrumento ainda se encontra disponível e irá responder com um resultado Verdadeiro/Falso dependendo de se o instrumento foi ou não adquirido com sucesso. Quando um instrumento é adquirido só pode ser usado pelo cliente que o adquiriu.

**UnlockInstrument:** Informa o servidor que o utilizador acabou de utilizar o instrumento especificado pelo número de instrumento no pacote de dados. O Servidor de

Interação irá desbloquear o instrumento, que irá ficar disponível nessa altura para os outros clientes.

**ImpactPacket:** Este é o pacote mais usado e um dos mais importantes. Informa o servidor que a bola em movimento bateu na parede e que pode gerar um som. O Servidor de Interação passa estes pacotes para o Motor de Síntese e Transformação e para o Motor Local de Representação Visual. De entre outra informação, estes pacotes especificam o número do instrumento, o valor dos *sliders* de parede (na interface), a velocidade da bola, o tamanho da mesma, a parede e o ponto da parede em que bateu, e o tamanho do rasto da bola. Nesse momento esta informação é usada pelo Motor de Síntese e Transformação para gerar um som segundo os parâmetros estabelecidos pelo utilizador na interface de Cliente PSOs. É também usada pelo Motor Local de Representação Visual para actualizar a representação visual daquele utilizador.

**ControlPacket:** Este tipo de pacote tem interesse somente para o Motor Local de Representação Visual. A informação que é enviada é a mesma do ImpactPacket mas os eventos que despoletam o envio são diferentes. Os ControlPackets são enviados sempre que o utilizador altere a velocidade, o tamanho ou tamanho do rasto da bola. O Servidor de Interação passa estes pacotes para o Motor Local de Representação Visual para que o servidor local de instalação possa ser actualizado.

**PingPacket:** Estes pacotes não contêm informação directa; o seu único objectivo é permitir ao Cliente PSOs determinar a latência de rede entre o cliente e o servidor. O Servidor de Interação só envia de volta ao cliente uma resposta vazia.

A outra tarefa principal do Servidor de Interação é gerir os Clientes PSOs que se encontram ligados. Se um cliente perdesse a ligação da rede sem ter enviado um UnlockPacket, o instrumento que se encontrava correntemente seleccionado nunca mais ficaria disponível. É a função do Servidor de Interação detectar tais situações e automaticamente libertar o instrumento. Isto é efectuado com *timeouts*, i.e., se um cliente permanece sem contactar o servidor mais do que uma quantidade de tempo fixa, esse cliente é removido da lista de clientes ligados naquele momento e o seu instrumento é disponibilizado.

#### IV.3 Motor de Síntese e Transformação

O Motor de Síntese e Transformação é responsável pela produção de som em resposta aos eventos gerados pelos Clientes PSOs. Este motor é um *patch* de PD automaticamente carregado pelo Servidor de Interação. Recebe ImpactPackets do Servidor de Interação (listas PD) e gera um som segundo os valores aí especificados. Os parâmetros retirados destes pacotes de dados são na realidade transmitidos a um dos nove módulos de síntese.

Neste momento, o motor tem nove módulos que correspondem aos nove instrumentos disponíveis para os utilizadores. Uma vez que cada módulo é diferente e

independente, o mesmo parâmetro pode ter um significado diferente em cada um dos módulos. Estes módulos são:

*Karplus-Strong Guitar*: Como nome sugere, este é uma implementação do algoritmo *Karplus-Strong* para um som de corda “puxada” implementado em PD.

*Sintetizador FM*: Um sintetizador de modulação em frequência.

*Vibrafone de Impacto Modal*: Uma tentativa de produzir sons semelhantes ao vibrafone usando modelos físicos de Impacto Modal implementados para PD, disponíveis do Projecto *Sounding Objects* [15].

*Piano, Percussão, Violino, Orquestra, Tabla e Poet Samplers*: Estes são de facto só um módulo, carregado com seis sons diferentes. O *sampler* foi implementado em PD com seis vozes, o que provou ser suficiente para não sobrecarregar o sistema, nos casos de pior cenário (nove utilizadores conectados com performances de ritmo elevado).

O som gerado é produzido por estes módulos no formato MP3, e transmitido usando o objecto PD [shoutcast~] para um servidor de *streaming* áudio. Foi utilizado o servidor de *streaming Icecast2* para Windows.

Cada utilizador pode escolher um destes módulos como o motor gerador de som do ecrã de entrada de PSOs ao nível do cliente.

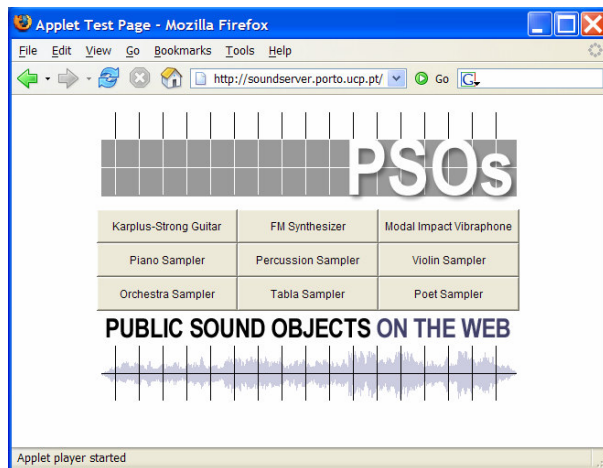


Figura 3. Ecrã de Entrada do Cliente de PSOs

Ao carregar a página web de entrada, se um Módulo de Som já estiver a ser usado por outro utilizador, o seu botão irá estar indisponível e o Módulo só será disponibilizado, quando o utilizador que o está a usar, o libertar.

#### IV.4 O Interface do Cliente

Uma vez seleccionado o motor gerador de som (instrumento), o web-browser carrega o *applet* do

interface controlador, que liga ao servidor de interacção, regista e inicializa uma sessão de utilizador.

A interface gráfica do utilizador (GUI - graphical user interface) pode ser diferente em desenvolvimentos futuros.

A implementação do GUI para o PSOs é desenvolvida segundo os seguintes requisitos: (a) deve permitir ao utilizador contribuir na performance musical que está a decorrer, podendo transformar as características de uma representação visual de um Objecto de Sonoro, enviando parâmetros normalizados para o motor de síntese através da rede; (b) a aplicação do interface deve ser capaz de permitir a manipulação de cada um dos parâmetros modificadores no motor de síntese em articulação com a configuração específica do local da instalação central; (c) o próprio GUI deve ser um interface metafórico orientado por comportamento, evitando um mapeamento directo de parâmetros, tais como potenciômetros ou botões, proporcionando o comportamento automático periódico para os Objectos Gráficos, que podem ser comandados pelo utilizador.

A Figura 4 mostra a interface do cliente na sua implementação actual.

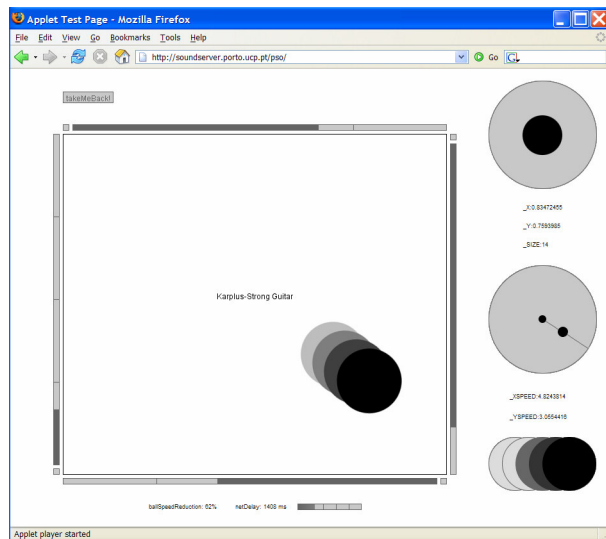


Figura 4. A interface do Cliente PSOs

A bola move-se continuamente e o utilizador pode manipular o seu tamanho, velocidade, direcção, extensão do rasto e a textura acústica de cada parede. Os valores normalizados são depois enviados para o servidor e mapeados nos parâmetros de síntese. A textura acústica da parede reflecte a tonalidade do Objecto de Sonoro (os valores individuais de tonalidade podem ser atribuídos a cada parede, permitindo a criação de estruturas de som melódicas e rítmicas) e a extensão do rasto da bola corresponde ao número de réplicas do atraso acústico aplicado ao Objecto de Sonoro.

#### IV.5 O MOTOR DE REPRESENTAÇÃO VISUAL

O Motor Local de Representação Visual cria a representação visual do modelo da bola de todos os Clientes PSOs, na localização física do servidor. Esta implementação local de um *patch* de PD usa o *external Ambiente Gráfico para Multimédia (Graphics Environment for Multimedia* ou GEM) para criação gráfica, usando a informação dos *ImpactPackets* e dos *ControlPackets* para actualizar a informação do estado para cada cliente.

A apresentação visual é composta por um *Videowall* com nove ecrãs dispostos numa matriz de 3 por 3. A cada ecrã é atribuído um instrumento na mesma ordem em que são apresentadas ao utilizador na interface do Cliente PSOs.



Figura 5. A Instalação Local do Sistema PSOs

Os clientes são representados no local da instalação como esferas com cores, tamanhos e velocidades diferentes. A cada cliente é atribuído um ecrã na parede de vídeo que também delimita o movimento da esfera correspondente, i.e., os limites de cada ecrã são mapeados até aos limites da janela do Cliente PSOs. Sempre que um novo cliente se liga, é escolhida uma cor ao acaso para o representar. A atribuição casual de cores aos clientes torna o site de instalação mais dinâmico e cativante para os utilizadores.

Foram escolhidos mais dois parâmetros para proporcionar *feedback* visual: a velocidade da bola e os eventos gerados na interface do cliente. Embora haja um *feedback* visual implícito na velocidade da bola, i.e., a esfera se mova mais rápida ou mais lenta no ecrã, foi acrescentada uma indicação adicional modificando a saturação da cor da esfera (foi utilizado o modelo de cor HSB). Às vezes quando a bola é ajustada para um tamanho maior e ocupa quase o ecrã inteiro, é difícil dizer qual é a sua velocidade porque tanto uma bola lenta como uma bola rápida irão saltar bastante. Mapear a velocidade até à saturação de cor – saturação alta para uma bola lenta, e saturação baixa para uma rápida – ajuda os espectadores a distinguir estas situações. Quando o Motor Local de Representação Visual recebe um pacote, significando que um evento foi despoletado na interface do cliente PSOs, transforma a esfera desse cliente, temporariamente, numa malha poligonal.

O motor só tem informação exacta quando os clientes enviam pacotes ao servidor. No tempo restante, a posição da bola tem de ser interpolada com base na informação do último pacote. Não é possível ter uma representação completamente exacta da bola do utilizador devido à latência da rede, os diferentes sincronismos entre clientes e servidor e também porque não podemos prever as acções do utilizador. Apesar disto, é possível obter uma representação bastante fiel dos vários clientes. O aspecto indesejável de representação mais notável é o ocasional “salto” da esfera, por exemplo, às vezes a representação é mais rápida do que a bola do cliente, por isso quando um pacote é recebido, a posição é actualizada de repente para a posição correcta causando um “salto” atrás da esfera.

#### V. RITMO ADAPTADO À LATÊNCIA DE REDE

A ideia de um instrumento de música em rede que se adapta dinamicamente à latência da rede foi implementada recentemente por Jörg Stelken no Software *PeerSynth* [16]. *PeerSynth* é um sintetizador de som *peer-to-peer* que suporta utilizadores múltiplos distribuídos pela Internet, medindo a latência entre cada ligação activa e baixando o volume de som dinamicamente de cada contribuição do utilizador na paisagem sonora, proporcionalmente à quantidade de latência medida na sua ligação. Stelkens seguiu uma metáfora do mundo real, onde de facto o volume de som de uma fonte de sonora diminui com a distância ao receptor, o que também implica um aumento da latência de comunicação acústica. Uma abordagem semelhante foi seguida no Sistema *AALIVENET* [17].

O sistema PSOs aborda esta mesma ideia, mas tratando uma relação entre características musicais e latência de comunicação menos imediata, porém igualmente relevante. Implementa um mecanismo de Ritmo Adaptável à Latência na Rede, que reduz dinamicamente a ritmo do tempo musical segundo a latência medida em tempo real entre o cliente e o servidor.

Na interface da bola em movimento o tempo musical corresponde à velocidade da bola. O factor de redução aplicado à velocidade da bola é apresentado na interface do utilizador e é calculado para que se garanta que em média a bola não bata duas vezes nas paredes sem o som, accionado na primeira pancada, chegue ao cliente. A ideia é que a bola seja tão rápida quanto a ligação o permita. Deste modo, o efeito de latência é muito menos confuso, permitindo ao utilizador ter uma melhor percepção da relação entre uma batida na parede e o som correspondente accionado.

#### VI. ESTUDO E AVALIAÇÃO DE UTILIZADORES

O Sistema PSOs, incluindo a instalação central física no local do servidor, foi instalado no Campus da Universidade Católica Portuguesa no Porto entre 7 a 14 de Outubro de 2004. Durante este período de ensaio foram instalados várias instâncias de cliente e 109 sujeitos testaram o sistema e responderam a questionários.

Os resultados médios extraídos desta sondagem de opinião são apresentados na figura 6.

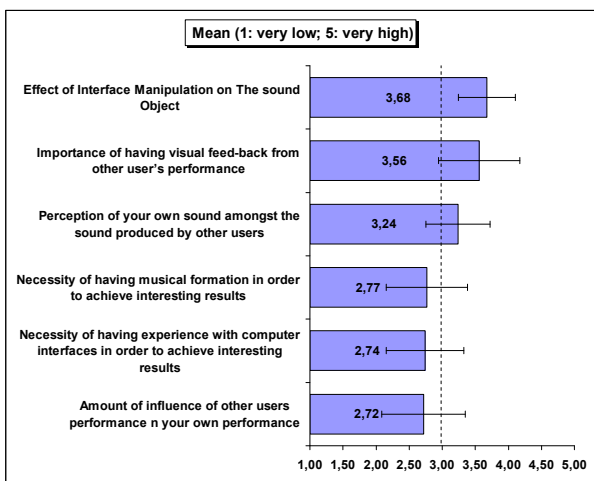


Figura 6. Avaliação preliminar de resultados do estudo de utilizadores do sistema PSOs

Alguns destes resultados vão de encontro às nossas expectativas: (1) A interface é eficaz para estabelecer uma relação entre a acção do utilizador e o seu efeito no Objecto de Som correspondente; (2) Os Objectos de Som disponíveis no formato actual permitem a diferenciação acústica na paisagem sonora global, (3) É um sistema acessível para o público em geral, sem requerer uma prévia formação musical ou capacidades prévias de manipulação de GUI.

Por outro lado parece que, de um modo geral, os utilizadores sentem a necessidade de ter uma representação visual do comportamento de outros utilizadores, e talvez por esta razão, o efeito colaborativo parece ser relativamente baixo.

## VII. CONCLUSÕES E TRABALHO FUTURO

Durante mais de dois anos o projecto de Public Sound Objects providenciou com sucesso um modelo experimental para implementação e teste de diferentes abordagens na comunicação musical on-line. O recente estudo de utilizadores e respectiva avaliação confirmaram algumas das principais ideias por trás do design do sistema. Contudo, também sugerem mais melhorias, tais como uma representação visual de outros utilizadores nos clientes, e possivelmente síntese de som no lado do cliente usando um *plug-in* que permita suportar PD num web-browser.

O sistema de Ritmo Adaptável à Latência na Rede, implementado na última versão do PSOs, representa uma melhoria significativa na usabilidade do sistema. Mais desenvolvimentos serão explorados para melhorar a tolerância de latência especificamente na interface da bola em movimento usando características de espacialização.

## VIII. AGRADECIMENTOS

Os Autores agradecem a Gunter Geiger e Martin Kaltenbrunner pelo trabalho desenvolvido na primeira fase de implementação

do projecto PSOs e Alexander Carôt pela sua colaboração no teste de Latência VS Ritmo. Este trabalho de investigação é apoiado pela Fundação para a Ciência e Tecnologia.

## XI. REFERÊNCIAS

- [1] A. Barbosa, *Displaced Soundscapes: A Survey of Network Systems for Music and Sonic Art Creation*. Leonardo Music Journal, Volume 13, Issue 1, pp. 53-59, MIT Press, Cambridge MA, 2003.
- [2] G. Weinberg. *The Aesthetics, History, and Future Challenges of Interconnected Music Networks*. 349-356. 2002. Proceedings of the International Computer Music Conference.
- [3] D. Tanzi, *Observations about Music and Decentralized Environments*. Leonardo Music Journal, Volume 34, Issue 5, pp. 431-436, 2001.
- [4] I. Hirsh, *Auditory Perception of temporal Order*. Journal of the Acoustical Society of America, Volume 31, pp. 759-767, 1959.
- [5] N. Schuett. *The Effects of Latency on Ensemble Performance*. 2002. Stanford University.
- [6] N. Lago and F. Kon. *The Quest for Low latency*. 33-36. 2004. Proceedings of the International Computer Music Conference (ICMC2004).
- [7] G. Föllmer. *Crossfade - Sound Travels on the Web - Soft Music*. 2001. San Francisco Museum of Modern Art; ZKM - Center for Art and Media - Karlsruhe; Walker Art Center - Minneapolis ; Goethe Forum - Munich. <http://crossfade.walkerart.org/>.
- [8] C. Chafe and G. Niemeyer. *Ping music installation, 2001, Walker Art Center / SFMOMA*. 2001. <http://crossfade.walkerart.org/>.
- [9] C. Chafe, S. Wilson and D. Walling. *Physical Model Synthesis with Application to Internet Acoustics*. 2002. IEEE - Signal Processing Society. Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Orlando - Florida (ICASSP2002).
- [10] C. Chafe, M. Gurevich, Grace Leslie and Sean Tyan. *Effect of Time Delay on Ensemble Accuracy*. 2004. Proceedings of the International Symposium on Musical Acoustics, Nara - Japan (ISMA2004).
- [11] A. Barbosa and M. Kaltenbrunner. *Public Sound Objects: A shared musical space on the web*. 2002. IEEE Computer Society Press. Proceedings of International Conference on Web Delivering of Music (WEDELMUSIC 2002) - Darmstadt, Germany.
- [12] P. Schaeffer, *Traité des Objets Musicaux*., 1966.
- [13] M. Chion, *Guide des Objets Sonores. Pierre Schaeffer et la Recherche Musicale*., 1983.
- [14] A. Barbosa, M. Kaltenbrunner and G. Geiger. *Interface Decoupled Applications for Geographically Displaced Collaboration in Music*. 2003. Proceedings of the International Computer Music Conference (ICMC2003).
- [15] D. Rocheso and F. Fontana, *The Sounding Object*, 2003.
- [16] J. Stelkens. *peerSynth: A P2P Multi-User Software with new techniques for integrating latency in real time collaboration*. 2003. Proceedings of the International Computer Music Conference (ICMC2003).
- [17] M. Spicer. *AALIVENET: An agent based distributed interactive composition environment*. 2004. Proceedings of the International Computer Music Conference (ICMC2004).

