

Mconf-Mobile: videoconferência BigBlueButton no Android

Felipe Cecagno, Valter Roesler
Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre, Rio Grande do Sul, Brasil
{fcecagno, roesler}@inf.ufrgs.br

RESUMO

Este artigo apresenta um dos resultados desenvolvidos no projeto Mconf. O Mconf-Mobile é um aplicativo de código aberto para dispositivos móveis com sistema operacional Android. Ele permite interação entre usuários de smartphones e tablets com usuários de computadores de mesa através do sistema de webconferência BigBlueButton. Serão apresentados alguns objetivos do projeto, a arquitetura da solução e suas principais funcionalidades.

ABSTRACT

This article presents a result of the Mconf project. The Mconf-Mobile is an application for mobile devices with Android operational system. It makes possible to interact with other users in a BigBlueButton videoconference. We present some goals of the project, the solution's architecture and its main functionalities.

Categories and Subject Descriptors

C.3 [Special-purpose and application-based systems]: Real-time and embedded systems; H.4 [Information systems applications]: Communications Applications: Computer conferencing, teleconferencing, and videoconferencing

Keywords

Videoconferência, Android, BigBlueButton, Código aberto, Mconf

1. INTRODUÇÃO

Dispositivos móveis inteligentes vêm ganhando a cada dia mais espaço e popularidade. A troca de telefones celulares “burros” por smartphones acontece por diversos motivos, e um deles é a grande variedade de aplicativos disponíveis. Dentre eles, os mais populares são os aplicativos de entretenimento [3], categoria que inclui os jogos, que aproveitam a capacidade de processamento dos dispositivos e oferecem distração aos usuários.

Além dos jogos, aplicações de vídeo em tempo real vêm crescendo no mercado de aplicativos. Clientes VoIP¹ como o Sipdroid e aplicativos populares por sua versão para desktop como o Skype já incluem a funcionalidade de videochamada nos seus aplicativos para dispositivos móveis. Além destes, o Fring permite chamada em grupos de até quatro pessoas.

Entretanto, tanto o Skype quanto o Fring permitem chamadas apenas para clientes de mesmo tipo, ou seja, Skype se comunica com Skype e Fring se comunica com Fring, mas não existe interação entre eles. Os clientes VoIP são mais genéricos, mas ao mesmo tempo mais limitados quanto a funcionalidades.

O principal objetivo do projeto Mconf é proporcionar uma solução de código aberto para webconferência com foco na facilidade de uso e integração com dispositivos móveis e redes sociais². A solução integra o sistema de webconferência BigBlueButton [1] com o portal web em forma de rede social Global Plaza, e prevê interação com dispositivos móveis através do Mconf-Mobile.

O projeto Mconf é desenvolvido pelo Laboratório de Projetos em Áudio e Vídeo da Universidade Federal do Rio Grande do Sul, teve início em novembro de 2010 e é financiado pela Rede Nacional de Ensino e Pesquisa no âmbito do programa de Grupos de Trabalho da RNP.

O código-fonte do Mconf-Mobile está disponível em um repositório público³ e é licenciado sob a *GNU General Public License*. O aplicativo está disponível para *download* no Google Code do projeto⁴ e futuras versões serão disponibilizadas gratuitamente na loja oficial do Android.

2. ARQUITETURA

Conforme descrito anteriormente, o Mconf integra dois sistemas existentes de código aberto voltados para videoconferência: o BigBlueButton e o Global Plaza. Entretanto, o aplicativo Android se comunica diretamente com o BigBlueButton e não realiza qualquer interação com o Global Plaza. Isso se deve a uma decisão estratégica de projetar o Mconf-Mobile como o cliente Android oficial do BigBlueButton. Dessa forma, usuários do BigBlueButton podem utilizar o aplicativo independente do portal web do Mconf. Tal decisão fez o projeto ganhar visibilidade e atrair a atenção da comunidade de desenvolvedores do BigBlueButton (que conta com mais de 950 membros⁵).

O BigBlueButton é um sistema de código aberto para webconferência voltado principalmente para ensino à distância.

²<http://inf.ufrgs.br/prav/gtmconf>

³<https://github.com/mconf/mconf-mobile>

⁴<http://code.google.com/p/mconf/>

⁵<http://groups.google.com/group/bigbluebutton-dev/about>

¹ *Voice over IP*, ou Voz sobre IP

Ele oferece funcionalidades como interação por áudio, vídeo e bate-papo (público e privado), além de compartilhamento da tela do computador e apresentação síncrona de slides e documentos.

O cliente web para desktop é desenvolvido na plataforma Adobe Flash e toda a comunicação com o servidor é feita através do protocolo proprietário RTMP⁶. Já o servidor do BigBlueButton utiliza o Red5⁷, que é uma implementação livre do Adobe Flash Media Server. Além da comunicação por vídeo, realizada de forma transparente entre o cliente Flash e o servidor Red5, são utilizados *Remote Shared Objects* (RSO) e *Remote Procedure Calls* (RPC) para as demais trocas de mensagens, como bate-papo e gerenciamento de status de participantes.

Para áudio, o sistema utiliza um servidor VoIP, que pode ser tanto o FreeSWITCH quanto o Asterisk. No cliente web é utilizada uma implementação de telefone SIP⁸ em Flash.

Apesar do Android oferecer suporte a plataforma Flash a partir da sua versão 2.2, optou-se por desenvolver um aplicativo nativo através da SDK padrão do Android para permitir compatibilidade com as versões anteriores. A arquitetura do aplicativo será detalhada nas subseções seguintes.

2.1 Flazr

Toda a comunicação RTMP entre o aplicativo e o servidor Red5 é feita através da biblioteca Flazr, uma implementação livre em Java de protocolos de *streaming* multimídia⁹. A biblioteca implementa partes importantes do protocolo RTMP, como por exemplo o *handshake* inicial e multiplexação/demultiplexação de mensagens de controle, áudio e vídeo, comandos de RPC e controle de fluxos de dados.

Entretanto, a biblioteca não oferece suporte a RSO, recurso indispensável para realização plena da comunicação com o servidor. Por exemplo, informações de entrada e saída de participantes, presença de vídeo de um determinado participante ou pedido de atenção através do recurso “levantar a mão” são ações informadas através do objeto **participantsSO**. Outro exemplo é a troca de mensagens públicas de bate-papo, que é realizada através do objeto **chatSO**.

Por isso, o suporte a RSO teve de ser adicionado à biblioteca. A implementação de gerência de objetos compartilhados e da multiplexação/demultiplexação de mensagens teve como base o código do Red5 e a especificação do protocolo RTMP [2].

2.2 BBB-Java

O BBB-Java é a biblioteca responsável pela interação entre um aplicativo genérico em Java e um servidor BigBlueButton. Essa biblioteca foi desenvolvida pelo projeto Mconf para ser utilizada no Mconf-Mobile, mas buscou-se manter um baixo acoplamento entre a biblioteca e o aplicativo Android exatamente para possibilitar e incentivar o desenvolvimento de

novos aplicativos não-web integrados ao sistema de webconferência. Além disso, a biblioteca pode ser utilizada na criação de aplicativos robôs que auxiliem a realização de testes de funcionamento e de carga.

Dentre os recursos oferecidos pela biblioteca estão:

- Gerenciamento de salas de conferência (pesquisa das salas existentes, criação de novas salas e acesso a uma sala específica);
- Atualização do status dos participantes da sessão;
- Troca de mensagens de bate-papo (público e privado);
- Recepção e envio de dados de vídeo.

O gerenciamento de salas de conferência é feito através de chamadas a um aplicativo web JSP (*mobile.jsp*), desenvolvido no projeto Mconf, que utiliza a API do BigBlueButton para executar as funções citadas. Um aplicativo intermediário é necessário, pois a API do BigBlueButton utiliza um *security salt* alfanumérico de 32 caracteres para gerar um *checksum* de cada requisição. Esse *salt* é conhecido apenas no ambiente web, não em aplicações externas como é o caso do BBB-Java, e não é plausível que um aplicativo exija do usuário uma senha de 32 dígitos para acesso a um servidor.

Inicialmente, em caráter experimental, o *mobile.jsp* removia a necessidade do *salt*. Essa brecha de segurança foi solucionada quando um novo *salt* de apenas 5 caracteres foi implantado, tornando factível a entrada desse dado pelo usuário.

Os demais recursos são oferecidos pela biblioteca através de uma comunicação RTMP com o servidor Red5. Para tal utilizou-se a versão modificada da biblioteca Flazr, com suporte a RSO.

2.3 BBB-Android

O BBB-Android é o aplicativo Android que utiliza o BBB-Java para interação com o servidor BigBlueButton. Esse aplicativo foi escrito principalmente em Java, utilizando o *Software Development Kit* (SDK) padrão do Android, e uma pequena parte crítica em desempenho foi desenvolvida em C/C++ com o *Native Development Kit* (NDK).

Considerando que o servidor BigBlueButton utiliza um servidor VoIP para lidar com os fluxos de áudio, para o BBB-Android adotou-se a estratégia de integrar ao aplicativo uma solução existente de VoIP de código aberto. Optou-se pelo aplicativo Sipdroid¹⁰, um dos telefones SIP mais populares do Android Market, adicionando assim a funcionalidade de interação por áudio. Dentre os codecs suportados pelo Sipdroid está o Speex, codec padrão de áudio do BigBlueButton, implementado na linguagem C e compilado através do NDK.

O codec de vídeo padrão utilizado no BigBlueButton é o H.263, e a estratégia de solução foi a mesma do Sipdroid. Optou-se por utilizar uma compilação otimizada da biblioteca FFmpeg¹¹ para Android, gerada através do NDK, que

⁶ Real-Time Messaging Protocol

⁷ <http://www.red5.org/>

⁸ Session Initiation Protocol

⁹ <http://flazr.com>

¹⁰ <https://market.android.com/details?id=org.sipdroid.sipua>

¹¹ <http://www.ffmpeg.org/>

tivesse apenas esse codec habilitado. O FFmpeg é uma biblioteca escrita na linguagem C que implementa diversos codificadores e decodificadores de áudio e vídeo.

3. PRINCIPAIS FUNCIONALIDADES

A figura 1 ilustra as principais funcionalidades do aplicativo, que serão descritas nesta seção.

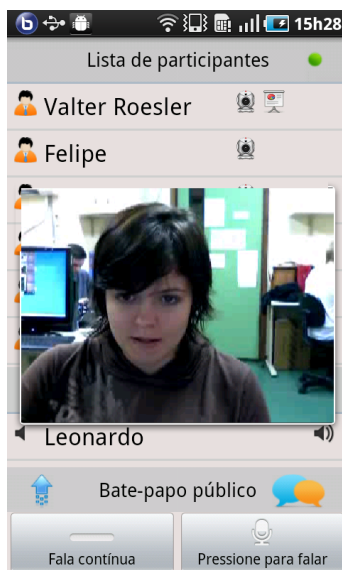


Figura 1: Tela principal do Mconf-Mobile

3.1 Bate-papo

Assim como o cliente web do BigBlueButton, o aplicativo possui a funcionalidade de bate-papo, que pode ser público ou privado. O bate-papo público pode ser observado na figura 1 como uma barra deslizante, que com um toque do usuário encobre a lista de participantes e deixa visíveis todas as mensagens trocadas (juntamente com o originador da mensagem e horário) e uma caixa de texto para envio de novas mensagens.

Já o bate-papo privado é acessado com um toque do usuário sobre o nome do participante com o qual ele deseja iniciar uma conversa. Uma nova tela é aberta, com as mesmas características do bate-papo público, mas com a particularidade de que somente aquele usuário terá acesso à conversa.

3.2 Áudio

O áudio é parte fundamental da interação em um ambiente de videoconferência, logo essa importância é refletida na interface principal do aplicativo. Ao habilitar a conferência de áudio através do botão **Menu** do aparelho, dois botões aparecem na parte inferior da interface (ver figura 1: “Fala contínua” e “Pressione para falar”).

Os botões de controle de áudio são indispensáveis para uma interação de qualidade, pois experimentalmente verificou-se que o microfone de dispositivos móveis é muito sensível a ruído. Além disso, a utilização do botão “Pressione para falar” atenua problemas de eco acústico que podem ocorrer se o usuário não utilizar fones de ouvido.

Através do menu também é possível habilitar o alto-falante do dispositivo, ajustar controle de volume e ganho do microfone, e também encerrar a conferência de áudio.

3.3 Vídeo

O recurso de vídeo, presente no cliente web do BigBlueButton, também é explorado no aplicativo Android. Como pode ser visto na figura 1, um ícone em forma de webcam informa se o participante possui sua câmera habilitada. Um toque longo sobre o nome deste participante faz aparecer um menu com a opção “Exibir vídeo”. Ao selecioná-la, o vídeo do participante é exibido em uma tela sobreposta.

3.4 Gerência de conferência

Caso o usuário entre na sessão como **Moderador**, é possível realizar uma série de operações administrativas, como promover um participante a Apresentador, colocar em mudo um participante que esteja emitindo ruído, abaixar a mão de um participante que pediu atenção e até expulsar um participante da sessão.

Esses recursos sugerem um novo caso de uso para o aplicativo: um professor utiliza o quadro negro para uma exposição aos seus alunos que estão remotamente conectados; ele não permanece em frente ao seu computador durante a exposição, mas pode utilizar um dispositivo móvel como controle remoto para gerenciar a videoconferência (checar mensagens de bate-papo, atender a um pedido de atenção, etc.).

4. CONCLUSÃO

Este artigo apresentou o Mconf-Mobile, um aplicativo para videoconferência em dispositivos móveis com sistema operacional Android. A principal contribuição do projeto Mconf é o desenvolvimento em código aberto de uma solução completa de videoconferência que integra dispositivos móveis e redes sociais. O desenvolvimento modular, conforme apresentado na seção 2, permite que as partes de software produzidas sejam utilizadas em outros projetos de forma independente.

Novas funcionalidades estão previstas para o futuro, como captura de vídeo a partir de um dispositivo móvel, possibilidade de visualização de vários vídeos ao mesmo tempo, integração do módulo de apresentação do BigBlueButton ao aplicativo e desenvolvimento de um novo módulo que permita edição cooperativa de texto através de um bloco de notas.

5. REFERÊNCIAS

- [1] BigBlueButton. Disponível em <http://bigbluebutton.org>. Acessado em julho de 2011.
- [2] Real-Time Messaging Protocol (RTMP) Specification. Disponível em <http://www.adobe.com/devnet/rtmp.html>. Acessado em maio de 2011.
- [3] Top categories on the Android Market. Disponível em <http://www.appbrain.com/stats/android-market-app-categories>. Acessado em julho de 2011.