

IMPLEMENTANDO O PROTOCOLO SIP COMO MÉTODO ALTERNATIVO DE TRANSMISSÃO DE ÁUDIO EM UM SISTEMA DE WEB CONFERÊNCIA DE CÓDIGO ABERTO.

Bruno Amaral da Rosa*
Roberto Wiest**

RESUMO

Web conferências são uma realidade na forma de comunicação atualmente, seja como forma de ensino completo e somente a distância (EAD), ou como uma forma de complemento à educação presencial. Nesse contexto, o software *open source* BigBlueButton surgiu como uma ferramenta completa para criar um ambiente de ensino virtual web, com recursos de áudio, vídeo, chat e slides em tempo real. Por ser *open source*, a solução de problemas ou modificações no sistema fica a cargo dos desenvolvedores e de terceiros que queiram participar do projeto, e um dos problemas do sistema é a não sincronização entre áudio e vídeo dos participantes. Esse artigo pesquisa a implantação de um sistema alternativo de transmissão de áudio para o sistema BigBlueButton, baseado em uma solução de VoIP utilizando o protocolo SIP – *Session Initiation Protocol*. A solução alternativa foi implantada em um sistema em paralelo à instalação padrão do BigBlueButton, e então realizado testes para averiguar a sincronia do áudio com o vídeo. Os resultados obtidos utilizando o protocolo SIP determinaram que é possível reduzir o *delay* em 80% transmitindo o áudio por essa maneira para o sistema BigBlueButton.

Palavras-chave: VoIP. Multimídia. Ensino à distância. Redes de computadores.

1 INTRODUÇÃO

Com a melhora gradual da conexão dos usuários à Internet, e o ensino à distância sendo uma necessidade, ferramentas que tornem possível uma interação aluno-professor tem se tornado foco de vários projetos de pesquisa para desenvolvedores de software. Uma das grandes questões nesse aspecto é: como tornar a experiência *online* o mais próximo possível da experiência de uma sala de aula?

Foi com esse questionamento que, de acordo com o site do projeto, dois desenvolvedores de software da Universidade de Toronto começaram a desenvolver um projeto, de código aberto, em 2007. Esse projeto chama-se “BigBlueButton”. O

*Graduando em Tecnologia em Sistemas para Internet pelo IFSUL campus Passo Fundo. E-mail: brunoadr@gmail.com

**Orientador, professor do IFSUL. Email: Roberto.wiest@passofundo.ifsul.edu.br

BigBlueButton cria um ambiente online, acessível por um computador, onde um professor pode compartilhar para seus alunos recursos de áudio, vídeo, texto, slides e compartilhamento de tela. Dessa maneira, os alunos têm um referencial ao vivo do professor, através de sua imagem e voz, e podem interagir com o professor por meio de um chat.

Segundo artigo publicado e apresentado na Segunda Conferência TICAL (Tecnologias da Informação e Comunicação na América Latina) pelo núcleo de suporte à projeto de áudio e vídeo da UFRGS, o software BigBlueButton vem sendo adotado por diversas universidades, como UFPR, USP e UFRJ, e hoje é projeto patrocinado pela RNP - Rede Nacional de Pesquisa. (MConf, 2012, p 4).

Apesar da proposta do software seja entregar todos os recursos previamente citados, por ser um software em desenvolvimento, ele tem falhas. Falhas que os próprios desenvolvedores do software instigam aos que estão estudando seu código a tentar corrigir.

Esta pesquisa visa corrigir uma falha do sistema que é considerada de alta importância: a disparidade entre o envio de voz e vídeo do transmissor, ou seja, do professor. O método atual de transmissão faz com que a voz do professor chegue atrasado em relação a seu vídeo, o que para os alunos gera um problema se os mesmos tentarem acompanhar o vídeo do professor, pois o que o professor falar não estará em sincronia com o movimento da sua boca ou com gestos que ele possa fazer. É possível corrigir isso usando uma abordagem de um método de transmissão de áudio IP chamado SIP – Session Initiation Protocol. Segundo Keller (2009, p 20) o protocolo SIP foi desenhado para reduzir ao máximo o tempo de transmissão, pois utiliza protocolo de rede UDP¹ para funcionar e codecs² de baixa latência.

O estudo aqui visa, portanto, esclarecer de que forma uma nova tecnologia como o SIP pode corrigir um defeito do BigBlueButton, expondo suas vantagens em relação a tecnologia atual e eventuais desvantagens. Acreditando que softwares para ensino a distância são componente essencial do futuro acadêmico, o trabalho busca ampliar o conhecimento de acadêmicos interessados nessa área de atuação e desenvolvimento

¹ User Datagram Protocol – Protocolo usado por alguns programas em vez do TCP para o transporte rápido, leve e não-confiável de dados entre hosts TCP/IP.

2 DESENVOLVIMENTO

Nesta seção é detalhado o processo de instalação do sistema BigBlueButton, o processo de instalação do software de base para o protocolo SIP, é definido o que são provas de conceito e que softwares serão usados para as provas de conceito, e é realizado uma análise dos resultados obtidos.

2.1 – BigBlueButton

Aberdour (2011, p. 7) define o projeto BigBlueButton como:

BigBlueButton é um sistema de web conferência de código aberto que permite que a criação de salas virtuais, baseadas em um site web, para reuniões e aulas. O objetivo do projeto é atender o ambiente acadêmico, porém existem empresas que dão suporte comercial à plataforma. O BigBlueButton suporta facilidades como compartilhamento de slides (PDF e PPT), vídeo, chat, voz e compartilhamento de desktops. Ele é construído usando mais de quinze componentes de código aberto, roda em Mac, Unix, e PC, e é apoiado por uma comunidade de código aberto que se preocupa com um bom design e uma experiência de usuário simplificada.

O projeto foi iniciado por Fred Dixon na Universidade de Carleton (Canadá) em 2007. Em 2009, o código do projeto foi aberto sob a licença GNU³. Desde que se tornou público, uma comunidade de desenvolvedores ao redor do mundo vem contribuindo ativamente para a melhoria constante do software, que costuma ter uma versão nova lançada por ano (Aberdour, 2011, p. 7).

A versão atual, 1.0, requer para utilização por parte do usuário, um computador com um navegador que tenha suporte a *Flash*. Para poder compartilhar seu áudio e vídeo, é requerido que o usuário tenha uma WebCam e um microfone plugados em seu computador.

Entre as funcionalidades do sistema, estão:

- *Chat*: bate-papo entre os usuários da mesma sala de aula *online*;

² Codec - É o acrônimo de Codificador/Decodificador, dispositivo de hardware ou software que codifica/decodifica sinais.

³ GNU – General Public License – designação da licença para software livre idealizada por Richard Matthew Stallman em 1989.

- Apresentação de slides: possibilidade do tutor da aula *online* carregar apresentações nos formatos mais comuns de arquivos, e apresentá-los aos alunos, com passagem de slides em tempo real;
- Compartilhamento de Áudio e Vídeo: os usuários podem compartilhar sua *webcam* e seu microfone se assim desejarem ou se o tutor na aula permitir;
- Compartilhamento de Desktop: o tutor pode compartilhar seu desktop para demonstração se necessário;
- Gravação: o tutor pode iniciar e parar a gravação da aula, posteriormente é gerado um *link* que pode ser repassado aos alunos que não acompanharam a aula.

2.2 – Implantando o BigBlueButton

A implantação do BigBlueButton é possível somente com um servidor dedicado ao funcionamento do mesmo, pois o sistema diversos módulos, como o servidor web *Apache*, que devem ser únicos e na versão homologada pelo BigBlueButton no servidor. Os requisitos mínimos para a implantação do sistema são (BigBlueButton Docs, 2016):

- Servidor Ubuntu 14.04 64-bit;
- 4 GB de memória RAM;
- Processador com 4 núcleos;
- Portas TCP⁴ 80, 1935, 9123;
- Portas UDP 16384 – 32768;
- Porta 80 não está em uso por nenhuma outra aplicação.
- 500GB de espaço em disco.

É também sugerido que se tenha disponível um link de Internet adequado para a quantidade de usuários irá utilizar o sistema simultaneamente; cada usuário consome, em condições padrão – compartilhando microfone, webcam e slides, 250 Kbps de banda de rede do servidor (BigBlueButton Docs, 2016).

Para a realização dos testes, foi necessário a instalação de dois servidores com o BigBlueButton. O primeiro servidor contém uma instalação padrão do

⁴ TCP - *Transmission Control Protocol* - conjunto de protocolos de comunicação de rede.

BigBlueButton, o segundo servidor, contará com a modificação do protocolo SIP.

Segundo o site de documentação do BigBlueButton (BigBlueButton Docs, 2016), a instalação padrão do BigBlueButton segue etapas pré-definidas por seus desenvolvedores, inicialmente é carregado uma série de repositórios extras (BigBlueButton Docs – Repositórios, 2016) no sistema operacional base Ubuntu 14, e a instalação é então realizada através de pacotes de repositórios particulares do BigBlueButton (BigBlueButton Docs – Repositórios, 2016), dessa maneira não é possível fazer um download de um arquivo instalador do sistema, e o servidor precisa estar conectado à internet para poder ocorrer sua instalação.

O BigBlueButton não possui nenhuma interface de administração, toda a comunicação com o sistema para verificação de credenciais de acesso e níveis de acesso (moderador ou cliente), é realizada através de requisições com a API⁵ nativa do sistema, permitindo a integração com sistemas desenvolvidos em PHP, WordPress, e sistema educacionais como o Moodle. (BigBlueButton Docs, 2016).

No entanto, para fins de testes, é disponibilizado para quem implanta o BigBlueButton uma interface de testes, chamada *bigbluebutton-demo* (BigBlueButton Docs, 2016). Essa interface, contém exemplos em JSP (Java Server Pages) de como criar salas no sistema e como obter dados sobre as conferencias em andamento.

2.3 – Transmissão de áudio no BigBlueButton

Com o sistema base do BigBlueButton implantado, o modelo de transmissão de áudio para o sistema passa a acontecer utilizando o *plugin* Flash Media Player para a captação do áudio (BigBlueButton Docs, 2016).

Para evitar possíveis bloqueios de rede em *firewalls*⁶ o sistema é configurado para enviar o sinal de áudio do participante por “HTTP Tunneling”. Segundo Franco (2009, p. 239), o “HTTP Tunneling” é uma pratica utilizada para burlar bloqueios de Internet ao servidor, porém como o próprio protocolo HTTP (TCP) é um protocolo lento comparado à, por exemplo, o UDP, o *streaming* utilizando HTTP também será bem mais lento, o que ocasiona em uma demora maior para o servidor processar e entregar a voz para os demais usuários. A imagem da webcam do usuário também é processado via

⁵ Application Programming Interface - conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseado na Web.

⁶ Ativos de rede responsáveis pela proteção entra a rede local e a internet.

flash, porém a imagem não é enviada por “HTTP Tunneling”, e sim diretamente via conexão na porta TCP 1935, o que faz com que a imagem seja processada mais rapidamente.

Segundo Franco (2009, p. 53), o codec speech possui uma taxa de amostragem de 120 milissegundos, ou seja, a cada 120 milissegundos um pacote de voz é gerado. Isso é importante porque o BigBlueButton utiliza essa amostragem para gerar um *stream*⁷ do usuário com o servidor.

Uma vez que o usuário solicita o envio do seu microfone para o sistema, o *plugin* do Flash solicita permissão ao usuário em seu navegador, capta o áudio do seu microfone, e cria um *stream* com o servidor utilizando o codec speex.(BigBlueButton Docs, 2016).

Um dos componentes do BigBlueButton é o software que une todos os *streams* de áudio dos participantes – o FreeSwitch. O FreeSwitch é uma plataforma de PABX virtual que suporta diversos tipos de codificação de áudio, como speex, GSM e SIP (Minessale, 2010, p. 2). Uma vez que o *stream* é estabelecido com o Flash Player ao servidor, o FreeSwitch une todos os *streams* que pertençam à mesma sala. Dessa maneira, todos os participantes da sala do BigBlueButton podem conversar por áudio, cabendo ao moderador controlar quem tem direito a fala no servidor. A figura 1 mostra uma lista de usuários de uma sala no BigBlueButton, onde um usuário compartilha recursos de áudio e vídeo:

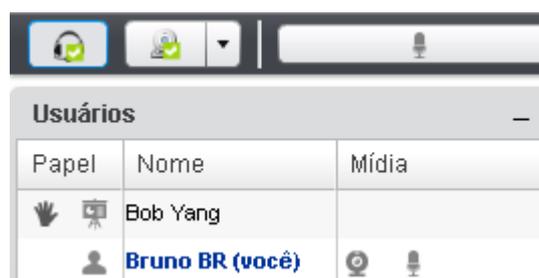


Figura 1 – Usuário compartilhando áudio e vídeo. Fonte: Autor.

2.4 – Implantando o protocolo SIP no BigBlueButton

Com a transmissão de áudio via Flash Player sendo impactada principalmente pela tecnologia “HTTP Tunneling”, onde a voz é transmitida usando o protocolo HTTP,

⁷ Fluxo contínuo de dados.

uma alternativa utilizando SIP – Session Initiation Protocol é proposta.

O software responsável pela montagem da conferencia de voz no BigBlueButton, o FreeSwitch, possui suporte a SIP. Dessa maneira, se o áudio puder ser gerado via SIP no lado do usuário, ele pode ser integrado a sala do BigBlueButton.

Segundo Kurose (2013, p 463), o Protocolo de Inicialização de Sessão (*Session Initiation Protocol – SIP*), é um protocolo aberto e simples, utilizado para transmissão de áudio com um mínimo *delay* – atraso - possível entre dois pontos distintos em uma rede. O SIP, diferentemente do Flash, não pode ser “tunelado” via HTTP, e utiliza portas UDP para trafegar na rede. (Kurose, 2013, p 464). Isso significa que o protocolo SIP não sofre com as mesmas dificuldades do HTTP na rede, o que significa um melhor desempenho e, conseqüentemente, menor *delay*. O codec padrão utilizado pelo SIP é o “PCM lei u”, com uma amostragem de 20 milissegundos. Esse codec é utilizado pois possui boa qualidade e não necessita ser licenciado. A tabela 1 compara os codecs, sua taxa de amostragem, e onde é implementado:

Formato de Áudio	Taxa de Amostragem	Vazão	Aplicação
GSM	120 milissegundos	4,8 kbps	Telefonia Móvel
Speex	120 milissegundos	16 kbps	Flash Player
PCM lei u	20 milissegundos	90 kbps	SIP, H323
G.722	20 milissegundos	160 kbps	SIP HD, Skype

Tabela 1 – Comparação de codecs para transmissão de áudio. Fonte: KUROSE, 2013, p 455.

Com uma taxa de amostragem menor e a utilização de um protocolo não “tunelado” e transmitido via UDP, o SIP se apresenta como uma melhor forma de transmissão de voz do que o Flash. O Flash é um protocolo mais leve e que se adapta melhor a condições adversas da rede, como bloqueios, porém a consequência disso é um desempenho pior em relação ao SIP.

Apesar do FreeSwitch ter suporte a SIP, o BigBlueButton implementa apenas o modulo de conferencia do FreeSwitch (BigBlueButton Docs, 2016). Logo, o FreeSwitch do BigBlueButton não fornece um dispositivo de entrada própria ao sistema, é necessário que a comunicação que for ser estabelecida com o FreeSwitch já esteja montada e codificada em SIP. Para isso, precisamos de um software base SIP para servir como ponto de conexão ao usuário, semelhante com o que acontece quando instalamos o Flash Player no computador do usuário e ele passa a ser usado para capturar o áudio do usuário.

O software livre Asterisk pode ser usado para essa função, o Asterisk é uma implementação de uma central telefônica PBX (*Private Branch Exchange*) em software. O Asterisk na prática funciona como um agrupador, controlador e encaminhador de chamadas de fluxo contínuo. Ele trata o VOIP com os protocolos já apresentados anteriormente, como o SIP (Keller, 2010, p 28).

Com o Asterisk, é possível que a geração do áudio para o BigBlueButton deixe de ser o plugin flash e passe a ser, por exemplo, um ramal telefônico pertencente ao Asterisk. Essa foi a aplicação implantada no Servidor 2 desse artigo.

A figura 2 mostra o diagrama da implantação do Asterisk em conjunto com o BigBlueButton:

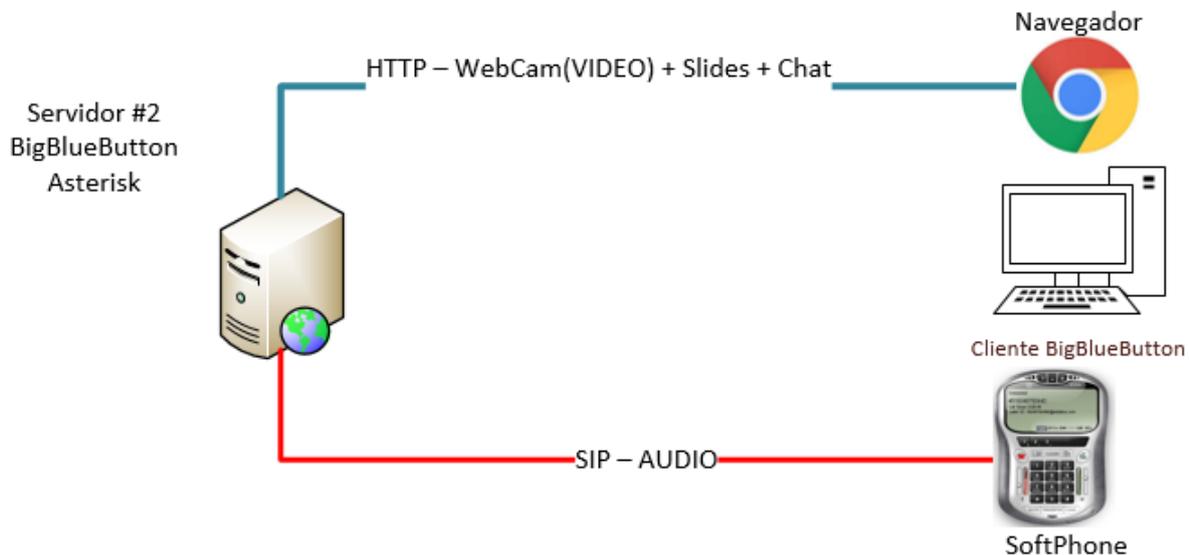


Figura 2 – Caminho alternativo da transmissão do áudio via SIP – Fonte: Autor.

Para a implementação da solução SIP com Asterisk, foi necessário à instalação dos pacotes do Asterisk no servidor do BigBlueButton que será modificado. A instalação padrão do Asterisk pode ser feita através de download do site oficial. A versão utilizada para a implementação foi a 13, através do pacote `asterisk-13-current.tar.gz`. A compilação do pacote é feita com o utilitário do Linux “make”. (Asterisk Wiki, 2016). Uma vez instalado o Asterisk no servidor, ele entrará em conflito com a instalação atual do FreeSwitch, pois, como dito anteriormente, o FreeSwitch também implementa o SIP, e por padrão requisições SIP são feitas na porta TCP 5060 (Keller, 2010, p 44).

O Asterisk contém um arquivo de configuração geral SIP chamado “`asterisk.conf`”, o qual possui a propriedade `port`. Alterando essa propriedade para outro

valor, como 5070, o problema do conflito com o FreeSwitch é resolvido. Como a implementação foi realizada em um ambiente sem NAT, não se leva em consideração nesse artigo que portas possam não estar disponíveis para utilização.

Solucionado o problema do conflito de serviços, pode se criar uma extensão cliente no Asterisk, que é o identificador que será utilizada para se conectar à conferência de áudio gerada pelo FreeSwitch. Após configurado uma extensão única a cada cliente no Asterisk, o Asterisk precisa se comunicar com o FreeSwitch, afim de encaminhar a chamada SIP. Segundo Keller (2010, p 45) o Asterisk pode encaminhar chamada para qualquer outro sistema que receba o protocolo SIP, com ou sem criptografia. Como o BigBlueButton não implementa o acesso seguro à sua plataforma, uma interligação SIP pode ser ativada entre o FreeSwitch e qualquer outro serviço SIP.

No Asterisk, é preciso informar o IP destino do FreeSwitch, bem como os codecs permitidos. Apesar do FreeSwitch aceitar qualquer conexão que tente se estabelecer com ele próprio, seja SIP ou speech (Flash), é necessário alterar alguns parâmetros no Red5.

O Red5 é o software que monta o *streaming* de vídeo e áudio no BigBlueButton, e disponibiliza isso para todos os usuários conectados à interface web do BigBlueButton. O Red5 também é disponibilizado em código aberto e disponível para a integração com qualquer sistema multimídia IP (Red5, 2016). Por padrão, o Red5 vem configurado para somente encaminhar conexões de *loopback*, ou seja, conexões que foram estabelecidas por um cliente previamente conectado ao sistema, como é o caso de um cliente que entrou no sistema pelo seu *browser*, e tentou compartilhar seu microfone utilizando o *plugin* Flash. É necessário alterar o parâmetro que informa o IP 127.0.0.1 (*loopback*) para o IP real do servidor, seja ele um IP válido (WAN⁸) ou um IP falso (LAN⁹). Feito essa alteração, o Red5 passa a aceitar conexões vindas de outros IP, logo a conexão SIP originada pelo Asterisk e montada pelo FreeSwitch, pode ser adicionada à lista de usuários web e ser controlado como mais um usuário do sistema.

Com o Asterisk implantado e interligado com o FreeSwitch, e o Red5 aceitando conexões externas, a comunicação via SIP passa a acontecer. Do lado do cliente, é necessário um software que tenha suporte à SIP, como o *softphone*, software que detecta o microfone do computador onde foi instalado e se conecta à um servidor SIP para poder estabelecer chamadas (Keller, 2010, p 50). Para a validação do ambiente

⁸ Wide Area Network – Internet

⁹ Local Area Network – Rede Local

foi utilizado o *softphone* X-Lite, por ser gratuito e ter suporte amplo ao Windows e suporte ao codec utilizado na integração SIP – “PCM lei u”.

A figura 3 mostra o escopo da interligação do Asterisk com o FreeSwitch e então o encaminhamento para o Red5 e o servidor web: Na API do BigBlueButton existe um parâmetro chamado “voiceBridge”, que é o número da conferencia no FreeSwitch referente a sala que foi criada na mesma chamada da API. É este número que o usuário deve discar em seu *softphone* para se juntar à conferência. A figura 3 mostra a o esquema:

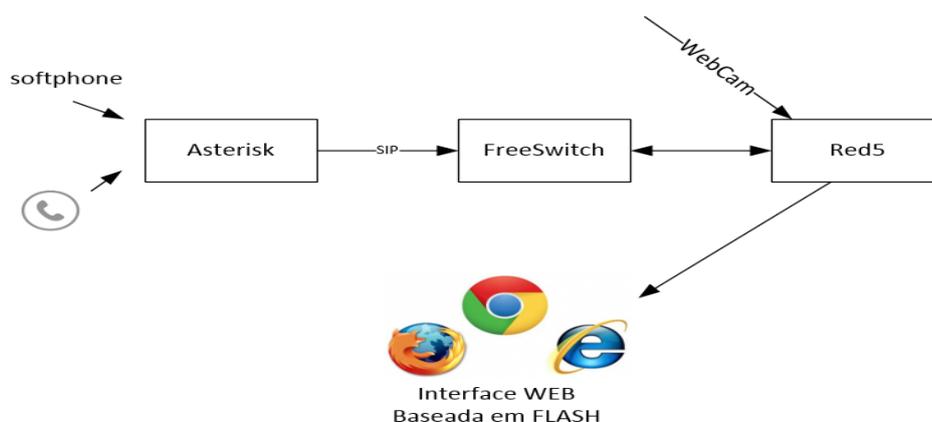


Figura 3 – Sequencia do envio de voz em no BigBlueButton via SIP. Fonte: Autor.

Na API do BigBlueButton existe um parâmetro chamado “voiceBridge”, que é o número da conferencia no FreeSwitch referente a sala que foi criada na mesma chamada da API. É este número que o usuário deve discar em seu *softphone* para se juntar à conferência. A figura 4 mostra a descrição desse parâmetro na documentação oficial do BigBlueButton:

voiceBridge	Optional	String	Voice conference number that participants enter to join the voice conference. The default pattern for this is a 5-digit number. This is the PIN that a dial-in user must enter to join the conference. If you want to change this pattern, you have to edit FreeSWITCH dialplan and <code>defaultNumDigitsForTelVoice</code> of <code>bigbluebutton.properties</code> . When using the default setup, we recommend you always pass a 5-digit voiceBridge parameter. Finally, if you don't pass a value for voiceBridge, then users will not be able to join a voice conference for the session. (see FAQ entry Users do not appear in listeners window).
-------------	----------	--------	---

Figura 4 – Descrição do parâmetro voiceBridge – API BigBlueButton

A figura 5 mostra o mesmo usuário com duas sessões, a primeira indica a presença Web do usuário em seu navegador, compartilhando recurso de vídeo através de sua webcam, e o segundo usuário indica o *streaming* SIP gerado por um *softphone*:



Papel	Nome	Mídia
	Bruno (você)	
	Bruno Softphone	🔊 🔒

Figura 5 – Usuários distintos para Web e SIP. Fonte: Autor.

2.5 – Provas de Conceito.

Implantado a solução com o protocolo SIP, é necessário aferir a diferença entre o sistema padrão e o sistema modificado. É necessário medir, em segundos e milissegundos, o atraso no envio do áudio dos dois sistemas, para então compara-los e averiguar se há melhoria na solução SIP. Para isso, foram usados 2 softwares: Wireshark – Software de análise de rede para determinar quando o *streaming* de pacotes de áudio começa no servidor; Audacity – software de análise de som para determinar a diferença no *delay* entre um *streaming* de áudio com Flash e um com SIP.

Para realizar os testes, um ambiente em LAN foi montado, com dois servidores, 1 com uma instalação padrão e outro com implantação do SIP, e dois computadores com o navegador Google Chrome acessaram os servidores, o primeiro fazendo o papel do professor, o qual compartilha sua webcam e microfone, e o segundo computador no papel do aluno, que recebe os recursos multimídia do professor. As medições foram realizadas todas no computador do aluno, visto que é quem assiste a aula que nota o atraso do áudio em relação ao vídeo. A figura 6 e a tabela 2 apresentam o mapeamento dessa implantação:

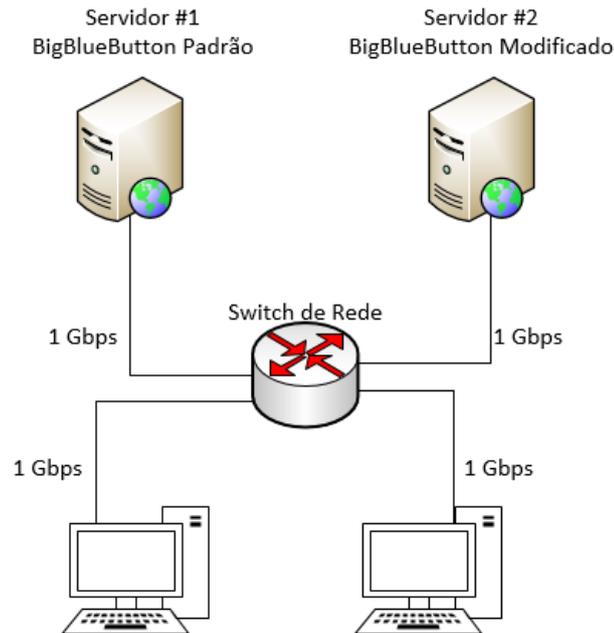


Figura 6 – Mapeamento dos componentes utilizados nos testes.

ATIVO	IP	SISTEMA OPERACIONAL	NAVEGADOR	MULTIMIDIA	CONEXÃO
Servidor BBB1	10.1.1.10	UBUNTU SERVER 14	N/A	N/A	1 Gbps
Servidor BBB2	10.1.1.20	UBUNTU SERVER 14	N/A	N/A	1 Gbps
PC - Gerador/Professor	10.1.1.30	WINDOWS 10	FIREFOX	WebCam + Microfone	1 Gbps
PC - Receptor/Aluno	10.1.1.40	WINDOWS 10	FIREFOX	N/A	1 Gbps

Tabela 2 – Descritivo dos componentes utilizados nos testes.

No ambiente de testes, o computador que gera o áudio e o vídeo já estava conectado a sala do BigBlueButton compartilhando esses recursos. No computador onde foi feita a medição, o computador do aluno, os softwares irão registrar os dados a partir do momento que o usuário entrar na sala de aula virtual. Dessa maneira foi possível medir o tempo exato que o usuário começa a receber dados do servidor.

2.5.1 – Usando o Wireshark para analisar o tráfego de rede.

Segundo Orebaugh (2006 p. 4), o Wireshark é um software para captura de pacotes de rede, análise de tráfego de rede e decodificação de tráfego de rede não criptografado. Como o tráfego direcionado ao BigBlueButton não é criptografado, o Wireshark consegue separar, por tipo de tráfego, pacotes de voz, áudio, ou texto puro, como HTML.

Para o teste com o servidor BigBlueButton padrão, foi executado o Wireshark no computador do aluno, com o *stream* do professor já ativo no servidor. A captura de pacotes foi ativada logo após a requisição de *login* na sala do BigBlueButton. Após a conexão, é encerrada a captura de pacotes, e aplicado filtros para identificar qual o primeiro pacote de áudio e qual o primeiro pacote de vídeo a trafegarem, e seus respectivos horários. A tabela 3 mostra os resultados:

Packet ID	Timestamp	SourceAdd	DstAdd	Protocol	Transport	Port	Application
735	28.407001	192.168.88.100	192.168.88.10	RTMP	TCP	1935	Video Data
825	30.107060	192.168.88.100	192.168.88.10	HTTP(Tunnel)	TCP	80	Áudio Data

Tabela 3 – Teste 1 - Resultados do filtro aplicado a captura de pacotes do Wireshark. Fonte: Autor.

A tabela 3 mostra que o *stream* de vídeo, proveniente da webcam do professor, começa a enviar dados ao computador do aluno no tempo 28.407001 da captura, enquanto que o *stream* de áudio começa a enviar dados para o mesmo computador no tempo 30.107060. Isso gera uma diferença, ou atraso, de aproximadamente 1,70 segundos. Esse atraso é perceptível na utilização do sistema, o que gera o problema que propõe-se resolver nesse artigo.

O mesmo teste foi realizado dessa vez com os computadores do professor e do aluno conectados a sala do BigBlueButton com a implantação do protocolo SIP. Para este teste, no computador do professor, foi necessário que além do navegador Google Chrome fosse carregado um *software softphone* SIP, para a captura da voz e envio ao Asterisk.

A tabela 4 mostra o resultado dos testes:

Packet ID	Timestamp	SourceAdd	DstAdd	Protocol	Transport	Port	Application
663	29.703033	192.168.88.200	192.168.88.10	RTMP	TCP	1935	Video Data
654	30.003029	192.168.88.200	192.168.88.10	SIP	UDP	5060	SIP Stream

Tabela 4 - Teste 2 - Resultados do filtro aplicado a captura de pacotes do Wireshark. Fonte: Autor

A tabela 4 mostra que a diferença entre o início do *stream* de vídeo e do *stream* de áudio, agora via SIP, caiu para aproximadamente 0,3 segundos, uma diferença de 1,4 segundos.

2.5.2 – Usando o Audacity para analisar o espectro de áudio.

O Audacity é um editor e gravador de áudio distribuído gratuitamente sob a licença GNU General Public License (Bohn, 2009, p. 4). É possível gravar com o Audacity não somente o microfone conectado à um computador, mas também o que é executado pelos autôfalantes. Da mesma maneira que o teste que foi executado com o Wireshark, o *stream* do computador já estava ativo, e o computador do aluno então se conecta à sala do BigBlueButton. A partir do *login*, o Audacity começa a gravar a saída de áudio do computador. Para auxiliar nos testes, um *beep* contínuo de 1 em 1 segundo é executado no microfone do professor. A figura 7 demonstra o resultado dos testes:

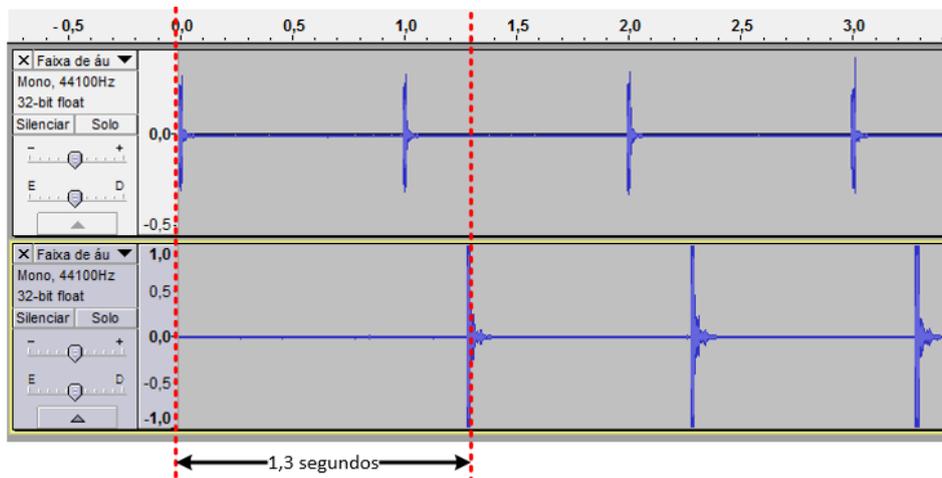


Figura 7 – Análise de espectro de som. Fonte: Autor.

Na figura 6, o primeiro espectro pertence a captura de áudio em SIP, a segundo pertence a captura de áudio em Flash. A gravação inicia assim que o computador do aluno ouve o primeiro *beep*. Existe uma diferença de 1,3 segundos de atraso em relação ao tempo que o *beep* é ouvido no *stream* SIP e no *stream* Flash.

3 – Considerações Finais

O objetivo maior foi a redução total ou quase total do *delay* entre áudio e vídeo no BigBlueButton. Pouco adiantaria reduzir um pouco o *delay*, uma vez que a falta de sincronia em um ambiente multimídia começa a ser percebida em milésimos de segundo.

Com a abordagem em SIP, juntamente com o Asterisk e *softphone*, foi possível reduzir o *delay* de 1700 milissegundos para 300 milissegundos. Nos testes, 300 milissegundos são detectados somente com muita atenção ao movimento dos lábios de quem está falando, portanto é seguro dizer que o sistema conseguiu um valor próxima da sincronia, com uma redução expressiva no *delay*. Nos testes de bancada, a percepção dos usuários do sistema com a modificação foi de uma melhora significativa.

Uma sincronia total somente seria possível se a captura de áudio e vídeo acontecesse pelo mesmo *stream*, porém não é dessa forma que o BigBlueButton é desenhado para funcionar, visto que o motor de funcionamento é baseado em Flash, o que gera o problema atual de funcionamento.

Desenvolvedores do BigBlueButton já buscam atualmente utilizar tecnologias com o HTML5 para resolver esses problemas (BigBlueButton Docs – HTML5, 2016), e um estudo aprofundado sobre novas tecnologias por certo trará outras soluções, que poderão ser implantadas no BigBlueButton como uma solução final ao problema do *delay*, não como no caso da implementação SIP que esse artigo propõe, que é uma adaptação ao BigBlueButton. O SIP por certo fará parte da solução final do *delay*, porém como método de transporte de áudio transparente, não dependendo de softwares terceiros, como o *softphone*.

Para trabalhos futuros, a questão mais latente fica na usabilidade do usuário, quando utiliza a solução SIP. Na solução proposta por esse artigo, é necessário que o usuário instale um *softphone* para transmitir seu áudio. Enquanto um navegador web é padrão existir em qualquer computador, um *softphone* demanda uma instalação a mais no computador e uma configuração desse software, o que torna a utilização do sistema BigBlueButton não mais uma experiência única ao navegador. Existem bibliotecas em Java que utilizam o SIP, e poderiam ser integradas no cliente Web do BigBlueButton, assim não necessitando mais o uso do *softphone* e tornando a experiência do usuário novamente única ao navegador.

IMPLEMENTATION OF THE SIP PROTOCOL AS A ALTERNATIVE WAY OF AUDIO TRANSMISSION IN A OPEN SOURCE WEBCONFERENCE SYSTEM.

ABSTRACT

Online educational environments is now a reality when it comes to a way of teaching, be that in the shape of full time e-learning, or as a way of complementing in site teaching. In this context, the open source software BigBlueButton comes as a complete tool to create an online educational environment, with audio, video, chat and slides resources in real time. Because it's open source, the troubleshooting of bugs or modifications in the system is also accounted to the online community, and one of the problems of the system is that the audio and the video of the participants are not synchronized. This article describes the implementation of an alternative system to transmit audio to BigBlueButton, based on a VoIP solution utilizing the SIP protocol. The alternative solution was implemented in a parallel system to a standard BigBlueButton installation, then tests were made to ensure that audio and video were synchronized. The results obtained by using the SIP protocol determined that is possible to reduce the *delay* by transmitting the audio this way to a BigBlueButton system.

Keywords: SIP. VoIP. Online learning. Multimedia.

REFERÊNCIAS

- ABERDOUR, M et, al – 2011: Virtual classrooms: an overview. Disponível em <[http://www.cedmaeurope.org/newsletter%20articles/Kineo/Virtual%20Classrooms%20Overview%20\(Feb%202011\).pdf](http://www.cedmaeurope.org/newsletter%20articles/Kineo/Virtual%20Classrooms%20Overview%20(Feb%202011).pdf)> Acesso em: 02 jun. 2016.
- Asterisk Wiki. 2016 – Disponível em <<https://wiki.asterisk.org>> Acesso em: 05 mai 2016.
- BigBlueButton Docs. 2016 – Disponível em <<http://docs.bigbluebutton.org>> Acesso em: 04 mai 2016.
- BigBlueButton Docs – HTML5 2016 – Disponível em <<http://docs.bigbluebutton.org/html/html5-overview.html>> Acesso em: 04 mai 2016.
- BigBlueButton, una alternativa de código abierto para la comunicación interactiva en actividades educativas – Disponível em <<http://www.aenui.net/jenui2014/32.pdf>> Acesso em: 20 mai. 2016.
- BOHN, Vanessa Cristiane Rodrigues – 2009: Seja Audacity na criação de Podcasts: a união do Software livre e da Web 2.0 no ensino de língua estrangeira. Disponível em <<http://www.periodicos.letras.ufmg.br/index.php/textolivre/article/view/19/7316>> Acesso em: 15 mai. 2016.
- FRANCO, Carlos Eduardo G. **Flex 3 + Flash Media Server 3.5**. 1. Ed. Rio de Janeiro: Brasport, 2009
- KELLER, Alexandre. **Asterisk na prática**. 1. Ed. São Paulo: Novatec Editora, 2009.
- KUROSE, James F. **Redes de computadores e a Internet: uma abordagem top-down**. 6. Ed. São Paulo: Pearson Education do Brasil, 2013.
- Mconf: sistema de multiconferência escalável e interoperável web e dispositivos móveis -http://mconf.org/m/wp-content/uploads/2012/08/Mconf_Conferencia_TICAL2012.pdf

MINESSALE, A, et al. **FreeSwitch 1.0.6**. 1 Ed. Reino Unido: Packt Publishing, 2010.

OREBAUGH, A et al. **Wireshark & Ethereal network protocol analyzer toolkit**. 1 Ed. Estados Unidos: Syngress, 2006.

Red5. 2016 – Disponível em <<https://red5.org>> Acesso em: 05 Jul 2016.