

MCU em software distribuído em nuvem

Valter Roesler
Instituto de Informática, UFRGS –
Universidade Federal do RGS
Porto Alegre, RS, Brasil
roesler@inf.ufrgs.br

Mário Gasparoni Júnior
Instituto de Informática, UFRGS –
Universidade Federal do RGS
Porto Alegre, RS, Brasil
mariogasparoni@gmail.com

Ronaldo Husemann
Instituto de Informática, UFRGS –
Universidade Federal do RGS
Porto Alegre, RS, Brasil
rhusemann@inf.ufrgs.br

Roberto Irajá T. da Costa Filho
Instituto de Informática, UFRGS –
Universidade Federal do RGS
Porto Alegre, RS, Brasil
rtcosta@gmail.com

Rafael Valle
RNP – Rede Nacional de Ensino e
Pesquisa
Rio de Janeiro, RJ, Brasil
rafael.valle@rnp.br

ABSTRACT

The presented tool is a fully virtualized videoconferencing MCU (*Multipoint Control Unit*) system using the standard SIP (*Session Initiation Protocol*). The proposed tool works in the cloud in a scalable way, with low deployment and maintenance costs. In addition, the proposed tool is more than an MCU, functioning as a universal framework for media forwarding.

KEYWORDS

Multimedia; MCU; SFU; videoconference; web conference; video collaboration.

1 INTRODUÇÃO

Hoje em dia as instituições utilizam soluções de videoconferência e videoconferência como uma forma de redução de custos operacionais (viagens, por exemplo) e também aumento na velocidade na qual a interação é efetuada. Atualmente o mercado vê um crescimento no uso de videoconferência, provocado muito pelo declínio do custo de ferramentas de videoconferência baseadas em software, como Microsoft Skype for Business e WebEx, que aproveitam equipamentos já existentes na empresa, como laptops, tablets e smartphones (equipados com câmeras). Essas soluções em software custam significativamente menos que as equivalentes em hardware, ampliando o acesso [1]. O modelo de salas de videoconferência onde várias pessoas participam juntas da interação é muito utilizado atualmente, em paralelo com o modelo de uso em dispositivos pessoais. O motivo é que, muitas vezes, a interação em um ambiente único em cada ponto facilita a comunicação. O fato é que existem diversos modelos atualmente para efetuar videoconferência, e diversas ferramentas para isso [2].

Ferramentas de videoconferência são úteis em comunicações ponto a ponto, com dois interlocutores, e também em

comunicações multiponto, com vários interlocutores. Para comunicações multiponto, um dos modelos existentes é o de MCU (*Multipoint Control Unit*), que permite a comunicação entre diversos participantes numa videoconferência, com fluxo de saída composto pelas mídias de diversos equipamentos finais de videoconferência (também chamados de *endpoints*) [3].

Os *endpoints* podem ser compostos por sistemas de hardware ou de software. Exemplos de *endpoints* de hardware são a linha HDX da Polycom [4], o Multipoint Bridge da Lifesize [5], e a linha MX da Cisco [6], entre outros. Exemplos de *endpoints* de software são o Jitsi [7], o Ekiga [8] e o RealPresence da Polycom [9].

Para interconectar os diversos *endpoints*, um MCU age como elemento centralizador, recebendo todas as mídias, compondo todas em um mesmo vídeo, e enviando esse vídeo para todos *endpoints*, como mostra a Figura 1. Observa-se que, em termos de rede, o MCU recebe um fluxo de mídia (áudio + vídeo) de cada usuário, e envia um fluxo de mídia composta (áudio mixado + vídeo composto) para cada participante.

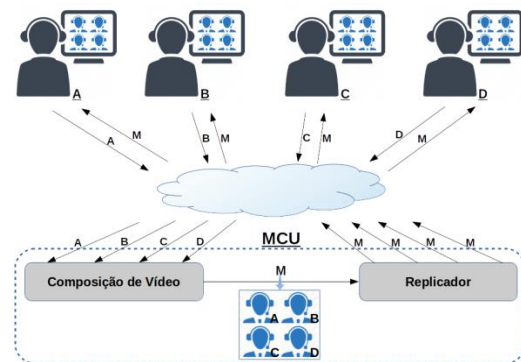


Figura 1: Funcionamento básico de um MCU

Essa arquitetura centralizada onera o servidor MCU, que precisa processar (decodificar e/ou transcodificar) os fluxos de todos *endpoints* e compor (codificar) o fluxo de saída. Devido à elevada carga de processamentos de vídeo que um MCU necessita atender, os equipamentos de MCU são produtos

bastante especializados, compostos normalmente por hardwares de DSPs (*Digital Signal Processors*) ou por módulos específicos de software instalados em servidores COST (*commercial off-the-shelf*), especialmente disponíveis e dedicados à decodificação e codificação de vídeo e áudio. Essa complexidade torna os produtos comerciais de MCU bastante caros para aquisição, e também caros para manutenção [10].

É possível fazer MCUs mais baratos, em software, e existem várias iniciativas para isso, como o projeto OpenMCU-ru (<https://videowitch.ru/eng.htm>), que apresenta uma plataforma completa em código aberto de um MCU. O problema de MCUs em software é que, por rodar em cima de um processador não especializado (genérico), o número de *endpoints* que uma máquina consegue atender é limitado, menor do que as soluções dedicadas, conforme será detalhado adiante neste documento. Esse fato se agrava quando se trabalha com qualidade HD.

A proposta deste artigo é apresentar um sistema de MCU em software que permite escalabilidade na nuvem, mantendo o custo de aquisição e manutenção baixo, por se tratar de software, mas permitindo escalabilidade, suportando um número muito grande de usuários, de acordo com o número de máquinas virtuais alocadas para o sistema.

A seção 2 apresenta uma visão geral da ferramenta, mostrando seus diferenciais. A seção 3 mostra sua arquitetura interna, detalhando ainda mais seu funcionamento. A seção 4 mostra alguns trabalhos relacionados. As seções 5 e 6 mostram o ambiente experimental de validação e os resultados obtidos.

2 ARQUITETURA GERAL DA FERRAMENTA

A Figura 2 mostra a arquitetura geral da ferramenta proposta, que é composta por um Gerenciador de Escalabilidade (GE) redundante e uma rede de Servidores de Mídia (SM) de MCUs em software, distribuídos na nuvem. A redundância do GE é necessária para que o sistema possua alta disponibilidade, visto que o GE gerencia toda a sinalização, e possui a inteligência de todas as conferências em andamento, tornando-se um elemento crítico no sistema. A redundância dos Servidores de Mídia é necessária para, além de fornecer alta disponibilidade, aumentar o poder de processamento do sistema, suportando mais *endpoints*. Os *endpoints* conectam com o GE (linhas pretas na figura), que é, basicamente, o ponto de entrada do MCU para os usuários. O GE escolhe um Servidor de Mídia disponível e direciona este usuário (*endpoint*) para ingresso na sala desejada, fazendo com que toda a transferência de mídia (vídeo e áudio) seja feita agora diretamente entre SM e *endpoint* (ver figura). A figura também representa várias salas independentes com vários usuários em cada sala, todos se comunicando através do MCU proposto.

Em termos de características gerais, a ferramenta contempla:

1. Funcionamento em software, em máquina virtual, de forma distribuída em nuvem.
2. Escalável através de um gerenciador de escalabilidade, permitindo diversas conferências simultâneas.
3. Baixo custo de implantação e de manutenção.

4. Funcionamento como um Servidor de Mídia Universal na rede, permitindo receber demandas de envio de áudio e vídeo e encaminhar para o(s) destino(s) desejado(s) no formato definido.
5. Alta robustez e disponibilidade para uso contínuo, ou seja, a ferramenta passou por baterias de testes de carga e funcionamento contínuo, provendo para o usuário uma experiência funcional, sem interrupções ou reinicializações.

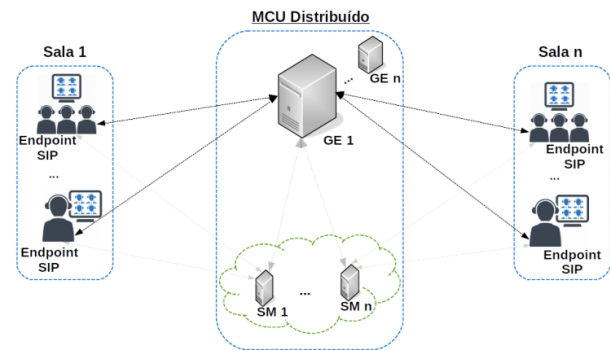


Figura 2: Visão geral do sistema MCU em nuvem.

3 ARQUITETURA INTERNA DO MCU

A Figura 3 exibe a arquitetura do MCU, mostrando os principais blocos do Gerenciador de Escalabilidade, resumidos a seguir:

- **MCS-SIP (*Media Control Server – Session Initiation Protocol*)**: atua como receptor das requisições SIP, traduzindo-as em chamadas de API para o servidor MCS. Além de traduzir as mensagens SIP para a API própria, esse bloco envia mensagens *SIP Redirect* para o *endpoint*, a fim dele trocar a mídia diretamente com o SM designado.
- **MCS (*Media Control Server*)**: corresponde a uma aplicação e um framework para desenvolvimento de aplicações multimídia. Uma de suas principais funções é fazer o balanceamento de carga dos servidores de mídia disponíveis. Além disso, o MCS define um modelo de abstração que é independente do tipo de comunicação de mídia realizado (RTP, WebRTC, etc.) permitindo a gerência de salas, usuários e mídias. Dessa forma, um usuário de uma webconferência (acessando uma sala de MCU através do navegador WEB de um PC) é capaz de se comunicar com um usuário de videoconferência (acessando a mesma sala de MCU através de um *endpoint* de videoconferência).
- **Gerente**: através de uma interface web, o Gerente permite o acesso Federado para controle do sistema. Esse módulo permite a criação de salas, desde que o usuário tenha permissão para isso. Além disso, cada sala tem um gerente local (moderador), responsável por mudanças de layout, início de gravação, expulsar um usuário, colocar em mudo, entre outras. Esse módulo também permite a gerência local da sala. A comunicação com o MCS é através da mesma API mencionada anteriormente.
- **BD Distribuído**: todos os dados instantâneos de todas as conferências em andamento (nome da sala, quem está na

sala, pipelines, entre outros) ficam armazenados em um banco de dados distribuído a fim de aumentar a disponibilidade do sistema. Caso uma VM de GE fique indisponível, outra VM de GE assume, e utiliza os dados instantâneos que estão armazenados no BD distribuído, que é um elemento crítico do sistema para manter a alta disponibilidade.

- **Monitor:** atua como um servidor que tem conhecimento do estado atual da nuvem de servidores de mídia. Seu papel é manter uma conexão com todos os agentes (ou clientes) monitores que estão presentes em cada uma das máquinas Servidores de Mídia (SMs) da nuvem, bem como receber, armazenar e informar o MCS sobre novos eventos detectados por esses agentes. Essa informação é fundamental para que o Balanceador de Carga (Gerenciador de Escalabilidade) consiga tomar uma decisão sobre qual Servidor de Mídia uma nova sala de videoconferência deve utilizar.

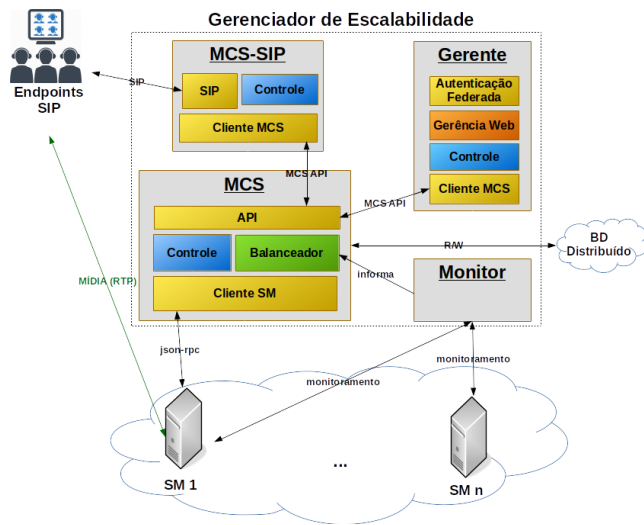


Figura 3: Esquema geral da arquitetura proposta

O padrão de sinalização proposto para o produto é o SIP (*Session Initiation Protocol*), permitindo a compatibilidade global do MCU proposto com os *endpoints* de mercado existentes.

Como pode-se perceber, a arquitetura proposta é facilmente escalável, totalmente baseada em máquinas virtuais na nuvem (tanto o GE e Banco de Dados como todos os SMs), permitindo futuras atualizações, troca de versões, aumento ou redução na capacidade do sistema, entre outras configurações. Isso torna o ambiente facilmente gerenciável caso seja oferecido como serviço por alguma NREN.

Adicionalmente à facilidade de gerência, a proposta de produto do GT-MCU é permitir acesso para a criação de salas virtuais de videoconferência de forma Federada. Dessa forma, a operação fica ainda mais facilitada, pois basta criar a regra e as próprias pessoas gerenciam suas salas.

A arquitetura proposta se destaca pois foi projetada de forma modularizada, com o intuito de facilitar a adição de novos componentes (por exemplo, módulos ou *plugins* do GStreamer,

ou novas aplicações desenvolvidas sobre o *framework*) que gerenciam e trabalham diretamente sobre as mídias (áudio, vídeo e conteúdo), independentemente do protocolo de sinalização das chamadas (SIP, H.323 ou novos que venham a surgir futuramente). Isso possibilita, por exemplo, que os vídeos de um ou mais usuários sejam interceptados e tratados antes e/ou após a sua composição, podendo se aplicar técnicas de visão computacional caso necessário (ex.: reconhecimento facial para identificação de pessoas, detecção de face para o posicionamento automático de câmeras frontais, etc).

Tecnologias Utilizadas

O sistema operacional utilizado na solução MCU é o Linux Ubuntu, que possui uma comunidade muito grande de desenvolvedores e dispensa maiores detalhes.

O principal componente do servidor de mídia é o software Kurento¹, que funciona através da licença LGPL v 2.1.

Para a decisão de utilização do Kurento como elemento principal do bloco Servidor de Mídia, o grupo comparou o mesmo com as seguintes soluções: a) FreeSwitch (<https://www.freeswitch.org/>); b) Licode (<http://lynckia.com/licode/>); c) Medooze (www.medooze.com/); e d) OpenMCU-ru (<https://videowitch.ru/eng.htm>), tomando a decisão pelo mesmo em função da flexibilidade que a solução permite, habilitando tanto questões de MCU como SFU, e integrando facilmente com WebRTC.

O projeto Kurento existe desde 2012, surgindo dentro do Laboratório FUN-LAB (*Future Networks Laboratory*, ou Laboratório de Redes do Futuro) da Universidade Rey Juan Carlos (www.urjc.es) e com o objetivo de se tornar um servidor de mídia de código aberto com suporte a diferentes tipos de tecnologias e processamento de imagens.

Em termos de arquitetura, o Kurento é desenvolvido sobre a plataforma GStreamer (<https://gstreamer.freedesktop.org/>), que corresponde a um *framework* para desenvolvimento de aplicações multimídia, já utilizado em diversas aplicações Desktop e aplicações embarcadas, além de ser compatível com diferentes tipos de sistemas operacionais.

Outro software integrado no âmbito da proposta é o sistema de monitoramento Zabbix (www.zabbix.com/). O Zabbix é uma plataforma de código aberto para monitoramento escalável de infraestruturas de TI. Também realiza monitoramento baseado em agentes, podendo ser configurado como agente nas VMs dos Servidores de Mídia (SMs). Conta com uma interface web robusta, possui desenvolvimento e suporte ativo, sendo utilizado por diversas empresas consolidadas no mercado, como a Dell, Renner, ICANN, dentre outras.

O componente SIP utilizado para executar o padrão SIP é o SIP.js (<https://sipjs.com/>), que é uma biblioteca SIP em javascript também de código aberto. Essa biblioteca foi escolhida pois já possui várias sinalizações do protocolo SIP, evitando retrabalho, agilizando o desenvolvimento e garantindo mais compatibilidade.

¹ <https://www.kurento.org/>

4 TRABALHOS RELACIONADOS

Alonso [10] argumenta que um MCU é uma entidade que gerencia diferentes aspectos de sistemas multimídia, como a Mixagem, Encaminhamento, Gravação e Transcodificação de fluxos de mídia. Em seguida, apresenta cenários de uso de MCU em nuvem, e vantagens do modelo. Ele descreve os seguintes usos avançados de MCUs:

- **Streaming:** seria o uso do MCU para receber um fluxo de áudio e vídeo de uma fonte de origem (*publisher*) e , enviar o mesmo para vários assinantes.
- **Composição:** ao gerar um único fluxo de vídeo ou áudio a partir dos fluxos de entrada, o MCU reduz a sobrecarga nos *endpoints* e a banda na rede.
- **Transcodificação:** permite adaptar o fluxo de saída do MCU para diferentes necessidades de *endpoints*, como banda reduzida ou uma menor resolução, o que garante adaptação à heterogeneidade de dispositivos e redes de acesso.
- **Gravação:** permite manter o histórico das reuniões efetuadas.

Posteriormente, Alonso apresenta dificuldades de sistemas na nuvem, como a gerência da escalabilidade vertical (muitos numa sala) e horizontal (muitas salas). Ele apresenta também algumas vantagens dos sistemas em nuvem, como a distribuição geográfica, que permite o uso de uma nuvem próxima ao local da conferência sendo realizada.

Akkus [3] apresenta uma solução de multiponto que busca distribuir a carga do sistema utilizando *endpoints* como entidades capazes de gerir e processar localmente recursos de multimídia. Isso pode ser conseguido explorando peer-to-peer (P2P) ou estratégias de encaminhamento seletivo (SFU).

Kesavaraja [12] apresenta um conceito que está crescendo atualmente, que é o de oferecer serviços de videoconferência na nuvem (VaaS – *Video as a Service*).

Sistemas baseados em VaaS têm sido muito procurados para prover recursos de armazenamento, segurança e conferência. Armazenamento é a aplicação mais tradicional. A segurança desponta como preocupação recente, uma vez que provedores de serviços de multimídia têm sido alvos frequentes de hackers. Na videoconferência em nuvem, ainda são necessários software e hardware locais (*endpoints* providos de câmeras), mas os aplicativos de videoconferência que gerenciam as conferências e os dados que eles geram residem em algum ponto da rede (ou seja, na "nuvem"). Com isto se ganha em flexibilidade e mobilidade [13].

É importante destacar, entretanto, que aplicações de videoconferência baseados em VaaS, apesar de desejadas, ainda são emergentes, visto necessitarem gerência muito robusta para que não se mostrem sensíveis a flutuações e/ou congestionamentos na rede [14].

Rob Scott [15] apresenta um conceito de VGaaS (*Video Gateway as a Service*), onde a nuvem oferece um elemento central que converte protocolos entre si, permitindo a interação entre vários padrões, como SIP, WebRTC, ISDN, entre outros. Ele propõe o conceito de BYOC (*Bring Your Own Codec*), permitindo

unir numa mesma reunião várias pessoas sem se preocupar com qual equipamento, software ou codec cada um está utilizando.

Como pode ser visto através dos trabalhos relacionados, o conceito de utilização da nuvem como um elemento que efetua o papel de MCU ou de conversor de protocolos (gateway) está crescente atualmente.

5 RESULTADOS

Em termos de resultados, a ferramenta está funcional e sendo utilizada pelo grupo, em fase de início de projeto piloto com diferentes instituições. São efetuadas reuniões semanais do grupo de forma remota, através do MCU desenvolvido. Essa metodologia foi muito importante para encontrar erros no sistema, que foram corrigidos.

A Figura 4 mostra uma videoconferência efetuada pelo grupo, mostrando o layout com 5 participantes. Foram feitos experimentos com participantes em Brasília e até mesmo na Espanha, obtendo-se baixo atraso.

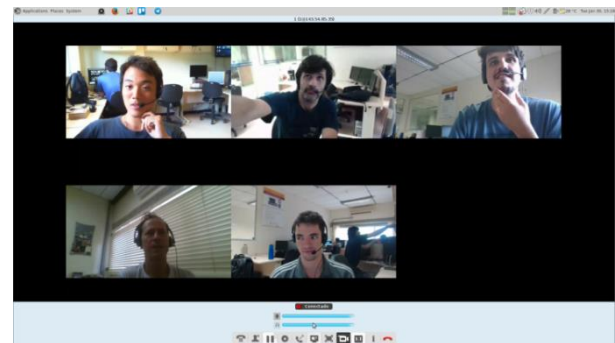


Figura 4: Videoconferência do grupo vista através do *endpoint* de software Jitsi

A Figura 5 mostra um resultado com o uso de um *endpoint* de hardware da Polycom. As quatro imagens representam o vídeo do Polycom (em cima à esquerda), um participante na mesma sala via Jitsi (embaixo à esquerda), e dois BOTs (à direita).



Figura 5: Videoconferência com um *endpoint* Polycom, um *endpoint* de software Jitsi e dois BOTS.

5.1 TESTES DE CARGA

Foram efetuados diversos testes de carga no sistema medindo CPU, memória, I/O, armazenamento, entre outros. Observou-se que o principal recurso computacional que oferece uma limitação ao número de usuários simultâneos é a CPU, visto a complexidade de codificação e decodificação de vídeo existente num MCU, conforme detalhado anteriormente. Fatores como memória e I/O não são afetados sensivelmente pelo processamento de vários vídeos.

É importante observar, também, que o ponto crítico de uso de CPU no sistema é nas máquinas virtuais (VMs) dos Servidores de Mídia (SM), e não do Gerenciador de Escalabilidade (GE). Isso se deve ao fato do GE trabalhar mais com sinalização (não onerando muitos recursos da máquina), e o SM trabalhar fortemente com codificação e decodificação de vídeo e áudio dos participantes das salas virtuais, onerando bastante os recursos computacionais das VMs.

Com base nisso, a Figura 6 apresenta o resultado obtido para o teste de carga específico de uso de CPU em uma sala virtual num dos servidores de mídia, tendo o número de vídeos sendo aumentado até 9. A partir desse ponto, a máquina somente recebia o vídeo do *endpoint*, mas o mesmo não entrava na composição. Obteve-se um limite aproximado de 25 usuários mantendo a qualidade inicial. Acima disso os vídeos começavam a diminuir o número de quadros por segundo (*frame rate*), diminuindo a qualidade da videoconferência.

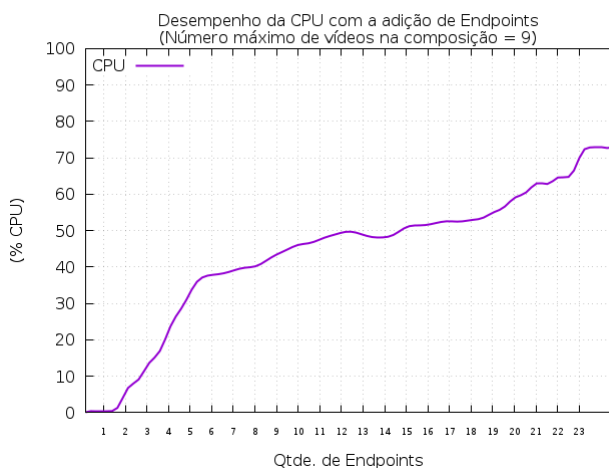


Figura 6: Resultado do teste de carga.

6 CONSIDERAÇÕES FINAIS

Este artigo apresentou um sistema de videoconferência do tipo MCU. Esse sistema já está em uso em forma de protótipo experimental na RNP. A arquitetura do MCU proposto é completamente inovadora em relação a MCUs tradicionais em hardware, possuindo funcionamento virtualizado em nuvem com possibilidade de escalabilidade. O fato de funcionar em ambiente virtualizado facilita muito a implantação e manutenção, diminuindo drasticamente os custos quando comparado a ambientes em hardware dedicado.

Algumas lições que podem ser extraídas do trabalho apresentado são:

- A arquitetura utilizada permite a escalabilidade em software de um sistema que antes do advento das máquinas virtuais seria muito difícil de implantar.
- O mecanismo de redundância proposto, onde os dados ficam num banco de dados redundante e distribuído, permite tolerância a falhas e aumenta a robustez da ferramenta.
- A limitação de uma sala do sistema é dada pela limitação de uma VM, o que deverá ser melhorado no futuro, fazendo com que o sistema permita uma sala suportada por múltiplas VMs.

Um vídeo que permite visualizar o sistema pode ser visto no início do texto em: <http://www.inf.ufrgs.br/prav/mcu.htm>.

AGRADECIMENTOS E LICENÇA

Este trabalho está sendo patrocinado pela RNP (Rede Nacional de Ensino e Pesquisa), a NREN brasileira, e pela Mconf Tecnologia. A licença de uso do sistema está para ser definida em conjunto com os parceiros envolvidos, entretanto, a base do sistema é o software Kurento e o Gstreamer, ambos abertos.

REFERÊNCIAS

- Kridel, Tim: The Value of Video Collaboration. Disponível em <https://www.avixa.org/insight/whitepapers/Details/the-value-of-video-collaboration/>. Novembro, 2017. Acesso em Maio, 2018.
- Sorokin, R.; Rougier, J-L. Video conference in the fog: an economical approach based on enterprise desktop grid. *Annals of Telecommunications*. 2017. pp. 1-12.
- Akkus, I. E. Civanlar, M. R. Ozkasap, O. "Peer-to-Peer Multipoint Video Conferencing using Layered Video", *Image Processing 2006 IEEE International Conference on*, Oct. 2006. pp. 3053-3056.
- <http://www.polycom.com.br/products-services/hd-telepresence-video-conferencing/realpresence-room/realpresence-room-hdx-series.html>. Acesso em Junho, 2018.
- <https://www.lifesize.com/en/video-conferencing-equipment>. Acesso em Junho, 2018.
- <https://www.cisco.com/c/en/us/products/collaboration-endpoints/telepresence-mx-series/datasheet-listing.html>. Acesso em Junho, 2018.
- <https://www.terena.org/activities/tf-wbrtc/meeting1/slides/Jitsi.pdf>
- <https://www.ekiga.org/ekiga-softphone-features>
- <http://www.polycom.com/content/dam/polycom/common/documents/datasheets/realpresence-platform-virtual-editions-ds-enus.pdf>
- Alonso, A. et al. Deploying a Multipoint Control Unit in the Cloud: Opportunities and Challenges. In *Fourth International Conference on Cloud Computing, GRIDs, and Virtualization*. 2013 pp. 173-178.
- Roesler, V.; Longoni, G.; Marins, A. Multipresença: um sistema de videoconferência adaptável, escalável e interoperável. In: *TICAL 2015, 2015, Vina del Mar. Quinta Conferencia de Directores de Tecnologia de Información, TICAL 2015, 2015*.
- D. Kesavaraja and A. Shenbagavalli, "Cloud video as a Service [VaaS] with storage, streaming, security and Quality of service: Approaches and directions," *2013 International Conference on Circuits, Power and Computing Technologies (ICCPCT)*, Nagercoil, 2013, pp. 1093-1098.
- Liu W, et al. Cloud and traditional videoconferencing technology for telemedicine and distance learning. *Telemed J E Health* 2015. n 21. pp. 422-426.
- Kang, J. A., Han, M., Jang, J. and Kim, H. K. (2016), Adaptive Speech Streaming Based on Packet Loss Prediction Using Support Vector Machine for Software-Based Multipoint Control Unit over IP Networks. *ETRI Journal*, 38: 1064-1073.
- Scott, Rob: Video 2020 - The Future of Video Conferencing. Disponível em <https://www.uctoday.com/news/insights/video-2020-future-video-conferencing/>. Junho, 2017. Acesso em Maio, 2018.