

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ALEXANDRE KREISMANN DOS SANTOS

**Soluções *Open Source* para
Interoperabilidade entre Sistemas de
Videoconferência e Webconferência**

Trabalho de Conclusão apresentado como
requisito parcial para a obtenção do grau de
Bacharel em Ciência da Computação

Prof. Dr. Valter Roesler
Orientador

Porto Alegre, maio de 2016

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Kreismann dos Santos, Alexandre

Soluções *Open Source* para Interoperabilidade entre Sistemas de Videoconferência e Webconferência / Alexandre Kreismann dos Santos. – Porto Alegre: Graduação em Ciência da Computação da UFRGS, 2016.

33 f.: il.

Trabalho de Conclusão (bacharelado) – Universidade Federal do Rio Grande do Sul. Curso de Bacharelado em Ciência da Computação, Porto Alegre, BR-RS, 2016. Orientador: Valter Roesler.

1. Interoperabilidade. 2. Videoconferência. 3. Webconferência. 4. Open source. 5. Mconf. I. Roesler, Valter. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Graduação: Prof. Vladimir Pinheiro do Nascimento

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do Curso de Ciência da Computação: Prof. Carlos Arthur Lang Lisboa

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

To do

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	7
LISTA DE FIGURAS	8
LISTA DE TABELAS	9
RESUMO	10
ABSTRACT	11
1 INTRODUÇÃO	12
1.1 Interoperabilidade	14
1.2 Organização	14
2 PADRÕES E TECNOLOGIAS DE VIDEOCONFERÊNCIA	16
2.1 Protocolos de Sinalização	16
2.1.1 Padrão H.323	17
2.1.2 Padrão SIP	22
2.2 Protocolos de Transporte de Mídia	28
2.2.1 Padrão RTP	28
2.2.2 Padrão RTMP	28
2.3 Codificação de áudio	28
2.3.1 Padrão PCMA	28
2.3.2 Padrão PCMU	28
2.3.3 Padrão G.711	28
2.3.4 Padrão Speex	28
2.3.5 Padrão OPUS	28
2.4 Codificação de vídeo	28
2.4.1 Padrão H.263	29
2.4.2 Padrão H.264	29
2.4.3 Padrão VP8	29
2.5 Tecnologias para Webconferência	29
2.5.1 Flash	29
2.5.2 WebRTC	29
3 SOLUÇÕES OPEN SOURCE PARA INTEROPERABILIDADE	30
3.1 Freeswitch	30
3.2 Asterisk	30
3.3 Red5	30

3.4	FFMPEG	30
4	ARQUITETURA	31
4.1	Mconf	31
4.1.1	Arquitetura anterior	31
4.1.2	Arquitetura modificada	31
5	RESULTADOS	32
	REFERÊNCIAS	33

LISTA DE ABREVIATURAS E SIGLAS

ASCII	American Standard Code
IETF	Internet Engineering Task Force
ITU	International Telecommunication Union
MCU	Multipoint Control Unit
RFC	Request for Comments
RTP	Real-time Transport Protocol
RTMP	Real Time Messaging Protocol
SIP	Session Initiation Protocol
SDP	Session Description Protocol
URI	Uniform Resource Identifier
WebRTC	Web Real-Time Communication
XML	eXtensible Markup Language

LISTA DE FIGURAS

Figura 1.1:	Exemplo de sistema de videoconferência de sala: equipamentos da Polycom.	12
Figura 1.2:	Exemplo de sistema de telepresença da Cisco	13
Figura 1.3:	Exemplo de softfones: Ekiga (esquerda) e Skype (direita)	13
Figura 1.4:	Exemplo de sistema de webconferência: Mconf.	14
Figura 2.1:	Unidades H.323	17
Figura 2.2:	Pacote H.225	18
Figura 2.3:	Estabelecimento de uma chamada H.323 básica	19
Figura 2.4:	Negociação de mídia H.323	20
Figura 2.5:	H.323 <i>Fast Connect Mode</i>	22
Figura 2.6:	Diálogo SIP	23
Figura 2.7:	SIP Invite	24
Figura 2.8:	SIP 100 Trying	26
Figura 2.9:	SIP 200 OK	27
Figura 2.10:	SIP ACK	27
Figura 2.11:	SIP BYE	27
Figura 2.12:	SIP 200 OK	28

LISTA DE TABELAS

Tabela 2.1:	SIP Requests	25
Tabela 2.2:	SIP Responses	25

RESUMO

A comunicação transparente entre webconferência - videoconferência executada em um navegador web - e outros sistemas de videoconferência - sejam eles hardwares dedicados ou sistemas desktop - é uma tarefa dificultada pelo fato de que o primeiro, muitas vezes, implementa diferentes padrões para a sinalização, transmissão e codificação de áudio e vídeo. Outro fator complicador é que soluções para interoperação resolvem apenas parcialmente o problema, de forma que é preciso integrá-las em uma arquitetura para alcançar um nível maior de interoperabilidade.

O objetivo deste trabalho, portanto, é a análise de soluções *open source* e o desenvolvimento de uma arquitetura que permita a interoperabilidade entre webconferência e sistemas de videoconferência em geral (equipamentos e softwares).

Para tal, primeiramente, é feito o levantamento de protocolos, codecs, e demais padrões e tecnologias do contexto de video e webconferência. Após, apresentação de soluções *open source* que possibilitem a comunicação transparente dos padrões estudados. Finalmente, projeto e desenvolvimento de uma arquitetura utilizando essas soluções. Será utilizado o sistema de webconferência open source *Mconf* para efetuar a implementação e validação. A validação deverá permitir a interoperabilidade da webconferência com softwares e equipamentos de videoconferência, como um equipamento comercial da Polycom, por exemplo.

Palavras-chave: Interoperabilidade, videoconferência, webconferência, open source, Mconf.

Open Source Solutions for Interoperability of Video Conference and Web Conference Systems

ABSTRACT

Transparent communication among web conference - video conference executed in a web browser - and other video conference systems - dedicated hardwares or desktop applications - is a task made difficult by the fact that the first one often implements different standards for media signaling, transmission and encoding. Another complicating matter is that interoperability solutions only partially resolve the problem, so we need to integrate them into an architecture to achieve a higher level of interoperability.

Thus the objective of this work is to analyze open source solutions and to develop an architecture allowing interoperability of web conference and different video conference systems (equipments and softphones).

To do so, firstly we study protocols, codecs and further standards and technologies of video and web conferencing context. Afterwards, we present open source solutions that allow transparent communication of studied standards. Finally, we develop an architecture utilizing those solutions. Implementation and validation will be performed by using the open source web conference system *Mconf*. Through validation, the web conference must interoperate with softphones and commercial video conference equipments (for instance, a Polycom equipment).

Keywords: interoperability, video conference, web conference, open source, Mconf.

1 INTRODUÇÃO

Sistemas de videoconferência possibilitam a comunicação de múltiplos usuários, independente de suas posições geográficas, através de áudio e vídeo. Com isso, aulas, treinamentos, apresentações e outros tipos de reunião podem ser realizados sem o deslocamento pessoal para um mesmo local físico, reduzindo custos e agilizando logísticas. Segundo (ROESLER et al., 2012), sistemas de videoconferência podem ser organizados em quatro grupos:

1. **Sistemas de videoconferência de sala:** São sistemas baseados em hardware e exigem o uso de salas físicas com um terminal de acesso - no caso, equipamentos de videoconferência acompanhados de televisões, monitores ou sistemas de som - como pode ser visto na Figura 1.1. Por serem *hardware-based*, têm sua flexibilidade prejudicada, podendo depender de extensões Desktop para funcionalidades extras, tais como compartilhamento de tela ou apresentação de slides. Entretanto, justamente por serem hardwares dedicados, equipamentos de videoconferência de sala garantem áudio de alta qualidade e vídeo de alta definição em tempo real.

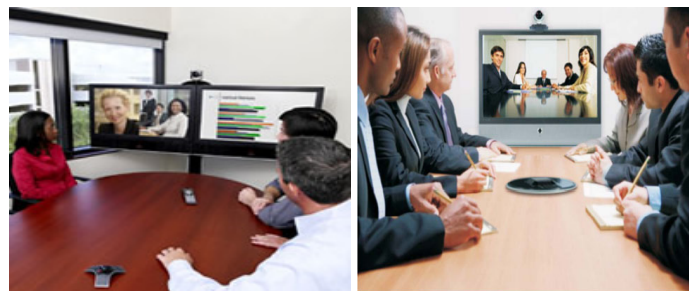


Figura 1.1: Exemplo de sistema de videoconferência de sala: equipamentos da Polycom.

2. **Sistemas de videoconferência por telepresença:** são uma especialização dos sistemas de sala - herdando, portanto, as características descritas no item 1 - e têm como objetivo reproduzir a sensação de que todos os participantes estão na mesma sala física. Para tal, certas configurações de ambiente da sala e equipamentos devem ser feitas, como:
 - Ajustar a câmera para mostrar o participante remoto em tamanho real;
 - Usar microfones e alto-falantes de uma forma que o som remoto venha a partir da posição do participante;
 - Usar vídeo em alta definição para mostrar os detalhes dos participantes;

- Usar um ambiente complementar, com o mesmo tipo de cadeiras, cores e mesas.



Figura 1.2: Exemplo de sistema de telepresença da Cisco

3. **Sistemas de videoconferência desktop:** também conhecidos como softfones (junção das palavras *software* e *telefone*), são alternativas *software-based* para sistemas de sala. Devem ser instalados em um computador pessoal, equipado com câmera e microfone. Embora tais sistemas não usem hardwares dedicados, a diferença de qualidade de mídia entre sistemas de sala e desktop tem diminuído, devido aos avanços na qualidade dos hardwares de PCs.

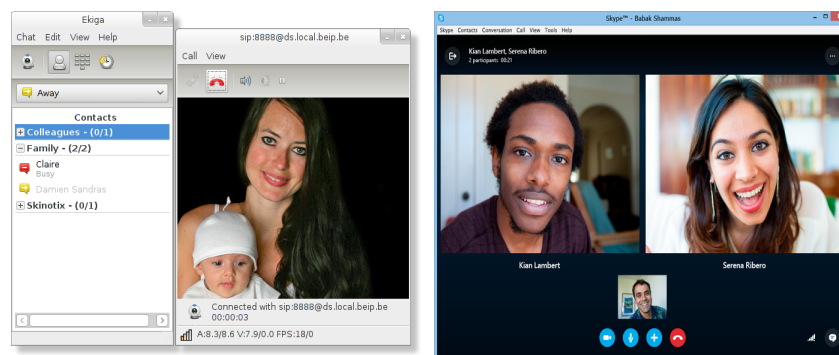


Figura 1.3: Exemplo de softfones: Ekiga (esquerda) e Skype (direita)

4. **Sistemas de videoconferência web ou sistemas de webconferência:** são sistemas de videoconferência executados em um navegador web, sendo compatíveis, portanto, com diferentes sistemas operacionais e não requerendo instalação. Para acessar um sistema de webconferência, o usuário, utilizando o sistema operacional de sua preferência, apenas precisa abrir uma URL em um bom browser, o que implica em uma grande facilidade de uso. Exemplos de tais sistemas são o Adobe Connect, Google Hangout e Mconf.



Figura 1.4: Exemplo de sistema de webconferência: Mconf.

1.1 Interoperabilidade

Por meio de padrões de videoconferência, um sistema desse tipo deve ser capaz de interoperar, através de áudio e vídeo, com diferentes sistemas (equipamentos ou softwares). Por exemplo, se um sistema utiliza os padrões SIP para sinalização e RTP para o transporte de mídia (itens que veremos mais adiante), espera-se que ele possa se comunicar com outro sistema que também os empregue.

Embora sistemas de videoconferência de sala e softfones trabalhem com um escopo similar de padrões (basicamente: SIP ou H.323, e RTP), sistemas de webconferência apresentam algumas particularidades. Por exemplo, se um sistema foi desenvolvido sobre a plataforma Flash, o transporte de mídia utilizará o padrão RTMP, em vez de RTP. Já webconferências utilizando-se da tecnologia WebRTC, que possibilita browsers utilizarem SIP e RTP, apresentam VP8, OPUS e outros padrões de codificação de mídia ainda não encontrado em muitos equipamentos de videoconferência e softfones.

Quando não há os mesmos padrões implementados, a interoperação fica dependente de soluções intermediárias que possibilitem a ponte entre esses padrões. Exemplificando, caso um equipamento de videoconferência utilize H.323 para sinalização e RTP para o transporte de mídia, e um sistema de webconferência empregue, respectivamente, SIP e RTMP, precisaremos de duas pontes para haver interoperabilidade: uma que viabilize a comunicação H.323/SIP e a outra, RTP/RTMP.

Este trabalho, portanto, apresentará os principais padrões e tecnologias de videoconferência, e, em seguida, soluções *open source* que possibilitem pontes entre eles, a fim da comunicação transparente de webconferência e videoconferência (softfones e equipamentos). Como soluções para interoperação normalmente implementam apenas algumas pontes, precisamos integrá-las em uma arquitetura para cobrir um número maior de padrões e, assim, alcançar um nível maior de interoperabilidade.

1.2 Organização

Para tanto, o trabalho está estruturado da seguinte forma:

- capítulo 2: Apresentação de padrões de videoconferência e tecnologias: protocolos de sinalização (H.323, SIP), protocolos de transporte de mídia (RTP, RTMP),

codecs de áudio (PCMA, PCMU, G.722, Speex, OPUS), codecs de vídeo (H.263, H.264, VP8) e tecnologias para webconferência (Flash, WebRTC);

- capítulo 3: Soluções de interoperabilidade *open source*: Freeswitch, Asterisk, Red5 e FFMPEG;
- capítulo 4: Desenvolvimento da arquitetura, integrando soluções de interoperabilidade, utilizando o sistema de webconferência Mconf;
- capítulo 5: Resultados da implementação da arquitetura, demonstrando a interoperabilidade do sistema Mconf com diferentes sistemas de videoconferência.

2 PADRÕES E TECNOLOGIAS DE VIDEOCONFERÊNCIA

Segundo (FIRESTONE; RAMALINGAM; FRY, 2007), as principais camadas de um sistema de conferência são:

- Interface de usuário e de administrador;
- Controle de Conferência: controle de criação/finalização de sessões, de roteamento de ligações e de alocações de recursos em geral;
- Plano de Controle: controle de chegada/saída de conexões e da negociação de parâmetros da sessão de acordo com a capacidade do servidor de mídia;
- Plano de Mídia: controle das *streams* de mídia, administrando o transporte e a mixagem de áudio e vídeo.

O principal desafio para interoperabilidade entre sistemas de videoconferência é considerar os diferentes Planos de Controle e de Mídia existentes. São essas camadas que definem de que maneira um sistema envia e recebe requisições para entrada e saída de usuários, bem como o modo no qual ele irá transmitir e receber áudio e vídeo. Um sistema de conferência com Plano de Controle e de Mídia flexíveis permitem que usuários utilizando diferentes equipamentos ou softwares de videoconferência - os chamados *end-points* - se conectem e se comuniquem de maneira transparente em uma sessão desse sistema.

Como afirmado em (KUROSE; ROSS, 2013), com a popularização das aplicações de tempo real, importantes órgãos, como o IETF e ITU, estabeleceram (e continuam estabelecendo) padrões para essa classe de aplicações. No contexto de videoconferência, padrões amplamente usados no Plano de Controle são os protocolos de sinalização SIP e H.323, os quais veremos na Seção 2.1. Já no Plano de Mídia, dependemos tanto de padrões de transporte como de codificação de mídia (Seções 2.2, 2.3 e 2.4). Já em termos de webconferência, a tecnologia que compõe o sistema determinará o escopo de padrões usados, como será detalhado na Seção 2.5.

A seguir, portanto, são apresentados os principais padrões e tecnologias para sistemas de conferência.

2.1 Protocolos de Sinalização

Os protocolos de sinalização têm como papel a implementação do Plano de Controle. Para isso:

1. Determinam-se as portas onde o sistema irá aguardar conexões de *endpoints*. Tais portas definem o **canal de sinalização** de um sistema de conferência;
2. No momento em que um *endpoint* se conecta no canal de sinalização, o Plano de Controle envia informações da capacidade de mídia do sistema, as quais o *endpoint* deve respeitar. Esse passo estabelece a **negociação de mídia**;
3. Após essa negociação (bem-sucedida), o Plano de Controle e o *endpoint* abrem um canal lógico para *streaming* de mídia, e o *endpoint* passa a se comunicar, então, com o Plano de Mídia, para enviar e/ou receber áudio/vídeo. O Plano de Controle será responsável por administrar esse canal lógico, avisando o Plano de Mídia se o endpoint sair da sessão, ou algum erro de conexão ocorrer.

Como dito na introdução desse capítulo, existem dois principais padrões para a sinalização. O primeiro, H.323, foi estabelecido pelo ITU em 1996. O segundo, SIP, foi projetado pelo IETF e primeiramente definido no RFC 2543 em 1999.

2.1.1 Padrão H.323

Como afirmado em (TANENBAUM; WETHERALL, 2011), H.323, mais do que um protocolo específico, é uma visão geral da arquitetura de telefonia da Internet, bem como um encapsulador de diversos outros protocolos (que veremos no decorrer desse trabalho). O padrão, primeiramente, define quatro tipos de unidades:

- **Terminais:** são os *endpoints*, emissores e receptores de mídia. É a única unidade obrigatória;
- **Gatekeeper:** define a *zona* - conjunto de terminais que o gatekeeper irá administrar -, controlando quais *endpoints* podem se comunicar entre si, provendo acesso e controlando a largura de banda utilizada pelos terminais;
- **Gateway:** permite a interoperação com terminais que utilizam redes de telefonia ou outros tipos de rede (que não a internet). Na prática, ainda permite a interoperabilidade com terminais que utilizem SIP, em vez de H.323, para sinalização;
- **MCU (Multipoint Control Unit):** sistema que permite que múltiplos terminais se conectem em uma sessão de conferência, distribuindo e mixando a mídia para os participantes.

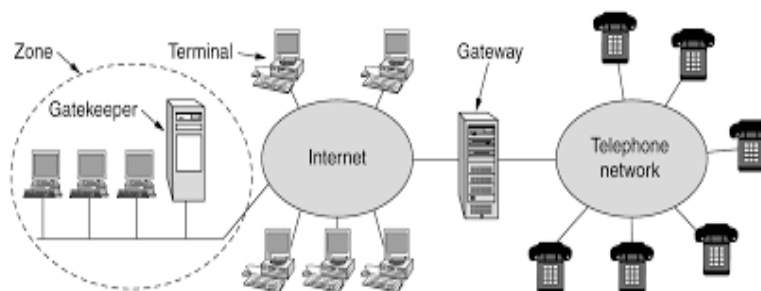


Figura 2.1: Unidades H.323

2.1.1.1 H.225

O protocolo H.225 define o canal de sinalização de uma sessão H.323, descrevendo, portanto, como uma chamada é iniciada, estabelecida e finalizada. Um terminal iniciando uma chamada (daqui para frente chamado de *caller*), deve, primeiramente, abrir uma conexão TCP com a unidade que se quer chamar (respectivamente, *callee*), que estará esperando conexões na porta 1720. Um pacote H.225 é definido da seguinte forma:

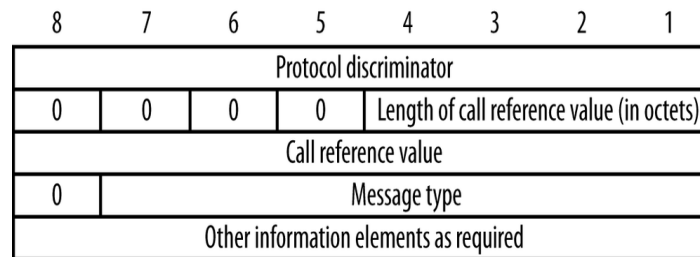


Figura 2.2: Pacote H.225

- *Protocol Discriminator*: Número que identifica a mensagem como uma mensagem H.225. O valor desse campo é 8.
- *Length of Call Reference Value*: comprimento do CRV, que pode ser 1 ou 2 bytes;
- *Call Reference Value*: CRV é o número identificador único da chamada;
- *Message Type*: tipo da mensagem que o pacote H.225 carrega (*Setup Message*, *Connect*, *Alerting*, etc - definidos mais adiante).
- *Information Elements*: espaço para parâmetros adicionais. Dependem do *Message Type*.

A figura abaixo demonstra o estabelecimento de uma chamada H.323 básica:

1. Após o estabelecimento da conexão TCP, o *caller* inicia a chamada, mandando um H.225 *Setup Message*. Esse tipo de *Message Type* carrega como parâmetro obrigatório o *Bearer Capability*, que define a natureza da chamada: se é apenas de áudio ou uma chamada audiovisual.
2. O *callee* responde o *Setup Message*, com um H.225 *Alerting*, indicando ao *caller* que o dispositivo chamado já iniciou o procedimento de alerta de que uma nova chamada chegou - em outras palavras, que o dispositivo está "tocando" (como um telefone; em inglês, *Ring*). Antes do *Alerting*, outro tipo de *Message Type* pode ocorrer (não é obrigatório) do *callee* para o *caller*: o *Call Proceeding*. CP tem o objetivo de informar o *caller* que a chamada está em processo de ser estabelecida e que não será mais aceito novas mensagens para estabelecimento de chamada.
3. O *callee* também responde um *Setup ACK*, indicando que o *Setup Message* foi devidamente recebido.
4. Quando a chamada é finalmente atendida, é mandado um H.225 *Connect* para o *caller*. É nesse momento que a conexão H.225 foi completamente estabelecida.

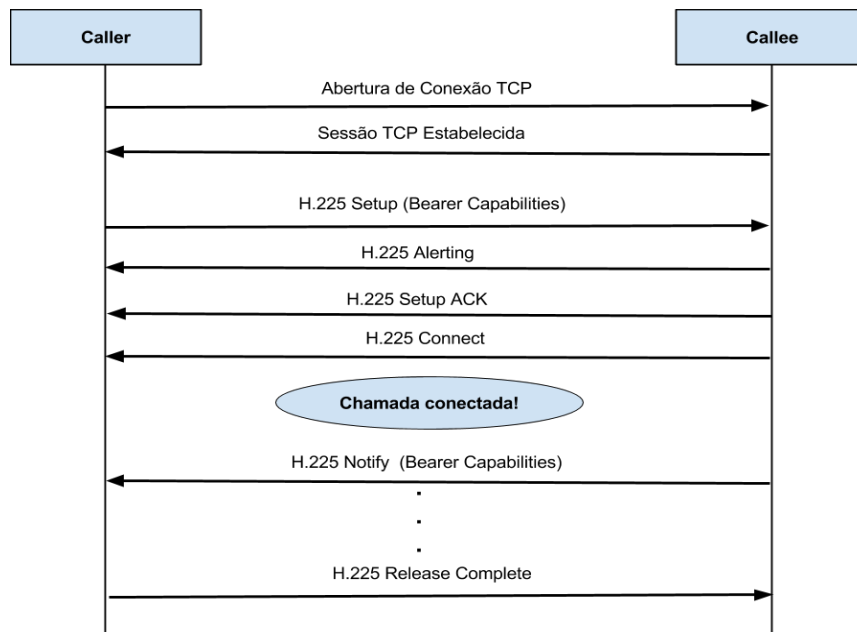


Figura 2.3: Estabelecimento de uma chamada H.323 básica

5. Mensagens H.225 *Notify* servem para troca de informações gerais entre *caller* e *callee* durante a chamada. Um uso é para o *callee* indicar ao *caller* o seu próprio *Bearer Capability*, como indicado na Figura 2.3.
6. Quando uma das partes envolvidas encerra a chamada, é enviado um H.225 *Release Complete*. Com isso, o CRV, identificador único da chamada, é desalocado - assim como quaisquer outros recursos alocados - e pode ser reusado em outra chamada. O H.225 *Release Complete* carrega consigo um código que determina qual foi o motivo do encerramento da chamada (a Recomendação ITU-T H.225.0 (ITU, 2009a) contém uma lista de todos os códigos de encerramento e seus significados).

2.1.1.2 H.245

O protocolo H.245 provê o mecanismo para a negociação de mídia e para a abertura dos canais RTP, onde os dados multimídia serão transmitidos/recebidos. Após o H.225 *Setup Message*, o *callee* pode usar o H.225 *Call Proceeding*, *Alerting* ou *Connect* para enviar ao *called* o IP e a porta para a abertura da sessão H.245. Quando o *caller* recebe essas informações, é aberta uma conexão TCP entre as partes envolvidas, onde haverá, então, a negociação de mídia.

Na figura abaixo, é demonstrado como é feita a negociação de mídia via sessão H.245:

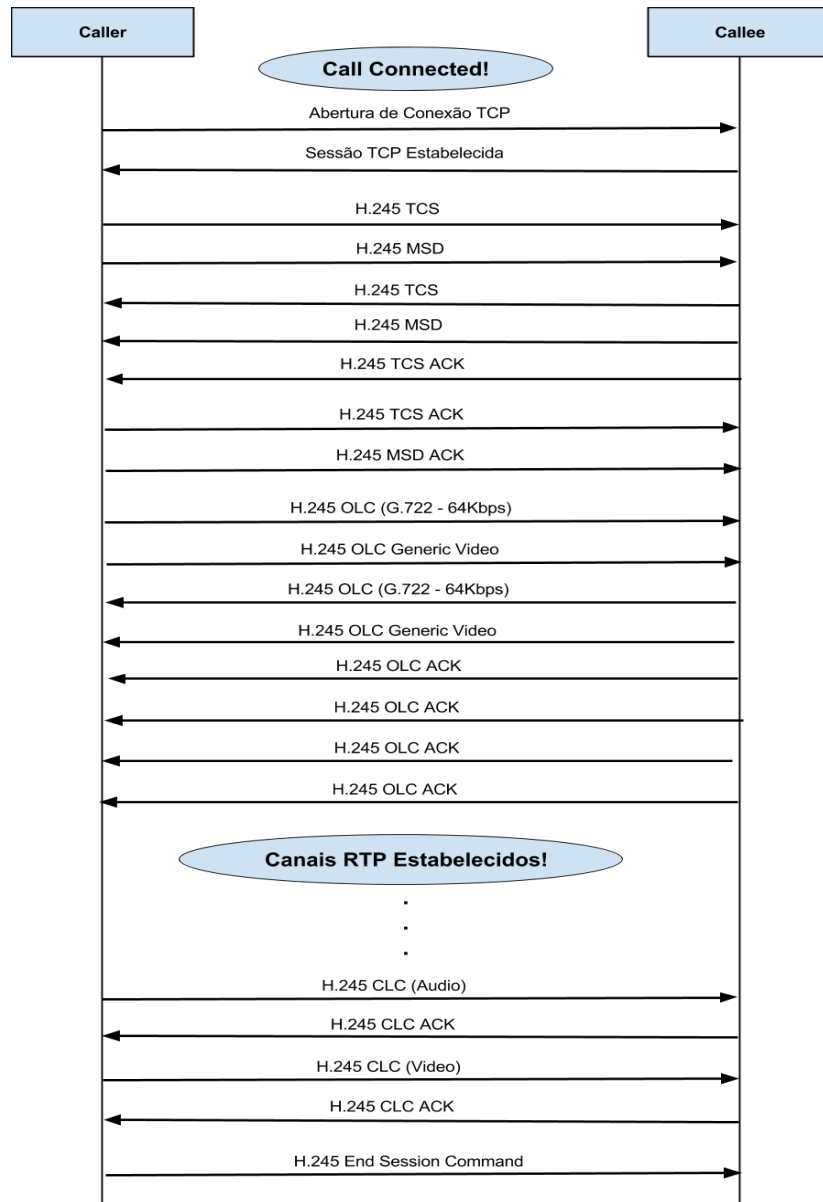


Figura 2.4: Negociação de mídia H.323

1. Após o estabelecimento da chamada - como visto anteriormente na Seção 2.1.1.1 - e da conexão TCP para a Sessão H.245, *caller* e *callee* trocam, primeiramente, 2 tipos de mensagens H.245: *TCS* (*Terminal Capability Set*) e *MSD* (*Master-Slave Determination*).
2. *TCS* contém a descrição da capacidade de mídia do remetente da mensagem. Inclui a lista de todos os codecs de áudio e vídeo suportados e suas características (tipos de payload, tempos de packetização, perfis e resoluções de vídeo, taxa de bits e de frames, etc - itens que veremos em 2.3 e 2.4). Também contém o *Simultaneous Capability Set*, que define quais grupos de codecs podem ser usados simultaneamente. A parte que recebe o H.245 *TCS* informa ao transmissor da mensagem o recebimento da mesma com um *TCS ACK*.
3. A troca de mensagens *MSD* define quem irá ser o 'mestre' e o 'escravo' da ses-

são. O mestre será responsável por atribuir um ID para o canal lógico, gerar uma chave quando há encriptação de mídia, entre outras tarefas de administração do canal lógico. Uma mensagem *MSD* contém um *Terminal Type* (um número) e um aleatoriamente escolhido *Determination Number*. Cada parte envolvida compara os números recebidos na mensagem *MSD* com os seus próprios números e quem tiver o maior *Terminal Type* será o mestre. Em caso de empate, o mestre será quem tiver o maior *Determination Number*. Caso haja novamente um empate, ocorre uma nova negociação, com novas mensagens *MSD*.

4. Posteriormente à determinação do mestre e escravo, o *caller* envia o H.245 *Open-LogicalChannel (OLC)*, requisitando a abertura do canal lógico. O canal lógico, no contexto H.323, representa o caminho usado para a transmissão de mídia. A mensagem *OLC* carrega consigo o codec de áudio e de vídeo que o *caller* usará na transmissão, bem como as características de cada codec (como explicado no item 2). Outros dados importantes do H.245 *OLC* são as portas RTCP (mais detalhes na Seção 2.2.1) que serão usadas no contexto RTP e o *LCN (Logical Channel Number)*, identificador único do canal lógico. No exemplo da Figura 2.4, tanto o *caller* quanto o *callee* enviam um *OLC* requisitando um canal para transmitir áudio usando o codec G.722, e um outro *OLC* requisitando um canal para transmitir vídeo H.264. *Generic Video Capability*, transportada na mensagem *OLC*, é uma estrutura H.245 específica para informar características do codec H.264. Existem, ainda, outras mensagens H.245 específicas para controle de vídeo, como o *Flow Control*, usado para reajustar a taxa de bits máxima, e o *Miscellaneous Command*, utilizado para pausar, congelar, avançar a stream de vídeo, bem como fazer outros ajustes, como, por exemplo, alterar a resolução ou a taxa de frames.
5. Quem recebe o *OLC* deve examinar os detalhes do pedido de abertura do canal lógico, alocar os recursos necessários e responder com um H.245 *OLC ACK*, contendo os IPs e as portas RTP e RTCP que serão usados para receber a mídia.
6. Quando uma das partes envolvidas encerra a chamada, é enviado um H.245 *Close Logical Channel (CLC)*, que deverá ser respondido com um *CLC ACK*.
7. H.245 *End Session* é a mensagem que indica o fim da sessão. Após essa mensagem, não há mais trocas de nenhuma outra mensagem H.245 (portanto, não há ACK para *End Session*).

2.1.1.3 H.323 Fast Connect Mode

No H.323 versão 2 existe a opção do *Fast Connect Mode*. O *caller* manda um H.245 *OLC* com um codec sugerido dentro do H.225 *Setup Message*. Se o *callee* suportar *Fast Connect Mode* e aceitar a sugestão do codec, será respondido um H.225 *Connect* também transportando um H.245 *OLC*. Com isso, as partes envolvidas já têm as informações necessárias para estabelecer a conexão de mídia, diminuindo significativamente o número de mensagens requeridas para estabelecer a sessão H.323. Segundo (FIRESTONE; RAMALINGAM; FRY, 2007), equipamentos de vídeo H.323 geralmente não suportam *Fast Connect Mode*.

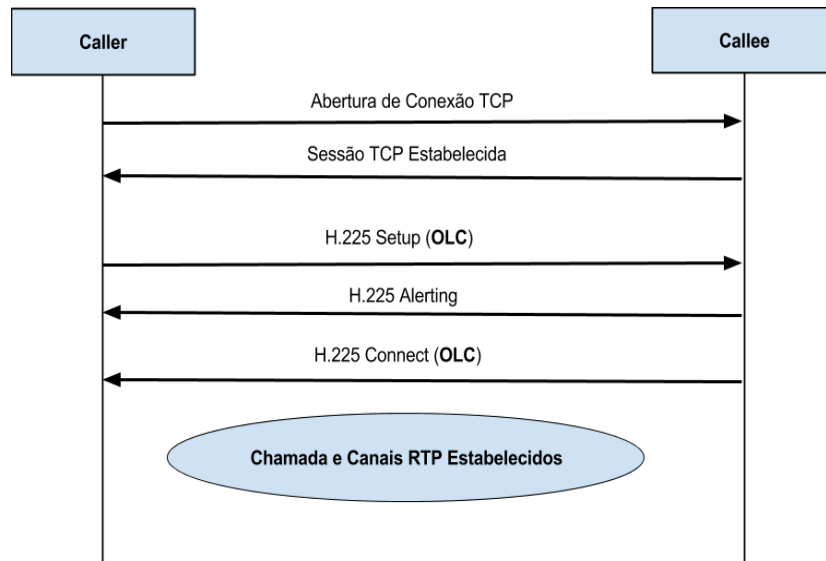


Figura 2.5: H.323 *Fast Connect Mode*

2.1.1.4 Mais informações sobre H.323

Para mais informações sobre o protocolo H.323, consulte (ITU, 2009a), (ITU, 2009b), (ITU, 2011) e também (FIRESTONE; RAMALINGAM; FRY, 2007) (detalhes nas Referências deste trabalho).

2.1.2 Padrão SIP

Segundo (TANENBAUM; WETHERALL, 2011), muitos viam o H.323 como um típico produto de uma companhia de telecomunicação: grande, complexo e inflexível. Nessa situação, o IETF projeta o padrão SIP, que, em vez de uma suíte de protocolos, é um módulo único. O protocolo SIP é *text-based*, apresentando todas as suas mensagens, atributos e métodos em ASCII. O padrão, primeiramente, define os seguintes elementos:

- **User Agent:** podem ser tanto os *endpoints*, emissores e receptores de mídia, quanto servidores de controle de chama (abordados nos próximos itens). Todo UA é formado por um *User Agent Client (UAC)* - responsável por iniciar as transações (que também definiremos mais adiante)- e por um *User Agent Server (UAS)* - responsável pelas respostas dessas transações;
- **Proxy Server:** Entidade que faz o roteamento de mensagens SIP. É responsável por determinar para onde mensagens SIP vão ser direcionadas (quais *endpoints* ou servidores receberão determinada mensagem), provendo também autenticação e autorização de mensagens. Um PS *stateful* armazena o histórico de todas as mensagens de uma transação, a fim de levantar dados para distribuição de sessões SIP em diferentes servidores e/ou para geração de estatísticas. Já um PS *stateless* faz o roteamento de mensagens sem guardar quaisquer informações.
- **Redirect Server:** Entidade que informa UAs para onde uma mensagem SIP deve ser encaminhada. Um UA manda uma mensagem SIP, que deve ser entregue a outro UA, para o *Redirect Server*. Este, então, retorna para o UA de origem o endereço alternativo do UA destino. Em posse da nova informação, o UA remetente remanda a mensagem SIP diretamente para o UA destino. Portanto, não é o *Redirect Server*

que faz o roteamento da mensagem (como o *Proxy Server* faria): o trabalho de fazer a nova ligação - utilizando o endereço certo advindo da resposta do RS - é do UA de origem.

- **Registrar Server:** Entidade para onde os usuários enviam o pedido de registro. Usuários são registrados como URIs em um banco de dados, onde cada URI é associado com uma localização (normalmente o endereço IP desse usuário). Com isso, *Proxy Servers* ou *Redirect Servers* podem usar essa base de dados para executar suas tarefas.

Na figura abaixo, é demonstrado um diálogo SIP:

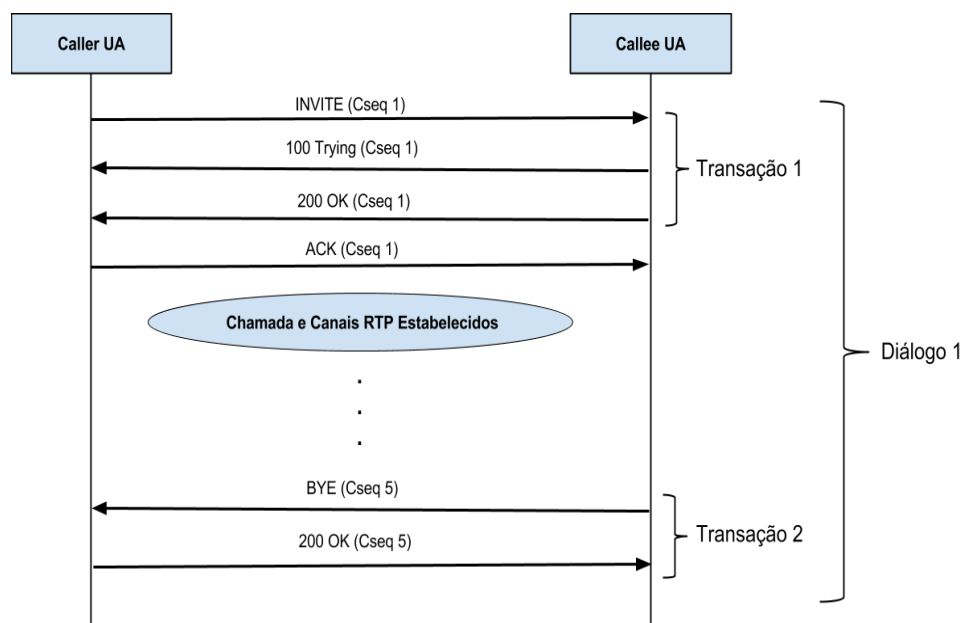


Figura 2.6: Diálogo SIP

1. Começamos com o UA callee recebendo um *INVITE* do UA Caller, para o início de uma chamada. Tipicamente, a porta que define o canal de sinalização SIP é a porta 5060 (podemos encontrar ambientes onde se usa a porta 5070 ou 5061 - esta última para sinalizações encriptadas). Um *INVITE* tem a seguinte estrutura:

```

INVITE sip:UAallee@172.27.14.53 SIP/2.0

! As linhas abaixo são os headers SIP:
Via: SIP/2.0/UDP 172.27.14.4:5060;branch=8dJXAX9MDw
Max-Forwards: 70
To: <sip:UAallee@172.27.14.53>
From: <sip:UAcallee@172.27.14.4>;tag=ds17aa9bd4
Call-ID: 11022705439144@172.27.14.4
CSeq: 1 INVITE
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE
Content-Length: 251
Content-Type: application/sdp

! Já as linhas abaixo estão no body do INVITE SIP:
v=0
o=UAcallee 1549546120 0 IN IP4 172.27.14.4
s=ExampleSession
c=IN IP4 172.27.14.4
t=0 0
m=audio 49220 RTP/AVP 118
a=rtpmap:118 Speex/16000/1
m=video 5068 RTP/AVP 104
a=rtpmap:104 H264/90000
a=fmtp:104 max-fs=6336;max-mbps=190080;profile-level-id=42801e

```

Figura 2.7: SIP Invite

O INVITE é acompanhado do endereço de destino (no caso, o usuário 'UAallee' que está no endereço IP 172.27.14.53) e tem os seguintes cabeçalhos:

- (a) *Via*: indica como transportar a mensagem (no caso, usando UDP) e para onde as respostas dessa mensagem devem ir (IP e porta de destino). Ainda contém o *branch*, identificador único da transação SIP. Uma transação, no contexto SIP, significa um *Request* - requisição - seguido de suas *Responses* - respostas (pode ser visto na Figura 2.6 que o ACK não está incluído na Transação 1, isso pois, em geral, ACK não é considerado parte de transações). Já um diálogo SIP é o conjunto de todas as transações entre 2 UAs. Nas Tabelas 2.1 e 2.2 são definidos os SIP *Requests* e os SIP *Responses* mais comuns segundo (FIRESTONE; RAMALINGAM; FRY, 2007);
- (b) *Max-Forwards*: usado para evitar *loops*. Toda vez que um *Request* é redirecionado, o *proxy* que recebe esse *Request* decrementa o *Max-Forwards* em 1. Se um *proxy* recebe um *Request* com *Max-Forwards* igual a 0, ele responde com uma mensagem de erro para o UA de origem;
- (c) *To*: identificador do destino: usuário seguido do seu endereço IP;
- (d) *From*: identificador da origem: usuário seguido do seu endereço IP. O parâmetro *tag* é usado para identificar o diálogo SIP;
- (e) *Call-ID*: identificador único da chamada SIP;
- (f) *Cseq*: o *Command Sequence* é um identificador de transação, sendo um número (que pode ser arbitrário) acompanhado do nome da *Request* que originou a transação. Todas as *Requests* e *Responses* de uma transação devem usar o mesmo *Cseq*;
- (g) *Allow*: lista de *Requests* que o UA pode enviar e receber;
- (h) *Content-Length*: número de caracteres do *body*;
- (i) *Content-Type*: do que consiste o *body*; no caso, do SDP.

SIP Request	Descrição
INVITE	Convida um UA para uma chamada.
BYE	Termina o diálogo SIP entre 2 UAs.
OPTIONS	Solicita informações das capacidades do UA remoto.
MESSAGE	Manda mensagens instantâneas (não faz parte de um diálogo).
ACK	Confirma que o UA <i>caller</i> recebeu uma <i>Response</i> definitiva (que não é da classe <i>1xx</i>) sobre o seu pedido de INVITE.
REGISTER	Solicitação para ser registrado em um servidor de registro.
CANCEL	Cancela um <i>Request</i> corrente.
INFO	Método para UAs trocarem informações gerais durante uma sessão SIP.
PRACK	Age como um ACK temporário, onde um UA confirma que recebeu alguma <i>Response</i> provisória (da classe <i>1xx</i>)
UPDATE	Atualiza o estado da sessão SIP.
SUBSCRIBE	Solicita inscrição para receber notificações da ocorrência de determinado(s) evento(s) SIP.
NOTIFY	Envia um evento de notificação para UAs inscritos.
REFER	Indica que um UA deve usar o endereço contido no REFER para refazer um determinado <i>Request</i> .

Tabela 2.1: SIP Requests

SIP Response Code	Reason	Response Class
100	Trying	1xx
180	Ringing	1xx
200	OK	2xx
301	Moved permanently	3xx
302	Moved temporarily	3xx
400	Bad Request	4xx
600	Busy	6xx
603	Decline	6xx
604	Does not exist	6xx

Tabela 2.2: SIP Responses

O SDP (*Session Description Protocol*) define a sintaxe da descrição da sessão SIP de mídia. É através da troca de SDPs entre origem e destino que é feita a negociação de mídia. No exemplo da Figura 2.7, o SDP contém os seguintes parâmetros:

- (a) *v*: versão do SDP;
- (b) *o*: identificador do criador da sessão;
- (c) *s*: nome da sessão;
- (d) *c*: informação da conexão. Basicamente, de qual IP virá a mídia. Esse é um exemplo de atributo que pode ser tanto:
 - *session-level*: vale para toda a sessão. Para isso, o atributo deve estar listado antes da descrição das mídias. No exemplo da Figura 2.7, o parâmetro "*c=*" está justamente como atributo de sessão, ou seja, todas as mídias que serão descritas posteriormente virão do mesmo IP;

- *media-level*: vale apenas para uma mídia específica. Para isso, o atributo deve estar listado depois de uma descrição de mídia ("*m*="). Com isso, por exemplo, poderíamos declarar diferentes informações da conexão para diferentes mídias (assim, áudio e vídeo poderiam vir de IPs distintos).
- (e) *t*: define o tempo de início e o tempo de fim da sessão SIP. Os valores de *t* são representações decimais dos valores de tempo do *Network Time Protocol* (NTP), que, por sua vez, são os valores de tempo em segundos desde 1900. No exemplo da Figura 2.7, ambos os valores são zero, significando que o começo da sessão é imediato (assim que negociação de mídia terminar) e sem tempo de término (sessão só será encerrada quando uma das partes envolvidas desligar a chamada);
- (f) *m*: descreve as mídias que serão usadas. No nosso exemplo, tanto áudio, quanto vídeo. A linha "*m*=" ainda define a porta de destino da mídia (49220 para áudio e 5068 para vídeo), o protocolo de transporte usado na transmissão da mídia (RTP sob AVP para ambos) e a lista de *payload types* (mais detalhes nas Seções 2.3 e 2.4) de cada codec de mídia;
- (g) *a*: descrição de um atributo. Atributos podem ser *session-level* (atributos de sessão) ou *media-level* (atributos de mídia). Um uso importante desse parâmetro é descrever os codecs de mídia que um UA suporta, logo após a descrição de mídia ("*m*="). Na Figura 2.7, é estabelecido que o UA Caller só suporta Speex a 16kHz como codec de áudio e H.264 a 90kHz como codec de vídeo. Ao final do SDP, há ainda alguns atributos específicos do codec H.264, como tamanho máximo de frame, taxa máxima de bits por segundo e *profile* de vídeo (ver Seção 2.4.2). Vale a menção que há diversos outros usos para atributos e existe uma lista extensa definida em (IETF, 2002).
2. Quando o *callee* recebe a chamada e entra em processo para determinar se a aceitará ou não, é mandado um 100 Trying para o *caller*:

```
SIP/2.0 100 Trying

Via: SIP/2.0/UDP 172.27.14.53:5060;branch=8dJXAX9MDw
To: <sip:UAcallee@172.27.14.4>
From: <sip:UAcallee@172.27.14.53>;tag=ds17aa9bd4
Call-ID: 11022705439144@172.27.14.4
CSeq: 1 INVITE
Content-Length: 0
```

Figura 2.8: SIP 100 Trying

3. Assim que o *callee* aceita a chamada, este manda um 200 OK para o *caller*. Nele, há o SDP descrevendo quais codecs de mídia, do conjunto que o *caller* enviou no SDP do INVITE, que o *callee* também suporta (no caso, o *callee* também suporta Speex e H.264). Há também, como importante atributo de mídia (tanto para áudio, quanto para vídeo), o IP e as portas RTCP que o *callee* usará para o controle do recebimento dos fluxos RTP (áudio e vídeo).

```

SIP/2.0 200 OK

Via: SIP/2.0/UDP 172.27.14.53:5060;branch=8dJXAX9MDw
To: <sip:UAcallee@172.27.14.4>
From: <sip:UAcallee@172.27.14.53>;tag=ds17aa9bd4
Call-ID: 11022705439144@172.27.14.4
CSeq: 1 INVITE
Content-Length: 317
Content-Type: application/sdp

v=0
o=UACaller 1549568353 0 IN IP4 172.27.14.53
s=ExampleSession
c=IN IP4 172.27.14.53
t=0 0
m=audio 31818 RTP/AVP 118
a=rtpmap:118 Speex/16000
a=rtcp:31819 IN IP4 172.27.14.53
m=video 29240 RTP/AVP 104
a=rtpmap:104 H264/90000
a=fmtp:104 max-fs=6336;max-mbps=190080;profile-level-id=42801e
a=rtcp:29241 IN IP4 172.27.14.53

```

Figura 2.9: SIP 200 OK

4. Quando o *caller* finalmente recebe o 200 OK, ele transmite o ACK. A partir daí, ambos os UAs estão numa chamada estabelecida e prontos para enviarem e receberem áudio e vídeo.

```

ACK sip:UAcallee@172.27.14.53 SIP/2.0

Via: SIP/2.0/UDP 172.27.14.4:5060;branch=8dJXAX9MDw
Max-Forwards: 70
To: <sip:UAcallee@172.27.14.53>
From: <sip:UAcallee@172.27.14.4>;tag=ds17aa9bd4
Call-ID: 11022705439144@172.27.14.4
CSeq: 1 ACK
Content-Length: 0

```

Figura 2.10: SIP ACK

5. Quando uma das partes desliga a chamada, é enviado um BYE. No caso da Figura 2.7, quem o faz é o *callee*:

```

BYE sip:UAcallee@172.27.14.4 SIP/2.0

Via: SIP/2.0/UDP 172.27.14.53:5060;branch=8dJXAX9MDv
To: <sip:UAcallee@172.27.14.4>
From: <sip:UAcallee@172.27.14.53>;tag=ds17aa9bd4
Call-ID: 11022705439144@172.27.14.4
CSeq: 5 BYE
Content-Length: 0

```

Figura 2.11: SIP BYE

6. No momento em que o *caller* recebe o BYE, é enviado o 200 OK para o *callee* e, desse modo, a chamada é encerrada.

```
SIP/2.0 200 OK

Via: SIP/2.0/UDP 172.27.14.4:5060;branch=8dJXAX9MDv
Max-Forwards: 70
To: <sip:UAallee@172.27.14.53>
From: <sip:UAcaller@172.27.14.4>;tag=ds17aa9bd4
Call-ID: 11022705439144@172.27.14.4
CSeq: 5 BYE
Content-Length: 0
```

Figura 2.12: SIP 200 OK

2.1.2.1 *Mais informações sobre SIP*

Para mais informações sobre o protocolo SIP, consulte (IETF, 2002) e (FIRESTONE; RAMALINGAM; FRY, 2007) (detalhes nas Referências deste trabalho).

2.2 Protocolos de Transporte de Mídia

Intro

2.2.1 Padrão RTP

Explica SIP

2.2.2 Padrão RTMP

Explica RTMP

2.3 Codificação de áudio

Intro

2.3.1 Padrão PCMA

Explica PCMA

2.3.2 Padrão PCMU

Explica PCMU

2.3.3 Padrão G.711

Explica G.711

2.3.4 Padrão Speex

Explica Speex

2.3.5 Padrão OPUS

Explica OPUS

2.4 Codificação de vídeo

Intro

2.4.1 Padrão H.263

Explica H.263

2.4.2 Padrão H.264

Explica H.264

2.4.3 Padrão VP8

Explica VP8

2.5 Tecnologias para Webconferência

Intro

2.5.1 Flash

Explica Flash

2.5.2 WebRTC

Explica WebRTC

3 SOLUÇÕES *OPEN SOURCE* PARA INTEROPERABILIDADE

Intro

3.1 Freeswitch

Explica FS

3.2 Asterisk

Explica Asterisk

3.3 Red5

Explica Red5

3.4 FFMPEG

Explica FFMPEG

4 ARQUITETURA

Escolhidos: Freeswitch, Red5 e FFMPEG *Implementados utilizando o Mconf*

4.1 Mconf

Intro

4.1.1 Arquitetura anterior

Arquitetura anterior do Mconf

4.1.2 Arquitetura modificada

Arquitetura do Mconf Modificada

5 RESULTADOS

Mostra interoperabilidade funcionando: Mconf com Ekiga, Polycom (SIP e H.323), MCU

REFERÊNCIAS

Adobe Connect: bringing together video and next-generation web conferencing. **White paper**, [S.l.], 2011.

FIRESTONE, S.; RAMALINGAM, T.; FRY, S. **Voice and Video Conferencing Fundamentals**. [S.l.: s.n.], 2007.

IETF. **SIP RFC 3261**. [Online; acessado em 4-Maio-2016], <https://tools.ietf.org/html/rfc3261>.

ITU. **H.225 Recommendation (12/09)**. [Online; acessado em 4-Maio-2016], <http://www.itu.int/rec/T-REC-H.225.0-200912-I/en>.

ITU. **H.323 Recommendation (12/09)**. [Online; acessado em 4-Maio-2016], <http://www.itu.int/rec/T-REC-H.323-200912-I/en>.

ITU. **H.245 Recommendation (05/11)**. [Online; acessado em 4-Maio-2016], <http://www.itu.int/rec/T-REC-H.245-201105-I/en>.

KUROSE, J. F.; ROSS, K. W. **Computer Networking: a top-down approach**, 6th edition. [S.l.]: Pearson, 2013.

ROESLER, V. et al. Mconf: an open source multiconference system for web and mobile devices. In: **Multimedia - A Multidisciplinary Approach to Complex Issues**. [S.l.]: InTech, 2012.

TANENBAUM, A. S.; WETHERALL, D. J. **Computer Networks, 5th Edition**. [S.l.]: Pearson, 2011.