Comparison of H.323 and SIP Video-Conferencing support

Norbert Oertel, Siemens AG, CT IC 2 MC (e-mail: norbert.oertel@siemens.com)

***Abstract*— ITU-T Rec. H.32x is still the predominant standards family for video conferencing room systems and multipoint controller units (MCU). Although most MCUs and room systems support SIP as a second signaling protocol some more advanced functionality is available with H.323 signaling only. Analysis reveals that this seems to be not due to missing, standardized signaling primitives. Lack of broad support of these features in existing end points and infrastructure components and the fear of potential interoperability issues seem to hinder their adoption.**

***Index Terms*— Conferencing, H.323, SIP, Video**

I. Introduction

VIDEO conferences using multipoint controller units (MCU) are tightly coupled; each conference participant establishes a one-to-one signaling (e.g. H.323 or SIP) and media (audio and video over RTP/RTCP) relationship with the central multipoint controller unit. Media streams are mixed within the multipoint controller unit so that each conference participant gets the notion of being part of one conversation space with all the other conference participants within which everybody can hear and see each other.

The fundamental requirement on the signaling protocols H.323 and SIP is to set up a basic call between the communication endpoint (room system, softclient, video enabled IP phone,...) and the MCU including the negotiation of audio, video, presentation and control streams. Call setup and tear-down is one of the most essential features which require either SIP or H.323 signaling. After this step there is either a SIP or H.323 based signaling as well as an in-band user interface relationship between the MCU and the endpoint.

A second requirement is security. Since ethernet is a shared medium in contrast to the POTS network, IP based communication which is not adequately protected by security measures could be easily subject to eavesdropping, SPIT and other attacks. Hence countermeasures have to be used in order to make voice and video over IP as reliable and secure as plain old telephony.

Third, far end camera control (FECC) based on H.224/H.281 media streams is often used to control the MCU (in a way similar to DTMF) as well as to allow remote parties to control pan, tilt and zoom of the local camera. Since the functionality is quite common for H.323 based room systems and it is the only way to enable others to control the local camera, it is required to be supported with SIP based signaling as well.

And fourth, since video conferencing is much more demanding with respect to required network bandwidth and computational resources, especially when HD video is used, flow control mechanisms and media renegotiation mechanisms are used by some MCUs in order to limit bandwidth, frame rate or size of the video of individual participants. This network adaptiveness increases the scalability of a video conferencing service and increases the user experience when congestion within the network occurs.

Based on these identified requirements we will first have a look at the SIP and H.323 based session setup and media negotiation procedures and how they compare. Then we will have a closer look at the media level and see how sharing desktops and presentations using a second video stream (H.239) and far end camera control (FECC) can be used together with SIP based signaling. Finally we will walk through the security features of SIP and H.323.

II. SIP signaling support

A. Session Setup

Session setup and control for a tightly-coupled conference using an MCU are less demanding than for e.g. the loosely-coupled, fully-meshed scenario since every participating client is just in a basic call with the MCU. Negotiating video, however, in contrast to audio only use cases, is a bit more complex because it may require more than just one round-trip in order for both end systems to agree on the video codec and parameters like profile or level among many other possible parameters.

One big difference between H.245, which is the signaling protocol within the H.323 framework, and SIP is the way capability exchange and media negotiation is done. With H.245 based signaling both parties first send their capability set describing exactly which media codecs they support and their limitations like e.g. profiles and levels as well as limitations of concurrent use of codecs. After that both parties can open logical channels given that the media encoding intended to be used for the channel does not exceed the now known limitations of the remote party.

SIP uses a combined approach instead. Together with the INVITE for a dialog usually a media description based on SDP is offered to the remote party. This offer describes the number of media streams intended to be used, the type of media and transport to be used, which codecs, their profiles and levels and other parameters the offering party is able to receive and/or send and network parameters like IP address and ports to be used for the RTP sessions. The offer thus combines the media streams the offerer would like to open and a description of a subset of the offerers capabilities.

The remote party responds with a media description based on the offer, within which the list of media streams, their codecs and parameters is limited according to its own capability set. If there is no common media format at all, the call setup is considered failed.

If one considers e.g. a SDP description with which the offerer expresses its willingness to send and receive an audio stream and a video stream using H.264 base profile at level 3.1 as the only offered video codec, then the remote party can agree with this offer only if it supports H.264 base profile at a level of 3.1 or higher. If it supports for instance just level 2.0 then is has to reject the video offer and accept the audio stream as the only common media between both parties.

However, the called now knows that its supported video encoding (H.264 base profile at level 2.0) is a subset of the capabilities which have been previously offered by the remote caller. It can now generate a re-INVITE reactivating the video media stream by offering H.264 baseline profile at level 2.0.

B. Dynamic flow control, packet loss and media re-negotiation

Depending on the mixing mode of the MCU it may be the case that certain conference participants are visible only within a small portion of the screen (e.g. at the side or the bottom of a continuous presence layout) or are not visible at all, which is the case for all participants but the currently active speaker when the MCU operates in voice activated switching mode.

Since high bandwidth for video is a scarce ressource and processing high resolution video uses a lot of computational power it makes perfect sense to reduce the bandwidth, the frame size and the frame rate sent by these participants to the MCU up to the point where currently invisible participants stop sending video at all. This behavior of the communication endpoints is triggered and controlled by the MCU, for which it uses signaling of flow control requests and re-negotiation of video codec parameters. This mechanism also allows the MCU to limit the bandwidth for participants for which it detects severe packet loss.

1) H.323 signaling approach

In the case of H.323 based signaling MCUs today make use of the H.245 FlowControlCommand in order to limit the bandwidth of a specific media channel or the whole multiplex. Endpoints are required to comply with this bandwidth limitation request and send a FlowControlIndication after having adjusted the outgoing bandwidth. If a FlowControlCommand requests an endpoint to limit the outgoing bandwidth of a media stream to 0, then the endpoint has to stop sending media on the logical channel. H.245 defines also LogicalChannelRateRequest, LogicalChannelRateAcknowledge, LogicalChannelRateReject and LogicalChannelRateRelease messages for requesting changes in bitrate of a specific logical channel in a more interactive way than the strict enforcement of a maximum bitrate limit as is done with the FlowControlCommand.

The H.245 RequestMode command allows the MCU to tell the endpoint about its preferences for video encodings, frame sizes, video bitrates and use of optional annexes of H.263. The frame rate is not specified within the RequestMode request - it is determined by the encoder dynamically based on frame size, available bitrate for the stream, the scene complexity and a temporal spatial tradeoff parameter, which is adjustable via the H.245 LogicalChannelRate command. The temporal spatial tradeoff parameter allows to specify a tradeoff between highest possible framerate and best image quality.

If packets have been lost, this may have severe impact on the video quality since video is coded predictively and decoding errors are propagated from one frame to the next until a refresh takes place. These ``refreshing frames" are called INTRA coded frames. H.245 provides several commands for a receiver to request complete or partial INTRA updates with the videoFastUpdatePicture command being the most widely used one.

2) SIP signaling approach

The wealth of H.245 flow control and mode request commands maps directly to a SIP re-INVITE initiated by the MCU as defined by RFC 3261 (``SIP: Session Initiation Protocol"), RFC 2327 (``SDP: Session Description Protocol") and RFC 3264 (``An Offer/Answer Model with the Session Description Protocol (SDP)"). The overall process of modifying parameters and encodings of an existing session is described in detail in section 8 of RFC 3264.

Flow control, that means adjusting the bandwidth of a single stream or the complete session, is achieved by generating a new SDP media description based on the last negotiated media description for which only the session or media level bandwidth attributes are changed to meet the new bandwidth requirements. If a bandwidth attribute specifies a zero bandwidth for a given media stream, the behavior is the same as with H.245 - the sender is requested to stop sending media for this stream. This procedure is described in detail in RFC 3264 section 8.3.4.

Requesting a video encoding with different codec, frame size or bitrate is done the same way by generating a new SDP media description based on the previously negotiated one with which it is possible to renegotiate essentially all parameters. However, the mapping from RTP payload type to media encoding has to stay the same for the whole duration of a session. This is described in detail in section 8.3.2 of RFC 3264.

Requesting INTRA updates is currently implemented by major MCU and room system vendors based on draft-levin-mmusic-xml-media-control. This draft describes a method of sending a fast INTRA update request as a XML body of a SIP INFO message. It is however controversially discussed because sending these update requests on the signaling path wastes server bandwidth and implies higher latency than signaling the message on the media path (the main reason for this is that there are usually several proxies within the signaling path).

The specification of codec control messages to be exchanged via the RTCP feedback channel is currently work in progress (draft-ietf-avt-avpf-ccm). It allows for timely feedback and signaling between sender and receiver of a video stream with lower overhead and shorter delay than the SIP INFO or H.245 based methods. Together with the already finished audio video profile with feedback specification (RFC 4585) this forms an extremely powerful framework for improving error robustness and visual quality over lossy networks.

III. Video conferencing specific media support

A. Far end camera control (FECC)

Far end camera control is the process of controlling camera parameters (pan, tilt and zoom) of a remote camera. The remote control of a video conferencing system supporting far end camera control usually provides arrow keys for controlling pan and tilt and two separate keys for controlling the camera zoom. Target for camera control can be either the local camera or the remote camera - in the latter case the video conferencing system generates far end camera control messages which it sends to the remote party. The remote party may act on reception of these messages by adjusting pan, tilt and zoom of its camera.

In the case of MCU based video conferencing it is the MCU which receives these FECC messages from the connected clients. It is able to use them in two different ways, depending on the context the MCU is in:

- For in-band user interface purposes. For instance for navigation through menus, selecting menu items or moving the focus between screen areas in continuous presence layouts. Usually this is combined with DTMF based user input generated via the keypad on the remote control.
- Routing the FECC messages received from one conference participant to another conference participant for actually controlling a far end camera.

Usage of far end camera control in the context of H.323 is defined by Annex Q of ITU-T Rec. H.323 which details on how H.281/H.224 messages are to be used and sent over an RTP/UDP/IP transport.

H.224 defines a real-time control protocol for applications requiring a low-latency and low-overhead means of sending unreliable messages between conference participants. The H.224 message header consists among other information of an identifier for the target application (Far end camera control as described in H.281 is the only registered application at this time) and the terminal address of the sender and recipient of the message consisting of a MCU / terminal ID tuple. The address 0/0 is used as a broadcast address.

H.281 is an application on top of H.224. It defines the message syntax and semantics for

- selecting a remote camera for coding and transmission
- starting and stopping camera movements
- storing and restoring the current camera position

IETF RFC 4573 in turn registers a ``h224" media type for use of H.281 over H.224 with SDP based media negotiation. This makes it possible to negotiate a H.224 session as described in H.323 Annex Q between a SIP client and an MCU in an interoperable way. This feature is already supported by major MCU and room system vendors.

B. H.239 presentations

H.239 is a technology which allows sharing graphical content like presentation slides, applications, complete desktops or other external video sources with other conference participants. The content is treated as a second video stream (subject to video compression as the primary video stream of a participant; thus the term DualVideo) provided by one source and distributed to all other participants in a conference. The most obvious difference to the ITU-T Rec. T.120 standard suite is that there is no backchannel for jointly manipulating documents. It is thus not possible to navigate or insert text within a document from the remote. It adds the following functionality to H.323:

- One-way transmission of media streams. Listeners to a presentation are just receivers of an additional video stream - they do not provide one.
- A method to attach roles to streams (``Live" and ``Presentation"). This allows both the client devices and the MCU to distinguish between the remote party camera and the presentation stream and choose an appropriate layout.
- A method to manage access to the presentation stream. A presentation role stream is subject to mutual exclusion. A sender needs to acquire a token in order to be allowed to provide a presentation stream to others.

SIP uses SDP (session description protocol) for negotiating media and thus cannot make direct use of H.239 signaling. It is however possible to map the functionality to corresponding SIP primitives.

The SDP offer/answer model allows streams to be negotiated as send only, receive only or for both sending and receiving. This is done by using the corresponding “sendonly”, “recvonly” and “sendrecv” attributes to the media description of the stream within the SDP body.

The IETF RFC 4796 defines a new media level attribute “content” which provides a means to label any stream negotiated via SDP with one or more content related attributes. RFC 4796 defines the content types “slides”, “speaker” and "sl" (short for sign language) as well as the content sources "main" and "alt" (short for alternate). The H.239 "Live" role maps to the "speaker" content type, whereas the "Presentation" role maps to the "slides" content type.

Mutual exclusion for access to the presentation stream (floor control) could be managed by using basic SIP signaling primitives (using SIP re-INVITE) for granting or revoking "sendrecv" or "sendonly" status to devices for the presentation stream (the same way as is done for implementing the hold/resume call control feature), but such behavior is not explicitly defined in RFC 4796. This signaling based mutual exclusion mechanism could be complemented by an MCU specific in-band or out-of-band floor control functionality.

IV. SIP SECURITY FEATURES

With both H.323 and SIP call signaling and media transfer is done over an insecure IP network shared with other applications and services. VoIP calls are subject to the following security threats (without qualifying for completeness):

- Denial-of-service
- Eavesdropping
- Tampering (e.g. Man-in-the-middle attack)
- Call or Registration hijacking (e.g. tearing down calls)
- Spam

ITU-T Rec. H.235 is an optional extension of the H.323 framework which addresses the specific security needs of VoIP applications. It provides for

- Authentication of users
- Integrity of signaling messages and data
- Confidentiality of the signaling channel via TLS or IPSec
- Confidentiality of media channels by providing means for key exchange and support for encrypted media transport

A. Authentication

SIP is in many respects very similar to HTTP. It is thus natural that SIP reuses the HTTP authentication mechanism described in RFC 2069 An Extension to HTTP: Digest Access Authentication and RFC 2617 HTTP Authentication: Basic and Digest Access Authentication. Its use for SIP is detailed in RFC 3261 section 22. Use of basic authentication is deprecated by RFC 3261 because of its inherent security issues.

There is a significant difference between SIP and HTTP in that SIP messages may be routed through many application layer hops. RFC 3261 details the use of digest authentication by proxies and user agent servers (UAS) and explicitly defines the proxy behavior when call forking takes place and several UAS or proxies might request the originating user to provide his credentials.

B. Integrity

SIP messages may contain MIME bodies. This is e.g. used to exchange SDP media descriptions between communication endpoints. Analogue to e-mail it is also possible to use S/MIME bodies instead which allows for digitally signing and/or encrypting message bodies. This is described in detail in RFC 3261 section 23.

Prerequisite for being able to use S/MIME message bodies is an existing public key infrastructure for distributing and managing the public and private keys of message recipients and a per user certificate asserting that the given user is identified by a given address-of-record.

One use case for using S/MIME is to protect the SDP message body between two communication endpoints. Another use case is to provide end-to-end integrity and to some extent privacy of SIP headers by tunneling a copy of the sip message as a signed or encrypted message/sip body. The ultimate message body like e.g. the SDP media description should be attached to the inner sip message in order to benefit from the additional security features. Section 23.4 of RFC 3261 details the exact procedure of integrity checking, which header fields have to be sent as plain text and how to merge the protected inner SIP message with probably modified outer SIP header fields. However this leads to problems at network boundaries since intermediate nodes do not know the ports on which audio/video will be exchanged.

On an application-layer hop-by-hop basis IPsec or transport level security (TLS) can be used to guarantee message integrity. This also guarantees message integrity along a path of trusted SIP entities. The end systems have however no means of determining whether a message has been tampered with even if both have sent and received them via a secure connection. This is because the message might get routed through multiple application layer entities, some possibly not trustworthy, and because there is no guarantee that for every hop TLS or IPsec security has been used.

C. Confidentiality

Secure media transmission between endpoints requires exchange of encryption keys between end systems. The key exchange makes in turn use of some pre-shared secret, an existing public key infrastructure or the confidentiality of the signaling channel.

As has been described earlier, S/MIME may be used to encrypt the message body and at least some of the SIP header fields which are not important for routing the SIP messages. IPsec and TLS can be used to provide for confidentiality and integrity of the complete SIP signaling channel on a hop-by-hop basis. A protected SDP part can in turn be used to exchange keys in a secure way in order to encrypt the media streams using SRTP. This is detailed in RFC 4568 Session Description Protocol (SDP): Security Descriptions for Media Streams.

Another key exchange procedure which can be used in conjunction with SIP is MIKEY (RFC 3830 MIKEY: Multimedia Internet KEYing and RFC 4567 Key Management Extensions for Session Description Protocol (SDP) and Real

Time Streaming Protocol (RTSP)).

Similar or even identical methods are also used for key exchange within the H.235 security framework for H.323. Use of MIKEY is specified in H.235.7 and a similar approach to sdes is described in H.235.8.

A method for key exchange within the media path (in contrast to the SIP signaling path which is used for both MIKEY and sDescriptions based key exchange) is ZRTP, currently being discussed at the IETF (draft-zimmermann-avt-zrtp-03.txt).

The actual secure media transfer is done using SRTP (RFC 3711 The Secure Real-time Transport Protocol (SRTP)) which provides for confidentiality of both the exchanged media and control messages, message authentication and for protection against replay-attacks. This is the same transport used within the H.235 security framework for H.323.

V. Summary and Conclusion

In the previous section we went through the SIP and H.323 based session setup and media negotiation procedures, media level functionalities like far end camera control (FECC), presentation sharing using a second video stream (H.239) and lip synchronization, and finally had a look at the security features of the two protocol suites. We found out, that from a standardization point of view both protocol suites are at a par, however the level of support for these functionalities in today's products, both endpoints and MCUs, differs substantially between H.323 and SIP.

Whereas all of the above discussed features are supported with H.323 based signaling in an interoperable way by most products and vendors, the situation with SIP is completely different:

• Session setup and media negotiation works as expected for most products, however media renegotiation and flow control on behalf of clients is not implemented within current MCU products despite being adequately specified in RFC 3261 and RFC 3264. Lack of support of the latter features seems to be mainly due to interoperability issues and concerns.

• Far end camera control using H.281 over H.224 as specified in RFC 4573 is implemented by only some of the vendors. From a specification point of view this feature seems to be ready to be adopted by other vendors soon.

• Sharing presentations using SIP via a second video stream similar to H.239 is not implemented by any of the vendors. At least some of the MCU products have the ability to feed presentations out of band e.g. using the VNC protocol and to view presentations as just another stream in continuous presence layouts or via an out-of-band stream offered via a co-located streaming server. The basic signaling primitives for implementing this feature are available and standardized, however how to build up the functionality from these primitives leaves a large amount of freedom - a best practices document and a reference implementation could help to remedy this situation.

• HTTP Digest Authentication is supported by most of the vendors. Support for signaling integrity using TLS is increasing, however support for S/MIME and media encryption is not implemented within the products of the major vendors. It is likely that the reason for this is problems with key management (secure exchange of encryption keys between communication partners) in a heterogeneous communication infrastructure environment.

For this reason H.323 can be considered more mature than SIP when used for video conferencing. In order to improve the adoption of the features missing in products right now we propose to

• Motivate vendors to implement proper behavior for SIP re-INVITES in order to support media re-negotiation and flow control for increased network adaptivity.

• Motivate vendors to implement FECC according to RFC 4573.

• Work on a Best Practices RFC within the IETF community and build a reference implementation for presentation sharing using a second video stream similar to H.239.

• Because of the diversity of the SIP specification including numerous IETF RFC's it is worth thinking about defining profiles and levels of support for these RFC's. This could range e.g. from a simple baseline profile including just the basic necessities up to a HD video conferencing profile with support for presentation sharing. Defining a strict conformance testing procedure and some sort of logo like ``SIP HD Videoconf profile compliant" could help distinguish well designed and behaved SIP systems from simple ones implementing just parts of the specifications.

References

[1] A. Noack, ``Vergleich von H.323 und SIP", Studienarbeit and der Universität Rostock; Rostock 2003.

[2] K. Kostas, G. Parissidis and B. Plattner, ``A Comparison of Frameworks for Multimedia Conferencing: SIP and H.323", in: 8th IASTED International Conference on Internet Multimedia Systems and Applications (IMSA 2004).

[3] P. Papageorgiou, ``A Comparison of H.323 vs SIP", unpublished, University of Maryland, 2001.

[4] J. Ho, J.-C. Hu, and P. Steenkiste, ``A Conference Gateway Supporting Interoperability Between SIP and H.323", in: MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia, p. 421-430.

[5] J. Rosenberg, ``A Hitchhiker's Guide to the Session Initiation Protocol (SIP)", IETF draft-ietf-sip-hitchhikers-guide-02, 2007.

[6] H. H. Taha, ``Architecture for a SIP-Based Conferencing Server", Diploma Thesis at Fachhochschule Mannheim; Mannheim 2005.

[7] D. Song, Y. Mo and F. Wang, ``Architecture of multiparty conferencing using SIP", in: International Conference on Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005, p. 1361-1364.

[8] K. Singh, G. Nair, and H. Schulzrinne, ``Centralized conferencing using SIP", in: Internet Telephony Workshop http://citeseer.ist.psu.edu/singh01centralized.html

[9] H. Schulzrinne, J. Rosenberg, ``A Comparison of SIP and H.323 for Internet Telephony", Network and Operating System Support for Digital Audio and Video (NOSSDAV), Cambridge, England, July 1998.

[10] I. Dalgic and H. Fang, ``Comparison of H.323 and SIP for IP Telephony Signaling", in: Proc. of Photonics East, Boston, Massachusetts, Sept. 1999.

[11] I. Miladinovic and J. Stadler, ``Multiparty Conference Signalling using the Session Initiation Protocol (SIP)",in: Proceedings INC 2002, p. 191 - 198.

[12] J. Glasmann, W. Kellerer and H. Müller, ``Service Architectures in H.323 and SIP – A Comparison", http://citeseer.ist.psu.edu/glasmann03service.html, 2003.

[13] J. Ott, ``SIP Conferencing", presentation at IIR SIP Congress 2001; Stockholm, 2001.

[14] J. Ott, ``SIP Conferencing", presentation at Upperside SIP 2003; Paris, 2003.

[15] International Telecommunications Union, ``Visual telephone systems and equipment for local area networks which provide a non-guaranteed quality of service", ITU-T Recommendation H.323, July 2003.

[16] International Telecommunications Union, ``Role management and additional media channels for H.3xx-series terminals", ITU-T Recommendation H.239, September 2005.

[17] International Telecommunications Union, ``Control protocol for multimedia communication", ITU-T Recommendation H.245, May 2006.

[18] International Telecommunications Union, ``Multiplexing protocol for low bit rate multimedia communication ", ITU-T Recommendation H.223, July 2001.

[19] International Telecommunications Union, ``A real time control protocol for simplex applications using the H.221 LSD/HSD/HLP channels.", ITU-T Recommendation H.224, February 2000.

[20] International Telecommunications Union, ``A far end camera control protocol for videoconferences using H.224", ITU- T Recommendation H.281, November 1994.

[21] International Telecommunications Union, ``H.323 security: Framework for security in H series (H.323 and other H.245-based) multimedia systems", ITU- T Recommendation H.235.0, September 2005.

[22] International Telecommunications Union, ``H.323 security framework: Usage of the MIKEY key management protocol for the Secure Real Time Transport Protocol (SRTP) within H.235", ITU- T Recommendation H.235.7, September 2005.

[23] International Telecommunications Union, ``H.323 security: Key exchange for SRTP using secure signalling channels", ITU- T Recommendation H.235.8, September 2005.

[24] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, ``SIP: Session Initiation Protocol", RFC 3261, June 2002.

[25] M. Handley and V. Jacobson, ``SDP: Session Description Protocol", RFC 2327, April 1998.

[26] J. Rosenberg and H. Schulzrinne, "An Offer/Answer Model with SDP", RFC 3264, June 2002.

[27] O. Levin, R. Even and P. Hagendorf, ``XML Schema for Media Control", draft-levin-mmusic-xml-media-control-10, May 2007.

[28] S. Wenger, U. Chandra, M. Westerlund and B. Burman, ``Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", draft-ietf-avt-avpf-ccm-09, August 2007.

[29] J. Ott, S. Wenger, N. Sato, C. Burmeister and J. Rey, ``Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.

[30] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, ``RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003.

[31] H. Schulzrinne and S. Casner, ``RTP Profile for Audio and Video Conferences with Minimal Control", RFC 3551, July 2003.

[32] J. Hautakorpi and G. Camarillo, ``The Session Description Protocol (SDP) Content Attribute", RFC 4796, February 2007.

[33] R. Even and A. Lochbaum, ``MIME Type Registration for RTP Payload Format for H.224", RFC 4573, July 2006.

[34] B. Reeves and D. Voelker, ``Effects of Audio-Video Asynchrony on Viewer's Memory, Evaluation of Content and Detection Ability", Research Report Prepared for Pixel Instruments, http://www.lipfix.com/file/doc/reeves_and_voelker_paper.pdf; Stanford University, October 1993.

[35] G. Camarillo, G. Eriksson, J. Holler and H. Schulzrinne, ``Grouping of Media Lines in the Session Description Protocol (SDP)", RFC 3388, December 2002.

[36] J. Franks, P. Hallam-Baker, J. Hostetler, P. Leach, A. Luotonen, E. Sink and L. Stewart, ``An Extension to HTTP : Digest Access Authentication", RFC 2069, January 1997.

[37] J. Franks, P. Hallam-Baker, J. Hostetler, P. Leach, A. Luotonen, E. Sink and L. Stewart, ``HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[38] J. Lennox, X. Wu and H. Schulzrinne, ``Call Processing Language (CPL): A Language for User Control of Internet Telephony Services", RFC 3880, October 2004.

[39] J. Arkko, F. Lindholm, M. Naslund, K. Norrman and E. Carrara, ``Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP)", RFC 4567, July 2006.

[40] F. Andreasen, M. Baugher and D. Wing, ``Session Description Protocol (SDP) Security Descriptions for Media Streams", RFC 4568, July 2006.

[41] P. Zimmerman, A. Johnston and J. Callas, ``ZRTP: Media Path Key Agreement for Secure RTP", draft-zimmermann-avt-zrtp-03, March 2007.

[42] M. Baugher, D. McGrew, M. Naslund, E. Carrara and K. Norrman, ``The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[43] J. Arkko, E. Carrara, F. Lindholm, M. Naslund and K. Norrman, ``MIKEY: Multimedia Internet KEYing", RFC 3830, August 2004.

[44] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.