Intelligent Multimedia Technologies for Networking Applications:

Techniques and Tools

Dimitris N. Kanellopoulos
University of Patras, Greece

Information Science
REFERENCE

Managing Director: Lindsay Johnston
Editorial Director: Joel Gamon
Book Production Manager: Jennifer Yoder
Publishing Systems Analyst: Adrienne Freeland
Development Editor: Myla Merkel
Assistant Acquisitions Editor: Kayla Wolfe
Typesetter: Christy Fic
Cover Design: Jason Mull

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Intelligent multimedia techologies for networking applications: techniques and tools/Dimitris N. Kanellopoulos, editor.
p. cm.
Includes bibliographical references and index.
Summary: “This book promotes the discussion of specific solutions for improving the quality of multimedia experience while investigating issues arising from the deployment of techniques for adaptive video streaming”--Provided by publisher.
ISBN 978-1-4666-2833-5 (hardcover) -- ISBN 978-1-4666-2834-2 (ebook) -- ISBN 978-1-4666-2835-9 (print & perpetual access) 1. Multimedia communications. 2. Integrated services digital networks. 3. Image transmission. 4. Webcasting. I. Kanellopoulos, Dimitris N., 1967-
TK5105.15I545 2013
006.7--dc23
2012032520

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

Chapter 3
Protocol Interactions among User Agents, Application Servers, and Media Servers:
Standardization Efforts and Open Issues

Alessandro Amirante
University of Napoli Federico II, Italy

Tobia Castaldi
University of Napoli Federico II, Italy

Lorenzo Miniero
University of Napoli Federico II, Italy

Simon Pietro Romano
University of Napoli Federico II, Italy

ABSTRACT

In this chapter, the authors focus on the complex interactions involving the various actors participating in a multimedia session over the Internet. More precisely, bearing in mind the current standard proposals coming from both the 3GPP and the IETF, they investigate some of the issues that have to be faced when separation of responsibilities comes to the fore. The scenario the authors analyze is one in which one or more user agents are put into communication with a media server through the mediation of an application server. In such scenario, the application server does play the role of a middlebox for all that concerns signaling, since it is responsible for the transparent negotiation of a session among the entities (the user agents on one side and the media server on the other) that will be exchanging media during the communication phase. In this chapter, the authors highlight that protocol interactions become really complex under the depicted circumstances. They provide a survey of the current standardization efforts related to media control, together with a discussion of open issues and potential solutions.

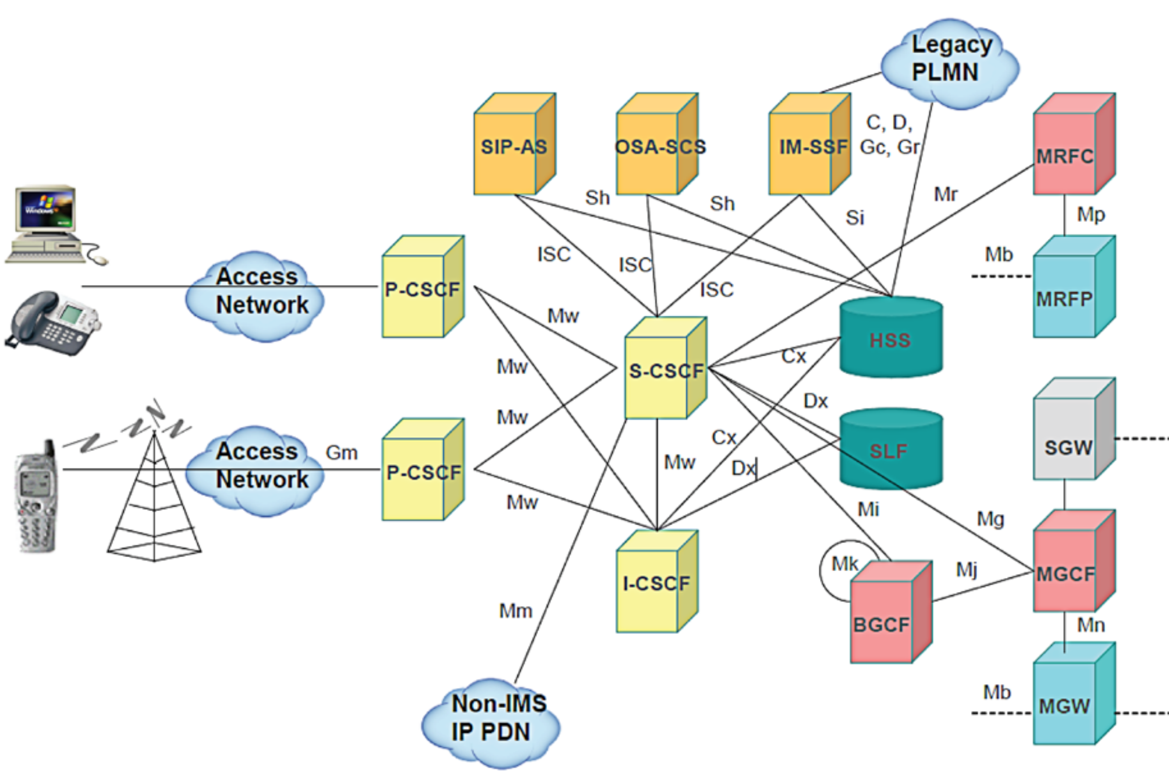
DOI: 10.4018/978-1-4666-2833-5.ch003

INTRODUCTION

Recently, advanced services have massively entered the Internet arena pushed by the revolutionary "global" approach envisaging the coexistence of a variegated portfolio of applications on top of an integrated IP-based network. Consequently, the Internet has become a place where an ever-increasing number of "dependent" or "correlated" transactions take place every day. This unexpected growth of complexity unavoidably unveils a number of less or more subtle issues that have to be faced when looking at the interactions among the various entities involved in the service delivery chain. Standardization bodies like the IETF (Internet Engineering Task Force) and the 3GPP (3rd Generation Partnership Project) are actively contributing both to the definition of an integrated framework for advanced service creation and deployment and to the solution of the above mentioned issues. As to the 3GPP, the consortium is currently standardizing the *IP Multimedia Subsystem* (IMS) architecture (see Figure 1), whose aim is to provide a common service delivery mechanism capable to significantly reduce the development cycle associated with service creation across both wireline and wireless networks.

The main objective of IMS resides in trying to reduce both capital and operational expenditures (i.e., CAPEX and OPEX) for service providers, at the same time providing operational flexibility and simplicity. Since the beginning, the IMS has chosen SIP (Session Initiation Protocol) (Rosenberg, et al., 2002) as the main signaling protocol among most of its components (3GPP, 2007). The envisaged portfolio of IMS services includes advanced IP-based applications like Voice over IP (VoIP), online gaming, videoconferencing, and content sharing. All such services are to be provided on a single, integrated infrastructure, ca-

Figure 1. The architecture of the 3GPP IP multimedia subsystem (IMS)



pable to offer seamless switching functionality between different services. It is worth noting that IMS is conceived as an access agnostic platform. This requirement clearly imposes a careful study of the core IMS components (such as Call/Session Control Function—CSCF, Home Subscriber Server—HSS, Media Resource Function—MRF, and Application Server—AS), which must be scalable and able to provide advanced features, like *five nine* reliability. A more-in-depth analysis of the IMS architecture is reported in Appendix A.

Similarly, the IETF is devoting a great effort to the definition of advanced frameworks for multimedia service delivery, starting from the effective utilization of the base functionality made available by the SIP protocol. SIP provides users with the capability to initiate, manage, and terminate communication sessions in an IP network. For a brief description of the SIP protocol and architecture, see Appendix B. The main working groups within the IETF involved in the standardization of advanced multimedia services belong to the Real-Time Applications and Infrastructure (RAI) Area. Among them, the *MEDIACTRL* Working Group focused on the general definition of an appropriate way for an Application Server (AS) to control a Media Server (MS) in order to provide users with a set of advanced services like Interactive Voice Response (IVR) (McGlashan, et al., 2011) and conferencing (McGlashan, et al., 2012). At the time of this writing, the working group is almost closed, since all the envisaged milestones have been met.

The chapter is organized as follows. In the next section, we illustrate both the overall context of our work and the motivations, which inspired us to focus on AS-regulated interactions for our analysis of advanced service provisioning frameworks. Afterwards, we delve into some of the details needed in order to have a clear vision of the issues hidden behind an architecture built based on the principle of the separation of concerns among its inner components. After that, we analyze a number of interesting scenarios, which help the reader to understand how the issues identified can be dealt with as effectively as possible. Conclusions are provided in the last section. Finally, three appendixes expand a little bit on the IMS architecture and on the SIP and BFCP protocols.

CONTEXT AND MOTIVATION

From the brief discussion above, it comes out that the SIP protocol is actually paving ground for both 3GPP and IETF standardization efforts. Indeed, though work inside these two bodies is proceeding along independent tracks, there is currently a trend towards convergence at least at the level of the approach adopted to cope with both complexity and heterogeneity. This includes the choice of a fully distributed paradigm envisaging the definition of independent yet tightly interacting components, each responsible for a specific function inside the overall service architecture. As an example, the idea of separating responsibilities has been applied at the outset in the IMS, which basically identifies the various functions needed in a multimedia-capable IP network and tackles the issue of standardizing the interfaces among the entities implementing them. The same holds for the IETF: the idea of clearly separating the business logic (Application Server) from the data manipulation functionality (Media Server) goes exactly in the same direction.

In the following of this chapter, we will further elaborate on the mentioned issues. We will take the two referenced frameworks as leading examples of the current approach towards the definition of components acting, at various levels, as mediators between users and services. Our focus will be on a specific function of this new generation of networks, namely the delivery of multimedia services. Starting from the consideration that both the IMS and the IETF have identified two distinct roles for Application Servers and Media Servers (the so-called *Media Resource Function* of the IMS, which actually is a macro-component made of the two sub-elements called, respectively, *Media Resource Function Controller*—MRFC—and *Media Resource Function Processor*—MRFP), we will identify the Application Server as a critical entity acting as a middlebox which can highly influence the communication between users on one side and media servers on the other. We will show that such a role naturally imposes a tradeoff between service composition flexibility and service management complexity, thus calling up network engineers in order to try and strike the balance between these two counteracting facets of any advanced service delivery framework.

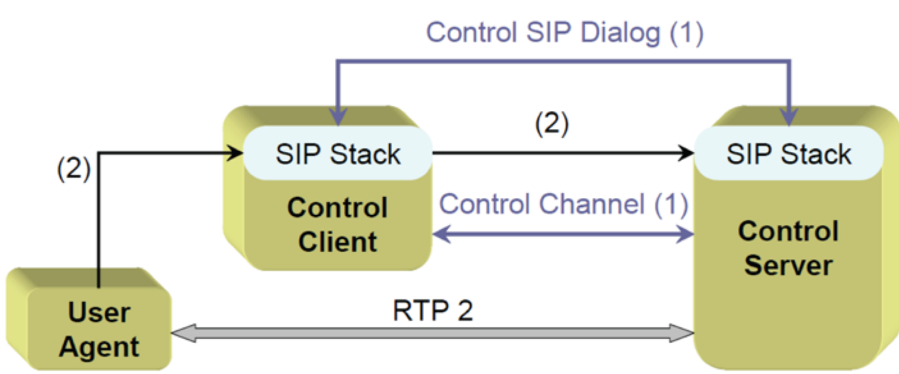
CONTROLLING A REMOTE MEDIA SERVER: ARCHITECTURE AND PROTOCOLS

The *Media Resource Function* (MRF) is logically decomposed in the MRFC and MRFP, whereas the former acts as a controlling interface towards the MRF, while the latter takes care of actually manipulating the media resources according to the directives provided by the controller. The interface between the above-mentioned components has been only partially defined so far in IMS. In fact, while it has been specified that the AS and the MRFC need to involve SIP in their interaction, the protocol details have not been standardized yet. To fill the gap due to such lack of standardization, the IETF Working Group called *MEDIACTRL* (Media Server Control) specified a complete architecture, called *Media Control Channel Framework* (MCCF) (Boulton, et al., 2011), for the interaction between Application Servers and Media Servers through the use of SIP. To map the work of the IETF onto the IMS architecture, the current specification maps the MCCF onto the MRF as a whole, meaning that it acts both as a controller and as a processor. Nevertheless, this distinction is not of great interest for the purpose of this chapter, since the MCCF actually acts as the SIP interface towards the MRFC as seen by IMS-compliant multimedia-aware Application Servers.

The architecture conceived in the MEDIACTRL specification is depicted in Figure 2. Please, note that the MCCF will be referred to as MS from now on.

The AS and the MS interact by means of two protocols: a dedicated protocol, called CFW (Control Framework), and SIP, as envisaged by the IMS specification. SIP is mainly used by the AS to attach UACs (User Agent Clients) needing multimedia resources to the MS, in order to let them negotiate media sessions: this is accomplished by making use of the 3rd Party Call Control mechanism (Rosenberg, et al., 2004). This call control mechanism envisages the AS as ter-

Figure 2. Protocols interaction between AS, MS, and UACs



minating endpoint for what concerns signaling, both for the UACs ad for the MS (while media flow directly via RTP between the UACs and the MS). Instead, CFW is the protocol used between AS and MS to explicitly manipulate the negotiated resources, e.g., to add users to a conference mix or to present them with IVR menus. This protocol is also negotiated by means of SIP, within the context of a so-called COMEDIA negotiation.

The use of 3PCC to give UACs access to media resources, and of CFW to drive how these media resources can be accessed or need to be presented, makes it quite clear that all the application logic resides in the AS, while media processing is achieved in MS. To make it even clearer, the AS can be seen as the brain, whilst the MS is the arm. Of course, considering that the directives the AS sends to the MS through the control channel assume they both are aware of the available resources (including UACs and their negotiated media streams), the 3PCC mechanism and the CFW protocol transactions need to be properly synchronized in some scenarios. Some of these scenarios and the issues they can arise will be dealt with in the following section.

For the sake of completeness, we inform the reader that all this study has been accomplished by also referring to our implementation of the MEDIACTRL framework. Such implementation (at the time of this writing) is the only available implementation of the framework, and is completely open source (MEDIACTRL, 2012).

SCENARIOS OF INTERACTION

Hereafter, we describe some typical use case scenarios, presenting the way they can be accomplished using the involved protocols. The scenarios are presented from a very high-level perspective, without delving too much into the details of the transactions contents. Actually, the focus is on the advantages and potential drawbacks of each presented approach, with special emphasis on the middlebox role of the Application Server. Sequence diagrams are added for a better understanding of the interactions among the involved parties. The scenarios are presented in order of increasing complexity:

- First, the interaction between two User Agent Clients (UAC) and a "self-sustained" Back-to-Back User Agent (B2BUA) is presented.
- Then, the focus will move to the interaction between two UACs through an Application Server (AS), which relays media manipulation to a separate Media Server.
- Finally, signaling is complicated by the addition of a floor control protocol to the list of resources to be negotiated.

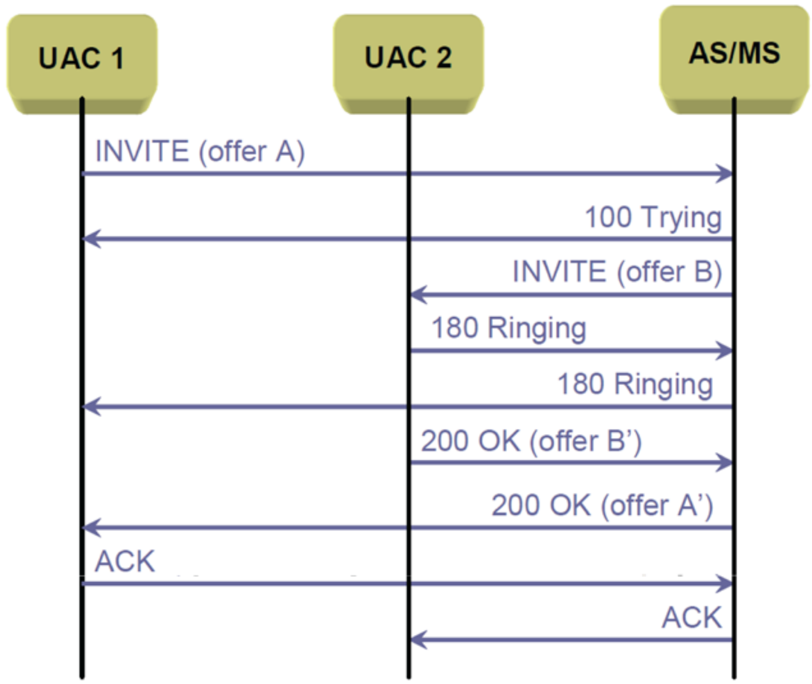
Self-Sustained B2BUA

The first scenario is by far the simplest one. In this scenario, a UAC (the caller) wishes to place a call to another UAC (the callee), and a B2BUA is in their way. The B2BUA presented in the scenario is a self-sustained one, in the sense that it acts as an integrated AS and MS: a typical example of such a middlebox is the popular open source PBX Asterisk2 (www.asterisk.org). The interaction between the caller and the callee is quite straightforward in this case, and is depicted as a sequence diagram in Figure 3.

There are very few issues in the signaling. In fact, considering that the AS is completely aware of the media functionality, since it actually is directly responsible for them, all the needed SDP answers and offers are immediately available, and the negotiation can be completed with no much hassle. The interaction can be summarized in the following steps (provisional responses are skipped for the sake of conciseness):

- The caller sends an INVITE addressed to the callee to the B2BUA; the INVITE con-

Figure 3. Phone call: back-to-back user agent



tains the SDP offer (A) with the media the caller wants to negotiate;

- The B2BUA matches the offer with its own capabilities (since it also acts as MS), and sends a new INVITE to the callee; this new INVITE carries as payload a new SDP offer (B), obtained by the previous match;
- In case the callee accepts the call (a 200 message), its SDP answer (B') is matched by the B2BUA as before; this answer is modified accordingly, if needed (A'), and then forwarded to the caller to complete the negotiation;
- As soon as the ACK from the caller is forwarded to the callee, the two UACs can interact; the B2BUA might have to act as a media transcoder between the two, if the negotiation presented such a need.

The call flow presents very few issues from a signaling point of view. The B2BUA has full control over both the signaling and the media capabilities, and can take care of all the decisions according to its policies, application logic, and resources availability. However, such approach presents more than one drawback. In fact, it implies that the media functionality has to be replicated at every AS needing it, since each such AS has to act as a completely self-sustained B2BUA. Besides, even focusing on a single AS, such an approach completely lacks in terms of scalability and locates in the B2BUA a single point of failure.

AS and MS Separated

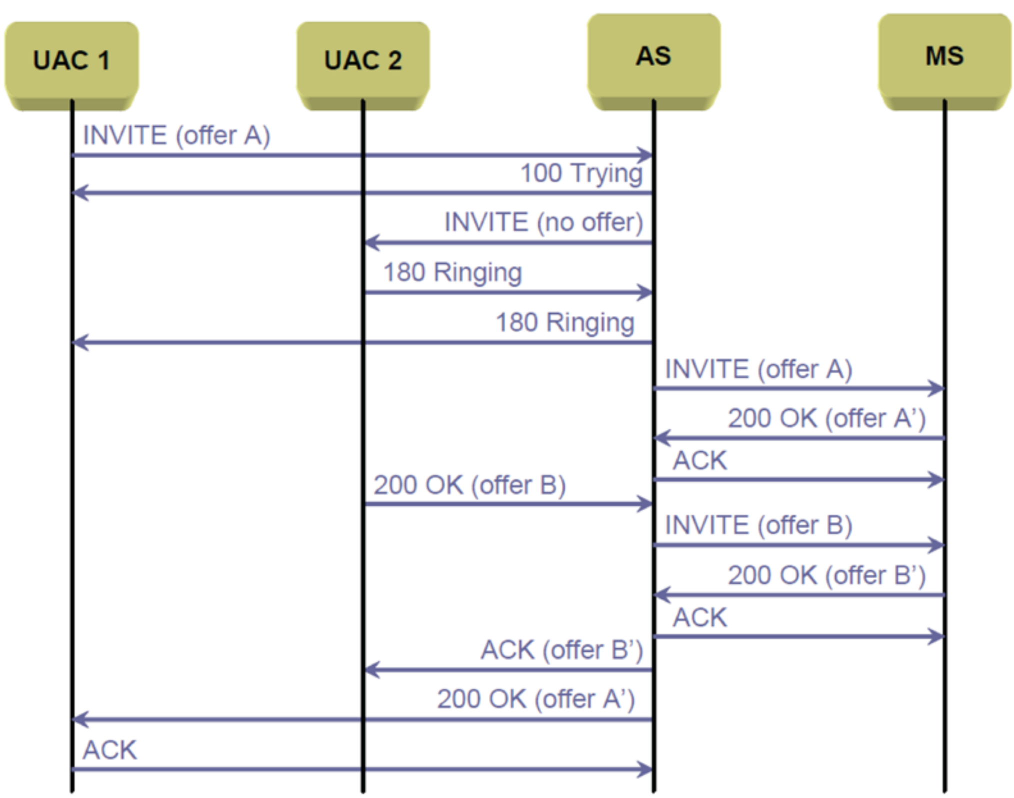
To somehow fill the gaps of the previously described B2BUA approach, a separation between the application logic and the media processing can be introduced. This can be done, as already explained in the introductory sections, with an approach a la MEDIACTRL. This is exactly the scenario that is presented hereafter. The provided example is again a phone call between two UACs, where the media flowing between them might need to be transcoded. In this scenario, the focus is on the so-called 3PCC (3rd Party Call Control). In fact, the AS needs to make use of 3PCC to attach both the calling and the invited UAC to the MS, in order to have the media negotiated between them, and subsequently manipulated through the control channel between the AS and the MS.

The 3PCC approach is quite more complex than the B2BUA one, considering that the SDP answers and offers to provide the UACs will in such case depend on the negotiation the AS relays to the MS: the AS might not be able to know in advance how the MS will negotiate the media with the UACs as it did as a self-sustained B2BUA, since in this case the negotiation may easily vary depending on policies in the MS and/or available resources. A sequence diagram of the scenario is depicted in Figure 4.

The call flow presents some more issues than those presented in the B2BUA case. Again, the scenario can be summarized in these steps:

- Once the AS receives an INVITE from the caller, it first invites the addressed callee to check if it is available; however, this time the AS sends a body-less INVITE to the callee, in order to have it offer all the media it supports; in fact, the AS cannot use the caller's offer in the INVITE, considering it might be heavily modified by the MS subsequently;
- At this point, the first 3PCC can take place; in fact, since an offer from the caller (A) is already available, the AS forwards it to the MS to have the caller negotiate its media with the MS; the SDP answer provided by the MS (A') is stored by the AS in order to

Figure 4. Phone call: allocating caller and callee



be able of relaying it later to the caller, that is as soon as the callee accepts the call;

- Once the callee accepts the call, providing in the 200 its offer (B), the second 3PCC can take place as well; the AS attaches the callee to the MS, and stores the negotiated answer (B');
- At this point, the AS has an SDP answer for both the caller and the callee. This means that the AS can complete the negotiation with both of them, sending the answer to the caller (A') in a 200 and the answer to the callee (B') in the final ACK;
- Now that the negotiation is complete, the AS can make use of the control channel to properly instruct the MS to attach the media connections of the two UACs with each other, thus allowing them to interact.

The additional complexity of the signaling in this scenario is immediately perceptible. It is worth noting that in this case the issue is not represented by the control channel interaction between the AS and the MS. In fact, even if such interaction always needs to be synchronized with the 3PCC in order to achieve the desired results, in the phone call case attaching the UACs can simply be done after both SIP negotiations have succeeded.

While the presented approach allows overcoming the limitations introduced by the B2BUA one, there are some drawbacks. One of them is related to the potentially premature allocation of resources. The caller, in the presented diagram, is allocated before knowing if the callee is available. This means that, if the callee rejects the call instead of accepting it as envisaged in the scenario, resources have been wasted, and must be de-allocated consequently. Moving the allocation of the caller after a 200 from the callee, solves this issue, but introduces a new one: in fact, the MS may be lacking the resources to allocate the UACs at that time, thus resulting in a media-less session between the UACs. Obviously, this is an undesirable behavior in an environment subject to billing policies.

Involving Moderation

The signaling can be further complicated by involving other protocols in the negotiation among the involved parties. In fact, all protocols in a session are often negotiated within the context of the same offer/answer. Besides, such protocols might need to be handled by the MS, meaning they would be part of the already mentioned 3PCC negotiation. An example of such protocols is the *Binary Floor Control Protocol* (BFCP) (Camarillo, et al., 2006). BFCP is a recently standardized floor control protocol, typically be involved in conferencing systems to allow moderated access to the available resources. More information about the BFCP protocol is reported in the Appendix C.

As specified in IMS, a MS may be invested with the additional role of floor control server. This means that the MS might not only have to deal with media in the SDP, but also with COMEDIA-based negotiations for BFCP as specified in Camarillo (2006). This negotiation is needed to provide the UAC with a list of attributes, including the transport address of the floor control server, as well as several BFCP-related identifiers. The problem introduced by this scenario is that, unlike media negotiation, the BFCP negotiation of a UAC with the floor control server can only be achieved after the UAC has been added to a conference through the control channel. However, the AS will not be able to correctly address the UAC in the control channel request before the UAC has been attached to the MS. This immediately suggests that the synchronization between 3PCC and the control channel transactions becomes of paramount importance. This section presents two possible approaches to deal with the described signaling issue (depicted in Figure 5 and Figure 6, respectively). Both use-cases present a single UAC interacting with the AS and the MS, instead of the caller and callee of the previous sections. In fact, in this case BFCP is assumed to be involved in a conferencing scenario, where the UAC places the INVITE to the AS to join a conference it is aware of. The approach (presented in Figure 5) relies on a re-INVITE generated by the MS to the UAC, in order to update the media session at a later moment.

Examining the signaling step by step:

- The UAC sends its INVITE to join the conference to the AS; the SDP offer includes BFCP-related lines as specified in Camarillo (2006); this INVITE is relayed to the MS as part of the 3PCC mechanism;
- As explained before, the BFCP negotiation is expected to fail, since no BFCP user identifier associated with the UAC is available yet; in fact, the UAC has not been added to the conference at the time of the INVITE; this results in the BFCP negotiation being refused;
- The SDP answer provided by the MS, which only includes the negotiated media is forwarded by the AS to the UAC;
- Now that the 3PCC negotiation is completed, the AS can add the UAC to the conference mix by means of the control channel; the control channel interaction between the AS and MS also includes directives related to BFCP, which results in a BFCP user identifier being associated with the UAC;
- At this point, the MS can trigger a re-INVITE addressed to the UAC, this time including all the BFCP-related identifiers as specified in Camarillo (2006); such re-

Figure 5. BFCP negotiation: reINVITE to the UAC

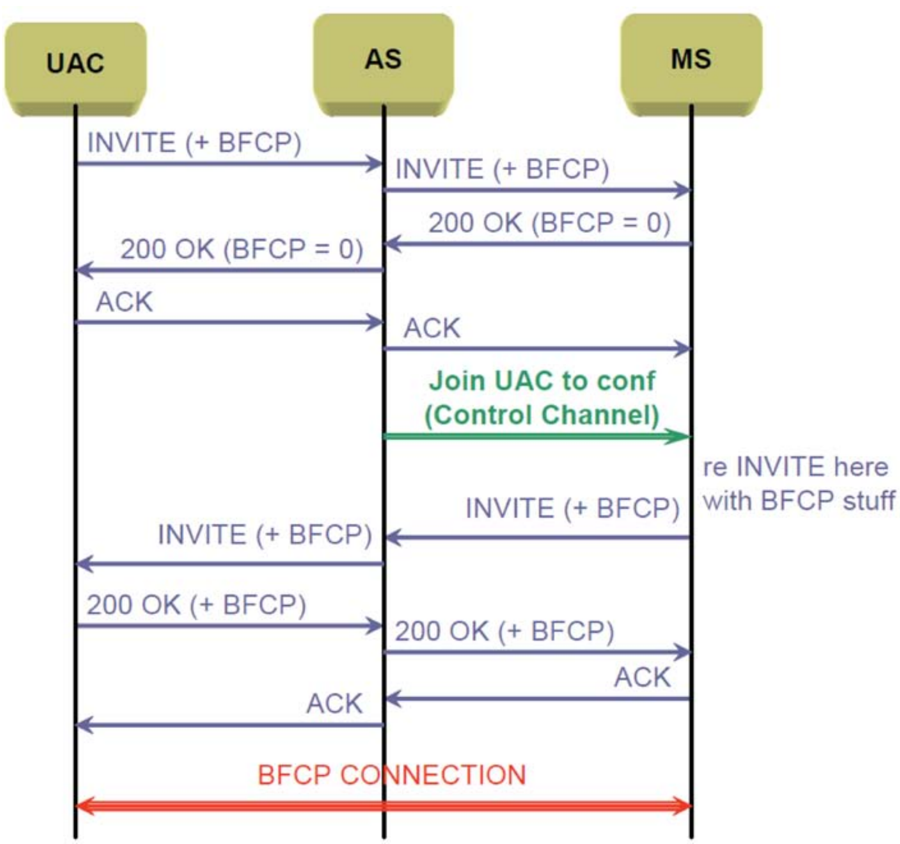
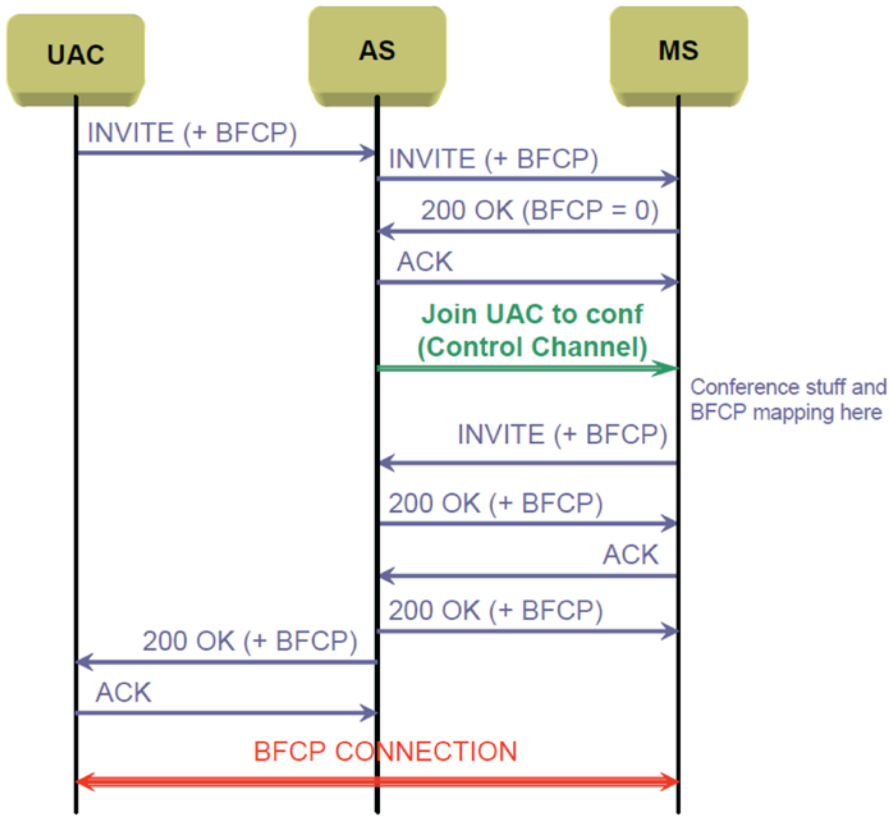


Figure 6. BFCP negotiation: transparent to the user



INVITE is then relayed by the AS to the UAC as part of a new 3PCC;

- Once this further negotiation is over, the UAC is able to open a BFCP connection with the floor control server in order to access its moderated resources.

This approach actually works (existing implementation efforts, e.g., our Confiance prototype [Buono, et al., 2007] seem to confirm this statement), and is how things would work anyway in any MS-generated re-INVITE involving BFCP. However, it is not transparent to the UAC, and so assumes that given functionality to be supported by the UAC, which can be seen, probably as a drawback, or at least as a limitation. Besides, the MS refusing the BFCP part of the first offer implies that the UAC first gets an answer with the BFCP media line set to 0, which is ambiguous: the UAC cannot know whether the media line has been set to 0 because the BFCP identifier is not available yet, or simply because BFCP is not supported by the AS/MS at all. A way to make the negotiation less ambiguous and transparent to the UAC is illustrated in Figure 6.

In this approach, the AS only provides the UAC with a complete answer when all of the needed identifiers are made available. Examining the signaling step by step as before:

- The UAC sends its INVITE to join the conference to the AS; the SDP offer again includes BFCP-related lines as specified in

Camarillo (2006); this INVITE is relayed to the MS as part of the 3PCC mechanism;
- The BFCP negotiation is refused as before;
- Unlike in the previous approach, the AS does not forward this answer to the UAC yet; since the UAC has just been attached to the MS, the AS can add the UAC to the conference mix and configure its BFCP settings through the control channel; from the UAC's perspective; however, the join attempt is still proceeding;
- At this point, the MS can trigger a re-INVITE addressed to the UAC, including all the BFCP-related identifiers as before; this re-INVITE is used by the AS to complete the negotiation with the UAC as part of the original 3PCC;
- At this point, the UAC is attached to the conference and is immediately able to open the BFCP connection.

This approach also works in theory, and has the advantage of being completely transparent to the UAC. The ambiguous BFCP media line set to 0 is still involved, but in this case it can be considered less of a problem than before: in fact, it can quite safely be assumed that the AS is aware of the functionality the MS can provide, including floor control. One of the drawbacks is instead the slowed down signaling from the UAC perspective, considering that the negotiation is completed only after a series of intermediate steps. Besides, further input from the UAC (e.g., a SIP CANCEL during the inner AS-MS interaction) might raise additional issues, which would need to be taken care of accordingly in the application logic.

CONCLUSION AND FUTURE WORK

Many services currently available over the Internet involve complex multimedia interactions among the interested parties. In the IMS, such services are implemented by the Application Servers, which terminate the signaling originated by User Agent Clients willing to access the functionality they provide. Typical examples of such Application Servers include conferencing systems, voice-mail services, and so on. This implicitly suggests that these Application Servers have somehow to be multimedia-aware. While there exist Application Servers, which offer multimedia support as an inner functionality, this is not usually the case in Next Generation Networks for several reasons, including a potential functional redundancy as well as lack of scalability.

In this chapter, we introduced the use of remote Media Servers (as fostered by both 3GPP and the IETF), with special focus on the framework architecture proposed by the MEDIACTRL Working Group. Starting from our implementation efforts, we provided the reader with a set of real world scenarios typically involving the complex interaction envisaged by such approach, focusing on the critical role of the Application Server. In all the presented scenarios, Application Server acts as a middlebox for all the signaling. For each scenario, a different 3PCC-based approach has been presented, with its strengths and drawbacks. We have explained how the right choice for the signaling is never simple, and is often dependent upon the specifics or policies of the scenario itself, as well as the capabilities of the involved parties. This is even truer when involving non-multimedia resources in the signaling, as the presented case of BFCP and its COMEDIA-based negotiation. Strong standardization efforts, especially in researching best common practices regarding the protocols of interaction, become of paramount importance in such scenarios, considering how the choice of a specific pattern of signaling instead of another can lead to weaker results. Future work will definitely include further study of the presented scenarios and patterns of interactions, as well as the introduction of additional scenarios to deal with.

REFERENCES

3GPP. (2007). *Internet protocol (IP) multimedia call control protocol based on session initiation protocol (SIP) and session description protocol (SDP): Stage 3*. Technical Report. Retrieved from http://www.3gpp.org

Boulton, C., Melanchuk, T., & McGlashan, S. (2011). *Media control channel framework*. RFC 6230. Retrieved from http://www.rfc-editor.org/rfc/rfc6230.txt

Buono, A., Castaldi, T., Miniero, L., & Romano, S. P. (2007). Design and implementation of an open source IMS enabled conferencing architecture. In *Proceedings of the 7th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN 2007)*. St. Petersburg, Russia: NEW2AN.

Camarillo, G. (2006). *Session description protocol (SDP) format for binary floor control protocol (BFCP) streams*. RFC 4583. Retrieved from http://tools.ietf.org/html/rfc4583

Camarillo, G., Ott, J., & Drage, K. (2006). *The binary floor control protocol (BFCP)*. RFC 4582. Retrieved from http://tools.ietf.org/html/rfc4582

Handley, M., Jacobson, V., & Perkins, C. (2006). *SDP: Session description protocol.* RFC 4566. Retrieved from http://tools.ietf.org/html/rfc4566

McGlashan, S., Melanchuk, T., & Boulton, C. (2011). *An interactive voice response (IVR) control package for the media control channel framework*. RFC 6231. Retrieved from http://tools.ietf.org/html/rfc6231

McGlashan, S., Melanchuk, T., & Boulton, C. (2012). *A mixer control package for the media control channel framework.* RFC 6505. Retrieved from http://www.rfc-editor.org/rfc/rfc6505.txt

MEDIACTRL. (2012). *IETF media server control prototype.* Retrieved from http://mediactrl.sourceforge.net

Rosenberg, J., Peterson, J., Schulzrinne, H., & Camarillo, G. (2004). *Best current practices for third party call control (3PCC) in the session initiation protocol (SIP)*. RFC 3725. Retrieved from http://tools.ietf.org/html/rfc3725

Rosenberg, J., Schulzrinne, H., Camarillo, G., et al. (2002). *SIP: Session initiation protocol*. RFC 3261. Retrieved from http://www.ietf.org/rfc/rfc3261.txt

ADDITIONAL READING

Ghandeharizadeh, S., & Muntz, R. (1998). Design and implementation of scalable continuous media servers. *Parallel Computing*, *24*(1), 91–122. doi:10.1016/S0167-8191(97)00118-X

Hasswa, A., & Hassanein H. (2012). Utilizing the IP multimedia subsystem to create an extensible service-oriented architecture. *Journal of Computational Science*.

Liao, J., Qi, Q., Xun, Z., Li, T., Cao, Y., & Wang, J. (2012). A linear chained approach for service invocation in IP multimedia subsystem. *Computers & Electrical Engineering*, *38*(4), 840–852. doi:10.1016/j.compeleceng.2012.03.010

Liao, J., Wang, J., Li, T., Wang, J., Wang, J., & Zhu, X. (2012). A distributed end-to-end overload control mechanism for networks of SIP servers. *Computer Networks*, *56*(12), 2847–2868. doi:10.1016/j.comnet.2012.04.024

Luo, A., Lin, C., Wang, K., Lei, L., & Liu, C. (2009). Quality of protection analysis and performance modeling in IP multimedia subsystem. *Computer Communications*, *32*(11), 1336–1345. doi:10.1016/j.comcom.2009.03.003

Pesch, D., Pous, M. I., & Foster, G. (2005). Performance evaluation of SIP-based multimedia services in UMTS. *Computer Networks*, *49*(3), 385–403. doi:10.1016/j.comnet.2005.05.013

KEY TERMS AND DEFINITIONS

Application Server (AS): It is a component in charge of appropriately controlling a Media Server (MS) in order to provide advanced services to end-users. As such, AS is where all the application logic related to the services resides. To make a very simple example, AS can be seen as the brain in the architecture, viz. as the entity making decisions and controlling all the actions accordingly.

Binary Floor Control Protocol (BFCP): It is an IETF designed protocol used to handle moderation of resources. The protocol envisages the so-called *floor* as a token that can be associated with one or more resources. Queues and policies associated with such floors are handled by a Floor Control Server (FCS), which acts as a centralized node for all requests coming from Floor Control Participants (FCP).

IP Multimedia Subsystem (IMS): It is a standardized Next Generation Networking (NGN) architecture for telecom operators that want to provide mobile and fixed multimedia services. IMS uses a Voice-over-IP (VoIP) implementation based on a 3GPP standardized implementation of SIP, and runs over the standard Internet Protocol (IP).

Media Server (MS): It is a component conceived to take care of every facet of the media processing and delivery. Its operations are realized according to the directives coming from the controlling Application Server. MS can be seen as the arm in the architecture.

Media Server Control (MEDIACTRL): It is a Working Group of the IETF which aims at specifying an architectural framework to properly cope with the separation of concerns between Application Servers and Media Servers in a standardized way.

Session Description Protocol (SDP): It is an IETF designed protocol intended for describing multimedia communication sessions for the purposes of session announcement, session invitation, and parameter negotiation. SDP does not deliver media itself, but is used for negotiation between end points of media type, format, and all associated properties. The set of SDP properties and parameters constitute a *session profile*. SDP is designed to be extensible to support new media types and formats.

Session Initiation Protocol: (**SIP**): It is an IETF defined signaling protocol widely used for controlling communication sessions such as voice and video calls over Internet Protocol (IP). The SIP protocol can be used for creating, modifying, and terminating two-party (unicast) or multiparty (multicast) sessions. Sessions may consist of one or several media streams.

APPENDIX A: THE IP MULTIMEDIA SUBSYSTEM

Figure 1 shows the architecture of the 3GPP IP Multimedia Subsystem. Both IMS entities and IMS interfaces are showed in the picture. In the following of this appendix, we briefly expand on the components, which came into play for the work described in this chapter. The *User Equipment* (UE) implements the role of a participant and might be located either in the Visited or in the Home Network (HN). In any case, it can find the P-CSCF via the CSCF discovery procedure. Once done with the discovery phase, the UE sends SIP requests to the *Proxy-Call Session Control Function* (P-CSCF). The P-CSCF in turn forwards such messages to the Serving-CSCF (S-CSCF). In order to properly handle any UE request, the S-CSCF needs both registration and session control procedures (so to use both subscriber and service data stored in the *Home Subscriber Server* – HSS). It also uses SIP to communicate with the Application Servers (AS). An AS is a SIP entity hosting and executing services. The *IP Multimedia Service Control* (ISC) interface sends and receives SIP messages between the S-CSCF and the AS. The two main procedures of the ISC are: (1) routing the initial SIP request to the AS; (2) initiating a SIP request from the AS on behalf of a user. For the initiating request the SIP AS and the OSA SCS (*Open Service Access – Service Capability Server*) need either to access user's data or to know a S-CSCF to rely upon for such task. As we already mentioned, such information is stored in the HSS, so the AS and the OSA SCS can communicate with it via the *Sh* interface. In the scenario described in this work, the MRFC (*Media Resource Function Control*) shall regard the MRFP (*Media Resource Function Processing*) as a mixer. When the MRFC needs to control media streams (creating a conference, handling or manipulating a floor, etc.) it uses the *Mp* interface. This interface is fully compliant with the H.248 protocol standard. S-CSCF communicates with MRFC via *Mr*, a SIP based interface.

APPENDIX B: SIP – SESSION INITIATION PROTOCOL

The *Session Initiation Protocol* (Rosenberg, et al., 2002) is an end-to-end, client-server protocol. The design base was HTTP (HyperText Transfer Protocol) and SMTP (Simple Mail Transfer Protocol), two lightweight text-based protocols. SIP was originally used to establish, modify, and terminate multimedia sessions over the Internet. It has evolved to be able to set up a broad range of sessions, like multimedia (e.g., voice, video, etc.), gaming, Instant Messaging, and presence. SIP messages are either *requests* or *responses*, and may carry zero or more *bodies*. The most common body carried by SIP messages is an SDP payload. It is noteworthy that the *Session Description Protocol* (SDP) (Handley, et al., 2006) is used to describe the set of media streams, codecs, and other media-related parameters supported by either party in a multimedia session. All SIP implementations must support SDP. SIP runs on any transport protocol (UDP, TCP, TLS, SCTP); the specifications mandates UDP and TCP, while other transport protocols are optional. SIP provides the following functionality:

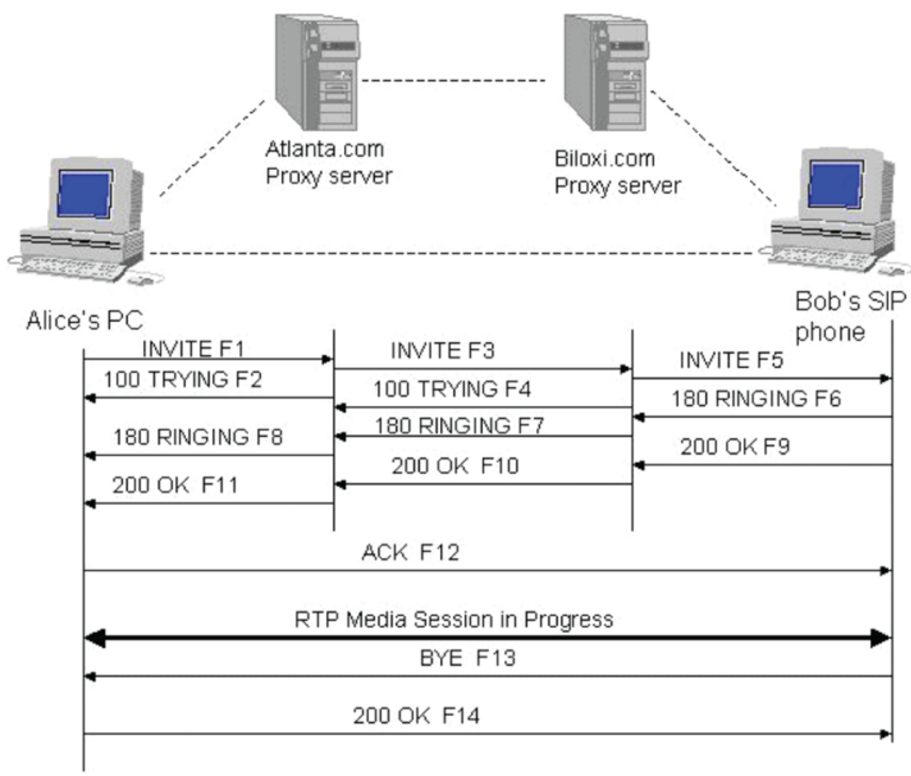
- User location.
- User availability.
- User capabilities.
- Session set up.
- Session management.

SIP does not provide services, but it enables the system to provide services in an easy way. The specifications envisage the following logical entities:

- **User Agent (UA):** An endpoint which can act by both a User Agent Client (UAS), when it sends requests and receives responses, and a User Agent Server (UAS), when it receives requests and sends responses;
- **Proxy Server:** A network host that proxies requests and responses. Hence, it acts as a UAC and a UAS;
- **Redirect Server:** A UAS that redirects requests to other servers;
- **Back-to-Back User Agent (B2BUA):** A UAS linked to a UAC. It acts as a UAS and as a UAC linked by some application logic;
- **Registrar:** A special UAS that accepts only registrations.

There are several types of SIP proxies, depending on the state they keep. A *stateless proxy* does not keep any state when forwarding requests and responses, while a *Transaction stateful proxy* stores state during the duration of the transaction. Finally, a *Call stateful proxy* stores all the state pertaining to a session (e.g., from the beginning to the end). SIP interactions usually happen by following the so-called *SIP trapezoid* depicted in Figure 7. We will not analyze the various SIP messages. The interested reader may refer to Rosenberg et al. (2002) for a detailed description.

Figure 7. The SIP trapezoid



APPENDIX C: BFCP – BINARY FLOOR CONTROL PROTOCOL

As the name already suggests, *floor control* is a way to handle moderation of resources. In fact, a floor can be seen, from a logical point of view, as the right to access and/or manipulate a specific set of resources that might be available to end-users. Introducing means to have participants request such a right is what is called floor control. A typical example is a lecture mode conference, in which interested participants might need to ask the lecturer for the right to talk in order to ask a question. The IETF standardized, within the context of the XCON WG, a dedicated protocol to deal with floor control, the *Binary Floor Control Protocol* (Camarillo, et al., 2006). This protocol envisages the above mentioned floor as a token that can be associated with one or more resources. Queues and policies associated with such floors are handled by a Floor Control Server (FCS), which acts as a centralized node for all requests coming from Floor Control Participants (FCP). Decisions upon incoming requests (e.g., accepting or denying requests for a floor) can be either taken on the basis of automated policies by the FCS itself, or relayed to a Floor Control Chair (FCC), in case one has been assigned to the related floor. These decisions affect the state of the queues associated with the related floors, and consequently the state of the resources themselves. To go back to the lecture mode scenario example presented before, a participant who has been granted the floor (i.e., the right to ask a question to the lecturer) would be added to the conference mix, whereas participants without the floor (or with pending requests) would be excluded from the same mix, thus being muted in the conference. An example of BFCP interaction is depicted in Figure 8.

Figure 8. BFCP in action