Validation of the Signaling Procedures of a Delivery Platform for IMS Services

Juan Miguel Espinosa Carlin and Dirk Thissen

Juan Miguel Espinosa Carlin
Communication and Distributed Systems, RWTH Aachen University, 52064 Aachen, Germany,
e-mail: `espinosa@i4.informatik.rwth-aachen.de`

Dirk Thissen
Communication and Distributed Systems, RWTH Aachen University, 52064 Aachen, Germany,
e-mail: `thissen@i4.informatik.rwth-aachen.de`

Abstract Defined by the 3^{rd} Generation Partnership Project (3GPP), the IP Multimedia Subsystem (IMS) is becoming the de facto overlay architecture for enabling service delivery in converged environments. In order to explore and to further develop the service delivery mechanisms defined by the IMS, there is the need to enable a reliable testing environment that allows to experiment with these new approaches, in order to correctly evaluate their impact in the standards already defined. With this goal in mind, this paper presents an evaluation of conformance of the session control mechanisms of an IMS testbed based on the UCT IMS Client developed at the University of Cape Town and the Open IMS Core implementation of the Fraunhofer FOKUS Institute at Berlin.

1 Introduction

Nowadays, users want to be able to access their services in a uniform way without regard of the type of subscription they have (e.g. fixed, mobile, data), the device / terminal that they are using for connecting to the network, or the network connectivity existing in their environments (e.g. WLAN, GPRS, UMTS) [10]. One of the main constraints imposed by this paradigm is the need to enable a unified service control architecture that allows the delivery of rich multimedia services that satisfy the expectations of the users. With this goal in mind, the 3^{rd} Generation Partnership Project (3GPP) introduced the IP Multimedia Subsystem (IMS) as a service control architecture aimed to realize network and service convergence.

Because the IMS is already in trial phases with operators adopting the standard worldwide, research efforts in the field are likely to gain support within a broad audience. In order to enable the development of IMS services and of trial of concepts around core the IMS elements, it is necessary to have a testing environment that allows evaluating the impact of the proposed approaches to the 3GPP standards. The aim of this paper is to evaluate the basic session signaling mechanisms of an

IMS testbed based on the UCT IMS Client [12] and on the Open IMS Core Project [2].

The rest of this paper is structured as follows. Section 2 gives an overview of the IMS architecture and of the IMS service delivery mechanisms. Then, Sect. 3 describes the hardware and software environment in which the tests were applied. Section 4 presents and analyzes the SIP dialogs traced in the testbed for basic registration, session set up, presence updating and messaging mechanisms. Finally, the conclusions and pointers towards future work are given on Sect. 5.

2 The IP Multimedia Subsystem

The IMS is based on protocols developed by the Internet Engineering Task Force (IETF). The basic ones are the Session Initiation Protocol (SIP) for session control and signaling, and the Diameter protocol for doing Authentication, Authorization, and Accounting (AAA). For an IMS core interconnecting only IP-based networks, the simplified architecture of the system is depicted in Fig. 1.

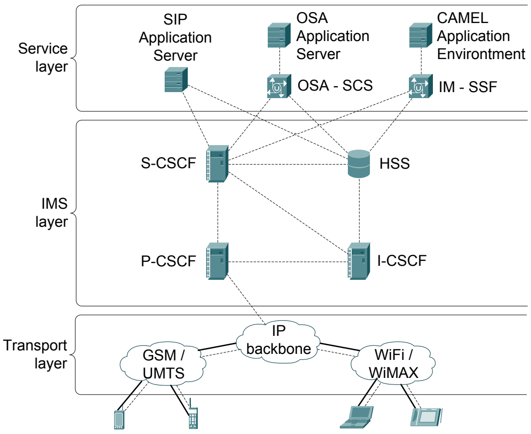

Fig. 1 Simplified IMS Architecture.

As shown, the lowest transport layer groups all the IP-based networks that allow users to access their services via the main IP backbone. For each one of these access networks, a specific technology is defined, and the device / terminal used to interact with the service must be compatible with these definitions (e.g. GSM, UMTS, WiFi, WiMAX).

Next, the IMS layer includes all the functions that implement the logic for enabling AAA and for allowing service delivery in the system. The main nodes in this level are:

- One or more SIP-based servers, collectively called Call / Session Control Functions (CSCF's), which are the essential nodes in charge of processing all the SIP signaling in the IMS. Depending on its role, a CSCF can be categorized as a Proxy-CSCF (P-CSCF) when it is the first point of contact between the IMS terminal and the network, as an Interrogating-CSCF (I-CSCF) when it is a proxy located at the edge of an administrative domain, or as a Serving-CSCF (S-CSCF) when it performs session control and provides routing services.
- One or more Home Subscriber Servers (HSS's), which are the repositories for the related subscriber information required to handle multimedia sessions.

Finally, the service layer contains the AS's that implement the services that will be delivered to the end users.

3 Evaluation Environment

This section gives an overview of the software components used for the testbed.

3.1 Open IMS Core Project

The Open Source IMS Core Project [2] is an implementation of the CSCF's and a lightweight HSS, which together form the core elements of all IMS architecture as specified by the 3GPP. The four components are all based on Open Source software. The central entities are the Open IMS CSCF's (Proxy, Interrogating, and Serving) which were developed as extensions to the SIP Express Router (SER) [7]. Since even basic signaling routing functionality for IMS requires information look-up in the HSS, the FOKUS Home Subscriber Server (FHoSS) is also part of the Open Source IMS Core project.

The testbed was configured with the default parameters for listening SIP requests: `4060` for the P-CSCF, `5060` for the I-CSCF and `6060` for the S-CSCF. Additionally, a DNS server was set up to properly resolve the default domain on which the testbed is running: `open-ims.test`. For the case of the HSS, the two IMS subscribers included by default (Alice and Bob) were used for the tests. For both of them, the proper service profiles were defined and a default iFC was configured with the proper service triggers for forwarding the SIP requests when doing the tests with the presence server.

3.2 UCT IMS Client

The *UCT IMC Client* [12] is developed by the Communications Research Group at the University of Cape Town, South Africa. The client was used with the default configuration for both Alice and Bob. For the case of Alice, the IMS configuration parameters are the following:

- Public User Identity: `sip:alice@open-ims.test`
- Private User Identity: `alice@open-ims.test`
- Proxy CSCF: `sip:pcscf.open-ims.test:4060`
- Realm: `open-ims.test`
- QoS: `Mandatory`
- QoS Type: `Segmented`
- Access Network: `IEEE-802.11b`

3.3 OpenSER

OpenSER [8] is an open source SIP server that can act as SIP registrar, proxy or redirect server. For the aim of the tests, the OpenSER was configured as a presence server. The necessary configuration changes were done in order for the OpenSER to support the necessary functions to correctly process the `PUBLISH` and `SUBSCRIBE` SIP requests, and to enable the persistent storage of subscriptions information in a database. The server was configure to listen requests at `localhost:5065`.

3.4 Testing Scenario

The architecture of the networking scenario is shown in Fig. 2.

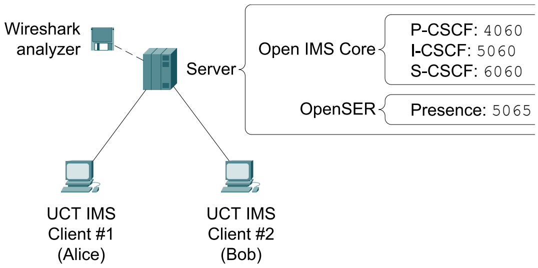

Fig. 2 Networking scenario deployed out for testing

To test the conformance of the testbed regarding the IMS SIP session control mechanisms, it is necessary to have a close look to the SIP flows that exchanged between the parties involved in the signaling. The tool used for this purpose, was Wireshark [11], a known protocol analyzer that features live capture and offline analysis of network streams.

4 Conformance Results

4.1 IMS Registration with Presence Service enabled

The *IMS Registration* procedure is defined in 3GPP TS 24.229 [4], and it is the first step that has to be done by subscribers in order for them to be able to access their services. The following assumptions were done during this test:

- Alice is an authorized subscriber in the network (i.e. a valid HSS profile exists).
- Neither Alice nor Bob are registered in the network.
- Alice is doing the Registration from the home network.
- Alice has her *Presence Service* enabled and has Bob in her buddy list.

The SIP sequence traced for Alice (or Bob) when registering in the testbed is shown in Fig. 3. This flow shows the SIP messages exchanged when Alice registers with the testbed. First, the UCT IMS Client #1 sends the proper `REGISTER` (1) request to the contact point with its IMS network, which as configured, is the P-CSCF running in the server at port `4060`. Then, the P-CSCF forwards the request to the I-CSCF (2), which performs the first authorization step and determines if a S-CSCF is already assigned to Alice. This authorization is achieved by exchanging the proper Diameter messages with the HSS (for simplicity, not shown in Fig. 3). Because dealing with a new registration, no S-CSCF has been yet assigned to Alice, so the I-CSCF performs a S-CSCF selection procedure according to the information received by the HSS, and proxies the request to the selected P-CSCF (3). In order to authenticate Alice, the P-CSCF contacts the HSS for downloading her authentication vectors. Additionally, it registers itself in the HSS, so that further queries to the HSS related to Alice are properly routed back to the S-CSCF. Then, the S-CSCF sends back a `401 Unauthorized` (4) SIP message that includes a challenge that must be correctly answered. This request is received by the I-CSCF which then forwards it to the P-CSCF (5). Finally, the P-CSCF sends the request Alice's IMS device (6).

After receiving the challenge, the Alice answers and its IMS device sends another `REGISTER` (7) with this information. After receiving it, the P-CSCF forwards it to the I-CSCF (8), which then sends it to the S-CSCF (9). It is important to mention that before forwarding the message, the I-CSCF queries the HSS to find out which S-CSCF has been assigned to Alice when the first `REGISTER` was sent (these Diameter messages are not shown in Fig. 3). After receiving the request, the S-CSCF verifies the answer given by Alice to the challenge and, if successful, sends back an `200 OK` (10) to confirm the registration. This request is forwarded by the I-CSCF (11) and finally by the P-CSCF (12).

An IMS requirement inherited from the GSM age, consists on letting Alice know whether or not she is reachable. Because the core SIP specification offers not solution to cope with this problem, the IETF created a registration package for the SIP event framework [9], which allows the IMS terminal to subscribe to its own registration information stored in the S-CSCF. After receiving the confirmation of

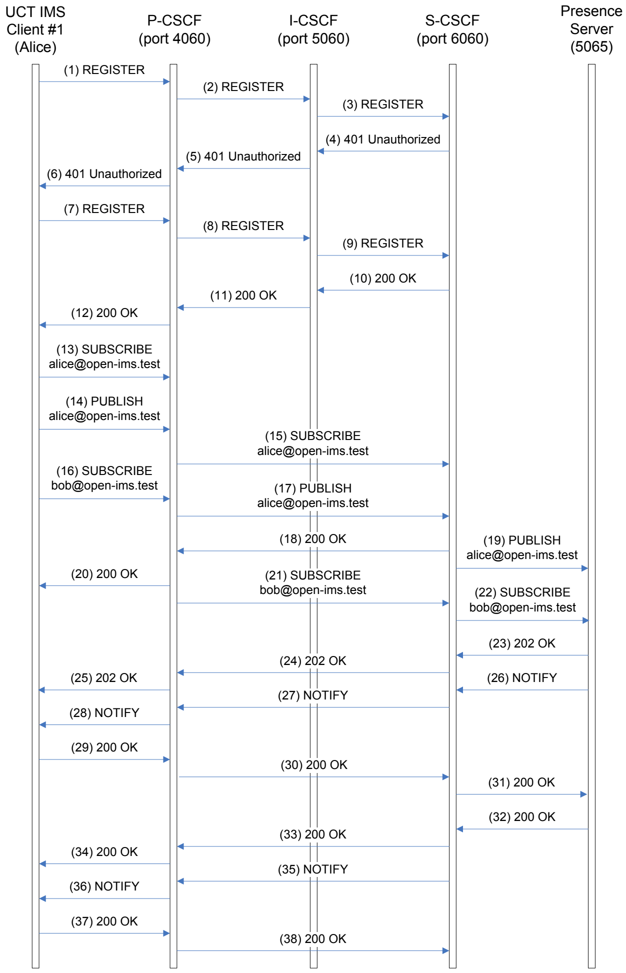

Fig. 3 SIP flow of an IMS Registration

the registration procedure, Alice's IMS terminal sends a `SUBSCRIBE` request for the `reg` event (13) addressed to the same PUI that the S-CSCF just registered. This request is directly forwarded from the P-CSCF to the S-CSCF (15), and upon reception, the S-CSCF installs the subscription and positively replies to user with the corresponding `200 OK` via the P-CSCF (18 and 20). Additionally, the S-CSCF sends a `NOTIFY` request (35) that contains the registration information. The request is forwarded to Alice (36) and answered by her client with the corresponding `200 OK` (37 and 38).

Because Alice has presence enabled on her device, two additional signaling sequences are traced. The first one consists on Alice publishing her information in the Presence Server. This is achieved by Alice's client sending a `PUBLISH` (14) request, which is then forwarded by the P-CSCF (17) and by the S-CSCF (19). When receiving the request, the Presence Server updates Alice's information and confirms her with the correspondent `OK 200` (32), which is sent back by the CSCF's (33 and 34).

The second sequence is the one dealing with subscribing to Bob's presence information. For achieving this goal, Alice's device sends a `SUBSCRIBE` (16) to the Presence Server to indicate that Alice wants to be notified about Bob's presence information. This `SUBSCRIBE` request is forwarded to the Presence Server as usual (16, 21 and 22). When the Presence Server receives this subscription it checks if Bob has already published some information. Then, the Presence Server sends the corresponding `202 OK` back to Alice (23, 24 and 25) for confirming the subscription. Additionally, the Presence Server sends Alice the corresponding `NOTIFY` related to Bob's installed subscription (26, 27 and 28). Upon reception of this notification, Alice's client sends back the corresponding `200 OK` (29, 30 and 31) to the Presence Server.

As it can be seen, based on the assumptions already given, the Registration procedure executed by the testbed complies with the rules defined in the correspondent standards of the 3GPP, and satisfies the constraint of executing registration in two round trips.

4.2 Session Setup with Mandatory QoS Requirements

The procedure that is usually followed by users after successfully registering is to establish sessions with other users in the IMS network. The operations done by each node in the IMS network for the aim of setting up a session are also described in 3GPP TS 24.229 [4]. This test consisted on Alice inviting Bob to take part in a voice call. The following assumptions were done:

- Alice and Bob have successfully registered in the testbed.
- The client configuration of both Alice and Bob is the same.
- The supported video and audio codecs are the same for both IMS clients.

The SIP sequence traced in the testbed from the point in which Alice invites Bob to the point in which Bob's client starts the mechanism to negotiate QoS of the session, is shown in Fig. 4.

The procedure starts with Alice's IMS device sending a `INVITE` (1) request directed to Bob. This request includes a SDP body describing the type of session that Alice's client wants to create. After receiving the request, the P-CSCF replies Alice with a `100 Trying` message (2), confirming that the session set up was started. Then, the `INVITE` request is forwarded by the P-CSCF to the S-CSCF (3), which replies the P-CSCF also with a `100 Trying` (4). Because Bob is registered at the same P-CSCF as Alice, the request is send by the S-CSCF to itself (5), and is answered with the proper `100 Trying` (6). The `INVITE` is further forwarded by the S-CSCF to Bob (7) via the P-CSCF (9). Each `INVITE` is answered with its

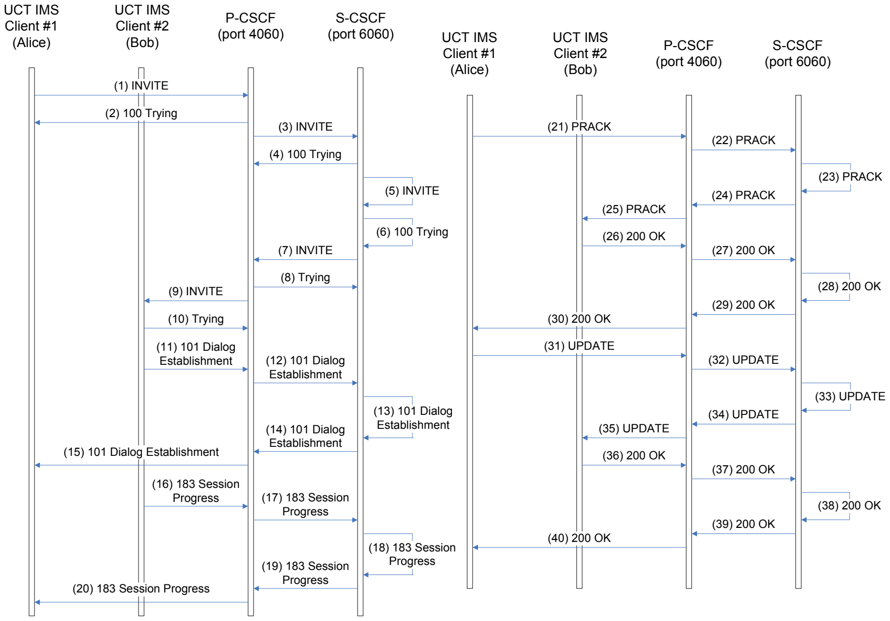

Fig. 4 Basic session setup, part 1

correspondent 100 Trying (8 and 10). Then, Bob's client sends a 101 Dialog Establishment message to inform about the set up status. This message is first sent to the P-CSCF (11) and then forwarded to the S-CSCF (12). As already mentioned, because both Alice and Bob are being served by the same S-CSCF, the 101 Dialog Establishment is sent by the S-CSCF to itself (13) and is further forwarded to the P-CSCS (14) and finally to Alice's IMS device (15).

Because the session requires QoS negotiation as described in [1], Bob's client sends back an additional 183 Session Progress provisional response. This message also includes a SDP [6] body, which contains the media streams and codes that Bob is able to accept for this session. At this point, Bob's client can start its resource reservation, because it knows the parameters needed for it. As in the case of the 101 Dialog Establishment message, the 183 Session Progress response is routed back to Alice via the same route (16, 17, 18, 19 and 20). The provisional response is received by Alice and, assuming that both clients agreed on the codes used, Alice sends a PRACK request including the definitive SDP configuration for the session. In parallel with the generation of this request, Alice's client starts at this point the mechanisms for resource reservation. The PRACK request is sent to Bob traversing the required proxies (21, 22, 23, 24 and 25). When the request is received by Bob, his client generates the corresponding 200 OK for confirming the media streams and codecs that will be used for the session. At this time, Bob's terminal may still be involved in its resource reservation process, and most likely it will not yet be complete. Additionally, the 200 OK response indicates Alice's client that Bob's client wants to receive an indication when Alice is ready with her resource reservation. The `200 OK` is then sent back to Alice (26, 27, 28, 29 and 30).

Once that Alice's terminal has finished with its resource reservation, it sends an `UPDATE` request to inform Bob's client that the procedures on its side are finished. This is indicated through a specific SDP body added to the request. The request is routed to Bob via the same proxies as before (31, 32, 33, 34 and 35). Upon reception, Bob's client generates the corresponding `200 OK` and, assuming that it has already finished with the resource reservation procedures on its side, it send the response back to Alice (36, 37, 38, 39 and 40).

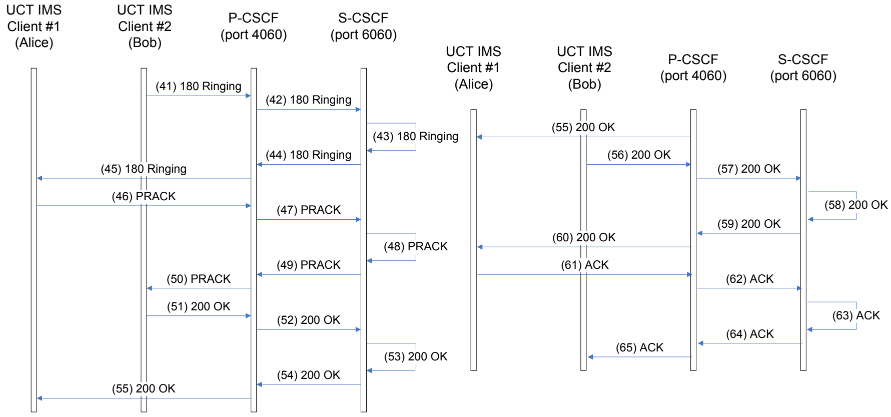

Fig. 5 Basic session setup, part 2

The SIP flow continues as shown in Fig. 5. The next step consists on Bob's IMS device alerting him about the incoming request. This action is indicated to Alice by sending her a `180 Ringing` message (41) with a SDP body for indicating Alice about the characteristics accepted by Bob's client. This notification is forwarded by the P-CSCF (42), twice by the S-CSCF (43 and 44), and by the P-CSCF (45) to Alice. After receiving this message, Alice's client will likely generate a locally stored tone to indicate Alice that Bob's terminal is ringing. Additionally, Alice's client sends back a `PRACK` message (46), which is routed back to Bob (47, 48, 49 and 50). When received, Bob's device sends back a `200 OK` confirmation. This message is routed back to Alice following the usual SIP signaling path (51, 52, 53, 54 and 55). After accepting the call, Bob's device sends a `200 OK` message (56) that is forwarded to Alice by the P-CSCF (57), twice by the S-CSCF (58 and 59), once more by the P-CSCF (60). Finally, Alice's client generates the last `ACK` to confirm that the generation of media-place traffic with the agreed codecs can be started. This request travels back to Bob through the usual path (61, 62, 63, 64 and 65). At this point the session setup is complete, and both Alice an Bob can generate their respective audio and video media streams.

As described above, the session set up process done by the testbed, subject to the mentioned restrictions, complies with the procedures described by the 3GPP when dealing with establishing a session with specific QoS requirements.

4.3 Presence Service

The architecture for supporting the *Presence Service* in the IMS is defined in 3GPP TS 23.141 [3]. As in the Internet, the Presence Service in the IMS can be considered as the cornerstone for service provisioning, due to the fact that most of the other services benefit from the presence information supplied. The following assumptions were made regarding the presence tests:

- Alice and Bob have the Presence Service enabled in their devices.
- Alice and Bob have each other in their Presence Buddy Lists.
- Alice and Bob have successfully registered in the testbed, including the necessary mechanisms to involve the presence server by sending the proper `SUBSCRIBE` and `NOTIFY` messages.

Under these constraints, the SIP flow traced in the testbed when Alice is updating her presence information, in shown in Fig. 6.

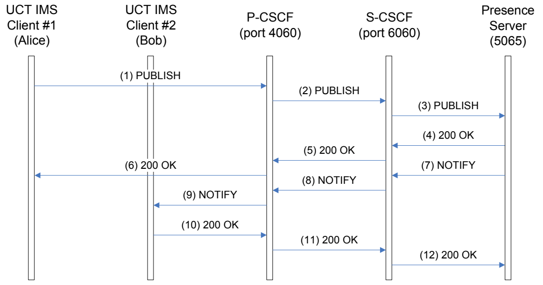

Fig. 6 Publishing presence information

In order to notify the system about a change in her presence status, Alice's client sends a `PUBLISH` request (1) to the PUI assigned to her during the registration phase. The request is received by the P-CSCF and then forwarded to the S-CSCF (2). Once received, the S-CSCF forwards the message to the Presence Server (3). The Presence Server sends a `200 OK` (4) back to Alice to inform her that the request arrived. This message is further forwarded by the S-CSCF (5), and finally sent to Bob (6). Additionally, the Presence Server determines which users have subscribed to changes in the presence status of Alice. Because Bob is included in this list, the Presence Server send the proper `NOTIFY` (7) to him, routing the request through the S-CSCF (8) and the P-CSCF (9) to finally deliver it to Bob (9). To acknowledge the receipt of the notification, Bob's client sends a final `200 OK`, that is routed back to the Presence Server (10, 11 and 12).

From these results, it can be concluded that the basic presence management mechanisms implemented by the testbed comply with those described by the 3GPP.

4.4 Messaging

The last procedure evaluated in the testbed was the *Messaging* service. It is defined in 3GPP TS 24.247 [5], and it allows a user to send some content to another user in almost real time. The basic content in an instant message is text, although it can be any other kind of media. For the aim of the results presented in this paper, only text-based messages were taken into account. This test assumes the following facts:

- Alice and Bob have successfully registered in the testbed.
- The client configuration of both Alice and Bob is the same.
- Alice and Bob authorized each other to use the messaging service.

The SIP flow found in the testbed when Alice and Bob exchange one instant message each, is depicted in figure 7.

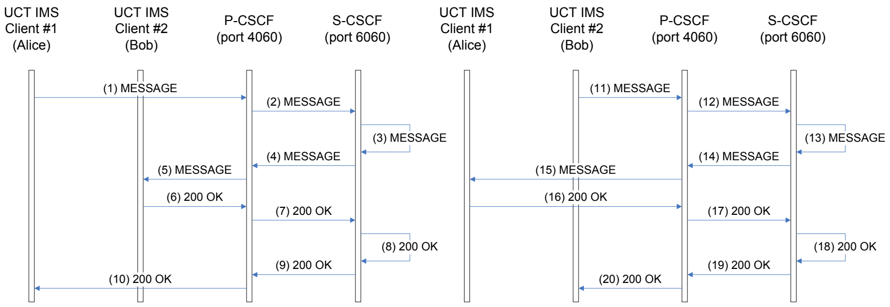

Fig. 7 Instant Messaging

As shown, the process is really simple, and the first step consists on Alice's client sending a `MESSAGE` request to Bob. This request is routed through the P-CSCF (2), twice through the S-CSCF (3 and 4), and finally sent to Bob (6). When Bob's device receives the request, it notifies Bob about the message sends and a `200 OK` reply. Finally, when Bob answers the message, the same procedure is done but in the opposite direction. The `MESSAGE` is sent by Bob (11, 12, 13, 14 and 15) and the correspondent reply is sent by Alice (16, 17, 18, 19 and 20) upon reception.

5 Conclusions and Future Work

This paper presented a conformance evaluation of the SIP signaling mechanisms realized by an IMS testbed based on well known software components. The tests were performed for the registration procedure, the initialization of a session between two registered users, the updating of presence information, and the sending of instant messages. It was shown that a testbed set up with the chosen software tools complies with the procedures developed by the 3GPP for the IMS regarding session signaling. Future work in the field includes further evaluation of the testbed for complex roaming scenarios and for scenarios involving the invocation of multiple services.

References

1. Camarillo, G., Marshall, W., Rosenberg, J.: Integration of Resource Management and Session Initiation Protocol (SIP). RFC 3312 (Proposed Standard) (2002). URL http://www.ietf.org/rfc/rfc3312.txt. Updated by RFCs 4032, 5027
2. Fraunhofer FOKUS: OpenIMSCore.org — The Open IMS Core Project. http://www.openimscore.org (2007). Last retrieved on the 20.04.2008
3. 3rd Generation Partnership Project: Presence service; Architecture and functional description; Stage 2. 3GPP TS 23.141 (2006)
4. 3rd Generation Partnership Project: Internet Protocol (IP) multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3. 3GPP TS 24.229 (2007)
5. 3rd Generation Partnership Project: Messaging service using the IP Multimedia (IM) Core Network (CN) subsystem; Stage 3. 3GPP TS 24.247 (2007)
6. Handley, M., Jacobson, V.: SDP: Session Description Protocol. RFC 2327 (Proposed Standard) (1998). URL http://www.ietf.org/rfc/rfc2327.txt. Updated by RFC 3266
7. iptel.org: SIP Express Router. http://www.iptel.org/ser (2006). Last retrieved on the 15.11.2007
8. OpenSER.org: OpenSER - the Open Source SIP Server. http://www.openser.org (2007). Last retrieved on the 20.04.2008
9. Rosenberg, J.: A Session Initiation Protocol (SIP) Event Package for Registrations. RFC 3680 (Proposed Standard) (2004). URL http://www.ietf.org/rfc/rfc3680.txt
10. Schonhowd, R.: Telecom And IT Strategists Must Pursue Converged Service Delivery To 2010 And Beyond. Forrester Research Report. Available from http://www.forrester.com/Research/Document/Excerpt/0,7211,41543,00.html (2007). Last retrieved on the 25.03.2008
11. The Wireshark Team: Wireshark. http://www.wireshark.org (2007). Last retrieved on the 15.11.2007
12. University of Cape Town: UCT IMS Client. http://uctimsclient.berlios.de (2007). Last retrieved on the 15.11.2007