REMOTE SERVICE USAGE THROUGH SIP WITH MULTIMEDIA ACCESS AS A USE CASE

Andreas Häber
Agder University College
Grimstad, Norway

Martin Gerdes
Ericsson Research
Aachen, Germany

Frank Reichert
Agder University College
Grimstad, Norway

Andreas Fasbender
Ericsson Research
Aachen, Germany

Ram Kumar
Agder University College
Grimstad, Norway

Abstract—**The IP Multimedia Subsystem is under deployment, as an IP-based service control and access infrastructure, but how it interconnects with residential appliances is currently unclear. With IMS access for the residential appliances they can be used as both service consumers and service providers. In this paper we present a protocol which allows residential services to be remotely invoked, through the IMS, and consumed in a different network, along with a prototype implementation and early results. With our protocol services of two distinct service protocol systems can cooperate.**

Index Terms—**IP Multimedia Subsystem, multimedia communication, telecommunication services, service islands, wireless LAN, multimedia systems**

I. INTRODUCTION

ERICSSON and Agder University College are looking into residential services in fixed-mobile converged networks. The IP Multimedia Subsystem (IMS) [1] is under deployment as an IP-based service control and access infrastructure. Devices, mobile or fixed, can register to central identification and access control nodes to obtain access to IP based services provided by the IMS infrastructure in a secure manner. IMS offers services through mobile or fixed Wide Area Networks (WAN) supporting Quality of Service (QoS).

In this paper, we start by presenting example use cases for providing and using remote services in Section II. Subsequently, in Section III, a protocol for enabling these use cases is proposed, based on the Session Initiation Protocol (SIP) [2]. We describe a prototype implementation of this protocol in Section IV, along with early results of the prototype in Section V.

In Section V an overview of related work is given, and finally, in Section VII, conclusions from our current results are stated, and we describe future work for both the protocol and the prototype.

II. USE CASES

What if the IMS connected device is in a house with access to a local network? In the local network, many services and resources are available, such as a TV, game console, home automation, etc. To enable a better user experience it is important that they can interconnect with the IMS. The device uses service discovery protocols, such as Service Location Protocol (SLP) [3], Universal Plug & Play (UPnP) [4], Bluetooth, etc, to be aware of the resources in the local network, as shown in Figure 1.

A. IMS Service consumption in a local network

Video service usage, where the consumer routes the content stream to a TV instead of its mobile phone, is an example of using IMS services together with the home network's resources. Consumers can preview movies, free of charge, on a mobile terminal and can pay to receive the content stream in a quality suitable to be viewed full screen on a TV screen. As the TV/Set-top Box (STB) typically will not host an IMS client, a gateway between IMS and the home network is required, referred to as Home IMS Gateway (HIGA) in Figure 1.

If the content stream is available in low quality only, for instance mobile originated video call, the rest of the display area can be used to show data related to the video call. In business scenarios, this can be spreadsheets, presentations, etc, while in family scenarios this can be a picture slideshow, browsing websites together for travel planning, etc.

B. Providing services from the local network

Home networks also provides services, such as access to Heat, Ventilation and Air Conditioning (HVAC) appliances, multimedia, and digital security cameras. With the expected pervasive deployment of IMS, users can therefore access their services and resources from everywhere: at the office, while visiting friends, when traveling, etc.
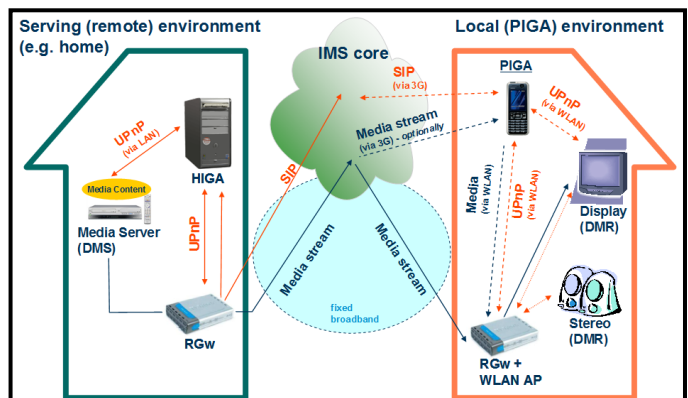

Figure 1 Remote media access use case.

This allows house owners on tour to not only receive a notification from sensors, which is available today with the second 2G Short Messaging Service (SMS), but also to consume the video stream of the digital security camera by taking advantage of the security and QoS provided by the IMS infrastructure. The identification support of IMS helps devices to authenticate and connect to the home network, independent of current location and access network.

C. Remote service cooperation

So far, we have considered a service in the local network to either consume or provide services. However, there are also scenarios where users would like services of two local networks to cooperate. Access to multimedia content stored in the home network is one scenario. The user has a home network that uses any type of service discovery mechanism (e.g., UPnP or SLP), including a media server where the family's multimedia files are stored: pictures, high definition videos, music, etc.

However, when family members leave their home they cannot access the services in their home network, because many service discovery mechanisms, such as UPnP, rely on all services being part of the same local network. This is good for security, because it makes it harder for intruders to access the services.

For example, a family member stays at a hotel while traveling. It is now common to expect a TV in hotel rooms today, and some hotels provide broadband access with Wireless Local Area Network (WLAN) for their guests. The hotel room network also supports a service discovery mechanism, and it must cooperate with the guest's home network to allow it to, for example, consume multimedia, as shown in Figure 1. IMS makes it easy to locate and access the guest's home network in a secure manner, and ensures quality for the content stream.

D. A gateway function for remote service invocation

As mentioned above, the service discovery mechanisms inside local networks cannot interoperate between backbone interconnections. For example, UPnP uses multicasting for service discovery but the multicast communication is restricted to the local network, and the service addresses should be from the private address space [5]. Moreover, IMS is not aware of the services provided within the local networks. Therefore, gateway functionality between the local networks and the wide area interconnection is necessary to enable the scenarios outlined above.

III. REMOTE SERVICE INVOCATION PROTOCOL

The remote service invocation protocol, which we have named PIRANHA, is based on SIP. It works between two Service Discovery Gateways (SDGs) connecting independent service discovery mechanisms together, through a common backbone. In the media plane, service commands and corresponding content are sent to the services controlled by the
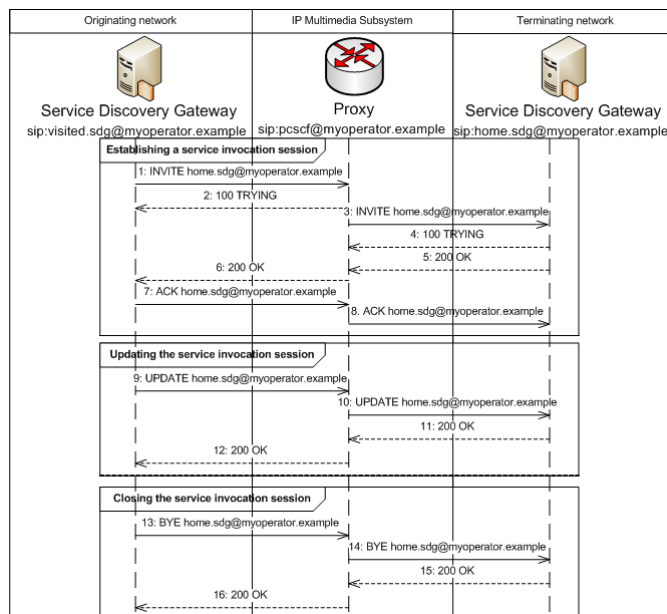

Figure 2 Message sequence at the signal plane for a remote service invocation session.

terminating SDG. This is different to other solutions, mentioned in Section VI, which use the session plane for service requests.

For remote connectivity scenarios, such as in the "hotel room to home network" scenario described in Section II.C, IMS can provide support for the necessary addressing, access control, and QoS. In other scenarios, for instance translation between two service discovery mechanisms in the same Local Area Network (LAN), intermediate nodes between the SDGs may not be necessary. Two SDGs establish service invocation sessions to be able to pass service invocation requests between them. In the following, we refer to the two SDGs as the originating SDG and the terminating SDG, where the originating SDG initiate the session and can pass requests to the terminating SDG.

There are three stages of the protocol: session establishment (Section III.A), update a session (Section III.B), and finally close a session (Section III.C). Command requests can be sent from the originating SDG to the terminating SDG, through the media plane, after a session has been established.

A. Establish service invocation sessions

Initiating a session is accomplished by sending an INVITE-

```
1   v=0
2   o=visited.sdg 3380446179 3380446179 IN IP4
192.168.168.31
3   s=-
4   c=IN IP4 /192.168.168.31
5   t=0 0
6   a=recvonly
7   m=application 9 TCP piranha
8   a=udn:uuid:9afb3231-345a-4cd1-b448-
8866b79ff91b
9
10  a=setup:active
```

Figure 3 SDP offer for remote service invocation session.

request with a Session Description Protocol (SDP) body [6], which describes the session, using the offer/answer model [7]. Figure 2 shows the complete message sequence.

In the INVITE-request, the originating SDG requests a device that it would like to control by specifying the udn-attribute in the SDP-offer, similar to the one shown in Figure 3. The session description part (lines 1 to 6) follows [7], therefore we don't describe it here. There is one media description included, starting at line 7, which specifies that the media type is *application* and the p*iranha* format. The terminating side can use this to see if it understands this protocol or not. If the terminating side does not understand application/piranha then it must return an appropriate response code, such as "488 Not Acceptable Here", and the session establishment will be cancelled. The media description line also specifies that we want to use Transmission Control Protocol (TCP) [8] as the transport protocol on port 9, the discard port, following [9], for service invocation requests. Three media-level value-attributes are also included: *udn (*from Unique Device Name (UDN)) is the device the SDG requests to control, and is a new attribute. The other two attributes specifies usage of the TCP port, and are specified in [9]. *Setup:active* means that the offering SDG will initiate an outgoing connection, and the *connection:new* attribute indicates that this should be a new connection. During the service invocation session this value–attribute can be changed to *existing* to indicate that there is an existing connection that should be reused.

If the terminating SDG accepts the INVITE-request, including its offer, it will add a port binding for this service invocation session at an arbitrary port at the local network's gateway, for example using UPnP. The SDP-answer, from the terminating SDG, looks similar to the offer, except that it specifies the destination port for the TCP-connection, and the setup and connection attributes specifies that it will listen for connections (setup:passive) and use a new connection (connection:new). The originating SDG will then receive a 200 OK response.

Now the originating SDG will check the answer in the response, and, if it is acceptable, it will send an ACK-request to the terminating SDG, via the proxy, to confirm it. Then it can start to use the media connection to invoke services. However, if the answer is unacceptable the originating SDG must terminate the session by sending a CANCEL-request instead, or, if supported, it can alter the session by sending an UPDATE-request.

Because the terminating SDG makes a Network Address Translation (NAT) [10, 11] mapping between the service's private internet address and the public internet address, it does not need to handle any of the traffic at the media plane. This makes the design and operation simple, but in some cases too simple because information that the service sends may include, for example, address information specific to its private network, causing problems when used in the originating network. An example of this issue is the "resource-uri"
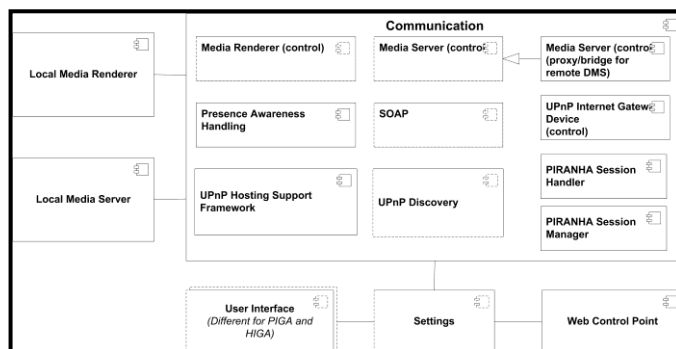

Figure 4 Logical view of xIGA Library, with user interface component for HIGA and PIGA. Dashed lines indicate components from our "Mobile UPnP Control Point" prototype.

included in browse-results from UPnP Media Servers, which we discuss below.

B. Updating service invocation sessions

During the session it might be necessary to alter the session description, for example to add or remove a media description for a media stream. To update the session an SDG can send a new INVITE-request for the same dialog, or, if supported, an UPDATE-request [12], including a new session description documents with the updated session information. Sequence 9-12 in Figure 2 shows this.

Being able to update the session is especially important when accessing IMS through cellular networks, where resources are scarce and it may be expensive for the customer to occupy resources.

C. Closing the service invocation session

Both SDGs can close the session, but normal behavior is that the originating SDG closes it. In exceptional situations, for example if the service becomes unavailable, the terminating side should close it instead.

Sending a BYE-request to the other SDG will close the session, as shown in sequence 13-16 in Figure 2.

All resources associated with the session must be released when the session is closed, including any port mappings setup during the course of events of the session.

IV. PROTOTYPE FOR REMOTE SERVICE INVOCATION

We have implemented a prototype of the establishment phase (Section III.A) by modifying our "Mobile UPnP Control Point" [13]. First, we had to port that prototype from a platform based on Java 2, Micro Edition (J2ME) Connected Limited Device Configuration (CLDC) to J2ME Connected Device Configuration (CDC).

In this section, we first describe why we changed our platform to J2ME CDC, and then we present the architecture of our new prototype.

A. Motivation for porting to J2ME CDC

As described in [4] the multicasting support in J2ME CLDC

is insufficient for UPnP, which was our main reason for switching to J2ME CDC as our platform.

In addition, we found that the reflection support in J2ME CLDC was too limited and made our API design for hosting UPnP devices awkward to use. In retrospect, our design for device hosting might be too powerful for this constrained platform. Our device hosting support allows any device, which uses our APIs, to be registered at runtime, which is why we require reflection. By altering the design, so devices are rather registered statically, we do not depend on so much reflection usage at runtime.

B. Scenario description

The scenario we consider here is users away from their home network, but the users still would like to access their multimedia. Examples of such situations are when visiting friends or staying at a hotel.

Even though mobile devices' storage space grows fast, is it hard to imagine that it will outgrow the pace of storage space required for a family's multimedia collection (family pictures, movies in high-definition format, and lossless music). Therefore, an access mechanism to a remote multimedia server is needed.

C. Deployment of PIRANHA Prototype

In our prototype system, we have three networks: home, visited, and an operator network connecting them together, as depicted in Figure 5. There is a gateway-device at the edge of both the home and visited networks, which implements the UPnP Internet Gateway Device (IDG) Device Configuration Protocol (DCP).

In the home network, the Digital Media Server (DMS) is located with the family's multimedia, whereas the Digital Media Player (DMP) is in the visited network.

D. Logical view of the PIRANHA Prototype Architecture

The prototype software is deployed at two nodes, HIGA[14] and Portable IMS Gateway (PIGA), which share a common library, named xIGA, for UPnP and the remote access functionality. This library is an evolution of the prototype described in [4], and the design is depicted in Figure 4. We will focus on describing the functionality relevant to the protocol described in Section III here.

1) PIRANHA Session Manager: This component implements the client part of the protocol. For instance, a user enters the address of the SDG in the user interface, which results in the PIRANHA Session Manager will establish a remote service invocation session with that SDG.

When creating the INVITE request it must also create the SDP-offer and setting the udn-attribute to a value known a priori to exist in the home network. In our testing, we only support UPnP AV Media Servers, and therefore do not use this value.

2) PIRANHA Session Handler: Handles requests about session establishment and, in the future, termination of sessions.

It will parse the SDP-offer and see if it can discover the requested device. If the device is available then it adds a port mapping for the device's private IP address at the IGD for the media plane.

3) PIRANHA Session: Common for both the PIRANHA Session Manager and PIRANHA Session Handler, and has information about port mapping and the SIP dialog. When a message comes in (SIP request or response) those components can look up the session information in a table, based on the dialog identifier.

4) Media Server Proxy: This component helps to make PIRANHA transparent for the user interface in PIGA. The user interface sees it as a normal media server and use it as the normal Media Server control objects, because the proxy inherits from that class. However, this proxy sends requests to the external DMS, in the media plane of the PIRANHA session.

When browsing the DMS the browse result includes a Uniform Resource Identifier (URI) [15]. Usually this URI contains a private network address, such as 192.168.1.3 in our scenario, because the DMS does not expect requests from external networks. Therefore, the Media Server Proxy must change this element of the browse result so the DMP can locate the resource, if the user requests to play it.

V. PROTOTYPE RESULTS

In our initial test configuration, with the home and visited networks one hop away from each other, we have successfully streamed music from a UPnP Media Server, TwonkyVision Media Server, to a UPnP Media Renderer, D-Link DSM320, with HTTP as the transport protocol, as required by DLNA.

As expected, video streaming over HTTP does not work very well over WAN. Our preliminary investigation of this issue indicates that the problem is related to the TCP sliding window mechanism.

With our prototype client, we have shown that it is feasible on a standard mass-market phone to support both the SIP and
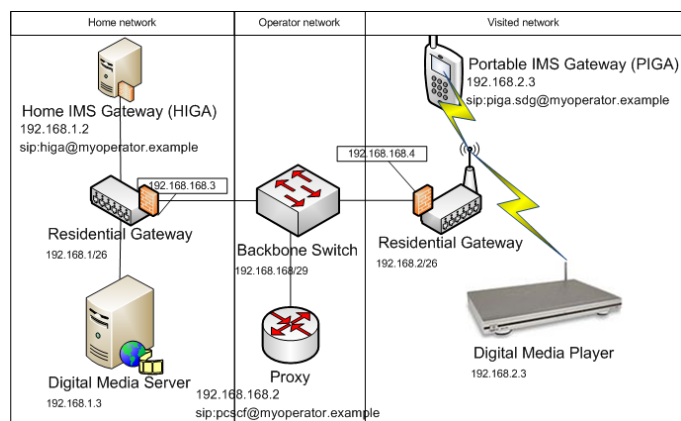

Figure 5 Deployment of the prototype system.

the UPnP stacks simultaneously. This makes it possible to realize new fixed-mobile converged applications.

VI. Related work

In [16] a similar protocol for remote service invocation is proposed, however with a different approach to handling the "resource-uri" issue. They rather let the terminating network's gateway function be part of the media plane, and mediate the information sent between the two networks.

Other solutions exist as well, such as [17, 18], where the service invocation requests are sent using the SIP MESSAGE-requests. In our opinion, it is however better to separate the signaling and the media plane for more flexibility, such as adding more media streams to the session.

VII. Conclusion and future work

Our prototype shows that our protocol works and allows services described by different service discovery mechanisms to cooperate, through the introduced Service Discovery Gateway function. However, HTTP streaming, which is the only mandated transport protocol of the Digital Living Network Alliance (DLNA), is causing issues related to the TCP flow control mechanism that should be further investigated.

We will later investigate how the protocol behaves over a longer distance, with more hops between the home and visited networks, and this might result in further TCP problems.

Our solution for fixing the "resource-uri" issue works, but we will continue to look for other options, which requires less knowledge of a service's semantics. Related to this issue is that our solution is limited by the cascading NAT problem [19], which we will also investigate further.

Finally, our current prototype does not handle updating and closing the session, and does not handle remote service discovery. Session updating and closing is described above, and the mechanism for selecting the service to be controlled is included in the protocol design (the udn-attribute). Remote service discovery is important to provide a good user experience.

Acknowledgment

References

[1] G. Camarillo and M. A. García-Martín, *The 3G IP multimedia subsystem (IMS) : merging the Internet and the cellular worlds*, 2nd ed. Chichester: Wiley, 2006.

[2] J.Rosenberg, H.Schulzrinne, G.Camarillo, A.Johnston, J.Peterson, R.Sparks, M.Handley, and E. Schooler, *SIP: Session Initiation Protocol*, IETF RFC 3261, http://www.rfc-editor.org/rfc/rfc3261.txt.

[3] E.Guttman, C.Perkins, J.Veizades, and M. Day, *Service Location Protocol, Version 2*, IETF RFC 2608, June, 1999; http://www.rfc-editor.org/rfc/rfc2608.txt.

[4] M. Jeronimo and J. Weast, *UPnP design by example : a software developer's guide to universal plug and play*. Hillsboro, Or.: Intel Press, 2003.

[5] Y.Rekhter, B.Moskowitz, D.Karrenberg, G. J. d. Groot, and E. L. L. rfc, "Address Allocation for Private Internets," 1996.

[6] M.Handley, V.Jacobson, and C. P. L. rfc, "SDP: Session Description Protocol," 2006.

[7] J.Rosenberg and H. S. L. rfc, "An Offer/Answer Model with Session Description Protocol (SDP)," 2002.

[8] J. P. L. rfc, "Transmission Control Protocol," 1981.

[9] D.Yon and G. C. L. rfc, "TCP-Based Media Transport in the Session Description Protocol (SDP)," 2005.

[10] P.Srisuresh and M. Holdrege, *IP Network Address Translator (NAT) Terminology and Considerations*, IETF RFC 2663, http://www.rfc-editor.org/rfc/rfc2663.txt.

[11] P.Srisuresh and K. Egevan, *Traditional IP Network Address Translator (Traditional NAT)*, IETF RFC 3022, http://www.rfc-editor.org/rfc/rfc3022.txt.

[12] J. R. L. rfc, "The Session Initiation Protocol (SIP) UPDATE Method," 2002.

[13] A. Häber, F. Reichert, and A. Fasbender, "UPnP Control Point for Mobile Phones in Residential Networks," in *15th IST Mobile & Wireless Communication Summit*. Myconos, Greece, 2006. http://mobilesummit2006.org/

[14] T. Cagenius, A. Fasbender, and L. Barriga, "An IMS Gateway for Service Convergence in Connected Homes," in *45th Congress of the Federation of Telecommunications Engineers of the European Community (FITCE)*. Athens, Greece, 2006.

[15] T. B. Lee, R.Fielding, and L. Masinter, *Uniform Resource Identifier (URI): Generic Syntax*, IETF RFC 3986, http://www.rfc-editor.org/rfc/rfc3986.txt.

[16] O. Yeon-Joo, L. Hoon-Ki, K. Jung-Tae, P. Eui-Hyun, and P. Kwang-Roh, "The DLNA Proxy System Architecture for Sharing In-Home Media Contents via Internet," 2006.

[17] B. Kumar and M. Rahman, "Mobility support for universal plug and play (UPnP) devices using session initiation protocol (SIP)," 2006.

[18] A. Brown, M. Kolberg, D. Bushmitch, G. Lomako, and M. Ma, "A SIP-based OSGi device communication service for mobile personal area networks," 2006.

[19] R. J. Vijay K. Gurbani, "Contemplating some open challenges in SIP," *Bell Labs Technical Journal*, vol. 9, pp. 255-269, 2004.