

An Evaluation of Architectures for IMS Based Video Conferencing

Richard Spiers, Neco Ventura
University of Cape Town
Rondebosch South Africa

Abstract—The IP Multimedia Subsystem is an architectural framework for delivering multimedia services over an Internet Protocol (IP) network. Originally, it was specified for wireless networks, but has since evolved to incorporate fixed line access as well. It forms part of a Next Generation Network (NGN) which is defined as a packet-based network where the service functionality is independent of the underlying transport technologies. This allows new converged services to be implemented on top of an existing packet switched network.

The IMS is not intended to act as a standard for services, but rather to aid the creation of new multimedia applications. As such, it has a horizontal control layer which separates the access network from the service layer. Each new service developed does not need its own control functions but can rather reuse the common infrastructure provided by the IMS. Examples of these features are: Quality of Service (QoS), user authentication, charging and security.

Thus the IMS provides a good platform to develop multimedia rich services. This paper evaluates two different architectures for providing a video conferencing service over the IMS. It discusses the different methods of handling the control signalling, as well as different ways of controlling the multimedia traffic. The first architecture analysed is modelled upon a standard server-client archetype. The second architecture discussed is based upon a distributed P2P model. It concludes that each architecture has its own set of advantages and disadvantages and both systems are viable for different environments.

Index Terms—conference,IMS,multimedia,video

I. INTRODUCTION

Currently, broadband penetration is on the rise. Faster connection speeds are being made available to consumers, with higher limits on the amount of data that any one consumer can transfer. This increase in bandwidth makes video rich services available to end users. Several different companies have been quick to take advantage of this capacity and have started providing advanced services to end users over the Internet. This can be seen by the success of YouTube and Skype. Network operators are in danger of becoming simple bit pipe providers, unless they innovate and come up with services that add value for the consumer. Consumers have ready access to free or low cost VoIP solutions through the Internet which is driving down network operator's revenue from these markets.

In order to create new streams of revenue by developing new services, as well as to reduce the cost of running and maintaining these services, network operators have begun to look to Next Generation Networks (NGNs) [1]. A NGN is defined by the International Telecommunication Union (ITU) as “a packet-based network” that makes use of “multiple broadband,

QoS-enabled transport technologies and in which service-related functions are independent from underlying transport-related technologies.” The Telecommunications and Internet converged Services and Protocols for Advanced Networks (TISPAN) standardises IMS as a subsystem of NGNs [2].

One of the advantages that the IMS has is that it uses standard Internet protocols with some extensions such as the Session Initialisation Protocol (SIP) and DIAMETER. This opens up the development environment to a much wider range of programmers, unlike previous telephony signalling protocols such as Signalling System #7 which required more specialised knowledge. SIP allows for the initialisation and modification of multimedia sessions, and can be used to set up an IPTV session, a video calling session, a video conferencing session or any form of multimedia session. This aids in the interoperability between different implementations of services, as well as allowing different sections of functionality to be reused for different services.

However, this flexibility leads to several different approaches that can be used to set up video conferencing services. Research is still needed in order to determine which video conferencing architecture is optimal for different scenarios. The control signalling needs to be examined as well as the flow of multimedia information such as the video feeds, as the IMS separates these into two different layers. Each of these layers can be routed along different paths. In a video conferencing system, the control signalling would contain information such as who is in the conference and how they can join it, whereas the multimedia signalling would contain the actual media feeds flowing between each participant (or between the server and a client). These media flows consist of audio and video streams which need to be handled differently. Each additional participant or client involved in the video conference needs its own unique signalling, which has an impact on the scalability of any video conferencing system.

The aim of this work is to implement, evaluate and discuss two different system architectures for creating a video conferencing system utilising the IMS. The first system discussed solves the problem by following a traditional server-client model. The second architecture examined follows a peer-to-peer method of communication, with the participants communicating directly with each other.

Both systems consist of a modified User Agent (UA) (or client) and a SIP Application Server (AS). A standard IMS testbed was created in order to evaluate the performance of the two architectures.

II. RELATED WORK

“A Framework for Conferencing with the Session Initiation Protocol (SIP)” has been proposed by the Session Initiation Proposal Investigation (SIPPING) work-group. This is available as RFC4353 [3]. It discusses a form of conferencing known as “tightly coupled SIP conferencing”. This system has a central control point or server to which each participant communicates control signalling, and is taken into account when designing the two implementations discussed in this paper.

A. Buono et al. have created an IMS compliant architecture for distributed conferencing [4]. Their approach consists of taking an existing solution for setting up conferences and mapping the actual system components with logical IMS functions. Thus there are potential differences between their implementation and the IMS paradigm. Another factor to consider is that they considered an audio only approach, and are making progress on integrating video into their system. However, it is not a true form of multi-party video conferencing as only the person who currently holds the “video-floor” is seen by the other participants. The implementations discussed in this paper were created from scratch, and are designed to fit into an existing IMS implementation. This paper aims to determine a viable implementation of the specific control and signalling needed by the IMS framework, as well as to examine the complexities that arise when working with multiple video feeds. These complexities include the fact that the bandwidth necessary is increased exponentially and that the video can not simply be mixed in an additive sense.

III. SUPPORTING SOFTWARE

An IMS testbed was created in order to execute the implementation of the different architectures. This testbed comprised of the core IMS components provided by the Open Source IMS Core project [5] created and released by the Fraunhofer Institute FOKUS. It comprises of the three IMS Call Session Control Functions (CSCFs) which are based upon the SIP Express Router (SER) [6], a high-performance,high-reliability SIP server. The Serving-CSCF (SCSCF), the Proxy-CSCF (PCSCF) and the Interrogating-CSCF (ICSCF) are all written in C, whereas the Home Subscriber Server (HSS) which contains information about the users is written in JAVA. They also provide a web user interface to the HSS which allows rapid configuration of application servers and initial filter criteria, which are used in both implementations. Modifications were made to the UCT IMS client [7],[8] which was used as the User Agent in each implementation. This client was the first IMS client released to the open source community, and is used in several research projects around the world. One of its goal's is to provide true IMS signalling in order to allow the development and testing of various different services. The client itself uses several free open source libraries and is released under the GNU Public License version 3 (GPLv3). In order to support the video conferencing capabilities the client was extended through the use of the Farsight framework. This framework has a generic API and several different plugins which allow programmers to create applications capable

of handling several different streaming protocols. All the necessary network traffic analysis was done with the assistance of the Wireshark Network Protocol Analyzer, a freely available open source network traffic recorder and analyser.

IV. UCT VIDEO CONFERENCING SERVER CLIENT IMPLEMENTATION

A. Architecture

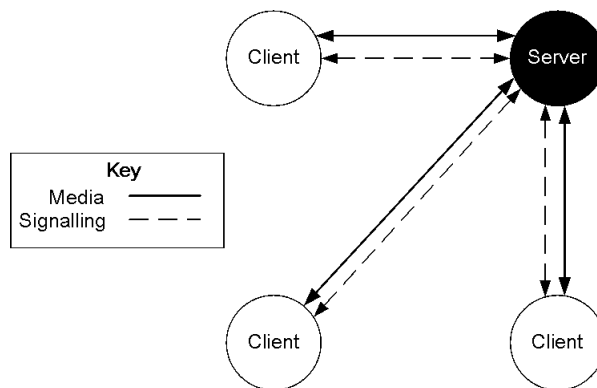


Figure 1. A server-client mode for signalling as well as the media traffic.

The first architecture discussed in this paper consists of a dedicated server that each client communicates with. The media stream consists of a single duplex stream that flows between the client and the server. The server receives each stream from each participant in the conference, mixes them together and then sends a mixed stream back to each participant. For the client->server stream, a unicast IP stream is required. The stream that the server sends back could be a unicast IP stream, or a multicast stream that each client received. If a more complicated video mixer was created, several different multicast streams could be setup. This would allow the client to receive the appropriate quality level for his connection.

In order to participate (join) a conference, the client needs to know about this conference server. The conference server could be contactable through a simple URI (e.g. video_conference@conference.open-ims.com) which is human readable (and therefore much easier to remember), or through a more advanced function. This function could be a web interface that the user selects a conference through, or even a hashing algorithm which generates once-off URIs (e.g. 23547adca72482@conference.open-ims.com). The particular hashing algorithm used would have to ensure that these URIs can not be easily guessed or constructed. Once a particular hash URI had been generated it could be sent to each participant to allow them to participate in this particular conference. The architecture behind these different forms of conference URIs remains the same. An Initial Filter Criteria (IFC) is set up which directs the SIP messaging containing the relevant URI to the appropriate AS.

All the clients that wish to participate in a conference using this architecture must share a common codec to which the conference server can transcode the resulting mixed video feed. Each client can not negotiate a preferred quality level

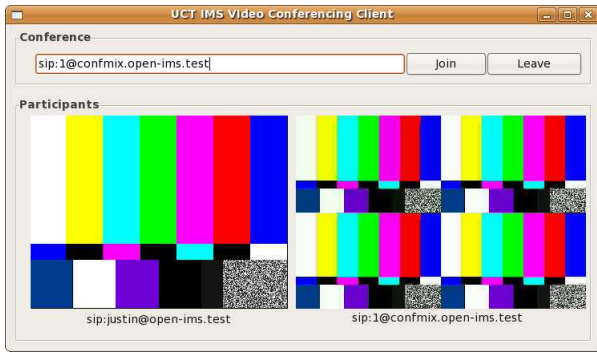


Figure 2. UCT IMS Server-Client Video Conferencing client in a four-way call (test-video).

with the server, unless a complicated video mixer is integrated. However, this would place a very high burden on the computational resources of the conference server. Through the standard IMS QoS mechanisms a form of QoS can be provided and the necessary resources allocated, however this will now be limited to the minimum quality level necessary to receive the existing video feed, and can not be reduced once the conference has begun.

B. Signalling

In both architectures, a SIP AS was developed. In this architecture, it acts as a terminating SIP User Agent (UA). Each participant initialises a session with the server itself in order to join the conference. This is done by sending a SIP INVITE request to the conference server. At this moment, the core IMS network allocates the appropriate network resources as well as the necessary charging information. Once the server has received the INVITE request it responds with a 200 OK response. The user is then added to the list of participants (or roster) and begins receiving the user's video feed. This video feed can be sent back to the client, manipulated according to the network operator's policies (e.g. with advertisements) or it can be delayed until there are two or more participants.

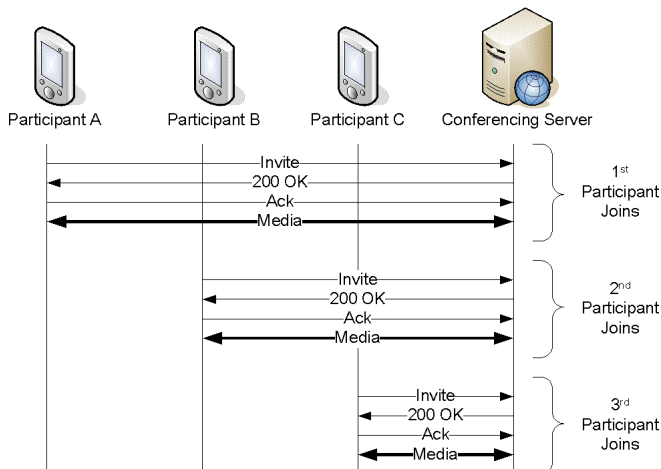


Figure 3. Diagram showing the necessary signalling between three participants and the conference coordination server.

This choice depends on the network operators preferences. This architecture allows for a high degree of control by the network operator. The network operator has full control over the media stream, and can manipulate it according to the network operators policies. For example, the network operator may wish to offer different quality levels at different prices, or may wish to overlay some text or video over the stream for advertising purposes. This implementation would also allow the users to remain anonymous if they so wished, as they would not be communicating directly with another participant, but rather through a conference server which could allow the person to contact somebody without giving any contact information away.

C. Signalling Overheads

In both architectures discussed in this paper, pure SIP signalling was used for the control signalling. This allows the IMS network operator to have a high degree of control over the sessions. There is minimal signal overhead in this implementation, as each client only has to set up one session, regardless of the number of current participants. This means that the signalling for each new participant remains constant, and does not increase. This is in direct contrast with the increase in signalling discussed under the P2P architecture below.

D. Traffic Overheads

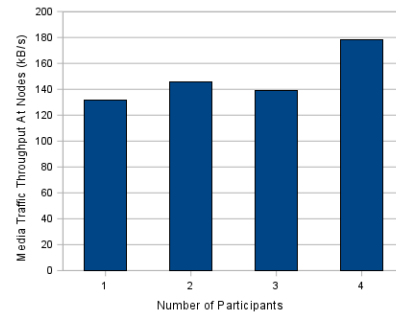


Figure 4. The measured media traffic at the conference client nodes.

The chosen resolution of the resulting mixed video feed was 640 x 480. This was fixed and was not dependant on how many video feeds we were mixing together (the input video feeds were scaled accordingly). In Fig. 4 we can see that there were some small variations in the bandwidth used for the media, as the complexity of the encoded video frame was increased depending on how many live video feeds we were mixing. We were encoding using a variable bit rate codec (h263+), which uses an appropriate amount of bandwidth depending on the complexity of any given scene. All the tests were conducted over a period of 60 seconds and were averaged. The tests were repeated for a varying number of participants, with varying test sequences to observe the behaviour of the implementation.

E. Discussion

This implementation sacrifices flexibility on the client side in order to create a framework that is more scalable. There is

very little signalling overhead in this implementation, and it allows for a roughly constant media stream to be generated. It also allows the network operator a much higher degree of control over the whole process, and allows for different price levels to be created. Another benefit of this approach is that it will allow a normal “video call capable” client to interact in a conference, as the conference server looks like another video call participant to the client. Thus, this system is recommended for conferences with limited client upload bandwidth, or for conferences with many different participants.

V. UCT VIDEO CONFERENCING P2P IMPLEMENTATION

A. Architecture

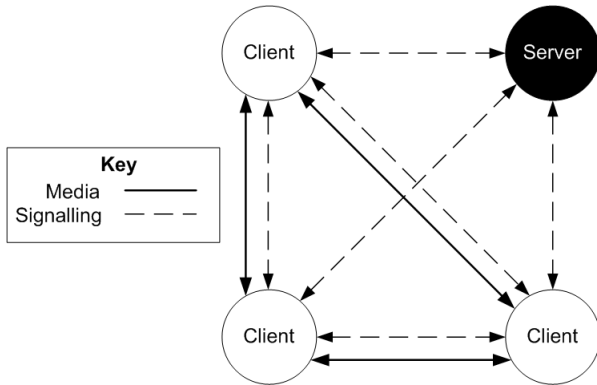


Figure 5. A distributed media and hybrid signalling scheme.

This architecture utilises a hybrid signalling scheme and a distributed approach to the media traffic distribution. When creating a distributed system, each client can either set up a duplex stream with each other client, or they can each broadcast to a multicast IP address. A single broadcast from each client to a multicast IP address would be the most efficient use of a client’s upload bandwidth, but unfortunately multicast has not been deployed extensively throughout the Internet and is thus more suited to closed environments. This would also reduce the flexibility of each client being able to negotiate the appropriate quality level for each session. For the evaluation of this model, we chose to utilise unicast duplex streams between each client.

In order for each of these distributed clients to communicate with each other, a central point is needed. A SIP AS serves as a conference coordination server in this design. If there was no central server, each participant would have to discover the other members of the conference through an external mechanism, or they would have to have prior knowledge of the future conference participants before the conference begins. The SIP AS maintains a list of who is currently in the conference, as well as notifying the current members whenever a new participant joins the conference or whenever an existing member leaves the conference.

Besides eliminating the need for a computationally powerful video mixing server, this design is beneficial for each client as it allows them to negotiate their own desired quality level between each client. The network can assign network resources through the existing IMS QoS mechanisms to guarantee this

level of bandwidth between each client, as well as allowing the clients or network to degrade the quality level depending on the currently available capacity (network or client side). As well as negotiating the necessary QoS, each client can also negotiate a set of codecs that they are both capable of utilising. This means that different clients with different codecs have a higher chance of interacting successfully in this design. If the network operator desires, this signalling can flow through a central conference server for more control over the system. However, this would place a high burden on the conference co-ordination server without much gain as the signalling will already have been captured by the appropriate charging mechanisms existing in the IMS. For these reasons, we chose to implement a hybrid signalling scheme as can be seen in Fig. 5

B. Signalling

As mentioned above, the coordination server is a SIP AS. This AS implements the SIP conference event package. This is functionally similar to the concept of “presence”. Each user sends a SIP SUBSCRIBE request to the conference server in order to be added to the current list of participants (roster). Whenever a new user subscribes, a SIP NOTIFY message is generated for all existing conference members. These NOTIFY messages contain a XML-formatted SIP URI of the new user in the body of a message. This tells the other participants who the new user is and how to contact him or her.

On receipt of a NOTIFY request the existing conference participants each send a SIP INVITE message to the newly discovered URI. Each of these INVITE messages are processed individually, with complete SDP negotiation to allow the clients to determine the best codec to use, together with the appropriate ports to use. As with any IMS session establishment, the core network can allocate the appropriate resources as well as trigger the necessary charging mechanisms. A 200 OK response is generated by the new member for each accepted INVITE message, and the media can begin to flow between these user agents.

Compared to the architecture discussed previously, this architecture allows for a much greater degree of flexibility,



Figure 6. A Four-way call implemented in a distributed fashion.

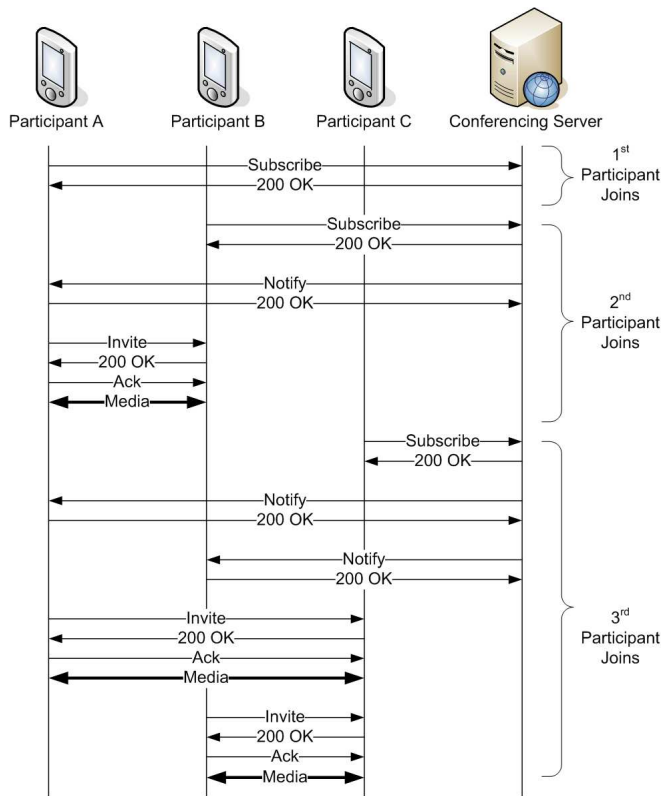


Figure 7. The necessary control signalling between the conference coordination server and three participants

allowing clients with mismatched bandwidth capabilities and different codec sets to inter-operate. Lower quality sessions can be established, without impacting on the overall experience of the other conference members. It can be noted that a client that only supports a subset of functionality, e.g. voice only can still participate in the conference as each session is negotiated separately. Likewise, a lower quality stream can be established between clients with less available bandwidth.

C. Signalling Overheads

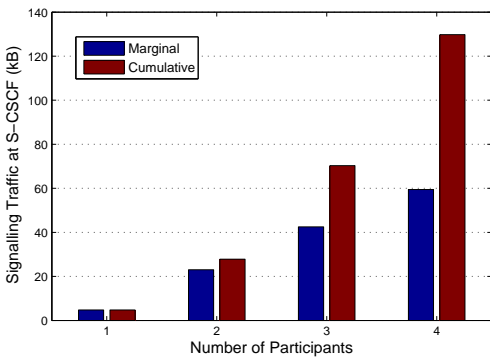


Figure 8. The measured SIP signalling overhead at the S-CSCF when a new participant joins the conference.

In contrast to the previously discussed server-client model, the hybrid signalling scheme discussed here grows exponentially with the addition of each new member. The signalling

is still based entirely on SIP which allows a great deal of control and the various different IMS charging mechanisms to be utilised. However, this does place an extra burden on the IMS core network.

As can be seen in Fig. 8 the marginal signalling overhead increases for each new participant added to the conference. The signalling traffic measured includes both the event subscriptions and notifications, and the subsequent signalling between the individual participants. The reason for this growth can be seen in Fig. 7. Each new member of the conference must be invited by all the existing members, resulting in an exponential explosion of signalling. All of these signal messages must flow through the S-CSCF according to the IMS paradigm, and thus this is the node that we chose to record our measurements.

D. Traffic Overheads

Having a duplex stream between each member of the conference means that there is a lot of bandwidth used in this system. While the overall bandwidth used is increasing exponentially with each new participant, the bandwidth at each client node only increases in a linear fashion as each new participant means that one new duplex stream is created at each client node. This increase of bandwidth can be seen in Fig. 9 which shows the averaged media throughput at a client node. The tests were once again run with several different participants and different video test sequences, and averaged over a 60 second period.

E. Discussion

As mentioned above, the marginal signalling overhead increases with the addition of every new conference participant. This leads to an exponential increase in the overall signalling used, and may place an unreasonable burden on the core network for large number of participants. However, the client bandwidth used scales linearly as can be expected due to the new duplex stream. Even though this is only a linear increase, it still places a heavy burden on the available upload capacity of each client. Thus it can be concluded that although this particular architecture is very flexible and offers several advantages including support for different quality video feeds,

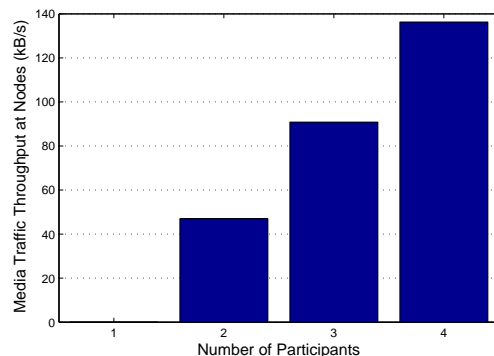


Figure 9. The measured media traffic at the conference client nodes.

it is only suitable for conference sessions with a low number of participants.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we presented research on two different architectures for an IMS-based video conferencing system. Each implementation was designed, implemented and evaluated. Although they both appear to be feasible models for an IMS-based video conferencing service they each have their advantages and disadvantages. A traditional server-client model offers finer network control together with a great reduction in signalling and media overheads, whereas a P2P implementation allows great flexibility, but at the expense of higher overheads. Further work will involve mixing these two architectures to produce a hybrid architecture that allows for a greater degree of flexibility at the client side, yet retaining the low signalling overheads associated with the server-client model. Another area of research can be identified as eliminating the need for a central conference focus point in the P2P model, as well as creating a video-mixer that is capable of producing several different quality levels.

REFERENCES

- [1] M. Weldon, "IMS and IPTV, Perfect Together," January 2006.
- [2] Gilles Bertrand, "The IP Multimedia Subsystem in Next Generation Networks." May 2007.
- [3] "A Framework for Conferencing with the Session Initiation Protocol (SIP)," February 2006. <http://www.ietf.org/rfc/rfc4353.txt>.
- [4] A. Amirante, T. Castaldi, L. Miniero, and S. P. Romano, "Improving the Scalability of an IMS-Compliant Conferencing Framework through Presence and Event Notification," in *Proceedings of the 1st International Conference on Principles, Systems and Applications of IP Telecommunications (IPTComm)*, New York City, NY, July 2007.
- [5] Fraunhofer Institute FOKUS, "Open Source IMS Core," <http://www.openimscore.org/>.
- [6] Iptel.org, "SIP Express Router," <http://www.iptel.org/ser/>.
- [7] University of Cape Town, "UCT IMS Client," <http://uctimsclient.berlios.de/>.
- [8] D. Waiting, R. Good, R. Spiers, and N. Ventura, "Open Source Development Tools for IMS Research," *Tridentcom*, 2008.

Richard Spiers is studying for his Master of Science Engineering Degree at the University of Cape Town, having just graduated with a B.Sc. in Electrical and Computer Engineering. His undergraduate thesis involved creating an IPTV system over the IMS, and he is currently one of the UCT IMS client developers.