Aggregation Efficacy of Resource List Servers in IMS Presence Services

Michael Pitman and Neco Ventura
University of Cape Town, Rondebosch, South Africa
Email: {mikep, neco}@crg.ee.uct.ac.za

***Abstract*—The IP Multimedia Subsystem (IMS) is the Internet Protocol-based service-provisioning framework for mobile and fixed-line convergence, as specified by the Third Generation Partnership Project (3GPP). The provision of novel applications and services to subscribers is an important growth area for telecoms providers in an age of connection commoditisation. Presence is an important growth service in IMS and is likely to become both an important stand-alone service, and plug in-component for other applications. The aspect of the presence architecture examined here, Resource List Servers, forms part of the presence framework, improving the bandwidth use for constrained wireless links.**

This paper covers the presence architecture and signalling in the IMS, before examining the operation of a Resource List Server. The operational states of the RLS are examined, and different methods for aggregation are discussed. Results of simulated aggregation are presented and discussed.

***Index Terms*—IP Multimedia Subsystem, Presence, Resource List Server, Aggregation**

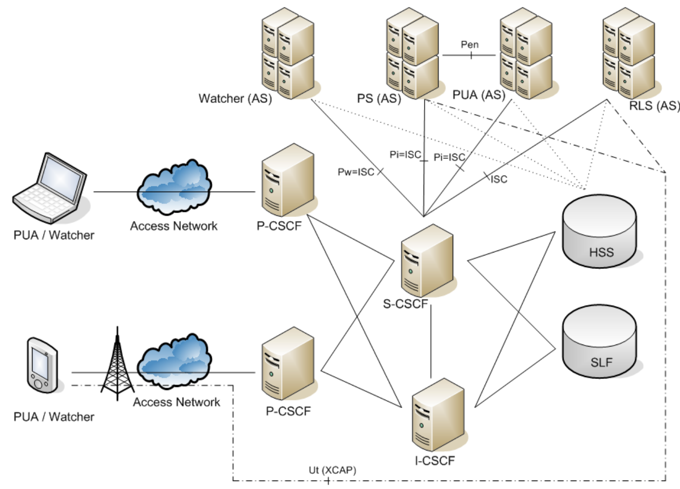

Figure 1. SIP-based Presence Architecture in the IMS [6]

I. Introduction

The IP Multimedia Subsystem is a service-provisioning framework delivering IP multimedia services to subscribers through the emerging mobile all-IP network [1]. Specified by the 3GPP, it allows service interoperability and roaming between networks, as well as providing bearer-level control. The IMS is a key enabler for fixed-mobile convergence, through the definition of interfaces and gateways between the respective bearer technologies, converging the operation of traditional circuit-switched telephone networks with packet-switched Internet technology [2]. Because the IMS abstracts service provisioning away from the bearer technologies, subscribers' mobile services and applications can be accessed from different end systems or through different access networks, for which QoS negotiation is supported.

The ease of service creation and provisioning in the IMS is the primary motivation for its development and deployment. The IMS Core Network abstracts away from the access network, linking subscribers to services and providing general network services [3]. The IMS application server plays the main role in service provisioning, providing network-provider services as well as any third-party services to users. Interaction between applications acting on a subscribers' behalf can be expected to occur, as applications can often act as proxies for their subscribers. However, the implementation of multimedia and other applications for third parties is considered to be outside of the 3GPP scope [1], and left to service providers.

This paper examines the functioning of one particular application server, the Resource List Server (RLS), existing within the presence framework in the IMS. The RLS proxies all presence information destined for a client on a bandwidth-restricted access network, and aggregates the information, to make more efficient use of the bandwidth available. The need for such an AS within the presence framework is examined, and the various methods of achieving the best aggregation reviewed.

The paper is organised from this point into 5 sections, beginning with aspects of presence in the IMS. An investigation into the reduction of traffic through the access network when using a Resource List Server (Section IV) is provided. The authors' conclusions are then given, with a description of future work.

II. Presence in the IMS

Presence in the Internet is a service that allows sets of users to be informed about the availability and means of communication of other users, and in particular their willingness to receive communications [2]. Presence has become a basic service incorporated into numerous Internet applications, but in many such cases the presence application operates over the developers' non-interoperable, proprietary protocols, hence restricting any interaction between other similar applications. This lack of interoperability and the absence of a dominant standard for presence has continued for some time [4], [5].

Presence in the 3GPP-specified IMS is closely based on the Request for Comment (RFC) presence specifications [7], released by the IETF regarding presence in the Internet. Many of the IETF protocols and logical entities are replicated in the IMS specification, although some of these IETF-specified entities can be mapped to existing IMS entities [6].

In the IMS, the network-based presence entities are implemented as application servers, and are located on the IMS application plane. All standard SIP-based presence traffic flows from the Presence User Agent (PUA) through the Proxy-Call Session Control Function (P-CSCF) and Serving-Call Session Control Function (S-CSCF) to the PS (Presence Server), or in reverse.

The presence information uploaded by a subscriber in the IMS is sensitive to a number of factors, such as the media types a subscriber may be able to accept, which watchers are viewing the presence information, and the subscribers' presence preferences [2]. Furthermore, the presence information for a subscriber may be aggregated and updated on their behalf by a network-based PUA, freeing the PS of this overhead. The PS in this case will act as an online presence information repository, rather than an actively intelligent application, increasing its potential throughput.

The primary entities in the IMS presence framework are the Presence Server (PS), the Resource List Server (RLS), the XCAP Presence Server, and the Presence User Agent (PUA) devices, such as the Presentity and Watcher user devices. The architectural arrangement of these entities is given in Figure 1.

A. Presence Signalling in the IMS

As stated, the SIP signalling for IMS presence follows the standard IMS signalling pathway from the PUA to the PS (or other AS), through the IMS Core. The PUA initiates the signalling to the PS for service registration, to subscribe to their resource list or additional presentities, and to upload presence information updates to the PS. The PS updates the presence information held by the subscribers' Watcher through NOTIFY messages, as well as updating the Watcher list of the Presentity. The Resource List Server acts as an aggregating proxy server between this traffic from the PS to the respective PUAs. The presence interfaces prefixed with "P" in Figure 1 are existing IMS SIP or Diameter interfaces, which are mapped to presence-oriented functions. The "Pen" interface is important as it allows an Application Server to publish presence information to a presentity's Presence Agent, acting in the role of a presence client or PUA.

The PUA is also able to interact directly with the PS using the HTTP-based XCAP interface. The client is able to upload their specific application configuration information to the server using XML, allowing for an individually tailored service. This interaction with the PS bypasses the IMS core because of the use of HTTP as the carrier protocol, which passes over the Ut interface [6]. Figure 2 shows the messaging performed during an upload of presence information to the PS from the PUA. An example of a subscription to the RLS is given in Figure 4. While the PUA only subscribes to it's resource list, the RLS subscribes to each Presentity in the resource list on behalf of the PUA, which is not directly indicated in the diagram.

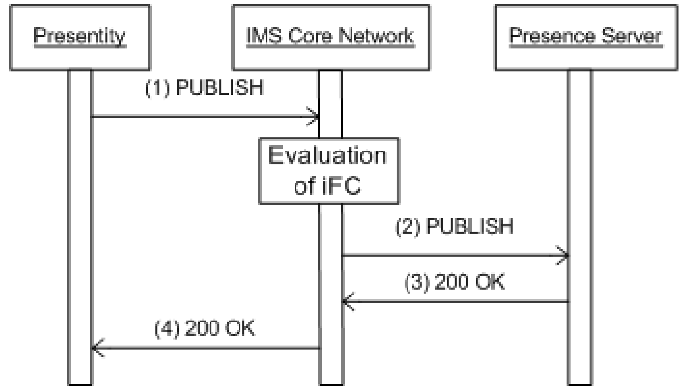

Figure 2. Signalling for a Presence PUBLISH

B. Presence Information Structure

The desirability and usefulness of presence information makes it an important feature to incorporate into applications. This was recognised by the IETF in 2000, and initial requirements for standards protocols were released [7], [4]. This specification accounted for the future growth of Internet and IP services in mobile and wireless networks, as well as the access restrictions such networks would impose, such as limited bandwidth, latency and intermittent network coverage, and device restrictions such as screen size or limited power. The need for a protocol design that was reasonably efficient for small payloads was also noted at this time. The Common Profile for Presence (CPP) [5] defined semantics and data formats for common presence services, to allow gateways between presence services. The Presence Information Data Format (PIDF) [8] expanded on the data formats in [5], introducing XML as the format for presence information encoding. By using XML, information is structured, and extensions to include additional information can be made. The Rich Presence Extension (RPID) [9] adds a framework for such extra information, much of it designed to be automatically derived by the system, such as location, device and device capabilities, or time zone. All extensions to the XML format must however not modify the base structure or semantics of the presence information data format, ensuring compatability between systems of different capabilities [8].

When using the rich presence enhancements defined in [9], the amount of information that can be held for a Presentity can become very large. The example given in Figure 3 can be considered a relatively simple, yet comprehensive example of presence information, which contains a total of 881 bytes. Presentities containing many of the rich presence extensions can become much larger, with examples given in [9] comprising almost 2 700 bytes. As was shown in Figure 3, much of this data is the overhead of using XML to encode a relatively small amount of pure presence information.

The hierarchical structure of XML becomes highly beneficial, however, when updates to the presence information

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
      xmlns:im="urn:ietf:params:xml:ns:pidf:im"
      xmlns:myex="http://id.example.com/presence/"
      entity="pres:someone@example.com">
   <tuple id="bs35r9">
     <status>
       <basic>open</basic>
       <im:im>busy</im:im>
       <myex:location>home</myex:location>
     </status>
     <contact priority="0.8">im:someone@mobilecarrier.net
</contact>
     <note xml:lang="en">Don't Disturb Please!</note>
          <note xml:lang="fr">Ne derangez pas, s'il vous
plait</note>
     <timestamp>2001-10-27T16:49:29Z</timestamp>
   </tuple>
   <tuple id="eg92n8">
     <status>
       <basic>open</basic>
     </status>
     <contact priority="1.0">mailto:someone@example.com
</contact>
   </tuple>
   <note>I'll be in Tokyo next week</note>
</presence>
```

Figure 3. Presence Information in XML with Status Extensions [8]

held by watchers is made. Only the specific tags and fields of the Presentity that have changed are updated, because XML enables snippets to be added to or replaced in the Presentity XML description in the correct place [10]. While the complete Presentity stored by the Presence Server would be sent to a subscribing Watcher when a new subscription is opened, any subsequent updates to the Presentity are much smaller in size.

Similar IETF data format specifications exist for Watcher information [11], which use XML to describe the current status of all the subscriptions to a Presentity. The size of this Watcher list can also become large, but again, the entire list is only sent to the PUA when it subscribes to the service. Subsequent updates to the Watcher list also benefit from the hierarchical XML, whereby updates contain only the specific changes to the Watcher list.

III. IMS Resource List Server

The Resource List Server (RLS) in the IMS acts as a core network-based proxy for PUAs, aggregating the XML-encoded presence or watcher information from the PS to the PUA [12]. Subscribers to the RLS service subscribe to their personal resource list, which contains a list of the Presentities they subscribe to. The RLS subscribes to all presentities in the list on the subscribers' behalf, and maintains the respective subscriptions while the PUA is active. Subscriptions with the RLS exist in one of two states: the initialisation state, and the steady state. During the initialisation state, large amounts of presence information, comprising all the subscribed Presentities' presence information, is sent to the PUA. During the steady-state, which begins after the peak of the initialisation state has subsided, updates by watched Presentities are forwarded to the PUA, which as explained in Section II-B, are considerably smaller than the full presentity.

A. Data Encoding and Session Protocol Overheads

As noted above in Section II-B, the design requirements of the presence protocol requires a protocol that is reasonably efficient for small payloads. The choice of XML was made because of its heirarchical nature, and because it is fully extensible, allowing new elements to be defined as required. However as shown in Figure 3, the encoding of presence information in XML leads to a considerable information-encoding overhead, even for relatively small amounts of presence information.

The choice of the Session Initiation Protocol (SIP) as the session control protocol for the Signalling Plane in the IMS [3] adds to the problem of bandwidth-heavy protocols, especially when used to transfer information with a significant encoding overhead. SIP messages are encoded in plain text in order to make them human readable, and to keep the protocol simple and easy to use. The operation of SIP is based on an HTTP-like request-and-response model [13], in which acknowledgements (ACKs) and OK messages are used to indicate if the communicating party received and understood the message. While [14] enables the use of compression for SIP messages, this approach to reducing the amount of data transmitted is not particularly feasible in a highly mobile environment, in which many of the devices sending or receiving presence information are likely to have limited processing capabilities and power reserves.

B. Resource List Server Signalling

The subscription of a watcher to a presentity through an RLS occurs in the same manner as a watcher subscribing directly to a Presentity, or its Presence Server. Once the subscription reaches the RLS, it handles all further signalling with the PS on behalf of the subscriber, and maintains an interface with the subscriber, as if it were the PS.

The purpose of the RLS, however, is to store addressable resource lists, allowing watchers using the RLS service to subscribe to their resource list held by the server [15], [16]. The resource lists contains XML encoded SIP URIs or other valid URIs for all the presentities the watcher maintains subscriptions to [16].

When a subscription to a resource list is made, the RLS subscribes to each presentity on the subscribers' behalf. The subscribed presence information is returned to the RLS in NOTIFY requests either by the Presentity, or their presence agents operating in the core network. The presence information received is stored by the RLS, and forwarded to the Watcher in NOTIFY requests [17]. The RLS ensures that the subscriptions to the presentities are maintained while the resource list subscription is active, and updates the Watcher regarding status changes when they occur in watched Presentities. The flow of

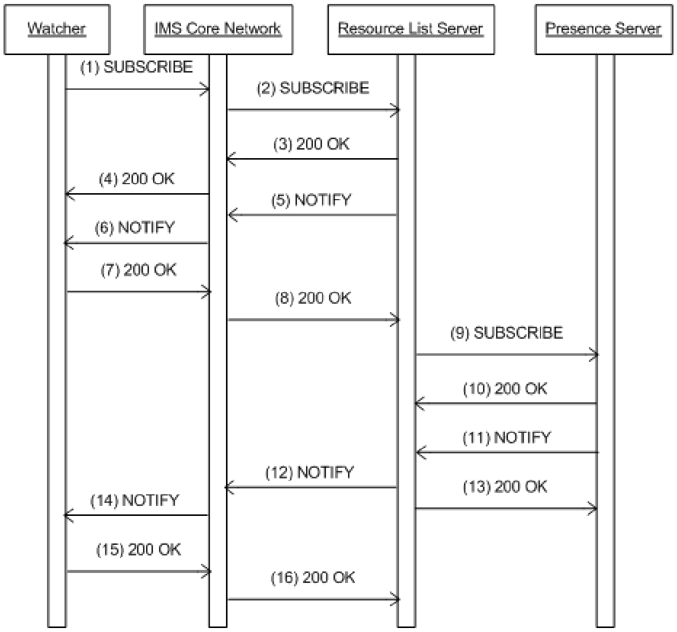

Figure 4. Subscription to a Presentity via a Resource List Server

updated information from the RLS to the PUA continues until such time as the PUA deregisters from the RLS.

The RLS is in the position, as a core-based proxy, to aggregate messaging sent to the client. By subscribing to a resource list at the RLS, the bandwidth-intensive subscription by the PUA to every presentity is removed from the access network, and instead occurs in the IMS core. It is still possible, and may be necessary, for the PUA to subscribe to additional presentities, but such subscriptions passing through the RLS can be added to the PUAs' resource list, making the individual subscription a once-off event.

As the RLS lies directly on the path of presence messaging from the respective PSs to the Watcher PUA, it receives and stores all the XML-encoded presence payloads carried by the SIP NOTIFY requests towards the PUA.

It is also possible to "poll" a Presentity, where a zero-timed subscription is made to the PS, and the presence information is obtained only once, with no subsequent updates sent to the watcher. Polling can either bypass the RLS or be handled by it, depending on the service providers' configuration for such events.

C. RLS Aggregation Methods

The RLS aggregates the stored presence information received from the PS, forwarding it on to the PUA by placing the presence or watcher information for several Presentities or Watchers into a single SIP NOTIFY packet payload. This results in a reduction in both the frequency of presence-related SIP packets arriving at the PUA as well as removing the attendant packet overhead of many small-payload packets. This reduction in traffic at the PUA can provide a significant performance enhancement for devices that access the service through low-bandwidth access networks, such as GPRS or fixed dialup connections.

The efficiency of the aggregation method employed depends on factors such as the operational state of the subscription, the number of Presentities or Watchers in the list, and the level of activity of the elements in the respective lists. The decision when to send aggregated methods can be made using a number of different methods. There are three primary methods, each of which will be implemented and examined in a testbed RLS implementation.

The first method examined is the use of time as a trigger for sending NOTIFY messages. The use of time gives a stable update rate for NOTIFY messages from the RLS to the PUA, and can ensure that the presence information is not stale, if the time period is well chosen. The drawback of this method however is that it does not react to bursty data, resulting possibly in either very large or very small NOTIFY updates being sent. The use of time-based aggregation is particularly useful for aggregation in steady-state subscriptions.

The trigger can also be based on the number of NOTIFY messages received by a particular Watcher or Presentity. By using the number of messages received it is possible to ensure that the NOTIFY messages sent contain the payloads of a few NOTIFY messages, as a guarantee that aggregation is being carried out, and that the SIP packet overhead per incoming NOTIFY packet at the RLS is reduced when forwarding to the PUA. The drawback of this method, however, is that there are no mechanisms to ensure that information delivered to the PUA is still fresh, and hasn't been stored at the RLS for long periods of time.

Basing the trigger on the maximum payload size that can be transmitted in a single message, using data size instead of the variable-sized payloads of a set number of packets, achieves the best possible SIP packet overhead to payload ratio. The payload size should be limited to that of the smallest Maximum Transmission Unit (MTU) along the path of transmission, to avoid breaking the message up further down the path. As the presence information for a Presentity will very often carry information regarding the type and capabilities of the device, it should not be difficult for the RLS to infer the MTU of the access network, and adapt the aggregation to that PUA accordingly. This MTU-based method also suffers from the problem of guaranteeing the timeliness of data. The low packet overhead to payload ratio makes it particularly applicable to initialisation-state aggregation, where large amounts of presence information pass through the RLS in a short period of time, rather than in a steady-state scenario with relatively low traffic volumes.

The use of time- and quantity-based methods individually have serious drawbacks in terms of the timely arrival of presence information and the relative efficiency of the aggregation, but when used together, can be used to eliminate the others' negative effects. This is most useful when a subscription is operating in a steady state, and the two methods can be used to reduce or eliminate the others' drawbacks. The ability to vary the respective parameters for the combined systems' operation will allow the aggregation to be tuned to specific subscribers needs, making the system dynamically efficient.

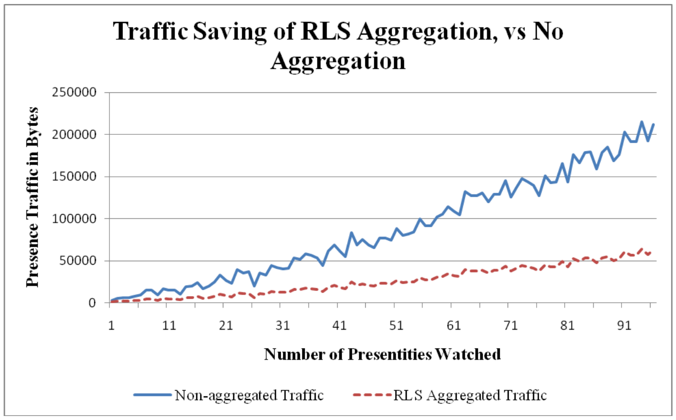

Figure 5. Traffic Reduction Achieved via RLS Aggregation

IV. Resource List Server Traffic Reduction

This work is being carried out as an experimental verification of the known concepts which led to the definition and specification of the RLS in literature [15], [17]. While the concept and operation of the RLS have been described, the performance of the RLS under varying operational conditions has not been widely tested.

Simulations of simplified aggregation scenarios carried out demonstrate the effect aggregation has on the efficiency of IMS presence signalling in the PUA access network.

The simulation presented in this paper examined the effect of time-based RLS aggregation for a stabilized, steady-state RLS subscription. The aggregation parameters in the simulation were held constant, while the size of the resource list varied. The duration of each iteration in the simulation was set at one thousand seconds, with an aggregation time limit of five seconds between NOTIFY messages sent to the PUA. After each iteration, the size of the resource list was increased by one, starting from one watched Presentity, up to one hundred Presentities. In order to model a group of dynamic presentities operating in real time, the list of Presentities in the simulation was split into two groups. Fifteen percent of the list were termed "very active" and given a probability of generating an update per second of 0.01. The remaining eighty-five percent of the group were "active", but with a lower presence information update probability of 0.0001 per second.

This provided both aggregated and non-aggregated "activity data" for a range of resource list sizes. Examples of basic XML presence information and SIP packets were captured while using the UCT IMS Client [18] and the OpenSER SIP Presence Server [19] as the PUA and application server respectively. The IMS signalling between the two entities was carried over the Frauenhofer FOKUS OpenIMSCore [20]. The presence payload sizes and SIP packet overhead obtained from the above system were added to the activity data to correctly weight the figures obtained. As the subscription is operating in a steady state, the amount of presence information generated per presentity is considerably smaller than the respective full Presentity, causing the size of the packet overhead to become significant.

The data obtained is graphed in Figure 5, with the aggregated packet flow illustrated by the red dashed line. The effect of the aggregation smooths the packet flow from the RLS to the PUA, and significantly reduces the volume of data transmitted as the resource list size increases. This occurs because the larger numbers of presence payloads per NOTIFY message sent reduces the packet overhead significantly.

The reduction of the presence-related traffic will have the largest impact on low-bandwidth access networks such as GPRS. As an enhancement of GSM, GPRS has much wider coverage than newer technologies such as UMTS 3G networks, especially out of heavily populated areas. Large amounts of presence information being delivered to the PUA will restrict the bandwidth available to other applications, many of which may be of a higher priority than the presence information being delivered.

A testbed implementation of an RLS is being developed currently, using Java and SIP Servlets. Methods for time-based, quantity-based and MTU-based aggregation are being included, as well as a combination of time- and quantity-based aggregation. The parameters controlling aggregation are configurable at run time, allowing simple configuration for testing different aggregation conditions or parameters.

V. Conclusions and Future Work

The IMS is a converged service delivery network which allows operators to rapidly deliver services to subscribers across a myriad of access networks. Presence is a basic network service applicable over all access networks and end terminals, likely to form an integral part of the IMS service offering. It is imperative that the presence framework adopted in the IMS adheres to recognized standards in order to maintain an acceptable level of interoperability. The presence framework for the IMS as it stands is relatively complete, but certain important challenges still remain.

This paper presents an examination of the IMS signalling with regards to the presence service. The data format of presence information is also examined, and the relative inefficiencies of both the data formatting and IMS signalling with regards to the abilities and resources of the access network described.

The proposal of the RLS as a means of reducing the amount of presence-related signalling that passes through the access network, and removing it to the IMS core, is shown to be highly effective. Furthermore, various methods of implementing aggregation are examined, and their efficacy when operating under different subscription states.

A simulation of RLS aggregation operating with increasing resource list lengths is presented, and the data obtained analysed and presented across the range of resource list lengths. This served to demonstrate the increased efficiency and efficacy of the RLS aggregation, as the amount of presence information dealt with per subscriber increased.

The development of a testbed implementation with which to test the aggregation methods is ongoing. This implementation will test the operation of all the methods described, with various parameters such as the subscription state, amount of information per NOTIFY received, the size of the resource list, and aggregation method parameters being adjusted. These results will be published when available. All software developed for the RLS is open source and freely available to all.

REFERENCES

[1] N. Blum and T. Magedanz, "PTT + IMS = PTM - Toward Community/Presence-based IMS Multimedia Services," in *International Symposium on Multimedia*, IEEE Computer Society, 2005.

[2] Ericsson AB, "IMS - IP Multimedia Subsystem. The value of using the IMS architecture," 2004.

[3] G. Camarillo and M. A. Garcia-Martin, *The 3G IP Multimedia Subsystem (IMS) - Merging the Internet and the Cellular Worlds*, ch. 17, pp. 323–328. John Wiley and Sons, Ltd, 2nd ed., 2006.

[4] M. Day, S. Aggarwal, G. Mohr, and J. Vincent, "RFC 2779, Instant Messaging / Presence Protocol Requirements," 2000.

[5] J. Peterson, "RFC 3859, Common Profile for Presence (CPP)," 2004.

[6] G. Camarillo and M. A. Garcia-Martin, *The 3G IP Multimedia Subsystem (IMS) - Merging the Internet and the Cellular Worlds*, ch. 17, pp. 323–328. John Wiley and Sons, Ltd, 2nd ed., 2006.

[7] M. Day, J. Rosenberg, and H. Sugano, "RFC 2778, A Model for Presence and Instant Messaging," 2000.

[8] H. Sugano, S. Fujimoto, G. Klyne, A. Bateman, W.Carr, and J. Peterson, "RFC 3863, Presence Information Data Format (PIDF)," 2004.

[9] H. Schulzrinne, V. Gurbani, P. Kyzivat, and J. Rosenberg, "RFC 4480, RIPD: Rich Presence Extensions to the Presence Information Data Format (PIDF)," 2006.

[10] 3rd Generation Partnership Project, "Technical Specification Group Services and System Aspects; Presence Service Stage 1 (Release 7)," 2005.

[11] J. Rosenberg, "RFC 3858, An Extensible Markup Language (XML) Based Format for Watcher Information," 2004.

[12] 3rd Generation Partnership Project, "Technical Specification Group Services and System Aspects; Presence Service; Architecture and Functional Description (Release 7)," 2006.

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

[14] G. Camarillo, "RFC 3486, Compressing the Session Initiation Protocol (SIP)," 2003.

[15] A. B. Roach, B. Campbell, and J. Rosenberg, "RFC 4662, A Session Initiation Protocol (SIP) Event List Notification Extension for Resource Lists," 2006.

[16] J. Rosenberg, "RFC 4826, Extensible Markup Language (XML) Formats for Representing Resource Lists," 2007.

[17] 3rd Generation Partnership Project, "Technical Specification Group Core Network and Terminals; Presence Service Using the IP Multimedia (IM) Core Network (CN) Subsystem; Stage 3 (Release 7)," 2006.

[18] "UCT IMS Client." http://uctimsclient.berlios.de.

[19] "Open Source SIP Server." http://www.openser.org.

[20] "Open Source IMS Core." http://www.openimscore.org.

M ichael Pitman obtained his BSc in Electrical and Computer Engineering at the University of Cape Town in 2006. He commenced his MSc in the Communications Research Group in 2007. His research interests are IMS, Presence, and IMS Presence Architectures. He is an Ubuntu Linux enthusiast and recently migrated from C and eXosip to Java SIP Servlets and Apache Tomcat for Web/SIP server implementations.