Cost Analysis of the IMS Presence Service

M. T. Alam, Z. D. Wu
School of IT, Bond University
malam@bond.edu.au

Abstract

IMS (IP Multimedia Subsystem) is the technology that will merge the Internet (packet switching) with the cellular world (circuit switching). Presence is one of the basic services which is likely to become omnipresent in IMS (IP Multimedia Subsystem). It is the service that allows a user to be informed about the reachability, availability, and willingness of communication of another user. The flow of messages will be massive for large amount of publishers and watchers joining an IMS system, because of the security architecture of the IMS. Although the IETF engineers have proposed several solutions to reduce the signalling overhead to facilitate the presence service, the heavy traffic flows have been compromised with several factors like real time view and information segregation etc. In this paper, we propose a mathematical model to analyse the system-performance of the IMS presence service during heavy traffic. The model derives the cost functions that are based on the real parameters of the Presence server. Simulation results have been shown that provide useful insight into the system behaviour.

1. Introduction

IP Multimedia Subsystem (IMS) is a new framework, basically specified for mobile networks, for providing Internet Protocol (IP) telecommunication services [1]. It is the technology that will merge the Internet (packet switching) with the cellular world (circuit switching). Presence is one of the basic services that is likely to become omnipresent in IMS. It is the service that allows a user to be informed about the reachability, availability, and willingness of communication of another user. The presence service is able to indicate whether other users are online or not and if they are online, whether they are idle or busy.

The presence framework defines various roles as shown in figure 1. The person who is providing presence information to the presence service is called a presence entity, or for short a presentity. In the figure, Alice plays the role of a presentity. The presentity is supplying presence information such as status, capabilities, communication address etc. A given presentity has several devices known as Presence User Agents (PUA) which provide information about her

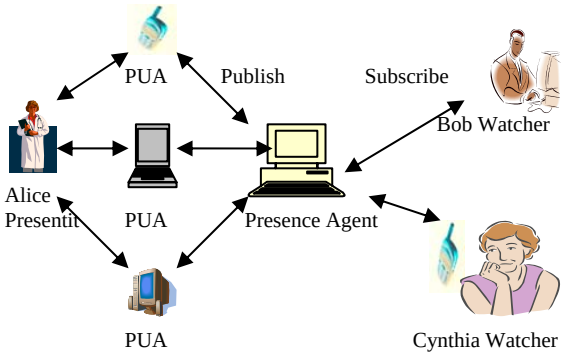


Figure 1-SIP Presence Architecture

presence. All PUAs send their pieces of information to a presence agent (PA). A presence Agent can be an integral part of a Presence Server (PS). A PS is a functional entity that acts as either a PA or as a proxy server for SUBSCRIBE requests. Figure 1 also shows two watchers: Bob and Cynthia. A watcher is an entity that requests (from the PA) presence information about a presentity or watcher information about his/her watchers. A subscribed watcher asks to be notified about future changes in the presentity's presence information, so that the subscribed watcher has an updated view of the presentity's presence information.

3GPP defined in 3GPP TS 23.141 [2] provides the architecture to support the presence service in the IMS. The used interfaces for this service are *Pen, Pw, Pi, Px, Ut* and the protocols used are SIP (Session Initiation Protocol), Diameter and XCAP (XML Configuration Access Protocol) as described by the IMS technical specification. The watcher subscription is done in number of steps. The watcher application residing in the IMS terminal sends a SUBSCRIBE request (1) addressed to her list for example sip:alice-list@home1.net. The request (2) is received at the S-CSCF (Serving Call/Session Control Function), which evaluates the initial filter criteria. One of those criteria indicates that the request (3) ought to be forwarded to an Application Server that happens to be an RLS (Resource List Server). A RLS can be implemented as an Application Server in IMS. The RLS, after verifying the identity of the subscriber and authorizing the subscription, sends a 200 (OK) response (4). The RLS also sends a notify request, although it does not contain any presence information at this stage. The RLS then subscribes one by one to all the presentities listed in the resource list and, when enough

information has been received, generates another NOTIFY request that includes a presence document with the aggregated presence information received from the presentities' PUAs.

The above IMS presence architecture indicates that the flow of messages will be massive for large amount of publishers and watchers joining an IMS system. A watcher should not be able to watch infinite amount of time to its presentities when the PS encounters heavy traffic. Since, every time a presentity changes position, its watcher will have to be notified, particularly with IMS any message generated from an IMS terminal will have to travel through several nodes for AAA (Authentication, Authorization and Accounting) functionality. Thus a model needs to be designed carefully to analyse the performance of the system during heavy traffic. In this paper we propose a mathematical model to analyse the performance of the IMS presence service.

The paper is structured as follows. Section 2 shows the related work and section 3 proposes an analytical model to analyse the cost of the presence service in IMS framework. Section 4 shows the simulation results for the proposed model. Section 5 concludes the paper with the outline of future work regarding the derived model.

2. Background and Related Work

SIP (Session Initiation Protocol) has been chosen to be used in IMS to play the key role for setting up the session while inter-working with other protocols. RFC 3265 [12] defines a framework for event notification in SIP. According to this document, the entity interested in the status information of a resource subscribes to that information. The entity that keeps track of the resource state will send a NOTIFY request with the current status information of the resource and a new NOTIFY request every time the status changes. A watcher receives NOTIFY message every time any of its presentity changes state. Although the event notification framework offers powerful tool in the IMS presence service that allows a watcher to be informed about changes in the state of a presentity, in some situations the amount of information that the Presence Server has to process might be large. Imagine, for instance, IMS presentities of a watcher are driving on highways. The corresponding IMS watcher will get very frequent updates, because of the rapid geographical change of the presentities.

The Presence Information Data Format (PIDF) is a protocol-agnostic document that is designed to carry the semantics of presence information across two presence entities. The PIDF is specified in the Internet-Draft "Presence Information Data Format (PIDF)" [4]. The PIDF encodes the presence information in an XML (Entensible Mark-up Language) document that can be transported, like any other MIME (Multipurpose Internet Mail Extension) document, in presence publication (PUBLISH transaction) and presence subscription/notification (SUBSCRIBE/NOTIFY transaction) operations. The Rich Presence Information Data Format (RPID) is an extension to the PIDF that allows a presentity to express detailed and rich presence information to his/her watchers. Like the PIDF, RPID is encoded in XML. The RPID extension is specified in [5]. Every time a watcher wants to subscribe to the presence information of a presentity, the watcher needs to exchange a SUBSCRIBE transaction and a NOTIFY transaction with the presentity's PUA, just to set up the subscription. Obviously, this mechanism does not scale well, particularly in wireless environment. In order to solve this problem the IETF has created a number of concepts as described below.

1. The concept of resource lists is one of the mechanisms to reduce excessive signals. A resource list is a list of SIP URIs that is stored in a new functional entity called the Resource List Server (RLS), sometimes known as an exploder for SUBSCRIBE requests. A SIP exploder receives a request from a user agent and forwards it to multiple users. SIP exploders used for subscriptions are described in [6].

Instead of sending a SUBSCRIBE request to every user in the presence list, a watcher in these type of systems, sends a single SUBSCRIBE request addressed to its presence list. The request is received by the SIP exploder, or RLS. The exploder sends a request to every user in the list. Later when the exploder receives the NOTIFY requests from them, it aggregates the presence information and sends a single NOTIFY request to the watcher. Although the mechanism saves bandwidth on a user's access network; the signalling impact is still there for massive number of publishers and watchers. Moreover, the PIDF/RPID documents are naturally large because they are rich in information. A watcher who is subscribed to a number of presentities may get one of these XML documents every time the presentity's presence information changes. When presence information reaches a small device that has constraints in memory, processing capabilities, battery lifetime and available bandwidth, the device may be overwhelmed by the large amount of information and might not be able to acquire or process it in real time.

2. Partial notification is another mechanism on which IETF engineers are working to reduce the amount of presence information transmitted to watchers. A weight or preference is indicated through a SUBSCRIBE request. The mechanism defines a new XML body that is able to transport partial or full state. Again, frequency of notification is reduced at the cost of information transmitted.

3. Event-throttling mechanism allows a subscriber to an event package to indicate the minimum period of time between two consecutive notifications. So, if the state changes rapidly, the notifier holds those notifications until

the throttling timer has expired, at which point the notifier sends a single notifications to the subscriber. However, with this mechanism the watcher does not have a real-time view of the subscription state information.

4. Compression of SIP messages is another technique to minimize the amount of data sent on low-bandwidth access. RFC 3486 [8], RFC 3320 [10], RFC 3321 [9] defines signalling compression mechanisms. Usually these algorithms substitute words with letters. The compressor builds a dictionary that maps the long expressions to short pointers and sends this dictionary to the de-compressor. However, the frequency of data transmission is not reduced in such techniques.

Although, the abovementioned works try to reduce the signalling overhead, they fail to draw attention to the behaviour of the system during heavy traffic. We propose a mathematical model next that will provide useful insight into the IMS presence service in terms of cost functions.

3. The Analytical Model

The model is divided into three stages: 1) finding the steady state probability vector of a presentity movement, 2) finding the signalling cost and 3) cost function in general to analyse the system performance.

Let the number of states for a presentity to change be arbitrary. The presentity can hop among any state from its initial state with arbitrary probability. However, the probabilities of coming back to its initial state are equivalent. The scenario is depicted in figure 2. We assume that state zero is the initial position of a presentity which may be thought of its actual residing position. The other states may represent the presentity's state change to busy, idle, not available etc. or even the location change for instance, availability in office etc. We also assume that the presentity initial state is saturated so that upon completion of one state change, it will enter to another statically identical state instantaneously. The probability of staying at state zero is q_0 and $\sum_{i=0}^{m} q_i = 1$.

With these assumptions, the system can be modelled as a discrete-parameter Markov chain. The transition probability matrix P of the Markov chain is given by:

$$\begin{vmatrix} q_0 & q_1 & \dots & q_m \\ 1 & 0 & \dots & 0 \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ 1 & 0 & 0\dots0 & 0 \end{vmatrix}$$

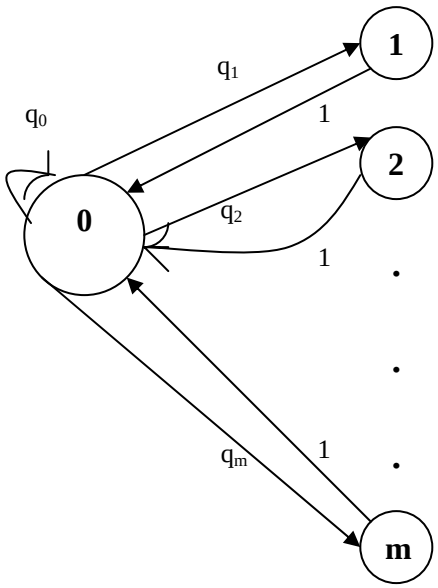


Figure 2-Markov chain for a Presenity's states

Since, $0<q_i<1 (i=0,1,\dots,m)$, then this finite Markov chain is both irreducible and aperiodic. The unique steady-state probability vector, $\boldsymbol{v}$, is obtained by solving the system of linear equations:

$$\boldsymbol{v} = \boldsymbol{v}P$$

or

$$v_0 = v_0 q_0 + \sum_{j=1}^{m} v_j, \qquad j = 0, \qquad (1)$$

$$v_j = v_0 q_j, \qquad j = 1,2,\dots,m. \qquad (2)$$

Using the normalization condition:

$$\sum_{j=0}^{m} v_j = 1,$$

We have (appendix A):

$$v_0 + v_0 \sum_{j=1}^{m} q_j = 1. \qquad (3)$$

Substituting $\sum_{j=1}^{m} q_j = 1 - q_0$, in equation (3), we get:

$$v_0(1+1-q_0) = 1$$

$$\Rightarrow \qquad v_0 = \frac{1}{2-q_0}, \quad j = 0,$$

From (2) we get, $v_j = \frac{q_j}{2-q_0}, \qquad j = 1,2,\dots,m. \quad (4)$

v_j, as defined above is the visit to state j of a presentity. Let r be the total number of IMS presentities observed by the IMS watchers via a Presence server in the system.

Therefore, the total average cost of presentities' movement for a Presence Server in a real-time interval *t*, in the long run is:

$$C_p = t\sum_r \frac{q_j}{(2-q_0)} \quad (5)$$

Where, both q_j and q_0 are variable for each *r*.

We recall that a state change at the presentity side will require a NOTIFY message to the watcher to be generated. We use the following delays as shown in [11] to establish the cost of a NOTIFY message, C_n via P-CSCF (Proxy-CSCF) at each hop. The parameters used are denoted as follows:

$\lambda_{i1,}$ *i=1,2,…n:* Notify message arrival rate at hop *i,*
$\lambda_{i2,}$ *i=1,2,…n:* arrival rate of messages other than Notify at hop *i,*
$\mu_{i1,}$ *i=1,2,…n:* serving rate for each Notify message in hop *i,*
$\mu_{i2,}$ *i=1,2,…n:* serving rate for messages other than Notify at hop *i,*
$\rho_{i1,}$ *i=1,2,…n:* load at hop *i* for Notify messages,
$\rho_{i2,}$ *i=1,2,…n:* load at hop *i* for messages other than Notify,
where $\rho_i = \lambda_i / \mu_i$, $\lambda_i < \mu_i$,
D_{1i}*:* the processing delay at hop *i,*
D_2*:* the propagation / internet transmission delay at each hop,
D_{3i}*:* the queuing delay at hop *i,*

We define $D_{1i} = \frac{1}{(\mu_{i1} - \lambda_{i1})}$

$$D_{3i} = \frac{\frac{1}{\mu_{i1}}(1-\rho_{i1}-\rho_{i2}) + R}{(1-\rho_{i1})(1-\rho_{i1}-\rho_{i2})} \quad \text{from [21]}$$

Where $R = \frac{\lambda_{i1}X_1^2 + \lambda_{i2}X_2^2}{2}$; X_1^2, X_2^2 are the second moments of $\mu_{i1}$and μ_{i2} respectively.

The propagation / internet transmission delay at each hop is considered to be a constant, $D_2 = \Delta$.

The cost of sending a NOTIFY message, C_n is measured as the sum of delays at each node that is involved to send the message between an IMS presentity and a watcher.

$$C_n = \sum_i [D_{1i} + D_{2i} + D_{3i}], \qquad i = 1,2,3...$$

Where, *i* is the number of hops. Assuming M/M/1 system at the PS, the expected number of watchers in the server is given by (appendix B):

$$E(X) = \frac{\rho_w}{1-\rho_w},$$

Where,

Traffic intensity, $\rho_w = \frac{\lambda_w}{\mu_w}$

$\lambda_{w:}$ Average watcher arrival rate (poisson), equivalently the watcher inter-arrival times are exponentially distributed with mean $\frac{1}{\lambda_w}$.

$\mu_{w:}$ Watcher service times are independent identically distributed random variables, equivalently the distribution being exponential with mean $\frac{1}{\mu_w}$.

It is noted that a Presentity may subscribe as a watcher and a presentity may be watched by several watchers.

Thus, the total average cost of sending NOTIFY messages, C_T for each watcher in the system (PS) is:

$$C_T = \frac{\sum_{x=1}^{r} N_x (v_{jx} C_{nx})}{E(X)} \quad (6)$$

Where, N_x is the number of NOTIFY messages generated for a state change of the x^{th} presentity to notify its watchers.

Let, *R* is the presentity subscription rate in the system and C_{S_K} is the subscription cost for the k^{th} presentity in the system. C_{S_K} can be derived similar to C_n.

Therefore, this total average cost (in terms of delays) in a real-time interval *T* is:

$$C(t) = \int_0^T \{(\sum_{x=1}^{r} N_x v_{jx} C_{nx}) + C_{S_K} R\} dt$$

$$\approx T(\sum_{x=1}^{r} N_x v_{jx} C_{nx}) + TC_{S_K} R \quad (7)$$

Note that both *r* and N_x are function of *R* i.e., the number of presentities and NOTIFY messages generated in the system at any period of time dynamically depend on the presentity subscription rate of the watchers. The presentities subscribed may be overlapped by watchers; in that case, the *r* will not vary but N_x will. Since we are only interested in the increasing traffic in this model, we do not address the issue of presentity un-subscriptions/deletions by the watchers.

4. Simulation Results

If the watcher subscription rate is constant, then $Y = tC_{S_K} R$ behaves like a straight line. Thus we centre our simulation over (6). BRITE was used to generate the mobile environment after every interval in a fixed area. Figure 3 shows total cost against number of watchers. The watcher was varied from 50 to 200 for 500 presentities. The message

costs were kept constant. We define the transition probability, $q_j = \frac{1-q_0}{j}$ where j is number of states and kept q_j as 0.06, 0.09 and 0.17. All costs go down as the number of watchers increases where the lower transition probability costs reduce to similar values for large number of watchers.

Next we varied r for 500 watchers and transition probability vector 0.2. The message costs were generated randomly. The total cost was found to behave linearly (see figure 4).

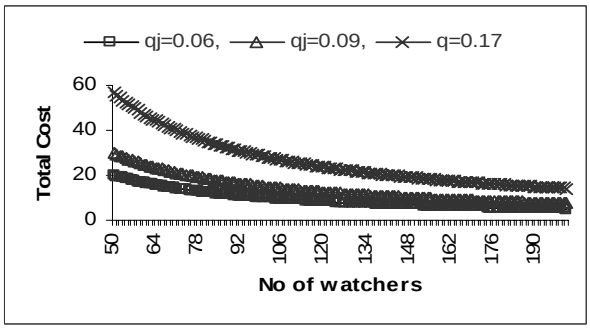


Figure 3-Cost for large watchers

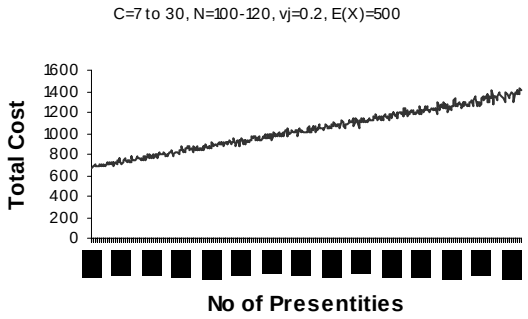


Figure 4-Cost for large presentities

In figure 5, the steady state probability was varied from 0.01 to 0.2 for 500 watchers. The number of messages was randomly generated. The cost goes up slowly with spikes.

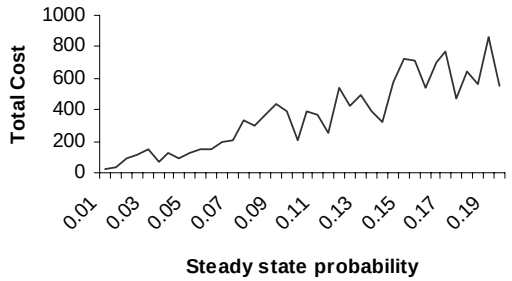


Figure 5-Cost for growing steady state vector

5. Conclusions and Future Work

It is recognised that a framework for reducing load is essential in IMS for large scale of traffic to facilitate presence service to the terminals. We have proposed an analytical model in this paper to analyse the performance of the IMS presence service. By using this model, the cost related to the traffic at the Presence Server can be estimated. Our future work will be to propose an optimal time solution for the Presence server to provide the joining IMS watchers to reduce the signalling overhead.

The Timed Presence extension is specified in the Internet-Draft "Timed Presence Extension to the Presence Information Data Format (PIDF) to indicate Presence Information for Past and Future Time Intervals" [7] and allows a presentity to express what they are going to be doing in the immediate future or actions that took place in the near past. A timed-status element that contains information about the starting time of the event is added to the PIDF XML document. The starting time of the event is encoded in a 'from' attribute, whereas an optional 'until' (alternatively Presence Server may supply this time extension to the watcher) attribute indicates the time when the event will stop. Figure 6 shows an example of the time status extension. Here, Alice is publishing that she will be offline from 13:00 to 15:00.

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pdf"
          xmlns:ts="urn:ietf:params:xml:ns:pidf:timed-
            status"
          entity="pres:alice@example.com">

 <tuple id="qoica32">
   <status>
         <basic>open</basic>
   </status>
   <ts:timed-status from="2004-02-
         15T13:00:00.000+02:00"
         Until="2004-02-15T15:00:00.000+02:00">
         <basic>closed</basic>
   </ts:timed-status>
  <contact>sip:alice@example.com</contact>
 </tuple>
</presence>
```

Figure 6: Example of the timed status extension

The constant time set may create bottleneck because of excessive message flow in the network. Specially, if an IMS watcher watches many presentities and if the watcher-subscription-time is not set carefully, it will be notified any changes made in its presentity list. Both long and short life time will introduce overhead in number of messages and

cache respectively. Our future work will be to derive an optimal procedure using the proposed cost functions we have defined in this research to set the timer of the watcher subscription life time for the IMS terminals.

Appendix A

$$\sum_{j=1}^{m} v_j = \sum_{j=0}^{m} v_j - v_0$$

$$\Rightarrow \quad \sum_{j=1}^{m} v_j = 1 - v_0$$

$$v_0 = v_0 q_0 + \sum_{j=1}^{m} v_j$$

$$\Rightarrow \quad v_0 = v_0 q_0 + 1 - v_0$$

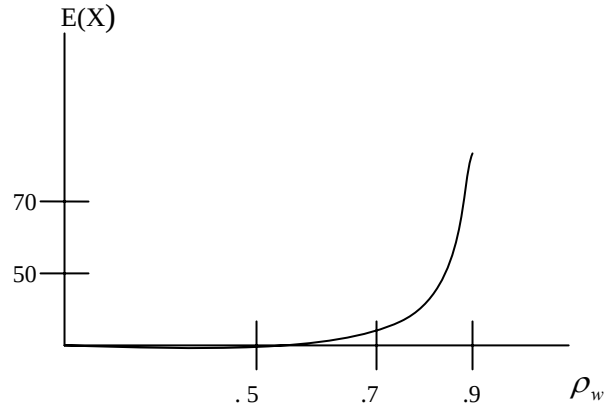
$$\Rightarrow \quad 2v_0 - v_0 q_0 = 1$$

$$\Rightarrow \quad 2v_0 - v_0 (1 - \sum_{j=1}^{m} q_j) = 1, \quad [q_0 = 1 - \sum_{j=1}^{m} q_j]$$

$$\Rightarrow \quad v_0 + v_0 \sum_{j=1}^{m} q_j = 1$$

Appendix B

The following figure shows the behaviour of expected number of watchers in the PS versus traffic intensity assuming that the PS is not idle when there are watchers waiting to subscribe.



References

[1] 3GPP TSG SSA, IP Multimedia Subsystem (IMS) – Stage 2 (Release 6), TS 23.228 v. 6.6.0, (2004-06)

[2] 3GPP. Presence service; Architecture and functional description; Stage 2. TR 23.141, 3rd Generation Partnership Project (3GPP)

[3] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler (2002). "SIP: Session Initiation Protocol". *Internet Engineering Task Force,* Internet Draft: http://www.ietf.org/proceedings/02mar/I-D/draft-ietf-sip-rfc2543bis-09.txt. (Work on progress)

[4] H. Sugano, S. Fujimoto, G. Klyne, A. Bateman, W. Carr, and J. Peterson (2003). "Presence Information Data Format (PIDF)." Internet-Draft draft-ietf-imppcpim-pidf-08, *Internet Engineering Task Force*, May 2003. work in progress.

[5] H. Schulzrinne (2004). "RPID – Rich Presence Extensions to the Presence Information Data Format (PIDF)." Internet-Draft draft-ietf-simple-rpid-03, *Internet Engineering Task Force*, March 2004. Work in progress

[6] A. B. Roach, J. Rosenberg, and B. Campbell (2003). "A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists." Internet-Draft draft-ietf-simple-event-list-04, *Internet Engineering Task Force*, June 2003. Work in progress

[7] H. Schulzrinne (2004). "Timed Presence Extension to the Presence Information Data Format (PIDF) to indicate Presence Information for Past and Future Time Intervals." Internet-Draft draft-ietf-simple-future-01, *Internet Engineering Task Force,* March 2004. Work in progress

[8] G. Camarillo (2003). "Compressing the Session Initiation Protocol (SIP)." RFC 3486, *Internet Engineering Task Force*, February 2003

[9] H. Hannu, J. Christoffersson, S. Forsgren, K.-C. Leung, Z. Liu, and R. Price (2003). "Signalling Compression (SigComp) – Extended Operations." RFC 3321, *Internet Engineering Task Force*, January 2003

[10] H. Hannu, J. Rosenberg, C. Bormann, J. Christoffersson, Z. Liu, and R. Price (2003). "Signalling Compression (SigComp)." RFC 3320, *Internet Engineering Task Force*, January 2003

[11] M. T. Alam, Z. D. Wu (2005). "Comparison of Session Establishment Schemes over IMS in Mobile Environment." *Fifth IEEE International Conference on Information, Communications and Signal Processing (ICICS 2005),* December 6-9, Bangkok, Thailand (to be presented)

[12] A. Roach (2002). "Session Initiation Protocol (SIP)-Specific Event Notification." RFC 3265, *Internet Engineering Task Force* (June 2002)