IBM Research Report

Peer-to-Peer Instant Messaging and Presence Services over Wireless Ad Hoc Networks

Nilanjan Banerjee*, Arup Acharya, Sajal Das*
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

*CReWMaN
University of Texas at Arlington
Arlingtown, TX

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Peer-to-peer Instant Messaging and Presence Services over Wireless Ad Hoc Networks

Nilanjan Banerjee

CReWMaN, University of Texas at Arlington

Arup Acharya

IBM T. J. Watson Research Center

Sajal K. Das

CReWMaN, University of Texas at Arlington

Abstract— Session Initiation Protocol (SIP) has been extended by IETF SIMPLE Working Group to enhance the basic protocol with Instant Messaging and Presence functionalities. These extensions are, however, dependent on the Internet and SIP infrastructure. Ad hoc networks being devoid of any such infrastructure, require auxiliary mechanisms to support such extensions. In this paper, we present a brief overview of the SIP extensions followed by a proposal to integrate a cluster based ad hoc routing protocol with the instant messaging and presence services.

Index Terms— Instant Messaging, Presence, Session Initiation Protocol, Ad Hoc Networks

I. INTRODUCTION

Instant Messaging and Presence Applications (IMP) have gained tremendous popularity over the last few years and contribute a significant amount of Internet traffic today. A presence system allows users to subscribe to each other and be notified of changes in state. Instant Messaging (IM), on the other hand, is defined as the exchange of content between a set of participants in near real time. The IMP applications are mostly based on proprietary IMP platforms and no common standard exists. As a result the IMP applications are not inter-operable, thus restricting the users to particular vendors only. Several working groups have been chartered at IETF to solve this problem. The IMPP (Instant Messaging and Presence Protocol) WG [1] defines the protocol requirements and the data format for building an Internet-wide scalable IMP system. Other WG leverage this framework for building their own IMP system. For example, the XMPP (Extensible Messaging and Presence Protocol) WG [3] defines an open XML-based protocol for extensible IMP applications. On the other hand, SIM-PLE (SIP for Instant Messaging and Presence Leveraging

Extensions) WG [2] has built their IMP platform on top of session Initiation Protocol (SIP) [14]. SIMPLE WG has proposed extensions to the SIP protocol, enabling it to exchange instant messages inside a SIP session and an event package mechanism for notification of presence information.

The advantages of SIP that is gaining favor for SIM-PLE WG are as follows. First, SIP is a matured, widely deployed and thoroughly tested protocol to serve as the platform for IMP applications. Second, SIP provides different types of mobility support and hence is suitable for providing IMP services in mobile devices. Third, SIP was designed to provide several services which are fundamentally similar to IMP services. Thus SIP could be reused or extended without any drastic change in the standard.

The SIMPLE framework follows the same mechanism as specified in the SIP standard and uses the Internet (e.g. DNS resolution) and SIP proxy based overlay infrastructure for SIP service discovery and the routing of the IMP messages. But this infrastructure is not available in ad hoc networks. Hence some auxiliary mechanism is needed to (i) discover the IMP services in the ad hoc networks and (ii) to route the SIMPLE messages.

In this paper, we discuss how to enable SIMPLE procedures for IMP applications in infrastructure-less ad hoc networks. An integrated approach where the SIMPLE procedures are integrated with a cluster based routing protocol has been proposed. A caching scheme leveraging the SIP call features has also been proposed. Some IEEE 802.11 MAC layer enhancements have also been discussed to increase the efficiency of IM services. The rest of the paper is organized as follows. Section II provides a brief overview of SIP. The extensions of SIP for instant messaging and presence services have been described in Section III. The issues related to the deployment of SIMPLE in ad hoc networks are discussed in Section IV. The proposed integration of SIMPLE

procedures with the routing protocol is presented in Section V. Section VI concludes the paper.

II. OVERVIEW OF SIP

SIP is a control protocol that allows creation, modification and termination of sessions with one or more participants. SIP is used for voice and video calls either for point-to-point or multiparty sessions. It is independent of the media transport which for example, typically uses Real-time Transport Protocol (RTP) over UDP [15]. It allows multiple end-points to establish media sessions with each other: this includes terminating the session, locating the end-points, establishing the session and then, after the media session has been completed. In recent times, SIP has gained widespread acceptance and deployment among wireline service providers for introducing new services such as VoIP; within the enterprises for Instant Messaging and collaboration; and amongst mobile carriers for push-to-talk service. Industry acceptance of SIP as the protocol of choice for converged communications over IP networks is thus highly likely. As shown in Figure 1, a SIP infrastructure consists of user agents, registration servers, location servers and SIP proxies deployed across a network. A user agent is a SIP endpoint that identifies services such as controlling session setup and media transfer. User agents are identified by SIP URIs (Uniform Resource Identifier), which is a unique HTTP-like URI of the form `sip:user@domain`. All user agents REGISTER its IP address with a SIP registrar server (which can be co-located with a SIP proxy). Details of the SIP protocol can be found in [14]. SIP defines a set of messages, such as INVITE, REFER etc., to setup sessions between user agents. These messages are routed through SIP proxies that are deployed in the network. DNS Service records help in finding SIP proxies responsible for the destination domain.
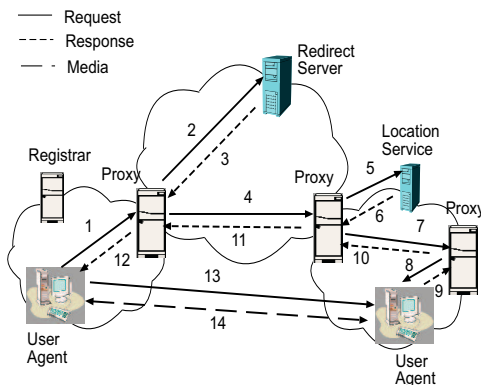

Fig. 1.   SIP architecture

A session is setup between two user agents following a client-server interaction model, where the requesting user agent acts as the client and is known as the user agent client (UAC), interacting with the target user agent known as the user agent server (UAS) acting as server. All requests from an originating UAC, such as an INVITE are routed by the proxy to an appropriate target UAS, based on the target SIP URI included in the `Request-URI` field of the INVITE message. Proxies may query location and redirect servers for SIP service discovery or in order to determine the current bindings of the SIP URI. Signaling messages are exchanged between user agents, proxies and redirect/location servers to locate the appropriate services or endpoints for media exchange. For reasons of scalability, multiple proxies are used to distribute the signaling load [9]. A session is setup between two user agents through SIP signaling messages comprising of an INVITE (messages 1,2,4,7, and 8 in Figure 1), an OK response (messages 9-12 in Figure 1) and an ACK (message 13 in Figure 1) to the response [14]. The call setup is followed by media exchange using RTP. The session is torn down through an exchange of BYE and OK messages.

SIP distinguishes between the process of session establishment and the actual session. A basic tenet of SIP is the separation of signaling (control) from media. Signaling messages are usually routed through the proxies while the media path is end-to-end. The session setup messages like INVITE contain user parameters using Session Description Protocol (SDP) [8] in the message body. SDP provides information about the session such as parameters for media type, transport protocol, IP addresses and port numbers of endpoints. The IP address and port numbers exchanged through SDP is used for the actual data transmission (media path) for the session. Any of these parameters can be changed during an ongoing session through a RE-INVITE message, which is identical to the INVITE message except that it can occur within an existing session.

III. SIMPLE: SIP FOR INSTANT MESSAGING AND PRESENCE LEVERAGING EXTENSIONS

According to the definition, an IMP system allows users to *subscribe* to each other and be *notified* of changes in state, and for users to send each other short instant messages in real-time.

Typically, in a presence service entities willing to receive the presence information of a given entity subscribe to the "presence service" of that entity so that they can be notified when a presence event related to that entity, e.g. coming on-line, occurs. The presence service is a system that accepts, stores and distributes the presence information to the interested parties.

SIP provides a general framework for event notifications [10], whose purpose is to allow SIP endpoints to receive notifications from remote endpoints indicating that an event has occurred. The framework does not define the nature of the events causing a notification, but is intended to be the general support on which specific events (event packages) can be built. Two SIP methods are used in the framework, SUBSCRIBE and NOTIFY. SIMPLE WG has defined in [13] the presence event package, with which SIP entities can subscribe to the presence service of a remote entity and be notified when a presence related event (e.g. entity goes on-line) occurs.

When a SIP entity (subscriber or watcher) wants to subscribe to the presence service of a remote SIP entity (presentity), also known as the Presence User Agent (PUA), it creates a SUBSCRIBE request, carrying the URI of the desired entity. The request traverses normally the SIP network (it passes through chain of proxies as the other requests) until it reaches a SIP presence server, which will generate a response for the SUBSCRIBE request.

The presence agent (PA) is the logical entity in charge of managing the presence information of a presentity, processing SUBSCRIBE requests, consequently notifying to the subscriber changes in the presence status of the presentity, with NOTIFY requests. The PA for a given presentity must have access to the presence information of that presentity. The way how this is achieved is out of scope of the SIMPLE framework. Upon authentication and authorization of the subscription, a PA sends a NOTIFY message to the subscriber including the presence information and whether the request was authorized. According to the definition given in [9], both SUBSCRIBE and NOTIFY methods are considered as SIP requests.

Fig. 2.   SIMPLE Presence Event Package

The message exchange is reported in Figure 2. In this example, we suppose that the PA and the PUA are not co-located. The way how PA and PUA exchange information is out of the scope of the Presence Event Package specification.

The PUA has already communicated to the PA the presence information of its presentity. At some point in the time, a subscriber sends a SUBSCRIBE request to the presentity. The presentity is identified by the URI carried in the SUBSCRIBE request. The request is routed in the SIP network and arrives to the PA of the queried presentity. The PA acknowledges the request and starts the authentication and authorization procedures of the subscriber (unless the subscriber had been previously authenticated and authorized).

After the, successful, completion of this step, the PA sends the NOTIFY request containing the presence document of the presentity. The subscriber acknowledges the message with a 200 OK SIP response. Afterwards, the PUA sends to the PA an update of the presence information of the presentity. Since the subscriber is regularly authorized, the PA delivers to the subscriber another NOTIFY message, containing the whole presence document of the presentity.

The SIP protocol has been extended for sending instant messages [6]. When one user wishes to send an instant message to another, the sender formulates and issues a SIP request using the MESSAGE method defined as an extension. MESSAGE requests normally carry the instant message content in the request body. The request message is routed based on the Request-URI of the request header, which contains the current information about the recipient's location. Like any other SIP request message, the MESSAGE request traverses a set of SIP proxies, using a variety of transports, before reaching its destination. Provisional and final responses to the request will be returned to the sender much in the same way as that of any other SIP request. Normally, a 200 OK response will be generated by the user agent of the request's final recipient.

However, this extension allows messages to be exchanged outside a session, independently each-other, and it is better suited for short messages exchanges. This approach is referred to as page-mode, as it resemble a pager-based exchange of messages. Session-mode messaging [7], where instant messages are exchanged in the context of a session, presents several advantages. Instant Messaging sessions in SIMPLE make use of the Message Session Relay Protocol (MSRP).

The MSRP protocol provides transport of instant messages in session-mode in an end-to-end fashion, running over a reliable transport protocol such as TCP or SCTP. MSRP sessions are managed using the Session Description Protocol (SDP) offer/answer model [12] combined
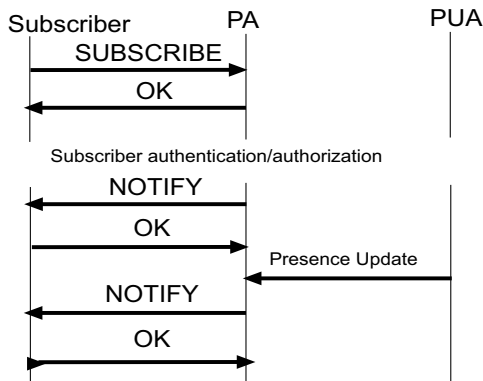
with SIP as message carrier. The MSRP protocol is rather simple, as it uses only two primitives: (i) SEND: for sending instant messages between endpoints (ii) VISIT: for establishing MSRP sessions. In order to establish a session, in a general use case with endpoint B, endpoint A sends in an offer message a temporary unambiguous URI, representative of the endpoint A. If B wishes to join the session, it opens a TCP connection to A, sends the VISIT message, addressed to the URI provided by A. After visiting the session, B sends the answer message for the offer received from A; the answer message contains a URI, where B can be contacted. After this exchange, instant messaging exchanges between A and B can begin; for each instant message sent with a SEND request, an OK reply will be sent by the interlocutor. More SEND messages can be sent even though they have not been (at SIP level) acknowledged yet. SEND requests are sent by either endpoint to the URI indicated by the peer; respectively.

Each message exchanged, shown in Fig 3, is labeled with a tag indicating whether it is a SIP or an MSRP message. Details on the exchanged messages can be found in [7].
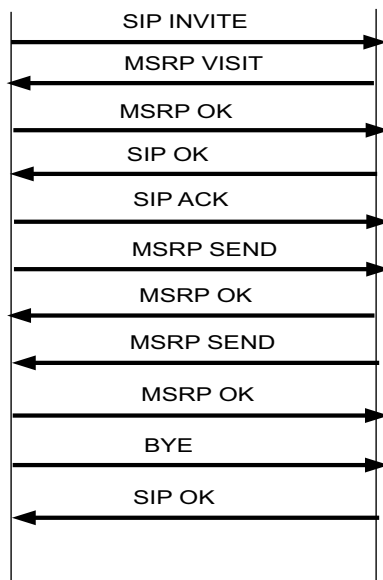

Fig. 3.   MSRP message exchange example

The offerer initiates the session using the SIP INVITE method; the INVITE request contains the parameters to be negotiated for establishing the session. In this case they are the transport protocol to be used (TCP for MSRP), the content type of the message bodies that can be accepted during the sessions, and the session parameter where the URI session descriptor, containing offerer address and port number is given to the answerer. The answerer-visitor opens a TCP connection with the offerer-host and sends the VISIT request.

The host acks the VISIT request (MSRP 200 OK response) and the visitor acks the INVITE request (SIP 200 OK request), that is, it answers to the offer. The SIP 200 OK response is further acknowledged by the generator of the INVITE request, according to the SIP model, to complete the SIP three-way handshake.

Host and visitor exchange one instant message each; the messages involved are two MSRP SEND requests and two MSRP 200 OK responses to acknowledge the safe receipt of the message. When the host wants to close the session, and sends a SIP BYE message, which will be answered by the visitor with a SIP 200 OK. Meanwhile, both endpoints release the local session state and the host drops the TCP connection.

IV. SIMPLE IN AD HOC NETWORKS

As described before, both the subscription to presence service and sending instant messages, involve the location of the IMP services corresponding to a unique URI, using the Internet and the overlay SIP proxy based infrastructure. None of these infrastructure is available in an ad hoc network. So, in order to enable such services in ad hoc networks, we must provision for (i) IMP service discovery in an infrastructure-less network and (ii) an efficient routing scheme for routing the SIP-based presence and instant messages. Now, some of these functions, such as node discovery and routing of data packets, are already done by the ad hoc routing protocol. So, instead of going for an auxiliary mechanism which performs the same task as the routing protocol, a better option is to integrate the desired functions with the routing protocol.

V. INTEGRATION WITH THE ROUTING PROTOCOL

The SIP-based IMP service discovery and the routing of IMP messages is integrated with a distributed cluster based routing protocol. The cluster based routing protocol creates a virtual topology with the cluster heads forming a backbone network, which is used in the routing of both SIP (the basic as well as the SIMPLE) messages and data packets. The basic assumption is that each node is equipped with a SIP user agent (with IMP services) and is able to take the extra responsibility of acting as a SIP proxy server and SIP registrar server.

A. Cluster based Routing Protocol

a) Virtual Topology Creation: The virtual topology creation involves two major steps of cluster head selection followed by cluster formation. They are described as follows.

- *Cluster Head Selection* Each node in the ad hoc network sends a periodic HELLO message to all its neighbors with the list of all of its neighbors. Each node on receiving the HELLO message computes the degree, or the number of adjacent neighbors of the node. This degree information is then broadcasted in the subsequent HELLO messages. Another data structure, called the adjacency table is used to embed 2-hop neighbor information in the HELLO messages. Thus each node gets to know about its degree, as well as that of its 1-hop and 2-hop neighbors. With this degree information each node selects itself as a cluster head if it satisfies any of the following two conditions.
  Condition 1: The node has the highest degree in its 1-hop neighborhood.
  Condition 2: The node has the highest degree in the 1-hop neighborhood of any of its 1-hop neighbors.
- *Cluster Formation:* Once the cluster heads get selected they assume the responsibility of a SIP proxy and a registrar server [14]. The 1-hop nodes adjacent to the cluster heads join the cluster identified by the cluster head with the highest node degree and the SIP UAC of the node registers its URI with the corresponding registrar server (in case of a tie, the cluster head with the lowest node address is chosen). It can be proved that the cluster heads are either 2-hops or 3-hops away from the nearest cluster heads. Each cluster head maintains connectivity with its neighboring cluster heads through gateway nodes selected by the following procedure. The HELLO message can detect the cluster heads which are 2 hops away but not those which are 3 hops away. For detecting the cluster heads 3 hops away, a cluster adjacency table is maintained at each node. Each cluster member gets information about its 2-hop cluster heads from the HELLO messages. It creates its own cluster adjacency table for its 2-hop away cluster heads with the intermediate 1-hop neighboring node, relaying the HELLO message, as the *gateway* node. The cluster adjacency table is then appended to the HELLO message as an extension and sent to all the 1-hop neighbors. Any cluster head in its 1-hop neighbor gets to know about the cluster heads which are 3 hops away and identifies the cluster adjacency table relaying node as the gateway node. In either case, there may be more than one candidate for the gateway node. In those cases, the node with the lowest node address is selected as the gateway node.

b) Routing Procedure: A presence service or a SIP UAS to which a SIP UAC wants to send an instant message, is uniquely identified by a SIP URI. The URI of the presence service or the target UAS is used as the `Request-URI` of the SIP requests *viz.* SUBSCRIBE, MESSAGE, and INVITE (session mode) messages. In our protocol, the cluster heads act as SIP proxies and as the forwarding nodes. The request messages are sent to the corresponding proxy of the requesting node. The proxy then sends this message to the neighboring cluster heads or proxies in order to discover the route to the target node. If any of the neighboring proxies has the target URI registered with itself, it sends the request message to the target node, otherwise it forwards the message to its neighboring cluster heads after recording the proxy address in the `Record-Route` field of the SIP message. The target node on receiving the request message sends back a SIP OK message via the reverse route specified by the list of traversing proxies in `Record-Route` header field. This is exactly the same as the typical proxy based routing of SIP messages [14]. The requesting node on receiving the SIP OK message, gets to know about the route to the target, which is used subsequently for both SIP session establishment and media packet delivery. The routes are also used to send the MESSAGE or the SEND messages of the instant messaging applications, directly to the target UAS. The NOTIFY messages of the presence applications are sent in the same fashion as that of the SIP OK response messages.

B. Cache Management

A cache is maintained at each of the proxies for already discovered routes corresponding to each unique URI. The cache, populated by the SIP OK message, essentially keeps a mapping of the neighboring cluster head to which the request message should be forwarded corresponding to a particular target URI. This saves the overhead associated with the discovery process each time a request or an instant message is sent to the same target. However, in an ad hoc network, the nodes are mobile and the cache need to be updated each time a node in the path to a target URI moves.

There can be potentially two cases of node movements: (1) the cluster member nodes or the endpoints move and (2) any of the proxies in the path between the two endpoints move. In either case the cache in the intermediate proxies need to be updated.

1) When the requesting node moves to a different cluster with a new cluster head, which does not has a mapping of the path to the target URI in

its cache, a new discovery process is initiated. The request message, with the target URI in the `Request-URI` field is sent through the network of cluster heads until an entry corresponding to the target URI is found in the cache of one of the cluster heads. Otherwise, the request message reaches the target node and the routing procedure, described previously, is employed.

If the target node moves to a different cluster, it registers with the registrar service of the corresponding cluster head, as well as with the previous cluster head's registrar service with updated contact header. Thus, when the request arrives at the old cluster head according to the cache entries, a 301 (Moved Permanently) response is sent back to the requesting client with updated contact information. All the cache entries, corresponding to the particular target URI in the path of the response message, are removed. The requestor, finally, redirects the future requests to the address in the updated contact information.

2) When any of the proxies in the path moves, the immediately upstream proxy would not receive a timely response from the downstream proxy. The upstream proxy would send a 408 (Request Timeout) response back to the requestor, thus removing all the cache entries for the particular target URI in the intermediate proxies. The requestor on receiving the 408 response initiates a fresh round of route discovery process.

C. Improving Routing Efficiency

The union of cluster heads and the gateway nodes define a fixed (or relatively static) connected multihop wireless network, where each of the nodes act as "wireless IP router" forwarding both the SIP messages and media packets. A forwarding node typically receives packets from the upstream nodes and then transmits them to the downstream nodes. Efficient and fast forwarding of messages is particularly important in the context of instant messaging services considering the heavy load and the real-time requirement of the instant messages. However, multihop IEEE 802.11 wireless LAN, the most dominant of the present-day multihop networking technology, pose several challenges in terms of the available system throughput due to multihop routing inefficiency. The current 802.11 Distributed Coordination Function (DCF) MAC algorithm has been designed implicitly for either receiving or transmitting a packet, but not for a forwarding operation (i.e., receiving a packet from an upstream node and then immediately transmitting the

packet to a downstream node as an atomic channel access operation). There are two key deficiencies:

- The forwarding node is involved in two separate RTS/CTS contention-based channel access attempts during the forwarding process: once to receive the packet (from the upstream node) and again to forward it (to the downstream node), and must thus suffer the contention resolution overhead twice.
- The same packet makes an unnecessary round-trip between the memory on the network interface card (NIC) and the hosts memory (accessed by the host software) to determine the next-hop MAC address. This round-trip not only loads the processor of the forwarding node, but also suffers from additional delays in transfers between the NIC and the host operating system.

A wireless IP forwarding architecture that uses MPLS [11] with modifications to IEEE 802.11 MAC has been proposed in [5] primarily to solve this problem and significantly improve the packet forwarding efficiency. The overheads of separate channel accesses is eliminated by defining the Data-Driven Cut-Through Medium Access (DCMA) protocol [5] as a simple extension of the 802.11 DCF. DCMA combines the Data ACK (to the upstream node) with the RTS (to the downstream node) in a single ACK/RTS packet that is sent to the MAC broadcast address. The problem of round-trip delay between the memory on the NIC and the hosts memory is solved by enabling the lookup for next hop within the NIC, without needing to perform the routing lookup in the host. MPLS, a well-known IP compatible technology has been used to perform next-hop lookup inside the NIC, by setting up labels that enables fast and scalable determination of the MAC address of the downstream node. This technology can be used in our proposed integrated routing protocol, when the route to a particular target has been established by populating the cache in the proxies.

The system throughput of the IEEE 802.11 multihop networks, another important parameter in the context of IM applications, can be further improved by increasing concurrent transmissions through better spatial reuse. The 802.11 MAC protocol and its variants are primarily designed for a single-hop wireless environment, where nodes typically form a clique and communication always takes place over a single wireless hop (often to a base station providing connectivity to the wired infrastructure). In such a single-cell environment, the 802.11 MAC contention resolution mechanism focuses primarily on ensuring that only a single sender-receiver node pair receives collision-free access to the channel at any single instant. The 802.11 MAC does not seek to exploit the

spatial diversity inherent in multihop networks, where different sets of nodes are able to concurrently communicate with different sets of neighbors. This can be achieved potentially by three different methods: use of power control algorithms, use of directional antennas and modification of the MAC itself to relax some unduly harsh restrictions of the IEEE 802.11 MAC. One such 802.11-like protocol is called MACA-P [4] that provides synchronized parallel transmissions by allowing neighboring nodes to synchronize their reception periods, so that 1-hop neighbors switch between transmitting and receiving roles in unison at explicitly defined instants, and thus avoid the problem of packet collisions. MACA-P can be used for efficient message routing through the cluster heads with enhanced system throughput.

VI. CONCLUSION

In this paper we have discussed the extensions to SIP proposed by the IETF SIMPLE WG in the context of ad hoc networks. We have presented a brief overview of the extensions and the problems to support them in ad hoc networks. An integrated approach, where the SIMPLE protocol operations are integrated with a cluster based routing protocol has been proposed. Caching and enhancement of IEEE 802.11 MAC layer have been discussed in the context of performance improvement of the integrated approach. We would quantify these improvements with results from simulation and practical experiments, in our future works.

REFERENCES

[1] Instant Messaging and Presence Protocol (IMPP) IETF Working Group. http://www.ietf.org/html.charters/impp-charter.html

[2] SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE) IETF Working Group. http://www.ietf.org/html.charters/simple-charter.html

[3] Extensible Messaging and Presence Protocol (XMPP) IETF Working Group. http://www.ietf.org/html.charters/xmpp-charter.html

[4] A. Acharya, A. Misra, and S. Bansal, "MACA-P : a MAC for concurrent transmissions in multi-hop wireless networks" *IEEE PerCom*, pp. 505–508, 2003.

[5] A. Acharya, A. Misra and S. Bansal, "A label-switching packet forwarding architecture for multi-hop wireless LANs", *WoW-MoM*, pp. 33-40, 2002.

[6] B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema, D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging". *RFC 3428*, December 2002.

[7] B. Campbell, R. Mahy, C. Jennings , "The Message Session Relay Protocol". *Internet Draft (Work in Progress) draft-ietf-simple-message-sessions-07.txt* , July 2004.

[8] M. Handley, and V. Jacobson, "SDP: Session Description Protocol", *IETF RFC 2327*, April 1998.

[9] W. Jiang, J. Lennox, H. Schulzrinne and K. Singh, "Towards Junking the PBX: Deploying IP Telephony", pp. 177-185 *NOSSDAV* 2001.

[10] A. B. Roach, "Session Initiation Protocol (SIP) Specific Event Notification",. *RFC 3265*, June 2002.

[11] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," *IETF RFC 3031*, Jan. 2001.

[12] J. Rosenberg, H. Schulzrinne, "An Offer/Answer Model with the Session Description Protocol (SDP)". *RFC 3264*, June 2002.

[13] J. Rosenberg, "A Presence Event Package for the Session Initiation Protocol (SIP)".*Internet Draft (Work in Progress), (In the RFC editor queue pending for publication), draft-ietf-simple-presence-10.txt*, July 2003.

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

[15] H. Schulzrinne et al. "RTP: A Transport Protocol for Real-Time Applications" *IETF RFC 1889* Jan 1996.