Session Initiation Protocol (SIP)

Jouni Soitinaho
Jouni.Soitinaho@nokia.com

Abstract

This paper describes the basic characteristics of the SIP protocol and especially its extension mechanism. Several Internet draft specifications are studied in order to get an overall picture of the maturity of the protocol. Some interesting application areas are examined for demonstrating how the SIP protocol suite can be used in a wider context.

1 Introduction

SIP is a simple but extendable signaling protocol for setting up, modifying and shutting down communication sessions between two or more participants. One or more media or even no media at all, can be transmitted in the session context. SIP is independent of the actual media and the route of the media can be different to the route of signaling messages. SIP can also invite participants to IP multicast session.

SIP is part of the IETF multimedia architecture and it's designed to cooperate with several other protocols, which is a fundamental principle of the SIP design. Other protocols include, for example, RTP and RTCP for media transport, RTSP for controlling streaming and SDP for describing the capabilities of the participants. Limiting the SIP protocol to the controlling of the session state is also more likely to keep it simple and easy to implement.

Another fundamental aspect of SIP design is the easy way it can be extended with additional capabilities. Actually, the basic protocol specification defines rather limited signaling protocol. It is missing several capabilities needed by real life applications. Several general extensions are being defined currently and some of these are expected to be included in the basic standard after reaching the required stability.

SIP was first developed within the Multiparty Multimedia Session Control (MMUSIC) working group and then continued in the SIP working group. Active communications with MMUSIC is important since the Session Description Protocol (SDP) is developed by MMUSIC. The working group has also close relationship with the IP telephony (iptel) working group, whose Call Processing Language (CPL) relates to many features of SIP, and the PSTN and Internet Internetworking (pint) working group, whose specification is based on SIP. Distributed Call Signaling Group (DCS) is giving input to SIP for distributed telephony services. Recently it was decided to split the SIP working group to two: SIP WG will concentrate on the basic protocol and general extensions and SIPPING WG will concentrate on applications and generate input to the SIP WG.

Besides all the activities taken by the IETF task forces 3GPP technical specification groups currently investigate SIP. Since SIP was chosen as the signaling protocol for the IP multimedia subsystem of 3G network 3GPP will set new requirements for the protocol.

The basic SIP protocol is defined in RFC2543 that is currently in "proposed" state. The corresponding Internet draft document [1] contains many updates and is the reference document for describing the basic protocol in the next section. Some of the current development activities are discussed in section three. Finally, a few application areas of SIP are studied in section four before conclusions in the last section.

2 Basic Protocol

2.1 Characteristics

The basic features of SIP:

- Locating user: determination of the end system to be used for communication;
- Determining user capabilities: determination of the media and media parameters to be used;
- Determining user availability: determination of the willingness of the called party to engage in communications;
- Setting up the call: "ringing", establishment of call parameters at both called and calling party;
- Controlling the call: including transfer and termination of calls.

Main technical properties and some implications of SIP:

- Text-based (ISO 10646 in UTF-8 encoding), similar to HTTP: Easy to learn, implement, debug and extend. Causes extra overhead, which is not a serious drawback for a signaling protocol. Header names can be abbreviated.

?? Recommended transport protocol is UDP: It is not meant to send large amounts of data.
?? Application level routing based on Request-URI: The signaling path through SIP proxies is controlled by the protocol itself not by the underlying network. Requires routing implementation in SIP proxies.
?? Independence on the session it initiates and terminates (capability descriptions, transport protocol, etc.): Cooperates with different protocols, which can be developed independently. It is not a conference control protocol (floor control, voting, etc.) but it can be used to introduce one.
?? Supports multicasting for signaling and media but no multicast address or any other network resource allocation.
?? Support for stateless, efficient and "forward" compatible proxies (re-INVITE carries state, ignore the body, ignore extension methods).

2.2 Operations

Protocol operations of SIP:

?? `INVITE` initiates session establishment
?? `ACK` confirms successful session establishment
?? `OPTIONS` requests capabilities
?? `BYE` terminates the session
?? `CANCEL` cancels a pending session establishment
?? `REGISTER` binds a permanent SIP URL to a temporary SIP URL for the current location.

The following diagram demonstrates SIP protocol operations for user registration and session handling.

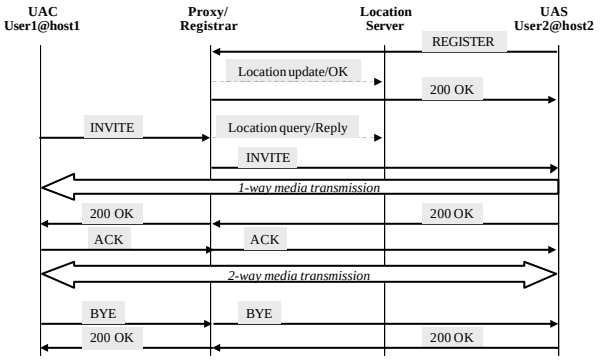

Figure 1. An example of SIP protocol operations.

2.3 Network elements

SIP has been designed for IP networking. The protocol makes use of standard elements like DNS and DHCP servers, firewalls, NATs and proxies. Special support in DNS and DHCP servers is not needed but it makes the protocol operations more efficient. The SIP protocol is implemented by the user agent client (UAC) and server (UAS), redirect servers, proxies and registrars. Registrars and location servers maintain the mapping between user's permanent address and current physical addresses.

The SIP specification does not actually define the network architecture. However, the logical elements and their relationships can be determined based on the protocol specification. The following figure demonstrates an example of inter-domain session setup. Both UAC and UAS are located in their home domains. Thin lines represent SIP signaling messages and thick lines represent media transmission and dotted line represent non-SIP protocol.

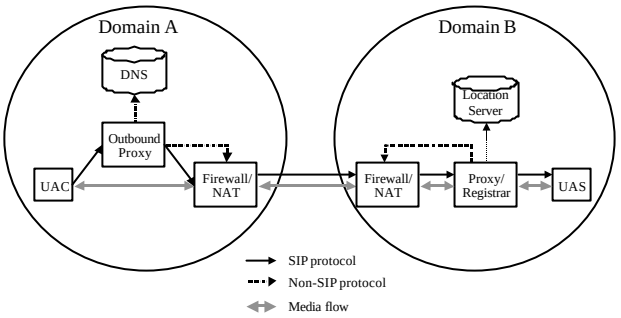

Figure 2. Logical network elements involved in an inter-domain session setup.

In this scenario UAC composes an INVITE message in order to set up a call with UAS. The message contains the session data in its headers and media descriptions in the body in SDP format [2]. INVITE is sent to Outbound Proxy whose address may have been configured in UAC using DHCP. Outbound Proxy uses DNS to resolve the recipient's address. It also controls Firewall/NAT to open the ports for media transmission. Domain B has configured all the incoming requests to go to Proxy/Registrar that controls Firewall/NAT of Domain B. Proxy/Registrar queries the current location of UAS from Location Server and forwards the message to UAS. In an intra-domain call a redirect server could be used instead of a proxy in Domain B to return the current location of UAS who could then be contacted directly by UAC without having any proxy involved in the communications.

Since the request carried the media descriptions of UAC and since the corresponding ports were opened in firewalls media can immediately flow back from UAS to UAC. The signaling response is routed along the same path as the request and it carries the media descriptions of UAS. UAC can now send media to UAS. Finally UAC has to send ACK message to UAS for acknowledging the successful session establishment.

2.4 Addressing and routing

SIP uses e-mail like addresses for users but it also includes the protocol keyword in the SIP URL. SIP URLs are used to identify the originator (From), current destination (Request-URI), final destination (To) and redirection address (Contact).

Two formats exist:

?? `sip:user@host`
when UA exists, e.g. From and To fields in INVITE

?? `sip:host`
when no UA exists, e.g. Request-URI in REGISTER

Including the protocol keyword in the URL allows SIP server use the Contact-header to redirect a call to a web page or to a mail server, for example. This facilitates integration of audio and video applications with other multimedia applications.

Routing of SIP messages is included in the protocol itself since finding the user is one of the primary functions of SIP. The host part of the SIP URL indicates the next hop for a request. Even if clients could send the request directly to this address in practice they are typically forced to go through a proxy for security or address translation reasons.

Furthermore two headers are in central position for routing SIP messages:

?? `Via` header indicates the request path taken so far. It prevents looping and is used for routing the response back the same path as request has traveled. Proxies must add "received" parameter in the top-most Via header if the field contains different address than the sender's source address. This feature supports NAT servers. Proxies can also forward the request as multicast by adding "maddr" parameter in the Via field.

?? `Route` header is used for routing all requests of a call leg along the same path, which was recorded in the `Record-Route` header during the first request. This is to guarantee that stateful proxies will receive all the subsequent messages that affect the call state.

SIP proxies can also fork the incoming request to several outgoing requests in order to accelerate the processing of INVITE method. The forking can create several simultaneous unicast INVITEs to the potential locations or one multicast INVITE to a restricted subnetwork. Even if forking is an efficient mechanism it is a potential source of difficult problems and needs to be paid special attention during implementation.

2.5 Registering

A client uses REGISTER method to bind its permanent address to one or more physical addresses where the client can be reached. The request is sent to the registrar, which is typically co-located with a proxy server. Alternatively the request can be sent to the well-known SIP multicast address "sip.mcast.net".

REGISTER method is also ideally suited for configuration and exchange of application layer data between a user agent and its proxy. This may produce modest amounts of data exchanges. However, because of the infrequency of such exchanges and their typical limitation to one-hop this is acceptable if TCP is used.

The most important fields for the REGISTER method:

?? `Request-URI` names the domain of the registrar. `user` part must be empty.

?? `To` indicates the user to be registered

?? `From` indicates the user responsible for the registration (typically equal to To header value)

?? `Contact` (optional) indicates the address(es) of the user's current location. List of current locations can be queried by leaving the Contact header empty in the REGISTER request. An optional `expires` parameter indicates the expiration time of the particular registration. By giving the wildcard address "*" in a single contact header a client can remove all the registrations. By giving zero as the value for the expires parameter a client can remove the corresponding registration.

?? `Expires` tell the default value for expiration unless the corresponding parameter is present in the `Contact` header. If neither one is present default value of one hour is used.

It is particularly important that REGISTER requestor is authenticated.

2.6 SIP Security

Security must be addressed at several levels. At the network level the security is based on regular firewalls and NATs since SIP is designed for IP networking. Controlling the firewall with a SIP proxy is an essential enhancement for the standard IP security mechanisms.

At the protocol level both the media security and signaling security must be addressed. Media encryption is specified in the message body with SDP [2].

Signaling security includes user authentication and encryption of the signaling messages. User authentication is based on HTTP authentication mechanism [3] with minor modifications as specified in [1]. Besides "Basic" and "Digest" authentication

schemes SIP supports also stronger authentication with "PGP" scheme [4]. It is based on public key cryptography, which requires the client to sign the request with the private key and the server to verify the signature with the public key. It is recommended to authenticate the REGISTER requestor with the PGP scheme instead of the other schemes.

SIP also supports PGP encryption of the signaling messages. By setting the "Encryption" header to "PGP" scheme all following headers can be encrypted as well as the message body. Note that sending the media encryption key in the body requires the message body to be encrypted. Note also that there are special considerations for the encryption of the Via header since it is used by the proxies.

Obviously, standard IPSec protocol can be used for IP level encryption.

2.7 Expandability

In order to keep the basic protocol compact SIP provides the protocol designers with means for extending its capabilities. Protocol elements that can be extended without change in the protocol version include:

?? Methods
?? Entity headers
?? Response codes
?? Option tags

In addition to the SIP extensions the session description (SDP) can be extended to contain new attributes and values for the session.

Several definitions in the protocol set the limits for the extensions. First of all, proxy and redirect servers treat all methods other than INVITE, CANCEL and ACK in the same way by forwarding them. User agent server and registrar respond with the "501 Not Implemented" response code for request methods they do not support.

SIP servers and proxies ignore header fields not defined in the specification [1] and they do not understand, i.e. treating them as entity headers. General headers, request headers and response headers are extended only in combination with a change in the protocol version. Furthermore, stateless proxies are required to recognize only the values defined in the basic protocol. They will forward new values without actions. Session stateful proxies need to support the extension if it can change the call state in a way, which is meaningful for the proxy.

SIP applications are not required to understand all registered response codes. They must treat any unrecognized response code as being equivalent to the x00 response code of that class, with the exception that an unrecognized response must not be cached.

Option tags are unique identifiers used to designate new extensions for SIP. These tags are set in Require, Proxy-Require, Supported and Unsupported header fields to communicate the signaling capabilities between UACs, UASs and proxies. The extension creator must either prefix the option with the reverse domain name or register the new option with the Internet Assigned Numbers Authority (IANA).

Clients can always call the OPTIONS method for explicitly querying the capabilities of the server and proxies lying on the path.

Since there are multiple ways to define a SIP extension special attention needs to be paid on the semantic compliance with the basic protocol. An informational Internet draft sets the guidelines for writing a SIP extension [5].

3 Protocol Extensions

About 30 extension drafts can be found on http://www.cs.columbia.edu/~hgs/sip/drafts_base.html. Some of these add reliability or functionality missing in the basic protocol for supporting real time services like VoIP. Examples of these are "reliable provisional responses", "resource management" and "INFO method". Some extensions add functionality for implementing existing PBX services, like call transfer. Examples are "call control-transfer" and "caller identity and privacy". Some extensions add new functionality for enabling new type of services, like presence based instant messaging. Examples are "event notification" and "caller preferences". Finally some extensions add resilience to the basic protocol for implementing reliable and scalable networks. Examples are "session timer" and "distributed call state".

3.1 Reliable provisional responses

When run over UDP, SIP does not guarantee that provisional responses (1xx) are delivered reliably, or in order. However, many applications like gateways wireless phones and call queuing systems make use of the provisional responses to drive state machinery. This is especially true for the 180 Ringing provisional response, which maps to the Q.931 ALERTING message.

The Internet draft document [6] specifies an extension to SIP for providing reliable provisional response messages ("100rel"). When a server generates a provisional response which is to be delivered reliably, it places a random initial value for the sequence number (RSeq).

The response is then retransmitted with an exponential backoff like a final response to INVITE.

The client uses a new method (PRACK) for acknowledging the provisional response. Unlike ACK, which is end-to-end, PRACK is a normal SIP message, like BYE. Its reliability is ensured hop-by-hop through each stateful proxy. PRACK has its own response and therefore existing proxy servers need no modifications. A new header (RAck) in the PRACK message indicates the sequence number of the provisional response, which is being acknowledged.

The following diagram demonstrates how the support and need for reliable provisional response is negotiated and implemented.

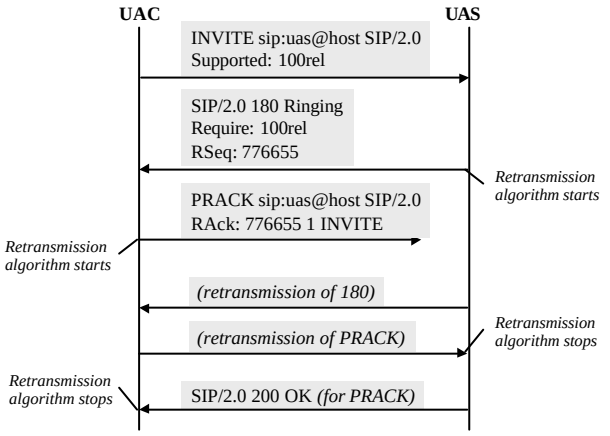

Figure 3. Reliable provisional response.

3.2 Resource Management

In order to become a successful service Internet telephony must meet the quality expectations based on the existing telephony services. This implies that the resources must be reserved beforehand for each call. Cooperation is therefore needed between call signaling, which controls access to telephony specific services, and resource management, which controls access to network-layer resources

The Internet draft document [10] discusses how network QoS and security establishment can be made a precondition to sessions initiated by SIP, and described by SDP. These preconditions require that the participant reserve network resources or establish a secure media channel before continuing with the session. In practical terms the "phone won't ring" until the preconditions are met. The draft proposes new attributes for SDP:

?? `"a=qos:" strength-tag SP direction-tag`

?? `"a=secure:" SP strength-tag SP direction-tag`

where the strength can have values "mandatory", "optional", "success" and "failure" and the direction can have values "send", "recv" and "sendrecv".

The document also proposes a new method to SIP. The COMET method is used to confirm the completion of all preconditions by the session originator. The following diagram presents the message flow for a single-media session setup with a "mandatory" quality-of-service "sendrecv" precondition, where both the UAC and UAS can only perform a single-direction ("send") resource reservation.

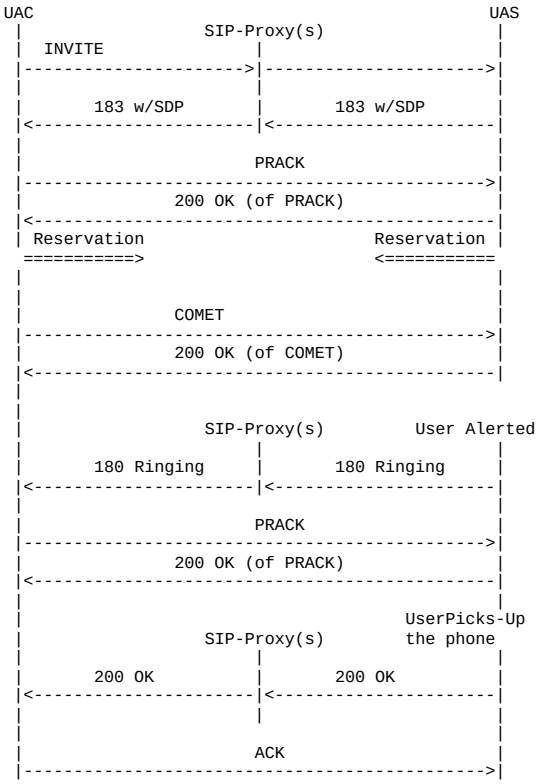

Figure 4. Resource management signaling.

The session originator (UAC) prepares an SDP message body for the INVITE describing the desired QoS and security preconditions for each media flow, and the desired direction "sendrecv." This SDP is included in the INVITE message sent through the proxies, and includes an entry "a=qos:mandatory sendrecv." The recipient of the INVITE (UAS), returns a 183-Session-Progress provisional response containing SDP, along with the qos/secure attribute for each stream having a precondition. The UAS now attempts to reserve the qos resources and establish the security associations. The 183-Session-Progress is received by the UAC, and the

UAC requests the resources needed in its "send" direction, and establishes the security associations.

The diagram also demonstrates the usage of PRACK and COMET methods for confirming the responses and resource allocations respectively.

3.3 INFO method

The SIP INVITE method can be called one or more times during the established session (re-INVITE) to change the properties of media flows or to update the SIP session timer. However, there is no general-purpose mechanism to carry session control information along the SIP signaling path during the session.

RFC2976 [14] defines the INFO method for communicating mid-session information during the call. It is not used to change the state of the session but it provides means for exchanging additional information between the peers. One example of such session control information is ISUP and ISDN signaling messages used to control telephony call services.

The information can be conveyed either in the header of the INFO message or as part of the message body. The definition of the message body and/or message headers used to carry the mid-session information is outside the scope of this document. However, consideration should be taken on the size of message bodies since it can be fragmented while carried over UDP bearer.

3.4 Call Control - Transfer

The basic SIP protocol does not support any of the multiple ways a call can be transferred to a third party. In an "unattended transfer" the transferor is not participating the call simultaneously with the transferee and transfer target whereas in an "attended transfer" the three actors participate the call simultaneously (ad-hoc conference). In an "consultation hold transfer" the transferor establishes and terminates a second call with the transfer target before performing the actual transfer.

The Internet draft document [11] proposes a SIP extension, which can be used, for example, to implement traditional unattended and consultation hold transfers. The attended transfer is not drafted yet since the call control framework has not addressed conferencing. The following figure presents the message sequence of unattended transfer with consultation hold.

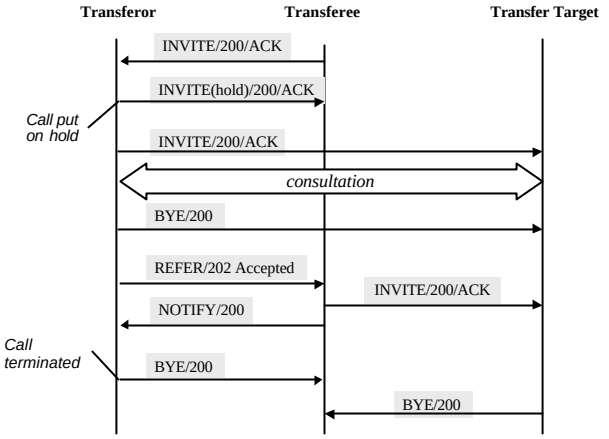

Figure 5. Unattended call transfer with consultation on hold.

The new REFER method indicates that the recipient (Request-URI) should contact a third party identified by the contact information (Refer-To). Once the transferee knows whether the transfer succeeded or failed it notifies the transferor by sending "refer" event using the NOTIFY mechanism as if the REFER message had established a subscription.

3.5 Caller Identity and Privacy

In order for SIP to be a viable alternative to the current PSTN, it must support certain telephony services including Calling Identity Delivery, Calling Identity Delivery Blocking, as well as the ability to trace the originator of a call. While SIP can support each of these services independently, certain combinations cannot be supported. The issue of IP address privacy for both the caller and callee needs to be addressed as well.

The Internet draft document [12] specifies two extensions to SIP that allow the parties to be identified by a trusted intermediary while still being able to maintain their privacy. A new general header, Remote-Party-ID, identifies each party. Different types of party information can be provided, e.g. calling, or called party, and for each type of party, different types of identity information, e.g. subscriber, or terminal, can be provided. Another new general header, Anonymity, is also defined for hiding the IP addresses from the other parties.

3.6 Caller preferences

When a SIP server receives a request, there are at least three parties who have an interest and each of which should have the means for expressing its policy:

?? The administrator of the server, whose directives can be programmed in the server.

?? The callee, whose directives can be expressed most easily through a script written in the call processing language (CPL)
?? The caller, who doesn't have obvious ways to express the preferences within the SIP server.

The Internet draft document [9] specifies an extension mechanisms by which the caller can provide its preferences for processing a request. These preferences include the ability to select which URIs a request gets proxied or redirected to, and to specify certain request handling directives in proxies and redirect servers. It does so by defining three new request headers, Accept-Contact, Reject-Contact and Request-Disposition. The extension also defines new parameters for the Contact header that describe attributes of a UA at a specified URI.

3.7 Event Notification

The ability to request asynchronous notification of events is useful in many types of services. Examples include automatic callback services (based on terminal state events), buddy lists (based on user presence events), message waiting indications (based on mailbox state change events), and PINT status (based on call state events).

The Internet draft document [13] proposes a framework by which notification of events can be ordered. The draft can't be used directly, i.e. it doesn't specify any event types and it must be extended by other specifications (event packages). In object-oriented terminology, this is an abstract base class which must be derived into an instantiatable class by further extensions.

The extension is based on two new methods: SUBSCRIBE and NOTIFY and a new header "Event" together with the "Expires" header. Neither SUBSCRIBE nor NOTIFY necessitates the use of "Require" or "Proxy-Require" header and no extension token is defined for "Supported" header. Clients may probe for the support of SUBSCRIBE and NOTIFY using the OPTIONS method.

There is no separate media transmission between the subscriber and notifier as in normal SIP session. The message body of the NOTIFY method is to carry the actual notification.

Removing and refreshing subscriptions are performed in the same way as for REGISTER method. Usage of the message body in SUBSCRIBE request is left up to the concrete extensions. It may be used to filter and set thresholds for the events.

The basic scenario of a notification session is presented in the following figure. Note that according to the SIP principle proxies need no additional behavior to support SUBSCRIBE and NOTIFY methods but they can act as subscribers and notifiers.

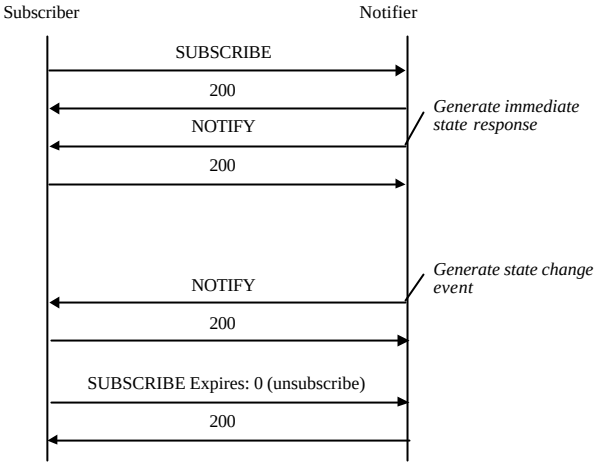

Figure 6. Event notification messages.

This extension is not targeted to very frequent notifications. The interval must be minutes instead of seconds. For better performance and for simplifying the subscriber implementation the new state after the event must be notified in addition to the event itself. The extension is not either for transferring large amounts of data since the preferred transport protocol is UDP. Therefore this extension is not fully in line with the SIP extension guidelines.

3.8 Session timer

SIP does not currently define a keepalive mechanism. The result is that call stateful proxies are not always able to determine whether a call is still active or not. For instance, when a user agent fails to send a BYE message at the end of a session, or the BYE message gets lost due to network problems, a call stateful proxy will not know when the session has ended.

This is especially important feature for proxies controlling firewalls or NATs or performing billing tasks. Holes and address bindings are dynamically created in firewall and NATs to allow the media for the session to flow. These settings represent state which must be eventually removed.

The Internet draft document [7] specifies the session timer extension ("timer") for solving the problem and improving the reliability of the basic SIP protocol. UAC, UAS and proxies communicate the support for the extension and assign the responsible party (UAC or UAS) for sending the re-INVITEs in the original

INVITE message. If UAC supports the extension it sets "timer" in the Supported header and if it wants to turn the extension on it sets the refresh interval in Session-Expires header. UAC will then be responsible for sending the re-INVITEs. A proxy may adjust the refresh interval to a smaller value and also require (Proxy-Require) UAS to send the re-INVITEs in case UAC does not support the extension. If a re-INVITE is not received before the refresh interval passes, the session is considered terminated, and call stateful proxies can release the session.

Note that using INVITE as the refresh method, as opposed to a new method, allows sessions to be recovered after a crash and restart of one of the UAs.

3.9 Distributed Call State

Many types of services require proxies to retain call state. Unfortunately, maintaining call state presents problems. It introduces scalability problems and makes fallback and load balancing more complex.

The extension proposed in the Internet draft document [8] allows proxies to encapsulate any state information they desire into a header, called State header. The header is sent to the user agents and reflected back in subsequent messages.

The idea is similar to the use of cookies with HTTP user agent clients and proxies. In essence, it allows proxies to behave as stateful proxies while still being stateless.

4 Applications

4.1 Call centers

There are multiple ways to implement a SIP based call center where more than one operators can provide the same service for incoming requests. In a very simple model a redirect server is used together with the registrar to redirect the calls to a free operator according to a round robin algorithm, for example. The server can use the Contact header with the maddr parameter to instruct the caller to send the next INVITE with the same Request-URI but connect to the host indicated by the maddr parameter.

This is a very limited solution since the redirect server has no automatic means to record the state of the operators. Of course, they could send re-REGISTER message whenever they are free for a new call but this is not according to the semantics of the REGISTER message. In fact, SIP provides a better way for implementing the application.

Using a SIP proxy instead of a redirect server the state of each call can be maintained by listening to the SIP messages. The address of the proxy is published externally and no direct connections to the operator addresses are allowed through the firewall. The proxy includes itself in the message path using Record-Route and Via headers in order to get the CANCEL and BYE requests as well as all the responses. When a new call arrives the proxy decides the operator based on its own call state information and information in the registrar.

Sending the INVITE message using IP multicast can accelerate the seeking of operator. Free operators generate a response within a random time interval. Since all operators will hear the first response they can drop the request without responding. If no operator is free proxy retries until one is free or the client terminates the call by sending CANCEL request which is responded by the proxy. The proxy generates all call statistics.

If the network does not support IP multicast yet another option is to fork the request in the proxy into simultaneous requests to the current locations of the free operators. In this case the cancellation of the other INVITE messages need to be performed by the proxy whenever the first operator responds.

4.2 Presence and Instant Messaging

Presence is considered as a promising application area in all-IP networks. When combined with instant messaging it creates a lot of opportunities for application developers. A new working group, called SIMPLE (SIP for Instant Messaging and Presence Leveraging), has been established in IETF for developing specifications in this area. 3GPP is also considering presence as one service for the IM subsystem.

Presence is defined as user's reachability, capabilities and willingness to communicate with other users. Presence application obviously has to provide the means to deliver this information to other users. A lot of room exists for differentiating applications from each other's. For example, intelligent filters for exposing the presence and accepting calls can be built based, for example, on user's location and caller's identity.

Instant messaging (IM) is defined as the exchange of content between a set of participants in real time, like in IRC. The content is mainly small textual messages but they can also contain pictures or audio or video clips. The main difference to emails is the real time nature requiring all the parties to be online.

It is very important to keep presence and IM separate from each other even if these are mixed in the existing, proprietary solutions. The separation enables

independent development of the two protocols. This is important also because of the existing IM applications (multiplayer online games).

SIMPLE bases its work on the existing SIP and extension drafts. The foundation of using SIP for the presence and IM protocols derives from two factors: the SIP registrars already hold some information about the user's presence and SIP networks already route messages from user to the proxy that can access this information [15,16]. Extending SIP for this area is rather small step in terms of protocol operations but semantically it is a bigger step, however.

The presence extension is an instantiation of the abstract notification extension. A new event package, named as "presence", is defined for this purpose. The body of the NOTIFY message contains a presence document. An XML data format and a MIME type will be defined for the document. The following figure shows the logical elements for SIP presence.

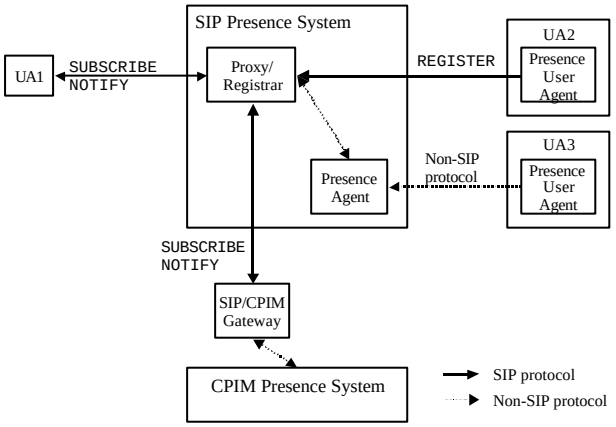

Figure 7. Logical network elements for SIP presence.

The presence agent (PA) is capable of storing the subscriptions and generating notifications based on the events. Present user agent (PUA) updates presence information.

Authorization is a critical component of a presence protocol. Authorization can be pushed to the server ahead of time or, more typically, determined at the time of subscription. Since this is not covered by the basic SIP protocol an Internet draft [17] proposes a new method (QUATH) for querying the authorization from the subscription authorizer (e.g. PUA). This draft seems to be arguable, however.

The IM protocol extensions are defined in the Internet draft [18]. When a user wishes to send an instant message to another, the sender issues a SIP request using the new MESSAGE method. The request URI can be in the format of "im: URL" or normal SIP URL. The body of the request contains the message to be delivered. Provisional and final responses will be returned to the sender as with any other SIP request. The following diagram shows two message exchanges between two users.

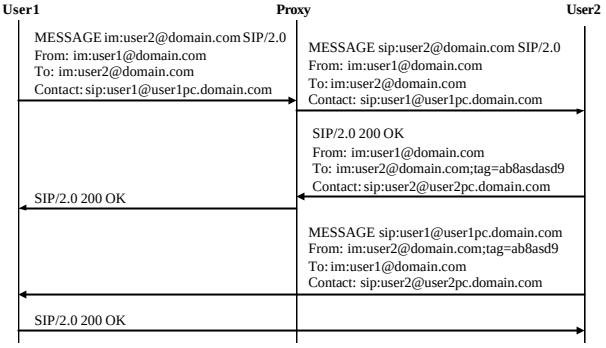

Figure 8. Instant messaging between users in the same domain.

Proxy looks up the registration database for the binding from im address to sip address of User2 and forwards the message to the current location. The response traverses the same path. Based on the Contact header of the message User2 can send the second message directly to User1's current location because Proxy added no Record-Route header in the first message. The From and To headers are reversed, however.

The specifications for presence and instant messaging are still rather insufficient. This is indicated by the long list of open issues listed in the drafts.

The semantic difference between presence and IM protocols and basic SIP protocol is in the type of session they create. Presence protocol creates a passive session which is used asynchronously for notifying the subscriber using the signaling channel without any media channel. Establishment and termination of the session is done differently to the basic protocol. IM does not create a session at all which is currently discussed in the working group. Surrounding the related MESSAGE requests with INVITE and BYE requests would be consistent with the basic protocol..

5 Conclusion

Simplicity is a key characteristic of SIP. It facilitates interoperable clients, servers and proxies coming from independent vendors. Sharing a lot of similarities with HTTP makes the understanding of SIP rather easy for a large developer community.

Expandability is another key characteristic. Being inbuilt in the basic protocol it provides the means for extending

the protocol capabilities. Network elements can dynamically negotiate their capabilities. The basic protocol specification can concentrate on its primary function.

Supporting different protocols for different purposes is yet another key characteristic of SIP. This facilitates protocol development independence between SIP and other protocols and makes the overall adoption of SIP more likely.

A lot of SIP related development activities are going on in IETF (over 70 drafts). This is an evidence of its potential on one hand but an evidence of its immaturity on the other hand. The potential is demonstrated by the application examples presented in this paper. The immaturity for IP telephony is demonstrated by the large number of suggested extensions described in this paper that are fundamental for this area.

Many extensions seem to be very useful and easy to specify at first sight. However, they may not share the semantics of the basic protocol and should not be defined as a SIP extension. The ability of IETF to respond to the needs and at the same time control the specification work will be tested in near future.

The slowness of the IETF process is indicated also by its inability to promote the basic SIP specification to "draft" state after being in "proposed" state over two years. At the same time 3GPP is stating its requirements for SIP in the IP multimedia subsystem of 3G. If these requirements are not included in the IETF specifications the risk of SIP fragmentation may come true.

References

[1] Internet Draft, SIP WG, Handley/Schulzrinne/Schooler/Rosenberg: SIP: Session Initiation Protocol, November 24, 2000, http://www.ietf.org/internet-drafts/draft-ietf-sip-rfc2543bis-02.txt

[2] RFC2327, Network WG, M. Handley, V. Jacobson: SDP: Session Description Protocol, April 1998, http://www.ietf.org/rfc/rfc2327.txt

[3] RFC2617, Network WG, J. Franks, et al: HTTP Authentication: Basic and Digest Access Authentication, June 1999, http://www.ietf.org/rfc/rfc2617.txt

[4] RFC2440, Network WG, J. Callas, et al: OpenPGP Message Format, November 1998, http://www.ietf.org/rfc/rfc2440.txt

[5] Internet Draft, SIP WG, J.Rosenberg, H.Schulzrinne: Guidelines for Authors of SIP Extensions, March 5, 2001, http://www.ietf.org/internet-drafts/draft-ietf-sip-guidelines-02.txt

[6] Internet Draft, SIP WG, J.Rosenberg,H.Schulzrinne: Reliability of Provisional Responses in SIP, March 2, 2001, http://www.ietf.org/internet-drafts/draft-ietf-sip-100rel-03.txt

[7] Internet Draft, SIP WG, S.Donovan, J.Rosenberg: The SIP Session Timer, November 22, 2000, http://www.ietf.org/internet-drafts/draft-ietf-sip-session-timer-04.txt

[8] Internet Draft, SIP WG, W. Marshall, et all: SIP Extensions for supporting Distributed Call State, February, 2001, http://www.ietf.org/internet-drafts/draft-ietf-sip-state-01.txt

[9] Internet Draft, SIP WG, Schulzrinne/Rosenberg: SIP Caller Preferences and Callee Capabilities, November 24, 2000, http://www.ietf.org/internet-drafts/draft-ietf-sip-callerprefs-03.txt

[10] Internet Draft, SIP WG, W. Marshall, et al: Integration of Resource Management and SIP, February, 2001, http://www.ietf.org/internet-drafts/draft-ietf-sip-manyfolks-resource-01

[11] Internet Draft, R. Sparks: SIP Call Control – Transfer, February 26, 2001, http://www.ietf.org/internet-drafts/draft-ietf-sip-cc-transfer-04.txt

[12] Internet Draft, SIP WG, W. Marshall, et al: SIP Extensions for Caller Identity and Privacy February, 2001, http://www.ietf.org/internet-drafts/draft-ietf-sip-privacy-01.txt

[13] Adam Roach: Event Notification in SIP, Internet Draft, February 2001, http://www.ietf.org/internet-drafts/draft-roach-sip-subscribe-notify-03.txt

[14] RFC2976, Network WG, S. Donovan: The SIP INFO Method, October 2000, http://www.ietf.org/rfc/rfc2976.txt

[15] RFC2778, Network WG, M. Day, J. Rosenberg, H. Sugano: A Model for Presence and Instant Messaging, February 2000, http://www.faqs.org/rfcs/rfc2778.html

[16] Internet Draft, SIMPLE WG, Rosenberg et al: SIP Extensions for Presence, March 2, 2001, http://www.cs.columbia.edu/sip/drafts/draft-rosenberg-impp-presence-01.txt

[17] Internet Draft, IMPP WG, Jonathan Rosenberg et.al: SIP Extensions for Presence Authorization, June 15, 2000, http://www.cs.columbia.edu/sip/drafts/draft-rosenberg-impp-qauth-00.txt

[18] Internet-Draft, J. Rosenberg, et al: SIP Extensions for Instant Messaging, February 28, 2001, http://www.cs.columbia.edu/sip/drafts/draft-rosenberg-impp-im-01.txt