A REVIEW ON SESSION INITIATION PROTOCOL (SIP)

Mandeep kaur*

Aarti Shar**

Abstract---

This paper describes Session Initiation Protocol SIP, its evolution, and the architecture for building the session initiation protocol. SIP is an application-layer control protocol for creating, modifying, and terminating sessions with one or more participants. The functionality performed by session initiation protocol includes the determination of location of target point, media capabilities, availability of target point.This paper also includes the responses, requests messages used for effective communication, the transactions handled by session initiation protocol and also describes the header and algorithm used to describe the working of SIP. This paper also provides the difference between various signaling protocols used in IP networks, and describes which is best in what aspect, and how session initiation can be used in real world applications.

Keywords—session initiation protocols, multimedia communication, quality of service.

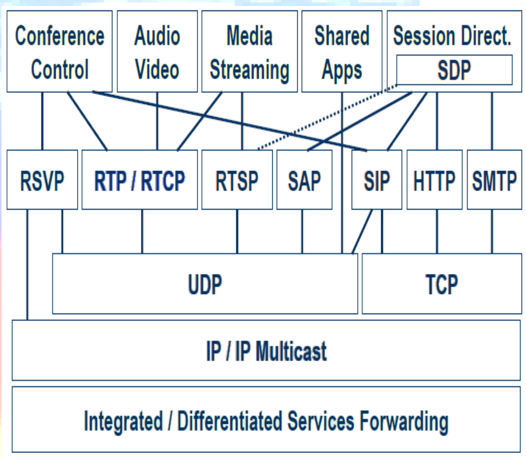
* Department of Information Technology GIMET Amritsar, Punjab, India

** Department of Computer Science, GNDU Amritsar, Punjab, India

I. INTRODUCTION

Session Initiation Protocol is an IETF-defined signaling protocol, widely used for controlling multimedia communication sessions such as voice and video calls over Internet Protocol. These sessions may be text, voice, video or a combination of these. SIP sessions involve one or more participants and can use unicast or multicast communication[8].Session initiation protocol is an Application Layer Protocol that is designed to be independent of the underlying transport layer. It can run on User Datagram Protocol, Transmission Control Protocol, or Stream Control Transmission Protocol which is a text-based protocol, incorporating many elements of the Simple Mail Transfer Protocol and Hypertext Transfer Protocol. Session initiation protocol is not a vertically integrated communications system.

It is rather a component that can be used with other IETF protocols to build a complete multimedia architecture[5]. The IETF architecture is as shown in figure. Session initiation protocol is based on HTTP protocol. The HTTP protocol has inherited the format of message headers from RFC822. HTTP is probably the most successful and widely used protocol in the Internet. In fact, HTTP can also be classified as a signaling protocol, because user agents use the protocol to tell a HTTP server in which documents they are interested in. SIP is used to carry the description of session parameters; the description is encoded into a document using Session direct protocol. Both protocols, HTTP and SIP have inherited encoding of message headers from RFC822[3].



All Session initiation protocol-based communication sessions share at least three typical separate activities and protocols such as:

- SIP provides the basic signaling between participants to set up the session.
- SIP uses the Session Description Protocol (SDP) to classify the nature of the communication utilize within session.
- SIP uses the suitable protocol to convey information in the session.

II. HISTORY OF SIP

Initially only the traditional switch-based telephone system was the main medium for transmitting messages. With the advent of the Internet, the need was felt to fabricate a system, which connects people over the IP based network. Different communities put forward different solutions but the solution presented by IETF was finally accepted as the most general one. This solution was the development of Session initiation protocol. However the development of SIP in IETF was not a one-step process. This was developed in various versions[6].The SIP approach exemplifies classic Internet-style innovation, build only what is needed, to address only what is lacking in existing mechanisms. Because the SIP approach is modular and free from underlying protocol or architectural constraints, and because the protocols themselves are simple, SIP has caught on as an alternative to H.323 and to vendor-proprietary mechanisms for transporting SS7 protocols over IP[1].

SIP was originally designed in the form of Session Invitation Protocol (SIP) by M.Handley, E.Schooler and Simple Conference Invitation Protocol (SCIP) by H.Schulzrinne. The original drafts of the SIP standard were started in February 1996 by IETF organization. These internet drafts of SIP were originally produced to create a mechanism for inviting people to large-scale multipoint conferences on the Internet Multicast Backbone (Mbone). At this stage, IP telephony did not really exist.

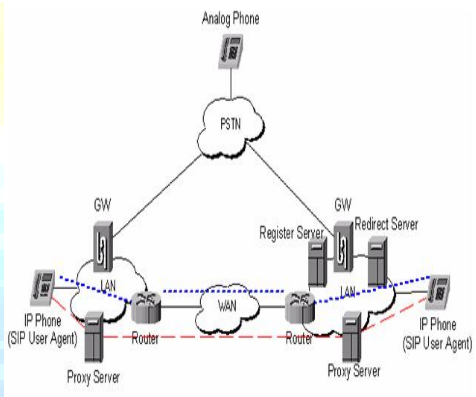
The first draft of SIP titled as "draft-ietf-mmusic-sip-00," included only one request type, which was a call setup request[6]&[3]. In December 1996, this draft was revised and a new draft named as "draft-ietf-mmusic-sip-01" was produced. This draft would still be unrecognizable to most people as the precursor to SIP. Eleven versions and 3 years later, the draft took shape as the SIP with which people are now familiar. The IETF published this draft, which was called "draft-ietf-mmusic-sip-12," in January 1999. It contained the six SIP requests. From "draft-ietf-mmusic-sip-12" to SIP’s publication as RFC 2543 in March 1999 was a small step. It was modified further to generate the more modern version of RFC 3261.

The latest version of the specification is RFC 3261 from the IETF Network Working Group. In November 2000, SIP was accepted as a 3GPP signaling protocol and permanent element of the IP Multimedia Subsystem (IMS) architecture for IP-based streaming multimedia services in cellular systems[6].

SIP revisions have been ongoing in recent years. Perhaps more important, new working groups have been formed to exploit SIP’s potential. Most notable is the SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE) Working Group. This working group expanded the original SIP specification so that it included the delivery of Instant Messaging (IM) information through the use of the MESSAGE request. Primarily, this request is the mechanism by which the Microsoft Real Time Communications (RTC) Server provides its IM capability[3].

III SIP ARCHITECTURE

SIP is a request-response protocol that works in the Application layer of the Open Systems Interconnection (OSI) communications model, and provides the capability to determine the location of the target end point, determine the media, capabilities of the target end point, determine the availability of the target end point establish a session between the originating and target end point, and handle the transfer and termination of calls[7]. The architecture of session initiation protocol has two basic components as shown in figure:



A. THE SIP USER AGENT (UA)

It is the endpoint component, which can be represented by a hardware or software device implementing SIP[9]. For example an IP phone. It consists of two main components:

I. USER AGENT CLIENT (UAC)

It initiates the calls. The UAC is an application that initiates up to six feasible SIP requests to a user agent server. The six requests issued by the UAC are: INVITE, ACK, OPTIONS, BYE, CANCEL and REGISTER[14].

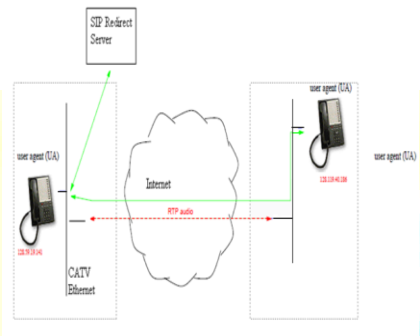
When the SIP session is being initiated by the UAC SIP component, the UAC determines the information essential for the request, which is the protocol, the port and the IP address of the UAS to which the request is being sent.

This information can be dynamic, which will make it challenging to put through a firewall. The UAC is also capable of using the information in the request URI to establish the course of the SIP request to its destination, as the request URI always specifies the host which is essential.

January 2015 IJMIE Volume 5, Issue 1 ISSN: 2249-0558

II. USER Agent Server (UAS)

It answers the calls. UAS is the Server that hosts the application responsible for receiving the SIP requests from a UAC, and on reception returns a response to the request back to the UAC. The UAS may issue multiple responses to the UAC, not necessarily a single response. Communication between UAC and UAS is client/server and (peer-to–peer)[7]. The peer to peer communication between user agent client and user agent server is as shown in figure 3.2.



B. THE SIP NETWORK SERVER

It handles signaling associated with multiple calls providing name resolution and user location. It consists of three main groups namely SIP register server, SIP redirect server, SIP register server. These three components are as follows:

1. **IP REGISTER SERVER**

It receives registration messages from endpoints regarding current user location and maps the SIP addresses with the physical location in the domain where the endpoint is located. These mapping data are stored in a database, which can reside on the same machine or on a remote server. The Registrar server makes it possible for users to alter the address at which they are contactable. This is possible through the SIP client sending a REGISTER request of change of an address to the registrar server, which then accepts the request and records the user's new address[3].

There are two ways in which the SIP clients can contact the registrar server. The first way is through a direct approach, by utilizing information that is configured into the client. Secondly through an indirect approach, which uses the multicast address to contact the registrar server.

2. SIP PROXY SERVER

It forwards the SIP messages to multiple proxy servers, creating a search tree, in order for the SIP messages to reach their destination. There are two different operating modes for these servers:

1. **STATELESS**:

   The server forgets all the information once the request is sent.

2. **STATEFUL**:

   The server save previous routing information and is able to use it for improving the message transfer.

The proxy server mostly acts as mediator that services the requests or forwards them to other UASs or UACs for servicing. Proxy server can use an intra-organizational configuration through which to route all its sip communications. Intra-organizational configuration can be described when users messages are routed through a proxy server before the messages are relayed to the destination SIP client. This occurs when initiating a SIP session to another user within the same organization. This can be useful for internal communication where security over an internet link can be a problem[4]&[7].

3. SIP REDIRECT SERVER

It helps endpoints to find the desired address by redirecting them to try another server. The redirect server allows for redirection which enables users to temporarily change geographic location and still be contactable through the same SIP identity. In the future this will be the way that telephone communications will work and with the arrival of wireless is an accommodating way to enable the client to be handed over from server to serve as the user moves around. The RTC server implements the proxy server and the redirect server on one server. A server with combined functions is called a SIP server. Determination of how the SIP messages will be processed, that is whether the messages go to the proxy or the redirect server, is determined through the configuration settings on the SIP server. Using this technology it is also possible to keep the service running while some of the servers are being worked on and maintained[2]&[14].

IV.FUNDAMENTAL ASSUMPTIONS:

The various fundamental assumptions are as follows:

A. REUSING EXISTING PROTOCOLS

SIP was designed to specifically reuse as many existing protocols and protocol design concepts. For example, SIP was modeled after HTTP, using URLs for addressing and SDP to convey session information.

B. MAXIMIZING INTEROPERABILITY

SIP was also designed so that it would be easy to bind SIP functions to existing protocols and applications, such as e-mail and Web browsers. SIP does this by limiting itself to a modular philosophy just like many other Internet protocols and focusing on a specific set of functions[9].

IV Functionality of Session layer Protocol

SIP is an application-layer control protocol that can establish, modify, and terminate multimedia sessions such as Internet telephony calls. SIP can also invite participants to already existing sessions, such as multicast conferences. Media can be added to an existing session. SIP transparently supports name mapping and redirection services, which supports personal mobility. Users can maintain a single externally visible identifier regardless of their network location[10].

The various functions performed by session initiation protocol are as follows:

A. Determine the location of the target end point:

SIP supports address resolution, name mapping, and call redirection.

B. Determine the media capabilities of the target end point:

By using Session Description Protocol (SDP); SIP determines the "lowest level" of common services between the end points. Conferences are established using only the media capabilities that can be supported by all end points.

C. Determine the availability of the target end point :

If a call cannot be completed because the target end point is unavailable, SIP determines whether the called party is already on the phone or did not answer in the allotted number of rings. It then returns a message indicating why the target end point was unavailable.

D. Establish a session between the originating and target end point:

If the call can be completed, SIP establishes a session between the end points. SIP also supports mid-call changes, such as the addition of another end point to the conference or the changing of a media characteristic or codec.

E. Handle the transfer and termination of calls:

SIP supports the transfer of calls from one end point to another. During a call transfer, SIP simply establishes a session between the transferee and a new end point (specified by the transferring party) and terminates the session between the transferee and the transferring party. At the end of a call, SIP terminates the sessions between all parties.

SIP normally runs over UDP or TCP, but it can run over other protocols such as IP, ATM, or X.25. It requires only a datagram service and is independent of the packet layer. It can provide "out-of-band" call setup services in which the SIP exchanges take place over UDP or TCP, but actual data transmission takes place over the public telephone network[11].

V Session Initiation Message

SIP is a text-based protocol with syntax similar to that of HTTP. There are two different types of SIP messages: requests and responses. The first line of a request has a method, defining the nature of the request, and a Request-URI, indicating where the request should be sent. The first line of a response has a response code[12]

A. **SIP Request Messages**

The various SIP request messages are as follows:

B. **INVITE**

It initiates the session. The session description is included in the message body. Re-INVITE is used to change session state. Thus it is used to establish a media session between user agents.

C. **ACK**

It confirms the session establishment and can be used only with INVITE, i.e. reliable message exchanges

D. **BYE**

It terminates the session.

E. **CANCEL**

It cancels a pending INVITE.

F. **OPTIONS**

It includes the capability inquiry, i.e. requests information about the capabilities of a caller, without setting up a call.

G. **REGISTER**

It binds a permanent address to current location and it may convey user data. It is used by a UA to indicate its current IP address and the URLs for which it would like to receive calls[13].

The tabular representation of these messages is as shown in figure.

SIP Method	Description
INVITE	Invites a user to a call
ACK	Used to facilitate reliable message exchange for INVITEs
BYE	Terminates a connection between users or declines a call
CANCEL	Terminates a request, or search, for a user
OPTIONS	Solicits information about a server's capabilities
REGISTER	Registers a user's current location
INFO	Used for mid-session signaling

V. SIP Message Response

The various SIP response messages are as follows:

A. **PROVISIONAL (1xx)**

It indicates that the request is received and is being processed.

B. **SUCCESS (2XX)**

It tells that the action was successfully received, understood, and accepted.

C. **REDIRECTION (3XX)**

It indicates whether the further action needs to be taken to complete the request.

D. **CLIENT ERROR (4XX)**

It tells that the request contains bad syntax or cannot be fulfilled at the server.

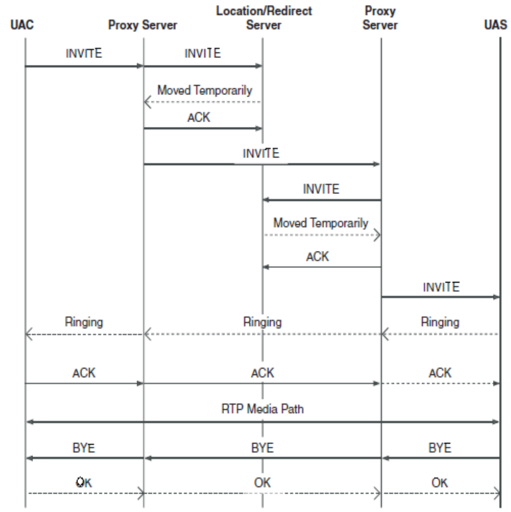
E. **SERVER ERROR (5XX)**

It defines that the server failed to fulfill an apparently valid request.

F. **GLOBAL FAILURE (6XX)**

The request cannot be fulfilled at any server is indicated by it.

The call setup procedure in session initiation protocol is as shown in figure.



If a call is to be routed through a number of different Proxy servers Redirect server is used. When a caller UA sends an INVITE request to the redirect server, the redirect server contacts the location server to determine the path to the called party, and then the redirect server sends that information back to the caller. The caller then acknowledges receipt of the information. The caller then sends a request to the device indicated in the redirection information (which could be the callee or another server that will forward the request). Once the request reaches the callee, it sends back a response and the caller acknowledges the response. RTP is used for the communication between the caller and the callee[12]&[13].

VI. Transactions

SIP makes use of transactions to control the exchanges between participants and deliver messages reliably. The transactions maintain an internal state and make use of timers. Client Transactions send requests and Server Transactions respond to those requests with one-or-more responses. The responses may include zero-or-more Provisional (1xx) responses and one-or-more final (2xx-6xx) responses[3].

Transactions are further categorized as either Invite or Non-Invite. Invite transactions differ in that they can establish a long-running conversation, referred to as a Dialog in SIP, and so include an acknowledgment (ACK) of any non-failing final response. For example 200 OK, as shown in figure.

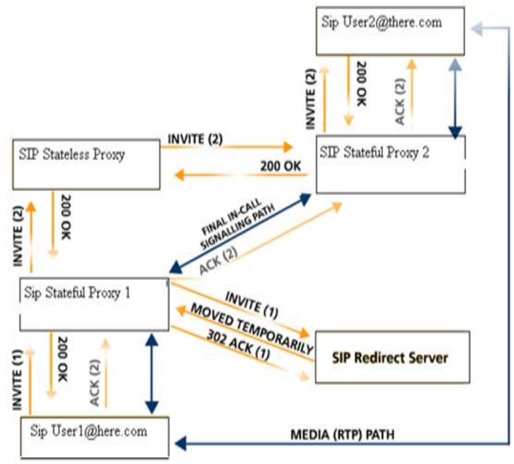


Figure : Transactions

Because of these transactional mechanisms, SIP can make use of un-reliable transports such as User Datagram Protocol (UDP). As in the above example, User1's UAC uses an Invite Client Transaction to send the initial INVITE (1) message. If no response is received after a timer controlled wait period the UAC may have chosen to terminate the transaction or retransmit the INVITE. However, once a response was received, User1 was confident the INVITE was delivered reliably. User1's UAC then must acknowledge the response. On delivery of the ACK (2) both sides of the transaction are complete. And in this case, a Dialog may have been established[1].

IX. Session Initiation Protocol Header

The header of session initiation protocol is of 32 bits. It comprises of various fields like version, flow label, payload length, payload type, hop limit, source address, and destination address as shown in figure.[15]

The various fields of the session initiation protocol are as follows:

A. VERSION

The version field in SIP distinguishes SIP from IP and any other internet protocol that uses the same link layer frame as IP such as Stream Protocol [17].

B. FLOW LABEL

The Flow Label is an expansion of the IP Service field. It is used to label packets as belonging to a particular traffic flow for which the server requires specific handling, for example real time service or non default quality of service[4].

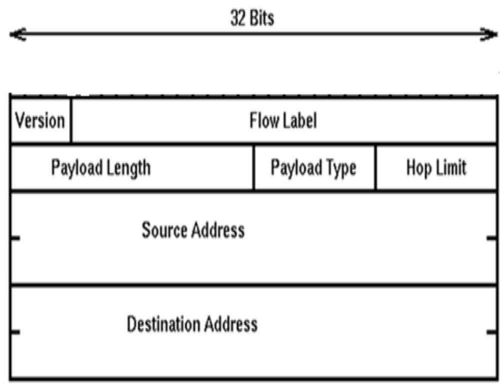


Figure : SIP Header

C. **PAYLOAD LENGTH**

This specifies the length of the SIP packet. It does not include the header. This is a change from IP, where the receiver had to check that the payload length was not less than the header size, this is not necessary in SIP, and results in one less thing to be checked[2].

D. **PAYLOAD TYPE**

This uses the same values as the IP Protocol field. It specifies the type of the header immediately following the SIP header, such as TCP or UDP. It has been renamed to avoid confusion as to what is being referred to as the IP protocol - the protocol field or IP itself[3].

E. **HOP LIMIT**

The Hop Limit is set to some nonzero value, and decremented by one by each system that forwards the packet. The packet is discarded if the hop limit reaches zero. This is to prevent the packet getting stuck in a forwarding loop.

Other uses include limiting the propagation of multicast packets, and it can also be used for diagnostic purposes. The "time to live" field in IP provided the same function, plus one extra one. This was to limit the amount of time that a packet spent in transit. This was discarded because it proved too costly to implement, and in some cases impossible to implement, for example in large subnets whose transit time is unpredictable. In practice many IP routers implemented time to live as hop limit, SIP legitimized this. Any higher level functions that cannot tolerate delivery delays, must provide their own method of recognizing old packets[12].

F. **SOURCE ADDRESS**

This is the 64 bit address of the sender. SIP addressing does not impose any strict class structure on addresses.

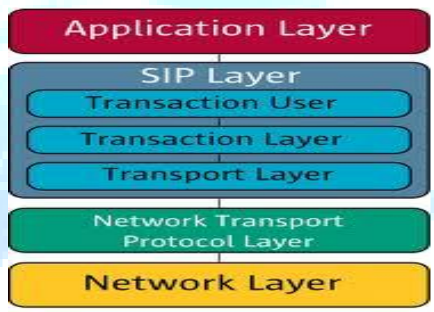
G. **DESTINATION ADDRESS**

This is the 64 bit address of the destination. SIP addressing does not impose any strict class structure on addresses[16].

X. Session Initiation Protocol Layer

SIP is structured as a layered protocol, which means that its behavior is described in terms of a set of fairly independent processing stages with only a loose coupling between each stage. The protocol behavior is described as layers for the purpose of presentation, allowing the description of functions common across elements in a single section. It does not dictate an implementation in any way. When an element "contains" a layer, it is compliant to the set of rules defined by that layer[24].

Not every element specified by the protocol contains every layer. Furthermore, the elements specified by SIP are logical elements, not physical ones. A physical realization can choose to act as different logical elements, perhaps even on a transaction-by-transaction basis[18].



LAYERS:

The layers of session initiation protocol are as follows:

A. **SYNTAX AND ENCODING**

The lowest layer of SIP is its syntax and encoding. Its encoding is specified using an augmented Backus-Naur Form grammar (BNF)[26]

B. **TRANSPORT LAYER**

The second layer is the transport layer. It defines how a client sends requests and receives responses and how a server receives requests and sends responses over the network. All SIP elements contain a transport layer[2].

C. **TRANSACTION LAYER**

The third layer is the transaction layer. Transactions are a fundamental component of SIP

A transaction is a request sent by a client transaction (using the transport layer) to a server transaction, along with all responses to that request sent from the server transaction back to the client. Any task that a user agent client (UAC) accomplishes takes place using a series of transactions. Transaction layer handles application-layer retransmissions, matching of responses to requests, and application-layer timeouts.

Stateless proxies do not contain a transaction layer. User agents contain a transaction layer, as do Stateful proxies. The transaction layer has a client component (referred to as a client transaction) and a server component (referred to as a server transaction), each of which are represented by a finite state machine that is constructed to process a particular request [4]&[24].

D. **TRANSACTION USER LAYER**

The layer above the transaction layer is called the transaction user (TU). Each of the SIP entities, except the stateless proxy, is a transaction user.

When a TU wishes to send a request, it creates a client transaction instance and passes it the request along with the destination IP address, port, and transport to which to send the request. A TU that creates a client transaction can also cancel it. When a client cancels a transaction, it requests that the server stop further processing, revert to the state that existed before the transaction was initiated, and generate a specific error response to that transaction. This is done with a CANCEL request, which constitutes its own transaction, but references the transaction to be cancelled[18].

In the example shown in figure 8.1, the rejection of the first INVITE request, followed by a valid INVITE request, enables the analysis of the processing of the ACK for these two situations.

XI. Future Scope

SIP IS THE FUTURE OF TELECOMMUNICATIONS….

Session Initiation Protocol (SIP) isn't just a protocol that is "popular" in Voice over IP (VOIP) environments. It is, for all intents and purposes, THE VOIP protocol, at least among traditional telecommunications providers. It is the protocol that networks choose when they want to IP-enable their largely SS7 environments (SS7 is the signaling protocol used in the global circuit-switched network used to communicate within and between almost every telecommunications company in existence). SIP plays a central role in the IP Multimedia Subsystem (IMS), a family of protocols which is supposed to define the architecture of next-generation mobile networks capable of streaming various kinds of text, voice and video data to mobile phone subscribers even as they roam between networks. By paying for a SIP trunk connection, you don't need to have any kind of phone line hooked into your office. You just pay for a SIP connection from the SIP trunk provider of your choice, and your SIP-compliant software has all the telephony connectivity it needs.

One of the advantages of this model is that you aren't tied to just a number in your local area or country. Given that SIP abstracts away the concept of where you are, it is easy to have numbers in multiple countries that feed back into your central office. This gives small companies the ability to seem like much larger companies, something from which my company, which is extremely small, could certainly benefit, but from which even home users can benefit.

SIP trunking also allows you to escape reliance on the owner of the phone or cable lines that run into your home for voice service, moving beyond cable-based VOIP services, and even consumer services like Vonage, to allow a web of VOIP providers to compete for your home phone business.

SIP trunk connections are likely to find their way into end-users homes once businesses - a group that serves as the canary in the mine for most new technologies - have worked out the kinks.

XII. Conclusion

In this paper, the session initiation protocol, its architecture, functionality is discussed. SIP is an application-layer control protocol for creating, modifying, and terminating sessions with one or more participants.SIP user agent which may be user agent client or user agent server, SIP network server are the components of session initiation protocol. SIP network server includes SIP register server, SIP proxy server, and SIP redirect server. SIP performs various functions like determination of locations of the target end point, media capabilities, availability of target end point, establishing sessions between the originating and target end point, handling the transfer, and termination of calls. Session initiation protocol has two different types of messages, SIP request and SIP response messages. SIP request messages are INVITE, ACK, BYE, CANCEL, and OPTIONS. SIP response messages include PROVISIONAL, SUCCESS, REDIRECTION, CLIENT ERROR, and GLOBAL FAILURE.

SIP makes use of transactions to control the exchanges between participants and deliver messages reliably. The header of session initiation protocol is of 32 bits. It comprises of various fields like version, flow label, payload length, payload type, hop limit, source address, and destination address. Session initiation protocol offers a wide variety of real world applications.

Session initiation protocol facilitates the voice communication over VOIP, but still there are still major problems such as unavailability of floor control, minimal capability exchange, undefined procedures for handling failures, and is not required for SIP multicast conferences. It provides limited support for video, and do not provide any support for data conferencing protocols.

References

[1] Perkins CE, Royer EM. Ad hoc on-demand distance vector routing. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, 1999.

[2] Perkins CE, Belding-Royer E, Das SR. Ad hoc on-demand distance vector (AODV) routing. http://www.ietf.org/rfc/rfc3561. txt, July 2003. RFC 3561.

[3] Perkins CE, Bhagwat P. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of ACM Sigcomm*, 1994.

[4] Raju J, Garcia-Luna-Aceves JJ. A new approach to on-demand loop-free multipath routing. In *Proceedings of Int'l Conference on Computer Communications and Networks (IC3N)*, 1999.

[5] Marina MK, Das SR. Routing performance in the presence of unidirectional links in multihop wireless networks. In *Proceedings of ACM MobiHoc*, 2002.

[6] Nasipuri A, Castaneda R, Das SR. Performance of multipath routing for on-demand protocols in mobile ad hoc networks. *ACM/Kluwer Mobile Networks and Applications (MONET)* 2001; **6**(4): 339–349.

[7] Ogier R, Shacham N.Adistributed algorithm for .nding shortest pairs of disjoint paths. In *Proceedings of IEEE Infocom*, 1989.

[8] Papadimitratos P, Haas ZJ, Sirer EG. Path set selection in mobile ad hoc networks. In *Proceedings of ACM MobiHoc*, 2002.

[9] Marina MK, Das SR. Performance of route caching strategies in dynamic source routing. In *Proceedings ofWorkshop onWireless Networks and Mobile Computing (WNMC) in conjunction with International Conference on Distributed Computing Systems (ICDCS)*, 2001.

[10] Samar P, Wicker SB. On the behavior of communication links of a node in a multi-hop mobile environment. In *Proceedings of ACM MobiHoc*, 2004.

[11] Tsirigos A, Haas ZJ. Multipath routing in the presence of frequent topological changes. *IEEE Communications Magazine* 2001; **39**(11): 132–138.

[12] Sidhu D, Nair R, Abdallah S. Finding disjoint paths in networks. In *Proceedings of ACM Sigcomm*, 1991.