Measurements & check the Performance of Secure RFC2961 Protocol

Rachana Kamble, R.K Pateriya

Abstract—RSVP (Resource ReSerVation Protocol) is an Internet protocol which is allowing applications reserving network resources. RSVP is used as a general purpose signaling control in the MPLS and Traffic Engineering areas. This paper describes our research on the Extension of RSVP (RFC2961) protocol overhead and applied security authentication by ESP (Encapsulating Security Payload) after then check the performance while sending messages on to the network. We specify network-layer protocol overhead and monitor the effects of increased modularity and security by use of ESP. We implement RSVP (RFC 2961 standard) and used ESP for security authentication and study its performance in a RedHat 7.0 Linux OS testbed. An ESP node helping to provide security for signaling sessions is found to consume small amounts of CPU time and memory. Individual routines in the ESP code are instrumented to obtain a detailed profile of their contributions to the overall system processing. Important factors in determining performance, such as the number of sessions, state management, refresh reduction capable bit, RSVP bundle message, summary refresh extension, timer management and signaling message size are further discussed. The IP Encapsulating Security Payload (ESP) Header provides integrity, authentication, and confidentiality to IP datagram. It does this by encapsulating either an entire IP datagram or only the higher-layer protocol (e.g., RSVP protocol) data inside the ESP, encrypting most of the ESP content, and then appending a new IP header to the now encrypted ESP Payload. This new IP header carries the protected data through the internetwork. Our work is based on RFC2961. The main idea of RFC2961 is to send a probe message from a source router in a domain to a destination router in another domain. The probe is passing from domain to domain through the network.

Index Terms— RSVP, RFC 2961, ESP (ENCAPSULATING SECURITY PAYLOAD) ,Performance Evaluation.

1 INTRODUCTION

THIS paper is based on security over RFC 2961 standard (Extension RSVP). RSVP is developed by IETF (Internet Engineering Task Force) proposed standard signaling protocol [1]. It is enabling unicast or multicast sending and receiving applications to full fill QoS (Quality of Service) requirements which is arrived on Internet nodes in the network path. The RSVP has designed in a manner that allows managing QoS information defined by the IntServ (Integrated Service) architecture's. RSVP has considered as a signaling protocol in the MPLS architecture as a label distribution protocol.

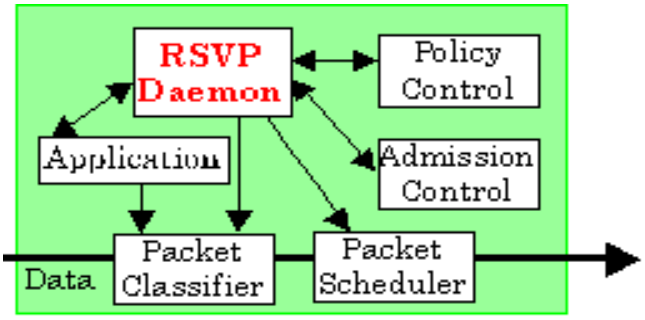


Fig.1. RSVP Daemon Network

RSVP is a "soft state" protocol; i.e., it maintains state in each router or host. State needs to be periodically refreshed thus Refresh Messages are required.

The general acceptance of the protocol has been slowed mainly by concerns about its scalability. The load generated thousands of RSVP sessions. The IETF RSVP Working Group believed that it was necessary to introduce some security provision extensions to the protocol which increase utilization with a large number of flows. The "RSVP Refresh Overhead Reduction Extensions" gathered in this research work. This RFC 2961 accomplish the inherent flexibility of the protocol and extends to reduce the dependency of the total overhead of the RSVP signaling on the number of RSVP sessions, though conserving the original flows isolation. Such techniques may also be used in the MPLS/Traffic Engineering. So that it is an important aspect to assessing the RSVP suitability as a signaling protocol for a large number of unicast flows. This paper discusses several operations of RFC2961 and their measurements of their influence in a real network. For providing the enhanced modularity and security we used ESP (ENCAPSULATING SECURITY PAYLOAD). The Encapsulating Security Payload (ESP) Header provides integrity, authentication, and confidentiality to IP datagram. It is encapsulating entire IP datagram. In this paper we investigate security authentication to improve message reliability over RFC 2961.

2 RELATED WORK

2.1 RSVP Description

The control messages in RSVP, Path and Resv messages is Created by the senders and receivers, respectively.

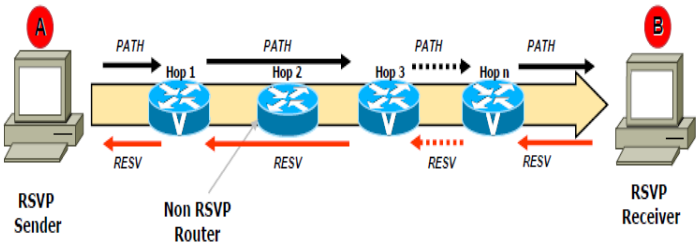


Fig.2. RSVP and Path Message.

- Rachana Kamble, (MTech Scholar), Department of Computer Science & Engineering, MANIT Bhopal (M.P), India, rachanakamble@gmail.com
- R.K Pateriya, Department of Information Technology, MANIT Bhopal (M.P), India, r_k_pateriya@indiatimes.com

Path messages follow the route computed by the routing protocol and provide receivers with the description of the .sender and traffic Row. Upon receipt of a valid Path message, each intermediate RSVP-capable router updates or creates a Path state entry for the sender before forwarding the appropriately updated Path message towards the receiver. After receiving a Path message, the receiver can make a reservation by sending a Resv message back to the source. RSVP is a sofl-state protocol. Hence, Path and Resv messages must be exchanged regularly between the routers in order to maintain the reservation. The frequency with which Path and Resv messages are sent is determined by the soft-state refresh period. Although the Path state and the Resv state will eventually timeout if not refreshed, PathTear or ResvTex messages may be used to tear down state promptly.

2.2 Related Work on Soft state Protocols

Some previous studies on soft-state protocols are suitable to RSVP and are relevant to the work in this paper. Scalable timers, where the refresh period increases proportionally with the amount of state to be refreshed. The main performance considered was the probability of the sender and receiver having consistent stale. A key finding was that incorporating feedback into soft-state protocols improves the network consistency without incurring excessive network resource consumption. Another analytical model for soft state signaling protocols was presented, allowing the comparison of a spectrum of signaling protocols from "pure" soft-state to soft-state augmented with explicit state removal and/or reliable signaling to "pure" hard-state protocols.

2.3 Related Work on RSVP Performance Evolution & Improvement

The performance of RSVP has been measure, using an industrial-strength RSVP implementation on a commercial IP router El 71. Performance metrics considered included the connection set up time, soft-state refresh overhead and the impact of real-time packet scheduling. The important thing was designing the RSVP protocol engine optimally and using the resulting RSVP implementation, the performance of RSVP was studied. Interestingly, the results suggested that the scalability of RSVP is better than is traditionally assumed [lS].

2.4 RSVP Extensions

A number of protocol improvements have been suggested to increase the performance characteristics of RSVP operations. An initial proposal to speed up the service establishment time in the presence of occasional packet loss and to reduce steady-state refresh signalling overhead has been made. One of the drawbacks of this approach is the requirement to change the protocol specification and to introduce an additional confirmation message security into RSVP. Which also deals with the general issue of reliability of RSVP messages, e.g., in case a service invocation is torn down. Instead of refreshing all the state information, neighbouring RSVP nodes only need to exchange ‘heartbeats’ denoting their liveliness. A slightly different suggestion addressing the same issue even more stringently is currently developed within the IETF RSVP working group. This mechanism addresses further details, such as how to discover a very short-termed node failure. It is beyond the scope of this work to rate these different techniques. However, they clearly bear the potential to drastically reduce RSVP’s processing requirements for steady-state refresh signalling. This eliminates one of the major performance limitations of the current RSVP specification. Other RSVP extensions, which are in the process of being standardized, encompass diagnostic messages, inter-operation with IP tunnels, cryptographic authentication and user identity representation.

3 Encapsulating Security Payload

Encapsulating Security Payload (ESP) is very much used for providing integrity and confidentiality for RSVP Extensions.

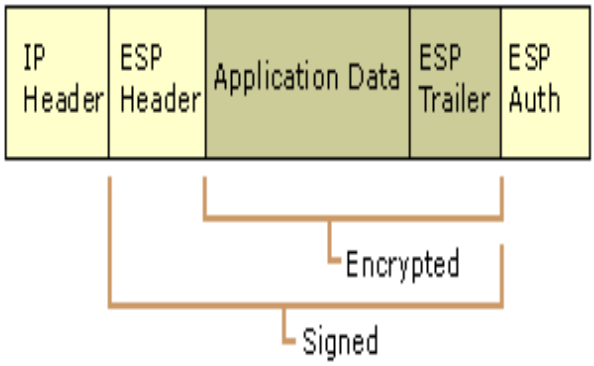


Fig.3. ESP used in RFC2961 and provide Encryption for Message

ESP is inserted after the IP header and before an upper layer protocol, such as RSVP or before any other IPSec headers that have already been inserted. Everything following ESP (the upper layer protocol, the data, and the ESP trailer) is signed. The IP header is not signed, and therefore not necessarily protected from modification. The upper layer protocol information, the data, and the ESP trailer are encrypted. In this paper we design to evaluate ESP Header Format used for authenticating RFC2961

TABLE 1
Shows the ESP Header Format

EP Header	Description
Security Parameters Index (SPI)	A 32-bit value field that identifies the SA for this datagram relative to the Destination
Sequence Number	. A 32-bit field that contains a counter value (sequence number). Before a cycle arises, the counter is reset by establishing a new SA thus a new key. This field is optional depending on whether anti-replay service is required.
Initialization Vector.	An unfixed-length field only required by certain encryption / decryption algorithms.
Payload Data	An unfixed-length field that contains data.
Padding.	A field for padding (margin-filling) Payload Data field if confidentiality is required, since then the block-size requirement for certain encryption/decryption algorithm has to be met.
Pad length.	A 8-bit field that identifies the size of the padding.
Next Header.	An 8-bit field that identifies the type of data contained in the Payload Data field.
Authentication Data.	An unfixed-length field that contains an Integrity Check Value (ICV) computed over the ESP packet (of course not including the field itself.) The mandatory-to-implement authentication algorithms, DES in Cipher-Block chaining (CBC).

3.1 Benefits of RFC2961 after providing ESP, Message Authentication

- **Improved Security**

The RFC2961 Message Authentication feature greatly reduces the chance of an RSVP-based spoofing attack and provides a secure method to control QoS access to a network.

- **Multiple Environments**

The RSVP Message Authentication feature can be used in traffic engineering (TE) and non-TE environments as well as with sub-network bandwidth manager (SBM).

- **Multiple Platforms and Interfaces**

The RFC2961 Message Authentication feature can be used on any supported RSVP platform or interface.

4. Description of the RSVP Overhead Reduction Extensions

About RFC 2961, Refresh Overhead Reduction extension defined addresses both the refresh volume, the reliability issues with mechanisms and adjusting refresh rate. Here a Bundle message is defined to reduce overall message handling load. A MESSAGE_ID object is defined to reduce refresh message processing by allowing the receiver to identify without delay an unchanged message.

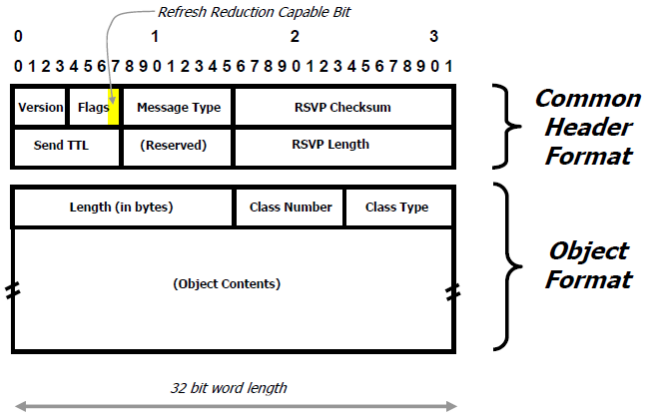


Fig.4. RSVP Extension Header/Objects

A MESSAGE_ACK object is defined which can be used to detect message loss and support reliable RSVP message.

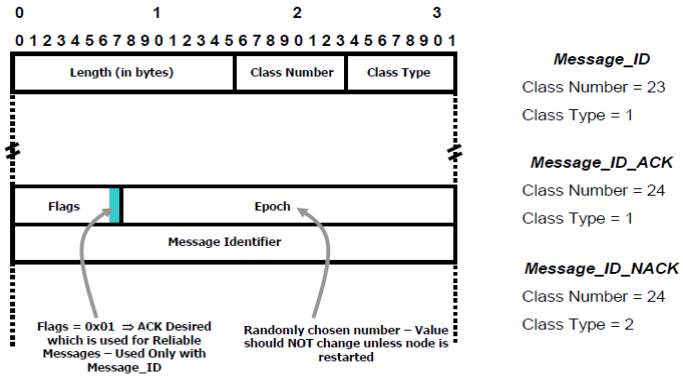


Fig.5. Message ID object Format

4.1. Bundling of Messages

The Bundle message explains in RFC2961, here a number of RSVP messages are bundled and sent to the same RSVP neighbor within a single larger RSVP message. For that a new RSVP Bundle message is defined. Each message has its own header and a body which is made up with a sequence of RSVP messages. The receiving router takes out the sub messages and processes them. The advantages of such mechanism are reduced usage of bandwidth, as a number of IP and datalink headers are replaced by a single one. The term "bundling" is used to avoid confusion with RSVP reservation aggregation. The following subsections define the formats of the bundle header and the rules for including standard RSVP messages as part of the message. RSVP Bundle messages are sent hop by hop between RSVP-capable nodes as "raw" IP datagrams with protocol number 46. The IP source address is an address local to the system that originated the Bundle message. The IP destination address is the RSVP neighbor for which the sub-messages are planned. RSVP Bundle messages SHOULD NOT be sent

with the Router Alert IP option in their IP headers. This is because Bundle messages are addressed directly to RSVP neighbors.

4.2. Refresh by a Message ID Extension

Three physical objects are defined as part of the MESSAGE_ID extension. The physical objects are the MESSAGE_ID object, MESSAGE_ID_ACK object, and the MESSAGE_ID_NACK objects. The first two objects are used to support the summary refresh extension. The MESSAGE_ID object can also simply shorthand indication of when the message carrying the object is a refresh message. Such knowledge can be used by the receiving node to reduce refresh processing requirements. Message identification and acknowledgment is done on a per hop basis. All types of MESSAGE_ID objects include a message identifier. The identifier MUST be unique on a per object generator's IP address basis. More than one MESSAGE_ID should not be included in an RSVP message. Each message containing a MESSAGE_ID object may be acknowledged via a MESSAGE_ID_ACK object, when so indicated. MESSAGE_ID_ACK and MESSAGE_ID_NACK objects may be sent piggy-backed in unrelated RSVP messages or in RSVP Ack messages. RSVP message carrying any of the three object types may be included in a bundle message.

4.3. Summary Refresh Extension

It encourages acknowledgments and reliable RSVP message delivery. The summary refresh extension allows the refreshing of RSVP state without the transmission of standard Path or Resv messages. The benefits of the SRefresh reduces the amount of information that must be transmitted and processed in order to maintain RSVP state synchronization. This extension cannot be used with Path or Resv messages that include any change from previously transmitted messages, example, trigger messages.

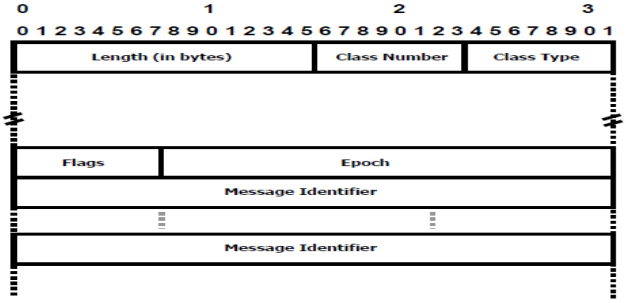


Fig.6. Summary Refresh Message format.

The summary refresh extension executes on the previously (already define) defined MESSAGE_ID extension. A state that was previously mentioned in Path and Resv messages containing MESSAGE_ID objects can be refreshed via the summary refresh extension. The summary refresh extension uses the objects and the ACK message previously defined as part of the MESSAGE_ID extension, and a new Srefresh message. The new message carries a list of Message_Identifier fields corresponding to the Path and Resv trigger messages that established the state. The Message_Identifier fields are carried in one of three Srefresh related objects. The three objects are the MESSAGE_ID LIST object, the MESSAGE_ID SRC_LIST object, and the MESSAGE_ID MCAST_LIST object. The MESSAGE_ID LIST object is used to refresh all Resv state, and Path state of unicast sessions. It is made up of a list of Message_Identifier fields that were originally advertised in MESSAGE_ID objects. The other two objects are used to refresh Path state of multicast sessions. A node receiving a summary refresh for multicast path state will at times need source and group information. These two objects provide this information. The objects differ in the information they contain and how they are sent. Both carry Message_Identifier fields and corresponding source IP addresses. The MESSAGE_ID SRC_LIST is sent in messages addressed to the session's multicast IP address. The MESSAGE_ID MCAST_LIST object adds the group address and is sent in messages addressed to the RSVP next hop. The MESSAGE_ID MCAST_LIST is normally used on point-to-point links. An RSVP node receiving an Srefresh message, matches each listed Message_Identifier field with installed Path or Resv state. All matching state is updated as if a normal RSVP refresh message has been received. If matching state cannot be found, then the Srefresh message sender is notified via a refresh NACK.

5. APPLICATION OF ESP IN RFC2961

For Authenticating RFC2961, ESP provides security concerns are Message integrity and node authentication, User authentication, secure data stream.

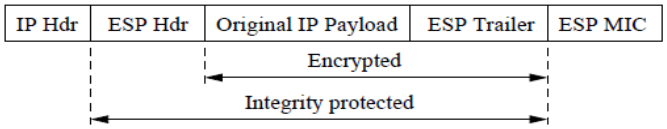
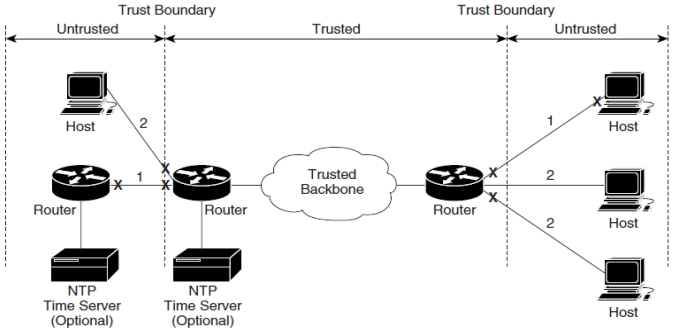


Fig.7. IPSec/ESP (tunneling mode)



1-Known Authentication key; Accepted
2-Does not known key; Rejected
X-RFC2961 Authentication Enabled

Fig.8. RFC2961 Message Authentication Configuration by ESP

In our work we concern with the message integrity and node authentication by ESP. The ESP uses the NTEGRITY object **in** the RSVP message in a hop-by-hop manner. In RFC2961 message authentication algorithm was suggested to use DES in Cipher-Block chaining (CBC). This is one of the most important and widely used cryptographic tools.

For Message authentication the communicating partners are sharing secret key. In DES Cipher-Block-Chaining, message is broken into blocks but these are linked together in the encryption operation. Each previous cipher blocks is chained

with current plaintext block, hence name use Initial Vector (IV) to start process
Ci = DESK1(Pi XOR Ci-1)

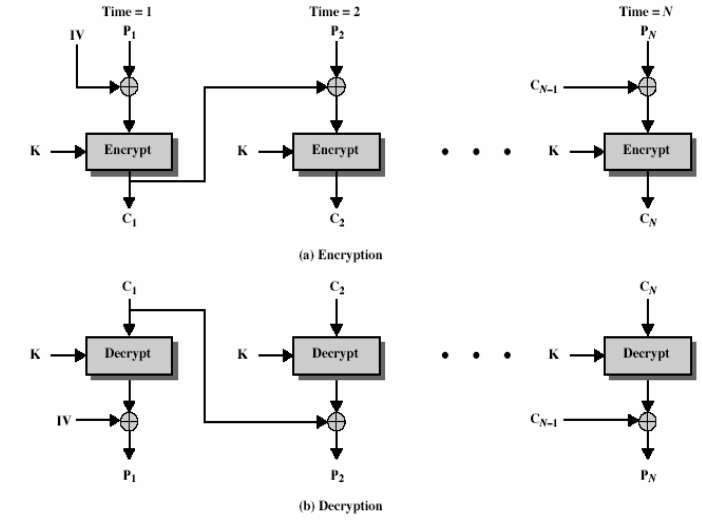


Fig.9. ESP encryption operation

5.1 Advantages of ESP over RFC2961

- It support bulk data encryption and authentication
- each cipher text block depends on all message blocks thus a change in the message affects all cipher text blocks after the change as well as the original block
- Initial Value (IV) should known to sender & receiver an IV must be a fixed value
- Authentication of the two neighboring protocol peers;
- Security association establishment to provide integrity, confidentiality and replay protection for signaling messages exchanged between these entities;
- Denial of service protection;
- Some security protection for the discovery mechanism.

6. PERFORMANCE OF THE RFC2961 WITH SECURITY IMPLICATION

We have implemented the extension of RFC2961 and improving security authentication by ESP (Encapsulating Security payload) and check the performance of secure RFC2961.

6.1 Description of the implementation of Extension RFC2961

We have aware of two, extensive, freely available implementations of the RSVP daemon. The first was ISI distribution [10], has been written at the ISI using the 'C' language was the first implementation of the RSVP protocol. The second was, the KOM RSVP engine [9], has been developed at the Darmstadt University of Technology using the C++ language. The ISI distribution implements all the functionality described in the RFC2205 while the KOM have several deficiency about some features (UDP encapsulation and IPv6 support) that is very important for the implementation. Both are available under a number of Unix OSs like Solaris, We chose Linux 7.0 for the implement of RFC2961 into the KOM RSVP engine and extended into java language. It sustains a higher maximum number of RSVP sessions. The RSVP ISI distribution under FreeBSD maintains a maximum of 5000 sessions while an optimized version of the KOM RSVP is able to maintain up to 50000 sessions.

6.2 Experiment test-bed

The performance experiments are worked out onto two standard PC-based workstations, running RedHat 7.0 Linux OS (kernel 2.2.16), which were configured to serve as routers and equipped as follows: Intel CORE i3 processor, 2Kb second-level cache, 128 Mbyte RAM, 100Mbps Ethernet links, 2interface cards. Monitoring was made by an iMac DV equipped by EtherPeek version 4, used of network monitoring tool. Each test was run with a fixed number of sessions. To create a number of RSVP sessions with a single command a modified version of "rtap", sending commands to RSVP API (application program interface) was used. All the sessions are supposed to be long-term ones. We measured data over a 5 minutes large temporal window the Refresh parameter is 30 seconds long as RFC2205 suggests. In our topology one computer was considered to be the upstream router for all the sessions and the other one the downstream router. These two routers and the network tool were linked to a hub. As the IP packets are encapsulated into ARPA Ethernet frames the MTU is 1500 bytes.

6.3 Performance Evaluation

We considered three parameters in our simulation: CPU load, throughput and signaling overhead. The CPU load time is where RSVP daemon is sending message with router. It is measured by periodically. The throughput represents the average number of bytes transmitted by the RSVP daemon. A high Throughput is responsible to reduce bandwidth waste. The number of messages processed over a given interval of time is called *system throughput.* Signaling represents the average number of packets transmitted by the RSVP daemon. For simulation we tested the performance of RFC2961 with security authentication. As per security authentication we implement ESP (Encapsulating Security payload) for that we applied authentication algorithm, DES (Data Encryption Standard) in Cipher-Block Chaning (CBC) and we tested the Bundle Message, Message ID and Summary Refresh extensions. Each test was run with a fixed number of sessions; for each test, we examine how CPU load time, Throughput and Signaling gets affected due to Bundle Message, MessageID and SRefresh. Table 2 shows the results we evaluated with security authentication of RFC2961. It is shown that the CPU Load time is little bit increased while sending the Bundle Message, MessageId extension and Refresh Extension. That's a obvious thing if we are applying security authentication in that time this mechanism is filtering each and every process and at the destination end we will get reliable and secure message. Table 3 shows the results we evaluated with security authentication of RFC2961. The throughput values bytes per second has been decreased while sending the Bundle Message, Message ID and SRefresh Extension. Table 4 shows the results of Signaling values (packet per second). As per our implementation we evaluated that the signaling increased by applying security authentication.

Table 2

CPU load is shown for different RFC2961 extensions at different number of sessions. Each value obtained from an extension such as Bundle Message, MessageID and SRefresh.

CPU Load (Value In Persentage)			
Session	**BundleMessage value (after applying Secutity premisis)**	**Message ID value (after applying Secutity premisis)**	**SRefresh Value (after applying Secutity premisis)**
100	3.2	2.8	2.84
200	3.38	3.7	4.32
500	13.42	9.1	5.6
1000	26.32	23.1	13.3
2000	41.32	38.9	18

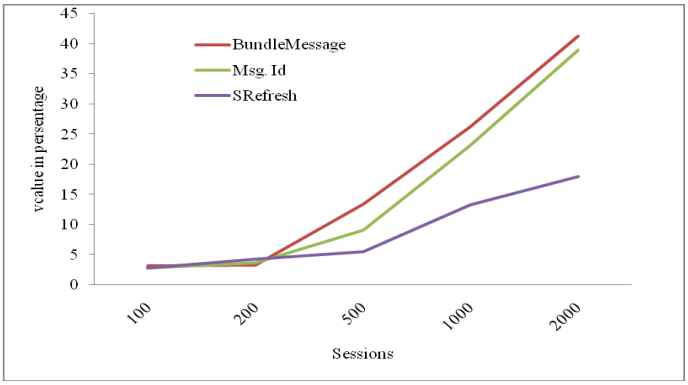


Fig. 10. Comparison of the track of the CPU load for Extensions RFC2961.

Table 3

Throughput is shown for different RFC2961 extensions at different number of sessions. Each value obtained from an extension such as Bundle Message, MessageID and SRefresh.

Throughput(values in byte/sec)			
Session	**Bundle value (after applying Secutity premisis)**	**Message ID value (after applying Secutity premisis)**	**SRefresh Value (after applying Secutity premisis)**
100	740	279	26
200	1588	589	49.0
500	3986	1979	128
1000	8580	4002	210
2000	18001	7988	418

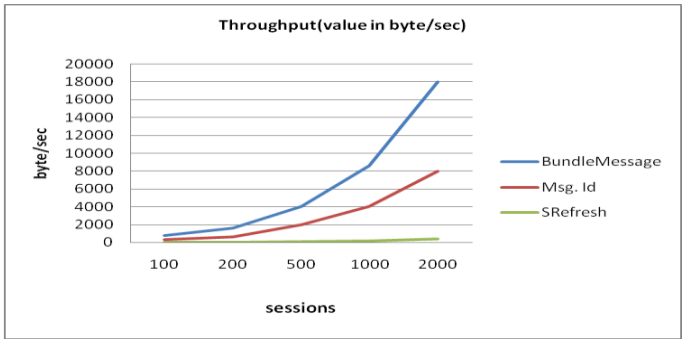


Fig.11. Comparison of the track of the Throughput utilization For Extension RFC2961

Table 4

RFC2961 signaling on the wire largely depends on the security configuration in terms of refresh interval and refresh reduction. Signaling value in packets/sec) has been shown different number of sessions. We obtained certain values.

Session	**Bundle value (after applying Secutity premisis)**	**Message ID value (after applying Secutity premisis)**	**SRefresh Value (after applying Secutity premisis)**
100	1.8	8.2	0.1
200	2.5	16.3	0.3
500	4.2	31.1	0.2
1000	7.0	61.8	0.2
2000	11.9	120	0.4

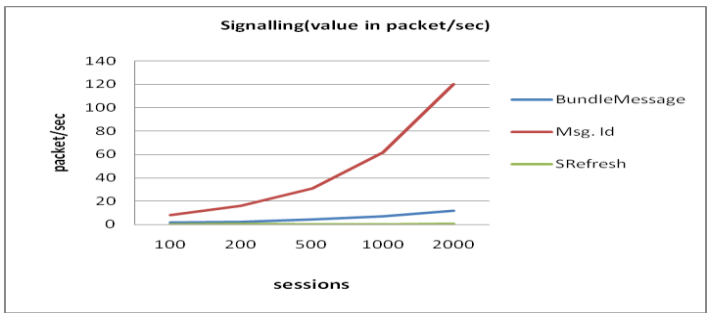


Fig.12. Comparison of the track of the signaling utilization for Extension RFC2961

6. CONCLUSIONS

As per our implementation on RFC2961 with security prevention will be very much helpful for improvement of QoS and reduces traffic on to the network. We have briefly described the RFC2961 with ESP (Encapsulating Security Payload), unicast communication. After then we measured the performance of these extensions with three parameters CUP load, Throughput, and Message Signalling. We demonstrate

that the use of RFC2961 with ESP reduce the bandwidth waste, provide message integrity & node authentication, secure data stream, providing compatibility and also improve the reliability of control message delivery. While measurement are valid for one specific implementation and one combination of hardware and software only. Our conclusion are not based on the absolute values of CPU load, Throughput and Signalling message but only point to provide security prevention for bundle message, message id, signaling message and node authentication. At future work it encourages the deployment of RFC2961 with security in large network also.

7. REFERENCES

[1] Berger L., "Generalized MPLS Signaling - RSVP-TE Extensions", Work in Progress, draft-ietf-mp generalized-rsvp-te-09.txt, September 2002.

[2] Awduche D., Berger L., Gan D., Li T., Srinivasan V., Swallow G., "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC3209, December 2001

[3] Awduche D., Chiu A., Elwalid A., Widjaja A., Xiao X., "A Framework for Internet Traffic Engineering", draft-ietf-tewg-pprrinciples-02.txt, May 2002.

[4] Awduche D., Chiu A., Elwalid A., Widjaja I., Xiao X., "Overview and Principles of Internet

[5] Traffic Engineering", RFC3272, May 2002. Awduche D., Hannan A., Xiao X., "Applicability Statement for Extensions to RSVP for LSP-Tunnels", RFC3210, December 2001.

[6] Berger L., Gan D., Swallow G., Pang P., Tommasi F., Molendini S. , "RSVP Refresh Overhead Reduction Extensions", RFC2961, April 2001.

[7] Braden R., Zhang L., Berson S., Herzog S., Jamin S., "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC2205, September 1997.

[8] Karsten M., Schmitt J., Steinmetz R., "Implementation and Evaluation of the KOM RSVP Engine", Proceedings of the

[9] 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'2001).

[10] KOM RSVP engine, http://www.kom.e-technik.tu-darmstadt.de/rsvp/ RSVP ISI distribution, http://www.isi.edu/rsvp/release.html

[11] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. RFC 2205. Sept 1997.

[12] J. Schmitt, M. Karsten, and R. Steinmetz. On the aggregation of deterministic service flows.Computer Communications, 24(1):2–18, 2001.

[13] B. E. Carpenter and S. Brim, "Middleboxes: Taxonomy and Issues,"
Engineering Task Force, RFC 3234, Feb. 2002. [Online] Available: http://www.rfc-editor.org/rfc/rfc3234.txt

[14] R. Braden, D. D. Clark, and S. Shenker, "Integrated services in the Internet architecture: an overview," Internet Engineering Task Force, RFC 1633, June 1994. [Online]. Available: http://www.rfceditor.org/rfc/rfc1633.txt

[15] L. Zhang, S. Deering, D. Estrin, S. Shen, and D. Zappala, "RSVP: A New Resource ReSerVation Protocol," *IEEE Network*, vol. 7, no. 5, pp.8–18, Sept. 1993.

[16] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification," *RFC 2205*, Sept. 1997. [Online]. Available: http://www.rfceditor.org/rfc/rfc2205.txt

[17] T. Chiueh, A. Neogi, and P. Stirpe, "Performance Analysis of an RSVPCapable Router," in *Proc of IEEE RTAS*, 1998, pp. 39–48.

[18] L. Berger, D. Gan, G. Swallow, P. Pan, F. Tommasi, and S. Molendini, "RSVP refresh overhead reduction extensions," RFC 2961, Apr. 2001. [Online]. Available: http://www.rfc editor.org/rfc/rfc2961.txt

[19] RFC 4860 - Generic Aggregate Resource ReSerVation Protocol (RSVP) Reservations.mht by IETF.

[20] RFC 2207 - RSVP Extensions for IPSEC Data Flows (RFC2207).mht by IETF.

[21] IP security: A Brief Survey by Zhijun Ni, *zhijunni@math.ohio-state.edu*