Proc. of the Intl. Conf. on Future Trends in Computing and Communication -- FTCC 2013

ISBN: 978-981-07-7021-1

A Detailed Study of Transport Layer SCT Protocol and its Security Solutions

Raghavendra Ganiga, Sanoop Mallissery
Department of Information & Communication Technology
Manipal Institute of Technology, Manipal University, Karnataka, India

Abstract- Among many reliable transport protocols, the Stream Control Transmission (SCT) Protocol is very much suitable in networking scenario. To transport telephone signaling messages over IP networks the SCT Protocol act as key role. Multimedia data such as speech, images and video is basic input to SCT Protocol and can be compared with the traditional protocols. The key factor of SCT Protocol will be the capacity to secure the transported data on the network. Security services like authentication, authorization and confidentiality are important and must be provided for SCT Protocol traffic. This paper is highlighting on working of SCT Protocol on transport layer and the security solutions.
Keywords—SCTP, RFC, S-SCTP, TLS, IPSec, SSH etc.

I. INTRODUCTION

SCT Protocol is a transport protocol with the standard security applications and implementations is having a great importance in today's networking scenario. To overcome functional and performance related problems used standard security protocol IPsec, TLS/SSL and SSH. The optimal solution to overcome the standard security issues used is called Secure-SCT Protocol (S-SCT Protocol) is introduced. To make the security solution perfect security should integrate directly with SCT Protocol.

The security protocols of SCT Protocol are level below the application layer and presentation. SSL and TLS security protocols are considered to be under the identical name of SSL or TLS. After receiving the standardization from SSL the IETF renamed the protocol as TLS. SSL offers communication data path between a SCT Protocol endpoints regardless of platform.

For IP networks SCT Protocol is a transport protocol which is uniformly standardized in RFC 2960 [1] by the Internet Engineering Task Force (IETF). SCT Protocol deceits between the application and network layer of the TCP/IP Reference model [2]. It is a consistent and message-oriented transport layer protocol also uses IP as the network protocol to send and receive packets from the peer instance. SCT Protocol combines the best features of UDP and TCP and at the same time detects duplicate data , lost data and out of order data which is required for the future IP network and it is more powerful protocol compare to TCP and UDP in the case of streaming data [3], [4], [5]. The SCT Protocol offers multiple stream services in each connection and if one of the streams is blocked, other stream can still deliver their data which is similar to multiple lanes on highway. Second service offered by the SCT Protocol is multi homing in which sending and receiving host can have multiple IP addresses in each end for an association using this approach when one path fails another interface can be used for the data delivery without interruption. These fault tolerant techniques are very much essential when using real time payload such as Internet Telephony. Also SCTP offers full duplex communication services where at the same time data can flow in same direction. [6], [7] [8], [9].

II. PACKET FORMAT

In SCT Protocol peer parties in SCT Protocol are called endpoints. The communication connections between these SCT Protocol endpoints are called association. The combination of an IP address and a port are called as SCTP transport address [3]. One SCTP endpoint with single interface with more than one IP address is called as multihoming. SCT Protocol endpoints with multihoming association end points can have several paths to connect dissimilar end points. The chunk carried both control information and user data from one endpoint to the other. A chunk is a structured data with blocks which is implanted in an SCT Protocol packet.

The basic format of a SCTP packet is shown in the Figure 1 it comprises an SCT Protocol general header. Combination of more chunks grouped behind the header which carries the control packet and user data. The general header in the SCT protocol contains the source and destination port number each of 2 bytes. The next 4 bytes contain the verification tag and checksum. Verification tag is a number that matches a packet to an association. This prevents a packet from a previous association from being mistaken as a packet in this association. In the SCT Protocol header checksum [6] is calculated over general header and group of chunks which prevents data updating problem in the destination endpoint.

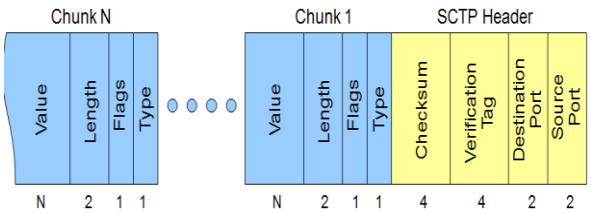


Figure 1: Basic format of SCT Protocol packet

Proc. of the Intl. Conf. on Future Trends in Computing and Communication -- FTCC 2013

ISBN: 978-981-07-7021-1

Chunks are shown in the Figure 1 which is designed in such a way they are self-descriptive and have an unvarying format. The chunks in the SCT Protocol packet format having two major types which are data chunks and control chunks. SCT Protocol endpoints can send control information with each other and also used to manage and control the association. DATA chunks are carry the payload between to SCT Protocol endpoints. One SCT Protocol packet contains both control chunks and data chunks.

The chunk length can vary with any size, but must complete a 32-bit word. The last word must probably be filled up with padding bits to make complete 32 bit words. The first byte sets the chunk type in which value 0 defines a DATA chunk. There were initially 14 other chunk types defined which are all control chunk types. The next byte contains the chunk flags of one byte used to identify the different chunk types. Byte number of chunk 3 and 4 contain the length of the complete chunk except the padding bytes at the end.

III. MAIN FEATURES

The main features of SCT Protocol are detailed below some of features not supported by TCP or UDP protocols are as follows.

Multi-streaming in SCT Protocol: An SCT Protocol with single association can consist of numerous message streams. Generally messages before passing to the Upper layer, the messages within a stream have to check for all missing order, arrived with a different order. For this checking SCT endpoint has to wait until it can reorder the message.

Multi-homing: SCT Protocol endpoint contains many network interfaces and each interface supports one IP address and several interfaces support many IP addresses. SCTP endpoint s with one IP address interface and SCTP endpoint B with one IP address with its own interfaces with A and B numbers of IP address interfaces with A*B different path of association it will support [10].

Bundling: Chunks can be put together in one SCT Protocol packet. This reduces the overhead since the SCT Protocol common header is just sent once. Control and data chunks in SCT Protocol packet can combine together. Grouping of control and data chunks is optional. MTU size has to take into account; because of association state change some combinations are not allowed [1].

Retransmission of lost packets in SCTP: SCT Protocol guarantees that transmitted data from source end to destination end and checks that really received by the destination. The Selective Acknowledge Chunk (SACK chunk) informs the sender about the sequence number of the last DATA chunk and informs also previous sequence numbers are perfect and no missing chunk involved in that is collective ACKNOWLEDGMENT point. If there is a missing chunk it will request again to send Data chunk which it is lost during transmission and also collective ACKNOWLEDGMENT point are informed. For the first and last chunk transmission sequence number should be well defined standard transmission. The effective ACKNOWLEDGMENT and the Selective ACK chunk evidence enable a very effective ACKNOWLEDGMENT and retransmission algorithm.

CC (Congestion control) and FC (Flow Control) in SCTP: CC and FC of SCT Protocol is having similar functionality to the transport layer protocol TCP [11], [12]. For flow control both sender window and receiver window should match by receiving the receiver window update message. When the receiver window value become zero , for sender stop sending the data because if still sender is sending the data chances of loss packet in the receiver side because unable to process the data which is send by the sender. For congestion control the main parameter used by the in SCT Protocols are congestion window and slow-start threshold because each endpoint can have many IP address and multiple path for data in the network but each path requires its own congestion window to support.

SCT Protocol association management: Transport layer in the TCP/IP layer with two protocols gives connection oriented services and which uses setup of association link, maintained and shut down. For the above reason should define exact for each procedures.

Association setup: Similar to TCP also SCT Protocol uses a handshake procedure for setting up the association, but unlike TCP it performs a four-way handshake instead of a three-way handshake in figure 2.

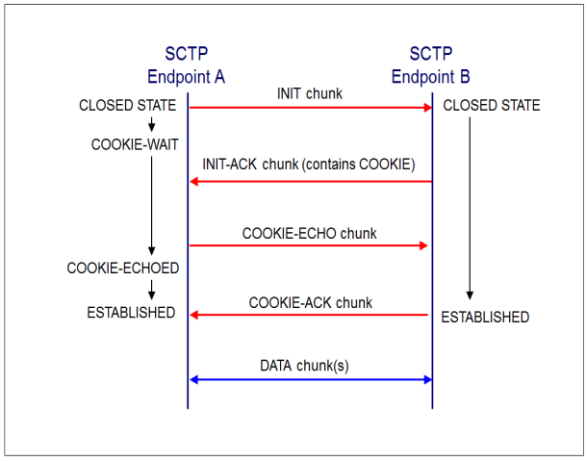


Figure 2: Association setup

Proc. of the Intl. Conf. on Future Trends in Computing and Communication -- FTCC 2013

ISBN: 978-981-07-7021-1

First endpoint A sends an INIT chunk, it includes an initiation tag which is used later for the common header. This simple feature prevents the endpoint peers from blind attacks. A SCTP packet which is having a verification tag not belonging to an established association is discarded immediately. Some other parameters, like the initial Transmission Sequence Number (TSN), the Advertised Receiver Window Credit or the list of the endpoints IP addresses are also contained in the INIT chunk. SCTP Endpoint B receives the INIT chunk and responses with an INIT-ACK chunk, which contains the same fields as the INIT chunk.

Association shutdown: There are two ways to SHUTDOWN the association one is graceful shutdown other one is abortive shutdown. The graceful shutdown usually initiated by the Upper Layer Protocol. SCT Protocol performs a three-way handshake, SHUTDOWN chunk, SHUTDOWN-ACK chunk and SHUTDOWN-COMPLETE chunk are send like it is shown in Figure 3.

The abortive shutdown: An unreliable best-afford shutdown to let the SCTP peer know that the association is stopping. The SCTP endpoint who wants to stop the association directs an ABORT chunk to the SCT Protocol endpoints and enters the 'closed' state.

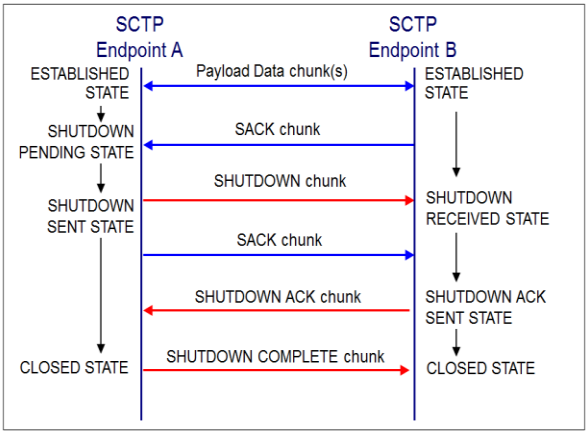
Path and peer monitoring: SCT Protocol endpoints monitor all paths to the peer SCT Protocol endpoint of an association. HEARTBEAT chunks are sent frequently over all paths except the primary path. Each HEARTBEAT chunk should be acknowledged by a HEARTBEAT-ACK chunk. [13] During association path where no HEARTBEAT-ACK takes place in a certain time gets the state becomes inactive.

SCT Protocol extensions: After the initial definition of SCTP in RFC 2960, SCTP has been continuously developed and some extensions have been proposed in internet drafts. These extensions may be important for the acceptance of SCT Protocol in future IP based network scenarios.

Partial reliable delivery: A feature that allows ignoring loss, unordered or late DATA chunks in case it is wanted. Computing late DATA chunks useful for application which works in real time where audio or video data is not usable if it is late. This feature is realized with the help of a new optional parameter type, which can be used in the INIT-ACK and INIT chunks to make sure both endpoints support this feature. In this case a new control chunk called FORWARD-TSN (Transmission Sequence Number) can be sent to perform the partially reliable service [7].

Dynamic address reconfiguration: An extension that allows the reconfiguration of IP addresses of an already established association. Two new control chunks are address configuration [8] change and address configuration acknowledge allow adding and dropping of IP addresses in the IP based network.

Figure 3: Graceful shutdown



IV. SECURITY SOLUTIONS FOR SCTP

i. SCT Protocol over IPsec

For the SCT Protocol is an internet protocol the security architecture for the SCTP [14] relies on a suite of protocols providing a system for securing communication channel. AS such, IPsec defines:

Security protocols: For Authentication and encapsulation SCT Protocol uses two security protocols Encapsulating Security Protocol [16] and Authentication header [15]. The AH provides protection of the data integrity which sent from a source to destination address and also checks for data origin authentication. In addition to these services, Encapsulating Security Payload provides confidentiality by encrypting the IP payload transport mode or in tunnel mode.

Security Associations (SAs) in SCT Protocol: A SA designates a single direction association characterized by the two Security parameter index, DA and SP. IN security parameter index (SPI) for each destination address different SA are available with its index and which uses security protocol AH or ESP. The list of all Security Associations is kept in a database called it as security association database (SAD). The inbound and outbound traffic which maps to an association and kept it in another database called security policy database (SPD).

Key and Security Association management procedures: IPsec implementation has two key distribution concepts those are manual and automatic key distribution. Automatic key distribution is based on a key management protocol [19], and a corresponding protocol for creating authenticated keying material based on the internet key exchange [18]. These protocols are used to establish initial

Proc. of the Intl. Conf. on Future Trends in Computing and Communication -- FTCC 2013

ISBN: 978-981-07-7021-1

security associations, i.e. secure, authenticated channels for further communication, also named phase-1 Security associations. Over these channels, further Security Associations for the IPsec security protocols may be established as phase-2 Security Associations, which involves distribution of appropriate keying material between the SCT Protocol endpoints. Typically, the setup of phase-2 Security Associations is a shorter process compared to the initial phase- 1 Security Association exchange.

ii. Transport Layer Security over SCT Protocol

The Transport Layer Security protocol [19] was specified for use on top of a transport protocol that gives data delivery with stringent reordering of chunks, Example is TCP. It introduces its own record marking in the Transport Layer Security record layer. In [20] it is defined how a standard TLS protocol implementation should use the services of SCT Protocol, and which limitations implementers and users would have to consider. As the Transport layer security standard does not readily support a multiplexing concept as required for the SCT Protocol streams, a separate Transport Layer Security connection has to be established for each stream that has to be protected resulting in multiple Transport Layer Security connections per SCT Protocol association. Additionally, Transport Layer Security connections always require a bidirectional communication. Therefore, two unidirectional SCT Protocol streams have to be combined logically to a bidirectional stream in order to support Transport Layer Security.

A benefit of the separate Transport Layer Security connections per stream is that in one association secure and insecure streams can be used within one association as essential by the application. By setting up a new transport layer security connection can be set up for the insecure communication to give secure communication domain. Two SCT Protocol endpoints that establish an SCT Protocol association with n bidirectional streams can have a maximum of n Transport Layer Security connections. Different way of establishing connection TLS over SCTP is implemented as follows:

- Full handshake for each bidirectional stream. The handshake is performed before the first data is to be transmitted securely. After that, each stream has an independent Transport Layer Security connection.
- Full handshake only for the first bidirectional stream. This results in a valid Transport Layer Security session identifier. The session identifier can be used for Transport Layer Security session resumption, which allows for establishing successive Transport Layer Security connections on the other streams with an abbreviated handshake. After this abbreviated handshake, each newly established Transport Layer Security connection has its proper set of security parameters.

Figure 4 shows a scenario where 3 applications use the same SCT Protocol association. In SCT Protocol endpoint each application uses one stream. Application number 2 uses plain SCT Protocol. Application 1 and Application 3 use secured transmission by setting up a Transport Layer Security session in their streams.

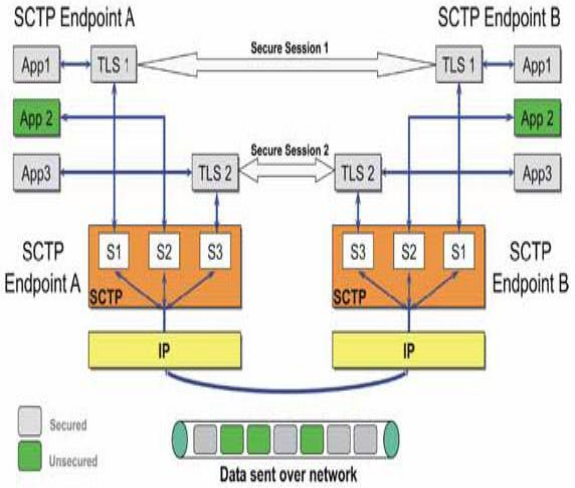


Figure 4: Transport Layer Security over SCT Protocol

iii. Secure-SCT (S-SCT) Protocol

S-SCT Protocol was designed to meet the following criteria:

Security: S-SCT Protocol offers both for user data and control chunk with secure data integrity and origin authentication which is transported by SCT Protocol, but also for the peer-to-peer control information used by SCT Protocol itself. Thus, vulnerabilities that may occur when using the Dynamic Address Re- configuration extension without security mechanism can be avoided. User chunk control also uses data confidentiality provided by mechanisms for flexible encryption of SCT Protocol. The conclusion whether encryption is used or not can be taken by the application on a per message basis. **Performance and scalability**: To add a security features into a SCT Protocol, overhead caused by the transmission overhead and computational effort can enhanced with application requirement. Both in the case of many streams that need to be secured, and in the case of many possible address combinations between endpoints, S-SCT Protocol shall provide a scalable solution which is achieved by establishing exactly one secure session per SCT Protocol

Proc. of the Intl. Conf. on Future Trends in Computing and Communication -- FTCC 2013

ISBN: 978-981-07-7021-1

association. Further, to minimize transmission overhead, an HMAC [21] is computed over the whole SCT Protocol packet, including all chunks and the common header, rather than over individual messages.

Ease of use: Application with limited configuration to take advantage of the full functionality of S-SCT Protocol defines several security stages:

- Security stages 0: At this stage S-SCT Protocol does not use any of the security functions and is fully compatible with standard SCTP.
- Security stage 1: At this stage all SCT Protocol chunks and the header of all SCT Protocol packets of the association are authenticated and integrity checked.
- Security stage 2: With the help of HMAC operation, Application gives the flag indication using this parameter user data chunks are encrypted.
- Security stage 3: In this stage all chunks within any SCT Protocol packet are encrypted, and the complete message is authenticated, and integrity checked.

Within an S-SCT Protocol session, two SCT Protocol endpoints may use different levels of security in which for different instance, if one SCT endpoint may require authentication and SCT endpoint require privacy level this can be configured using this protocol. The first endpoint select security stage 1 and other means second would select security stages 3. Any time during an SCT Protocol association lifetime, the user can change the security level as shown in the Figure 5.

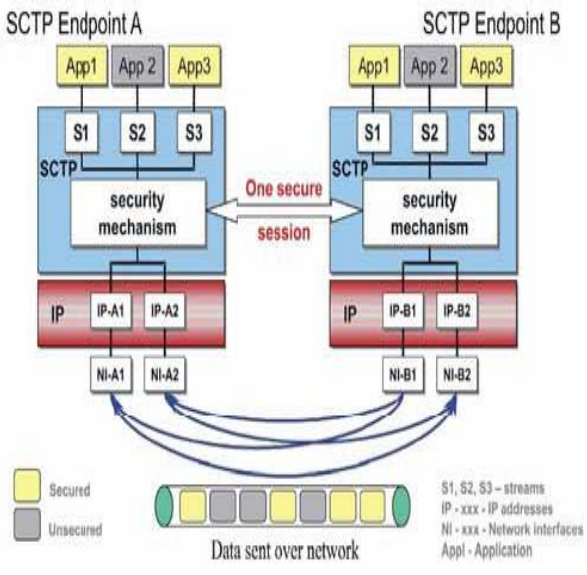


Figure 5: S-SCTP over SCTP

iv. SSH over SCTP

Secure Shell (SSH) is a protocol for secure data exchange between two hosts. SSH a routine to use secure connection which is multiplexed from the SSH TL and multiple channels uses its services [22]. Protocol uses single ordered transport channel for its secure shell transport layer.

The SCT Protocol is a transport protocol with good reliable and substitute to TCP, can also be used with SSH. This allows to profits from SCT Protocol multi homing features, for increased consistency or bandwidth, without further effort and without causing any security issues.

Another feature of SCT Protocol is multiple streams, which is unidirectional channels within single connection. To check the order messages are kept within a single stream, not across multiple streams, so lost messages only delay the streams they belong to. Therefore, mapping Secure Shell channels onto SCT Protocol streams is a possible optimization. Since reordering across streams, the cipher used for Secure Shell has to be altered to work without using information from the previous message. Mapping the channels also discloses them to potential attackers because the stream information is detectable in the SCT Protocol header. A probable solution is to encrypt the SCT Protocol data messages and their headers.

V. SECURITY SOLUTIONS COMPARISON

SCT Protocol uses bidirectional communication, sending multiple streams in one message, multiple IP address in one endpoint with secure total transmission can accepted by using TLS over SCTP or by the SCTP over IPsec respectively. Without any encryption both IPsec and TLS gives data integrity and authentication. For highly asymmetrical scenarios with many clients which have to be authenticated frequently by one or few servers, complex and computationally costly authentication mechanisms may contribute to possible scalability problems.

If confidentiality is mandatory when using IPsec, the Encryption Security protocol with a non-null encryption algorithm can be used to protect the user data. Obviously, TLS also provides encryption with different algorithms by using one of the existing cipher suites.

With respect to transmission and delivery, the TLS over SCTP concept requires in sequence and reliable service. The TLS was designed with these TCP properties in mind. Unordered delivery or the Partially Reliable Transport extension of SCTP cannot be supported on streams protected by TLS, unless suitable TLS extensions are standardized and provided in implementations. However, these features as well as unidirectional communication can be used on non-secure streams within a protected association. One

Proc. of the Intl. Conf. on Future Trends in Computing and Communication -- FTCC 2013

ISBN: 978-981-07-7021-1

consequence of using the TLS over SCTP concept is that although TLS protects the SCTP user data, neither the control chunks exchanged within the corresponding SCTP association nor any IP layer information can be protected by TLS. This is a general problem with security solutions operating above the transport layer which possibly allows some forms of denial of service attacks against the SCT Protocol association itself.

VI. CONCLUSION AND FUTURE WORK

The standard security solutions Transport Layer Security and IPsec are developed for Transmission Control Protocol and hence have functional limitations and cannot support all SCT Protocol features. S-SCT Protocol can be assumed to be an optimal solution because it integrates the security functionalities directly into SCT Protocol. We also compared the security solutions and show the strengths and weaknesses of each solution. The functional limitations of TLS and IPsec cannot be changed fully, because modification of the protocol specifications that already existing. But TLS and IPsec solutions do from functional limitations and from performance related ones.

The performance drawbacks of the security solutions are caused by SCT Protocol features which are not supported in an effective way. The security solutions can be used in many areas, e.g. in local area networks or the internet, so the solutions have also to be evaluated for different environments. SCT Protocol uses a cookie mechanism as a protection against sightless DoS attacks. But until now, this effectiveness of this cookie mechanism has not been validated in measurements. Another motivating point which could be investigated is the possibility of new DoS because of SCT Protocol specific characteristics, i.e., DoS using the cookie mechanism by sending fake cookies to a server, or replaying a valid cookie.

REFERENCES

[1] Stewart, R.; Xie, Q.; Morneault, K.: RFC 2960 – "Stream Control Transmission Protocol", IETF, Network Working Group, October 2000.

[2] ISO 7498:1984 Open Systems Interconnection - Basic Reference Model.

[3] Stewart, R.; Xie, Q.: "Stream Control Transmission Protocol – A Reference Guide", Addison-Wesley, November 2001.

[4] Coene, L.: RFC 3257 - "Stream Control Transmission Protocol Applicability Statement", IETF, Network Working Group, April 2002.

[5] Ong, L.; Yoakum, J.: RFC 3286 – "An Introduction to the Stream Control Transmission Protocol (SCTP)", IETF, Network Working Group, May 2002.

[6] Stone, J.; Stewart, R.; Otis, D.: RFC 3309 – "Stream Control Transmission Protocol (SCTP) Checksum Change", IETF, Network Working Group September 2002.

[7] Stewart, R.; Ramalho, M.; Xie, Q.; Tüxen, M.; Conrad, P.: RFC 3758 – "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", IETF, Network Working Group, May 2004.

[8] Stewart, R.; Ramalho, M.; Xie, Q.; Tüxen, M.; Conrad, P: Internet-Draft – "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration", draft-ietf-tsvwg-addip-sctp-14 (work in progress), IETF, Network Working Group, March 2006.

[9] Tüxen, M.; Stewart, R.; Lei, P.; Rescorla, E.: Internet-Draft – "Authenticated Chunks for Stream Control Transmission Protocol (SCTP)", draft-ietf-tsvwg-sctp-auth-02 (work in progress), IETF, Network Working Group, March 2006.

[10] Unurkhaan, E.: "Secure End-to-End Transport – A new security extension for SCTP", Dissertation, March 2005

[11]Balliache,L.:PracticalQOS,http://www.opalsoft.net/qos/TCP-1021.htm.

[12] Floyd, S.: RFC 2914 – "Congestion Control Principles", IETF, Network Working Group, September 2000.

[13] Jungmaier, A.: "SCTP for Beginners", http://tdrwww.exp-math.uniessen.de/inhalt/forschung/sctp_fb/sctp_multihoming.html, 2003.

[14] S. Kent and R. Atkinson, Security architecture for the Internet protocol, RFC 2401, 1998.

[15] S. Kent and R. Atkinson, IP authentication header, RFC 2402, 1998.

[16] S. Kent and R. Atkinson, IP encapsulating security payload (ESP), RFC 2406, 1998.

[17] D. Maughan et al., Internet security association and key management protocol (ISAKMP), RFC 2408, 1998.

[18] D. Harkins and D. Carrel, The Internet key exchange (IKE), RFC 2409, 1998.

[19] T. Dierks and C. Allen, The TLS protocol version 1.0, RFC 2246, 1999.

[20] A. Jungmaier, E. Rescorla and M. Tüxen, Transport layer security over stream control transmission protocol, RFC 3436, 2002.

[21] H. Krawczyk, M. Bellare and R. Canetti, HMAC: Keyed-hashing for message authentication, RFC 2104, 1997.

[22] Robbib Segglemann, Michael Tuxen, Erwin P. Rathgeb, SSH over SCTP-Optimizing a multichannel protocol by adapting it to SCTP, 8th IEEE international Symposium on Communication system and networking , 2012.