International Journal of Applied Research 2016; 2(2): 470-473

International Journal of Applied Research

ISSN Print: 2394-7500
ISSN Online: 2394-5869
Impact Factor: 5.2
IJAR 2016; 2(2): 470-473
www.allresearchjournal.com
Received: 18-12-2015
Accepted: 20-01-2016

Elhabib Elhabib A
Department of Communications, College of Engineering, Al-Neelain University

Amin B Abdel Nabi
Department of Communications, Al-Neelain University

Abdelrasouljabar Alzubaidi
Sudan University of science and technology, Department of electronics Engineering, College of Engineering

Correspondence
Elhabib Elhabib A
Department of Communications, College of Engineering, Al-Neelain

Performance of SCTP Congestion Control over Network

Elhabib Elhabib A, Amin B Abdel Nabi, Abdelrasouljabar Alzubaidi

Abstract
Congestion control scheme is key algorithms used to regulate the load in modern packet networks. The Stream Control Transmission Protocol (SCTP) inherited these algorithms from the Transmission Control Protocol (TCP). This paper is related to SCTP congestion control mechanisms, specifically in the Congestion Avoidance algorithm. In the congestion avoidance algorithm, the size of the congestion window increases additively until congestion is detected. However, this increase may lead to enter fast retransmit congestion control algorithm and fast recovery algorithm. For this reason we propose a new enhancement to the Congestion Avoidance algorithm by adjusting the congestion window size based on available network capacity, that can reduce congestion window rather than slow start threshold, then the cwnd is fully utilized and immediately triggers slow-start phase work again, also the SCTP sender has no wait- for-long time to receive the acknowledging and all available resources can utilize data sender rate Efficiently, by using tools of OMNet++ simulation and finally will compare the performance of the SCTP congestion control, between the proposed method and original method.

Keywords: congestion avoidance algorithm, SCTP congestion control algorithm, OMNeT++

1. Introduction

The transport layer is an essential part of the International Organization for Standardization/Open Systems Interconnection reference model (ISO/OSI), as well as the Internet Protocol stack (TCP/IP) [1]. Different transport layer protocols were already introduced [2]. The responsibilities of the transport protocols include end-to-end message transfer capabilities independent of the underlying network, along with error control, segmentation; flow control and congestion control [3]. The properties and characteristics of the different transport protocols must be studied for delay sensitive multimedia applications. Simple and fast protocols are recommended, while for reliable data transfer the information must be delivered, ordered and without any error. The multiple features of SCTP make SCTP a good alternative of next generation transport layer protocol [4]. The main differences between TCP and SCTP are multihoming and multistreaming features. SCTP multihomed hosts can be bound multiple IP addresses to use multiple network interfaces effectively and make data transmission more robust. SCTP multistreaming feature, which delivers unidirectional data independently, can avoid Head-of-Line blocking and benefit data delivering in time [5]. SCTP also provides partial reliability service that can deliver various data of applications in one association [6]. SCTP has more excellent features than TCP but it inherits TCP's congestion control scheme. In the SCTP connection the sender uses the slow start algorithm, According to this equation: Congestion window (cwnd) = cwnd + Acknowledged (acked) chunk size from receiver. The congestion window limits the number of bytes the sender is allowed to transmit before waiting for a new acknowledgement, that means, that not more than cwnd bytes may be outstanding if the value of cwnd is less than the Slow start threshold (ssthresh) value, which is set to an arbitrary value (mostly the advertised receiver window of the peer during association setup) at the beginning of an association. If the value of cwnd is greater than the ssthresh value, the network is forced to drop one or more packets due to overload, in this time the sender uses the congestion avoidance algorithm According to this equation: cwnd = cwnd + Maximum Transmission Unit (MTU) in this mechanism the current congestion window actually does not decrease but increase by MTU. However, this increase may lead to entering fast retransmit algorithm and fast recovery algorithm. For this reason we propose a new enhancement to the Congestion

Avoidance algorithm. As given by equation: cwnd= cwnd-[0.144*cwnd], instead of cwnd = cwnd + MTU. However the performance of the proposed modification was evaluated by limited set of simulations called OMNeT++ is an open source discrete event simulation environment with a modular component, based architecture written in C++ [7]. The types of component are channels (described by the parameters delay, packet error rate and data rate), network definitions, simple and compound modules [8]. The components can be assembled into more complex modules via connected gates, network are the result of combined module types that communicate through messages.

The paper is organized as follows: section II Overview of SCTP Congestion Control, while in section III, Modified Congestion Avoidance Algorithm of SCTP in section IV. Behavior of congestion control through Modification Congestion Avoidance, in section V. Network Simulation Model the objectives in section VI, the Expected results of the proposed: Modified Congestion Avoidance Algorithm in section VII, and finally, the Conclusions in section VIII.

II. Overview of SCTP Congestion Control

The SCTP uses congestion control algorithms to effectively handle network failures or unexpected traffic surges, and ensure quick recovery from data congestion. SCTP and TCP support the same set of congestion control algorithms. However, when using SCTP, the congestion control algorithms are modified to suite the protocol-specific requirements [9]. The SCTP supports the following congestion control algorithms:

- Slow Start and Congestion Avoidance

The slow start and congestion avoidance algorithms are used to control the amount of outstanding data being injected into the network. The SCTP uses the slow start algorithm at the beginning of the transmission, when the network condition is unknown, and also in repairing loss detected by the retransmission timer. The SCTP slowly probes the network to determine the available capacity of the network to avoid congestion in the network. If it detects congestion in the network, it switches to the congestion avoidance algorithm to manage the congestion [10]. The slow start and congestion avoidance algorithms use the following congestion control variables

- ✓ cwnd: Specifies the limit on the amount of data the sender can transmit through the network, before receiving an acknowledgement. This variable is maintained for each destination address.
- ✓ Receiver window (rwnd): Specifies the receiver's limit on the amount of outstanding data.
- ✓ ssthresh: Determines whether the slow start or congestion avoidance algorithm must be used to control data transmission [11].

In a SCTP connection, the sender uses the slow start algorithm if the value of cwnd is less than the ssthresh value. If the value of cwnd is greater than the ssthresh value, the sender uses the congestion avoidance algorithm. If the values for cwnd and ssthresh are same, the sender can use either the slow start or congestion avoidance algorithm, in the TCP the congestion avoidance procedure allows the window size to increase linearly until a time-out occurs or the maximum window size is reached [12] illustrated in figure (1), the time-out occurs when the window size is 20. At this moment, the multiplicative decrease procedure takes over and reduces the threshold to one-half of the previous window size. The previous window size was 20 when the time-out happened so the new threshold is now 10. TCP moves to slow start again and starts with a window size of 1, and TCP moves to additive increase when the new threshold is reached. When the window size is 12, a three-ACK event happens. The multiplicative decrease procedure takes over again. The threshold is set to 6 and TCP goes to the additive increase phase this time, it remains in this phase until another time-out or another three ACKs happen, also The SCTP have the same behavior of Additive Increase Multiplicative Decrease (AIMD) it is produce poor perform over broadband high latency networks in case of multiple packet losses [13].

- Fast Retransmit and Fast Recovery

The fast retransmit congestion control algorithm is used to intelligently retransmit missing segments of information in a SCTP association. When a receiver receives a data chunk out of sequence, the receiver sends a SACK packet with the unordered Transmission Sequence Number (TSN), to the sender [14]. The fast retransmit algorithm uses four SACK packets to indicate loss of data, and retransmits data without waiting for the retransmission timer to timeout. After the fast retransmit algorithm sends the data that appears to be missing, the fast recovery algorithm controls the transmission of new data until all the lost segments are retransmitted [15]. The whole SCTP congestion control scheme Show in Figure (2).

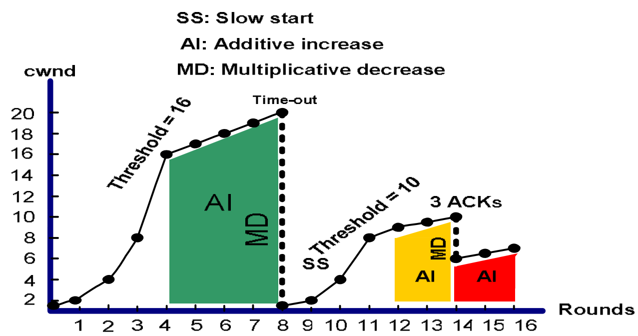

Fig 1: Additive increase/ Multiplicative-decrease (AIMD)

SCTP Congestion Control Algorithm: Initially:
cwnd = 2*MTU;
ssthresh = infinite;
New ack received:
if (cwnd<= ssthresh) /* Slow Start*/
cwnd = cwnd + acked chunk size from receiver;
if (cwnd>ssthresh) /* Congestion Avoidance */
cwnd = cwnd + MTU;
if (received 4 duplicate SACK) /*Fast retransmission*/
ssthresh = max(cwnd/2, 2*MTU);
cwnd = ssthresh;
Timeout: /* Multiplicative decrease */
ssthresh = max(cwnd/2, 2*MTU); cwnd = MTU;
Figure (2). SCTP Congestion Control Algorithm
	The pseudo code ofModifiedSCTP Congestion AvoidanceAlgorithm:
	
	Initially:
	cwnd = 2*MTU
	ssthresh = infinite
	$\boldsymbol{pba}$=**0**
	
	If (Data chunk loss) {
	/ / Congestion
	
	If (cwnd>ssthresh)& ($\boldsymbol{pba}$>$\boldsymbol{cwnd}$) {
	/ / Congestion Avoidance
	
	cwnd = cwnd - 0.144*cwnd
	}
	
	else ifcwnd<= ssthresh {/ / Slow Start cwnd = cwnd + acked chunk size from receiver}
	Figure (3).The pseudo code ofModifiedSCTP Congestion AvoidanceAlgorithm

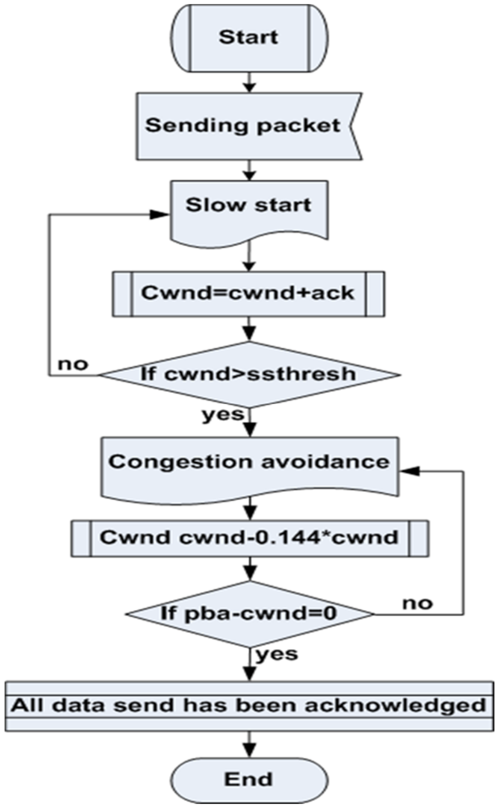

Fig 4: The flow chart o Modification SCTP Congestion Avoidance Algorithm

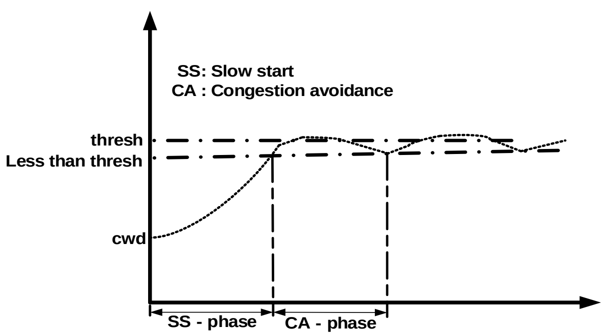

Fig 5: Behavior of congestion control through Modification Congestion Avoidance

III. Modified Congestion Avoidance Algorithm of SCTP

During slow- start, the cwnd is increased faster (roughly one MTU per received SACK chunk), but in the congestion avoidance algorithm, the size of the congestion window increase by roughly one MTU per round-trip-time (rtt) until congestion detected. However, this increase may lead to enter fast retransmit congestion control algorithm and fast recovery algorithm. For this reason the authors have proposed and modification of the Congestion Avoidance algorithm, we assume, During congestion avoidance algorithm for each acknowledgement received in a round trip time, the congestion window is decreased As given by equation: cwnd= cwnd- [0.144*cwnd],instead of cwnd = cwnd + MTU, illustrated in figure (3). This is will reduce congestion window less than slow start threshold, then the cwnd is fully utilized and immediately triggers slow-start phase work again, also the SCTP sender has no wait for a long time to receive the acknowledging and all available resources can utilize data sender rate Efficiently.

IV. Behavior of congestion control through Modification Congestion Avoidance

Before any data transfer takes place, $\boldsymbol{cwnd}$ is initialized to$cwndinit$, and $\boldsymbol{ssthresh}$ is initialized to some arbitrarily large value, the $\boldsymbol{cwnd}$ is incremented by $\Delta cwndss$, when: Current $\boldsymbol{cwnd}$ is fully utilized, Incoming Selective acknowledgement (SACK) advances the cumulative transmission sequence number(TSN)acked point and the sender is not in fast recovery the slow start phase is a self clocking mechanism, and congestion avoidance phase Starts to work, initialized Partial Bytes Acknowledged ($\boldsymbol{pba}$) = $\mathbf{0}$then $\boldsymbol{pba}$incremented by the total amount of data bytes acknowledged in SACK chunks, If $\boldsymbol{pba} \geq \mathbf{cwnd}$ that is means$\boldsymbol{cwnd}$>ssthresh will be decreased According to this equation 0.144*cwnd, if $\boldsymbol{pba} - \mathbf{cwnd}$ = 0 that means all data sent has been acknowledged by receiver, then SCTP$\boldsymbol{cwnd}$ reverts back to the slow start phase, illustrated in figure (4) and figure (5).

V. Network Simulation Model

The simulation evaluations were carried out with the use of OMNeT++ (in version 4.5) simulation framework extended with the INET package. The OMNeT++ is the modular, component-based simulator, with an Eclipse-based IDE and a graphical environment mainly designed for simulation of communication networks, queuing networks, congestion control and performance evaluation the framework is very popular in research and for academic purposes [8]. The INET Framework is the communication networks simulation extension for the OMNeT++ simulation environment and contains models for several Internet protocols: UDP, TCP, SCTP, IP, IPv6, Ethernet, PPP, IEEE 802.11MPLS, OSPF, etc. [9]. The simulated network was based on the example provided with the INET package to evaluate the behavior of the SCTP Congestion Control Algorithm a simple network with different traffic scenario The INET built in SCTP Congestion Control Algorithms are drop tail queue and RED tail drop algorithm. We add some improvement to the INET implementation. The network's topology is presented on Figure (6).

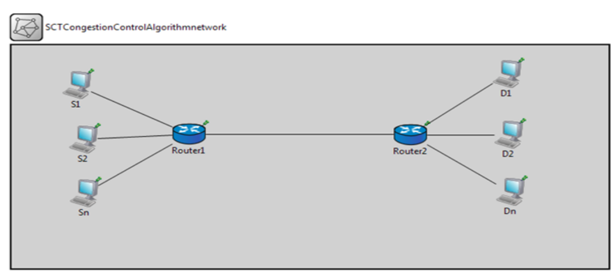

Fig 6: Network topology- testing congestion control through Modification Congestion Avoidance

VI. Objectives

We would like to study the Modified Congestion Control Algorithm for more adaptive the cwnd through Congestion Avoidance, that can be applied to an SCTP and that can utilize new, characteristic SCTP Congestion Control features. It aims to achieve the following specific contributions.

- To utilize data sender rate efficiently.
- To identify the limitations of congestion avoidance algorithm.
- To improve the good put values.

- To evaluate the AIMD mechanism for delivering over the network.

VII. Expected results of the proposed " Adaptive streams mechanism"

The new packets are injected into the network depending upon the size of the congestion window according to bandwidth size. During slow-start cwnd is increased exponentially to probe the network for capacity. If cwnd Became greater than ssthresh, then cwnd will be decreased by 0.144*cwnd, However, to become less than ssthresh, after that SCTP$\boldsymbol{cwnd}$ reverts back to the slow start phase. The expected results of the proposed mechanisms are as follow: The cwnd with Modification Congestion Avoidance is always close to thresh than original Congestion Avoidance this means more data could be injected in the network, therefore the SCTP with Modification Congestion Avoidance is ahead of SCTP without Modification Congestion Avoidance in its good put. Moreover, the SCTP with Modification Congestion Avoidance is ahead of SCTP with normal Congestion Avoidance in its percentage value of packet losses this means that SCTP with Modification Congestion Avoidance have lower values of packet losses, also SCTP Modification Congestion Avoidance is better than SCTP without Modification Congestion Avoidance in its delay. This is because of in short time can converting from Congestion Avoidance phase to the Slow Start phase.

VIII. Conclusions

The choice of congestion window increase and decrease mechanism requires careful analysis of the SCTP's response curve, bandwidth allocation properties. Thus the researchers proposed to reduce congestion window less than slow start threshold through the congestion avoidance algorithm to Avoid congestion quickly and utilize data sender rate efficiently. However, SCTP still needs more modifications to improve its throughout performance over high bandwidth delay networks Therefore, we propose a further modification for slow-start and congestion control mechanisms of SCTP by adjusting the congestion window size in first time.

IX. References

1. William Stallings, data and computer communications (Upper Saddle River, NJ 07458: Pearson Prentice Hall, Eighth Edition).
2. Behrouz Forouzan A. Data Communications and Networking (McGraw-Hill, Fourth Edition).
3. Jianping Zou MU. Mit Uyar, Mariusz A. Fecko, Sunil Samtani, Preferential Treatment of SCTP Subflows: Analysis and Simulation, Proc. IEEE Int. Symp. Comput. Commun. (ISCC), Alexandria, Egypt, 2004.
4. Tine Stegel, JanezSterle, JanezBešter, Andrej Kos, SCTP association between multi-homed endpoints over NAT using NSLP, Elektrotehniški vestnik 2008; 75(5):277-284.
5. Fang Cheng, Xiao-Fei Hang, Hong-Jiang Lei. a Congestion Control Scheme for LTE/SAE, International Conference on Computer Science and Information Technology (ICCSIT), 2011.
6. Katalin Tarnay, Gusztáv Adamis, Tibor Dulai. Advanced Communication Protocol Technologies: Solutions, Methods, and Applications, Information Science Reference (an imprint of IGI Global, ISBN) 978-1-60960-732-6 (hardcover)
7. OMNeT++ User Manual, Version 4.5 http://www.omnetpp.org/doc/omnetpp/Manual.pdf.
8. INET Framework Documentation. Retrieved from http://www.omnetpp.org/staticpages/index.php? Page=20041019113420757.
9. Meifang Shen, Yuanlong Cao, Mingchuan Zhang, Kaixuan Wang, Mingming Yu5. Evaluation of Performance of Background Traffic-based CMT-SCTP with Active Queue Management Algorithms, IJCSI International. Journal of Computer Science Issues. 2012; 9(6):2.
10. Nandgaonkar Vikas N, Patil Chandrashekhar G, Patil Sonali C. Comparative Analysis of TCP over Multi Streaming Protocol (SCTP) Based on Data Centers, International Journal of Electronics, Communication & Soft Computing Science and Engineering ISSN: 2277-9477, 2(6).
11. Syed Yasmeen Shahdad, Gulshan Amin, Pushpender Sarao, multihoming andmultistream protocol incomputer networks, IJCSMC, 2014; 3(3):1183-1187.
12. Stewart R, Tuexen M, Lei P. Stream Control Transmission Protocol (SCTP) Stream Reconfiguration, IETF Internet Engineering Task Force, 2012.
13. Firas Sabah Alturaihe. Reliable New Transport Protocol, IOSR Journal of Engineering (IOSRJEN), 2013; 3(10).
14. Osama Ahmed Bashir, Yahia Abdalla Mohamed, Mohamed Awad. The delay experienced by packets on routers, SUST Journal of Engineering and Computer Science (JECS), 2015, 16(2).
15. Michael Tüxen. Considerations on Handling Link Errors in SCTP, ICB-Research Report No. 35, ISSN 1860-2770, 2009.