International Journal of Applied Research 2017; 3(6): 737-749

ISSN Print: 2394-7500
ISSN Online: 2394-5869
Impact Factor: 5.2
IJAR 2017; 3(6): 737-749
www.allresearchjournal.com
Received: 11-04-2017
Accepted: 12-05-2017

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

Amin B Abdel Nabi
Department of Communications, College of Engineering, Al-Neelain University, Khartoum, Sudan

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

Avoidance frequent failure over SCTP multi-homing

Elhabib Elhabib A and Amin B Abdel Nabi

Abstract
The multi-homing feature enables an SCTP session to be established over multiple interfaces identified by multiple IP addresses. SCTP normally sends packets to a destination IP address designated as the primary address, but can redirect packets to an alternate secondary IP address if the primary IP address becomes unreachable. However, the advantages in the use of multi-homing will waste. If temporary congestion occur may confuse an SCTP endpoint as to which loss indicates either that the network path to the primary destination is congested, or the primary destination is unreachable. For this reason we modified SCTP congestion control mechanism to prevent unnecessary failovers due to temporary congestion may lead to false SCTP failure detection mechanism in multi-homing to appeared, and make spurious retransmissions and stalling path at long time continuous packet timeouts to reach the inactive threshold. The modified of SCTP congestion control mechanism depends on adding new stage to congestion control mechanism. The new stage set after Congestion Avoidance phase and before fast retransmission phase. When the congestion window become large through Congestion Avoidance phase and transmitter does not received heartbeat-ack, the new stage of SCTP congestion control mechanism trigger enable to reduce congestion window related to packet to improve the transmissions on the path according to the available bandwidth as a long time and get advantage of multi-homing feature. Therefore, we can improve performance of multi-homing, over the default SCTP congestion control mechanism through proposed method, by using OMNet++ network simulator showing that Modification can effectively improve the multi-homing performance. Proposed method is simple and provides a more effective performance than SCTP default SCTP congestion control mechanism.

Keywords: congestion window size, multi-homing, failure, OMNeT++

1. Introduction

Stream Control Transmission Protocol (SCTP) is an internet standard transport protocol that includes the functions of TCP and UDP, as well as additional functionalities; one of these is supporting multi-homing, which allows multiple source and destination IP addresses to be associated with an SCTP connection [1]. The multi-homing support, only the primary path interface is used and alternate paths interfaces are considered as secondary and used only to provide fault tolerance; so the retransmit lost packets increase the probability of successful reception and transmission of new packets, when the primary is declared inactive, in which case the secondary is turned primary [2]. SCTP can deal with multiple interfaces and unfortunately inherits congestion control scheme from TCP. The SCTP has a multi-homing feature and there are two factors have effect on SCTP transmission efficiency: the size of the congestion window of each transmission path and the active status of each path [3]. In this paper we focuses on optimizing the congestion window size due to Packet loss, in wired networks is mainly the result of network congestion. In the standard SCTP congestion control the sender begin with slow start algorithm, According to equation: Congestion window (cwnd) = cwnd + Acknowledged (acked) chunk size from receiver [4]. The congestion window limits number of bytes, the sender is allowed to transmit before waiting for a new acknowledgement that means, more than cwnd bytes may be outstanding [5]. 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 [6]. If the value of cwnd is greater than the ssthresh value, the Congestion Avoidance phase trigger to work and if the network still congested, the network is forced to drop one or more packets due to overload, In this case all facilities provided by multi-homing will be useless and possibly may lead to unnecessary failovers due to temporary congestion may lead to false SCTP failure detection mechanism to work, make spurious retransmissions and stalling path at long time continuous packet timeouts to reach the inactive threshold. However the sender uses the congestion avoidance algorithm according to equation: cwnd = cwnd + Maximum Transmission Unit (MTU), actually it does not solved the problem because due this mechanism the current congestion window increase by MTU, and this increase may lead to high timeout and higher timeout takes longer time to detect a path failure make the interval of heartbeat affect by this delay, so for this reason we propose new stage of SCTP congestion control mechanism depends on adjusting the congestion window size after Congestion Avoidance phase and before fast retransmission phase, the adjusting based on available network capacity, through equation: cwnd= cwnd- [0.144*cwnd]. The author looking for minimum delay between each retransmission on the path that elapses after a packet has been sent, to get high effectively to handle network failures or unexpected highly traffic, and ensure quick recovery from data congestion also look for multi-homing operation stay along time in statues of first primary path selected. So the optimum multi-homing feature achieved through adding a new stage to the current SCTP congestion control mechanism, based on available network capacity. However the performance of the proposed enhanced SCTP multi-homing and evaluated by OMNeT++ simulations. The Omnet++ is an important simulator programs which is used to simulate various network domain [7]. The remainder of the paper is organized as follows. Section II: Overview SCTP multi-homing operation, while in section III: SCTP Congestion Control mechanism. The section IV: described behavior of proposed method, and section V: presented the multi-homing behavior. In section VI: Multi-homing through current Congestion Control and the Multi-homing through proposed method in section VII. The section VIII: is the Objectives, in section IX: the Network Simulation Model and in section X: the Results and Analysis of the proposed and finally, the Conclusions and Future Work in section XI.

2. Overview SCTP multi-homing operation

Multi-homing support is one of the major advantages of SCTP which is not supported in other transport protocols such as TCP or UDP [8]. If a multi-homed end-point has redundant network connections, SCTP sessions can survive from the network failures by migrating inactive path to active one, this feature can be expected to be a driving force for deploying SCTP [9]. SCTP can utilize multiple IP addresses for single SCTP association each SCTP endpoint exchanges the list of available addresses on the node during initial negotiation Show in Figure (1). After this endpoints select one address from the list and define this as the destination of the primary path is used for initial data transmission and in the case of retransmission an alternate path is used see Figure (2) [10]. SCTP's current retransmission policy attempts to improve the chance of success by sending heartbeat messages to monitor an SCTP endpoint to indicate that the path status active or inactive mode Show in Figure (3). After a specific number of retransmissions called path.max.retrans (PMR) which is the number of maximum retransmissions on a path before it is considered unreachable, when the path is considered inactive the new path is chosen [11]. The Retransmission Timeout (RTO) duration represents the delay between each retransmission on the path as shown in Figure (4) [12]. Basically the initial RTO between endpoints as recommended value equal 3000 ms which is the value of the RTO before any RTT measurements have been made, and 60000 ms as maximum value which is the highest allowed value for RTO. If an SCTP sender does not receive a response for an SCTP data chunk from its receiver within the time of RTO, the sender will consider this data chunk lost and then RTO is calculated for each destination address separately based on the Smoothed Round-Trip Time (SRTT) and Round-Trip Time Variation (RTTVAR) of the path [13]. The SRTT and the RTTVAR of a path are calculated by the measurement of RTT of the path. The RTT measurement for a path is made for every round trip time [14]. When SCTP gets the first measurement of RTT, the RTT.1st, SRTT and RTTVAR are initialized as: SRTT = RTT.1st (1), RTTVAR = $\frac{\text{RTT.1st}}{2}$ (2) And RTO is updated according to RTO = SRTT+4*RTTVAR (3), for each time SCTP gets a new measurement of RTT: RTT. new. The SRTT and RTTVAR will be updated as follows: RTTVAR. new = (1-β) * RTTVAR. old+ β * (SRTT. old- RTT. new) (4) and calculated new RTO according to SRTT. new = (1- α) * SRTT. old+ α * RTT. new (5) Where β and α are constants and their recommended values are 1/4 and 1/8 respectively. Then the new RTO is: RTO = SRTT. new + 4 * RTTVAR. new (6) If the new RTO is less than RTO. Min which is the lowest allowed value for RTO equal 1000ms as the Recommended Value, it will be set to RTO. Min. If the new RTO is greater than RTO. Max, it will be set to RTO. Max [15]. Every time a transmission timeout occurs for an address, the RTO for this address will be doubled: RTO = RTO*2 (7) and if the new RTO is greater than RTO. Max, RTO. Max will be used for the new RTO. If the sender gets a response from the receiver and a new RTT is measured, SCTP will use this new RTT to calculate RTTVAR, SRTT and finally RTO calculated by the equations (4) to (6) [16].

3. SCTP Congestion Control mechanism

SCTP and TCP support the same set of congestion control algorithms. The slow-start, congestion avoidance, and the fast retransmit mechanisms of SCTP have been almost directly inherited from TCP. SCTP uses three control variables Similar to TCP: rwnd, cwnd, and ssthresh. However, unlike TCP, SCTP's cwnd reflects how much data can be sent, but not which data to send; thus, the cwnd in SCTP truly windows the amount of data that may be sent. SCTP also introduces an additional variable, partial bytes acked (pba), to calculate cwnd growth during the Congestion Avoidance phase [17]. In multi-homing mode, SCTP has a separate set of congestion control parameters for each of multiple transport paths within an association [18]. The goal of congestion control mechanisms is to prevent senders from blocking links by reducing the rate of sending packets [19]. A very important feature to control this rate, is the cwnd, it determines the amount of data that may be in flight on the link that is the data, which has not been acknowledged yet, plus the data, that may be injected into the network [20]. The first congestion control mechanism operates is the slow start, It operates for cwnd values less than or equal to the slow start threshold, which is set to an arbitrary value (mostly the advertised receiver window of the peer during association setup) at the beginning of an association [21]. Slow start is characterized by an exponential increase of the congestion window. Every time an incoming SACK-chunk announces that the cumulative TSN ack parameter has advanced and the cwnd is fully utilized, such as the number of outstanding bytes is greater than cwnd, the minimum of the path MTU and the acknowledged bytes is added to cwnd [22]. When cwnd exceeds the slow start threshold, congestion avoidance algorithm, makes for a linear increase of cwnd that means increase by roughly one MTU per round-trip-time (rtt), packet loss is the consequence. SCTP uses pba to facilitate this mechanism. Initially, pba is set to zero. When a SACK that advances the cumulative ack arrives, pba is incremented by the number of bytes newly acked in the SACK, as determined by the cumulative and selective ack feedback [23]. While fast retransmissions result in halving the congestion window, a timer based retransmission leaves cwnd at the size of the path MTU and in slow start again. Thus cwnd follows usually a zigzag curve in the lifetime of a connection as Shown in Figure (5). However, this increase may lead to enter fast retransmit congestion control algorithm and fast recovery algorithm.

4. Behavior of proposed method

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. The receiver send SACK and the sender not received SACK 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 $\boldsymbol{pba}\geq\mathbf{cwnd}$ this is means $\boldsymbol{cwnd}$>ssthresh, so the congestion avoidance phase trigger to work allows the window size to increase linearly by one MTU. If the network still congested the proposed congestion avoidance phase Starts to and the new congestion window will be decreased According to our proposed algorithm, cwnd = cwnd - 0.144*cwnd. The choice of this value 0.144 is very important things because if it is too small does not help enough the network to reduction congestion and if it is too big lead to degradation of the performance in the network. So if the value of congestion window equals 950 then the new congestion window will be 950 - 0.144*950 therefore the new congestion window equal 813.2. If the network still congested the new congestion window continue will decrease to 696.0992, if the $\boldsymbol{pba} - \mathbf{cwnd}$ = 0 this means all data sent has been acknowledged by receiver, then SCTP$\boldsymbol{cwnd}$ update from last cwnd decrease and reverts back to the slow start phase, else fast retransmission trigger to work and reduce the threshold to one-half of the previous window size. So the minimum Delay, less packet error rate and high data rate achieved during adding new stage after the congestion avoidance phase. The network appear staying along time in slow start phase see figure (6) and figure (9).

5. The multi-homing behavior

Multi-homing and Path Supervision a feature that is unique to SCTP is multi-homing, which means that an endpoint may have several IP addresses that can be used as destination addresses to reach this endpoint. During the association setup the endpoints announce their addresses in the address parameter of the INIT or INIT ACK chunk, respectively. One address is selected as primary path, either explicitly by the upper layer or the destination address of the first message is used. The availability of the paths is supervised by sending HEARTBEAT chunks every (HBInterval + RTO) seconds to all known endpoint addresses. If the corresponding HEARTBEAT_ACK chunk arrives, this means path is stated confirmed. The value for $HB_{Interval}$ is usually 30 $_{second}$. The first value for RTO is RTO_{Min} which is one $_{second}$ by default. The next values are calculated by taking the round trip time (RTT) into account. The RTT is measured by either calculating the time difference between the sending of a HEARTBEAT chunk and the arrival of the corresponding HEARTBEAT $ACK_{chunk,}$ or the first transmission of a user message and its acknowledgment in the SACK chunk, considering only the DATA chunks that have not been retransmitted. The computation of RTO in SCTP follows closely the rules for TCP in RFC 2988 [24]. HEARTBEAT chunks are only sent when the path has been idle for some time. In case the acknowledgment does not arrive, RTO is doubled and a path error counter is increased. The maximum number of trials for the path supervision is configured in the Maximum Path Retransmission (MPR) parameter. After a specific number of retransmissions PMR which is the number of maximum retransmissions on a path before it is considered unreachable, when the path is considered inactive the new path is chosen. RTO duration represents the delay between each retransmission on the path. The next confirmed path is used. As the properties and the usage of the paths are different from each other, the calculation of the RTO has to be done for each path individually.

6. Multi-homing through current Congestion Control

The congestion control mechanism that SCTP uses is in most parts derived from TCP. Yet some important differences are due to special SCTP features. As SCTP allows a host to be multi-homed, the congestion control mechanism has to be applied to each path separately. This means that a path has its own congestion window, slow-start threshold (ssthresh), the slow start threshold (ssthresh), which is set to an arbitrary value (mostly the advertised receiver window of the peer during association setup) at the beginning of an association. Slow start is characterized by an exponential increase of the congestion window. Every time an incoming SACK chunk announces that the Cumulative TSN Ack parameter has advanced and the cwnd is fully utilized, that is the number of outstanding bytes is greater than cwnd, the minimum of the path MTU and the acknowledged bytes is added to cwnd. When cwnd exceeds the slow start threshold. The variable partial Bytes Acked counts the number of acknowledged bytes, and only if they reach the size of cwnd, may an MTU be added. All other cases lead to a decrease of cwnd. When a fast retransmission is necessary, the window is halved or set to 4. MTU, in case of a timer based retransmission it is even left at one MTU. The cwnd timer expires if the cwnd has not been updated for a set time indicating that the path has been idle. Therefore, the condition of the path is not known, and the cwnd is decreased to the value at initialization time as shown in figure (8). The cwnd may not grow beyond the number of outstanding bytes plus maxBurst times the MTU. Max Burst is set to 4 by default and limits the number of packets that may be sent at once.

7. Multi-homing through proposed method

SCTP computes RTO values effect by RTT measurements illustrated in figure (3). The high RTT measurements lead to high RTO values and the high delay in RTO caused the standard path failure detection strategy trigger to change the status of primary path to in inactive mode as shown in figure (4), this problem appeared during the standard path failure detection strategy does not distinguished between the network congested and the destination unreachable. The cwnd in the current Congestion Control be increased roughly by one MTU per RTT through congestion avoidance algorithm and in fast retransmission The cwnd decrease to half theses mechanism make that path has been idle hastily so for this reason in our proposal we added new stage set after congestion avoidance phase and before fast retransmission phase, we assume that, the congestion window decreased by cwnd= cwnd – (0.144*cwnd), the decreasing of cwnd will be continues until the network appear without congestion see figure (10) otherwise the standard path failure detection strategy trigger to change the status of primary path to in inactive mode as shown in figure (7).

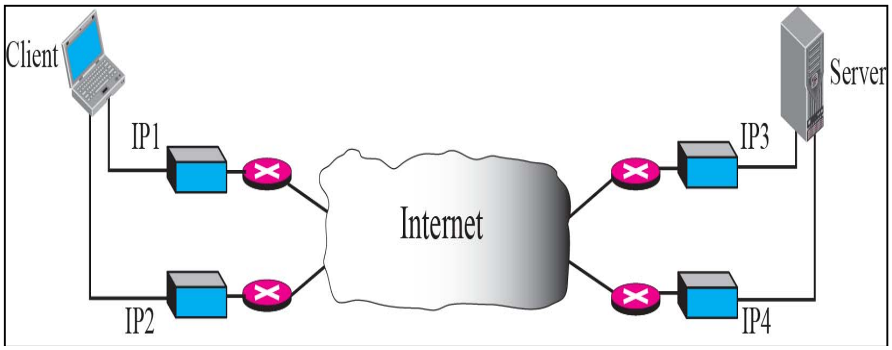

Fig 1: Multi-homing concept

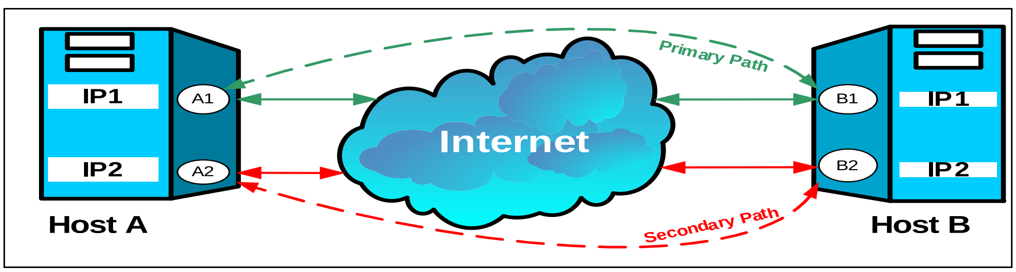

Fig 2: Primary and Secondary paths

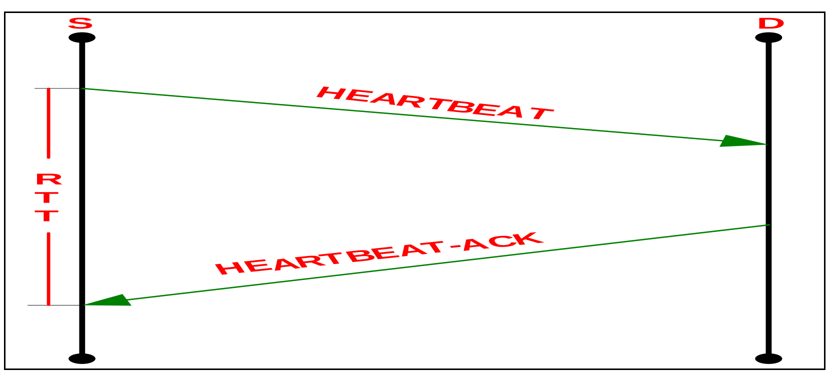

Fig 3: Heartbeat messages

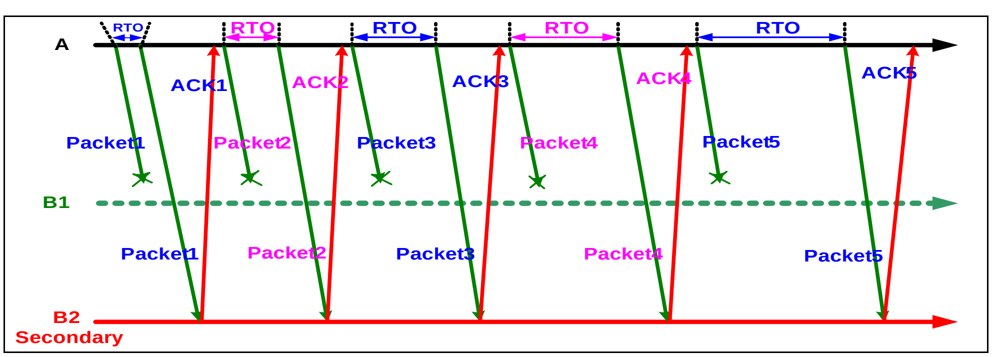

Fig 4: The RTO duration

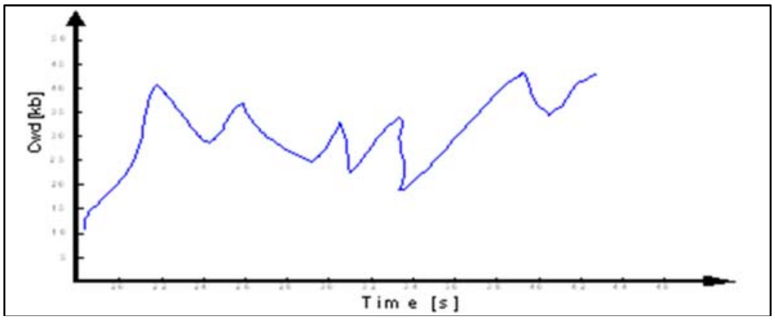

Fig 5: The lifetime of a connection windows

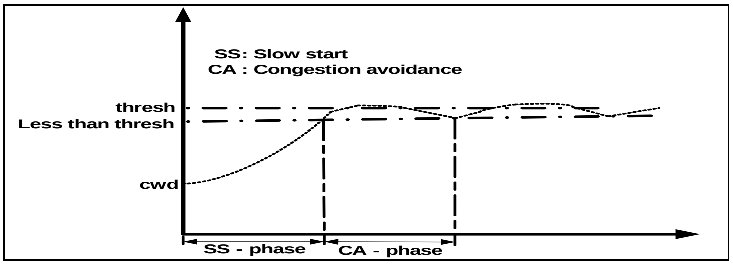

Fig 6: Behavior of congestion control through Modification Congestion Avoidance

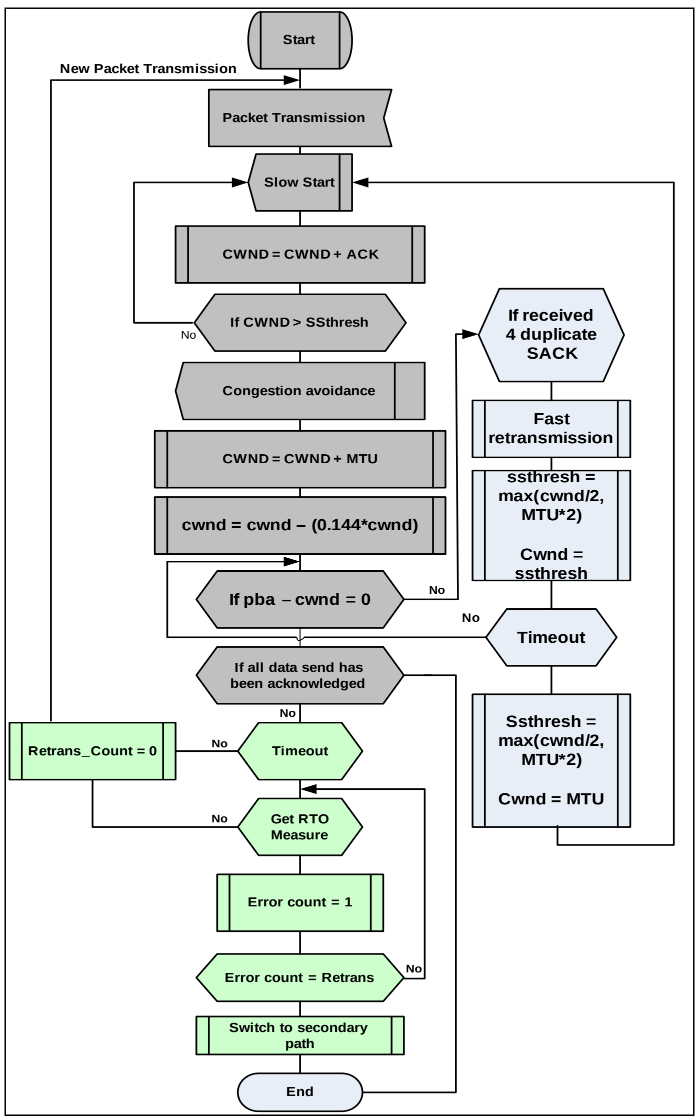

Fig 7: the flow chart of multi-homing through proposed method

SCTP Congestion Control Algorithm:
Initially:

```
cwnd = 2*MTU;
pba = 0;
ssthresh = infinite;
{
New ack received:
If (cwnd<= ssthresh)
New cwnd = cwnd + acked;
{
else (cwnd>ssthresh) & (pba > cwnd)
{
cwnd = cwnd + MTU;
else
(received 4 duplicate SACK)
ssthresh = max(cwnd/2, 2*MTU);
cwnd = ssthresh;
Timeout: /* Multiplicative decrease */
ssthresh = max(cwnd/2, 2*MTU);
cwnd = MTU;
}
```

Figure (8). SCTP Congestion Control Algorithm

The pseudo code of Modified SCTP Congestion Avoidance Algorithm:
Initially:

```
cwnd = 2*MTU;
pba = 0;
ssthresh = infinite;
{
New ack received:
If (cwnd<= ssthresh)
New cwnd = cwnd + acked;
{
else (cwnd>ssthresh) & (pba > cwnd)
{
cwnd = cwnd + MTU;
If (cwnd>ssthresh)
{
cwnd = cwnd - 0.144*cwnd;
else
(received 4 duplicate SACK)
ssthresh = max(cwnd/2, 2*MTU);
cwnd = ssthresh;
Timeout: /* Multiplicative decrease */
ssthresh = max(cwnd/2, 2*MTU);
cwnd = MTU;
}
```

Figure (9) The pseudo code of Modified SCTP Congestion Avoidance Algorithm

8. Objectives

The authors would like to study the behavior of SCTP on switchover in a multi-homed environment in order to identify the limitations of current SCTP switchover management such as switchover delayed and Offer solutions to enhance switchover performance to allow SCTP to detect the path switchover earlier than the standard mechanism. Also one of the main objectives of the study to overcome the shortcoming related to current RTO Calculation, Considering that RTO selection has significant impact on switchover may cause degradation in network quality as path failure. So the modified of SCTP congestion control mechanism to prevent unnecessary failovers due to temporary congestion and we would also like to increase the performance in networks and avoiding service interruption. Our assumption is that the solution could be based on Control Algorithm for more adaptive the cwnd, that can be applied to an SCTP and that can optimized Congestion utilize new, characteristic SCTP Congestion Control features to utilize data sender rate efficiently and improve the good put values.

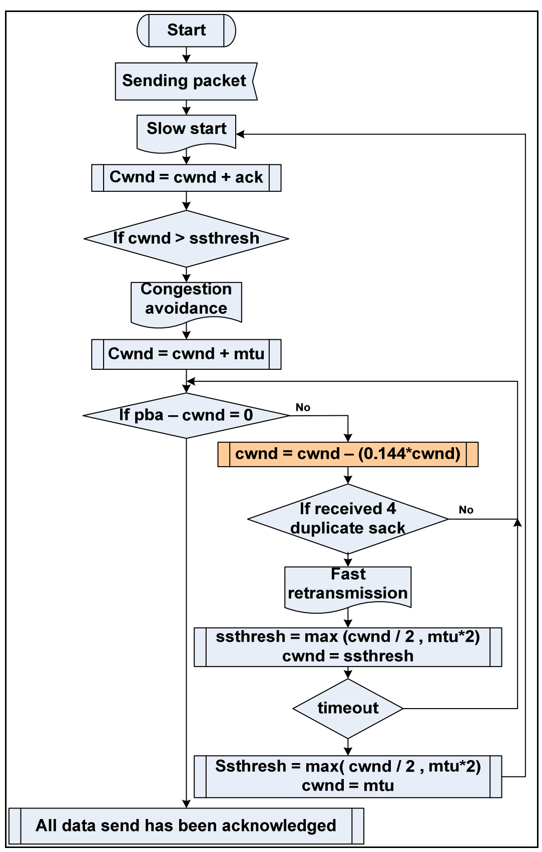

Fig 10: the flow chart of Modification SCTP Congestion Avoidance Algorithm

Simulations were carried using the OMNeT is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators, includes wired and wireless communication networks, on-chip networks and queuing networks [25]. The INET framework for the widely used OMNeT++ simulation environment supports discrete event simulation for IP-based networks [26]. Also has been extended to support external interfaces. These interfaces allow setting up hybrid scenarios where simulated nodes communicate with real external IP-based nodes [27]. The simulated network was setup in OMNeT++ to simulate the SCTP multi-homing environment each endpoint is defined as an SCTP-HOST and an association connection was set up between these two hosts. Duplex links were set up between the hosts in different endpoints. Each link has a bandwidth of 1Gbit/second and a propagation delay to 30 ms, the router queue is set to 20 packets. The path MTU is set to 3500 Bytes and one of these links were set to be the primary path. A heartbeat is sent out to all the nodes at 60-second intervals, using the default SCTP *RTO* parameters and parameter and PMR as default, The INET built in SCTP Congestion Control Algorithms are drop tail queue and RED tail drop algorithm. The network's topology is presented on Figure (10). We add some improvement to the INET implementation through SRC file, transport file, and SCTP file in SCTPAssociationBase.cc file.

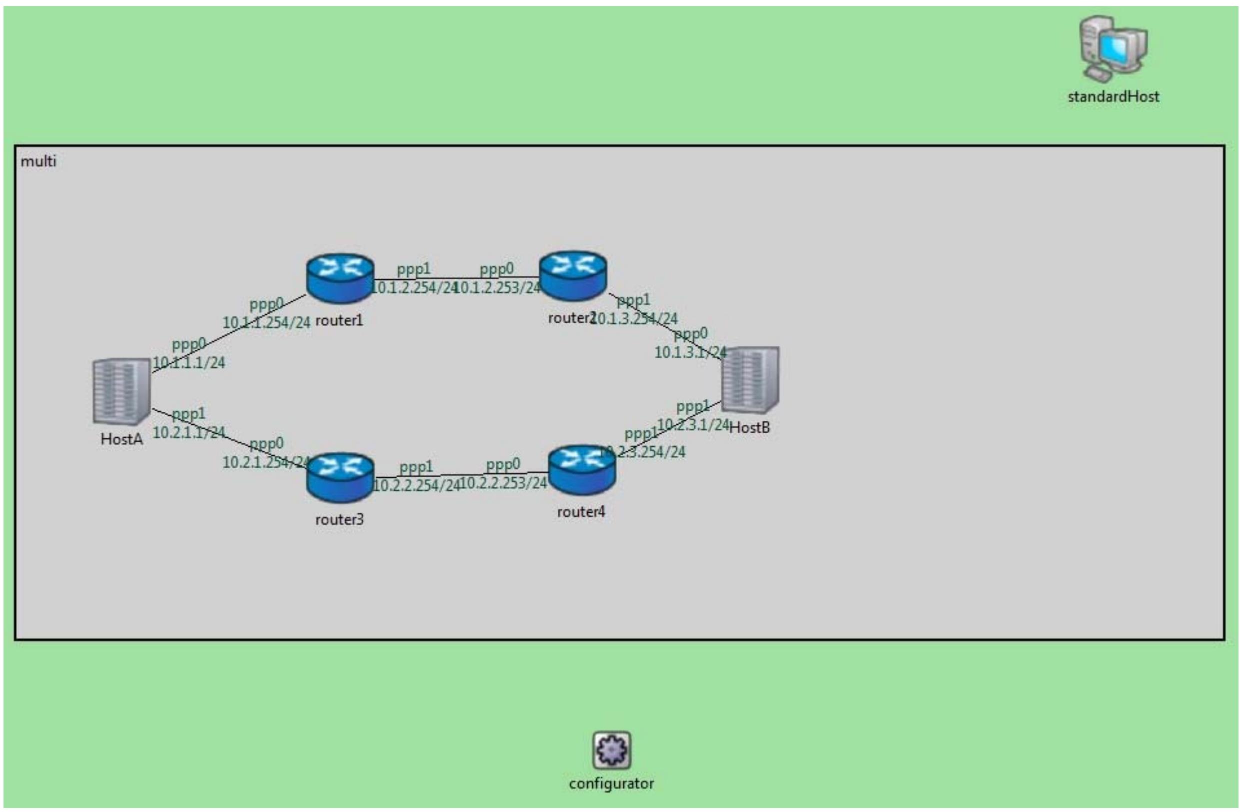

Fig 11: simulation topology for impact of Congestion control through multi-streaming performance.

10. Results and Analysis

The results appear in two scenarios. The first scenario implemented before we applied the new stage between the Congestion Avoidance phase and the fast retransmission phase, all scenarios Configured same Parameters, in The first scenario Host (A) in SCTP App [0] opened sessions (1) for transmitting 528000 bytes to Host (B) SCTP App [0] Host (A) retransmitted 128000 bytes of 528000 bytes. All byte sent to Host (B) 400000 with the life time of Association 8.31 second and it has 384921 bit/s throughput. In the other hand Host (B) received 400000 bytes with the life time of Association 8.25 second and 387879 bit/sec throughput. The second scenario Host (A) in SCTP App [0] opened sessions (1) for transmitting 516000 bytes to Host (B) SCTP App [0] Host (A) retransmitted 116000 bytes of 516000 bytes. All byte sent to Host (B) 400000 with the life time of Association 5.36 second and it has 596921 bit/s throughput. In the other hand Host (B) received 400000 bytes with the life time of Association 5.29 second and 604063 bit/sec throughput. We get the simulation results after the simulation run and wait for certain time. The results are captured inside the project's result folder as the tow type of result. The vector tab consist of the values captured during the simulation like cwnd, RTO, RTT, end to end delay, received bytes and advertised receiver window. Similarly the scalar tab contains all the scalar values like Association Lifetime, throughput, Transmitted Bytes, Packets, Retransmissions and Heartbeats. All results in this paper as the vector results in the two scenarios.

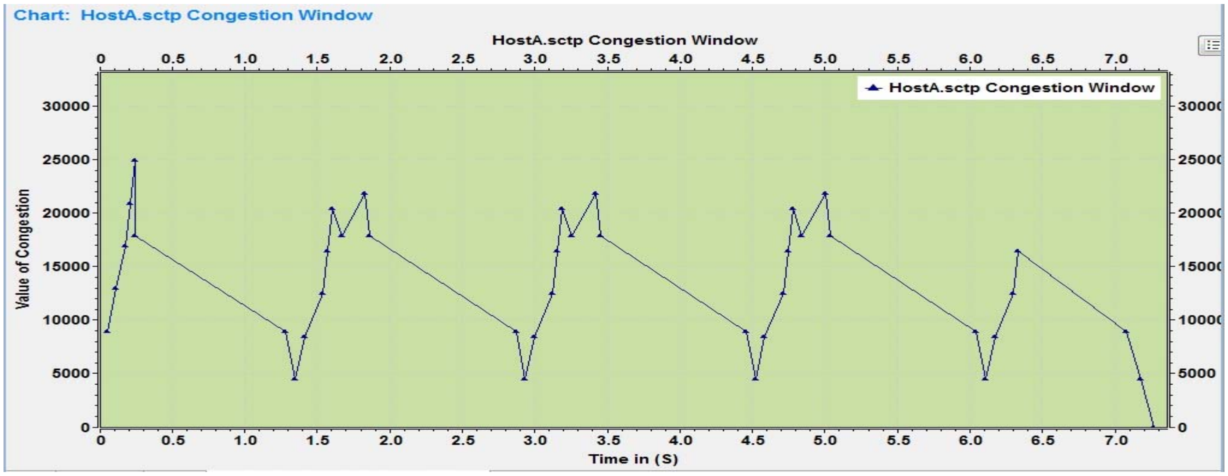

Fig 12: the cwnd during the standard of to SCTP congestion control mechanism.

Before any data transfer takes place, the cwnd is initialized and take 24940.0 as the max value equal the cwnd during new stage and the Mean value equal 13502.44 less than the mean value when we applied new stage this indicate the new stage has high throughput, also the StdDev value and the variance indicate the new stage has high throughput.

Table 1: parameters of cwnd during the standard of to SCTP congestion control mechanism.

Count	Start time	End time	Max	Mean	Min	StdDev	Variance
41	0.0521744	7.26556	24940.0	13502.44	0.0	6213.951	3.861321

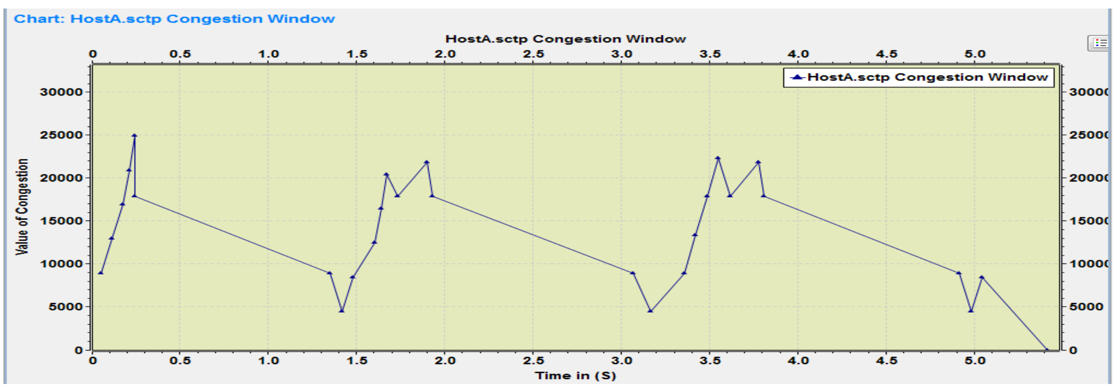

Fig 13: the cwnd during the new stage of to SCTP congestion control mechanism.

Table 2: parameters of cwnd during the new stage of to SCTP congestion control mechanism.

Count	Start time	End time	Max	Mean	Min	StdDev	Variance
28	0.0521744	5.41302	24940.0	13822.143	0.0	6580.729	4.33051

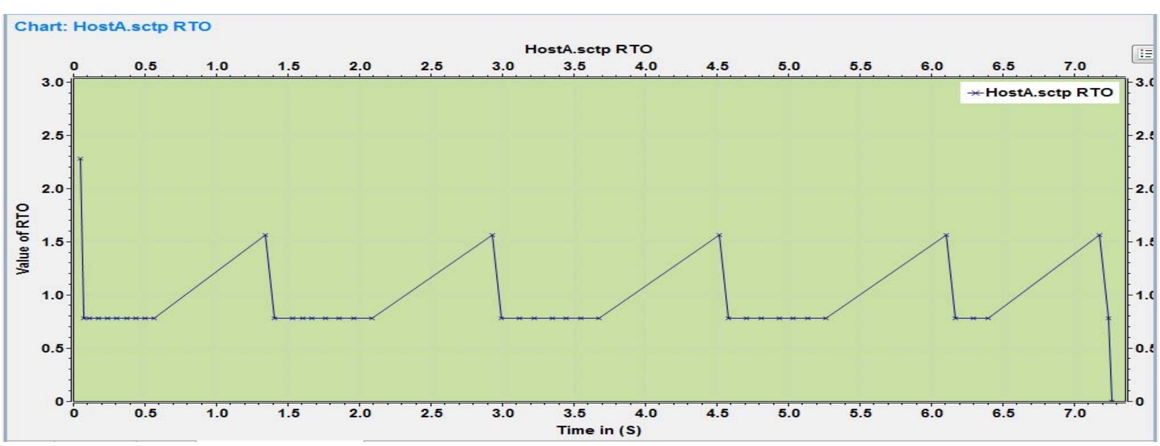

Fig 14: the RTO during the standard of to SCTP congestion control mechanism

The RTO duration represents the delay between each retransmission on the path; so the SCTP detects path failure through the duration of RTO, the max delay of the RTO during the standard of to SCTP congestion control mechanism is 2.28, when the max delay of the RTO during the new stage of to SCTP congestion control mechanism is 2.14, so the new stage of to SCTP congestion control mechanism offered minmum RTO duration and also the new stage of to SCTP congestion control mechanism produced minmum mean values, StdDev values and variance values therefore the performance of multi-homed switchover management enhance and able to detect path failure in earlier time.

Table 3: parameters of RTO during the standard of to SCTP congestion control mechanism.

Count	Start time	End time	Max	Mean	Min	StdDev	Variance
42	0.0521744	7.26556	2.28	0.89	0.0	0.362653	0.131517

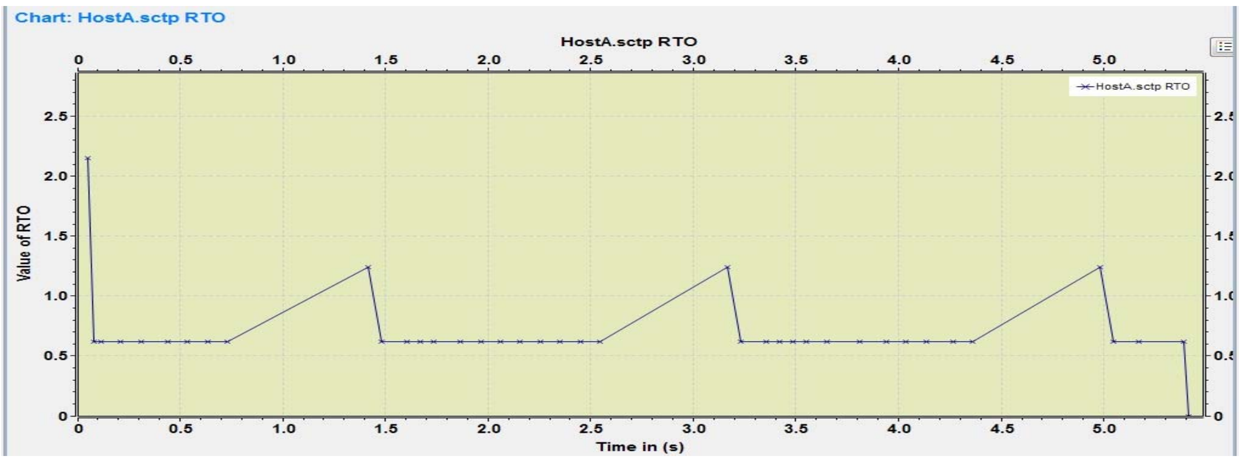

Fig 15: the RTO during the new stage of to SCTP congestion control mechanism.

Table 4: parameters of RTO during the new stage of to SCTP congestion control mechanism.

Count	Start time	End time	Max	Mean	Min	StdDev	Variance
40	0.0521744	5.41302	2.143456	0.68925	0.0	0.30746	0.094530

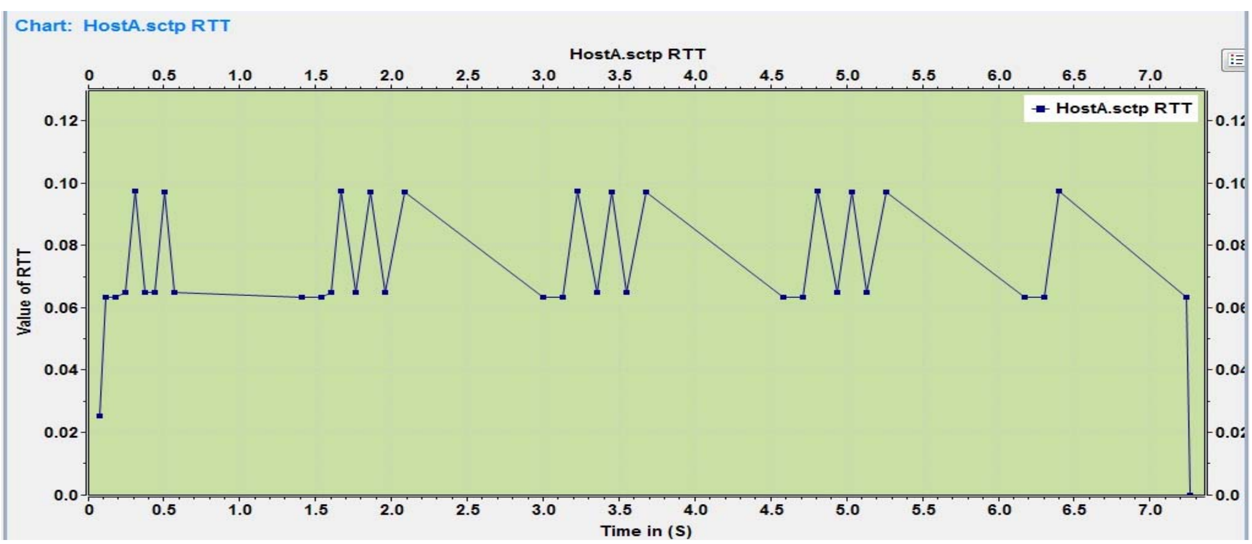

Fig 16: the RTT during the standard of to SCTP congestion control mechanism.

The RTT reflects the degree of congestion and packet loss rate on the path, also used to measure the delay difference between each path to estimate the bandwidth. the max delay of the RTT during the standard of to SCTP congestion control mechanism is 0.974, when the max delay of the RTT during the new stage of to SCTP congestion control mechanism is 0.183, this means the new stage of to SCTP congestion control mechanism offered minmum RTT duration for all period and minmum in the mean values, StdDev values through data transmission see table (6) and tables (5).

Table 5: parameters of RTT during the standard of to SCTP congestion control mechanism.

Count	Start time	End time	Max	Mean	Min	StdDev	Variance
36	0.077336	7.26556	0.973968	0.0723688	0.0	0.021651	4.687733

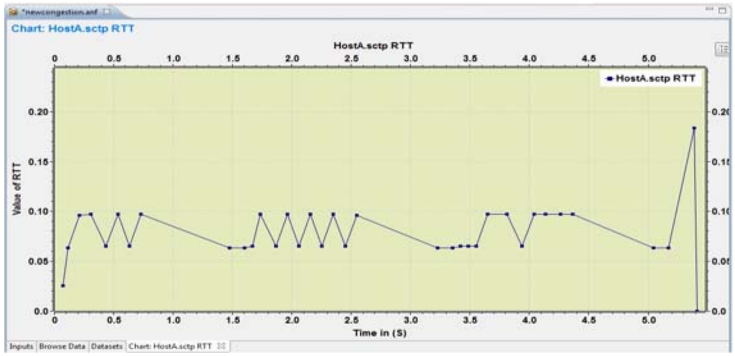

Fig 17: the RTT during the new stage of to SCTP congestion control mechanism.

Table 6: parameters of RTT during the new stage of to SCTP congestion control mechanism.

Count	Start time	End time	Max	Mean	Min	StdDev	Variance
36	0.077336	5.41302	0.183456	0.078441	0.0	0.028706	8.24057

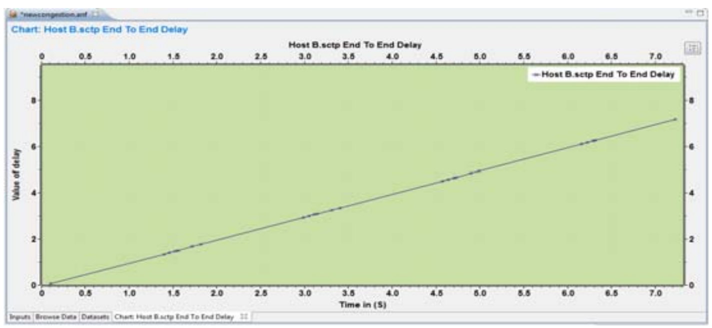

Fig 18: the end to end delay during the standard of to SCTP congestion control mechanism.

The end to end delay between the two scenarios they represent the delay of transmission data and they have the same a count and the start time but differ in time of complete transmission, the standard of to SCTP congestion control mechanism has end time 7.23, when the end time during the new stage of to SCTP congestion control mechanism is 5.17, so the new stage of to SCTP congestion control mechanism offered end to end delay and also the new stage of to SCTP congestion control mechanism produced minmum mean values, StdDev values and variance values therefore the performance of multi-homed switchover management enhanced see table (8) and tables (9).

Table 7: parameters of end to end delay during the standard of to SCTP congestion control mechanism.

Count	Start time	End time	Max	Mean	Min	StdDev	Variance
100	0.103146	7.22817	7.1760	3.8243	0.05097	1.8607	3.4623

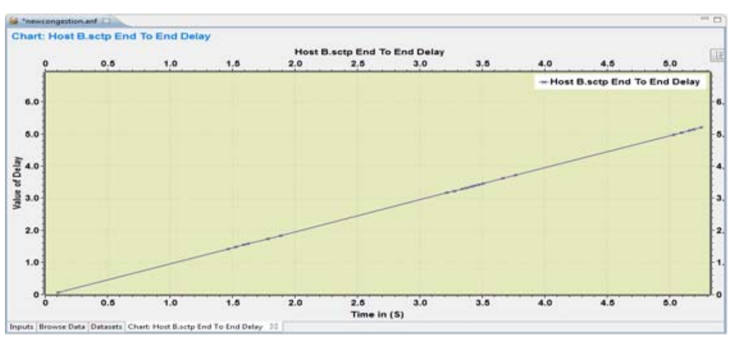

Fig 19: the end to end delay during the new stage of to SCTP congestion control mechanism.

Table 8: parameters of end to end delay during the new stage of to SCTP congestion control mechanism.

Count	Start time	End time	Max	Mean	Min	StdDev	Variance
100	0.103146	2.2556	5.203455	3.188	0.05097	1.402	1.966

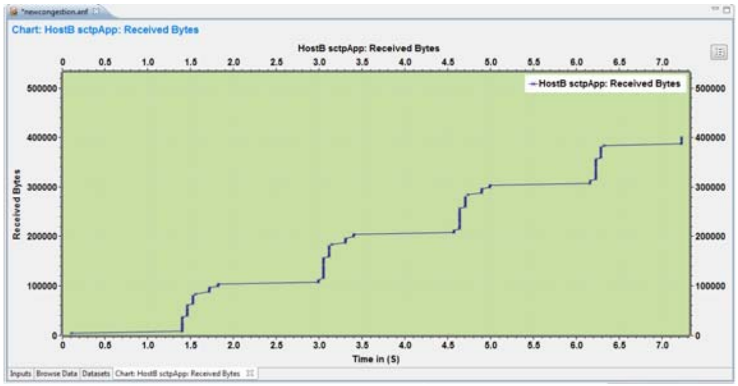

Fig 20: the received bytes during the standard of to SCTP congestion control mechanism.

We can see the difference between the two scenarios especially in the total of receiving bytes. The difference appear the new stage of to SCTP congestion control mechanism produced minmum delay 5.25, when we compared with the standard of to SCTP congestion control mechanism it has 7.23 delayed, so the new stage ofred 1.98 second this indicate that the new stage has high throughput.

Table 9: parameters of received bytes during the standard of to SCTP congestion control mechanism.

Count	Start time	End time	Max	Mean	Min	StdDev	Variance
100	0.103146	7.22817	400000.0	202000.0	4000.0	116045.9	1.347

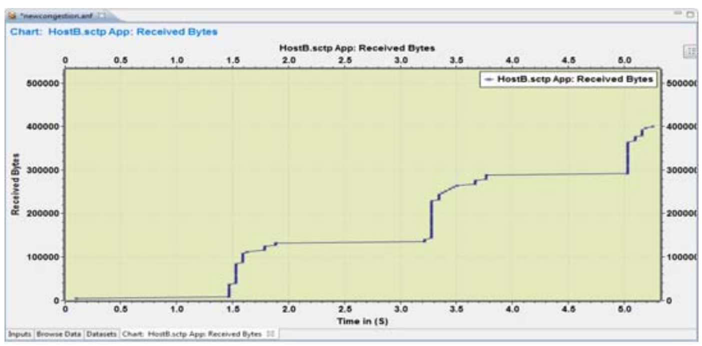

Fig 21: the received bytes during the new stage of to SCTP congestion control mechanism.

Table 10: parameters of received bytes during the new stage of to SCTP congestion control mechanism.

Count	Start time	End time	Max	Mean	Min	StdDev	Variance
100	0.103145	5.2557	400000.0	202000.0	4000.0	116045.9	1.347

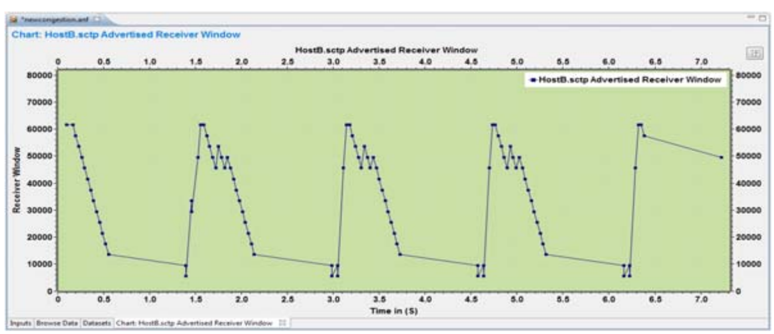

Fig 22: the advertised receiver window during the standard of to SCTP congestion control mechanism.

The advertised receiver window between the two scenarios they represent the throughput of transmission data between the end point. The two scenarios have the same a count and the start time but differ in time of complete transmission, the standard of to SCTP congestion control mechanism has end time 7.23, when the end time during the new stage of to SCTP congestion control mechanism is 5.37, so the new stage of to SCTP congestion control mechanism offered high performance in end time value, mean values, StdDev values and variance values therefore the performance of multi-homed enhanced see table (11) and tables (12).

Table 11: parameters advertised receiver window during the standard of to SCTP congestion control mechanism.

Count	Start time	End time	Max	Mean	Min	StdDev	Variance
95	0.103146	7.228170	61535.0	36987.6	5535.0	18074.178	3.266759

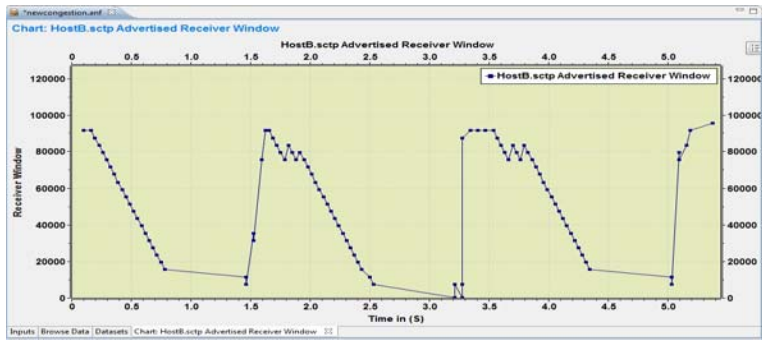

Fig 21: the advertised receiver window during the new stage of to SCTP congestion control mechanism.

Table 12: parameters of advertised receiver window during the new stage of to SCTP congestion control mechanism.

Count	Start time	End time	Max	Mean	Min	StdDev	Variance
96	0.103145	5.3766	95535.0	54882.9	0.0	28744.13	8.26225

XI. Conclusions and Future Work

In this paper we applied new stage to SCTP congestion control mechanism. The new stage set between after Congestion Avoidance phase and before fast retransmission phase, exactly after Congestion Avoidance phase and before fast retransmission phase as shown in figure (7) to prevent unnecessary failovers due to temporary congestion and achieve more efficient communication performance in multi-homing environment**.** when we applied the new stage for certain file sent to host (B) we get minimum than the standard of to SCTP congestion control mechanism we offer (7.26556 - 5.41302 = 1.85254 second) also the congestion window became more stable through the data transmission, therefore the better variance of congestion window done through The new stage also the best Averages achieved through The new stage**.** The minimum and maximum RTO during the standard of to SCTP congestion control mechanism are 0.78 and 2.78 respectively but when we applied the new stage. The minimum and maximum RTO decrease to 0.62 and 2.16 respectively, so the minimum delay in RTO achieved through applying the new stage of SCTP congestion control mechanism. and also the variance and averages of RTO through applied the new stage are seem when compared with the standard SCTP congestion control mechanism. The minimum and maximum RTT during the standard of to SCTP congestion control mechanism are 0.97and 0.025 respectively but when we applied the new stage. The minimum and maximum RTT decrease to 0.18 and 0.025 respectively, so the minimum delay in RTO achieved through applying the new stage for SCTP congestion control mechanism. The end to end delay between host A and host B has max value of delay 7.17 through standard SCTP congestion control, but the max value of delay through applied the new stage is 5.20 and the variance and averages of end to end delay between host A and host B are 3.46 and 3.82 respectively during standard SCTP congestion control but when we applied the new stage the variance and averages decrease to 1.96 and 3.18 respectively, so the performance of host B for receiving by increase during applied the new stage, the total delay is 5.25 second when we copaired with 7.23 second in standard SCTP congestion control**.** The performance clearly enhanced when host B advertised to receive max window 95535.0 compared with 61535.0 in standard SCTP congestion control, also other enhancement the variance and averages are 54882.9 and 8.26 respectively compared with 36987.6 and 3.266759 in standard SCTP congestion control**.** However, there are still some aspects that require further research in order to analyze the efficiency and the ability of this multi-homing feature. One of these the shortcoming related to current RTO Calculation, Considering that RTO selection has significant impact on switchover may cause degradation in network quality as path failure. Therefore, we propose a further work is to design an algorithm to Calculate RTO depend on variable values parameters instead of constant values parameters beside the current enhancement.

References

1. Stewart R, Xie Q *et al*. RFC 2960: Stream Control Transmission Protocol, The Internet Society, 2000.
2. Arsh Arora, Lekha Bhambhu. Evaluation of Active Queue Management Algorithms (IJCST). 2014; 2:4.
3. Behrouz Forouzan A. Data Communications and Networking (McGraw-Hill, Fourth Edition).
4. Łukasz Budzisz. Stream Control Transmission Protocol (SCTP) a proposal for seamless handover managementat the transport layer in heterogeneous wireless networks, Barcelona, 2009.
5. Victor Leung CM, Eduardo Parente Ribeiro, Alan Wagner, Janardhan Iyengar. Multihomed Communication with SCTP, CRC Press International Standard Book Number-13: 978-1-4665-6699-6, 2013.
6. Rahim Rahmani, Christer Ahlund, Theo Kanter. Design of Active Queue Management for Robust Control on Access Router for Heterogeneous Networks, EURASIP Journal onWireless Communications and Networking, 2011, 7.
7. Syed Yasmeen Shahdad, Gulshan Amin, Pushpender Sarao. multihoming and multistream protocol incomputer networks, IJCSMC, 2014; 3(3):1183-1187.
8. Fei Song, Hongke Zhang, Sidong Zhang, Fernando Ramos, Jon Crowcroft. An estimator of forward and backward delay for multipath transport, UCAM-CL-TR-747 ISSN 1476-2986, 2009.

9. ITU T. Recommendation Q.700: Introduction to CCITT Signalling System No. 7, International Telecommunication Union, Geneva, 1993.
10. Mériem Afif, Philippe Martins, Sami Tabbane, Philippe Godlewski. A SCTP – Layer 2 Cross Layer Mechanism for Data Handover in Wireless Networks (application to EGPRS), Research Gate, 2016.
11. Ahmed Abd El Al, Tarek Saadawi, Myung Lee LS-SCTP. a bandwidth aggregation techniquefor stream control transmission protocol, Computer Communications Elsevier, 2004.
12. Rammohan Bandaru, Debashis Barman. Performance Evaluation of SCTP as a Transport Layer Protocol, Technical report, IDE1117, 2011.
13. Yuansong Qiao, Enda Fallon, John Murphy, Liam Murphy, Xiaosong Zhu, Gregory Hayes *et al*. Performance Analysis of Multi-homed Transport Protocols with Network Failure Tolerance, IET Communications, 2016.
14. Afsane Zahmatkesh, Amir Hossein Mohajerzadeh. An Optimized Congestion Control and Path Management Schemefor SCTP, International Journal of Future Generation Communication and Networking. 2011; 4(4).
15. Katalin Tarnay, Gusztáv Adamis, Tibor Dulai. Advanced communication protocol technologies: solutions, methods, IGI Global, ISBN 978-1-60960-733-3 (ebook), 2011.
16. Meifang Shen, Yuanlong Cao, Mingchuan Zhang, Kaixuan Wang, Mingming Yu. Evaluation of Performance of Background Traffic-based CMT-SCTP with Active Queue Management Algorithms, IJCSI International Journal of Computer Science. 2012; 9(6):2.
17. William Stallings. Data and computer communications (Upper Saddle River, NJ 07458: Pearson Prentice Hall, Eighth Edition).
18. Stewart R, Tuexen M, Lei P. Stream Control Transmission Protocol (SCTP) Stream Reconfiguration, IETF Internet Engineering Task Force, 2012.
19. Syed Yasmeen Shahdad, Gulshan Amin, Pushpender Sarao. multihoming and multistream protocol in computer networks, IJCSMC, 2014; 3(3):1183-1187.
20. Jianping Zou MU, mit Uyar, Mariusz Fecko A, Sunil Samtani. Preferential Treatment of SCTP Subflows: Analysis and Simulation, Proc. IEEE Int. Symp. Comput.Commun. (ISCC), Alexandria, Egypt, 2004.
21. 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.
22. Tine Stegel, Janez Sterle, Janez Bešter, Andrej Kos. SCTP association between multi-homed endpoints over NAT using NSLP, Elektrotehniškivestnik. 2008; 75(5):277-284.
23. 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.
24. 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.
25. OMNeT++ User Manual, Version 4.5 http://www.omnetpp.org/doc/omnetpp/Manual.pdf.
26. OMNET++ homepage, http://www.omnetpp.org/.
27. Thomas Dreibholz, Martin Becke, Hakim Adhari, Erwin Rathgeb P, Irene Rungeler, Robin Seggelmann *et al*. Improvements to the sctp environment in the inet framework. University of Duisburg-Essen, Institute for Experimental Mathematics, OMNeT++ Code Contribution, 2012.