An Optimized Congestion Control and Path Management Scheme for SCTP

Afsane Zahmatkesh and Amir Hossein Mohajerzadeh

Dept. of Computer Engineering, Ferdowsi University Mashhad, Iran
Afsane.Zahmatkesh@gmail.com, AH.Mohajerzadeh@stu-mail.um.ac.ir

Abstract

In this paper, a new SCTP congestion control mechanism is introduced in multi-homing mode. Congestion in each route can be avoided or can be controlled, based on an Active Queue Management (AQM) method. Also, routers compute probability of congestion for the sources on the paths and then notify them. Therefore, the sources can adjust their sending rates on each path effectively and if necessary, can switch to an alternate path to prevent congestion. Simulations have been conducted with Opnet linked with MATLAB. The simulation shows that the new method can decrease packet loss, increase the amount of transmissions and stabilize queue length, as compared with standard SCTP.

Keywords: *SCTP; AQM; congestion control; sending rate*

1. Introduction

Stream Control Transmission Protocol (SCTP) is a transport protocol that has been proposed by the IETF Signaling Transport (SIGTRAN) working group [1]. SCTP inherited much of its design from TCP but improves several features to make its signal transmission more efficient. For example, SCTP is defined as an alternative transport protocol for the Session Initiation Protocol (SIP). SIP is a internet telephony signaling protocol [2]. Although it was initially developed for telephone signaling, it is gradually expanded into a general-purpose transmission layer. Nowadays, SCTP is a mature protocol standardized in RFC 4960 [3].

Multi-homing is one of the features that SCTP natively supports. This feature makes it possible to obtain a high reliability and robustness against single interface and network failures. An SCTP endpoint is considered multi-homed if there are more than one transport address that can be used as a destination address to reach that endpoint.

SCTP uses an end-to-end window based flow and congestion control mechanism similar to TCP [4]. SCTP can support multi-homing and has respective congestion control for each multiple transport paths.

Congestion occurs when the amount of data injected by sources in the network are larger than the amount of data delivered to destinations.

Similar to TCP algorithms, SCTP uses only implicit congestion information such delays or losses. Congestion control can be implemented as a distributed control strategy. Some mechanisms like Active Queue Management (AQM), executed by routers, detect congestion problems and inform sources (either implicitly or explicitly with the mechanism of Explicit Congestion Notification ECN [5]). These techniques are designed to reduce packet loss and the end-to-end delay as well as to improve network utility. An AQM algorithm regulates the queue length by drops/marks incoming packet with a given probability related to a congestion index (such as queue length or delay).

In this paper, a novel rate adjustment method is proposed to improve SCTP congestion control. Here, characteristic of SCTP multi-homing is considered. In this case, two paths are chosen as primary path and secondary path between each source and each destination. In this algorithm, based on an Active Queue Management (AQM) method, congestion on each route can be avoided or can be controlled by probability of dropping. Also, routers compute probability of congestion, which is named congestion degree, for each source on each path and feedback to the sources. Then, the sources can adjust their sending rates effectively on each route by receiving the feedback from paths. After that, the senders consider condition of path switch to prevent congestion.

The remainder of the letter is organized as follows: main features of SCTP are briefly introduced in Section 2. The related works are discussed in Section 3. In section 4, the proposed algorithm is presented. Simulation results are brought in Section 5 and the paper is concluded in Section 6.

2. Overview of SCTP

As mentioned before, SCTP is very similar to TCP. It is connection oriented and a connection in SCTP called an association. Data are transmitted in chunks which are a unit of user data or control information within a SCTP packet consisting of a specific chunk header and specific contents dependent of its usage.

Multi-homing is a feature of SCTP. Based on this feature, an association may comprise multiple source and destination IP addresses. During association setup, one path is selected as the primary path, and provided that this path is available, all data are sent in this way. Any remaining paths serve as backup or alternate paths. On these paths, only Heartbeat packets are sent regularly to control reach-ability. Often, one of the alternate paths is defined as a secondary path and retransmitted data are passed on this route. The secondary path avoids additional and unnecessary congestion at the primary path.

In SCTP, the sender keeps an error counter for the primary path which counts the number of timeouts that occurs consecutively. If the error counter of the primary path reaches a set threshold, Path.Max.Retrans (PMR), the primary path is considered unavailable or unreachable and a failover is performed.

In case of a failure, SCTP should quickly switch the transfer to an alternate path, but in case of mild congestion, it continues to use the same path. If the secondary path is used when the primary path failure is detected, the primary path is unusable and the secondary path is used as the primary path. Also, the next secondary path is selected from the alternative paths. Figure1 shows the SCTP multi-homing concept.

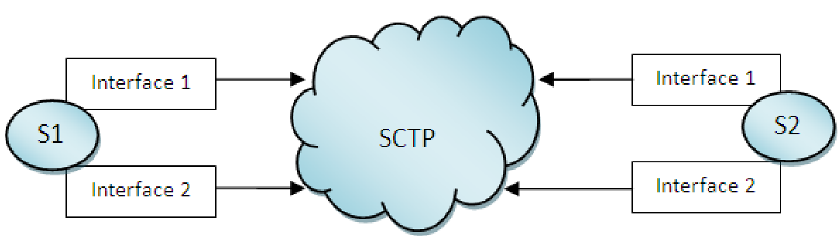

Figure 1. SCTP Multi-homing

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. Additionally, the use of selective acknowledgements (SACK), similar to TCP SACK [6], is mandatory in SCTP.

In multi-homing mode, SCTP has a separate set of congestion control parameters for each of multiple transport paths within an association.

3. Related Works

Results of preliminary and in-depth studies on various features of SCTP have been reported in the literature. The most studies have been focused on the performance of SCTP in different situations [7-9]. Some efforts in SCTP congestion control with multi-homing features are as follows.

Dahal and Saikia [10] have represented an adaptation of an RTT based Congestion Control scheme on SCTP. This eliminates drops due to congestion, decreases packet latency and necessary packet retransmissions. In this method, based on RTT measurements, level of traffic load have been computed to avoid driving the network into congestion. Also, a scheme has been considered that SCTP can switch to the alternate path to prevent congestion.

A protocol named WiSE has been proposed by Fracchia and Chiasserini [11]. This is a sender-side transport-layer protocol that modifies the standard SCTP protocol through the use of bandwidth estimation techniques. WiSE tries to infer whether losses are due to congestion or radio channel errors. The protocol computes available bandwidth for current path and an alternate path. If the current path is severely congested and the alternate path is lightly loaded, WiSE switches the transmission onto the alternate path using SCTP's flexible path management capabilities.

Ho and Cheng [12] have proposed a new enhancement SCTP called RSCTP (Receiver Bandwidth Estimation SCTP) based on receiver-side available bandwidth estimation. RSCTP discriminates wireless loss from congestion loss over error prone wireless link by bandwidth estimation. This changes the principle of Heartbeat-Request of SCTP to send Heartbeat-Request on primary path periodically. Receiver then utilizes the interval of Heartbeat-Request to calculate available bandwidth. Also, when loss rate exceeds the maximum path loss rate, primary path switches immediately to keep data transmission.

A new path management (quality-aware SCTP) has been represented by Chen et al. [13] for wireless networks. This includes a new path failure detection method and ICE (idle path congestion window size estimation) mechanism. The new method uses cycle counting rather than single counting as in standard SCTP to detect path failures. Cycle counting improves the original path failure detection method in a wireless environment. Also, the ICE mechanism can estimate the path quality and provides information for path switching decisions.

Chen et al. [14] have been proposed a jitter-based congestion control scheme with end-to-end semantics over wired-wireless networks. The new protocol, JSCTP adopts jitter ratio to differentiate wireless loss from congestion loss. Because of multi-homing, different paths should maintain their parameters of jitter ratio when transmitting through the path. Available bandwidth estimation scheme will be integrated into their congestion control mechanism to make the bottleneck more stabilized.

This study is different from the previous studies in considering AQM method for SCTP. Also, routers inform the sender probability of congestion for paths to adjust its sending rates on the paths. Here, the sending rate on each path depends on its packet loss rate and its probability of congestion. Because these conditions affect the size of congestion window that shows the amount of data can be transmitted. So, by selecting appropriate load on each of the paths, throughput of the network can be increased and probability of packet loss can be decreased.

4. The Proposed Algorithm

To understand the proposed algorithm, the network topology for one sender is shown in Figure 2. There are a sender (S), a receiver (R) and several paths between the sender and the receiver. Three routers are on each route. The first route is the primary path, the second is the secondary path, and the others are the backups. Data are transmitted between the primary and the secondary paths.

In this paper, the algorithm is implemented in the routers and the sources over wired networks. In what follows, the proposed algorithms for the routers and the senders will be described, respectively.

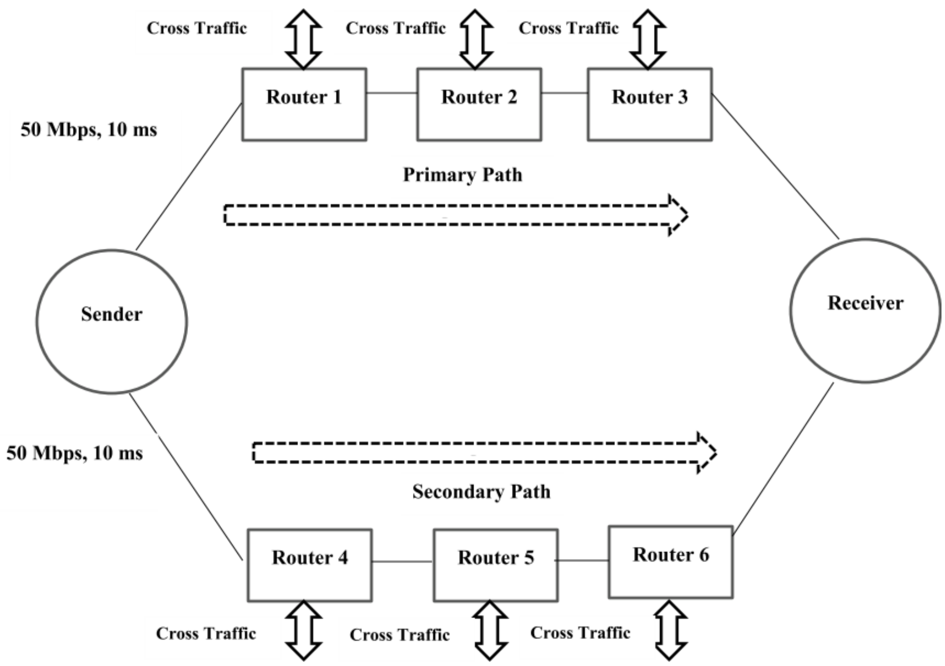

Figure 2. Network Topology

4.1. The Proposed Algorithm for Routers

In this study, an AQM method is implemented in the routers. AQM algorithm is a router-based congestion control mechanism which aims to signal congestion early by dropping packets before the buffer becomes completely full. This method has been proved to be an efficient way to alleviate or control network congestion and to maintain the queue sizes stability [15].

The RED (Random Early Detection) [16] algorithm, the earliest well-known AQM scheme, was developed to eliminate the flow-synchronization problem and attenuate the traffic load by monitoring the average queue length. Also, it goes to achieve fairness among sources with different transmission rates.

Similar to the existing AQM schemes, the proposed algorithm adjusts the packet drop probability based on the deviation of the queue length. Also, the routers compute congestion degree for the senders on each path as a congestion indicator. Then, the routers inform the sender to adopt its sending rates on both routes.

Each router has one queue with maximum size of QL. This queue is shared between the senders. Each sender has a proportional contribution in each queue based on its packet's application. These proportional contributions are considered as follows.

$$\sum_{i=1}^{k} Po[i] = 1, \qquad (1)$$

with k being the number of senders and Po[i] being the portion of sender[i] at queue of router[j]. Where, j being the router's number.

When a packet from sender[i] arrives in the queue[j], some phases should be done. At the first phase, q_rate[i] and QL_free[j] are computed as,

$$q\_rate\,[i] = (Pck\_num\_sndr\,[i]) \,/\, (Po\,[i] * Q_L), \qquad (2)$$

$$O_L\_free\,[j] = Q_L - \left(\sum_{i=1}^{k} Pck\_num\_sndr[i]\right), \qquad (3)$$

with q_rate[i] being the packet rate of sender[i] in the queue of router[j]. Pck_num_sndr[i] is the number of packets from sender[i] and Po[i] is the specified portion of sender[i] in the queue in router[j]. QL is maximum size of this queue and QL_free is the free space of the queue.

For the second phase, two major thresholds are concerned, namely, QL(MAX) and QL(MIN). Here, the maximum and minimum corresponds to the average queue length. The simulation set the best values to 30 and 45, respectively.

In this phase, by receiving a packet in the queue, the current queue size is compared with the two thresholds and based on condition packet is inserted or is dropped. If the length of current queue size be lower than the minimum threshold, packet will be inserted and no packet is dropped. If the length be between the maximum and the minimum threshold and Pck_num_sndr[i] be less than Po[i]*QL, packet is inserted. Otherwise, the packet is discarded.

The last condition occurs when the current queue size is greater than the maximum threshold. In this case, if Pck_num_sndr[i] be less than Po[i]*QL, a packet is inserted with a probability P(n). Otherwise, the packet should be discarded.

$$\begin{aligned} P(n) &= (Current\_queue\_size - Q_L(min)) * \\ &(0.5 / (Q_L(MAX) - Q_L(MIN)) + 0.5), \end{aligned} \qquad (4)$$

The proposed AQM algorithm is described in Figure 3.

The proposed AQM algorithm

```
If (Current_queue_size < Q_L (MIN)) {
    /*Insert the new packet
    Pck_num_sndr[i] ++;
  }
If (Current_queue_size > Q_L (MIN)) {
  If (Current_queue_size < Q_L (MAX)) {
          If (Pck_num_sndr[i] < Po[i]*Q_L) {
               /*Insert the new packet
               Pck_num_sndr[i] ++;
          }
```

```
            Else
                /*Drop packet
        }
      }
    If (Current_queue_size > QL (MAX)) {
     If (Pck_num_sndr[i] < Po[i]*QL) {
            /*Insert the new packet with a probability of
            P (n)
            Pck_num_sndr[i] ++;
        }
     Else
          /*Drop packet
    }
```

Figure 3. The Proposed AQM Algorithm

After the decision for dropping or inserting packet, cd_evaluate(i) function is used in the last phase. Where, i being the sender's number.

The function evaluates congestion degree for the senders of arriving packet. This value is between 0 and 1. Based on statistical methods, q_rate[i], QL_free, and by employing some threshold, congestion degree for sender[i] at the queue is calculated in Figure 4.

cd_evaluate(i) function

```
If (q_rate [i] <= 1) {
            If (QL_free [j] < thα * QL) {
                congestion_degree = 0; }
Else {
          If (0 <= q_rate [i]) {
                        If (q_rate [i] < 0.1) {
                        congestion_degree = 0.07; }
            }
          If (0.1 <= q_rate [i]) {
                        If (q_rate [i] < 0.2) {
                        congestion_degree = 0.14; }
            }
          If (0.2 <= q_rate [i]) {
                      If (q_rate [i] < 0.3) {
                        congestion_degree = 0.21; }
            }
          If (0.3 <= q_rate [i]) {
                      If (q_rate [i] < 0.4) {
                       congestion_degree = 0.2; }
           }
          If (0.4 <= q_rate [i]) {
                     If (q_rate [i] < 0.5) {
                      congestion_degree = 0.35; }
           }
```

```
        If (0.5 <= q_rate [i]) {
                If (q_rate [i] < 0.6) {
                  congestion_degree = 0.42; }
        }
        If (0.6 <= q_rate [i]) {
                If (q_rate [i] < 0.7) {
                   congestion_degree = 0.49; }
        }
        If (0.7 <= q_rate [i]) {
                If (q_rate [i] < 0.8) {
                   congestion_degree = 0.56; }
        }
        If (0.8 <= q_rate [i]) {
                If (q_rate [i] < 0.9) {
                  congestion_degree = 0.63; }
        }
        If (0.9 <= q_rate [i]) {
                If (q_rate [i] < 1) {
                   congestion_degree = 0.7; }
        }
        If (q_rate [i] == 1) {
                congestion_degree = 0.77; }
        }
}
If (q_rate [i] > 1) {
        If (Q_L_free [j] < thβ * Q_L) {
            congestion_degree = 0.9;
        }
      Else {
          If (Q_L_free [j] < thλ * Q_L) {
            congestion_degree =1;
              }
          Else {
            congestion_degree = 0.84;
           }
      }
}
```

Figure 4. Congestion Degree Calculation Algorithm

Using simulations, value of the thresholds are adopted as thα =o.5, thβ = 0.5 and thλ = 0.7. Then, this value is put in the packet. One byte from reserve option in the packet is used for transmission the congestion degree value. If the packet does not contain an ACK, while arrives in the destination, the congestion degree in the packet will be put in ACK of the packet. Then, the ACK is transmitted to the sender of the packet. When the packet or ACK passes through the routers, it gets maximum value of the congestion degree for its source from the routers.

A given service rate is responsible to send packets at queues to the destination of the packet. When the sender receive the ACK packet, it is responsible for adjusting its sending rates based on receiving information and packet loss rate on each route.

4.2. The Proposed Algorithm for Senders

As seen in Figure 2, there are several routes between the endpoints. One route is the primary path while the other route is the secondary path and the other routes are alternate. Therefore, the data are sharing between the primary and the secondary path. SCTP has a separate set of congestion control parameters for each of transport paths

At the first, the sending rates are the same for both routes. The senders receive the ACK packets and reserve their congestion degree value. Here, Heartbeat-request is sent to the receiver, periodically. In this algorithm, the sender selects time of arrival of Heartbeat-request's ACK to evaluate costs of the routes. Interval of Heartbeat-requests is utilized. The receiver puts packet loss rate in the ACK of Heartbeat-request. The parameters to evaluate costs are packet loss rate and congestion degree on the paths. The probability of congestion is computed from the reserving congestion degrees based on pattern recognition method. To obtain the new sending rates using costs of each route, an optimization function is employed. For using optimization function, Opnet [17] is linked with MATLAB to send costs of each route as inputs and receive outputs to change congestion window size of the routes.

The function is expressed as follows:

$$\text{Minimize } F = ((1\text{-}Af_1/k_1) / (1+af_1/k_1)) * Cost_1 + ((1\text{-}Af_2/k_2) / (1+Af_2/k_2)) * Cost_2, \quad (5)$$

$$\text{S.T. } Af_1+Af_2 = 1,$$

Here, Cost1 and Cost2 are the costs of paths as inputs for the function. k1 and k2 denote coefficients and have a constant value of 0.5. Af1 and Af2 are the outputs and depend on desired sending rates on the first and the second paths, respectively. The sender determines the sending rate depending on the values of congestion window and receiver’s received window. It adjusts the congestion window size of each route by receiving the new Af1 and Af2 based on Figure 5.

Rate adjustement
If Af_New[i] < Af_Old[i] *Cwnd = Af_New[i] * Max_size_Cwnd // Immediately change* *If Af_New[i] > Af_Old[i] // Gradually change* *Cwnd(t) =((t – t1) (Af_New(i , t1)* Max_size_Cwnd –Cwnd(t1)))/15 + Cwnd(t1)*

Figure 5. Rate Adjustment Algorithm

where Af_New[i] is the new value of Af for path[i] and Af_Old[i] is the previous calculated Af for path[i]. Max_size_Cwnd is the maximum size of Cwnd and it is determined based on bandwidth of links and the length of packets. Cwnd(t) shows the value of Cwnd at time t and t1 is the time of calculating the previous Af.

If the previous value of Afi is greater than the new value, the congestion window size is decreased immediately. Otherwise, the sender increases the size gradually by receiving every ACK.

After adjusting the new sending rates, the sender considers conditions of path switch. The conditions is based on the error counter in the standard SCTP and also, using Afi. If the error counter on the primary path exceeds the threshold value, then, Af1 and Af2 are compared. If Af1 is lower than Af2, the primary path switches to the secondary path. Otherwise, switch is not necessary. Based on this condition, in addition to packet loss, the congestion conditions of the routes are considered and switch is done effectively.

The Block diagram of the proposed algorithm is shown in Figure 6.

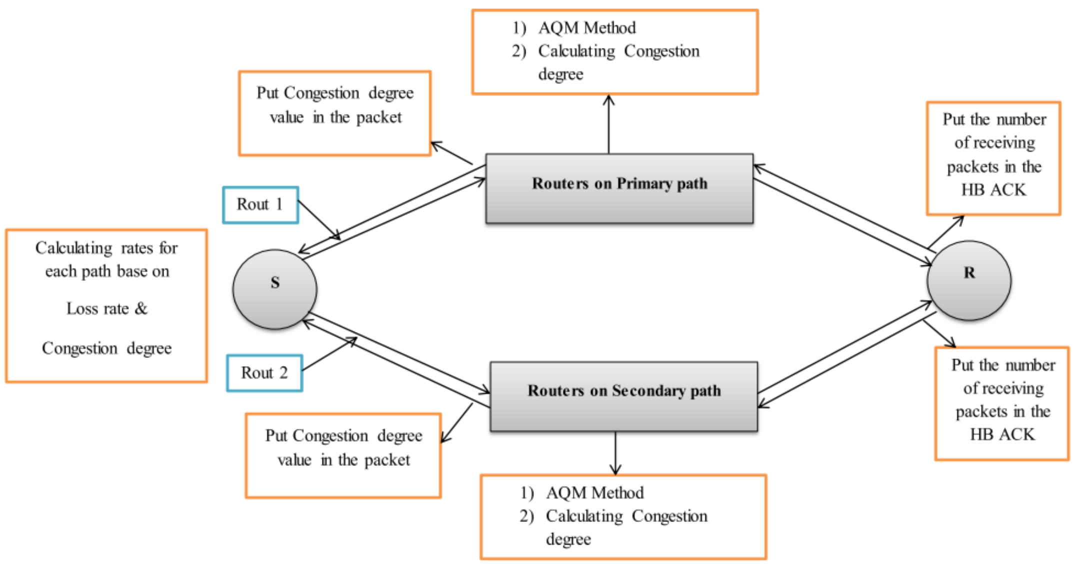

Figure 6. Block Diagram of the Proposed Algorithm

5. Simulation Results

To evaluate the performance of the proposed algorithm, Opnet modeler is linked with MATLAB. The network topology used in the simulation is the same as in Figure 2 for each sender. The endpoints are SCTP agents. The capacity of links set 50Mbps and 20Mbps and the link delay is set 10ms. Three routers are concerned on each route. The maximum buffer size of each router is taken 50 packets. The current simulation focuses on three performance metrics: Throughput, Cumulative packet loss and Queue length.

Both TCP and SCTP control congestion by changing the congestion window size to control the quantity of packets being transmitted. In addition, the packet transmission condition affects the size of the congestion window. For example, by occurring congestion or by increasing packet loss rate, the size of the congestion window is decreased. So, selecting the appropriate load on each of path in terms of packet loss rate and congestion occurrence, affects amount of data can be transmitted. The volume of data transmitted to the receiver per unit time is named throughput. Hence, with higher throughput, more data can be transmitted. Figure 7 compares the throughput of the new SCTP and the standard SCTP.

As shown in Figure 7, with increasing transmission rates, throughput is increased in both protocols. Also, in the same transmission rates, the protocols have almost the same throughput values. However, it is clear that throughput degrades as the transmission rate exceeds 4Mbps. This occurs since congestion or drop occurs in the network happened. Notice that the new SCTP scheme shows a higher throughput than standard SCTP after the network becomes congested. Because with a proper load on the routes after congestion, the new protocol can increase the transmission rates.

Figure 8 depicts the cumulative packet loss for the new SCTP as well as the standard SCTP. The cumulative packet loss at any time of the simulation is concerned as the sum of the packet loss since the beginning of the simulation until that time.

It is evident that at different times, the new SCTP has a lower cumulative packet loss. This shows that the probability of the occurrence of the packet loss in the proposed algorithm is less than the SCTP. This occurs since the proposed algorithm implements proper AQM method and appropriate rate adjustment to degrade the congestion as well as the packet loss.

In Figure 9, the experiment shows the instantaneous queue size for both of the protocols. The simulation is performed for 200 seconds. The results demonstrate that SCTP is unable to control the oscillations in the instantaneous queue. In spite of that it is clear that the proposed AQM provides reasonable stability to the instantaneous queue. The proposed algorithm can stabilize the queue length around a desired level and achieves the lower standard deviation and based on Figure 8, the new scheme has lower value of cumulative packets loss.

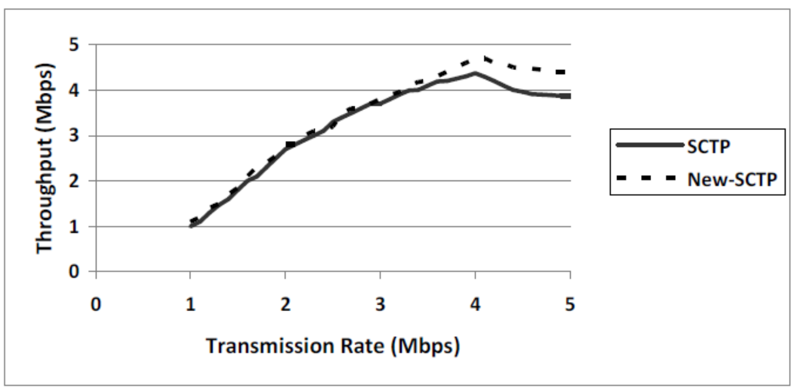

Figure 7. Throughput Comparisons

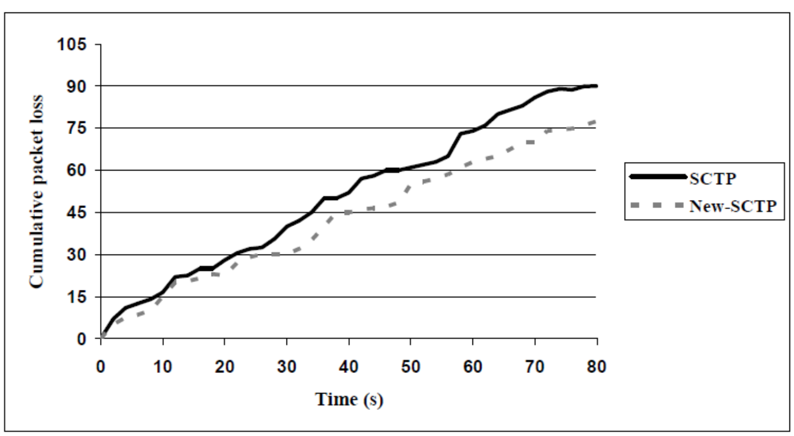

Figure 8. Cumulative Packet Loss

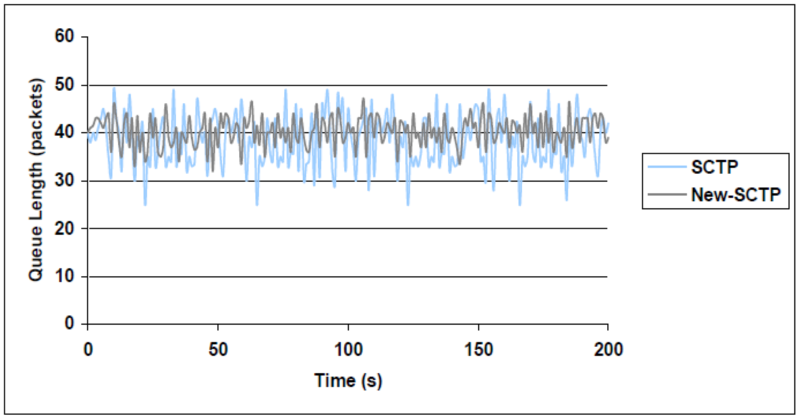

Figure 9. Instantaneous Queue at New-SCTP and Standard SCTP

6. Conclusion

In this paper, a new rate adjustment method for SCTP congestion control in multi-homing mode is presented. The data are sharing between two paths as the primary and the secondary path, the others are alternate.

The algorithm is implemented on the sender and the routers. The routers calculate the probability of congestion for each sender and inform them. Also, based on an AQM method, incoming packets are dropped / marked with a given probability to prevent congestion.

The sender defines the amount of data transmitted on each path depending on its packet loss rate and its probability of congestion on each routes.

The simulation is performed in Opnet modeler linked with MATLAB. The experiments show that the new algorithm increases the amount of data that can be transmitted so, this achieves a better throughput. In addition, the proposed algorithm has a lower cumulative packet loss than the standard SCTP. The new SCTP can also stabilize the instantaneous queue around a desired value with reduced queue length oscillation while the standard !!! SCTP is unable to control the oscillations in the instantaneous queue.

With more accurate calculation of the congestion probability and with more effective optimization function, the new method can demonstrate even better performance and this can be an area of future research.

Acknowledgment

The authors wish to express their gratitude to Research Institute of ICT - ITRC for their financial support.

References

[1] R. Stewart, Q. Xie, et al., “Stream Control Transmission Protocol,” RFC 2960, October 2000.

[2] J. Rosenberg, H. Schulzrinne, G. Camarillo, “The Stream Control Transmission Protocol (SCTP) as a Transport for the Session Initiation Protocol (SIP),” RFC 4168, October 2005.

[3] R. Stewart, “Stream Control Transmission Protocol,” RFC 4960, September 2007.

[4] M. Allman, V. Paxon, and W. Stevens, “TCP congestion control,” RFC 2581, April 1999.

[5] K. Ramakrishnan, S. Floyd, D. Black, “The Addition of Explicit Congestion Notification (ECN) to IP,” RFC 3168, September 2001.

[6] M. Mathis, J. Mahdavi, S. Floyd, A. Romanov, “TCP Selective Acknowledgment Options,” RFC 2018, October 1986.

[7] M. Kozlovszky, T. Berceli, L. Kutor, “Analysis of SCTP and TCP based communication in high-speed clusters,”ACAT 05, Vol. 559, Apr. 2006, pp. 85-89, doi:10.1016/j.nima.2005.11.107 .

[8] M. Scharf , S. Kiesel, “Head-of-line Blocking in TCP and SCTP: Analysis and Measurements,” Proc. Global Telecommunications Conference (GLOBECOM '06), IEEE Press, Nov. 2006, pp. 1 – 5, doi:10.1109/GLOCOM.2006.333.

[9] T. Hung, T. Khanh, “Analyze and Evaluate the performance of SCTP at transport layer,” Proc.International Conference on Advanced Communication Technology (ICACT 2010), IEEE Press, Feb. 2010, pp. 294 - 299.

[10] M. Dahal, D. Saikia, “RTT Based Congestion Control and Path Switching Scheme for SCTP,” Proc. International Conference onCommunication Technology (ICCT 2006), IEEE Press, Nov. 2006, pp. 1-4,doi:10.1109/ICCT.2006.341983.

[11] R. Fracchia, C. Chiasserini, “WiSE: Best-Path Selection in Wireless Multihoming Environments,” IEEE TRANSACTIONS ON MOBILE COMPUTING, Vol. 6, Oct. 2007, pp. 1130-1141, doi:10.1109/TMC.2007.1027 .

[12] Y. Ho, Q. Cheng, “An Adaptive SCTP Congestion Control Scheme Based on Receiver Available Bandwidth Estimation,” Institute of Computer and Communication Engineering,National Cheng Kung University, Dec. 2007.

[13] J. Chen, C. Chu, E. Wu,P. Lin and K. Lu,“Quality-Aware SCTP inWireless Networks,” EURASIP Journal on Wireless Communications and Networking, vol. 2010, Feb. 2010, doi:10.1155/2010/820578.

[14] J. Pan, M. Chen, and et al.“Improving SCTP Performance by Jitter-Based Congestion Control over Wired-Wireless Networks,” EURASIP Journal on Wireless Communications and Networking, vol. 2011, Jan. 2011, doi:10.1155/2011/103027 .

[15] B. Braden et al., “Recommendations on Queue Management and Congestion Avoidance in the Internet,” RFC 2309, Apr. 1998.

[16] S. Floyd, V. Jacobson, “Random early detection gateways for congestion avoidance,” IEEE/ACM Transactions on Networking (TON), Vol. 1, Aug 1993, doi:10.1109/90.251892.

[17] “Opnet Modeler.” www.opnet.com.