An Adaptive SCTP Congestion Control Scheme Based on Receiver Available Bandwidth Estimation

Yu-Kun Ho
Institute of Computer and Communication Engineering
National Cheng Kung University, Taiwan, R.O.C.
ykho@mail.ncku.edu.tw

Qiao-Hung Cheng
Institute of Computer and Communication Engineering
National Cheng Kung University, Taiwan, R.O.C.
u5330509@tknet.tku.edu.tw

***Abstract* —Streaming control transmission protocol (SCTP) is one kind of new generation transport layer protocol, is a little modification form TCP. Like TCP, the congestion control mechanism of current SCTP relies on packet loss as an indicator of network congestion. When congestion been detected the sender will reduce congestion window size to half to avoid congestion. In wire/wireless network environment, the packet losses are not all ways because of congestion. If SCTP reduces the congestion window inappropriately in an error prone wireless network, it may not use the network resource efficiently. For the purpose to improve the performance of SCTP congestion control, especially in an error prone wireless network, we propose a new enhancement SCTP called RSCTP (Receiver Bandwidth Estimation SCTP) based on receiver-side available bandwidth estimation. RSCTP relies on bandwidth estimation to discriminate wireless loss from congestion loss over error prone wireless link, it also reduces congestion window and slow start threshold appropriately. In multihoming mode, using chunk loss rate estimation, RSCTP can switch path actively over the high loss rate environment. We use NS2 to implement RSCTP and the simulation results reveal that our scheme improves performance efficiently over error prone wireless network in evidence.**

Keywords: SCTP, Congestion Control, Wireless Error, Bandwidth Estimation, NS2

I. INTRODUCTION

With the evolution of communication networks grow up, wired/wireless heterogeneous networks gradually become the most popular communication networks in next generation Internet. Mobile devices may be equipped with multiple wired or wireless network interfaces, such as Ethernet, 802.11, 3G etc., to adapt different environment [1].

The common transport layer protocol, such as TCP and UDP, can bind only one IP address. It may be a waste of other network paths that could provide alternative to parallel transmission. Thus, more and more researchers aim at this solution of multihoming issue. For the variable applications of transport layer protocol, Internet Engineering Task Force (IETF) has proposed one can cope with multihoming feature, called Streaming Control Transmission Protocol (SCTP) [18].

SCTP is originally designed to carry telephony signaling over IP networks. It also provides reliable, unreliable and error-free services like TCP and UDP. SCTP overcomes several security deficiencies of TCP by using four-way handshake and cookie mechanism. 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 [19]. SCTP also provides partial reliability service that can deliver various data of applications in one association [23]. The multiple features of SCTP make SCTP be a good alternative of next generation transport layer protocol.

For the terminal users access Internet over wired/wireless heterogeneous network, transport layer protocol should carry end-to-end semantics and it’s can’t receive any message below itself. Packet losses were interpreted as symptoms of congestion by TCP. In the wired portion of the network a congested bottleneck network is indeed the likely reason of packet loss; on a wireless link, on the other hand, a noisy, fading ratio channel is the more likely cause of loss. Hence the poor performance of wireless TCP gets more and more attention [15].

SCTP has more excellent features than TCP [20], but it inherits TCP’s congestion control scheme [17][21]. TCP-like congestion control scheme cannot work well because of its inability to differentiate wireless loss from congestion loss [2]. As figure 1-1, if all wireless losses are treaded as congestion losses, congestion window will be reduced to half unnecessarily. SCTP cannot utilize the network resource efficiently.

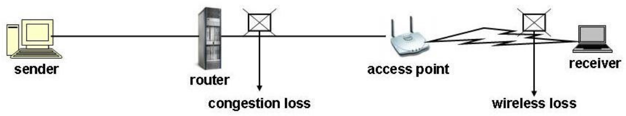

Fig. 1 congestion loss and wireless loss

There are many effective error and congestion control for wired/wireless networks have been proposed. They can roughly be classed into three kinds [8]: End-to-End, Link Layer, and Split Connection approaches. End-to-End is it reacts to send ack back until final destination receive packet. Such as TCP Veno [10] belongs to this kind. It adds loss event distinction in TCP Vegas [9] to prevent unnecessary congestion window reducing over wireless network. Link Layer has Snooping [8] and MEW SCTP [26], etc. Snooping caches copies of TCP data packets for local retransmissions to reduce end-to-end retransmissions. MEW SCTP adds a control chunk on link layer to notify the cause of packet losses. It helps SCTP to prevent the reducing in congestion windows size.

Although end-to-end schemes are not as effective as other two schemes in handling wireless losses, it can be achieved without intermediate node support at the network layer in routers and base stations. It maintains the principle of end-to-end feature. This thesis proposes a SCTP congestion control scheme variation using receiver bandwidth estimation. Utilizing the feedback information of receiver available bandwidth, sender can judge wireless loss form congestion loss and the performance over error prone wireless network will be improved. We use binomial control to decrease congestion window while wireless losses and prevent unnecessarily congestion window decreasing [12]. In the multihoming mode, on the other hand, SCTP standard switches paths passively. The scheme we proposed considered chunk loss rate. We switch path actively if chunk loss rate exceeds one threshold on primary path and data transmission will be more stable.

The rest of this thesis is organized as follows. SCTP overview and several transport layer protocol enhancement will be introduced in section 2. In section 3, we propose a receiver-based bandwidth estimation congestion control scheme of SCTP that improves the performance over wireless error prone link. We use NS2 to simulate our scheme versus original SCTP and describe the performance in section 4. The final section is conclusion.

II. BACKGROUND AND RELATED WORK

2.1 SCTP overview

The development of the Internet network has already been exceeded for more than 20 years. Most applications use TCP or UDP as transport layer protocol, but what TCP and UDP offered which have not accorded with some demands of service gradually. SCTP is a kind of developing transport layer protocol, it offers the functions between TCP and UDP and includes the advantages of the two. SCTP is in the same layer with TCP and UDP in OSI, and the IETF has already formed it in RFC2960 [17].

SCTP has inherited most functions of TCP, such as reliable transmission, inorder transmission, error recovery and congestion control. SCTP needs to set up an association before data transmission. Then the data reaches receiver by its sending sequence and does error recovery while loss happening. SCTP will close the association after finishing data transmission.

SCTP was originally designed for the transport of message based signaling information over IP networks. SCTP also features some functions more advantaged than TCP, such as partial reliable, unordered transmission, multihoming and multistreaming, etc. SCTP gets more stronger security by these features. Hence SCTP can offer larger range application service, especially in the multimedia transmission. SCTP can totally substitute the function of TCP, and replace partial function of UDP too, so it may replace TCP and become next generation transport layer protocol [18].

2.1.1 SCTP congestion control

SCTP congestion control is based on TCP, including Slow-Start, Congestion Avoidance, and Fast Retransmission are the same as TCP. TCP-like congestion control scheme also been generalized as additive-increase/multiplicative-decrease (AIMD) [12][14]. If sending rate is too fast between two communication hosts, it is easy result in congestion situation. The routers will start to discard packets to avoid congestion. As the sender detects to packet loss, it infers that congestion happens in the network. SCTP sender will start a succession of congestion controls at this moment. And then reduces sending rate or stops sending packet. Every congestion control introduces as follows.

(1) Slow Start

Beginning data transmission into a network with unknown conditions or after a sufficiently long idle period requires SCTP to probe the network to determine the available capacity. There are three situations to trigger slow start, including (a) while beginning a new connection, (b) while resuming a transmission after a long idle time, (c) after repairing loss detected by the retransmission timer.

When slow start begins, initial congestion window is set as MTU of twice. During slow start, whenever receives one SACK the congestion window is increasing by the amount of data that SACK acknowledged. When congestion window exceeds or equals slow start threshold, or detects chunk loss, finish slow start and enters the procedure of congestion avoidance.

(2) Congestion Avoidance

When congestion window exceeds or equals slow start threshold, the state enters congestion avoidance. The congestion window is increased by one MTU every RTT (Round-Trip-Time) until congestion occurs.

(3) Fast Retransmission

As chunk loss happen, sender receives 4-duplicated SACKs and triggers fast retransmission immediately, then sender does not wait for retransmission timeout to send lost chunk back. Besides, slow start threshold will be set as the maximum of 1/2 of congestion window and double MTU. And then set up congestion window as slow start threshold. Fig.2-1 shows the whole SCTP congestion control scheme.

```
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 2-1: SCTP Congestion Control

2.2 Binomial Congestion Control [12]

In [12] the authors proposed a class of non-linear TCP compatible congestion control schemes called Binomial Congestion Control Schemes, which are well suited for real time streaming applications. AIMD can be considered as one of congestion control schemes in the subset of binomial algorithm. Formally, every TCP-like congestion control equation can be generalized by the binomial algorithm as the following two equations:

$$w' = w + \alpha / w^k \; ; \; \alpha > 0 \text{ and if no loss} \quad (1)$$
$$w' = w - \beta w^l \; ; \; 0 < \beta < 1 \text{ and if loss} \quad (2)$$

Where w' is the congestion window after adjusting, k and l are window scaling factors for increasing and decreasing respectively, and α and β are proportionality constants. For any given value of α and β and $k + l = 1$ and $l \leq 1$, this class of congestion control will be TCP-Friendly. Furthermore, all the binomial control protocols converge to fairness as long as $k \geq 0$, $l \geq 0$ and $k + l > 0$.

2.3 Wireless enhancement on transport layer protocol

TCP is still the most common transport layer protocol currently. TCP is also the main target to that transport layer protocol enhancement on wireless network. Many methods such as End-to-End, Link Layer and Split Connection, etc. are proposed successively. End-to-end is not as efficient as other two methods with regard to packet loss on wireless network, but end-to-end is totally set up in transport layer. The one does not need any assistance below transport layer. Hence the thesis regards SCTP as the target of improving and keeps the characteristic of end-to-end. There are several kinds of improvement methods as follows.

Snooping [8] is a kind of Linker Layer method. It caches copies of TCP data packets for local retransmissions to avoid premature retransmissions at TCP sender. Snooping can reduce end-to-end retransmissions and prevent the reducing in congestion windows size, but it requires a proxy in the base station and some modifications of TCP.

TCP Vegas [9] attempts to solve network congestion by limiting window based on measurements of round trip delay but it suffers fairness problems and unable to distinct loss events. TCP Veno[10] derived from TCP Vegas. It attempts to solve the wireless loss problem by distinguishing wireless loss from congestion-related loss. But it still has different round trip delay measurement on different connection.

TCP Westwood [15] is a sender-side modification of TCP in wire as well as wireless network. It can estimate the end-to-end bandwidth and the improvement is most significant in wireless network with lossy links. The sender monitors the ACK reception and from it estimates the data rate. The sample of bandwidth B_k is

$$B_k = d_k / (t_k - t_{k-1}) = d_k / \triangle t_k \quad (3)$$

Where t_k is the current SACK reception time and t_{k-1} is the previous SACK reception time respectively, and d_k is the amount of data size acknowledged by the receiver at time t_k. The sender also uses the bandwidth estimate to properly set the congestion window and slow start threshold.

In reference [27] the authors proposed one kind of cross-layer method. It proposed that MAC layer retransmission is always a little earlier than Transport layer. If MAC layer packet loss occurs, a new packet MEW will be generated at MAC layer before the chunk is discarded, which will notify the upper layer the failed transmission. MEW SCTP can distinct wireless loss from congestion loss by MEW and trigger retransmission without degrading the congestion window. MEW SCTP needs assistance of intermediary. It will not maintain end-to-end principle.

In order to improve SCTP congestion control on error prone wireless link, this thesis proposes a loss event differentiation method using receiver bandwidth estimation based on SCTP. The bandwidth estimation refers to TCP Westwood, but different from TCP Westwood we put bandwidth estimation on receiver. And congestion window adjusting while packet losses cause of wireless error adopts binomial control algorithm. It prevents congestion window from decreasing unnecessarily over error prone wireless link.

III. RECEIVER BANDWIDTH ESTIMATION SCTP

In order to solve the problem that congestion control of SCTP exhibits poor efficiency over wireless error prone link, this thesis proposes one adaptive congestion control, called RSCTP (Receiver Bandwidth Estimation SCTP). To stabilizing the demand for sending traffic, RSCTP reduces congestion window using binomial congestion control. The congestion window will not be reduced largely if not congestion result in packet loss, and the sending traffic can be kept steady. As regards the discrimination policy of RSCTP packet loss cause of wireless error, RSCTP includes the function of available bandwidth estimation. The packet loss events can be differentiated by the comparison of available bandwidth and sending traffic over bottleneck network and then adjusts congestion window properly. In addition to when the network is steady, bandwidth estimation can be used to adjust slow start threshold to make congestion window not increase excessively. And in order to keep the fairness of bottleneck network bandwidth competition, RSCTP adopts the same congestion control as SCTP in congestion situation. On the other hand, RSCTP avoids reducing congestion window unnecessarily over wireless error prone link. In the multihoming mode, RSCTP will consider chunk loss rate on the primary path and switch path actively when chunk loss rate exceeds a self-defined threshold to ensure the transmission will not be terminated. Fig.3-1 shows RSCTP revises the function of original SCTP and its some new functions.

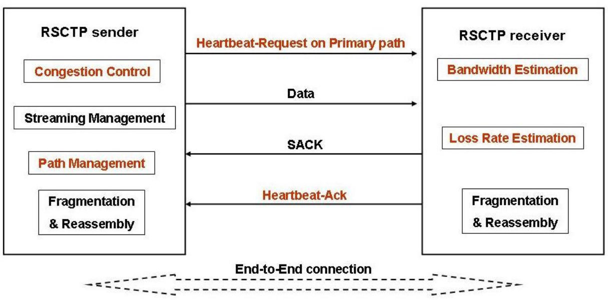

Figure 3-1: RSCTP Architecture

3.1 Bandwidth Estimation on primary path

The purpose to use bandwidth estimation is in order to judge the cause of packet loss and share bottleneck network bandwidth fairly. The bandwidth estimation is on the receiver. In multihoming mode, the packet samples are different on primary path and secondary path. So we adopt different policies on primary path and secondary path.

The bandwidth estimation method is refer to TCP Westwood [12], the author makes use the amount of data size acknowledged by SACK and calculates available bandwidth through the time interval of arriving SACK. But in the real network environment, bandwidth and latency of upload and download are not usually the same [7]. As Fig. 3-2 shows, if the time interval of receiving SACK is smaller than the time interval that data chunk sampling takes, it may over-evaluate real available bandwidth. And via the experimental result, the time interval of two successions SACK that SCTP sender received is about 0.02 seconds. It is probably unable to get real available bandwidth within such short time. For this reason we revise the method, as Fig.3-3 shows, after RSCTP association is set up, sender uses primary path to send data chunk and also send Heartbeat-Request chunk periodically. The receiver receives Heartbeat-Request and calculates available bandwidth once each time. The way to calculate available bandwidth is counting the amount of data size d_k between the interval of succession Heartbeat-Request chunk and using d_k divided by Heartbeat-Request interval $\triangle t_k$. And then sends bandwidth estimation result back to sender through Heartbeat-Ack chunk. We define B_k as available bandwidth of estimation at time t_k. The formula of calculation of B_k is defined as follows:

$$B_k = \frac{d_k}{t_k - t_{k-1}} = \frac{d_k}{\Delta t_k} \quad (4)$$

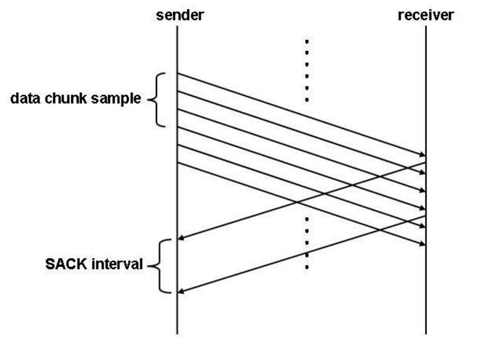

Figure 3-2: Bandwidth Estimation of TCP Westwood

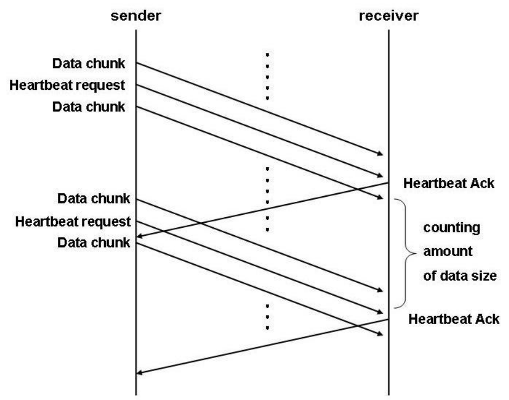

Figure 3-3: Bandwidth Estimation of RSCTP

3.2 Bandwidth Estimation on secondary path

In the original SCTP, sender will send Heartbeat-Request chunk to secondary path periodically to get reachability. In order to process available bandwidth estimation on secondary path, RSCTP changes Heartbeat-Request period from once to twice. The receiver calculates available bandwidth through the time interval of the two arriving Heartbeat-Request chunks, but it must prevent the problem of time compression of Heartbeat-Request chunk arriving time.

The time compression means the time interval of the two arrival chunks is smaller than the time interval of the sending time. The available bandwidth estimation will be higher at this moment, and it will not send back to sender. The formula of bandwidth estimation *Bs* defined as follows:

$$Bs = \frac{s_2}{t_2 - t_1} \quad (5)$$

Among them *Bs* represents available bandwidth while secondary path receives the second Heartbeat-Request chunk, s_2 is the size of the second chunk, t_1 and t_2 are the arrival time of the first and second Heartbeat-Request chunk respectively.

In the mode of multihoming, the path management of SCTP is counting retransmissions of chunk. If retransmissions of chunk of one path reach PMR (Path Maximum Retransmission), the path will be marked inactive. If primary path is inactive, new chunk transmission is through secondary path. This secondary path becomes new primary path at this moment. As regards congestion control, the new primary path increases congestion window starting at slow start phase, is the same as initial phase of what the connection is set up.

Because transmission has already lasted for some time before path switch, RSCTP sender can obtain available bandwidth of secondary path from Heartbeat-Ack chunk. Hence RSCTP modifies the practice that congestion window is increased since slow start phase after path switch. Whenever RSCTP sender gets available bandwidth estimation of secondary path, it sets up congestion window of secondary path as the product of available bandwidth value and round-trip time. As soon as path switch happens, congestion window of secondary path can be increased at higher value and throughput will be improved.

3.3 Chunk Loss Estimation

In wireless network environment, it would cause wireless signal fading that communication node may move or be block of the buildings. It will make the situation of packet loss more serious. This moment can utilize multihoming of SCTP properly. This thesis adds chunk loss rate estimation in RSCTP. RSCTP does path switch immediately as chunk loss rate of primary path exceeds a threshold MPLR (Maximum Path Loss Rate). In order to obtain chunk loss rate during data transmission, RSCTP refers to the method of [28]. The sender can count the amount of losing packets through sequences numbers of retransmission. And then packet loss rate can be calculated as follows:

$$l = \frac{COUNT(Sr)}{COUNT(P)} \quad (6)$$

In the formula *l* denotes packet loss rate, *Sr* is sequence number of retransmission, *P* denotes sending packet, and *COUNT()* denotes counting function.

In order to obtain more correct chunk loss rate, and utilize the transmission of Heartbeat-Request chunk, RSCTP processes estimation on receiver. Receiver counts the number of chunk loss in the interval of Heartbeat-Request, and then sends the result back to sender by Heartbeat-Ack chunk. The amount of sending chunk can be got from Heartbeat-Request chunk. The revised formula is as follows:

$$l = \frac{COUNT(Sp) - COUNT(Rp)}{COUNT(Sp)} \quad (7)$$

In the formula *l* denotes chunk loss rate, *Sp* is the chunk that sender sends, *Rp* denotes the chunk that receiver receives.

3.4 Modification of Heartbeat-Request/Ack chunk

Because of RSCTP congestion control added bandwidth estimation and chunk loss rate estimation, and the two communication nodes need to exchange control information. We have to do some modifications of Heartbeat-Request chunk and Heartbeat-Ack chunk.

According to RFC2960, it can be added extra control information after sixth row of Heartbeat-Request/Heartbeat-Ack chunk format. In order to accord with the demand for RSCTP, as Fig.3-4, 3-5 shows, we added current time and chunk count in the Heartbeat-Request chunk and the two fields each occupied 4 bytes. The Heartbeat-Ack chunk was added available bandwidth and chunk loss rate in addition and the two fields are 4 bytes.

Source Port Number		Destination Port Number	
Verification Tag			
Checksum			
Type	Chunk Flags	Heartbeat Length	
Heartbeat Info Type		HB Info Length	
current time		chunk count	

Figure 3-4: Heartbeat-Request chunk format of RSCTP

Source Port Number		Destination Port Number
Verification Tag		
Checksum		
Type	Chunk Flags	Heartbeat Length
Heartbeat Info Type		HB Info Length
available bandwidth		chunk loss rate

Figure 3-5: Heartbeat-Ack chunk format of RSCTP

3.5 Congestion control scheme of RSCTP

In order to insure chunk loss is cause of congestion event in wireless network, RSCTP does available bandwidth estimation on primary path. Whenever chunk loss occurs, the sender compares sending rate and available bandwidth currently. The sending rate can be obtained through congestion window divided by round-trip time. If sending rate is greater than available bandwidth, it indicates that data traffic exceeds the load of bottleneck network. The sender can conclude congestion event happens. And check if current chunk loss rate exceed MPLR and bandwidth of secondary path exceeds primary path then switch path. If chunk loss rate is lower than MPLR then standard SCTP congestion control is processed. On the contrary, if sending rate is smaller than available bandwidth, it indicates that chunk loss is cause of wireless error. We adopt binomial congestion control to reduce congestion window and check whether chunk loss rate of primary path exceeds MPLR at this moment. If chunk loss rate exceeds MPLR then switch path immediately. The procedure of the whole RSCTP congestion control is as Fig.3-7 shows.

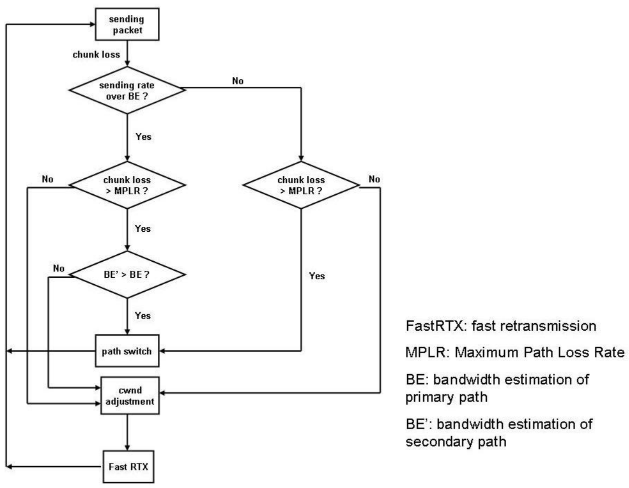

Figure 3-6: The flow chart of RSCTP congestion control

The design object of RSCTP is for steady sending rate and judges the cause of chunk loss. And if chunk loss is serious, it will check the need of path switch. The design is based on standard SCTP and accord with the end-to-end principle. Both communication nodes work the whole congestion control scheme together, so it does not need the assist of intermediate (network layer or link layer). The detail operations of sender and the receiver are as follows.

3.5.1 The operation of sender

RSCTP congestion control procedure after sender and receiver set up association by 4-way-handshake. RSCTP increases congestion window of slow start and congestion avoidance phase is the same as SCTP. The variation of congestion window expresses by equations as follows: cwnd' = cwnd + min(MTU, previous acked chunk size) (slow start) ;

cwnd' = cwnd + MTU (congestion avoidance).

During data chunk transmission, in order to obtain available bandwidth of receiver, sender sends Heartbeat-Request chunk periodically and gets available bandwidth of receiver through Heartbeat-ACK chunk. Whenever chunk loss occurs, sender compares sending rate and available bandwidth to judge chunk loss event. If congestion event occurs, the congestion window must be reduced. Otherwise it is wireless error that causes chunk loss and then reduces congestion window by binomial congestion control scheme. At the same time if chunk loss rate exceeds MPLR and available bandwidth of secondary path is greater than primary path then switch path actively to keep data transmission. The slow start threshold must be set up as product of available bandwidth and RTT if there is no chunk loss. The purpose is to avoid the excessive increase of congestion window.

There are two kinds of situations of chunk loss, one is receiving four duplicated SACKs and other is retransmission timeout. When receiving four duplicated SACKs or retransmission timeout is indeed cause of congestion event, RSCTP processes standard congestion control of SCTP. Upon detection of chunk losses from SACK, an endpoint should do the following:

ssthresh = max(cwnd/2, 2*MTU)

cwnd = ssthresh

If wireless error causes chunk loss, then reduces congestion window with SQRT binomial congestion control algorithm. The variation of congestion window expresses with equations as follows:

cwnd' = MAX(cwnd / 2, 2 × MTU); (congest and received 4-duplicated SACK)

cwnd' = MTU; (congest and retransmission timeout)

cwnd' = cwnd – $\beta \times cwnd^{1/2}$; (wireless error)

```
The pseudo code of sender's operation:
begin
   sending Heartbeat request periodically

      if (! chunk loss) {
       standard SCTP congestion control
       ssthresh = Bandwidth Estimation × rtt
      }

   if (Data chunk loss or Heartbeat Ack loss) {
      // congestion
      if (current sending rate > Bandwidth Estimation) {
       if (4 Duplicated SACK are received) {
          ssthresh = MAX(cwnd/2, MTU)
          cwnd = ssthresh
       }
       else if (RTX timeout) {
          ssthresh = MAX(cwnd/2, MTU)
          cwnd = MTU
       }
       retransmit lost packet

      if (packet loss rate > MPLR and BE < BE')
       path switch

      } // wireless error
      else if (current sending rate <= Bandwidth estimation)
{
            cwnd = cwnd - β×cwnd^(1/2)
            retransmit lost packet

        if (packet loss rate > MPLR)
           path switch
      }
```

Figure 3-7: The pseudo code of sender operation

3.5.2 The operation of receiver

In a situation that the network is steady, the work of receiver is to receive and acknowledge Heartbeat-Request chunk. But in order to do congestion control with sender in coordination, RSCTP added the function of available bandwidth estimation on receiver. If that association is in multihoming mode, the bandwidth estimation methods of primary path and secondary path are different. The detailed works has already been explained in section 3.1 and 3.2. The receiver triggers bandwidth estimation by Heartbeat-Request chunk, and sends the result back to sender through Heartbeat-Ack chunk. Then sender uses the available bandwidth value to process congestion control of RSCTP.

Under the mode of multihoming, in order to switch path actively when network congestion and wireless error are serious, RSCTP includes the function of chunk loss rate estimation on receiver. The way of calculation has already introduced in chapter 3.3. The receiver receives once Heartbeat-Request and does chunk loss rate estimation once, and then sends the results including available bandwidth back to sender. The sender can determine whether to switch path or not by these information when chunk loss occurs. The pseudo code of receiver operation is as Fig. 3-8.

```
The pseudo code of receiver's operation:
begin
   if (Received Heartbeat-Request) {
      doing Available Bandwidth Estimation
      doing Chunk Loss Rate Estimation
      }

   Attach Available Bandwidth and Chunk Loss Rate to
     Heartbeat-Ack chunk

   Send Heartbeat-Ack Back

end

Available Bandwidth Estimation()
begin
  primary path:
   Available Bandwidth
      = received data size / timing gap of Heartbeat Request

  secondary path:
   if (! time compression)
     Available Bandwidth
       = size of Heartbeat / timing gap of Heartbeat Request

   cwnd = Available Bandwidth × rtt
```

Figure 3-8: The pseudo code of receiver operation

3.6 Convergence to fairness

According to the conclusion of binomial algorithm, all the binomial control protocols converge to fairness as long as the indexes $k \geqq 0$, $l \geqq 0$ and $k + l > 0$. We can express congestion control equations of RSCTP with binomial algorithm as following (The α and β are constants). List the equation of no chunk loss at first, $k = 0$ among it:

cwnd' = cwnd + α

The following binomial congestion control equation is cause of congestion event; the index l is equal to 1:

cwnd' = cwnd – β×cwnd

The following binomial congestion control equation is cause of wireless error; the index l is equal to 1/2:

cwnd' = cwnd – β×cwnd$^{1/2}$

Therefore no matter what kind of situation of chunk losses, the indexes k and l satisfy $k \geq 0$, $l \geq 0$ and $k + l > 0$. It is indicated RSCTP can converge to fairness.

IV. SIMULATION RESULTS

This thesis uses NS2 (Network Simulator version 2) [29][31] as platform of performance simulation of network. NS2 is a set of open source, it was built by two kinds of language including C++ and Otcl(Object Tool Command Language). It utilizes C++ to compile and process fast, network protocol of every layer is built by C++. And Otcl is a kind of interpreter language. It's be used for dealing with network scenario and it can be convenient to revise the network environment of simulation. NS2 combines the advantages of these two kinds of language. The user uses C++ to develop network protocol, and use Tcl to set up network environment. Because simulation environment and result of NS2 are very close to the truth, more and more research about performance of network adopts NS2 to work on network simulation.

We revise SCTP mould [30] of NS2 to build RSCTP then use NS2 to process performance simulation. We need to decide some parameters correlated with RSCTP at first, including β of decreasing congestion control equation, and Heartbeat-Request chunk interval of primary path and secondary path. And observe the accuracy of bandwidth estimation and chunk loss rate estimation too. Then simulate the situations of network congestion and wireless error to compare the performance of RSCTP and SCTP, and determine a suitable MPLR. Finally we evaluate Fairness and TCP-Friendly of RSCTP.

4.1 Simulation experiment environment set up

This thesis investigates mainly into end-to-end congestion control, we do not consider the moving of mobile host and its relative position located in the range of wireless signal. We suppose the position of mobile host is fixed, and the wireless reception is normal. The wireless error causes packet losses utilizes the Error Model of NS2.

The simulation topology is as Fig.4-1 shows. Two hosts have two network interfaces each. Bandwidth and latency are noted near by every section. The sender is one fixed host located in wired network. The receiver is a mobile host, which two network interfaces communicate with access point. The Error Model is set up between receiver and wireless access point. The bottleneck network between router and access point may have other cross traffic. The dotted line arrows express end-to-end paths between sender and receiver. One of them is primary path, another one is secondary path after SCTP association is set up.

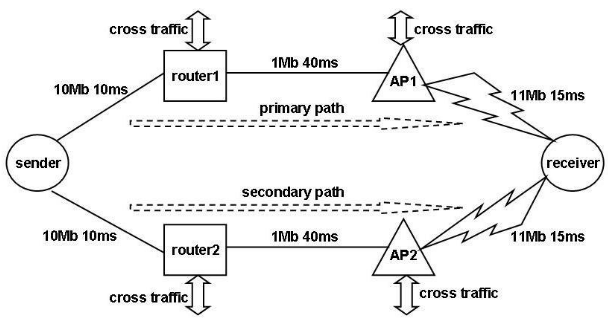

Figure 4-1: Simulation topology

4.2 RSCTP parameter settings

Before evaluate the performance of RSCTP, we must decide the relevant parameters of RSCTP. They are including β of congestion control decreasing function, and Heartbeat-Request interval of primary path and secondary path. The simulation result shows that throughput was not influenced deeply of different β, so we set up β value is 0.2.

In order to process available bandwidth estimation on receiver, RSCTP will send Heartbeat-Request chunk to primary path periodically. When the interval of Heartbeat-Request chunk changes, the amount of data size that receiver can take also varies with it. It will have great influence on bandwidth estimation result. We adjust different Heartbeat-Request interval and determine proper Heartbeat-Request interval according to the accuracy of bandwidth estimation. Fig.4-2 shows the experimental result of RSCTP bandwidth estimation adaptability. The experiment method is to join cross traffic in bottleneck network to change available bandwidth of sender. The brown curve of Fig.4-2 shows the change of real available bandwidth of bottleneck network, other curves show the bandwidth estimation results while using different Heartbeat-Request interval of RSCTP.

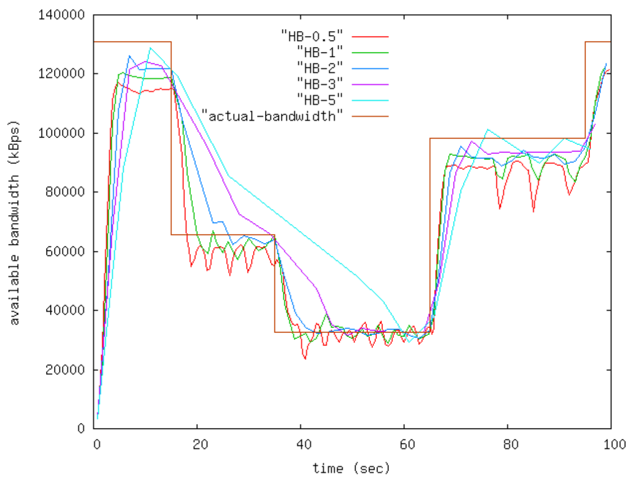

Figure 4-2: Bandwidth estimation adaptability

Table 4-1 shows bandwidth estimation error and overhead of RSCTP sender uses different Heartbeat-Request interval. The experimental shows that the interval of one second can get the minimum error of 13.55%. For this reason we determine Heartbeat-Request interval is one second and its overhead is 0.07%. Experimental results showed that the actual bandwidth of bottleneck network is 1/4 of original while simulation time from 15 seconds to 35 seconds. As Heartbeat-Request interval exceeds more than 3 seconds sender is unable to obtain available bandwidth of receiver immediately.

And then we determine Heartbeat-Request interval of secondary path. In the multihoming mode, SCTP will send Heartbeat-Request chunk to secondary path to get reachability. RSCTP changes to send two Heartbeat-Request chunks by once in order to process bandwidth estimation of secondary path. Hence Heartbeat-Request interval and bandwidth estimation of secondary path do not have direct relation. If the Heartbeat-Request interval is shorter, the more times of available bandwidth of secondary path that sender can get. Because of secondary path is idle while primary path is in transmission. So its Heartbeat-Request interval does not need to be as frequent as primary path. We set up Heartbeat-Request interval of secondary path as 4 seconds finally which were 4 times of primary path. We also adjust congestion window while RSCTP processes bandwidth estimation on secondary path. The experimental result shows RSCTP can adjust congestion window of secondary path to reach 17.3 times of initial value. It will improve performance when secondary path transmission begins.

Table 4-1: Error and overhead of different Heartbeat-Request interval

Heartbeat-Request interval(sec)	0.5	1	2	3	5
frequency of bandwidth estimation	192	92	48	26	14
error (%)	14.26	13.5 5	14.1 3	15.2 6	33.0 1
overhead(%)	0.14	0.07	0.04	0.02	0.01

4.3 Wireless loss performance comparison

After determined the parameters that RSCTP needs, this section compares performance in case of wireless error of SCTP with RSCTP. The chunk loss rate is set up from 0% to 10% to observe the performance of RSCTP and SCTP.

Fig.4-3 shows the throughput variation when wireless error is applied only for primary path of RSCTP and SCTP and there is no cross traffic. The simulation result shows RSCTP can improve average throughput to get up to 20.8%. It is because that RSCTP has more chances to prevent unnecessary degrading congestion widow size cause of wireless error. But when chunk loss rate is higher, both RSCTP and SCTP can succeed to send chunk is lower.

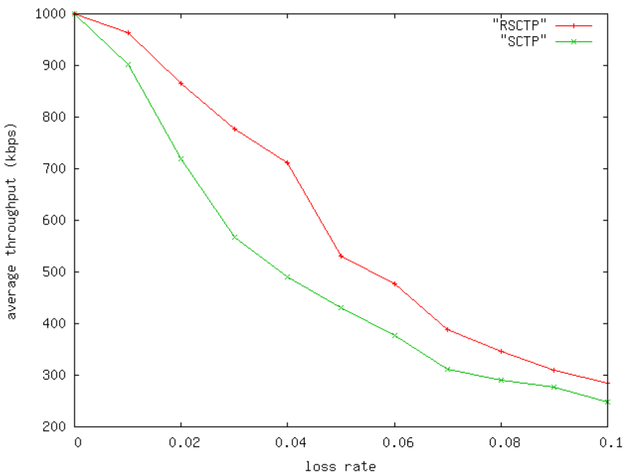

Figure 4-3: throughput comparison with 0~10% loss on primary path

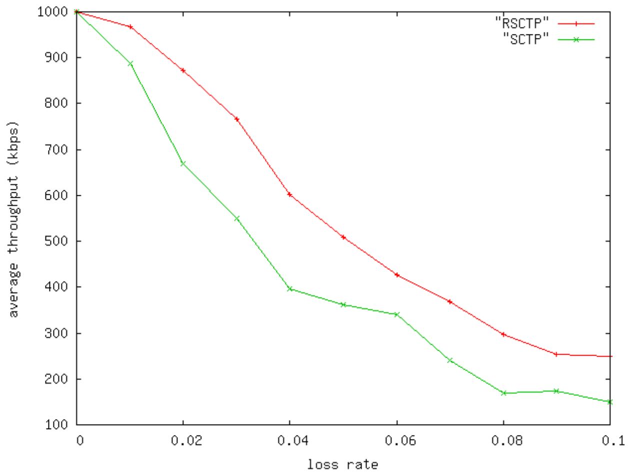

Figure 4-4: throughput comparison with 0~10% loss on both paths

And then wireless error is operated both on primary and secondary path. The purpose is to let duplicated SACK and retransmission timeout appeared simultaneously. The experimental result of Fig.4-4 shows the variation of throughput as 0%~10% chunk loss rate of primary path and secondary path. RSCTP can improve average throughput up to 40%. This experiment efficiency is superior to the situation of wireless error will take place only on primary path. The reason is when chunk loss happens for a long time on secondary path of SCTP, it is more frequent to result in retransmission timeout. And RSCTP have more chance to distinguish wireless error from congestion, so it improves performance efficiency more obviously.

The result of Fig.4-3 shows that throughput of RSCTP is close to 1/2 of available bandwidth of 5% chunk loss rate. We add the function of chunk loss rate estimation and active path switch to RSCTP. It utilizes a self-defined Maximum Path Loss Rate (MPLR) that is set up as 0.05. RSCTP will process path switch actively when chunk loss rate exceeds 0.05. And secondary path will become new primary path.

The result of Fig.4-5 is the throughput variation of RSCTP versus RSCTP path switch and the loss rate is 5%. Loss rate is on primary path and operated at 30 seconds of simulation time. The red curve is RSCTP with path switch function. It detected that chunk loss rate exceeded 0.05 at 47 seconds and then switched primary path. Because of secondary path has no wireless error the performance is obviously superior to RSCTP without path switch (green curve).

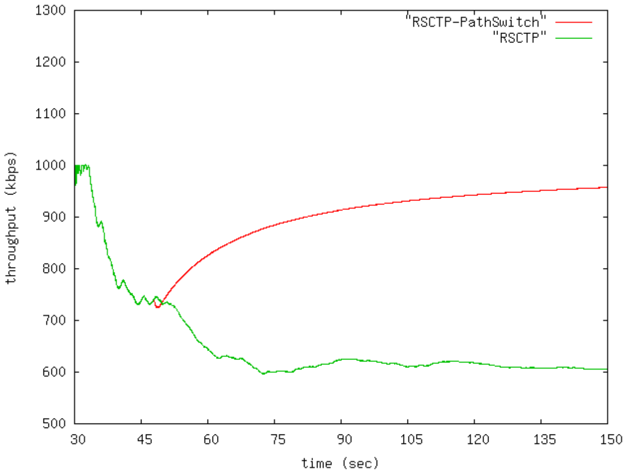

Figure 4-5: throughput comparison with 5% loss of RSCTP versus RSCTP path switch

4.4 Fairness

No matter which kind of network environment, connections of the same protocol must keep fairness for bandwidth competition. This section processes the experiment of competition fairness of bandwidth for RSCTP. The design of the experiment situation is two RSCTP connections were started at the same time to compete bandwidth of bottleneck network. Through the variation of throughput of two connections to observe whether bottleneck bandwidth is shared fairly.

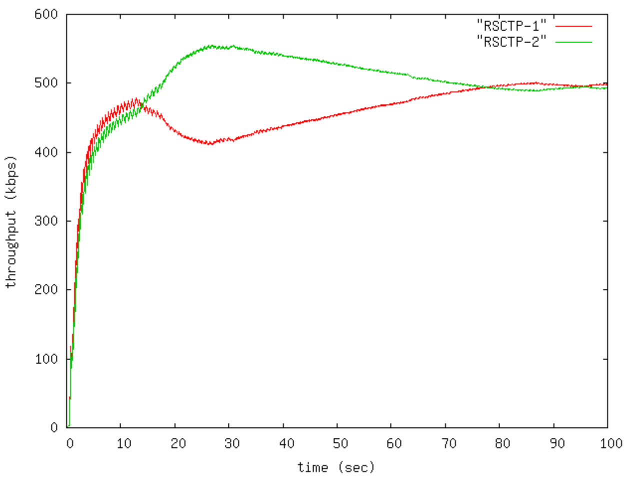

Figure 4-6: RSCTP fairness of two connections

Fig.4-6 is the experimental result of fairness. Two connections of RSCTP-1 and RSCTP-2 started at 0 seconds and transmitted for 100 seconds. Two's throughput behaves equally before 14 seconds. RSCTP-2 got more bandwidth than RSCTP-1 from 14 seconds to 30 seconds, so its throughput is superior to RSCTP-1. Because of two connections can adjust slow start threshold to closing to available bandwidth. The bandwidth competition of two connections reached the fairness gradually after 30 seconds. Therefore RSCTP is accord with the fairness of bandwidth competition.

4.5 TCP-Friendly

Most internet applications use TCP to be transport layer protocol mainly. We proposed RSCTP needs to follow TCP friendly characteristic to avoid compressing TCP transmission in wireless network. The scenario in this chapter is set up loss rate of wireless section as 1% and 5% respectively. RSCTP connection is started 30 seconds later after TCP connection. Then observe the throughput variation of the two connections.

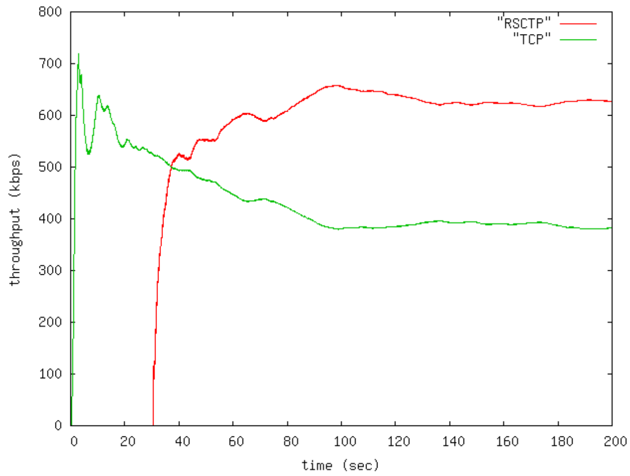

Figure 4-7: 1% loss rate of TCP-Friendly

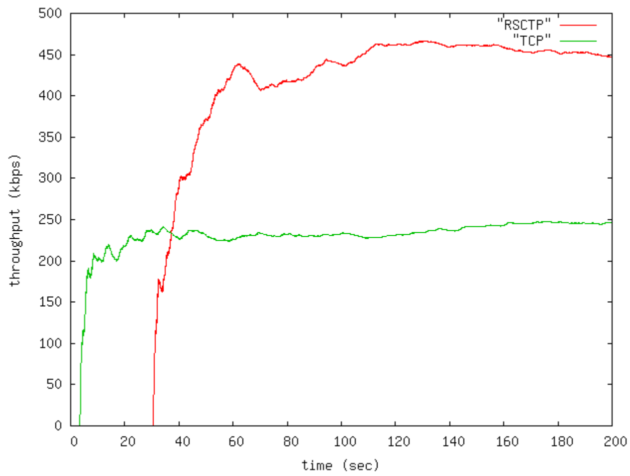

Figure 4-8: 5% loss rate of TCP-Friendly

Fig.4-7 is the experimental result of 1% loss rate. TCP and RSCTP start at 0 seconds and 30 seconds respectively. Two connections keep transmission until the end of simulation time. Because of TCP congestion control can't distinguish the reason of packet loss. As simulation time being longer, wireless error causes more packet losses and result in more times of unnecessary congestion window reducing. TCP is unable to use the network resources fully. After RSCTP is triggered, because RSCTP can prevent reducing congestion window unnecessary cause of wireless error. It can use the network resources that TCP has not been utilized properly. In addition RSCTP can adjust slow start threshold equal to available bandwidth to avoid the congested window not increasing excessively. Hence RSCTP will not influence the throughput of TCP and maintain the characteristic of TCP friendly. As Fig.4-8, we can observe RSCTP gets higher throughput than TCP in 5% wireless loss rate environment. TCP's performance is worse over high wireless error rate environment. And consequently RSCTP can achieve higher throughput.

CONCLUSION AND FURTHER WORKS

This thesis proposes one adaptive congestion control scheme with bandwidth estimation on receiver based on SCTP, called RSCTP. Sender judges chunk loss events by comparing available bandwidth of receiver with sending rate. The similar method in the past is to process bandwidth estimation on sender. RSCTP moves bandwidth estimation to receiver. It can avoid over-evaluating available bandwidth among asymmetric bandwidth network. And in order to let receiver process bandwidth estimation and let sender obtain the estimation results. We change 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. In addition, we also modify Heartbeat-Request/Heartbeat-ACK chunk to exchange control information.

Primitive SCTP is unable to switch primary path actively under multihoming mode. We add chunk loss rate estimation in RSCTP. When sender detects that chunk loss rate exceeds Maximum Path Loss Rate will switch primary path immediately to keep data transmission. And we enable secondary path to adjust congestion window to make congestion window be improved to the peak load of bottleneck bandwidth. When secondary path becomes new primary path it can use higher initial value of congestion window.

The experimental result shows that wireless network has certain chunk loss rate, RSCTP can really improve throughput of SCTP. Only in a situation that primary path sets up wireless error, RSCTP can improve wireless network throughput is up to 20.8%. When primary path and secondary path all set up wireless error, RSCTP that improved throughput is up to 40%. It shows RSCTP is suitable for wireless network environment. In addition we also proved RSCTP can reach fairness and TCP-friendly.

Via the adaptability experimental result of secondary path bandwidth estimation, RSCTP can probably estimate bandwidth of secondary path. But when other traffic joins to compete with bandwidth of bottleneck network at the same time, RSCTP can't adapt to the variation of available bandwidth. The estimated result is still primitive bandwidth of bottleneck network. This is because RSCTP uses Heartbeat-Request chunk as the sample source of calculating bandwidth. And Heartbeat-Request chunk only has 44Bytes. Data size is too small and unable to fill all available bandwidth. RTT will not extend cause of cross traffic join in bottleneck network. So we will do further discussion to the adaptability of bandwidth estimation of secondary path in the future.

As the more wireless network technology, the higher demand of mobility for mobile communication. Packet loss in wireless network is more and more variety too. The user in moving needs to process handover between different base stations also meets packet loss easily. SCTP multihoming can offer the good solution for handover. But SCTP is unable to do path switch actively. Then how to apply SCTP to handover and keep data transmission is a topic that is worth further investigating. Load balancing of SCTP is paid much attention at present. RSCTP can be applied on the topic in the further.

REFERENCES

[1] G. Xylomenos, G.C. Polyzos, P. Mahonen and M. Saaranen, TCP performance issues over wireless links, *IEEE Communications Magazine 39*, pp. 52-58., 2001 (4)

[2] Y. Tian, K. Xu, and N. Ansari, "TCP in Wireless Environments: Problems and Solutions," *IEEE Commun. Mag.*, vol. 43, no. 3, pp. S27-S32. , 2005

[3] M. Allman, V. Paxson, and W. Stevens. TCP Congesfion Control, RFC 2581, April 1999.

[4] H. C. Lee and Fongray Frank Young, "Congestion Scheme Investigations on TCP/IP Network," *2003 Symposium on Century Digital Life and Internet Technologies*, Taiwan., P0317 Page 1-8, 2003

[5] S. Floyd, "HighSpeed TCP for Large Congestion Windows" , IETF Internet Draft, Feb 2003.

[6] Balakrishnan, H., et al., "Improving TCP/IP Performance over Wireless Networks. in *Int. Conf. on Mobile Computing and Networking.* Berkeley, CA 1995.

[7] K. Xu, Y. Tian, and N. Ansari, "Improving TCP performance in integrated wireless communication networks," vol. 47, pp. 219-237, February 2005.

[8] C. Liu and R. Jain, "Approaches of Wireless TCP enhancement and a new proposal based on congestion coherence," in *Proc. 36th Annu. Hawaii Int. Conf.*, pp. 307-316. , Jan. 2003

[9] Lawrence S. Brakmo, and Larry L. Peterson., "TCP Vegas: End to End Congestion Avoidance on a Global Internet", *IEEE J. Select. Areas. Commun* , vol. 13, pp. 1465-1480, Oct. 1995.

[10] Cheng Peng Fu, and Soung C. Liew, "TCP Veno: TCP Enhancement for Transmission over Wireless Access Networks", *IEEE J. Select. Areas. Commun*, vol. 21 , pp. 216-228, Feb. 2003.

[11] A. Jayananthan, and H.R. Sirisena, "TCP Performance Enhancement over WLAN with Wireless Loss Detection Proxy", *Information, Communications and Signal Processing, 2005 Fifth International Conference*, Page(s):654-658, 06-09 Dec. 2005

[12] D. Bansal and H. Balakrishnan, "TCP-friendly congestion control for real-time streaming applications," Technical Report MIT-LCS-TR-806, Massachusetts Institute of Technology, Cambridge, Massachusetts, U.S.A., May 2000.

[13] D. Bansal and H. Balakrishnan, "Binomial Congestion Control Algorithms," *In Proceedings INFOCOM 2001*, April 2001.

[14] CHIU, D.-M., AND JAIN, R. Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks. *Computer Networks and ISDN Systems 17* , pp.1-14. , 1989

[15] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth estimation for enhanced transport over wireless links," in *Proc. ACM Mobicom*, Rome, Italy, pp. 287-297. , July 2001

[16] Grieco, L. A. , and Mascolo, S., End-to-End Bandwidth Estimation for Congestion Control in Packet Networks. Second International Workshop, QoS-IP 2003, Milano, Italy, February 2003.
[17] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. Stream Control Transmission Protocol. RFC2960, October 2000.
[18] A. L. Caro et al., "SCTP: A Proposed Standard for Robust Internet Data Transport," *IEEE Comp.*, vol. 36, no. 11, pp. 56-63. , Nov. 2003
[19] Shaojian Fu; Atiquzzaman, M.; "SCTP: state of the art in research, products, and technical challenges," *Computer Communications*, pp. 85-91, Oct. 2003.
[20] R. Stewart et al., Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration, Internet Draft, draft-ietf-tsvwg-addip-sctp-08.txt, Sept. 2003.
[21] J. Shi, Y. Jin, H. Huang, D. Zhang, "Experimental performance studies of SCTP in wireless access networks," in *Proc. Of ICCT'03*, vol. 1, pp. 392-395. , Apr. 2003
[22] I Ahmed, O Yasuo, K Masanori, "Improving Performance of SCTP over Broadband High Latency Networks," *Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks* , 2003
[23] G. Camarillo, R. Kantola, and H. Schulzrinne, "Evaluation of Transport Protocols for the Session Initiation Protocol", *IEEE Network* , Volume 17, Issue 5, Page 40-46, Sept.-Oct. 2003.
[24] Stewart,R., M. Ramalho, Q. Xie, M. Tuexen and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", RFC 3758, IETF, May 2004.
[25] H. Huang, J. Ou, and D. Zhang, "Efficient Multimedia Transmission in Mobile Network by using PR-SCTP", *Communications and Computer Networks*, Marina del Rey, USA, 2005
[26] S. Kashihara, K. Iida, H. Koga, Y. Kadobayashi, and S. Yamaguchi. "End-to-End Seamless Handover using Multi-path Transmission Algorithm", *In IC*, Tokyo, Japan, 2003.
[27] Dapeng Wang, Shoubao Yang, Weifeng Sun, "A Mac-Error-Warning Method for SCTPCongestion Control over High BER Wireless Network", *Proceedings 2005 International Conference on Wireless Communications, Networking and Mobile Computing*, pp.513-516, 2005
[28] Tomohiko Ogishi, Akira Idoue, Toru Hasegawa, and Toshihiko Kato,"Design of Performance Monitor Collecting TCP Level Statistics from Unidirectional IP Traffic," *IEICE General Conference*, B-7-3, March 2000.
[29] The Network Simulator - ns-2: http://www.isi.edu/nsnam/ns/
[30] SCTP module of ns2 simulator: http://pel.cis.udel.edu