

Research Article

Background Traffic-Based Retransmission Algorithm for Multimedia Streaming Transfer over Concurrent Multipaths

Yuanlong Cao,¹ Changqiao Xu,^{1,2} Jianfeng Guan,¹ and Hongke Zhang^{1,3}

¹ State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

² Institute of Sensing Technology and Business, Beijing University of Posts and Telecommunications, Jiangsu, Wuxi 214028, China

³ National Engineering Laboratory for Next Generation Internet Interconnection Devices, Beijing Jiaotong University, Beijing 100044, China

Correspondence should be addressed to Yuanlong Cao, ylcao@bupt.edu.cn

Received 1 December 2011; Accepted 2 May 2012

Academic Editor: János Tapolcai

Copyright © 2012 Yuanlong Cao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The content-rich multimedia streaming will be the most attractive services in the next-generation networks. With function of distribute data across multipath end-to-end paths based on SCTP's multihoming feature, concurrent multipath transfer SCTP (CMT-SCTP) has been regarded as the most promising technology for the efficient multimedia streaming transmission. However, the current researches on CMT-SCTP mainly focus on the algorithms related to the data delivery performance while they seldom consider the background traffic factors. Actually, background traffic of realistic network environments has an important impact on the performance of CMT-SCTP. In this paper, we firstly investigate the effect of background traffic on the performance of CMT-SCTP based on a close realistic simulation topology with reasonable background traffic in NS2, and then based on the localness nature of background flow, a further improved retransmission algorithm, named *RTX_CSI*, is proposed to reach more benefits in terms of average throughput and achieve high users' experience of quality for multimedia streaming services.

1. Introduction

The content-rich multimedia streaming, such as video-on-demand (VoD) [1, 2] and Internet Protocol Television (IPTV) will be the most attractive services in the next-generation networks. Most researches have proved the Stream Control Transmission Protocol (SCTP) will be the most promising technology for the large bandwidth consumption of multimedia streaming services [2–4]. Particularly in the future wireless heterogeneous network that the terminals will be equipped with multiple network interfaces and attached multiple heterogeneous access capability at the same time, the SCTP can provide the effective transmission for multimedia streaming services and balance the overhead among multiple access networks.

The SCTP [5] has been proposed and standardized by the Internet Engineering Task Force (IETF) in order to effectively utilize the multihoming environment and support real-time signaling transmission over IP networks, since

SS7 has been the only bearer for the signaling traffic in telecommunication networks [6] for many years. SCTP has some important features including: (1) multi-homing. The destination nodes can be reached under the several IP addresses (multi-homed). In SCTP, both sides of the association provide multiple IP addresses combined with a single SCTP port number [7]. (2) Multistreaming which means the parallel transmission of messages over the same association between sender and the receiver. The stream independently carries fragmented messages from one terminal to another, which can achieve a cumulative throughput [8] than other protocols (e.g., TCP). SCTP manages more than one communication path with two major functions: (a) using SACK (selective acknowledgment) to probes primary path connectivity and HEARTBEAT to probe the alternative paths, respectively; (b) fail-over which means once the primary path breaks and selects an alternative path as the primary path.

As an improved version of SCTP, Concurrent Multipath Transfer (CMT) [9] uses the SCTP's multi-homing feature to distribute data across multiple end-to-end paths in a multi-homed SCTP association. CMT is the concurrent transfer of new data from a source to a destination via more than one end-to-end paths, and it is used between multi-homed source and destination hosts to increase the throughputs. Moreover, a CMT sender can maintain more accurate information (such as available bandwidth, loss rate, and RTT) of all the paths, since new data are sent to all destinations concurrently. This feature allows the CMT sender to better decide where to retransmit once data is lost.

There is more and more researches pay attention to multimedia streaming, and CMT-SCTP had been employed as transport protocol as well to study the performance of multimedia streaming services. For example, Stegel et al. [10] proposed solutions on how to provisioning SCTP multi-homing in converged IP-based multimedia environment. Huang and Lin [11] proposed a partially reliable-concurrent multipath transfer (PR-CMT) protocol for multimedia streaming in order to improve the throughput and video quality degrade. In our previous work, we designed a novel Evalvid-CMT platform [3, 4] to investigate and evaluate the performance of CMT for real-time video distribution, and then a meaningful suggestion was pointed out on which strategies for real-time video concurrent multipath transmissions.

Although the advantages of CMT-SCTP has been investigated in variety of attractive services, however, existing evaluation works [1–16] of CMT-SCTP do not consider the impact of background traffic. Actually, Internet measurement studies showed complex behaviors of Internet traffic [17, 18] that are necessary for realistic testing environments. There are several reasons why background traffic is important in performance testing. First, the aggregate behavior of background traffic can induce a rich set of dynamics such as queue fluctuations, patterns of packet losses, and fluctuations of the total link utilization at bottleneck links and can have a significant impact on the performance of CMT-SCTP. Second, network environments without any randomness in packet arrivals and delays are highly susceptible to the phase effect [19], and a good mix of background traffic reduces the likelihood of synchronization [20]. Third, the core of the Internet allows a high degree of statistical multiplexing. Therefore, the performance evaluation of network protocols with little or no background traffic does not fully investigate the CMT-SCTP behaviors that are likely to be observed when it is deployed in the Internet.

On the other hand, there are five retransmission algorithms proposed in [12] to enhance the performance of CMT-SCTP. Previous work [9, 14] take major researching focus on the effects of different retransmission algorithms with different limited receive buffer (*rbuf*) sizes. However, all of the five retransmission algorithms are designed with only one of paths' condition as metric. Liu et al. [16] combines some paths' conditions to select the retransmission path but with an unreasonable metric since loss rate is not be recommended according to RFC4460.

In this paper, taking reasonable background traffic into account, we firstly investigate the effect of background traffic on the performance of CMT-SCTP based on a more realistic simulation topology in NS2 [21]. Considering the nature of background traffic and taking paths' previous states into account, we further propose an improved retransmission algorithm named *RTX_CSI* to achieve more benefits in terms of average throughput and high users' experience of quality for multimedia streaming services.

The rest of paper is organized as follows. Section 2 explains our experimental design for network redundancy in CMT-SCTP. Section 3 presents how effects occurred by designed background traffic. Section 4 addresses the proposed *RTX_CSI* algorithm and its performance evaluation. Section 5 concludes this paper and discusses the future work.

2. Preliminary Work

2.1. Background Traffic Design. In accordance with the Internet survey [22], TCP traffic on the Internet is about 80–83%, and UDP traffic is about 17–20%. Moreover, the content-rich multimedia streaming will be the most attractive services in the future networks, more and more multimedia encoded by VBR will be deployed in Internet. Thus, more reasonable background traffic consists of TCP traffic, CBR traffic, and VBR traffic should be taken into account to evaluate the performance of data delivering.

With the purpose of investigating the effect of background traffic on the performance of CMT-SCTP, our experiments adopt a more realistic simulation scenario for network redundancy in CMT-SCTP, that is, TCP traffic, CBR traffic, and VBR traffic will be employed in our simulation topology designing. Test scenario consists of one path with TCP traffic and UDP/CBR traffic as background traffic (TCP:UDP/CBR is 4:1) and another with TCP traffic and UDP/VBR traffic as background traffic (TCP:UDP/VBR is 4:1) which is represented by *TCP+UDP/CBR* and *VBR* in below.

2.2. VBR Traffic Generator Loading. Since NS2 still cannot support VBR traffic, in order to enable VBR traffic generator in NS2, we add *PT_VBR* as packet enumeration and then set *VBR* for *PT_VBR*'s value in packet information function [23]. The default values for VBR traffic are set as shown in Table 1.

2.3. Simulation Topology Setup. To investigate the impact of background traffic on the performance of CMT-SCTP completely, a more realistic simulation topology with reasonable background traffic is proposed, which is shown in Figure 1. In the dual dumbbell topology, each router ($R1 \sim R4$) connects to five edge nodes. The edge nodes are single interfaces and connect to the routes to generate the background traffic. Each edge node attaches with a traffic generator, and four edge nodes generate 80% TCP traffic and one edge node generates 20% UDP traffic (CBR or VBR). According to [24], the propagation delay between the edge nodes and routers is set to 5 ms in order to create the maximum effect occurred by background traffic, and bandwidth is set to 100 Mb. The propagation delay between

TABLE 1: Parameter settings of VBR traffic.

Variable	Value
<i>Application/traffic/VBR set rate</i>	448 Kb
<i>Application/traffic/VBR set random</i>	0
<i>Application/traffic/VBR set max pkts</i>	268435456
<i>Application/traffic/VBR set max Size</i>	200
<i>Application/traffic/VBR set min Size</i>	100
<i>Application/traffic/VBR set intervaltime</i>	200

two routers is set to 45 ms with 10 Mb of bandwidth in accordance with article [25] (CMT-PF addressed in this article is not employed in our experiments for pure study of the impact occurred by background traffic).

The *S* and *R* stand for CMT-SCTP sender and receiver, respectively, and connected to the network through two interfaces. CMT-SCTP uses concurrent multipath transfer to send data on both paths with the default parameters recommended by RFC4460. After 0.5 seconds of simulation, the CMT-SCTP sender begins to initiate the association with CMT-SCTP receiver. At 1.0 seconds, edge nodes generate the background traffic, and the total simulation time is 30 seconds.

3. Study of the Impact of Background Traffic

To analyze the impact of the background traffic, this section evaluates the average throughput (delay) of CMT-SCTP with and without background traffic, respectively. To measure the presence of background traffic affecting the performance of CMT-SCTP, we define a metric called *Impact Degree* (denoted as Θ) which can be expressed by

$$\Phi(\Theta_\alpha) = \frac{|\Phi(x_\alpha) - \Phi(y_\alpha)|}{\Phi(x_\alpha)} \times 100\%, \quad (1)$$

where α stands for *rbuf*; x stands for CMT-SCTP without background traffic condition and y for CMT-SCTP with *TCP + UDP/CBR* and *VBR* traffic condition; $\Phi(x_\alpha)$ is on behalf of average throughput or average delay achieved by CMT-SCTP without background traffic under different α ; $\Phi(y_\alpha)$ for average throughput or average delay created by CMT-SCTP with *TCP + UDP/CBR* and *VBR* traffic under different α ; $\Phi(\Theta_\alpha)$ stands for impact degree arisen by *TCP + UDP/CBR* and *VBR* traffic. High $\Phi(\Theta_\alpha)$ means that background traffic has high side effects on CMT-SCTP in terms of throughput and delay, that is, low average throughput (high average delay) CMT-SCTP will be reached.

Since default *rbuf* size commonly used in operating systems today is varied from 16 KB to 64 KB and beyond. Herein, we investigate the impact of background traffic on CMT-SCTP under *rbuf* size with 16 KB, 32 KB, 64 KB, 128 KB, and 256 KB. Figures 2, 3, 4, 5, and 6 show throughput reached by CMT-SCTP with or without background under different size of *rbuf*, respectively (the measuring interval is 0.5 s).

As illustrated in Figures 2, 3, 4, 5, and 6, we can point out that: (1) the background traffic presents an impact on

throughput clearly; (2) with the increase of the receive buffer, the impact of the background traffic is increased.

Figure 7 shows the comparison on average throughput with and without background traffic under different *rbuf*.

Figure 8 shows the comparison on average delay with and without background traffic under different *rbuf*, respectively.

Based on (1) and above simulation results, Figure 9 shows the corresponding impact degree which occurred by the designed background traffic.

As it shown in Figure 7, when *TCP + UDP/CBR* and *VBR* is employed as the background traffic, the impact degree on average throughput can be calculated as $\Phi(\Theta_{16}) \approx 0.0025$, $\Phi(\Theta_{32}) \approx 0.0047$, $\Phi(\Theta_{64}) \approx 0.0278$, $\Phi(\Theta_{128}) \approx 0.2652$, and $\Phi(\Theta_{256}) \approx 0.5238$. Figure 9 illustrates that larger *rbuf* will lead to larger impact degree, namely, larger side effect will occur by background traffic in terms of average throughput.

As it shown in Figure 8, when *TCP + UDP/CBR* and *VBR* is employed as the background traffic, the impact degree on average delay can be calculated as $\Phi(\Theta_{16}) \approx 0.05019$, $\Phi(\Theta_{32}) \approx 0.0721$, $\Phi(\Theta_{64}) \approx 0.1235$, $\Phi(\Theta_{128}) \approx 0.4941$, and $\Phi(\Theta_{256}) \approx 0.3037$. From Figure 9, we note that larger *rbuf* will lead to larger impact degree. However, when *rbuf* is set more than 256 KB, the impact will be reduced, the reason maybe that data can be received timely as greater receive buffer is used.

From above experiments and analysis, we can conclude that the background traffic can present an obvious impact on CMT-SCTP's performance in terms of throughput and delay, and it will lead to some known problems like congestion. Thus, we need to take background traffic condition into account during designing the retransmission algorithm.

4. RTX_CSI Algorithm

Retransmission algorithms play a more important role in achieving high users' experience of quality for multimedia streaming services. As mentioned in Section 1, there are five retransmission schemes proposed [12] for CMT-SCTP, we call them as *existing retransmission algorithms*. However, all of *existing retransmission algorithms* do not consider the nature of background traffic. This section will simply introduce the *existing retransmission algorithm* firstly, and then a further improved retransmission algorithm named *RTX_CSI* will be addressed with considering background traffic condition, a necessary performance evaluation will be presented lastly.

4.1. Existing Retransmission Algorithm.

RTX-SAME. Once a new data chunk is scheduled and sent to a destination, all retransmissions of the chunk thereafter are sent to the same destination (until the destination is deemed inactive due to failure).

RTX-ASAP. A retransmission of a data chunk is sent to any destination for which the sender has *cwnd* space available at the time the retransmission needs to be sent. If the sender has available *cwnd* space for multiple destinations, one is chosen randomly.

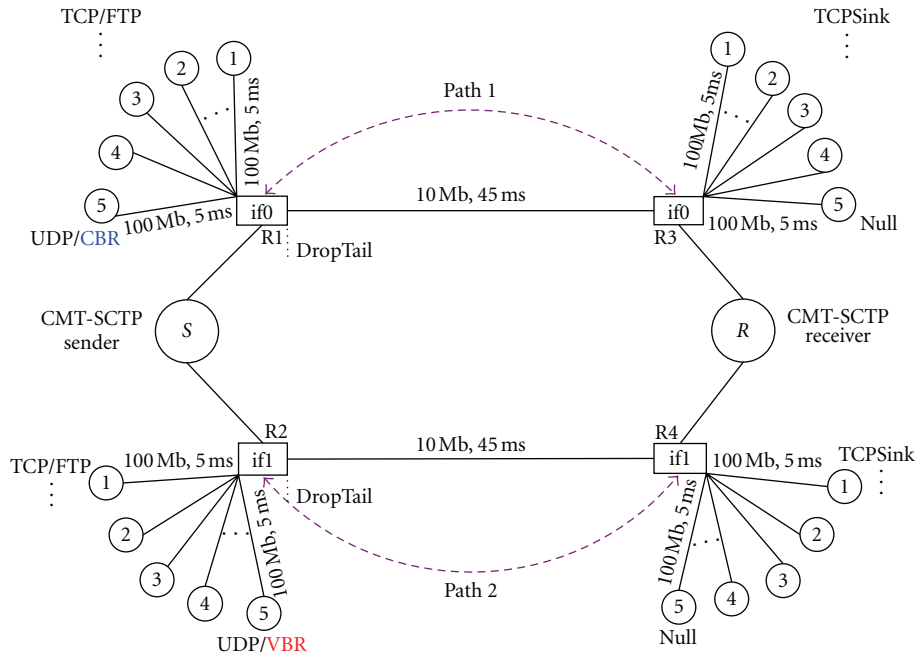


FIGURE 1: Simulation topology for studying the impact of background traffic.

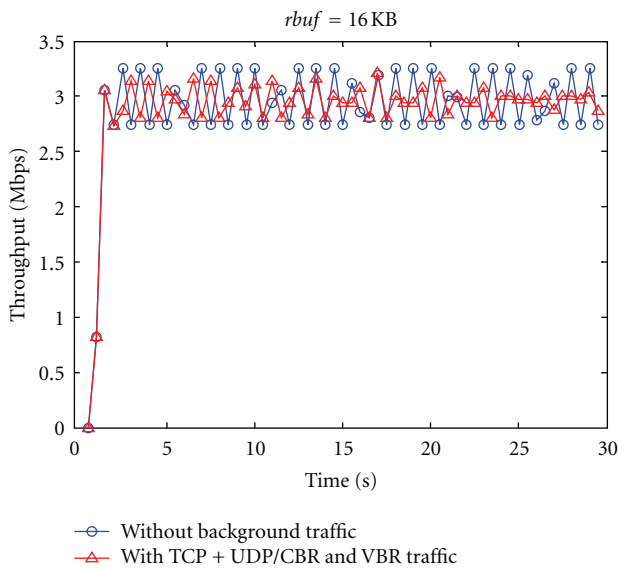


FIGURE 2: Comparison with $rbuf = 16$ KB.

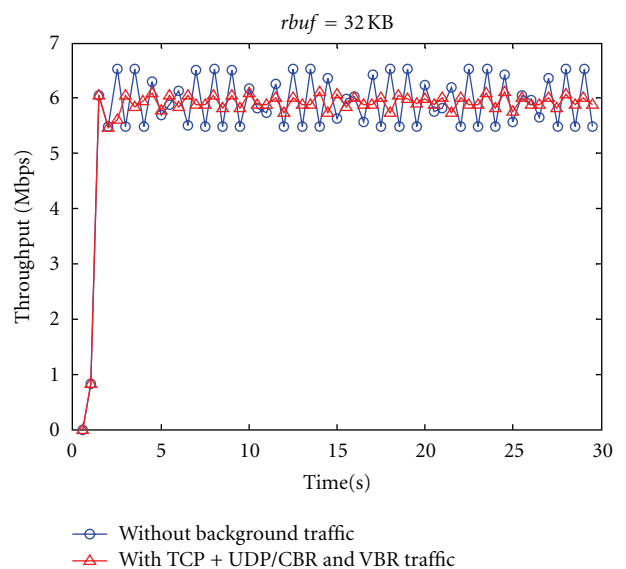


FIGURE 3: Comparison with $rbuf = 32$ KB.

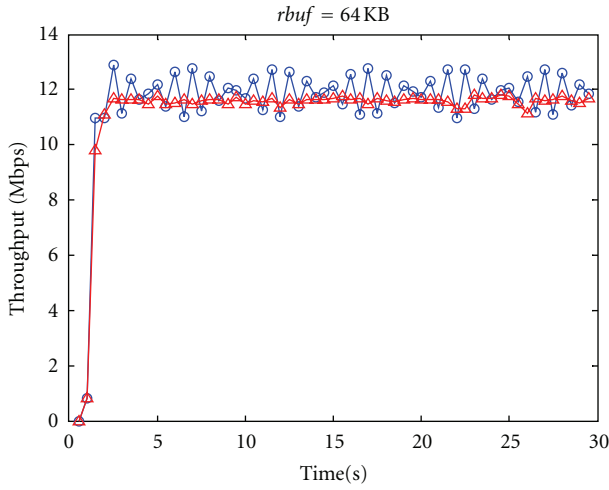
RTX-LOSSRATE. A retransmission of a data chunk is sent to the destination with the lowest loss rate path. If multiple destinations have the same loss rate, one is selected randomly.

RTX-CWND. A retransmission of a data chunk is sent to the destination for which the sender has the largest *cwnd*. A tie is broken randomly.

RTX-SSTHRESH. A retransmission of a data chunk is sent to the destination for which the sender has the largest *sssthresh*. A tie is broken randomly.

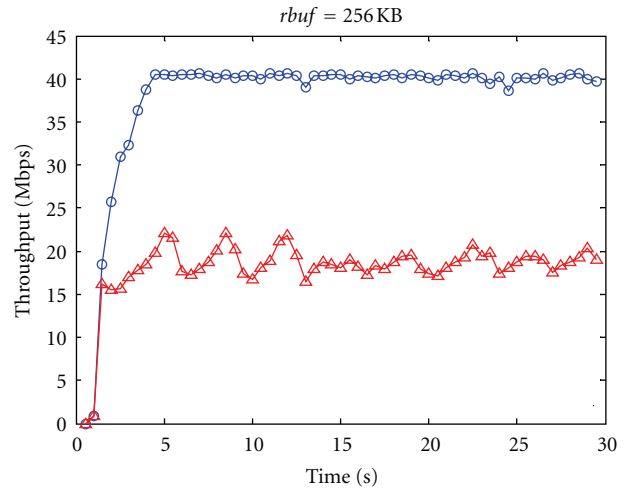
However, according to RFC4460, only the *RTX-CWND* and *RTX-SSTHRESH* are recommended retransmission policies, and the others are just for experimental sake. Moreover, the *RTX-CWND* is recommended as the default retransmission strategy since it can present the best performance [12].

4.2. RTX-CSI Description. As mentioned in Section 1, all of existing retransmission algorithms do not take the impact of background traffic into account. To fix this issue, we consider



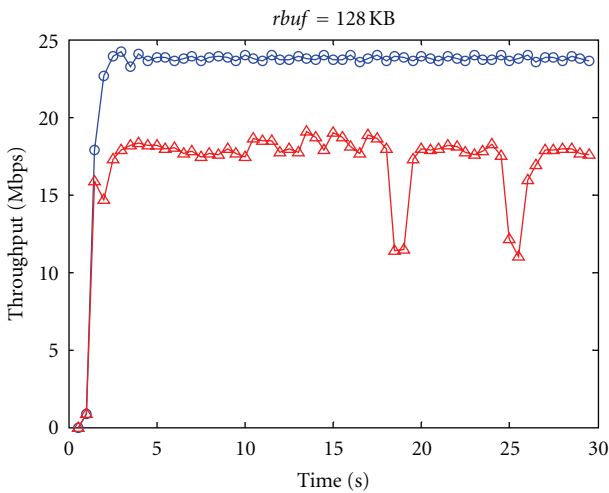
—○— Without background traffic
—△— With TCP + UDP/CBR and VBR traffic

FIGURE 4: Comparison with $rbuf = 64$ KB.



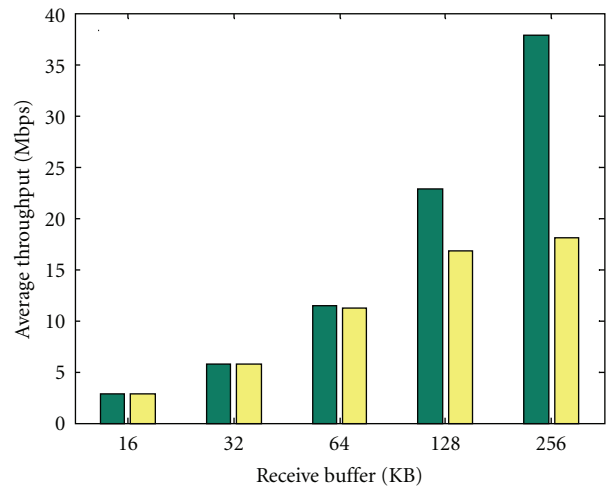
—○— Without background traffic
—△— With TCP + UDP/CBR and VBR traffic

FIGURE 6: Comparison with $rbuf = 256$ KB.



—○— Without background traffic
—△— With TCP + UDP/CBR and VBR traffic

FIGURE 5: Comparison with $rbuf = 128$ KB.



■ Without background traffic
■ With TCP + UDP/CBR and VBR traffic

FIGURE 7: Comparison on average throughput.

the localness nature of background flow [19], that is, coming packets will belong to the flow as previously arrived ones in a short period. Correspondingly, the side effect on CMT-SCTP caused by background flow will be the same in short period. As countermeasure of the localness nature of background traffic, paths' previous states should be considered during designing of retransmission algorithm. Therefore, we take paths' previous states into account to design an improved retransmission algorithm named *RTX_CSI* which consists of four more reasonable paths' conditions during selecting retransmission destination. *RTX_CSI* follows the below steps to select candidate path for data retransmission.

- (1) A retransmission is sent to the destination that has the largest *cwnd*;

- (2) if more than one destination has the largest *cwnd*, then a retransmission is sent to the one that has the largest *ssthresh* value;
- (3) if there are more than one destination has the largest *ssthresh* value, then a retransmission is sent to the one that has the lowest *Time Out Records (tor)* in specified time span (denoted as τ);
- (4) if more than one destination has lowest timeout records in specified timeslot, then a retransmission is sent to the one that has the largest interval $t_{interval}$ ($t_{interval}$ stands for the interval between the last timeout's time and current time);

Definition:

d_i : the i_{th} path in the destination list of core node
 d_{list} : the active destination list of core node
 d_i^{cwnd} : the *cwnd* value of the i_{th} path
 d_i^{ssth} : the *ssthresh* value of the i_{th} path
 d_i^{tor} : total timeouts on d_i in timeslot τ
 $d^{interval}$: the $t_{interval}$ of the i_{th} path
 d_{rxDest} : the destination selected to retransmit loss data
 Once having packet to resend
 1: **for** each destination d_i **do** //scan all paths' condition
 2: **if** status of d_i == ACTIVE **then**
 3: put d_i into d_{list} ; //ignore the inactive paths
 4: **end if**
 5: **end for**
 6: set $d_s^{cwnd} = d_{list(0)}^{cwnd}$, $d_t^{ssth} = d_{list(0)}^{ssth}$, $d_w^{interval} = d_{list(0)}^{interval}$;
 7: **for** ($i = 1$, $i <= \text{count}(d_{list})$, $i++$) **do**
 //select the destination which has largest *cwnd* value in loop
 8: **if** ($d_s^{cwnd} < d_{list(i)}^{cwnd}$) **then**
 9: set $s = i$; set $d_s^{cwnd} = d_{list(i)}^{cwnd}$;
 10: **end if**
 //select the destination which has largest *ssthresh* value in loop
 11: **if** ($d_t^{ssth} < d_{list(i)}^{ssth}$) **then**
 12: set $t = i$; set $d_t^{ssth} = d_{list(i)}^{ssth}$;
 13: **end if**
 //select the destination which has lowest *tor* value in loop
 14: **if** ($d_t^{tor} > d_{list(i)}^{tor}$) **then**
 15: set $t = i$; set $d_t^{tor} = d_{list(i)}^{tor}$;
 16: **end if**
 //select the destination which has largest $t_{interval}$ value in loop
 17: **if** ($d_w^{interval} < d_{list(i)}^{interval}$) **then**
 18: set $w = i$; set $d_w^{interval} = d_{list(i)}^{interval}$;
 19: **end if**
 20: **end for**
 21: **if** !(($d_j \in d_{list}$) && ($d_j^{cwnd} == d_s^{cwnd}$) && ($j < > s$)) **then**
 //path with largest *cwnd* is set as retransmission destination
 22: set $d_{rxDest} = d_s$;
 23: **else if** !(($d_j \in d_{list}$) && ($d_j^{ssth} == d_t^{ssth}$) && ($j < > t$)) **then**
 //path with largest *ssthresh* is set as retransmission destination
 24: set $d_{rxDest} = d_t$; //
 25: **else if** !(($d_j \in d_{list}$) && ($d_j^{tor} == d_t^{tor}$) && ($j < > t$)) **then**
 //path with lowest *tor* is set as retransmission destination
 26: set $d_{rxDest} = d_t$;
 27: **else if** !(($d_j \in d_{list}$) && ($d_j^{interval} == d_w^{interval}$) && ($j < > w$))
 //path with largest $t_{interval}$ is set as retransmission destination
 28: set $d_{rxDest} = d_w$;
 //a tie will be broken by random selection
 29: **else** set d_{rxDest} with random d_i ($d_i \in d_{list}$);
 30: **end if**
 let loss data be retransmitted on d_{rxDest}

ALGORITHM 1: RTX_CSI Algorithm.

- (5) if multiple destinations have the largest $t_{interval}$, then a tie will be broken by random selection.

The details of RTX_CSI algorithm are shown in Algorithm 1.

4.3. *Simulation Topology Setup.* In this section, we adopt the average throughput as the metric in our experiments. Figure 10 shows the simulation topology. Related simulation parameters are set the same as that mentioned in Section 3.

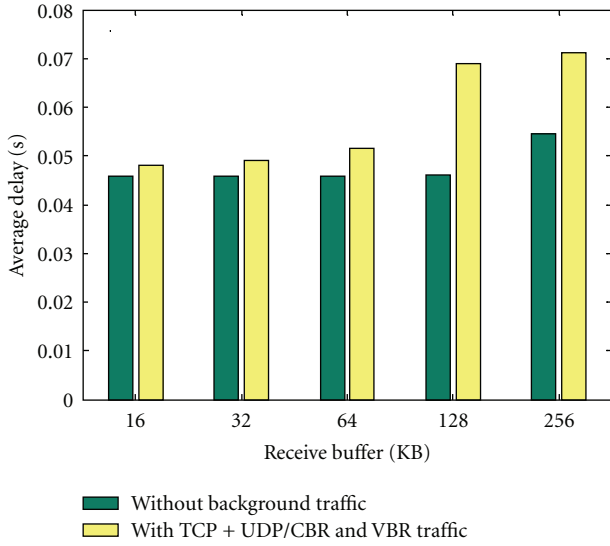


FIGURE 8: Comparison on average delay.

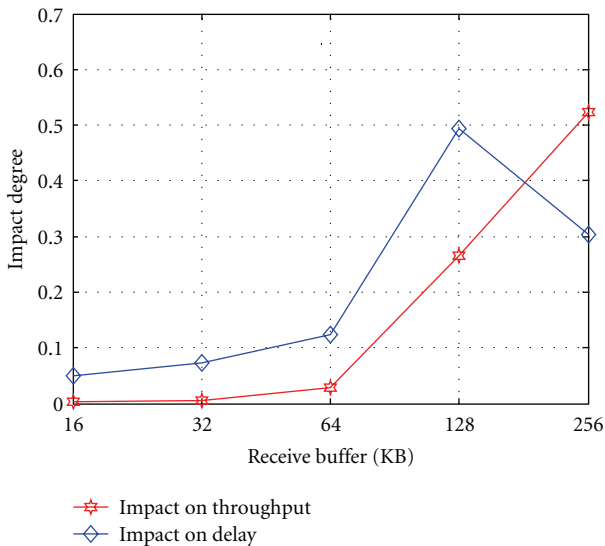


FIGURE 9: Impact degree on average throughput and delay.

We perform two experimental scenarios named Case 1 and Case 2 are examined as follows to study the performance of *RTX_CSI*. In our experiments, the *rbuf* is set to 16 KB, 32 KB, 64 KB, 128 KB, and 256 KB, respectively, and τ is set to 30 s.

Case 1. The loss rate on *TCP + UDP/VBR* traffic path is always kept at 1%, and on *TCP + UDP/CBR* traffic path, it is varied from 1% to 10%;

Case 2. The loss rate on *TCP + UDP/VBR* traffic path is varied from 1% to 10%, and on *TCP + UDP/CBR* traffic path, it is always kept at 1%.

4.4. Performance Evaluation. As mentioned above, according to RFC4460, only the *RTX-CWND* and *RTX-SSTHRESH*

are recommended retransmission policy. So we compare the performance of *RTX_CSI* with the *RTX-CWND* and *RTX-SSTHRESH*.

To compare conveniently, we use (2) illustrated below to express the advantage in terms of average throughput achieved by algorithm *A* (denoted as A_{adv}) over algorithm *B*.

$$A_{adv} = \frac{A_{aveTh} - B_{aveTh}}{B_{aveTh}} \times 100\%, \quad (2)$$

where A_{aveTh} and B_{aveTh} are on behalf of average throughput achieved by retransmission algorithm *A* and *B*, respectively.

Firstly, we evaluate the performance of *RTX_CSI*, *RTX-CWND*, and *RTX-SSTHRESH* with experimental condition illustrated in Case 1; Figure 11 shows the performance of the three algorithms under different *rbuf*. We can get that the average throughput achieved by the *RTX-CWND*, *RTX-SSTHRESH*, and proposed *RTX_CSI* will rise with the increase of the *rbuf*. But when *rbuf* is set to more than 64 KB, the increments of average throughput are reduced whichever the three algorithms is employed. This phenomenon verifies again that the background traffic presents more serious side effects on performance of CMT-SCTP as larger *rbuf* is used. However, the *RTX_CSI* performs the best performance in the three algorithms, the *RTX-CWND* comes next, and the *RTX-SSTHRESH* presents the worst behavior.

For Case 1, per calculated by (2), detailed comparison on the average throughput can be pointed out as follows.

- (1) Comparing to the *RTX-SSTHRESH*, the *RTX-CWND* achieves a more advantage about -4.42%, 4.94%, 18.2%, 56.39%, and 55.15% when *rbuf* is 16 KB, 32 KB, 64 KB, 128 KB, and 256 KB, respectively. So, it can be concluded that the *RTX-CWND* can also present better performance over the *RTX-SSTHRESH* [12] even under background traffic condition (Case 1).
- (2) Comparing to the *RTX-CWND*, the proposed *RTX_CSI* achieves more benefits about 1.14%, 0.56%, and 1.62% when *rbuf* is set to 16 KB, 32 KB, and 64 KB, respectively. Since larger *rbuf* leads to less packet loss; therefore, when the *rbuf* is set to 128 KB and 256 KB, the proposed *RTX_CSI* presents same performance in terms of throughput as the *RTX-CWND* algorithm.

Secondly, we compare the performance of the *RTX-CWND* and *RTX-SSTHRESH* with the proposed *RTX_CSI* under different *rbuf* with designed experimental scenario addressed in Case 2. As it shown in Figure 12, the average throughput achieved by the three algorithms will rise with the increase of *rbuf*. But with same reason mentioned in Case 1, when *rbuf* is set to more than 64 KB, the increments of average throughput are reduced whichever the three algorithms are employed. In this case, the *RTX_CSI* still performs the best performance. But different from conclusion addressed in [12] and Case 1, only if *rbuf* is larger than 64 KB, the *RTX-CWND* presents a better performance than the *RTX-SSTHRESH*.

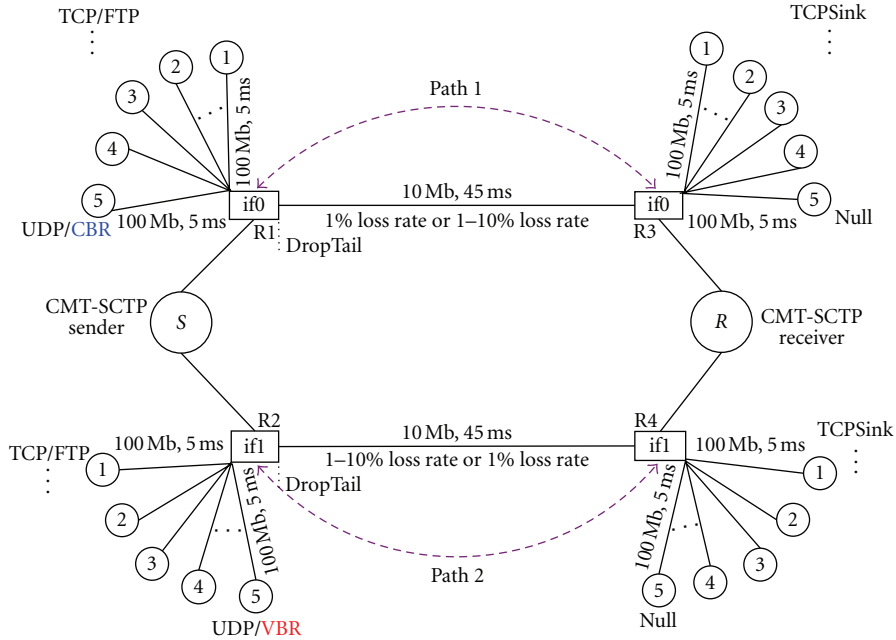


FIGURE 10: Simulation topology for evaluating RTX_CSI .

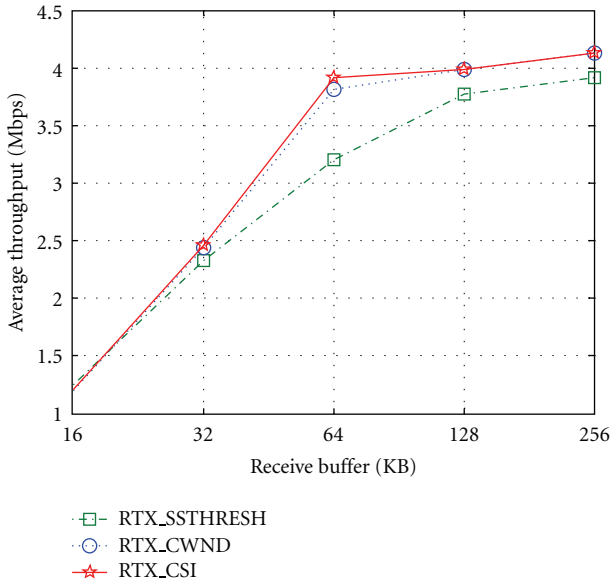


FIGURE 11: Path 1 loss rate is varied from 1–10%, Path 2 is always kept at 1%.

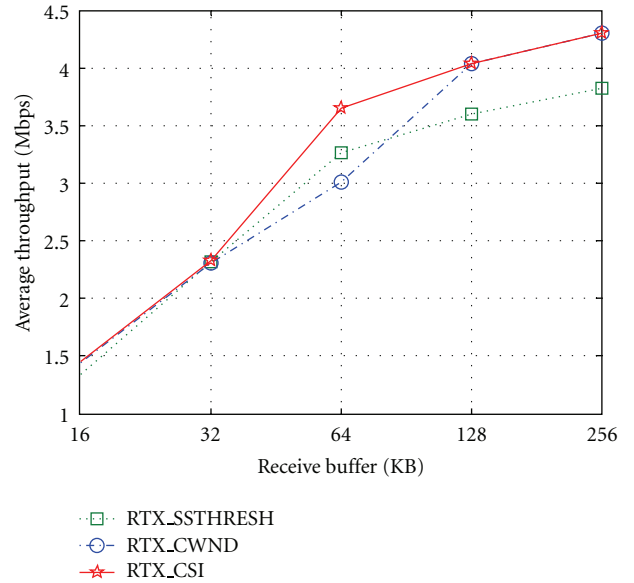


FIGURE 12: Path 2 loss rate is varied from 1–10%, Path 1 is always kept at 1%.

Likewise, for Case 2, detailed comparisons are pointed out per calculated by (2).

- (1) Comparing to the $RTX_SSTHRESH$, the RTX_CWND achieves a more advantage about 7.26%, -0.15%, -8.02%, 12.05%, and 12.57% when $rbuf$ is 16 KB, 32 KB, 64 KB, 128 KB, and 256 KB, respectively. The comparison results are different from the conclusion mentioned in [12], in our experiment (Case 2), comparing to the $RTX_SSTHRESH$, the RTX_CWND

only can achieve obvious more advantages in terms of average throughput when $rbuf$ is set to the value which is larger than the default $rbuf$ (64 KB). The reason may be that when enormous of variable bite rate packets are lost, their retransmissions not only deteriorate the path’s quality but also enlarge the unpredictability of path’s condition. Those unexpected conditions make the CMT-SCTP sender cannot tune its congestion window accurately. But

when the $rbuf$ is set to larger one (more than 64 KB), the sender can correct its congestion window since lots of packets can be received and acknowledged timely thus, the RTX_CWND can outperform the $RTX_SSTHRESH$. This phenomenon further verifies the proposed RTX_CSI more reasonable with considering the nature of background traffic. Our future work will investigate the reason in detail.

- (2) Comparing to the RTX_CWND , the proposed RTX_CSI achieves more benefits about 0.47%, 0.57%, and 17.56% when $rbuf$ is set to 16 KB, 32 KB, and 64 KB, respectively. As Case 1, when the $rbuf$ is set to 128 KB and 256 KB, the proposed RTX_CSI presents same performance in terms of throughput as the RTX_CWND with the reason that larger $rbuf$ leads to less packet loss.

From experiments and analysis for Case 1 and Case 2 respectively, we can conclude that the proposed RTX_CSI algorithm can achieve a better performance than the *existing retransmission algorithm*, especially for $rbuf$ such as 16 KB, 32 KB, and 64 KB which are used commonly. The reason RTX_CSI can achieve a more advantage over the two *existing retransmission algorithm* is that the RTX_CSI can select a more efficient path as the retransmission destination with considering more reasonable rules such as paths' $cwnd$, $ssthresh$ value, and historical states to meet some known problems like congestion and packet loss caused by background traffic.

5. Conclusions

In this paper, we designed realistic simulation topologies and examined the performance of CMT-SCTP in terms of throughput, end-to-to packet delay by considering reasonable background traffic. We discussed *how* the presence of background traffic affects the performance of CMT-SCTP in detail, which are generally ignored by most current researchers.

Based on above work, we proposed an improved retransmission algorithm called RTX_CSI for CMT-SCTP. RTX_CSI takes background traffic into account and considers paths' comprehensive characteristics during selecting retransmission destination to meet the localness nature of background traffic. The simulation results show RTX_CSI achieves better efficiency than CMT-SCTP's original retransmission algorithm. So the proposed RTX_CSI can be employed to improve the users' experience of quality for multimedia streaming service when CMT-SCTP is used for the multimedia transport protocol.

Acknowledgments

This work is partially supported by the National High-Tech Research and Development Program of China (863) under Grant no. 2011AA010701, in part by the National Natural Science Foundation of China (NSFC) under Grants nos. 61001122 and 61003283 and the Beijing Natural Science Foundation under Grant no. 4102064, in part by the

Fundamental Research Funds for the Central Universities under Grants nos. 2012RC0603 and 2011RC0507, and in part by the Natural Science Foundation of Jiangsu Province under Grant no. BK2011171.

References

- [1] C. Xu, G. M. Muntean, E. Fallon, and A. Hanley, "Distributed storage-assisted data-driven overlay network for P2P VoD services," *IEEE Transactions on Broadcasting*, vol. 55, no. 1, pp. 1–10, 2009.
- [2] Z. Liu, C. Wu, S. Zhao, and B. Li, "UUSee: large-scale operational on-demand streaming with random network coding," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '10)*, San Diego, Calif, USA, March 2010.
- [3] C. Xu, E. Fallon, Q. Yuansong, Z. Lujie, and M. Gabriel-Miro, "Performance evaluation of multimedia content distribution over multi-homed wireless networks," *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 204–215, 2011.
- [4] C. Xu, E. Fallon, M. Gabriel-Miro, X. Li, and A. Hanley, "Performance evaluation of distributing real-time video over concurrent multipath," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '09)*, Budapest, Hungary, April 2009.
- [5] K. Zheng, M. Liu, Z. C. Li, and G. Xu, "SHOP: an integrated scheme for SCTP handover optimization in multihomed environments," in *Proceedings of the IEEE Global Telecommunications Conference*, pp. 1–5, New Orleans, La, USA, December 2008.
- [6] R. Stewart, Q. Xie, K. Morneault et al., "Stream control transmission protocol," IETF RFC 2960, October 2000.
- [7] P. Natarajan, F. Baker, P. D. Amer, and J. T. Leighton, "SCTP: what, why, and how," *IEEE Internet Computing*, vol. 13, no. 5, pp. 81–85, 2009.
- [8] Y. Wang, R. Injong, and H. Sangtae, "Augment SCTP multi-streaming with pluggable scheduling," in *Proceedings of the 30th IEEE International Conference on Computer Communications Workshops*, pp. 810–815, Shanghai, China, April 2011.
- [9] J. R. Iyengar, P. D. Amer, and R. Stewart, "Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths," *IEEE/ACM Transactions on Networking*, vol. 14, no. 5, pp. 951–964, 2006.
- [10] T. Stegel, J. Sterle, U. Sedlar, J. Bešter, and A. Kos, "SCTP multihoming provisioning in converged IP-based multimedia environment," *Computer Communications*, vol. 33, no. 14, pp. 1725–1735, 2010.
- [11] C. M. Huang and M. S. Lin, "Multimedia streaming using partially reliable concurrent multipath transfer for multihomed networks," *IET Communications*, vol. 5, no. 5, pp. 587–597, 2011.
- [12] J. R. Iyengar, P. D. Amer, and R. Stewart, "Retransmission policies for concurrent multipath transfer using SCTP multihoming," in *Proceedings of the 12th IEEE International Conference on Networks (ICON '04)*, pp. 713–719, Singapore, November 2004.
- [13] J. R. Iyengar, P. D. Amer, and R. Stewart, "Performance implications of a bounded receive buffer in concurrent multipath transfer," Tech. Rep., CIS Department, University of Delaware.
- [14] J. Liao, J. Wang, and X. Zhu, "cmpSCTP: an extension of SCTP to support concurrent multi-path transfer," in *Proceedings of*

- the IEEE International Conference on Communications*, pp. 5762–5766, Beijing, China, 2008.
- [15] L. Budzisz, R. Ferrús, F. Casadevall, and P. Amer, “On concurrent multipath transfer in SCTP-based handover scenarios,” in *Proceedings of the IEEE International Conference on Communications*, pp. 1–6, Dresden, Germany, June 2009.
 - [16] J. M. Liu, H. X. Zou, J. X. Dou, and Y. Gao, “Reducing receive buffer blocking in concurrent multipath transfer,” in *Proceedings of the IEEE International Conference on Circuits and Systems for Communications (ICCSC '08)*, Shanghai, China, May 2008.
 - [17] P. Barford and M. Crovella, “Generating representative web workloads for network and server performance evaluation,” in *Proceedings of ACM SIGMETRICS*, pp. 151–160, Madison, Wis, USA, June 1998.
 - [18] S. Floyd and V. Paxson, “Difficulties in simulating the Internet,” *IEEE/ACM Transactions on Networking*, vol. 9, no. 4, pp. 392–403, 2001.
 - [19] S. Floyd and E. Kohler, “Internet research needs better models,” *ACM Computer Communications Review*, vol. 33, no. 1, pp. 29–34, 2003.
 - [20] S. Ha, L. Le, I. Rhee, and L. Xu, “Impact of background traffic on performance of high-speed TCP variant protocols,” *Computer Networks*, vol. 51, no. 7, pp. 1748–1762, 2007.
 - [21] The Network Simulator—ns-2, <http://www.isi.edu/nsnam/ns/>.
 - [22] M. Fomenkov, K. Keys, D. Moore, and K. claffy, “Longitudinal study of Internet traffic in 1998–2003,” in *Proceedings of the Winter International Symposium on Information and Communication Technologies (WISICT '04)*, pp. 1–6, Cancun, Mexico, January 2004.
 - [23] <http://www.isi.edu/nsnam/archive/ns-users/webarch/2001/msg05051.html>.
 - [24] A. Caro, P. Amer, and J. Iyengar, “Retransmission policies with transport layer multihoming,” in *Proceedings of the 11th IEEE International Conference on Networks*, pp. 255–260, Sydney, Australia, November 2003.
 - [25] P. Natarajan, J. R. Iyengar, P. D. Amer, and R. Stewart, “Concurrent multipath transfer using transport layer multihoming: performance under network failures,” in *Proceedings of the Military Communications Conference (MILCOM '06)*, pp. 1–7, Washington, DC, USA, 2006.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

