Evaluation of Performance of Background Traffic-based CMT-SCTP with Active Queue Management Algorithms

Meifang Shen[1], Yuanlong Cao[2], Mingchuan Zhang[2,3], Kaixuan Wang[2,4], Mingming Yu[5]

[1] Talend Beijng Technology Co., Ltd.
Beijing, 100004, China

[2] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
Beijing, 100876, China

[3] Electronic & Information Engineering College, Henan University of Science and Technology
Luoyang 471003, China

[4] School of Information Management, Shanxi University of Finance and Economics
Taiyuan 030006, China

[5] Commercial Aircraft Corporation of China Ltd.
Shanghai 200436, China

Abstract

Existing researches on performance analysis of SCTP's Concurrent Multipath Transfer (CMT-SCTP) usually use DropTail algorithm as queue management algorithm without considering the impact of the background traffic. However, the background traffic of realistic network environments has an important impact on the QoS of SCTP. Besides, more and more Active Queue Management (AQM) algorithms have been proposed as a router-based mechanism for early congestion detection to keep the stability of the whole network. This paper investigates the effect of background traffic on the performance of CMT-SCTP, and evaluates the performance of CMT-SCTP under two realistic simulation topologies with reasonable background traffic and different AQM algorithms in NS-2. The simulation results show that: 1) the performance of CMT-SCTP depends on characteristic of background traffic; and 2) the different AQM algorithms used as queue management algorithm under same background traffic have the different effects. Finally, this paper summarizes the proposals to satisfy the QoS requirements in terms of throughput, end-to-end packet delay and loss rate. Since CMT-PF2 is recommended by RFC4960 but without taking impact of cross traffic into account. In the second part, we use the most promising topology which meets the developing network and base on result of analysis mentioned in the first part to analyze the performance CMT-PF1/2/3/4 played respectively, in this part, the most common scenario, symmetric CMT-SCTP, is adopted and CMT-PF algorithm is turned on. A conclusion had been nailed down that, CMT-PF3 can get more advantage in terms of average throughput than CMT-PF2 which is recommended by RFC4960. Per reasonable analyzing, we lastly recommend a more reasonable resolution for realistic network in order to reaching more satisfied QoS.

Keywords- *CMT-SCTP; background traffic; TCP traffic; UDP/CBR traffic; UDP/VBR traffic; AQM algorithm; NS2;CMF-PF*

1. Introduction

The Stream Control Transmission Protocol (SCTP) [1] has been proposed and standardized by the Internet Engineering Task Force (IETF) in order to effectively utilize the multi-home environment and increase availability and for the purpose of transporting of real time signaling over IP networks. For many years SS7 has been the only bearer for the signaling traffic in telecommunication networks [2]. Some important features of SCTP are briefly addressed as follows: 1) Multi-homing. The destination nodes can be reached under the several IP addresses (multi-homed). In SCTP both sides of the association provides multiple IP addresses combined with a single SCTP port number [3]. 2) Multi-Streaming 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 the other, which gives an advantage to SCTP over others protocol (e.g. TCP) and it achieves a cumulative throughput [4].

As an improved version of SCTP, Concurrent Multipath Transfer (CMT) [5] 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 host via two or more end-to-end paths, and it is used between multi-homed source and destination hosts to increase an application's throughputs. Moreover, a CMT sender can maintain more accurate information (such as available bandwidth, loss rate, and RTT etc.) about all paths, since new data are being sent to all destinations concurrently. This information allows a CMT sender to better decide where to retransmit once data loss occurred [6-8].

Most of researches in CMT-SCTP lack of taking background flows into account and only use DropTail algorithm as congestion control algorithm during performance study. Actually, according to the Internet survey, the TCP traffic is about 80%~83% and UDP traffic is about 17%~20% [9]. So, background traffic should be taken into account during studying of CMT-SCTP's performance.

On the other hand, the IETF have proposed active queue management (AQM) as the mechanism for detecting congestion inside the network. Further, they have strongly recommended the deployment of AQM in routers as a measure to preserve and improve WAN performance. In practice, most of the routers being deployed use simplistic DropTail algorithm,

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

which is simple to be implemented with minimal computation overhead, but provides unsatisfactory performance.

To solve the problem, Random Early Detection (RED) [10], an active queue management algorithm was recommended by the IETF for deployment in IP routers/networks [11]. The basic idea behind an active queue management algorithm is to convey congestion notification early enough to the senders, so that senders are able to reduce the transmission rates before the queue overflows and any sustained packet loss occurs. It is now widely accepted that a RED-controlled queue performs better than a drop-tail queue. However, the inherent design of RED makes it difficult to parameterize RED queues to give good performance under different network scenarios. Several algorithms, like Flow Random Early Drop (FRED) [12], BLUE [13], Random Exponential Marking (REM) [14] and Proportional-Integral control (PI) [15] discard packets with a load-dependent probability whenever the queue buffer in a router appears to be congested.

Even though the performance of CMT-SCTP adopted in variety of attractive services has been studied widely. However, existing evaluation works [16-20] of CMT-SCTP only adopt the simplistic DropTail algorithm and seldom consider which AQM algorithm can get better performance. In this paper, we use combination of TCP traffic, UDP/CBR traffic and UDP/VBR traffic as cross background traffic to design a realistic-like simulation topology in Network Simulation (NS2) [21], and then analyze the performance of network redundancy in CMT-SCTP with several typical AQM algorithms (DropTail, RED, FRED, BLUE, REM and PI). Two goals will be reached in this paper: 1) which AQM algorithm can get better performance in CMT-SCTP with specified cross traffic; and 2) how affect the specified cross traffic plays in CMT-SCTP with different AQM algorithm. Throughput, delay and loss rate is used as metric in our experimental.

2. Preliminary work

In this section, we firstly give out a guide about how to load VBR traffic generator into NS2, and then describe the installation guide of new two algorithms (FRED and BLUE) since they are not included into NS2 as well.

2.1 VBR traffic generator loading

Since NS2 still cannot support VBR traffic, in order to enabling 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 [22]. The default values for VBR traffic are set by following Table 1.Table 1 shows parameters set for VBR traffic in the *ns-default.tcl* which will be used in our experimental.

2.2 FRED and BLUE algorithm modules loading

In order to getting FRED and BLUE algorithm supported by NS2, we need to load the two algorithms using follow steps:

1) Add the two algorithm's .h and .cc to *ns-2/queue*;
2) Edit *ns-default.tcl*, add default parameters for the two new algorithm (refer to Table 2);
3) update *makefile* with including *queue/fred*.o and *queue/blue.o*;
4) Remake NS2.

Table 2 shows partial parameters configuration for the FRED and BLUE algorithm used in our experimental. Others algorithm (DropTail, RED, REM and PI) use default parameters which are given in NS2.

Table 1 Parameter Configuration for vbr

Variable	Value
*Application/Traffic/VBR set rate_*	448Kb
*Application/Traffic/VBR set random_*	0
*Application/Traffic/VBR set maxpkts_*	268435456
*Application/Traffic/VBR set maxSize_*	200
*Application/Traffic/VBR set minSize_*	100
*Application/Traffic/VBR set intervaltime_*	200

Table 2 Parameter Configuration

	Parameters	Default Values
BLUE	Minimum threshold	10
	Maximum threshold	30
	Queue weight	0.002
	Mean Packet Size	500
	Increase drop probability	0.02
	Decrease drop probability	0.002
FRED	Minimum threshold	10
	Maximum threshold	30
	Queue weight	0.002
	Mean Packet Size	500
	Increase drop probability	0.02
	Decrease drop probability	0.002

The rest of paper is organized as follows. Section 3 describes a more realistic simulation topology design for network redundancy. Section 4 analyzes the effect of the certain background traffic. Finally, section 5 concludes this paper.

3. Simulations without CMT-PF and Loss

3.1 Background traffic design

To get a realistic-like network simulation scenario, according to the Internet survey mentioned in section 1, TCP traffic on Internet is about 80%-83%, and UDP traffic is about 17%-20%. So, we take combinations of TCP traffic, UDP/CBR traffic and UDP/VBR traffic as network background traffic in order to implementing a more realistic simulation topology. In our

IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No 2, November 2012
ISSN (Online): 1694-0814
www.IJCSI.org

experimental, we use tow paths in symmetric/asymmetric CMT-SCTP to transmit data:

1) Both paths with TCP traffic and UDP/CBR traffic as background traffic (TCP: UDP/CBR is 4:1) which is represented by *TCP+UDP/CBR* in below.
2) Both paths with TCP traffic and UDP/VBR traffic as background traffic (TCP:UDP/VBR is 4:1) which is represented by *TCP+UDP/VBR* in below.
3) 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&VBR* in below.

3.2 Simulation topology setup

We adpot both symmetric and asymmetric two paths to evaluate the performance of CMT-SCTP (as shown in Fig. 1 and 2). In the two dual dumbbell topology, each router node R1-R4 (R1-R3 in Fig. 2) is connected to five edge nodes. The S and R stands for CMT-SCTP's sender and receiver, respectively. The other edge nodes are single homed for the background traffic at the routers. The propagation delay between the edge nodes and routers is set to 5ms with 100Mb of bandwidth [23-24]. Each single homed edge node is attached with a traffic generator, introducing cross traffic with 80% (four nodes on each edge) of TCP traffic and 20% (one node on each edge). R1 and R2 are bottleneck for the whole traffic and their buffer size is set to 64Kb. The propagation delay between dual homed interfaces is set to 25ms. The two paths between the end points are fully separated. The path between R1 to R4 (in aysmmetric CMT-SCTP it's R1 to R3) is set as primary path, and CMT-SCTP uses concurrent multipath transfer on both paths. After 0.5 seconds of simulation CMT-SCTP Sender starts initiating association with receiver CMT-SCTP. At 1.0 seconds other cross traffic that is TCP and UDP is started in the network. Simulation time is 30 seconds.

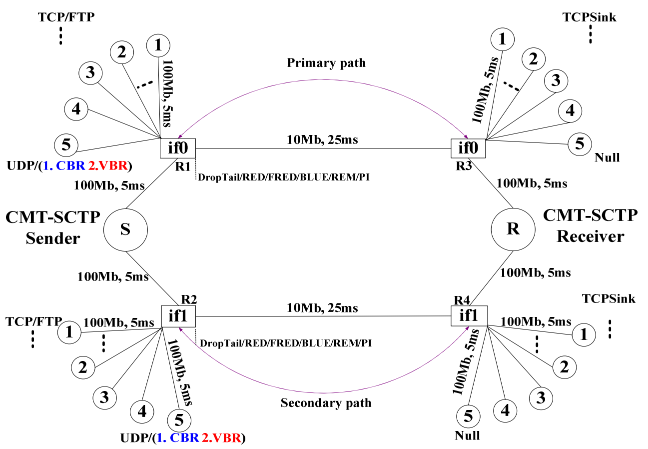

Figure 1. Symmetric CMT-SCTP simulation topology

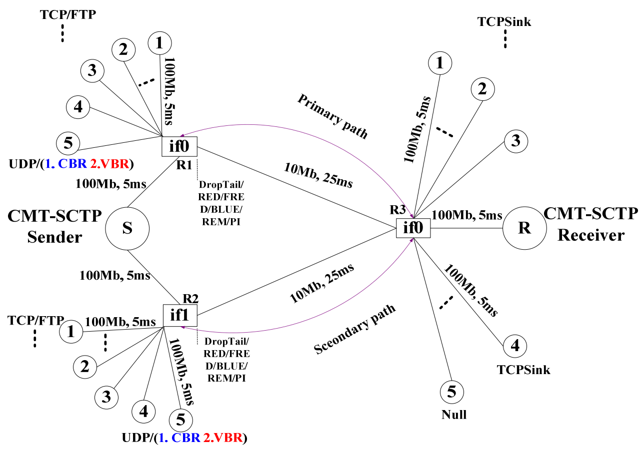

Figure 2. Asymmetric CMT-SCTP simulation topology

4. Performance Analysis

To analyze how the effects the specified cross traffic plays in CMT-SCTP with different AQM algorithm, we define an Impact Factor (*if*) equation:

$$if_i = \frac{|QoS_{x_i} - QoS_{y_i}|}{QoS_{x_i}} \quad (1)$$

Where *i* stands for which AQM algorithm adopted; *x*, *y* for average throughput and delay respectively; $QoSx_i$ is on behalf of average throughput or average delay created by CMT-SCTP with specified AQM algorithm under non- background traffic condition; $QoSy_i$ for average throughput or average delay created by CMT-SCTP with AQM algorithm under specified background traffic; if_i stands for impact factor arisen by certain AQM algorithm under specified background traffic.

Since loss rate may be 0 in CMT-SCTP under no background traffic. Thus, we define an Incremental Analysis (*IA*) equation as below:

$$IA_i = lossRate_m - lossRate_n \quad (2)$$

Where *i* stands for which AQM algorithm adopted as well; $lossRate_m$ is on behalf of loss rate occurred by CMT-SCTP with *m* algorithm under certain background traffic. $lossRate_m$ is for loss rate occurred by CMT-SCTP with *n* algorithm under no background traffic condition. IA_i represents for incremental in terms of loss rate between certain background traffic and no background.

To figure out which AQM algorithm adopted for certain background traffic can get CMT-SCTP get better QoS for CMT-SCTP than others AQM algorithm listed in this paper, we define a expression as below:

$$\textbf{if } \beta \textbf{ then } \text{set } i \text{ -> } Optimized(QoS) \quad (3)$$

Expression (3) means that if background traffic is β (β stands for TCP+UDP/CBR, TCP+UDP/VBR or TCP+UDP/CBR&VBR), then set *i* (*i* is DropTail, RED, FRED, BLUE, REM or PI) as Queue management will get a more satisfied *QoS* for CMT-SCTP in terms of average throughput, average delay or loss rate.

4.1 Test for symmetric CMT-SCTP

Using simulation topology shown in Fig. 1, we use AQM algorithm (DropTail, RED, FRED, BLUE, REM and PI) to get average throughput and delay under certain background traffic condition which shown in Fig. 3 and 5, respectively. Per calculating by equation (1), a corresponding *if* statistic is shown in Fig. 4 and 6, respectively.

1）In terms of average throughput. As shown in Fig. 4, when background traffic is '*TCP+UDP/CBR*', comparison with average throughput in CMT-SCTP without background traffic, $if_{DropTail} \approx 0.5816$; $if_{RED} \approx 0.5523$; $if_{FRED} \approx 0.5127$; $if_{BLUE} \approx 0.5900$; $if_{REM} \approx 0.4913$; and $if_{PI} \approx 0.5374$.

When background traffic is '*TCP+UDP/VBR*', comparison with average throughput in CMT-SCTP without background traffic, $if_{DropTail} \approx 0.3875$; $if_{RED} \approx 0.5031$; $if_{FRED} \approx 0.4097$; $if_{BLUE} \approx 0.3863$; $if_{REM} \approx 0.4025$; and $if_{PI} \approx 0.4181$.

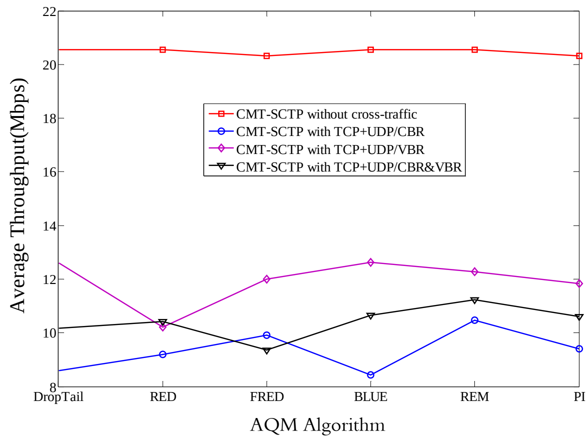

Figure 3 Comparison of average throughput on symmetric CMT-SCTP with different cross traffic

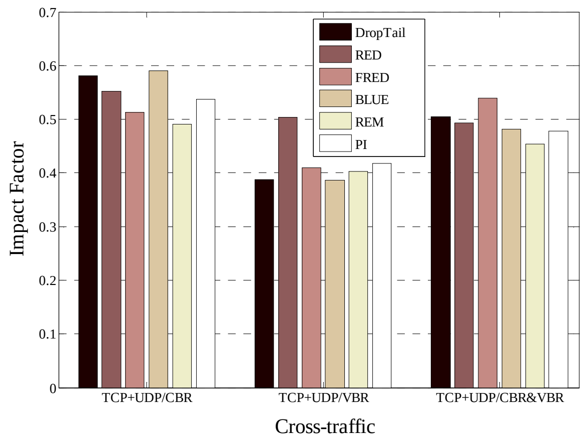

Figure 4 Impact factor of cross traffic with certain AQM algorithm in symmetric CMT-SCTP (in terms of average throughput)

When background traffic is '*TCP+UDP/CBR&VBR*', comparison with average throughput in CMT-SCTP without background traffic, $if_{DropTail} \approx 0.3875$; $if_{RED} \approx 0.5031$; $if_{FRED} \approx 0.4097$; $if_{BLUE} \approx 0.3863$; $if_{REM} \approx 0.4025$; and $if_{PI} \approx 0.4181$.

2) In terms of average delay. As show in Fig. 6, when background traffic is '*TCP+UDP/CBR*', comparison with average delay in CMT-SCTP without background traffic, $if_{DropTail} \approx 0.9164$; $if_{RED} \approx 0.1922$; $if_{FRED} \approx 0.1543$; $if_{BLUE} \approx 0.9753$; $if_{REM} \approx 0.7121$; and $if_{PI} \approx 0.8973$.

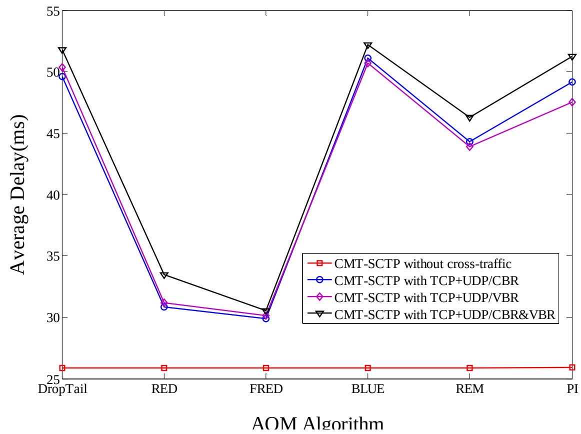

Figure 5 Comparison of average delay on symmetric CMT-SCTP with different cross traffic

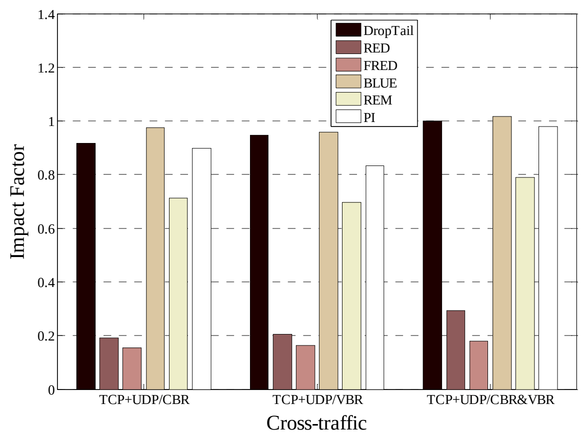

Figure 6 Impact factor of cross traffic with certain AQM algorithm in symmetric CMT-SCTP (in terms of average delay)

When background traffic is '*TCP+UDP/VBR*', comparison with average delay in CMT-SCTP without background traffic, $if_{DropTail} \approx 0.9464$; $if_{RED} \approx 0.2043$; $if_{FRED} \approx 0.1644$; $if_{BLUE} \approx 0.9584$; $if_{REM} \approx 0.6971$; and $if_{PI} \approx 0.8337$.

When background traffic is '*TCP+UDP/CBR&VBR*', comparison with average delay in CMT-SCTP without background traffic, $if_{DropTail} \approx 1.001$; $if_{RED} \approx 0.2933$; $if_{FRED} \approx 0.1799$; $if_{BLUE} \approx 1.018$; $if_{REM} \approx 0.7893$; and $if_{PI} \approx 0.9789$.

3) In terms of loss rate. As show in Fig. 8, when background traffic is '*TCP+UDP/CBR*', comparison with loss rate in CMT-SCTP without background traffic, $IA_{DropTail} \approx$

0.8124; $IA_{RED} \approx 0.6755$; $IA_{FRED} \approx 1.7709$; $IA_{BLUE} \approx 0.7605$; $IA_{REM} \approx 0.4187$; and $IA_{PI} \approx 0.6245$.

When background traffic is '*TCP+UDP/VBR*', comparison with average delay in CMT-SCTP without background traffic, $IA_{DropTail} \approx 0$; $IA_{RED} \approx 0.6849$; $IA_{FRED} \approx 1.1614$; $IA_{BLUE} \approx 0.0144$; $IA_{REM} \approx 0.1337$; and $IA_{PI} \approx 0.0976$.

When background traffic is '*TCP+UDP/CBR&VBR*', comparison with average delay in CMT-SCTP without background traffic, $IA_{DropTail} \approx 0.3377$; $IA_{RED} \approx 0.6096$;

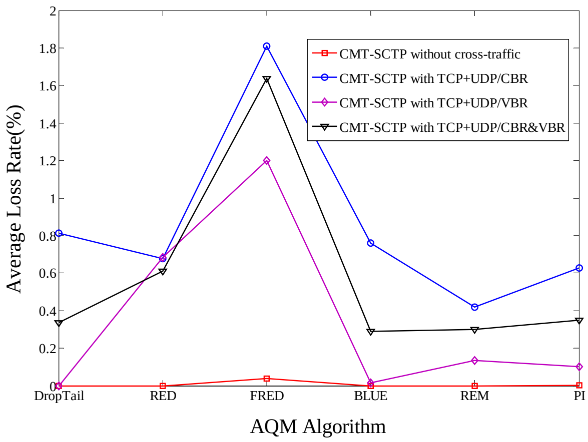

Figure 7 Comparison of loss rate on symmetric CMT-SCTP with different cross traffic

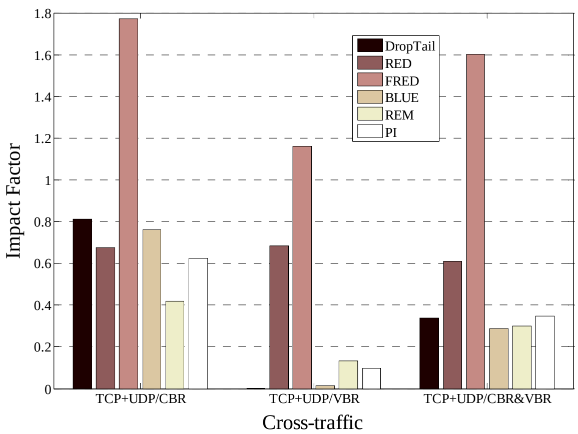

Figure 8 Incremental of cross traffic with certain AQM algorithm in symmetric CMT-SCTP (in terms of loss rate)

$IA_{FRED} \approx 1.601$; $IA_{BLUE} \approx 0.2887$; $IA_{REM} \approx 0.2981$; and $IA_{PI} \approx 0.3483$.

Conclusion 1: Per analyzing if_i and IA_i for symmetric CMT-SCTP with certain AQM algorithm and background traffic, conclusion expressed by expression (3) can be nailed down as follows:

In terms of Average throughput:

a) **if** *TCP+UDP/CBR* **then** set *REM -> Optimized*(QoS)

b) **if** *TCP+UDP/VBR* **then** set *BLUE -> Optimized*(QoS)

c) **if** *TCP+UDP/CBR&VBR* **then** set *REM -> Optimized*(QoS)

In terms of Average delay:

a) **if** *TCP+UDP/CBR* **then** set *FRED -> Optimized*(QoS)

b) **if** *TCP+UDP/VBR* **then** set *FRED -> Optimized*(QoS)

c) **if** *TCP+UDP/CBR&VBR* **then** set *FRED-> Optimized*(QoS)

In terms of loss rate:

a) **if** *TCP+UDP/CBR* **then** set *REM -> Optimized*(QoS)

b) **if** *TCP+UDP/VBR* **then** set *BLUE -> Optimized*(QoS)

c) **if** *TCP+UDP/CBR&VBR* **then** set *BLUE-> Optimized*(QoS)

4.2 Test for asymmetric CMT-SCTP

For simulation topology shown in Fig. 2, we use AQM algorithm (DropTail, RED, FRED, BLUE, REM and PI) to get average throughput and delay under certain background traffic condition which shown in Fig. 9 and 11 respectively. Per calculating by equation (1), a corresponding *if* statistic is shown in Fig. 10 and 12, respectively.

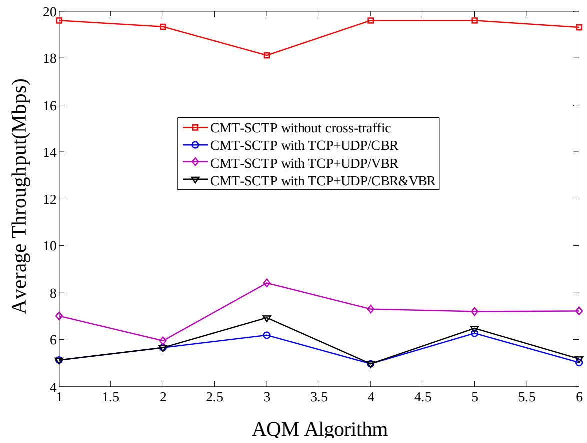

Figure 9 Comparison of average throughput on asymmetric CMT-SCTP with different cross traffic

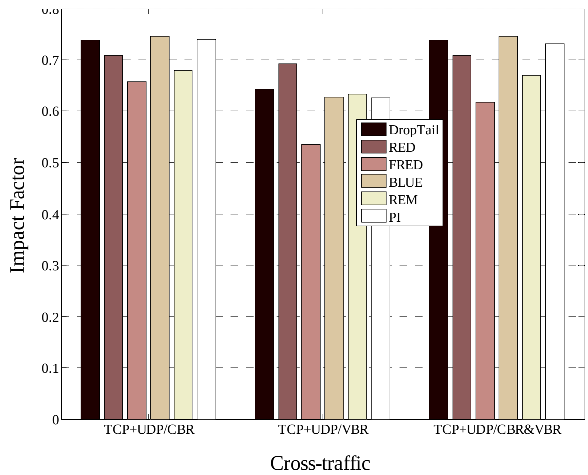

Figure 10 Impact factor of cross traffic with certain AQM algorithm in asymmetric CMT-SCTP (in terms of average throughput)

1）*In terms of average throughput*. As shown in Fig. 10, when background traffic is '*TCP+UDP/CBR*', comparison with average throughput in CMT-SCTP without background traffic, $if_{DropTail} \approx 0.7389$; $if_{RED} \approx 0.7081$; $if_{FRED} \approx 0.6580$; $if_{BLUE} \approx 0.7466$; $if_{REM} \approx 0.6796$; and $if_{PI} \approx 0.7401$.

When background traffic is '*TCP+UDP/VBR*', comparison with average throughput in CMT-SCTP without background traffic, $if_{DropTail} \approx 0.6429$; $if_{RED} \approx 0.6928$; $if_{FRED} \approx 0.5357$; $if_{BLUE} \approx 0.6270$; $if_{REM} \approx 0.6332$; and $if_{PI} \approx 0.6256$.

When background traffic is '*TCP+UDP/CBR&VBR*', comparison with average throughput in CMT-SCTP without background traffic, $if_{DropTail} \approx 0.7389$; $if_{RED} \approx 0.7079$; $if_{FRED} \approx 0.6176$; $if_{BLUE} \approx 0.7466$; $if_{REM} \approx 0.6700$; and $if_{PI} \approx 0.7312$.

2) In terms of average delay. As show in Fig. 12, when background traffic is '*TCP+UDP/CBR*', comparison with average delay in CMT-SCTP without background traffic, $if_{DropTail} \approx 0.7917$; $if_{RED} \approx 0.0566$; $if_{FRED} \approx 0.0828$; $if_{BLUE} \approx 0.8377$; $if_{REM} \approx 0.5520$; and $if_{PI} \approx 0.7788$.

When background traffic is '*TCP+UDP/VBR*', comparison with average delay in CMT-SCTP without background traffic, $if_{DropTail} \approx 0.9621$; $if_{RED} \approx 0.1425$; $if_{FRED} \approx 0.0745$; $if_{BLUE} \approx 0.9564$; $if_{REM} \approx 0.7687$; and $if_{PI} \approx 0.9560$.

When background traffic is '*TCP+UDP/CBR&VBR*', comparison with average delay in CMT-SCTP without background traffic, $if_{DropTail} \approx 1.0637$; $if_{RED} \approx 0.1068$; $if_{FRED} \approx 0.0886$; $if_{BLUE} \approx 1.1172$; $if_{REM} \approx 0.7305$; and $if_{PI} \approx 1.0587$.

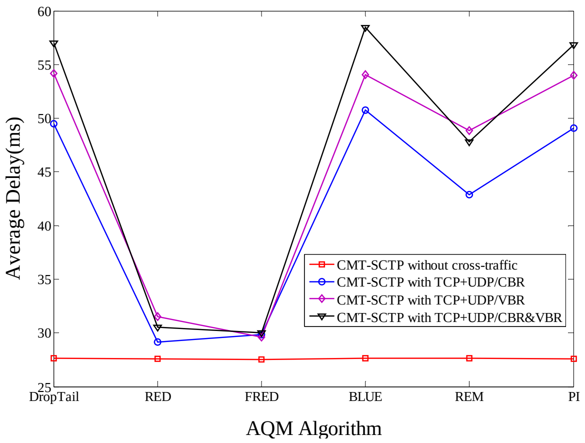

Figure 11 Comparison of average delay on asymmetric CMT-SCTP with different cross traffic

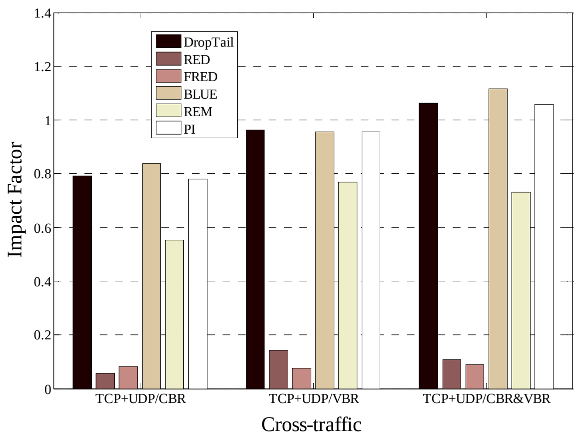

Figure 12 Impact factor of cross traffic with certain AQM algorithm in asymmetric CMT-SCTP (in terms of average delay)

3) In terms of loss rate. As show in Fig. 14, when background traffic is '*TCP+UDP/CBR*', comparison with loss rate in CMT-SCTP without background traffic, $IA_{DropTail} \approx 0.7087$; $IA_{RED} \approx 0.7263$; $IA_{FRED} \approx 1.1174$; $IA_{BLUE} \approx 0.7642$; $IA_{REM} \approx 0.5254$; and $IA_{PI} \approx 0.7218$.

When background traffic is '*TCP+UDP/VBR*', comparison with average delay in CMT-SCTP without background traffic, $IA_{DropTail} \approx 0.3440$; $IA_{RED} \approx 0.5423$; $IA_{FRED} \approx 0.8895$; $IA_{BLUE} \approx 0.3907$; $IA_{REM} \approx 0.3226$; and $IA_{PI} \approx 0.3298$.

When background traffic is '*TCP+UDP/CBR&VBR*', comparison with average delay in CMT-SCTP without background traffic, $IA_{DropTail} \approx 0.7087$; $IA_{RED} \approx 0.7801$; $IA_{FRED} \approx 1.118$; $IA_{BLUE} \approx 0.7642$; $IA_{REM} \approx 0.5241$; and $IA_{PI} \approx 0.8056$.

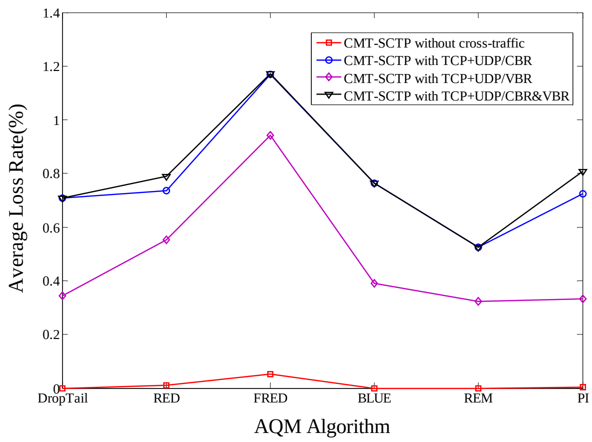

Figure 13 Comparison of loss rate on asymmetric CMT-SCTP with different cross traffic

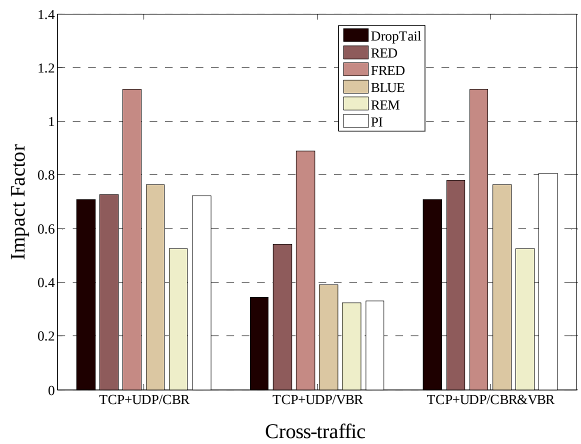

Figure 14 Incremental of cross traffic with certain AQM algorithm in asymmetric CMT-SCTP (in terms of loss rate)

Conclusion 2: Per analyzing if_i and IA_i for asymmetric CMT-SCTP with certain AQM algorithm and background traffic, conclusion can be nailed down as follows:

In terms of Average throughput:

a) **if** *TCP+UDP/CBR* **then** set *REM -> Optimized*(QoS)

b) **if** *TCP+UDP/VBR* **then** set *FRED -> Optimized*(QoS)

c) **if** *TCP+UDP/CBR&VBR* **then** set *FRED-> Optimized*(QoS)

In terms of Average delay:

a) **if** *TCP+UDP/CBR* **then** set *RED -> Optimized*(QoS)

b) **if** *TCP+UDP/VBR* **then** set *FRED -> Optimized*(QoS)

c) **if** *TCP+UDP/CBR&VBR* **then** set *FRED-> Optimized*(QoS)

In terms of loss rate:

a) **if** *TCP+UDP/CBR* **then** set *REM -> Optimized*(QoS)

b) **if** *TCP+UDP/VBR* **then** set *REM -> Optimized*(QoS)

c) **if** *TCP+UDP/CBR&VBR* **then** set *REM -> Optimized*(QoS)

5. Analysis with CMT-PF and Loss

5.1 CMT with Potentially-Failed State (CMT-PF)

To mitigate the recurring instances of receive buffer (*rbuf*) blocking, [15] and [16] introduced a new destination state called "potentially-failed". It is based on the rationale that loss detected by a timeout implies either severe congestion or failure in route. After a single timeout on a path, a sender is unsure, and marks the corresponding destination as "potentially-failed" (PF). A PF destination is not used for data transmission or retransmission. CMT's retransmission policies are augmented to include the PF state. CMT with the new set of retransmission policies is called CMT-PF. Details of CMT-PF are: (1) If a Transport Protocol Data Unit (TPDU) loss is detected by RFC4460's threshold number of missing reports, one of CMT's current retransmission policies is used to select an active destination for retransmission; (2) If a TPDU loss is detected after a timeout, the corresponding destination transitions to the PF state. No data is transmitted to a PF destination; (3) Heartbeats are sent to a PF destination with an exponential backoff of RTO (Retransmission TimeOut) after every timeout until (i) a heartbeat ack transitions the destination back to the active state, or (ii) an additional PMR (Path.Max.Retrans) consecutive timeouts confirm the path failure, after which the destination transitions to the failed state, and heartbeats are sent with a lower frequency as described in RFC4460; (4) Once a heartbeat *ack* indicates a PF destination is alive, the destination's *cwnd* is set to either 1 Maximum Transmission Unit (MTU) (CMT-PF1), or 2 MTUs (CMT-PF2), and data transmission follows the slow start phase; (5) Acks for retransmissions do not transition a PF destination to the active state.

For 2 MTUs (CMT-PF2) is recommended by RFC4960 for CMT-PF and employed into lots of researches [18-20]. However, this value is recommended with ignoring effect of cross traffic. In this section, we set 1 MTUs (CMT-PF1), 2 MTUs (CMT-PF2), 3 MTUs (CMT-PF3) and 4 MTUs (CMT-PF4) respectively to investigate which value can get more satisfied QoS in terms of throughput.

5.2 Simulation Topology Design

As more and more application encoded by VBR arise in Internet. So, cross traffic consist of TCP traffic, CBR traffic and VBR traffic should be more reasonable to be taken into simulation topology designing, in addition, transmission paths with varied loss rate in realistic network should be taken into our list as well. In this section, we use the most common scenario, that is symmetric CMT-SCTP, with *TCP+UDP/CBR&VBR* cross traffic to analyze the QoS under CMT-PF is turned on. And REM algorithm will be employed as queue management algorithm since it is recommended on benefit of QoS in terms of throughput in *TCP+UDP/CBR&VBR* condition as mentioned above, only throughput adopted as the metric simply in our experiments.

Simulation topology is shown in Fig. 15. All simulation parameters are set as mentioned in section 3.

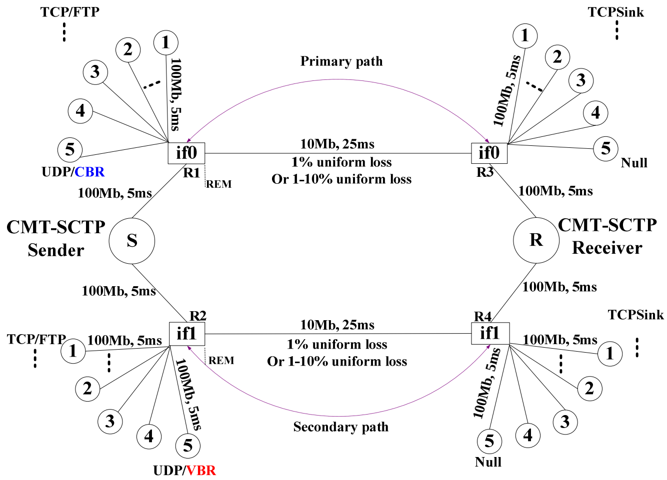

Figure 15 Symmetric CMT-PF simulation topology

5.2.1 During Short-term Failure

We perform one experiment where both path 1 and 2 fail for a brief period (path1 fails from 5 to 10 seconds, path 2 from 15 to 20 seconds) since different cross traffic specified on the two paths. In this experiment, both paths experience a low 1% loss rate, and sender S transfers data to receiver R by FTP means. Fig. 16 plots the average throughput, measured during the 5 seconds short-term failure on both two paths for various *rbuf* values. Since *rbuf* blocking increases as the *rbuf* size decrease [17], CMT-PF's throughput improvement as *rbuf* size increases. Evidently, CMT-PF3 performs the best performance when *rbuf* is greater than 32K.

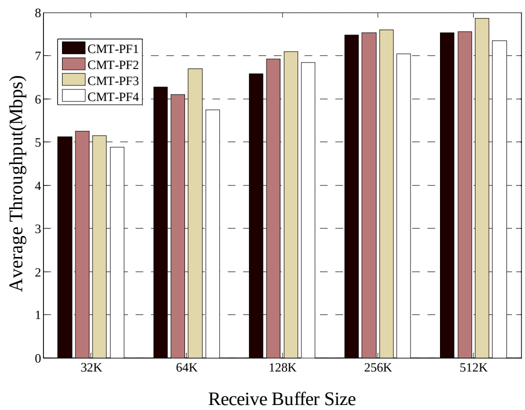

Figure 16 Average throughput during short-term failure

When *rbuf* is 32K, CMT-PF3 can get more advantage about 0.593% than CMT-PF1, about 5.3953% than CMT-PF4, but get less advantage about 2.02% than CMT-PF2.

When *rbuf* is 64K, CMT-PF3 can get more advantage about 6.7046% than CMT-PF1, about 9.9322% than CMT-PF2, and about 16.6259% than CMT-PF4.

When *rbuf* is 128K, CMT-PF3 can get more advantage about 7.8351% than CMT-PF1, about 2.5437% than CMT-PF2, and about 3.7703% than CMT-PF4.

When *rbuf* is 256K, CMT-PF3 can get more advantage about 1.5071% than CMT-PF1, about 0.8394% than CMT-PF2, and about 7.8628% than CMT-PF4.

When *rbuf* is 512K, CMT-PF3 can get more advantage about 4.5494% than CMT-PF1, about 4.1852% than CMT-PF2, and about 7.0392% than CMT-PF4.

5.2.2 During Congestion

We perform a final set of experiments to study CMT-PF1/2/3/4's performance when timeouts on a path are due to congestion rather than failure. Loss rate with three reasonable conditions are employed as follows, just default 64KB is used as *rbuf* in this experiment:

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

Condition 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%;

Condition 3: The loss rate on both two paths is varied from 1% to 10%.

We investigate the performance of CMT-SCTP with CMT-PF1, CMT-PF2, CMT-PF3 and CMT-PF4 under *condition 1*, 2 and 3 respectively. Corresponding average throughput are shown in Fig. 17, 18 and 19.

When paths' loss rate is low, most of the TPDU losses on the path can be recovered throughput fast retransmits, result in very few timeout recoveries. Hence, in Fig. 17, 18 and 19, CMT-PF1/2/3/4 performs a more satisfied performance, but average throughput will decrease as loss rate increase.

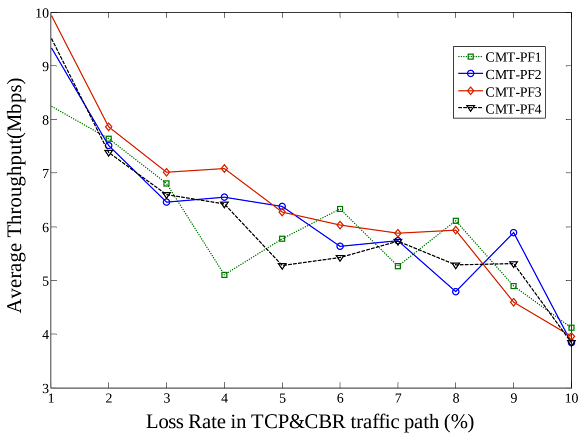

Figure 17 Average throughput with asymmetric path condition (Loss Rate on *TCP+UDP/CBR* traffic path)

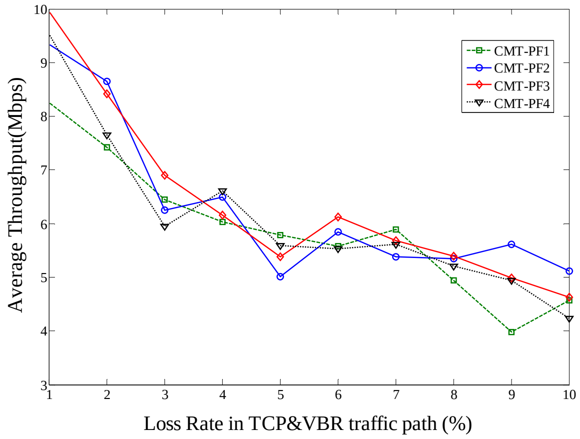

Figure 18 Average throughput with asymmetric path condition (Loss Rate on *TCP+UDP/VBR* traffic path)

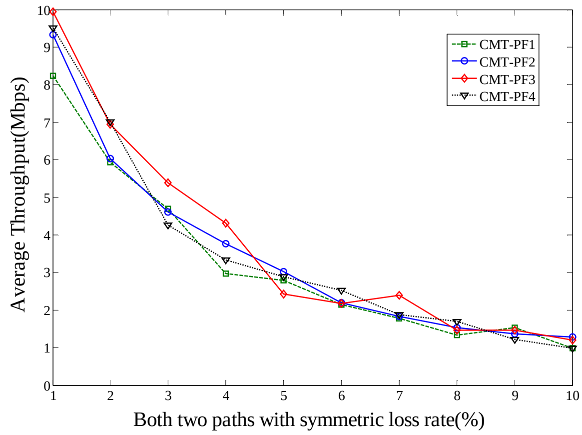

Figure 19 Average throughput with asymmetric path condition (Loss Rate on both two paths)

Accordance with those experiments addressed above, average throughput using CMT-PF1/2/3/4 under Condition 1/2/3 respectively, table 3 tabulates the average throughput during path's(s') loss rate varied from 1-10%.

Table 3 Average throughput under CMT-PF1/2/3/4

	CMT-PF1	CMT-PF2	**CMT-PF3**	CMT-PF4
CONDITION1	6.0309	6.2138	**6.4562**	6.0806
CONDITION2	5.8889	6.3047	**6.3625**	6.0844
CONDITION3	3.2395	3.4950	**3.7727**	3.5317

As shown in above table, when CMT-PF3 adopted in *condition 1*, comparison with CMT-PF1, it can get advantage is about 16.4592%; compare to CMT-PF, it's about 7.9459%; compare to CMTP-PF4, it's about 6.823.

When CMT-PF3 adopted in *condition 2*, comparison with CMT-PF1, it can get advantage is about 7.0515%; compare to CMT-PF2, it's about 3.9009%; compare to CMTP-PF4, it's about 6.8230%.

When CMT-PF3 adopted in *condition 2*, comparison with CMT-PF1, it can get advantage is about 8.0427%; compare to CMT-PF2, it's about 0.9171%; compare to CMTP-PF4, it's about 6.1760%.

Conclusion 3: Base on analysis in this section, a conclusion can be nailed down that, CMT-PF3 can get more advantage in terms of average throughput than CMT-PF2 which is recommended by RFC4960. So, when loss rate arise and CMT-PF is turned on, 3 Maximum Transmission Unit (MTU) (CMT-PF3) is recommended to be set as the value of the destination's *cwnd* once a heartbeat *ack* indicates a PF destination is alive.

6. Conclusions

In this paper, per designed realistic-like simulation topology, Firstly, we analyzed the QoS of network redundancy in symmetric/asymmetric CMT-SCTP with typical AQM algorithm. Two goals had be reached in this part: 1) work out how effect the specified cross traffic plays in CMT-SCTP with different AQM algorithm thereby impact factor; and 2) found out which AQM algorithm can get better QoS for CMT-SCTP under certain cross traffic.

Secondary, we use the most common scenario, that is symmetric CMT-SCTP, with *TCP+UDP/CBR&VBR* cross traffic to analyze the average throughput under CMT-PF is turned on. And REM algorithm was employed as queue management algorithm since it is recommended on benefit of QoS in terms of throughput in *TCP+UDP/CBR&VBR* condition, a conclusion can be nailed down that, CMT-PF3 can get more advantage in terms of average throughput than CMT-PF2 which is recommended by RFC4960. So, CMT-PF3 is recommended when loss rate arise and CMT-PF is turned on.

For realistic network which with loss rate and more and more applications encoded by VBR deployed, REM algorithm and CMT-PF3 is recommended for the common symmetric CMT-SCTP in order to get more satisfied performance.

References

[1] R. Stewart, "Stream Control Transmission Protocol," *IETF RFC 4960* (Proposed Standard), Sep. 2007.

[2] Y. Cao, C. Xu, J. Guan, *et al.*, "Relational Analysis Based Concurrent Multipath Transfer Over Heterogeneous Vehicular Networks," *International Journal of Computer Science Issues*, Vol.9, 2012, pp.1-10.

[3] Y. Cao, C. Xu, J. Guan, H. Zhang, "Background Traffic-based Retransmission Algorithm for Multimedia Streaming Transfer over Concurrent Multipaths", *International Journal of Digital Multimedia Broadcasting*, Vol.2012, Article ID 789579, 10 pages, 2012.

[4] Y. Cao, C. Xu, J. Guan, H. Zhang, "Cross-layer Retransmission Approach for Efficient VoD Transfer over Multi-homed Wireless Networks", *International Journal of Digital Content Technology and its Applications* (in press), Aug. 2012.

[5] J. Iyengar, P.Amer, and R. Stewart, "Concurrent Multipath Transfer Using SCTP Multihoming Over Independent End-to-End Paths", *IEEE/ACM Transactions on Networking*, Vol.5, No.14, 2006, pp.951-964.

[6] H. M. Tahir, A. M. Said, M. A. A. Seman, S. N. Shelena, M. S. Selan and S. H. Abdi, "Performance Comparison of SCTP and UDP Over Mobile Ad Hoc Networks", *International Journal of Computer Science Issues*, Vol.9, No.2, 2012, pp.443-448.

[7] Y. Cao, C. Xu, J. Guan, F. Song, H. Zhang, "Environment-aware CMT for Efficient Video. Delivery in Wireless Multimedia Sensor Networks", *International Journal of Distributed Sensor Networks* (in press), Oct. 2012.

[8] Y. Cao, C. Xu, and J. Guan, "A record-based retransmission policy on SCTP's Concurrent Multipath Transfer," in *Proceedings of 2011 International Conference on Advanced Intelligence and Awareness Internet (AIAI 2011)*, pp.67-71, Oct. 2011.

[9] M. Fomenkov, K. Keys, D. Moore and K. Claffy(Jan.2004). Longitudinal study of Internet traffic in 1998-2003: *proceedings of the Winter International Symposium on Information and Communication Technologies (WISICT)*, 2004, pp.1-6, Cancun, Mexico.

[10] S. Floyd, V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Trans on Networking*, Vol.1, 1993, pp.397-143.

[11] B. Braden, D. Clark, J. Crowcroft, B. Davie, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. K. Ramakrishnan, S. Shenker, J. Wroclawski, L. Zhang, "Recommendation on Queue Management and Congestion Avoidance in the Internet," *IETF RFC 2309*, Apr. 1998.

[12] L. Brakmo and S. O'Malley. TCP-Vegas: New techniques for congestion detection and avoidance. in *Proceedings of ACM SIGCOMM'94*, 1994, pp. 24-35.

[13] W. Feng, D. D. Kandlur, D. Saha, and D. G. Shin. BLUE: A New Class of Active Queue Management Algorithms. *Technical Report CSE-TR-387-99*, Dept. of EECS, University of Michigan, April 1999.

[14] S. Athuraliya, V.H. Li, S.H. Low, et al. REM: active queue management. *IEEE Network*, 2001,15(3):48-53.

[15] C. V. Hollot, V. Misra, D. Towsley and W. Gong. Analysis and design of controllers for AQM routers supporting TCP flows. *IEEE Trans. AutomaticControl*, vol. 47, pp. 945-959, 2002.

[16] Lin Cui, S.J. koh, W.J. Lee, "Fast selective ACK scheme for throughput enhancement of multi-homed SCTP hosts," *IEEE Communications Letters*, Vol.14, No.6, 2010, pp.587-589.

[17] C.M. Huang, M.S. Lin, "Fast Retransmission for Concurrent Multipath Transfer (CMT) over Vehicular Networks", *IEEE Communication Letters*, Vol.15, No.4, 2011, pp.386-388.

[18] J. Liao, J. Wang, X. Zhu, cmpSCTP: An Extension of SCTP to Support Concurrent Multi-Path Transfer. *IEEE International Conference on Communications,2008 (ICC'08)*, pp.19-23 May 2008.

[19] L. Budzisz, R. Ferrus, F. Casadevall, P. Amer, On Concurrent Multipath Transfer in SCTP-Based Handover Scenarios. *IEEE International Conference on Communications (ICC'09)*, 2009, pp.14-18.

[20] F. Song, H. Zhang, S. Zhang, F.M.V. Ramos, J.Crowcroft, Relative Delay Estimator for SCTP-Based Concurrent Multipath Transfer. *IEEE Global Telecommunications Conference (GLOBECOM'08)*. 2010, pp.6-10.

[21] The Network Simulator-- ns-2. *http://www.isi.edu/nsnam/ns/*

[22] *http://www.isi.edu/nsnam/archive/ns-users/webarch/2001/-msg05051.html*

[23] P. Natarajan, N. Ekiz, P. Amer, J. Iyengar, R. Stewart, "CMT using SCTP Multihoming: Transmission Policies using a Potentially-failed Destination State," *Technique Report*, CIS Dept, University of Delaware, Feb. 2007.

[24] P. Natarajan, N. Ekiz, P. Amer, J. Iyengar, R. Stewart, "Concurrent Multipath Transfer using SCTP Multihoming: Introducing the Potentially-failed Destination State," *NETWORKING 2008*, 2008, pp.727-734.

Meifang Shen received his B.S. degree from Hebei Normal University of China in 2006, received his M.S degree from Beijing University of Posts and Telecommunications (BUPT) in 2008. During 2007-2008, she worked as an intern in Pacific Century Cyber Works Ltd. (PCCW). During 2008-2010, he worked as a software engineer in Wonders Information Co., Ltd. (Beijing). She is currently working in the Talend Beijng Technology Co., Ltd. as a senior software test developer. She is broadly interested in computer networks, software testing and development, as well as next generation Internet technology.

Yuanlong Cao received his B.S. degree from Nanchang University of China in 2006, received his M.S degree from Beijing University of Posts and Telecommunications (BUPT) in 2008. During 2007-2009, he worked as an intern in BEA China Telecommunications Technology Center (BEA TTC) and IBM China Development Lab (IBM CDL) and received the best intern award from IBM Enterprise Content Management (ECM) team and the special prize in the public testing held by the IBM WebSphere Commerce China Solution team. During 2009-2010, he worked as a software engineer in DT Research (Beijing). He is currently working toward the Ph.D. degree in the Institute of Network Technology, BUPT. He is broadly interested in computer networks, multimedia communications, wireless networking, network security, and next generation Internet technology.

Mingchuan Zhang received his B.S. degree in computer application from Harbin Engineering University of China in 2005. He is a Lecturer in Henan University of Science and Technology, China. He is currently working toward the Ph.D. degree in the Institute of Network Technology, Beijing University of Posts and Telecommunications (BUPT). In particular, his research interests include ad hoc network, cognitive network, and next generation Internet technology.

Kaixuan Wang received his M.S. degree from Shanxi University of China in 2005. He is a Lecturer in Shanxi University of Finance and Economics, China. He is currently working toward the Ph.D. degree in the Institute of Network Technology, Beijing University of Posts and Telecommunications (BUPT). He is broadly interested in cognitive network, network fault management, Smart Grid communication network and next generation Internet technology.

Mingming Yu received his B.S. degree from Nanchang University of China in 2006, received his M.S degree from Beihang University of China in 2010. He is an engineer in Commercial Aircraft Corporation of China Ltd. (COMAC). He is broadly interested in computer networks, software development and Architecture.