Journal of Computer Science 7 (12): 1859-1866, 2011
ISSN 1549-3636

# Performance Metrics of Multipath State Aware Concurrent Multipath Transfer using Redundant Transfer in Stream Control Transmission Protocol Multihoming for Symmetric Paths

M. Subramaniam and D. Manjula
Department of Computer Science and Engineering, CEG,
Anna University, Chennai, India

**Abstract: Problem statement:** In multihomed host, if data transfer starts using CMT, unaware of paths status causes delay in data transfer or packet loss due to path failure. Transmission Path (one or more paths) failures result in out-of-order data delivery causing receiver buffer blocking and preventing sender from transmitting further data. **Approach:** Multipath State Aware Concurrent Multipath Transfer using Redundant Transfer (MSACMT-RT), status of multipath is determined, initially and periodically before transmission. Data transfer begins immediately after determining the path status, for a particular interval of time and later transfers with CMT only for predefined period. **Results:** We discuss MSACMT-RT performance in symmetric paths with the constrained receiver buffer (rbuf) value of 128 and 256KB's. Our simulation result shows Percentage of throughput Increase between 5-15% for transfer of 20 and 40MB file sizes. **Conclusion:** By using MSACMT-RT we infer that our algorithm outperforms on par or better but never worse than Concurrent Multipath Transfer Potentially Failed (CMT-PF). MSACMT-RT mechanism in alternate paths increases the throughput during aggressive failures and Non-failure scenarios. Larger the file sizes of file transfer, greater the degree of throughput with less transfer time is achieved.

**Keywords:** Receiver buffer, Stream Control Transmission Protocol (SCTP), Round Trip Time (RTT), symmetric paths, Concurrent Multipath Transfer Potentially Failed (CMT-PF), aggressive failures, non-failure scenarios

## INTRODUCTION

The Stream Control Transmission Protocol (SCTP) over an IPv4/IPv6 networks is a general purpose transport protocol featuring multi-homing and multi streaming (Al-Talib *et al*., 2009) support. SCTP is capable of transmitting different types of data's simultaneously on a single stream with its Multi-streaming feature to destination through multiple paths as Multi-homing supports multiple IP address within its single association. A host is multihomed (Stewart and Xie, 2002) if it can be addressed by multiple IP address, as in the case when the host has multiple network interfaces.

Multipath transport protocols have the potential to greatly improve the performance and resilience of Internet traffic flows. To fully utilize available resources and increase application throughputs, multi-path transmission between sources and destinations is important. In the time varying network conditions of each path, such as bottleneck bandwidth, delay and congestion must be detected and taken into account before transmission. Distributing packets along multiple paths with different bandwidths and delays leads to a large number of out-of-order packets arriving at the receiver. In multi-path environments, the out-of-order packets problem will become serious. In order to solve the out-of-order packets issue, a buffer is used to accomplish the packet reordering at the receiver. However, when using a limited buffer at the receiver, the out-of-order arrival of packets can lead to missing data for real-time multimedia applications. Reordering can also increase buffer requirements at the receiver and can also increase end-to-end delay. But our objective is to use memory efficiently in high end computing environment. We present some modifications to CMT-PF to prevent side-effects of the RTO expiry for improving the throughput efficiency.We work on the foundations of superior system already proposed by (Natarajan *et al*., 2008) (Yilmaz *et al*., 2010), CMT-SCTP the SACK chunks gap were Non-Renegeable SACK (NR-SACK), which allows a receiver to declare gap acknowledgements.

**Corresponding Author:** M. Subramaniam, DCSE, College of Engineering, Anna University, Chennai, (India)

We discuss our flow in the following style, literature review discusses from evolution to progress of study. We discuss our network simulation topology for symmetric paths followed by flaws in the existing system in problem description and the method for overcoming this problem. Next we discuss our proposed algorithm for MSACMT-RT and assumptions made. Finally we evaluate our performance during Non-failure and failure at regular and irregular intervals scenarios. Finally we conclude with our results and Conclusion.

**Literature review:** We know that SCTP supports Multihoming; the CMT-SCTP extension (Iyengar *et al*., 2005) provides CMT facility. Concurrent Multipath Transfer (CMT) is the concurrent transfer of new data from a source to a destination host via two or more end-to-end paths. Concurrent Multipath Transfer (CMT) between multihomed source and destination hosts also increases the application"s throughput (Iyengar *et al*., 2004). SCTP"s Multihoming feature having multiple interfaces with multiple IP"s, allows data to be transmitted through multiple interfaces (having multiple IP"s) and in case of any failure in its primary path, data is automatically transmitted through its Alternate Path (IP) or in its secondary path (Stewart and Xie, 2002). Proposes a default Path.Max.Retransmit (PMR =5), based on a network"s loss rate which translates to ≈63s (6 Consecutive timeouts for failure detection). Adopting the value PMR = 5, whenever the value goes beyond the set threshold (PMR = 5), the sender retransmits in alternate path.

Iyengar *et al*., (2006) (Stewart and Xie, 2002) (Iyengar *et al*., 2005) explored five retransmission policies for CMT and their overall analysis revealed that RTXLOSSRATE, RTX-SSTHRESH and RTX-CWND outperforms better than others. They also assumed strongly that the receiver's advertised window does not constrain the sender and since the bottleneck queues on the end-to-end paths used in CMT are independent (Caro *et al*., 2006), (Iyengar *et al*., 2004). Jiemin *et al*. (2008) (Caro, 2006), further alleviated some of the throughput degradation caused by rbuf blocking problem, by combining the parameters of Congestion window, Slow start thresholds and Lossrate (CSL) and suggested compound parameter Retransmission policy (i.e., Rtx-CSL policy) and improved the goodput of infinite rbuf CMT association, however they have not discussed with finite rbuf (Constrained receiver buffer).

However to achieve faster yet robust failure detection, (Iyengar *et al*., 2006) argues for varying Path Max Retransmit (PMR) based on a network's loss rate and suggests PMR = 3 for the Internet. Also, a tradeoff exists on deciding the value of PMR-a lower value reduces rbuf blocking but increases the chances of spurious failure detection, whereas a higher PMR increases rbuf blocking and reduces spurious failure detection in a wide range of environments (Natarajan *et al*., 2009). Trong *et al*. (2010) achieved good throughput for single radio multi-channel multi-path wireless mesh networks, by utilizing scheduling (Gao *et al*, 2004) availability with only one wireless card i.e., in Single path, therefore about concurrent transfer in multipath is not discussed.

Our work rests on the foundations of excellent system already proposed by (Natarajan *et al*., 2008) (Yilmaz *et al*., 2010), CMT-SCTP the SACK chunks gap acknowledgement were renegable, for which the receiver may decide not to acknowledge them, so called Non-Renegeable SACK (NR-SACK), which allows a receiver to declare gap acknowledgements. This NR-SACK significantly improved the CMT transport's performance over dissimilar paths-as shown in (Dreibholz *et al*., 2010) (Adhari *et al*., 2011) by allowing a sender to remove gap-acknowledged chunks from its sender buffer. In order to handle sender buffer and receiver buffer, buffer splitting is also done based on outstanding bytes (Adhari *et al*., 2011). We include the above discussed concept in our MSACMT-RT mechanism for improving the overall performance of the system.

**Simulation topology:** We consider an SCTP multihoming association over the typical network topology as shown in the Fig.1. In Fig 4, The MSACMT-RT sender A has three interfaces- A1, A2 and A3 upon which the data is transmitted to the MSACMT-RT receiver B via its interfaces-B1, B2 and B3. All the interfaces A1-A3 and B1-B3 are identified by unique IP address in the network. These six IP addresses are bound in the MSACMT-RT"s association such that sender employs the three independent paths- path1, path2 and path3 for data transmission. We evaluate our experiment for symmetric paths, i.e., paths having almost same QoS characteristics subject to minor variation, as Internet setups are more realistic.

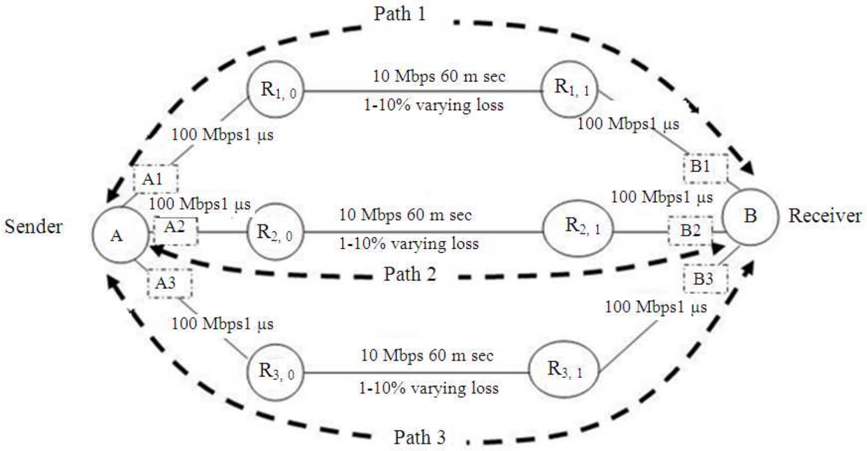

Fig. 1: Simulation topology

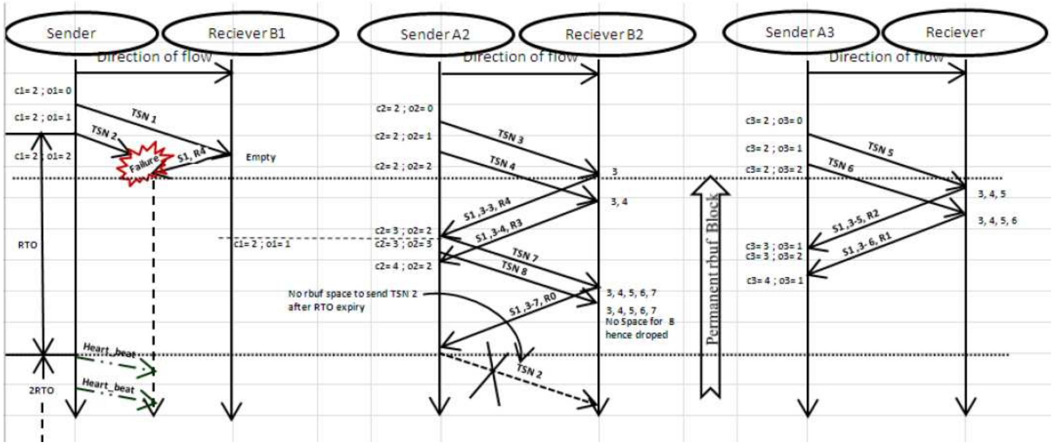

Fig. 2: CMT-PF rbuf blocking during path failure using three paths

## MATERIALS AND METHODS

Interface A1 of the sender A is connected to the router $R_{1,0}$ has bandwidth of 100 Mbps with a delay of 1µs and no loss. $R_{1,0}$ is connected to $R_{1,1}$ has a bandwidth of 10 Mbps and 60ms delay, with a varying loss between 1-10%. Receiver B is connected to the Router $R_{1,1}$ through the interface B1. $R_{1,1}$ to B1 has the bandwidth of 100 Mbps with a delay of 1µs and no loss. The route from sender A to receiver B via A1, $R_{1,0}$, $R_{1,1}$ and B1 defined as Path 1. Since our experimental setup is to have almost identical path QoS, the second interface A2 of the sender A is connected to the receiver B via $A_2$, $R_{2,0}$, $R_{2,1}$ and $B_2$ defined to be Path 2. Similarly, the third interface $A_3$ of the sender A is connected to the receiver B is through the interface via A3, $R_{3,\ 0}$, $R_{3,\ 1}$ and B3 defined to be Path 3. Both paths, Path 2 and Path 3 have the identical connectivity characteristics as defined for Path 1. MSACMT-RT receiver maintains a single receiver buffer (rbuf), which is shared across the sub-association flows in an association. Irrespective of the layer at which Multipath transfer is performed, a similar shared buffer would exist at a receiver.

**Problem description:**

**Permanent receiver buffer blocking in CMT-PF:** Natarajan *et al* (2009), claims for mitigating the failure-induced receiver buffer (rbuf) blocking problem by introducing a new „„ Potentially-Failed" (CMT-PF) and improved the performance of data transmission comparing with receiver buffer blocking in CMT. When using three paths their failure induced rbuf problem depicted through their experiment resulted permanent receiver blocking, which is explained via time line diagram of Fig. 2. The CMT" s sender (A) has three interfaces-A1, A2 and A3 and transmits data to receiver (B) with three interfaces-B1, B2 and B3. All six IP addresses are bound in the CMT association such that the sender employs the three independent paths- path 1, path 2 and path 3 for data transmission. Each TPDU consists of an MTU-sized(1 MTU =1500 Bytes) data chunk, which is assigned a unique Transmission Sequence Number (TSN).Both forward and reverse paths between A1 and B1 fail just after TSN 2 enters the network. Hence, TSN 2 and the SACK for TSN 1 are presumed lost. $C_i$ and $O_i$ denote the congestion window (cwnd) in number of MTUs and the number of outstanding TPDUs, respectively, on path i. A SACK labeled ($S_a$,b-c;$R_d$) cumulatively acknowledges all TSNs up to and including "a", selectively acknowledges TSNs b through c (missing report for TSNs a+1 through b-1) and advertises a receiver window capable of buffering d more TSNs. In Fig. 1"s example, the transport layer receive buffer can hold a maximum of 5 TSNs and its contents are listed after the reception of every TSN. TSNs 3 and 4 are received out of order and stored in the receive buffer at B2, similarly TSNs 5 and 6 are stored in the receiver buffer of the B3. Due to various associated delays in the network (Processing, Transmission, Queuing, Propagation delays) on each path, transmitted TSNs arrive at the receiver at different intervals. Here in our case TSN 1 arrived first and TSN 2 is lost due to induced failure, TSNs 3 and 4 arrives second and TSNs 5 and 6 arrived last. Each of these TSNs triggers a SACK to the sender. The CMT sender uses the Cwnd Update for CMT (CUC) algorithm (Iyengar *et al*., 2006) to decouple a path"s cwnd evolution and data ordering. Based on SACK triggered by TSN 3 the sender uses CUC and increments C2 = 3 from 2 and decrement O2 = 1 from 2, now the available receive buffer space for new data is calculated as advertised receive window (R) = 4-total outstanding on O2 (= 2), triggers the sender to transmit two TSNs, 7 and 8 on path 2. Since TSNs 3, 4, 5 and 6 are already in receiver buffer, only one TSN could be accommodated and hence TSN 7 is accommodated and TSN 8 is dropped due to unavailability of receiver buffer space.

Similarly SACK triggered by TSN 5 the sender uses CUC and increments C3 = 3 from 2 and remains O3 = 2, since TSN 6 and TSN 2 are outstanding, now the available receive buffer space for new data is calculated as advertised receive window (R) = 2-total outstanding on O2 (= 2) which is equal to zero, prevents further data transmission through path 3. On path 1, even though 1 MTU worth of new data could be transmitted (C1 > O1), rbuf blocking, i.e., flow control, terminates data transmission. When path 1's retransmission timer expires for TSN 2, the CMT-PF sender transitions path 1 to the PF state and transmits a heartbeat on path 1. Now there is no possibility of attempting to transfer TSN 2 in path 2 since there is no buffer space at the receiver and therefore unable to transfer TSN 2, fails to clear buffer. Hence receiver B encounters indefinite blocking in its predefined buffer.

**Defeating the problem:** The similar timeline scenario as described in Fig. 2 is described in Fig. 3 using MSACMT- Here we use the least prioritized path as the standby path for least priority path. This standby path is only temporary. After ten consecutive successful transmissions in the standby path, MSACMT-RT ensures the path durability and used from this time onwards the standby path is used as normal CMT path for further data transfer). Detailedly we discussed MSACMT-RT algorithm in forth coming section. In the defined symmetric paths between sender and receiver, due to the burst nature of internet traffic flow, there is minor magnitude of change in paths quality. Hence based on the initial parameter values of paths that have least Round Trip Time (RTT), largest Congestion Window (cwnd), largest slow start thresh (ssthresh) and lower loss rate (loss rate), the paths priority is defined before beginning transmission. Initial setup parameters ascertain, path having highest priority has better QoS comparing to the path having its next priority.

Since Path 1 has the highest priority that receives the first set of TSN's scheduled to transmit and Path 3 having least priority is used as redundant path for the highest priority path, Path 1. Hence TSN 1 and TSN 2 are transmitted in Path 1 and Path 3. The second set of TSN's TSN 3 and TSN 4 are transmitted in Path 2. In Path 1, network path fails just after TSN 2 is transmitted into the network, this is done by bringing the router $R_{1,0}$ and $R_{1,1}$ after 10s of data transfer, hence, TSN 2 and the SACK for TSN 1 are presumed lost. TSN 3 and TSN 4 which is arrived out-of- order is received by the receiver and stored in receiver buffer. From this time onwards rbuf blocking starts. Few times later after TSN 4 arrived, duplicated TSN 1 and TSN 2 are received by the receiver B through Path 3. TSN 1 is discarded by the receiver as it was already received through Path 1. Receipt of TSN 2 in Path 2 clears the receiver buffer, thereby blocking of rbuf is eliminated. SACK triggered by TSN 3, increases the cwnd by 1, i.e., from C = 2-C2 = 3 and O2 = 2. The available receive buffer space for new data, calculated as advertised receive window (=4)-total outstanding (= 2), allows the sender to transmit two TSNs, 5 and 6, on path 2. SACK triggered by TSN 1, increases the cwnd by 1, i.e from cwnd = 2 to C3 = 3 and O2 = 2. SACK triggered by TSN 1 on Path 3 in shows the receipt of Number of Duplicate TSNs (X) which triggers the Path 3 to assign as standby path for Path 2. For the calculated buffer space allows the sender A3 to transmit two duplicate TSNs, 5 and 6, on path 3.

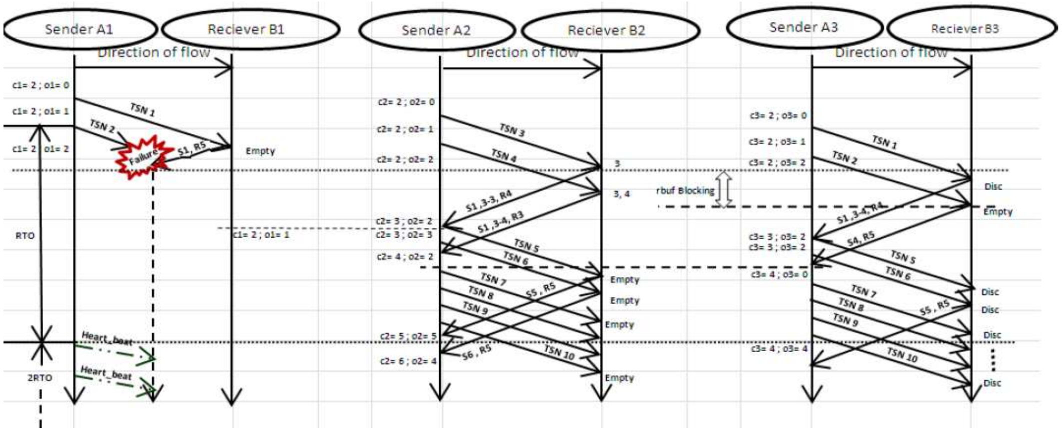

Fig. 3: MSACMT-RT smooth flow during path failure using three paths

After progression of growth in cwnd for two consecutive times and positive acknowledgements for the transmitted TSNs, now both the two paths are used for CMT transfer without any redundancy upto ten positive cycles. The process of observing the status of multiple paths by examining the parameters such as RTT/RTO, cwnd, ssthresh and lossrate after successive ten transmissions is said to be Multipath State Aware Concurrent Multipath Transmission using Redundant Transmission (MSACMT-RT. This redundant transmission ensures the paths durability and alternates to exercise itself for CMT along with other paths.

**MSACMT-RT algorithm:**

Step 1: The sender A transmits the INIT Chunk (Type = 1) over multiple paths. (Here path 1, 2 and 3, this chunk is used to initiate an SCTP association between two end points.

Step 2: The receiver B acknowledges the sender A using INIT ACK Chunk (Type = 2) for the initiation of the association.

Step 3: Determine Round Trip Time (RTT)/RTO, Congestion Window (cwnd), slow start thresh (ssthresh) and loss rate (lossrate).

Step 4: Assign path the priority from highest to least that is having least Round Trip Time (RTT), largest Congestion Window (cwnd) and largest slow start thresh (ssthresh) and low loss rate (lossrate).

Step 5: The cwnd determined for the least prioritized path size is taken and assigned uniformly for all paths on a per-destination-address, before transmission starts. Assign least prioritized path as redundant path for highest prioritized path.

Step 6: Start/Resume transmitting in all paths.

Step 7: Is transmission successful in all Paths? Step 8: Step 3 (C language syntax)

Step 8: Count (Consecutive successful transmission < = 10)? Step 6: Step 3

**Assumptions:**

- If a TPDU loss is detected after a timeout, the corresponding destination is stated dead route and the sender never transmits data through this path
- Heartbeats are sent to dead destination (s) with an exponential backoff of RTO after every timeout until either (i) A heartbeat ack transitions the destination back to the active state, or (ii) An additional PMR consecutive timeouts confirm the path dead, upon which the destination transitions to the dead state and heartbeats are sent with a lower frequency
- If ever a heartbeat “ack” indicates a dead destination is alive, that destination”s cwnd is set to either 1 MTU or 2 MTUs and the sender follows the slow start algorithm to transmit data to this destination
- Ack’s for retransmissions do not transition a dead destination back to the active state, since a sender cannot disambiguate whether the “ack” was for the original transmission or the retransmission (s)

## RESULTS AND DISCUSSION

**Performance evaluation:**

**Citation revisited:** Natarajan *et al*. (2009)Computer Communications 32 (2009) 1577-1587 (Caro *et al*., 2006), In page 1579 Fig. 2, in Path 2after SACK received for TSN 3, later transmitting TSN 5, the cwnd c = 2 is correct, because $C_i$ and $O_i$ is maintained per destination basis only, similarly Fig. 4 in page 1581 adds little bit confusion. Their experimental works described in page 1580, last paragraph says, Path 2 is failed by bringing down the bidirectional link between routers $R_{20}$ and $R_{21}$. And also in page 1581 last paragraph author say “CMT-PF transition path 2 to PF state after first timeout failure is detected", which is not correct, since in page 1581 their timeline diagram shows only the path 1 link is failed during transmission of TSN 2. The timeline scenario has to be discussed with path 1, but the author describes wrongly. Hence experimental study seems bewildered; therefore their results may be uncertain.

**Evaluation with no failures:** In our simulation, we transfer various sizes of files 20, 40 and 60 MB file from the sender A to the receiver B using Path 1, Path 2 and Path 3concurrently with reference to the network simulation topology as shown in Fig. 1. This file transfer uses a single streamed MSACMT-RT association such that all data is delivered in sequence to the receiving application.

The duplicate packets received by the receiver are discarded, but cumulative acknowledgement is sent in the path upon receiving any data packet. The graph is plotted for cumulative Megabytes transferred vs. time in seconds and compared with MSACMT-RT Vs. CMT-PF, by adopting the procedures discussed in last paragraph of litrature review (Natarajan *et al*., 2008) (Yilmaz *et al*., 2010). Fig. 8 is the graph plotted for a 20MB file transfer using MSACMT-RT and CMT-PF. The initial congestion window is set to 2MTU"s (1MTU =1500 Bytes) with a constrained receiver buffer of 128KB. The above case is discussed for paths with no failures during its transmission period. Figure 4-6 is the graph plotted for transfer of 20 and 40 MB file transfer using MSACMT-RT and CMT-PF. Figure 3-5 shows the transfer of 20MB file for rbuf size of 128 and 256KB, evaluation with 128KB file MSACMT-RT increases the transfer time by 9.37% and with 256MB the transfer time increases by 14.06%. Here we understand that larger the buffer size, greater the transfer time decreases. Great difference can be observed while transferring larger files and buffer size.

**Evaluation in failure scenarios:** We simulate the failure scenarios by bringing down one of the Path during the file transfer; this failure is simulated by bringing down the bidirectional link between routers $R_{1,0}$ and $R_{1,1}$ or $R_{2,0}$ and $R_{2,1}$ or $R_{3,0}$ and $R_{3,1}$ at various intervals and the link is reverted at various intervals, which is shown from the simulation topology of Fig. 1.

**Path failures at regular intervals:** Figure 7 shows the time taken for transfer of 20MB file using rbuf size of 128KB with MSACMT-RT and CMT-PF. Failures were induced on path 1 at regular intervals lasting for 5 sec.

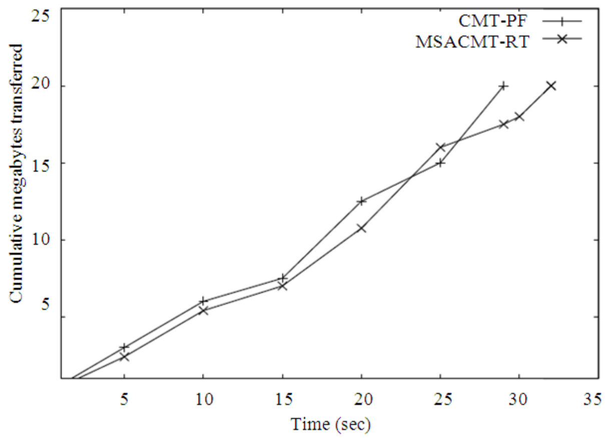

Fig. 3: MSACMT-RT Vs CMT-PF 20 MB file transfer; rbuf= 128 KB

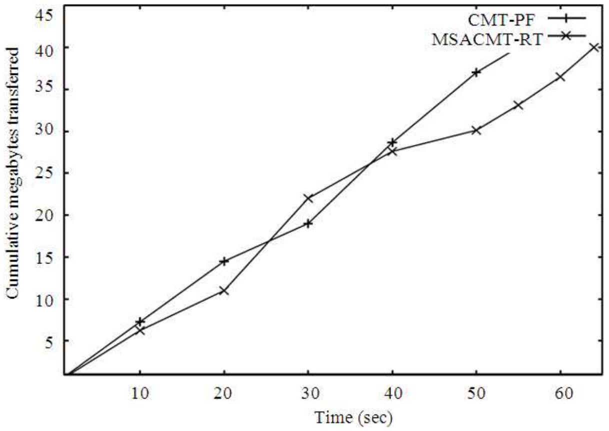

Fig. 4: MSACMT-RT Vs CMT-PF 20MB file transfer; rbuf= 128KB

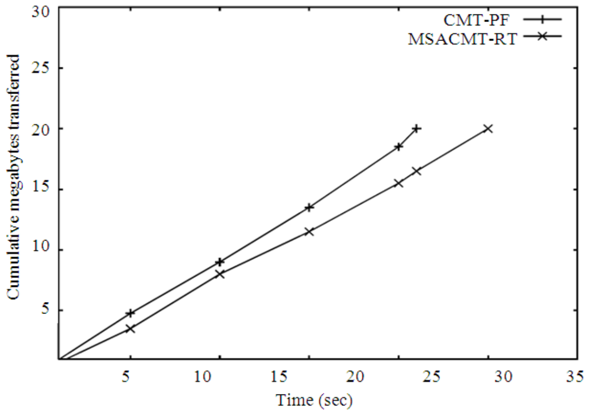

Fig. 5: MSACMT-RT Vs CMT-PF 20MB file transfer; rbuf= 256KB

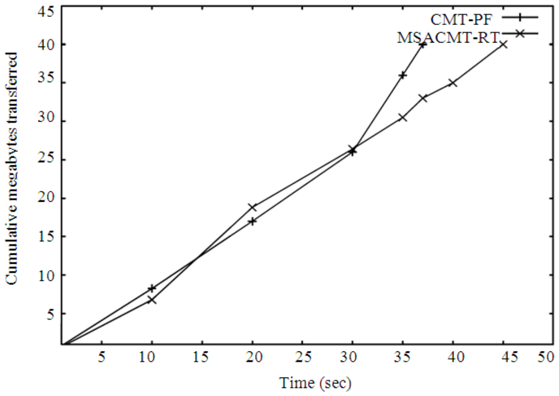

Fig. 6: MSACMT-RT Vs CMT-PF 40MB file transfer; rbuf = 256KB

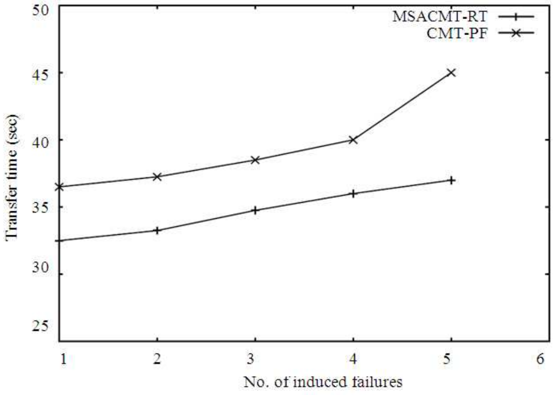

Fig. 7: Path 1 Fails at regular intervals for every 5 sec; failure duration = 5 sec; rbuf = 128KB

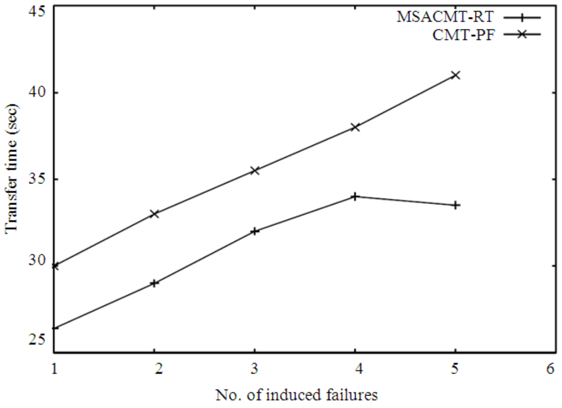

Fig. 8: Path 1 Failure Regular intervals for every 5 sec; failure duration = 5 sec; rbuf = 256KB

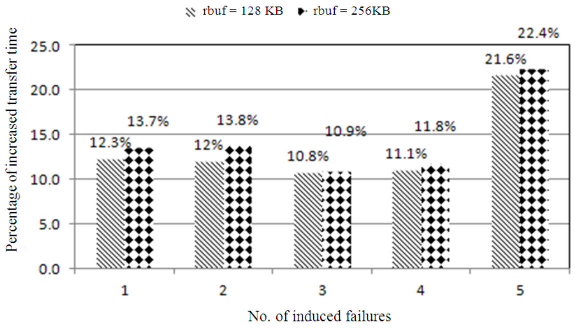

Fig. 9: Percentage of transfer time increased for rbuf = 128, 256KB Vs. No. induced failures

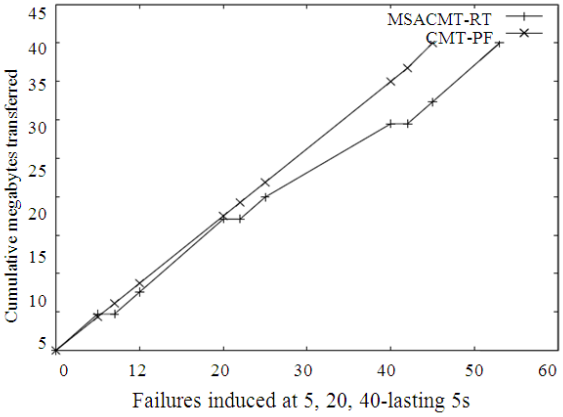

Fig. 10: Path 1 induced failures at irregular intervals 5,20,40s lasting for 5s each;file = 40MB

We induced 1-5 numbers of failures during the file transfer and measured the total time taken for transfer of 20MB file. Similarly for varying rbuf size of 256MB is also evaluated and shown in Fig. 8, both in Fig. 7 and 8 shows that when using MSACMT-RT, file transfer completes much earlier than CMT-PF. Figure 9 shows the percentage of transfer time increased when using rbuf size of 128 and 256KB for 1, 2...5 number of induced failures at regular intervals each lasting for 5s. After 5s break the link is brought active for CMT in all the three paths.

**Path failures at irregular intervals:** Figure 10. shows the time taken for transfer of 420MB file using rbuf size of 128KB with MSACMT-RT and CMT-PF. Failures were induced on path 1 at irregular intervals lasting at each point for 5 sec. Failures are induced at 5, 20 and 40th sec which is considered to be irregular intervals. It is inferred that MSACMT-RT do not get affected much and also transfers the file much earlier than CMT-PF. File transfer time is increased by 17.78%. We also transferred with several file sizes and on varying buffer size. We conclude that for larger file size with larger buffer size the performance seems to be better and the failures induced has not affected the file transfer.

## CONCLUSION

In this study we proposed MSACMT-RT. We demonstrated using simulation. MSCMT-RT, before the data transfer begins from source to destination the status of the independent network path is examined. After keen examination priority of the path is assigned and data transfer takes place. Path having least priority is used for redundant transmission for certain period. This is called multipath state aware CMT using redundant transmission.

The Investigation resulted MSACMT-RT outperforms better, improving overall throughput and less delay. This experiment also completes its data transfer much earlier than CMT-PF. We have tested the experiment both in non-failure and failure scenarios. There is a progressive growth in the performance of throughput for larger files. During failures, with varying intervals and regular intervals also increased transfer time from 10-22% compared with CMT-PF. MSACMT-RT transfer time increase by 17.8% when failure occur during irregular intervals. Hence MSACMT-RT claims to have better throughput and faster file transfer time. And our further research work is still in progress.

## REFERENCES

Adhari, H., T. Dreibholz, M. Becke, E.P. Rathgeb and M. Tuxen, 2011. Evaluation of concurrent multipath transfer over dissimilar paths. Proceedings of Workshops of International Conference on Advanced Information Networking and Applications, 22-25 March, IEEE XPloor, Biopolis pp: 708-14, DOI: 10.1109/WAINA.2011.92

Al-Talib, S.A., B.M. Ali and S. Khatun, 2009. An approach to improve the state scalability of source specific multicast. Am. J. Applied Sci., 6: 1347-1351. DOI: 10.3844/ajassp.2009.1347.1351.PDF:

Caro, A., P. Amer and R. Stewart, 2006. Retransmission policies for multihomed transport protocols. Comput. Communi., 29: 1798-810, DOI: 10.1016/j.comcom.2005.10.017

Caro, A., 2006. End-to-End Fault Tolerance using Transport Layer Multihoming. PhD Dissertation, CIS Dept, University of Delaware, 08/05. Dreibholz, T., M. Becke, E.P. Rathgeb and M. Tuxen, 2010. On the use of concurrent multipath transfer over asymmetric paths. Proceeding of IEEE Communications Society subject matter experts for publication in the IEEE Globecom.pp: 1-6.

Yilmaz, E., N. Ekiz, P. Natarajan, P.D. Amer and J.T. Leighton *et al*., 2010. Throughput analysis of non-renegable selective acknowledgments (NR-SACKs) for SCTP. J. Comput. Communi., 33: 1982-1991. DOI: 10.1016/j.comcom.2010.06.028

Trong, H.C., S. Lee and C.S. Hong, 2010. End-to-end throughput improvement for single radio multi-channel multi-path wireless mesh networks. A cross layer design. Anna. Telecommuni., 65: 635-646. DOI 10.1007/s12243-010-0178-y

Iyengar, J.R, P.D. Amer and R. Stewart, 2004. Retransmission policies for concurrent multipath transfer using SCTP multihoming. Proc. 12th IEEE Int. Conf. Networks. 2: 713-719, DOI: 10.1109/ICON.2004.1409269

Iyengar, J.R, P.D. Amer and R. Stewart, 2005. Receive buffer management for concurrent multipath transport using SCTP multihoming. Porceeding of the IEEEGlobal Telecommunications Conference, GLOBECOM ‘05. 28 Nov. pp: 25-40, IEEE XPloor, USA. DOI: 10.1109/GLOCOM.2005.1577365

Iyengar, J.R., P.D. Amer and R. Stewart, 2006. Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths. J. IEEE/ACM Trans. Networking, 14: 951-964. DOI: 10.1109/TNET.2006.882843

Jiemin, L., H. Zou, J. Dou and Y. Gao, 2008.Reducing Receive Buffer Blocking In Concurrent Multipath Transfer. proceeding of the 4th IEEE International Conference on Circuits and Systems for Communications. 26-28 May’08, IEEE XPloor, Shanghai, pp: 367-71. DOI: 10.1109/ICCSC.2008.85

Natarajan, P., N. Ekiz, E. Yilmaz, P.D. Amer and J. Iyengar, 2008. Non-renegable selective acknowledgments (NR-SACKs) for SCTP. Proceedings of the 16th IEEE International Conference on Network Protocols (ICNP). Oct. 2008, Orlando, Florida/U.S.A, pp: 187-196, DOI: 10.1109/ICNP.2008.4697037

Natarajan, P., N. Ekiz, P. Amer and R. Stewart, 2009. Concurrent multipath transfer during path failure. Comput. Communi., 32: 1557-87. DOI:10.1016/j.comcom.2009.05.001

Gao, Q.L., J. Brown and L.F. Capretz, 2004. Extending UML-RT for control system modeling. Am. J. Applied Sci., 1: 338-347.DOI: 10.3844/ajassp.2004.338.347.PDF:

Stewart, R.R. and Q. Xie, 2002. Stream Control Transmission Protocol (SCTP): A Reference Guide. 1st Edn. Addison-Wesley, New York, ISBN-10: 0201721864, pp: 351.