Performance Evaluation of Distributing Real-time Video over Concurrent Multipath

Changqiao Xu[3,2,1] Enda Fallon[3] Yuansong Qiao[3,1]
[1]Institute of Software [2]Graduate University
Chinese Academy of Sciences
Beijing, China

Gabriel-Miro Muntean
School of Electronic Engineering
Dublin City University
Dublin, Ireland

Xiaoguang Li Austin Hanley
[3]Software Research Centre
Athlone Institute of Technology
Athlone, Ireland

Abstract—**Recent research on Concurrent Multipath Transfer (CMT) and CMT with a Potentially-Failed destination state (CMT-PF) uses the transport layer multi-homing protocol Stream Control Transmission Protocol (SCTP) to increase application throughput by distributing transmitted data across multiple end-to-end paths. This paper investigates and evaluates the performance of CMT with Partial Reliability (CMT-PR) and CMT-PF with Partial Reliability (CMT-PF-PR), novel extensions of SCTP for real-time video distribution. The Evalvid-CMT platform was implemented in the University of Delaware's SCTP/CMT ns-2 module to perform emulation experiments in order to compare CMT and CMT-PR, CMT-PF and CMT-PF-PR, respectively. The results presented in the paper show how the CMT-PR and CMT-PF-PR outperform CMT and CMT-PF respectively. Consequently the former are suggested as strategies for real-time video concurrent multipath transmissions.**

Keywords-CMT; CMT-PF; CMT-PR; CMT-PF-PR; Evalvid-CMT

I. INTRODUCTION

It becomes increasingly common for a wireless device to be connected to more than one access networks employing either a homogeneous technology or heterogeneous forms of access such as GPRS, 3G, WiFi, WiMax, etc. The characteristics of mobile environments, with the possibility of frequent disconnections and fluctuating bandwidth, pose significant issues for mobile application developers and therefore the path redundancy offered by multi-homing protocols has a clear attraction. The multi-homing features of the Stream Control Transmission Protocol (SCTP) [1] appear to be key enablers for improving mobile communications [2]. Multihoming technologies, where a host can be addressed by multiple IP addresses, achieves definite improvements when link/path failures occur or temporary loss is experienced. Especially, the mobile extension of SCTP (mSCTP) [3], enables a more flexible way to reconfigure connection paths by allowing new source/destination IP addresses to be added to an ongoing SCTP association on the fly and invoked for reasons other than link failure, thus providing an elegant method to support terminal mobility across different networks.

SCTP provides a good framework to support multi-homing. Its original perspective was resilience as it was designed to address the link failure scenario by allowing alternate paths to be associated with a connection during its establishment. SCTP uses multiple interfaces only for redundancy. A host chooses a primary destination address, normally used for the data transmission, whereas the alternate addresses are considered as secondary, whose conditions are periodically monitored with the transmission of probe chunks called Heartbeat. The backup path is used only to retransmit lost data chunks for increasing the probability of successful retransmissions, or to transmit new data chunks temporarily if there is consecutive timeouts ("inactive") on the primary path. As soon as the Heartbeats reception on the primary interface is confirmed ("active"), the transmission is resumed into primary path again.

Recent many works focused on the Concurrent Multipath Transfer (CMT) [4]-[7] of SCTP. CMT is designed to use all the available paths for the association at the same time, instead of using only the primary path. Therefore it can provide full multi-homing support through the simultaneous utilization of all available paths, which achieves load sharing and increases an application's throughput.

The authors of [4] and [5] explored the receive buffer (rbuf) blocking problem in CMT, where Transport Protocol Data Unit (TPDU) losses throttle a sender once the transport receiver's buffer is filled with out-of-order data. Even though the congestion window would allow new data transmission, rbuf blocking stalls the sender, causing throughput degradation. To reduce rbuf blocking effects during congestion, [4] and [5] proposed different retransmission policies that use heuristics for faster loss recovery. To mitigate rbuf blocking during path failures, [6] and [7] modified CMT's failure detection process to include a "potentially-failed" (PF) destination state calling the resulting solution CMT-PF. A PF destination is not used for data transmission or retransmission. Only Heartbeats are sent to the PF destination. If a Heartbeat ACK returns, the PF destination returns to active state.

The growing availability of different wireless access technologies provides the opportunity for real-time distribution of multimedia content using parallel transfer mechanism in particular. End-to-end delay and packet loss are vital QoS requirements for multimedia applications. In order to investigate the behavior and quality of such applications under concurrent multipath transfer and heavy network load, it is therefore necessary to create genuine traffic patterns, both at network/transport layer and application. To setup a true multimedia parallel transfer network with multi-homing feature is a challenge and of little flexibility. Thus, networking simulations, using tools like the ns-2 seem tempting. A local

978-1-4244-2948-6/09/$25.00

simulation environment enables the researcher to build customized effective networks at a low cost.

In this paper, we have implemented a platform called Evalvid-CMT in University of Delaware's SCTP/CMT ns-2 module [8]. Evalvid-CMT gives a good solution to simulate the real-time video transmission over SCTP concurrent multipath, and it realizes the video quality evaluation after simulation.

Based on the Evalvid-CMT platform, we investigate and evaluate the performance of CMT with Partial Reliability (CMT-PR) and CMT-PF with Partial Reliability (CMT-PF-PR), novel extensions of SCTP for real-time video distribution. As multimedia is associated with time sensitive (real-time) applications, consideration of end-to-end delay and jitter becomes a major issue when using the transport protocol for concurrent multipath transmission. As the results presented in the paper show how the CMT-PR and CMT-PF-PR outperform CMT and CMT-PF respectively, the former are strongly suggested as strategies to be used for multimedia distribution.

The rest of the paper is organized as follows. We introduce the related work in section II, the CMT-PR and CMT-PF-PR polices are presented in section III. In section IV, we discuss the design of the Evalvid-CMT platform. The performance simulations are presented in section V. Section VI gives the conclusions.

II. RELATED WORK

A. *Concurrent Multipath Transfer (CMT)*

A transport layer receiver maintains a single rbuf which is shared across the sub-association flows. The rbuf is used to handle out-of-order data and receive data at a rate higher than that of the receiving application's consumption. [4] demonstrated how a shared rbuf results in a sub-association flow on a higher quality path getting lower throughput than expected. Multiple paths present a sender with a choice where to send a retransmission of a lost transmission. With CMT, new data is being sent to all destinations concurrently, [5] shows that rbuf blocking cannot be eliminated; it can be reduced by choice of retransmission policy. The retransmission policies are: (1) RTX-SAME - Once a new data chunk is scheduled and sent to a destination, all retransmissions of the chunk are sent to the same destination (until the destination is deemed inactive due to failure). (2) RTX-ASAP - A retransmission of a data chunk is sent to any destination for which the sender has cwnd space available at the time of retransmission. If multiple destinations have available cwnd space, one is chosen randomly. (3) RTX-CWND - A retransmission is sent to the destination for which the sender has the largest cwnd. A tie is broken by random selection. (4) RTX-SSTHRESH - A retransmission is sent to the destination for which the sender has the largest ssthresh. A tie is broken by random selection. (5) RTX-LOSSRATE - A retransmission is sent to the destination with the lowest loss rate path. If multiple destinations have the same loss rate, one is selected randomly.

B. *CMT with Potentially-Failed State (CMT-PF)*

To mitigate the recurring instances of rbuf blocking, [6] and [7] 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.

III. CMT-PR AND CMT-PF-PR POLICIES

The Partial-Reliable SCTP (PR-SCTP) [9] is an unreliable data mode extension of SCTP, PR-SCTP allows an SCTP sender to assign different levels of reliability to data so that lost data can be retransmitted until a predefined reliability threshold is reached. When the reliability threshold is reached for unacknowledged data, the sender abandons that retransmission of the data and notifies the receiver (with Forward TSNs) to neglect the outstanding data and move the cumulative ACK point forward.

For streaming multimedia content over SCTP, the reliability and thereby the retransmission of missing data could be the bottleneck for the guarantee of a desired frame rate. Sometimes it is necessary to drop a few missing packets, instead of waiting for them to be retransmitted, because they are no longer needed for the stream.

In this paper, we investigate and evaluate considerations in implementing CMT and CMT-PF with Partial Reliability extension of SCTP, which called CMT-PR (CMT-Partial Reliability) and CMT-PF-PR for real-time video distribution.

The following variations are considered for evaluations between CMT with CMP-PR: (1) CMT-PR(0/1/2)-ASAP - CMT with RTX_ASAP retransmission policy and reliability threshold setting to 0, 1, 2 respectively. (2) CMT-PR(0/1/2)-SAME - CMT with RTX_SAME retransmission policy and reliability threshold setting to 0, 1, 2 respectively. (3) CMT-PR(0/1/2)-SSTHRESH - CMT with RTX_SSTHRESH retransmission policy and reliability threshold setting to 0, 1, 2 respectively. (4) CMT-PR(0/1/2)-LOSSRATE - CMT with RTX_LOSSRATE retransmission policy and reliability threshold setting to 0, 1, 2 respectively. (5) CMT-PR(0/1/2)-

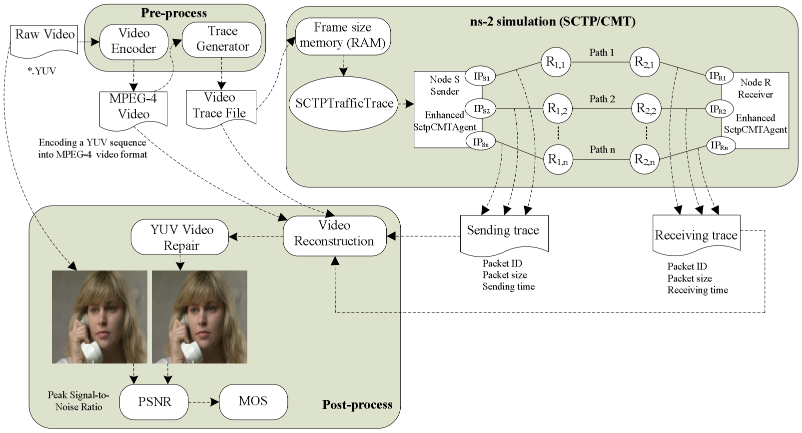

Figure 1. Evalvid-CMT framework

CWND - CMT with RTX_CWND retransmission policy and reliability threshold setting to 0, 1, 2 respectively.

The following variations are considered for evaluations between CMT-PF with CMP-PF-PR: (1) CMT-PF(1/2)-PR(0/1/2)-SSTHRESH - CMT with RTX_SSTHRESH retransmission policy, PF destination's cwnd setting to 1 or 2 MTU when it is alive, and reliability threshold setting to 0, 1, 2 respectively. (2) CMT-PF(1/2)-PR(0/1/2)-CWND - CMT with RTX_CWND retransmission policy, PF destination's cwnd setting to 1 or 2 MTU when it is alive, and reliability threshold setting to 0, 1, 2 respectively.

We will evaluate and compare above different policies for MPEG-4 video transmission with SCTP/CMT by using Evalvid-CMT platform. Evalvid-CMT will give the simulation results and accomplish the video quality evaluation.

IV. EVALVID-CMT

The quality of a video transmission depends on the impression a human observer receives of the delivered video. Measures of quality can be divided into two categories. The first category contains subjective measures, where individuals assess the quality on the basis of their personal experience. One of the most common subjective measure is created through a Mean Opinion Score (MOS) approach. The second category of measures of quality contains objective measures, where an algorithm is used to calculate a value of the measure. Note that the object measurements are repeatable, that is the measurement of quality can be repeated by someone else and the same results should be expected. The basis of many objective measures is the Peak Signal to Noise Ratio (PSNR).

In recent years, papers have been published on the topic of the simulation of multimedia. In [10], MPEG-4 trace files are used to calibrate a TES (Transform Expand Sample) mathematical model, and rate adaptation is incorporated by adjusting the frame size output by a scalar (from rate-distortion curve). H.263 video trace files are used in [11], and the sending rate is controlled by DCCP TCP-like. Authors of [12] proposed a tool-set called Evalvid-RA to support rate adaptive MPEG-4 VBR video simulation. Evalvid tools-set [13] is an open-source project, and supports trace file generation of MPEG-4 as well as H.263 and H.264 video. Using Evalvid together with the ns-2 interface code suggested by C.-H. Ke [14], perceived quality and objective measure like PSNR calculation can be obtained after network simulation.

In our previous work [15], we proposed a platform called Evalvid-SCTP, which achieved trace driven simulation of MPEG-4 video transmission over SCTP network. Evalvid-SCTP was used to study and analyze the performance of multimedia transmission over SCTP implementing single path mechanism. Based on our previous work and above related work, we have implemented Evalvid-CMT platform in University of Delaware's SCTP/CMT ns-2 module [8].

Evalvid-CMT enables simulation of multimedia concurrent multipath transfer based on the generation of MPEG-4 video trace files. The trace files consist of real compressed video characteristics including frame number, frame type, size, fragmentation into segments and timing for each video frame. These characteristics can be utilized to construct mathematical traffic models and traffic generators for network simulators since they determine the packet sizes and time schedules. The simulation utilises pre-generated media trace files. While running SCTP/CMT, Evalvid-CMT records packet throughput at each node including the receiver. Using this information and the original compressed video file, Evalvid-CMT reconstructs the video as if it was received on a real network. This reconstruction enables the video to be inspected visually as well as allowing for the calculation of PSNR and MOS for the transferred video. As figure 1 shows, the framework of Evalvid-CMT has three processes namely, pre-process, ns-2 simulation and post-process. In pre-process, the original YUV format video is compressed into MPEG-4 format video, then video trace file is generated which includes information about each frame (I/P/B-frame) in the video. In the ns-2 simulation, we designed an ns-2 Agent called SCTPTrafficeTrace to send data to SCTP/CMT network following video trace file. We modified and enhanced SctpCMTAgent to record the sender trace file including sending time, packet type, packet id and size, and records the receiver trace file including receiving time, packet type, packet id and size. In the post-process phase, the video can be reconstructed and converted in to the YUV video, and then the video quality evaluation can be given by the calculation of PSNR and MOS, compared with the original raw video.

V. SIMULATION-BASED ASSESSMENT

A. *Simulation Setup*

The work of [4] shows that if two paths are used for CMT, the lower quality (i.e., higher loss rate) path degrades overall throughput of an rbuf-constrained CMT association by blocking the rbuf. In order to investigate the behavior and quality of real-time video concurrent multipath transmission better, we focus on this scenario of asymmetric path condition.

The simulation topology is shown in figure 2 and includes node S and node R which are the SCTP sender and receiver respectively. Both SCTP endpoints have two addresses. $R_{1,1}$, $R_{1,2}$, $R_{2,1}$ and $R_{2,2}$ are routers. The implementation is configured with no overlap between the two paths. The MTU of each path is 1500B. The queue length of bottleneck links in both paths is 50 packets. The queue length of other links is set to 10000 packets. Default rbuf values in commonly used operating systems today vary from 16KB to 64KB and beyond. We study and analyze performance of the different policies with an rbuf of 64KB.

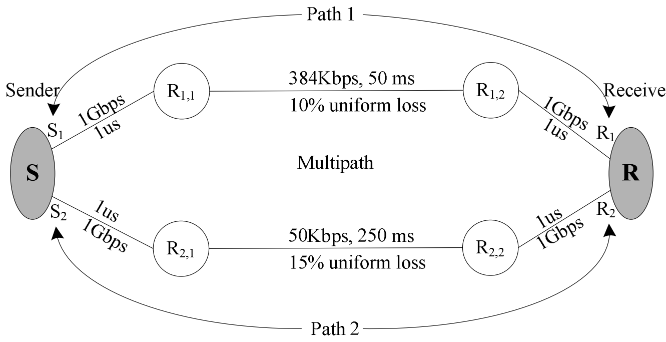

Figure 2. Simulation network topology

In the simulations, wireless link parameters are used as the references for network configurations. A computing node has a hybrid of GPRS, 3G connections. The path 1's bandwidth is 384Kbps and the bandwidth of secondary path is 50Kbps. The delay on the path 1 is 50ms, and the delay of secondary path is 250ms. The path 1 and path 2 experience 10% and 15% loss rate with Bernoulli Loss Model respectively.

A YUV video sequence is used with a QCIF format (resolution 176x144 pixels) and 2000 frames. After pre-processing in Evalvid-CMT, a MPEG-4 video trace file is produced. In ns-2 simulation, node S begins to send data from this video trace file at 30 frames/second to node R at t=5 second. The simulation will stop after 5+1.0*1000/30*(2000+1)/1000= 71.7 seconds. 10 random seeds are used for simulation and testing results are calculated by averaging the results of 10 runs.

B. *CMT vs CMT-PR*

This section studies the performance between CMT with CMT-PR (CMT-Partial Reliability). [15] shows that less aggressive failover settings is preferred for asymmetric path condition. In this simulation, the Path.Max.Retrans (PMR) is set 1. The results for PMR=2, 3, 4, 5 are not shown in this paper. The reason for this omission is that the path failure detection time is long, such as for PMR=5, it means that SCTP needs 6 consecutive transmission timeouts to detect path failure. RTO will be doubled for each transmission timeout and ranges between the SCTP parameters RTO.Min and RTO.Max. The default values for RTO.Min and RTO.Max are 1s and 60s respectively. If RTO is 1s (RTO.Min) in the case of a path failure, the minimum time for detecting path failure is 1+2+4+8+16+32=63s. It is not reasonable for real-time video transmission which is time sensitive.

Table I and figure 3 show the comparison results of average PSNR (dB) values and the numbers of different dropped frames (I-frame/P-frame/B-frame), which employed CMT and CMT-PR policies respectively. The dropped frames are those frames which can not be decoded after transmission. They may be the lost frames with network transfer or the discarded frames with the delay/jitter handling. Three reliability thresholds with 0, 1, and 2 are simulated for CMT-PR. As the table and figure show, in most cases, CMT-PR performs better than CMT with five different retransmission policies, except the case of CMT-PR0 (the reliability threshold is 0). Such as the average PSNR of CMT-SSTHRESH is just only 27.15, but the average PSNRs of CMT-PR1-SSTHRESH and CMT-PR2-SSTHRESH can arrive at 35.72 and 32.52 respectively. The number of total dropped frames of CMT-SSTHRESH is 983, but there are no dropped frames for CMT-PR1-SSTHRESH and only 436 dropped frames for CMT-PR2-SSTHRESH.

TABLE I. POST-PROCESSING RESULTS OF CMT VS CMT-PR

Policies	**Average PSNR(dB)**	**Dropped Frames**			
		Total	***I***	***P***	***B***
CMT-ASAP	35.35	57	6	13	38
CMT-PR0-ASAP	33.21	355	40	79	236
CMT-PR1-ASAP	33.79	256	29	57	170
CMT-PR2-ASAP	35.72	1	0	0	0
CMT-SAME	34.36	185	21	41	123
CMT-PR0-SAME	31.96	517	58	115	344
CMT-PR1-SAME	33.21	355	40	79	236
CMT-PR2-SAME	34.5	177	20	39	118
CMT-SSTHRESH	27.15	983	109	219	655
CMT-PR0-SSTHRESH	33.21	355	40	79	236
CMT-PR1-SSTHRESH	35.72	0	0	0	0
CMT-PR2-SSTHRESH	32.52	436	49	97	290
CMT-LOSSRATE	33.83	239	27	53	159
CMT-PR0-LOSSRATE	33.21	355	40	79	236
CMT-PR1-LOSSRATE	35.7	4	1	1	2
CMT-PR2-LOSSRATE	34.6	150	17	33	100
CMT-CWND	34.53	172	19	39	114
CMT-PR0-CWND	33.21	355	40	79	236
CMT-PR1-CWND	35.55	27	3	6	18
CMT-PR2-CWND	35.44	43	5	10	28

The simulation of [4] [5] show the performance of RTX-ASAP is less than RTX-SSTHRESH, RTX-LOSSRATE and RTX-CWND for file distribution. However, in our simulation, there are some interesting results, if partial reliability mechanism did not be implemented, only five retransmission policies are considered and compared, the average PSNR of

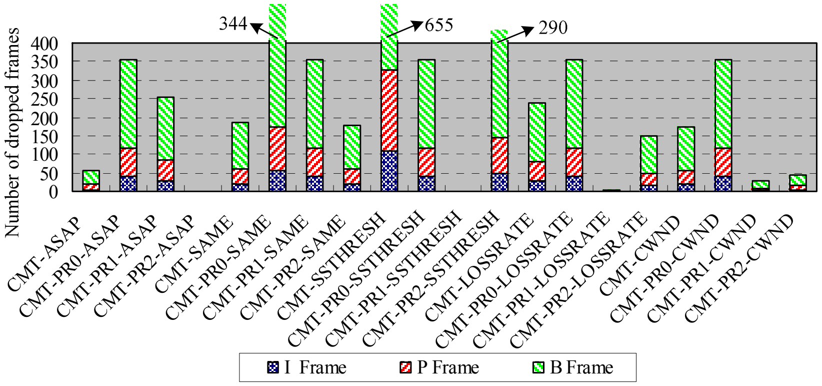

Figure 3. Number of dropped frames for CMT vs CMT-PR

RTX-ASAP can achieve 35.35, and the number of total dropped frames is just only 57, RTX-ASAP performs best. RTX-ASAP is a “hot-potato” policy, which retransmit data unit as soon as possible. This mechanism of RTX-ASAP is helpful for time sensitive video transmission.

The performance of CMT-PR0 is unstable, such as the performance of CMT-PR0-SSTHRESH (33.21) is better than CMT-SSTHRESH (27.15), but CMT-PR0-ASAP (33.21) performs worse than CMT-ASAP (35.35). In most cases, the performance of CMT-PR0 is less than CMT-PR1 and CMT-PR2 with different retransmission policies. CMT-PR1 performs better than CMT-PR2. CMT-PR1-SSTHRESH is the best strategy in all of policies between CMT and CMT-PR.

C. *CMT-PF vs CMT-PF-PR*

The authors of [7] considered and recommended RTX_SSTHRESH and RTX_CWND variants of CMT and CMT-PF, so in this section, we evaluate CMT-PF-PR (CMT-PF Partial Reliability) with RTX_SSTHRESH and RTX_CWND retransmission policies respectively. We also investigate the impact of PF destination’s cwnd with setting to 1 or 2 MTU respectively when PF destination is alive. The other simulation parameters are set same with above experiment.

Table II and figure 4 illustrate the comparison results for average PSNR (dB) values as well as dropped frames (I-frame/P-frame/B-frame) between CMT-PF and CMT-PF-PR. As the table and the figure show, in all the cases, the performance of CMT-PF-PR is better than CMT-PF with both of RTX_SSTHRESH and RTX_CWND retransmission policies respectively (CMT-PF-PR0 is an exception). Such as the average PSNR of CMT-PF1-SSTHRESH is 22.14, the number of total dropped frames arrive at 1669, however the average PSNRs of CMT-PF1-PR1-SSTHRESH and CMT-PF1-PR2-SSTHRESH are 35.47 and 33.62, and the number of total dropped frames are only 39, 269 respectively. The performance of CMT-PF-PR0 is unstable, such as the average PSNR of CMT-PF1-PR0-SSTHRESH and CMT-PF2-PR0-SSTHRESH are 33.07 and 32.80 respectively, and both of them are better than CMT-PF1-SSTHRESH and CMT-PF2-SSTHRESH. However, for CMT-PF1-PR0-CWND and CMT-PF2-PR0-CWND, the average PSNR of them are 33.07 and 32.80 respectively, which are worse than CMT-PF1-CWND (35.72) and CMT-PF2-CWND (35.02). But compared with CMT-PF-PR1 and CMT-PF-PR2, the performance of CMT-PF-PR0 is always worse.

TABLE II. POST-PROCESSING RESULTS OF CMT-PF VS CMT-PF-PR

Policies	**Average PSNR (dB)**	**Dropped Frames**			
		Total	***I***	***P***	***B***
CMT-PF1-SSTHRESH	22.14	1669	186	371	1112
CMT-PF1-PR0-SSTHRESH	33.07	370	41	83	246
CMT-PF1-PR1-SSTHRESH	35.47	39	4	9	26
CMT-PF1-PR2-SSTHRESH	33.62	269	30	60	179
CMT-PF2-SSTHRESH	22.14	1669	186	371	1112
CMT-PF2-PR0-SSTHRESH	32.80	400	45	89	266
CMT-PF2-PR1-SSTHRESH	33.34	330	37	73	220
CMT-PF2-PR2-SSTHRESH	33.65	274	31	61	182
CMT-PF1-CWND	35.72	1	0	1	0
CMT-PF1-PR0-CWND	33.07	370	41	83	246
CMT-PF1-PR1-CWND	35.72	0	0	0	0
CMT-PF1-PR2-CWND	35.72	1	0	1	0
CMT-PF2-CWND	35.02	103	12	23	68
CMT-PF2-PR0-CWND	32.80	400	45	89	266
CMT-PF2-PR1-CWND	35.09	92	10	21	61
CMT-PF2-PR2-CWND	35.71	3	0	1	2

For investigating the impact of PF destination’s cwnd with different size when PF destination is alive, there are interesting results, as the table II shows, CMT-PF1 performs better than CMT-PF2, which is different with the simulation results of [7] for file transfer. In CMT-PF, no data is sent on the PF path, after the timeout. CMT-PF sends a heartbeat on PF path and retransmits lost TPDUs along with new data on other active paths. PF path is marked active when the heartbeat ack arrives at the sender. Therefore, for CMT-PF1, at the end of 1 Round Trip Time (RTT) after the timeout, (i) congestion window on PF path is 1 MTU and (ii) no new data has been sent on PF path. Thus, CMT-PF1 has a 1 RTT “lead” in its congestion window evolution. CMT-PF2 which initializes the congestion window to 2 MTUs after a heartbeat ack arrives. Though it avoids the 1 RTT lag in CMT-PF1’s congestion window evolution, for the asymmetric path condition, because one path is worse than another one, CMT-PF1 let the better conditions path has more chance for use, which achieves better real-time video transmission performance than CMT-PF2. CMT-PF-PR1 performs more stable than CMT-PF-PR2, and CMT-PF1-PR1-CWND performs best in all policies.

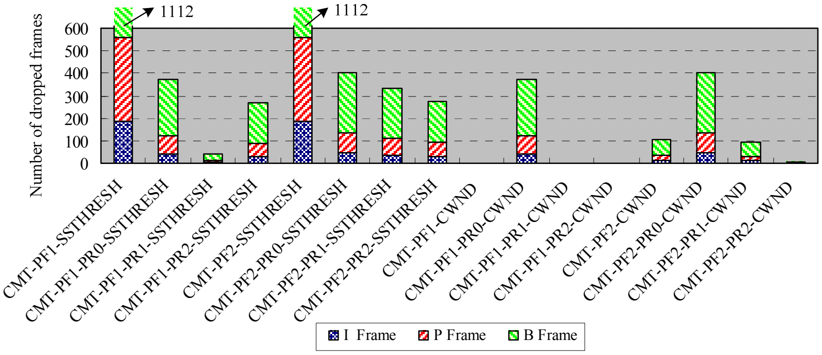

Figure 4. Number of dropped frames for CMT-PF vs CMT-PF-PR

Both of the simulation results of subsection B and C show that for distributing real-time video over concurrent multipath, complete reliability, excessive reliability or no reliability will degrade the quality of video for transmission. The policy with no reliability does not retransmit any data unit, which is an extreme strategy and not reasonable.

The simulation results show that CMT-PR1 and CMT-PF-PR1 performs more stable than CMT-PR2 and CMT-PF-PR2 respectively, it is because fast retransmission is used for the former. Since recovery via fast retransmission can happen only once for a given TSN, with other partial reliability thresholds, the number of consecutive timeouts would be high via the timeout retransmission, then the complete or excessive reliability of the retransmission of missing data could be the bottleneck for the guarantee of a desired frame rate. For real-time video, sometimes it is necessary to drop a few missing packets, instead of waiting for them to be retransmitted, because they are no longer needed for the stream.

The simulation results illustrate that CMT-PR1-SSTHRESH and CMT-PF1-PR1-CWND perform best with CMT-PR and CMT-PF-PR respectively. As it is discussed in [5], using RTX-SSTHRESH ensures that most of the data is sent over the better path with CMT, and RTX_CWND appears to be a better policy than RTX_SSTHRESH during failure with CMT-PF [7]. Combining the analysis of PF destination's cwnd and our real simulation results, CMT-PR1-SSTHRESH and CMT-PF1-PR1-CWND are strongly suggested as the strategies to be used for multimedia distribution with parallel transfer.

VI. CONLUSIONS

This paper investigates the performance of real-time multimedia transmission over multi-homing transport protocols utilizing concurrent multipath transfer (CMT). CMT with Partial Reliability (CMT-PR) and CMT-PF with Partial Reliability (CMT-PF-PR) extensions of SCTP were realised and an Evalvid-CMT platform implemented in University of Delaware's SCTP/CMT ns-2 module was used to perform emulation experiments. These aimed at comparing CMT and CMT-PR, CMT-PF and CMT-PF-PR, respectively. The simulation results show that CMT-PR and CMT-PF-PR perform much better than CMT and CMT-PF respectively for video transmissions. CMT-PR1-SSTHRESH and CMT-PF1-PR1-CWND are strongly recommended as the schemes to distribute real-time video over concurrent multipath.

REFERENCES

[1] R. Stewart, "Stream Control Transmission Protocol," IETF RFC 4960, September 2007.

[2] A. Dutta, S. Das, D. Famolari, Y. Ohba, K. Taniuchi, T. Kodama, H. Schulzrinne, "Seamless Handoff across Heterogeneous Networks - An 802.21 Centric Approach," Proc. of IEEE International Symposium on Wireless Personal Multimedia Communications (WPMC), Aalborg Denmark, September 2005.

[3] M. Riegel, M. Tuexen, "Mobile SCTP," IETF Draft, July 2005, draft-riegel-tuexen-mobile-sctp-05.txt.

[4] J. Iyengar, P. Amer, R. Stewart, "Concurrent Multipath Transfer using SCTP Multihoming over Independent End-to-end Paths," IEEE/ACM Transaction on Networking, Oct. 2006, 14(5): 951-964.

[5] J. Iyengar, P. Amer, R. Stewart, "Performance Implications of Receive Buffer Blocking in Concurrent Multipath Transfer," Computer Communications, Feb 2007, 30(4): 818-829.

[6] P. Natarajan, N. Ekiz, P. Amer, J. Iyengar, R. Stewart, "CMT using SCTP Multihoming: Transmission Policies using a Potentially-failed Destination State," TR 2007/338, CIS Dept, University of Delaware, Feb. 2007.

[7] P. Natarajan, N. Ekiz, P. Amer, J. Iyengar, R. Stewart, "Concurrent Multipath Transfer using SCTP Multihoming: Introducing the Potentially-failed Destination State," NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet, LNCS Springer, Singapore, May, 2008.

[8] A. Caro, J. Iyengar, "ns-2 SCTP Module," Version 3.6, June 2006. http://pel.cis.udel.edu.

[9] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen, P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension," IETF RFC 3758, May 2004.

[10] C. H. Liew, C. Kodikara, A. M. Kondoz, "Modelling of MPEG-4 Encoded VBR Video Traffic," IEE Electronic Letters, March 2004: 40(5).

[11] C. Xu, J. Liu, C. Zhao, "Performance analysis of transmitting H.263 over DCCP," Proc. of IEEE International Workshop VLSI Design and Video Technology, May 2005.

[12] A. Lie, J. Klaue, "Evalvid-RA: Trace Driven Simulation of Rate Adaptive MPEG-4 VBR Video," ACM/Springer Multimedia Systems Journal, Nov. 2007, 14(1): 33-50.

[13] J. Klaue, B. Rathke, A. Wolisz, "Evalvid - A Framework for Video Transmission and Quality Evaluation," Proc. of the 13th International Conference on Modeling Techniques and Tools for Computer Performance Evaluation, Urbana, Illinois, USA, Sept. 2003.

[14] C.-H. Ke, "How to evaluate MPEG video transmission using the NS2 simulator," [Online]. Available: http://hpds.ee.ncku.edu.tw/smallko/ns2/Evalvid in NS2.htm.

[15] C. Xu, Y. Qiao, E. Fallon, G.-M. Muntean, P. Jacob, A. Hanley, "Comparative Study of Real-Time Multimedia Transmission over Multihoming Transport Protocols," Proc. of 11th IFIP/IEEE International Conference on Management of Multimedia and Mobile Networks and Services (MMNS), Springer LNCS, September 2008: 64-76.