Comparative Study of Real-Time Multimedia Transmission over Multi-homing Transport Protocols

Changqiao Xu[1,2], Yuansong Qiao[1,2], Enda Fallon[1], Gabriel-Miro Muntean[3], Paul Jacob[1], and Austin Hanley[1]

[1] Athlone Institute of Technology, Athlone, Ireland
[2] Institute of Software, Chinese Academy of Sciences, Beijing, China
[3] School of Electronic Engineering, Dublin City University, Dublin, Ireland
{cqiaoxu,ysqiao,efallon}@ait.ie, munteang@eeng.dcu.ie,
{pjacob,ahanley}@ait.ie

Abstract. The availability of multimedia applications suitable for deployment using 3G and GPRS networks has led to a requirement for end-to-end quality of service. More efficient mechanisms are needed in order to provide the required end user quality of service in wireless data networks. This paper investigates the performance implications of transmitting real-time multimedia content over a multi-homed transport protocol in a manner which is tolerant of network failure. It evaluates video quality with different retransmission policies combined with various path failure detection thresholds, path bandwidths, delays and loss rate conditions through Partial Reliable Stream Control Transmission Protocol (PR-SCTP). A solution called Evalvid-SCTP, which is a trace driven simulation of MPEG-4 video over SCTP, was designed to achieve the performance evaluation.

Keywords: PR-SCTP, Multi-homing, MPEG-4 Video, Congestion Control Algorithm, Network simulation.

1 Introduction

In recent years the number of mobile and wireless networks available to devices have increased to the extent that there is a near pervasive deployment of networks capable of supporting IP communication. The availability of such networks creates significant technical opportunities for real time distribution of multimedia content. There are technical challenges however, as the networks available to a mobile device can have significantly differing performance characteristics in terms of signal propagation and bandwidth. In such an environment where the availability and capability of network changes dramatically in a short period of time it is logically to employ a seamless network migration strategy which can select the most appropriate network type at a given point in time. Seamless handover between heterogeneous networks can be achieved by using multi-homing technologies [1, 2, 3]. Currently, two multi-homing transport protocols have been proposed. They are Stream Control Transmission Protocol (SCTP) [4] and Datagram Congestion Control Protocol (DCCP) [5]. Especially with the SCTP, an extension named mobile SCTP (mSCTP), which facilitates mobility has been drafted in [2]. DCCP is an unreliable transport protocol with congestion

G. Pavlou, T. Ahmed, and T. Dagiuklas (Eds.): MMNS 2008, LNCS 5274, pp. 64–76, 2008.
© IFIP International Federation for Information Processing 2008

control, SCTP is a reliable transport layer protocol and employs a similar congestion control mechanism to TCP, Multi-homing feature of SCTP provides a basis for mobility support since it allows a mobile node (MN) to add a new IP address, while holding the old IP address already assigned to itself. On top of SCTP multi-homing feature, mSCTP utilizes ADDIP and DELETEIP functions which enables dynamically adding and deleting an IP addresses to and from the list of association end points in the middle of association [1].

The Partial-Reliable SCTP (PR-SCTP) [6] 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. The authors of [7] investigated MPEG-4 video transmission by partial reliable transmission in SCTP in GPRS network and studied QoS interactions between SCTP, RTP and application. This paper investigates the capability of partial-reliability scheme defined by PR-SCTP to support the real time distribution of multimedia content.

2 Handover and Retransmission Algorithms in SCTP

SCTP is designed to tolerate network failure and therefore provides a mechanism to detect path failure. The path failure detection time is determined by SCTP parameters *Path.Max.Retrans* (*PMR*) and *Retransmission Timeout* (*RTO*). For an idle destination address, the sender periodically sends a heartbeat chunk to that address to detect if it is reachable and update the path *Round Trip Time* (*RTT*). The heartbeat chunk is sent per path *RTO* plus SCTP parameter *HB.interval* with jittering of +/- 50% of the path *RTO*. The default value of *HB.interval* is 30s. *RTO* is calculated from *RTT* which is measured from non-retransmitted data chunks or heartbeat chunks. For a path with data transmission, it can be determined if it is reachable by detecting data chunks and their SACKs. When the acknowledgement for a data chunk or for a heartbeat chunk is not received within a *RTO*, the path *RTO* is doubled and the error counter of that path is incremented. For a data chunk timeout, the sender retransmits data chunks through the timeout retransmission algorithm. For a heartbeat chunk timeout, the sender sends a new heartbeat chunk immediately. When the path error counter exceeds *PMR*, the destination address is marked as inactive and the sender sends a new heartbeat chunk immediately to probe the destination address. After this, the sender will continuously send heartbeat chunks per *RTO* to the address but the error counter will not be incremented. When an acknowledgement for an outstanding data chunk or a heartbeat chunk sent to the destination address is received, the path error counter is cleared and the path is marked as active. If the primary path is marked as inactive, the sender will select an alternate path to transmit data. When the primary path becomes active, the sender will switch back to the primary path to transmit data.

The SCTP congestion algorithms [4] are inherited from SACK TCP [8], which include slow start, congestion avoidance and fast retransmit. In [9], the authors present a detailed comparison between the congestion algorithms of SCTP and TCP. SCTP

defines two retransmission algorithms: fast retransmission and timeout retransmission. SCTP sends all lost data chunks first before sending new data chunks to ensure that multiple paths are not used in parallel.

When an SCTP sender discovers a data chunk is lost on the primary path through the fast retransmission algorithm, it will enter fast recovery phase. The sender will adjust Congestion Window (*cwnd*) and Slow Start Threshold (*ssthresh*) of the primary path and then fast retransmit the data chunk immediately via a selected path, no matter what the current congestion window size of that path is. If multiple data chunk losses are detected simultaneously, the sender will only send one packet via the fast retransmission algorithm. The rest of the lost data chunks will be retransmitted when the path *cwnd* allows. After all the lost chunks have been retransmitted, the sender will send new data chunks on the primary path if the primary path *cwnd* allows. As long as the congestion window is not full and the receiver window size (*rwnd*) maintained in the sender is not zero, the sender can continuously send new data.

According to path selection strategies during retransmissions, three retransmission policies have been proposed and investigated in [10]. They are:

AllRtxAlt. All Retransmissions to an Alternate Path;
AllRtxSame. All Retransmissions to the Same Path;
FrSameRtoAlt. Fast Retransmissions to the Same Path, Timeout Retransmissions to an Alternate Path.

The authors of [10] evaluated these three retransmission policies with different extensions and the default SCTP parameters in various lossy environments. The results show that *FrSameRtoAlt* with the Multiple Fast Retransmission algorithm and the Timestamp or the Heartbeat after *RTO* extension performs best amongst the three policies and their respective extensions. *AllRtxAlt* performs worst because of the stale *RTO* problem [10].

In [11], the authors studied the performance of different *PMR* settings with *FrSameRtoAlt* and the Multiple Fast Retransmission extension [10]. The results show that *PMR*=0 can achieve best throughput in various path failure or non-failure situations. The authors of [12] investigate SCTP's throughput performance in different path scenarios and proposed a change to the protocol's heartbeat mechanism to improve the performance. The effect of path delay on SCTP performance was studied in [13]. [14] indicates that retransmission of all data on the same path with the path failure detection threshold set to one or zero gives the most stable performance in all path configurations. However, all of above researches focus on the performance of "FTP over SCTP".

This paper investigates real-time multimedia transmission performance of SCTP's retransmission policies with different *PMR* values, path bandwidths, delays and loss rates in various symmetric and asymmetric path conditions. In the simulations, 3G and GPRS link parameters are used as the references for network configurations. Although SCTP is designed to provide transport services over connectionless packet networks, not merely over IP, this paper assumes the underlying network is IP. In this way, the simulation scenarios are IP over 3G and GPRS. For symmetric path conditions, this paper focuses on the situations where a computing node has two 3G connections. For asymmetric path conditions, the tests cover different path delay,

bandwidth configurations and the situations where a computing node has a hybrid 3G or GPRS connection. Uniform loss is used to simulate network congestion.

The rest of this paper is structured as follows. In Section 3, we propose the solution for evaluating quality of MPEG-4 video transmission over SCTP. Section 4 compares different retransmission policies with various *PMR* settings in symmetric and asymmetric path conditions, and then gives the analysis of the results. The conclusions are presented in Section 5.

3 Evalvid-SCTP

With the increasing deployment of real-time applications over 3G and GPRS networks, end-to-end delay and packet loss are vital QoS requirements for these applications. In order to investigate the behavior and quality of such applications under heavy network load, it is therefore necessary to create genuine traffic patterns, both at network/transport layer and application. To setup true multimedia test networks is expensive and offer little flexibility. Instead network simulations using tools like the NS-2 [15] enable building customized effective networks at a low cost at their testing in details.

In recent years, many papers have studied multimedia delivery through simulations. In [16], MPEG-4 trace files are used to calibrate a Transform Expand Sample (TES) mathematical model, and rate adaptation is incorporated by adjusting the frame size output by a scalar (from rate-distortion curve). The simulation model however has no on-line rate controller, and since the traffic is synthetic, perceived quality cannot be investigated. H.263 video trace files are used in [17], and the sending rate is controlled by DCCP TCP-like. In [18] models are derived for pre-recorded media streaming over TFRC and compared to simulations. The models focus on the impact of the TFRC rate changes to the probability of rebuffering events, i.e. events where the receive buffer is emptied. Authors of [19] proposed a tool-set called Evalvid-RA to support rate adaptive MPEG-4 VBR video simulation. Evalvid tools-set [20] 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 [21], perceived quality and objective measure like Peak Signal-to-Noise Ratio (PSNR) calculation can be obtained after network simulation.

The quality of a video transmission depends on the impression a human observer receives of the delivered video. The subjective video quality test results are expressed by means of e.g. the mean opinion score (MOS) as defined by the ITU. The MOS is a scale from 5 (excellent) to 1 (bad). In contrast, objective video quality metrics are calculated by computers. Basically, these can be divided into pixel-based metrics, like SNR or PSNR, and psycho-visual metrics.

Based on related work, we designed Evalvid-SCTP, a solution for MPEG-4 video investigation over SCTP in NS-2, it is based on modifications to Evalvid and Delaware University's SCTP module [22] for NS-2 [15], and then Evalvid is integrated into SCTP.

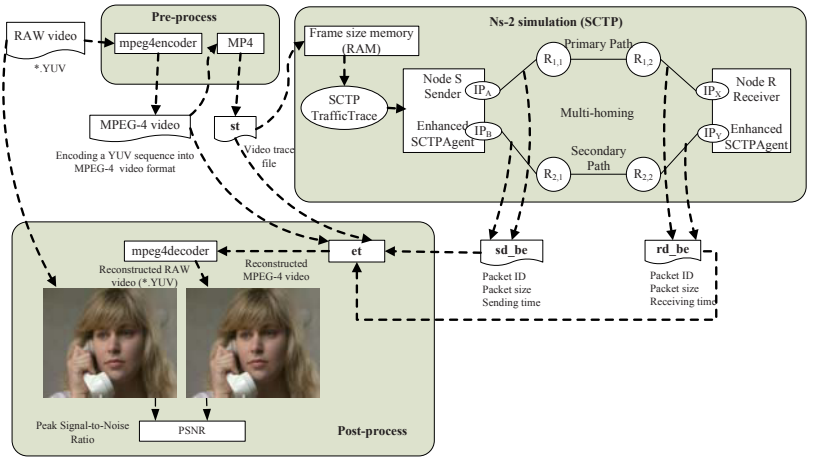


Fig. 1. The pre-process, NS-2 simulation and post-process of the Evalvid-SCTP framework

Table 1. The Evalvid-SCTP Tools Overview: Pre-process, NS-2 simulation, Post-process

Tool	Purpose
mpeg4encoder.exe	Pre-process: Encode video file into MPEG-4 video
mp4.exe	Pre-process: Create frame size trace file of encoded file from previous step
SCTPTrafficeTrace	Ns-2 simulation: A extended Agent which send data to the enhanced SctpAgent following the trace file from previous step
Enhanced SctpAgent	Ns-2 simulation: Modified sctp.h and sctp.cc in where sender trace file is written, including sending time, packet type, id, size, and receiver trace file is written, including receiving time and packet type, id, size.
et.exe	Post-process: From Evalvid to reconstruct video after transmission.
fixyuv.exe	Post-process: Inserts missing frames due to drop or late arrival so that sent and received video has the equal number of frames
psnr.exe	Post-process: calculate the PSNR
mos.exe	Post-process: Map MOS values from PSNR

Evalvid-SCTP enables simulation of multimedia 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, NS-2 records packet throughput at each node including the receiver. Using this information and the original compressed video file, Evalvid-SCTP reconstructs the video as if it were received on a real network. This

reconstruction enables the video to be inspected visually as well as allowing for the calculation of PSNR and Mean Opinion Score (MOS) for the transferred video.

As figure 1 shows, the framework of Evalvid-SCTP has three stages 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-frame, P-frame, B-frame) in the video. In the NS-2 simulation, the Agent SCTPTrafficeTrace sends data to SCTP network following video trace file, the enhanced SCTPAgent records the sender trace file including sending time, packet type, packet id, size. It also records the receiver trace file including receiving time, packet type, packet id and size. In the post-process phase, the video is reconstructed and converted in to raw video (YUV), the video quality evaluation is given by the calculation of PSNR. Table 1 lists some primary tools used by Evalvid-SCTP.

4 Simulation-Based Assessment

4.1 Simulation Setup

The simulations in this section consider different network path conditions. 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 Maximum Transmission Unit (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 1000 packets.

The SCTP parameters are the default ones [4]. The initial slow start threshold is set large enough to ensure that the full primary path bandwidth is used. SCTP is set as PR-SCTP, the default reliability level of PR-SCTP is set to 1. Only one SCTP stream is used and the data is delivered to the upper layer in order. For simulations with an infinite receive buffer, the receiver window (*rwnd*) is set to 100 MB as this size is larger than the data size transmitted by the sender. Network congestion is simulated by varying the path loss rate (2%, 4% and 8%).

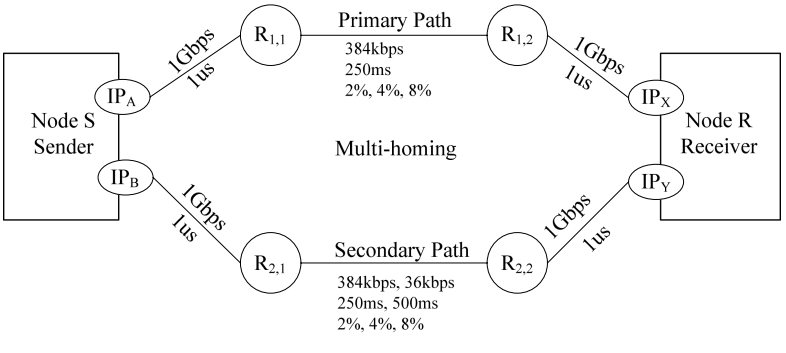


Fig. 2. Simulation network topology

A YUV video sequence is used with a QCIF format (resolution 176x144 pixels) and 2000 frames. After pre-processing in Evalvid-SCTP, a MPEG-4 video trace file is produced. In NS-2 node S sends data from this video trace file at 30 frames/sec to node R at t=5 sec. 10 random seeds are used for simulation and testing results are calculated by averaging the results of 10 runs.

4.2 Symmetric Path Bandwidth and Path Loss Rate

This section studies the performance of retransmission policies and *PMR* settings in symmetric path conditions. A computing node has two 3G connections and an infinite buffer. The bandwidth of both bottleneck links is set as 384kbps. The delays on the primary and secondary path are 250ms, and the paths loss rates are set to 2%, 4% and 8%. Aggressive (*PMR*=0) and less aggressive (*PMR*=1) failover settings are set respectively. The results for *PMR*=2, 3, 4, 5 are not shown in this paper. One 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. However, the initial *RTO* could be 60s (*RTO.Max*). Therefore, the maximum path failure detection time is 6*60=360s! Another reason is that the data transmission time for *PMR*>0 is similar.

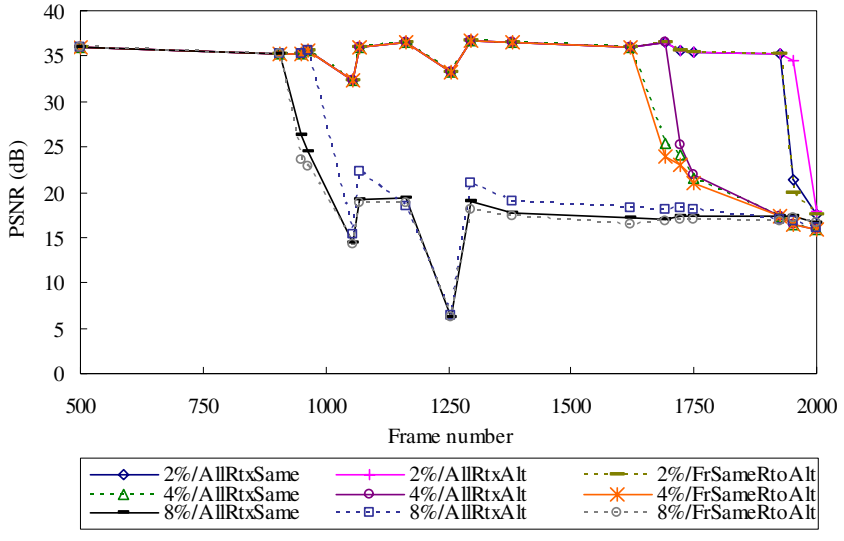
Table 2 and 3 show the comparison results of average PSNR (dB) values and the numbers of different lost frames (I-frame/P-frame/B-frame) after transmission, which

Table 2. Post-processing results (*PMR*=0)

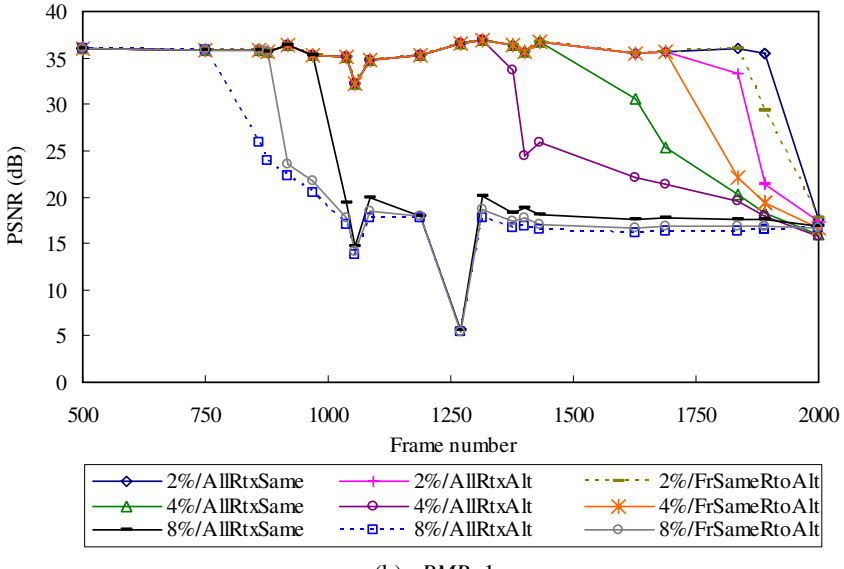
	Path loss rate =2%		**Path loss rate =4%**		**Path loss rate =8%**	
	Average PSNR (dB)	**Frames dropped (I/P/B)**	**Average PSNR (dB)**	**Frames dropped (I/P/B)**	**Average PSNR (dB)**	**Frames dropped (I/P/B)**
AllRtxSame	35.20	8/17/49	33.02	42/85/252	26.60	115/229/686
AllRtxAlt	35.39	6/11/32	33.39	34/69/204	27.17	108/216/647
FrSameRtoAlt	35.20	8/17/49	33.02	42/85/252	25.82	122/243/728

Table 3. Post-processing results (*PMR*=1)

	Path loss rate=2%		**Path loss rate =4%**		**Path loss rate =8%**	
	Average PSNR (dB)	**Frames dropped (I/P/B)**	**Average PSNR (dB)**	**Frames dropped (I/P/B)**	**Average PSNR (dB)**	**Frames dropped (I/P/B)**
AllRtxSame	35.06	11/21/62	33.02	42/85/252	26.60	118/235/705
AllRtxAlt	34.55	19/37/110	31.08	70/139/416	25.05	128/256/766
FrSameRtoAlt	34.94	13/25/74	33.42	35/69/206	25.59	125/250/749



(a) *PMR*=0



(b) *PMR*=1

Fig. 3. PSNR (dB) for symmetric paths (primary path: 384kbps & 250ms; secondary path: 384kbps & 250ms; paths loss rate change with 2% , 4%, 8%)

employed different retransmission policies: *AllRtxSame*, *AllRtxAlt* and *FrSameRtoAlt* with *PMR* 0 and 1 respectively. Figure 3 shows the corresponding PSNR values frame-by-frame (only the last 1500 frames are shown). As the tables and figure illustrate, with the increasing path loss rate PSNR decreases and the numbers of lost

frames increase in all cases. However, in most cases, setting *PMR*=0 performs better than setting *PMR*=1. Retransmission of all data on an alternate path with *PMR*=0 performs best; however, its performance degrades with *PMR*=1. Retransmission of all data on the same path with the *PMR* set to zero or one performs in a more stable manner than other configurations.

4.3 Asymmetric Path Bandwidth and Path Loss Rate

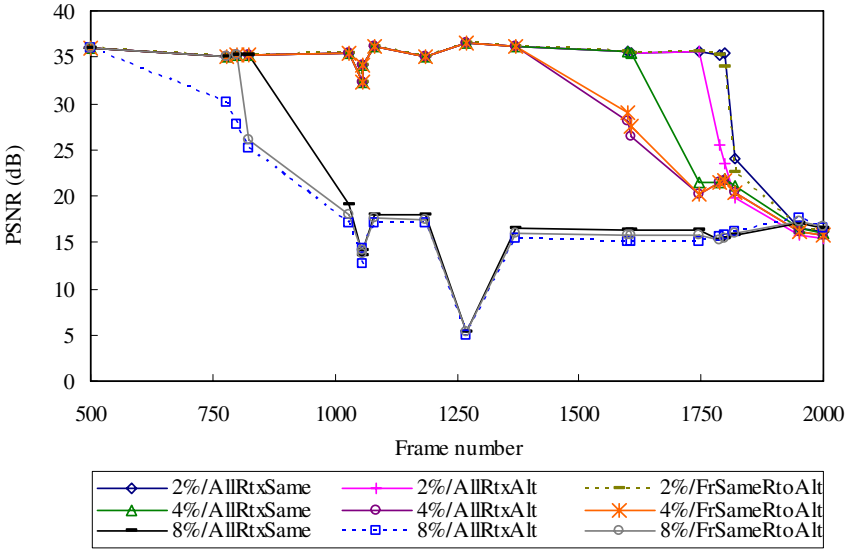
This section studies the performance of retransmission policies and *PMR* settings in asymmetric path conditions. A computing node has a hybrid of 3G or GPRS connections and an infinite buffer. The primary path bandwidth is 384kbps and the secondary path bandwidth is 36kbps. The delay on the primary path is 250ms, the delay of secondary path is 500ms, and the loss rates of both paths are set to 2%, 4% and 8%. Other settings are the same as for previous tests. Table 4 and 5 illustrate the comparison results for average PSNR (dB) values as well as lost frames (I-frame/P-frame/B-frame), for different retransmission policies: *AllRtxSame*, *AllRtxAlt* and *FrSameRtoAlt* with *PMR* 0, 1 respectively. Figure 4 shows the corresponding PSNR values frame-by-frame (only the last 1500 frames are shown). As the table and figure show, with the increasing of path loss rate the PSNR decreases and the number of slipped frames increase in all the cases. The total average video quality degrades compared with symmetric path conditions. In most cases however, *PMR*=1 performs better than *PMR*=0. Retransmission of all data on an alternate path performs worst. Retransmission of all data on the same path with the *PMR* set to zero or one performs in a more stable manner than other configurations.

Table 4. Post-processing results (*PMR*=0)

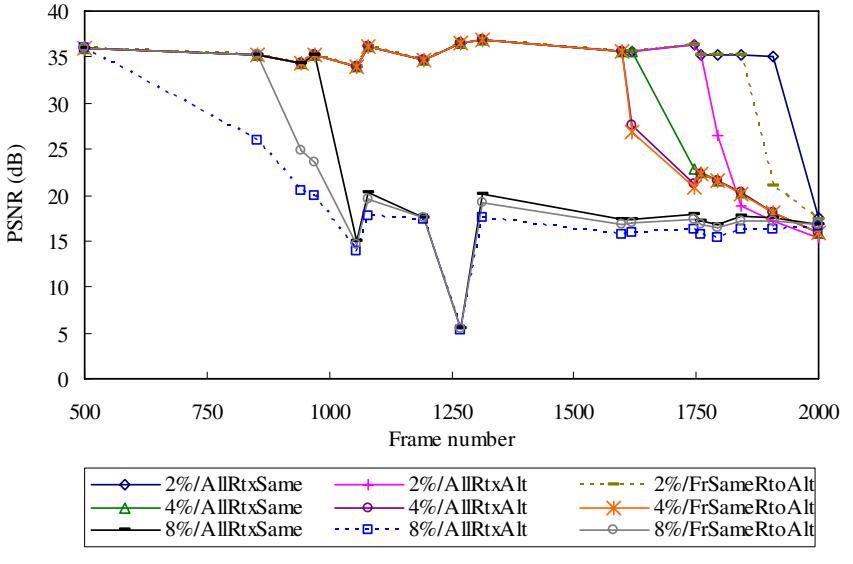
	Path loss rate =2%		**Path loss rate =4%**		**Path loss rate =8%**	
	Average PSNR (dB)	**Frames dropped (I/P/B)**	**Average PSNR (dB)**	**Frames dropped (I/P/B)**	**Average PSNR (dB)**	**Frames dropped (I/P/B)**
AllRtxSame	34.06	23/45/136	32.96	44/87/262	24.72	131/262/785
AllRtxAlt	33.83	25/49/217	32.69	46/91/274	24.66	137/275/822
FrSameRtoAlt	34.06	23/45/136	32.69	46/91/274	24.72	133/266/796

Table 5. Post-processing results (*PMR*=1)

	Path loss rate=2%		**Path loss rate =4%**		**Path loss rate =8%**	
	Average PSNR (dB)	**Frames dropped (I/P/B)**	**Average PSNR (dB)**	**Frames dropped (I/P/B)**	**Average PSNR (dB)**	**Frames dropped (I/P/B)**
AllRtxSame	35.06	11/21/62	33.02	42/85/252	26.60	115/229/686
AllRtxAlt	33.83	25/49/148	32.96	44/87/262	24.90	129/257/770
FrSameRtoAlt	34.58	18/35/105	32.88	44/89/264	26.08	119/239/714



(a) *PMR*=0



(b) *PMR*=1

Fig. 4. PSNR (dB) for asymmetric paths (primary path: 384kbps & 250ms; secondary path: 36kbps & 500ms; paths loss rate change with 2%, 4%, 8%)

4.4 Analysis of the Results

The test results illustrate that in most cases, aggressive failover setting (*PMR*=0) performs better than less aggressive failover setting (*PMR*=1) regardless of the path loss rates in symmetric path conditions. As we know, the underlying advantage of aggressive failover is that handover occurs with less time blocked during failure detection. With *PMR*=0, a single timeout migrate new data transmission to the alternate path quickly while the primary destination is probed with heartbeats. Though aggressive failover setting could increases the possibility of spurious failover where a small number of lost packets is interpreted to mean that the destination address is no longer reachable and sender mistakenly concludes a failure has occurred, however the alternate path has the same good path conditions with the primary path, which avoids negative impact by unnecessary failovers. So this scenario of symmetric paths with *PMR*=0 is actually a concurrent multi-path transmission, then it achieves better performance.

However, the investigation revealed that when a bandwidth asymmetry did exist, setting *PMR*=0 was not good advice. Less aggressive failover setting (*PMR*=1) generally outperforms aggressive failover setting (*PMR*=0) in asymmetric path conditions. As the secondary path conditions are worse than primary path with less bandwidth and larger delay. In this scenario, as it is discussed in [12], there is a substantial advantage to sticking with the higher speed primary path, despite the fact that it is not functioning, and waiting for it to be restored, rather than switching over to the lower speed alternate path. The reason for this is that when SCTP stays with the primary path, it will more quickly discover when the path is again functional (by retransmitting user data using exponential back-off) than if it fails over to the alternate and relies upon the slower heartbeat (HB) mechanism to probe for the primary's recovery. So with *PMR*=1, the worse secondary path is seldom used, which achieves better performance than that of *PMR* setting to 0.

The results show that all retransmissions to an alternate path with *PMR*=0 performs best in symmetric path conditions and degrades seriously in asymmetric path conditions. For *AllRtxAlt*, the lost data will be retransmitted on the secondary path, even for *PMR*=0. Therefore, the performance will be degraded when the secondary path conditions are significantly worse than the primary path conditions. For *PMR*>0, *AllRtxAlt* performs worst because the stale *RTO* problem as indicated in [10] and can be explained as follows. A retransmission timeout on the alternate path will double the *RTO*, whereas a successful retransmission will not refresh the *RTO* which can only be updated by the heartbeat chunks. Consequently, the *RTO* on the alternate path is usually a large value which causes the data loss detection time to become very long and degrades the performance. However, SCTP can avoid the stale *RTO* problem with *PMR*=0. Every time a packet is lost, the destination address is marked as inactive. The sender will transmit a heartbeat chunk to the inactive destination address immediately, which can get a new measurement for the path *RTT* and *RTO*. The HeartbeatAfterRTO extension proposed in [10] can be achieved automatically through *PMR*=0. *AllRtxAlt* retransmits all lost data on an alternate path. In the fast retransmit phase, the lost data are retransmitted immediately irrespective of the current path *cwnd*. This is actually a concurrent multipath transmission. Therefore, *AllRtxAlt* with *PMR*=0 in symmetric path conditions gives the best performance.

5 Conclusions

This paper proposed a solution called Evalvid-SCTP to analyse the performance of real-time multimedia transmission over multi-homing transport protocols utilizing network failure tolerant mechanisms. In particular, we focus on the extension of SCTP, Partial Reliable Stream Control Transmission Protocol (PR-SCTP). The three PR-SCTP retransmission policies: *AllRtxSame*, *AllRtxAlt* and *FrSameRtoAlt* were evaluated using a number of path failure detection threshold values in various symmetric and asymmetric path conditions through SCTP simulations. Uniform loss is used to simulate network congestion. The results indicate that an aggressive failover setting (*PMR*=0) performs better in symmetric path conditions, however a less aggressive failover strategy (*PMR*=1) performs better in asymmetric path conditions. Retransmission of all video on an alternate path with *PMR*=0 performs best in symmetric path conditions but with *PMR*=1 performs worst among all retransmission policies and *PMR* settings. Retransmission of all video on the same path with the path failure detection threshold set to zero or one is recommended since it gives the most stable performance in all path situations.

References

1. Stewart, R., Xie, Q., Tuexen, M., et al.: Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration. IETF draft, draft-ietft-tsvwg-addip-sctp-20.txt, draft-ietf-tsvwg-addip-sctp-15.txt (2007)
2. Riegel, M., Tuexen, M.: Mobile SCTP. IETF Draft, draft-riegel-tuexen-mobile-sctp-05.txt (2005)
3. Dutta, A., Das, S., Famolari, D., et al.: Seamless Handoff across Heterogeneous Networks - An 802.21 Centric Approach. In: Proc. IEEE WPMC, Aalborg Denmark (2005)
4. IETF RFC 2960: Stream Control Transmission Protocol (2000)
5. IETF RFC 4340: Datagram Congestion Control Protocol (DCCP) (2006)
6. Stewart, R., Ramalho, M., Xie, Q., et al.: Stream Control Transmission Protocol (SCTP) Partial Reliability Extension. IETF RFC 3758 (2004)
7. Huang, H., Ou, J., Zhang, D.: Efficient Multimedia Transmission in Mobile Network by using PR-SCTP. In: Proc. of Communications and Computer Networks, CCN, Marina del Rey, USA (October 2005)
8. IETF RFC2018: TCP Selective Acknowledgement Options (1996)
9. Fu, S., Atiquzzaman, M.: SCTP: State of the art in Research, Products, and Technical Challenges. IEEE Communications Magazine 42(4), 64–76 (2004)
10. Caro, A., Amer, P., Stewart, R.: Retransmission Policies for Multihomed Transport Protocols. Computer Communications 29(10), 1798–1810 (2006)
11. Caro, A., Amer, P., Stewart, R.: Rethinking End-to-End Failover with Transport Layer Multihoming. Annals of Telecommunications 61(1-2), 92–114 (2006)
12. Grace, K.H., Pecelli, D., Amelia, J.D.: Improving Multi-homed SCTP Mobile Communication Performance. Technical Papers, The MITRE Corporation (2006)
13. Qiao, Y., Fallon, E., Murphy, L., et al.: SCTP Performance Issue on Path Delay Differential. In: Boavida, F., Monteiro, E., Mascolo, S., Koucheryavy, Y. (eds.) WWIC 2007. LNCS, vol. 4517, pp. 43–54. Springer, Heidelberg (2007)

14. Qiao, Y., Fallon, E., Murphy, L., et al.: Performance Analysis of Multi-homed Transport Protocols with Network Failure Tolerance. IET Communications 5(2), 336–345 (2007)
15. UC Berkeley, LBL, USC/ISI, and Xerox Parc ns-2 documentation and software, Version 2.29 (2005), http://www.isi.edu/nsnam/ns
16. Liew, C.H., Kodikara, C., Kondoz, M.A.: Modelling of MPEG-4 Encoded VBR Video Traffic. IEE Electronic Letters 40(5) (2004)
17. Xu, C., Liu, J., Zhao, C.: Performance analysis of transmitting H.263 over DCCP. In: Proc. of IEEE Int. Workshop VLSI Design and Video Technology (2005)
18. Xu, L., Helzer, J.: Media Streaming via TFRC: An Analytical Study of the Impact of TFRC on User-Perceived Media Quality. In: Proc. of IEEE Infocom (2006)
19. Lie, A., Klaue, J.: Evalvid-RA: Trace Driven Simulation of Rate Adaptive MPEG-4 VBR Video. ACM/Springer Multimedia Systems Journal (2007)
20. Klaue, J., Rathke, B., Wolisz, A.: Evalvid - A Framework for Video Transmission and Quality Evaluation. In: Proc. of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, Urbana, Illinois, USA (2003)
21. Ke, C.H.: How to evaluate MPEG video transmission using the NS2 simulator. [Online], http://hpds.ee.ncku.edu.tw/smallko/ns2/Evalvid in NS2.htm
22. Caro, A., Iyengar, J.: ns-2 SCTP module, Version 3.5, http://www.armandocaro.net/software/ns2sctp/