On the Stability of Delay-Centric Multihomed Transmission

Pedro Mantovani Antunes, Guilherme de Souza Miguel,
Carlos Marcelo Pedroso, Eduardo Parente Ribeiro
Federal University of Parana - UFPR
P.O. Box 19011 – 81531-980
Curitiba – PR – Brazil
edu@ufpr.br

***Abstract*— The use of multiple network interfaces for multimedia transmission can reduce packet delay by choosing the less congested end-to-end path for transmission. Delay-centric is a simple mechanism that selects the path with current lowest delay. When multiple users employ this mechanism there is a possibility of instabilities due to excessive path changes that increases overall packet delay. In this paper we investigate some modifications on the delay-centric algorithm to reduce overall transmission latency in a scenario with multiple independent transmissions.**

***Keywords*— multihome, end-to-end multipath, delay-centric, SCTP.**

I. INTRODUCTION

Multihoming has been gaining increasing popularity in the recent years. Users can be connected to the Internet by different network access technologies such as ADSL (Asymmetric Digital Subscriber Line), Wi-Fi, WiMaX (Worldwide Interoperability for Microwave Access), 3G and LTE (Long Term Evolution). Multimedia communication can benefit from multiple end-to-end paths by selecting the best available path for transmission at a particular time, thus avoiding congestion and reducing packet loss.

A framework for multihomed communication was established by the Stream Control Transmission Protocol (SCTP) [1], but other approaches are possible, such as Multipath Transmission Control Protocol (MPTCP) [2][3] and application layer handover [4]. The standard SCTP protocol monitors the availability of each end-to-end path and automatically switches transmission to a secondary path in case of failure in the primary path to increase resilience. Other utilizations of multihoming already proposed in the literature include concurrent multipath transfer (CMT) [5][6][7], seamless handoff for mobile users [8][9] and delay-centric transmission for low-delay communication [10][11]. The latter is suitable for real-time multimedia transmissions. The benefits of the delay-centric algorithm for a single multimedia transmission have already been shown by several authors [12][13][14][15].

One topic that has not been fully investigated is the stability issue that arises when multiple users employ the delay-centric mechanism. An initial investigation has shown that under high link utilization, overall packet delay can increase due to excess of path changes of the transmitting sources [16]. In this paper, we investigate some preventing measures that can be taken to mitigate these instabilities and to allow lower end-to-end delay for all users.

The remainder of this paper is divided into four sections. Section II describes some fundamentals of delay-centric SCTP transmission. Section III explains the implemented methods. Results are shown in Section IV e conclusion is presented in Section V.

II. MULTIHOMED LOW-DELAY COMMUNICATION

A. Latency Estimation

Latency estimation of the end-to-end path is performed in both SCTP and TCP to calculate the retransmission timeout (RTO) of each packet [1][17]. To achieve this estimation, these protocols measure the round trip time (RTT) of a packet every time an acknowledge is received. As this value is highly variable, the protocol uses the smoothed round trip time (SRTT), defined as

$$SRTT_i = (1-\alpha)SRTT_{i-1} + \alpha RTT \tag{1}$$

where α value is 0.125, as recommended by RFC 4960 [1] and RFC 6298 [18].

B. Delay-centric

Delay-centric method compares the SRTT of all paths to decide where to transmit the packets [10]. The SRTT of the primary path is updated frequently, because it is changed every time a packet is transmitted and its ACK is received. In the secondary paths, the SRTT is updated only after a heartbeat (HB) is sent and a heartbeat-ack (HB-ACK) is received. This does not give these paths a proper estimation of the SRTT, as the standard interval value between HBs is 30 s. To improve the algorithm's responsiveness, most authors employed the value of 1 s for the HB interval [12][16][19].

The path handover in delay-centric can occur when the difference of the SRTT of the current path and the SRTT of an alternate path becomes greater than zero, or greater than a threshold. This threshold in milliseconds is simply called hysteresis.

A variation of the pure delay-centric method is the setting of a guard-time [20]. This is the period, in milliseconds, to wait after the SRTT comparison indicated a path handover. At the end of this period, the SRTTs are compared again, to check if the current SRTT is still greater than the SRTT of the alternate path, to confirm the handover or to cancel it, and keep the transmission in the same path. We propose the use of a random guard-time instead of a fixed guard-time. The idea is avoid synchronous operation of the sources and to promote better distribution of the handover decision over the time. We also propose the reduction in the heartbeat interval to 20 ms during guard-time, in order to improve the estimative of the SRTT of the alternate path.

C. Predictive delay-centric

A recent improvement in the delay-centric method also considers the SRTT trend [19]. The moving average convergence divergence (MACD) method is used. For each path, two SRTTs are calculated. A short time version, with $\alpha = 0.667$, named $SRTT_S$, and a long time version with $\alpha = 0.154$, named $SRTT_L$. When the $SRTT_S$ becomes greater than $SRTT_L$, there is a trend to increase latency. The proposed handover algorithm takes this trend into account and performs the handover only when these three conditions hold:

i) Current path SRTT is increasing ($SRTT_S > SRTT_L$).
ii) $SRTT_S$ of current path is greater than a threshold, set at 150 ms.
iii) $SRTT_S$ of current path greater than $SRTT_S$ of alternate path.

Simulations with typical delays gathered from Wi-Fi, 3G and Ethernet, have shown that this strategy, called predictive delay-centric (PDC), improves the quality of video transmission and reduces the number of handovers compared to pure delay-centric, which is a reactive method (RDC).

In this paper, we propose a modified version of predictive delay-centric for better performance. We replace condition (i) by: $(SRTT_S - SRTT_L) > (srtt_S - srtt_L)$, where upper case SRTT is the SRTT of the primary path, while the lower case is the SRTT of the alternate path. This condition is a strong indicator that the trend of increasing latency is greater in the current path than it is in the alternate path. This change requires the calculation of SRTT trend on both primary and secondary paths. Switchover only occurs if the alternate path has a better tendency to reduce the latency than the primary path, and if conditions (ii) and (iii) are also true. This modified method was used all along this paper, however both methods were tested and presented at the end of Results section, in order to provide proper comparison.

III. Methods

The test bed consisted of two computers connected to each other by two Ethernet links, as illustrated in Figure 1. In our last simulation, the same methodology was applied to check the algorithms' behavior in the presence of a third path.

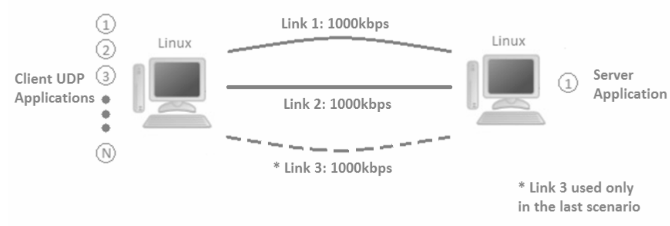

Fig. 1. Topology used in the experiments.

Both machines operate with Linux operating system (distribution Debian 7.1, Linux kernel 3.2.63-2). Test programs were written in C and compiled with gcc version 4.7.2-5. The multimedia transmission data consisted of 6 sources that independently transmitted UDP packets with 250 bytes of size, including headers. Each source also transmits heartbeat packets of 55 bytes in the alternate path. All ACK packets have 52 bytes. Each multimedia source starts its transmission randomly during the first second of the experiment. In order to work with lower bit rates, the bandwidth of the transmission was limited with the kernel traffic control (tc).

A total of 3000 packets were sent on each experiment, which was repeated 200 times to estimate the mean packet delay, its standard deviation and confidence intervals. In each experiment, the mean packet delay was measured in function of the link utilization ρ. Its definition was extended to consider multipath scenario, by considering the aggregated utilization according to

$$\rho = \frac{\sum_{i=1}^{t} B_i}{\sum_{i=1}^{n} C_i} \tag{2}$$

where 't' is the number of active transmissions and 'B' is the transmission traffic bit rate. 'N' is the number of paths and 'C' is the individual path capacity. In the experiments, 'i' was 6 and 'C' capacities were kept at 1000 kbps. The value of 'n' was 2, except in the last experiment, that had a third path. In the simulations, ρ was varied from 0.72 to 0.99 altering the values of 'B'.

The following mechanisms were tested:

i) Pure delay-centric, hysteresis = 10 ms.
ii) Delay-centric, hysteresis = 10 ms, guard-time = 1 – 4 s (heartbeat interval = 20 ms).
iii) Predictive delay-centric.
iv) Predictive delay-centric, guard-time = 1 – 4 s (heartbeat interval = 20 ms).

Heartbeat interval on all mechanisms were set to a random value between 0.5 s and 1.5 s sorted after each heartbeat, except when in guard-time period, when it is changed to 20 ms. Because the bandwidth was limited only in the sender machine, every queue delay was originated in the sender side as well. This doesn't reflect the actual Internet scenario, in which queue delays are originated in both ways of the links. To compensate this problem caused by our topology, we changed the handover threshold in the predictive delay-centric experiments to 70 ms.

The mechanisms were also simulated with the presence of background traffic. The background traffic consisted of fixed size UDP packets (250 bytes) with random, exponentially distributed inter-arrival time. Different proportions of background and foreground traffic were analyzed, but in order to simplify the results, we decided to present only the proportion of 40% background traffic against 60% of foreground traffic, which gives a representative comparison between both mechanisms in a different scenario.

IV. Results

A. Algorithms' Behavior

A first round of experiments was executed without background traffic. For low utilization ($\rho < 0.75$) the distributed delay-centric mechanism was able to promote an even distribution of the transmissions over the two paths. For higher utilization ($\rho > 0.75$), instabilities occur and the excessive path changes increase the overall mean delay.

For a particular high utilization, after 200 transmissions, the mean delay for each transmission was recorded. The obtained histogram of empirical end-to-end delay did not exhibit a single mode, but two modes. One mode is close to zero (minimum delay) and the other mode is between zero and the maximum delay. The interpretation of this result is that with some initial conditions, delay-centric mechanism was able to distribute all flows on the available paths and achieve low mean delay for all flows. But other initial conditions led to non-stable behavior and delay-centric method kept changing paths. This resulted in large mean delays for all flows. Figure 2 displays the mean delay as a function of the utilization.

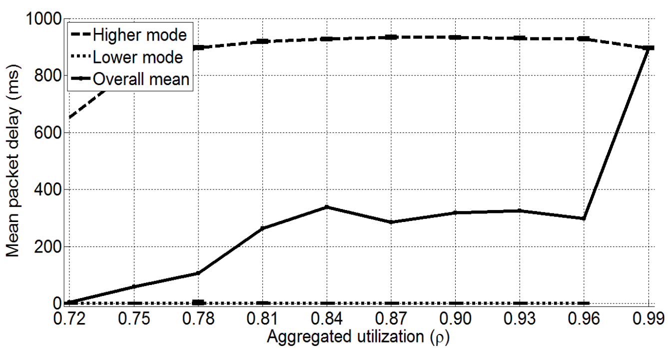

Fig. 2. Pure delay-centric (RDC).

In Figure 2, the dashed curve on top is the higher mode, which represents cases of instabilities. The dotted curve in the bottom is the lower mode, which clearly stands for the cases of good stability, as the delay goes to almost 0 ms. For each curve, confidence intervals (95%) are plotted.

The central curve is the overall mean packet delay. This is the most important information of the Figure, though the modes are also interesting information to check how the cases of stability and instability are distributed along the axes.

The next experiment, represented in Figure 3, was executed with the guard-time allied to the delay-centric algorithm, with the reduction of HB interval during guard-time, as explained in Section III. The two modes were still present in the experiment, which indicates instability in some cases.

As can be noted in Figure 3, this latter algorithm, compared to the pure delay-centric, had a great reduction of overall delay

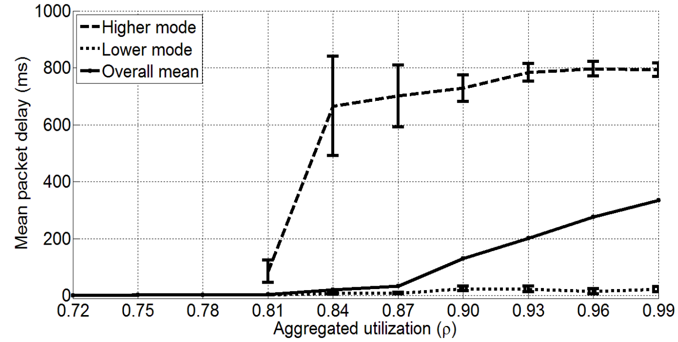

Fig. 3. RDC with Guard-time and HB interval reduction.

of the packets, which may be due to the decrease of HB interval. This may have given the sources a better estimative of the SRTTs, leading to a better path decision.

The predictive delay-centric was evaluated next. The MACD algorithm was tested twice: a simple predictive delay-centric, and a predictive delay-centric with guard-time and HB interval reduction. These experiments are represented respectively in Figure 4, and Figure 5.

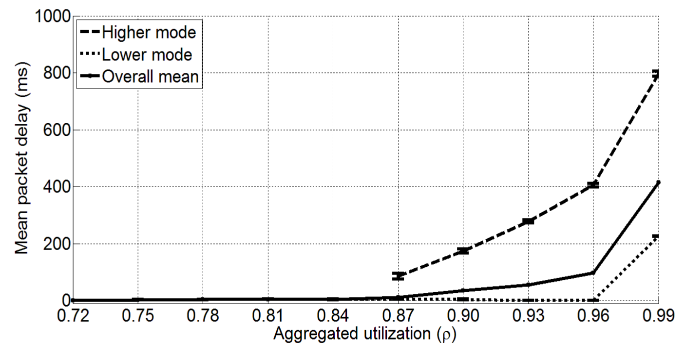

Fig. 4. Simple predictive delay-centric (PDC).

The pure MACD algorithm resulted in lower mean packet delay than the pure delay-centric, which indicates its natural trend to avoid instabilities. Even in cases of instability, it is noticed that in the predictive delay-centric the higher mode delay is lower than in the reactive delay-centric.

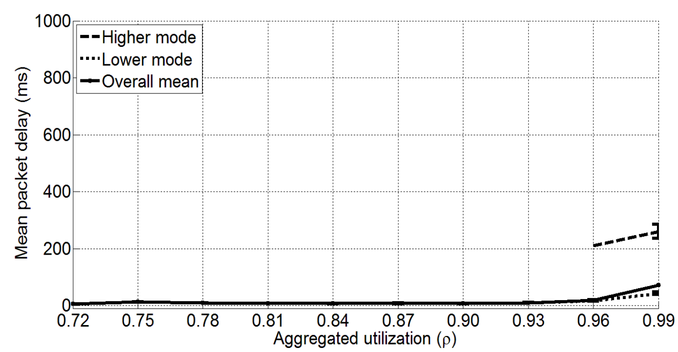

Fig. 5. PDC with guard-time and HB interval reduction.

Following the tendency from the delay-centric, the guard-time implemented in predictive delay-centric helped even more to lower the overall mean packet delay, eliminating almost every case of instability. With this algorithm, even in the highest utilizations, the overall delay is not high enough to greatly affect the QoS of a VoIP call, for example.

B. Background Traffic Scenario

Afterwards, experiments were made with background traffic to investigate the superiority of the last mechanism (Figure 5) in a different and more realistic scenario. In this situation, random delays were caused in the sending queue by the exponentially distributed interval between packet sends. We noted that the addition of background traffic improved the stability of the algorithms. Different and random transient delays may have helped unbiased path choice.

Figure 6 presents the comparative between all the previous tested methods. In this simulation, all the sources represent 60% of the total traffic and the exponential background traffic represents 40% of total traffic. Mean packet delay of several repetition for each method resulted in a unimodal distribution. The overall mean delay and confidence intervals for each method were plotted.

Also in this more realistic scenario, with background traffic, the predictive delay-centric with guard-time had better stability than all the other algorithms. It was able to keep low delay for all sources of transmission, even for utilization as high as 0.99.

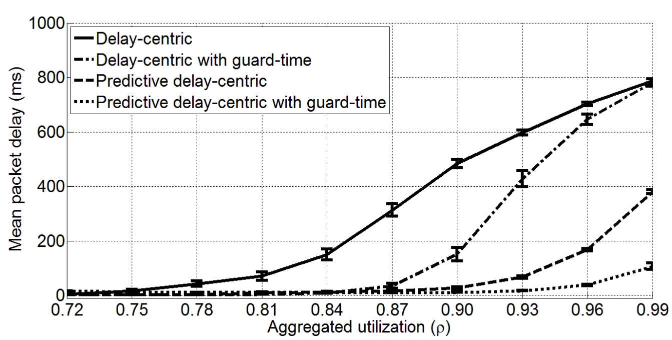

Fig. 6. Scenario with random background traffic.

C. Comparison of Predictive delay-centric methods

As explained in Section II, the predictive delay-centric method used so far was the algorithm that analyzes the latency trend of all paths. In the original method only the trend of the primary path is analyzed. In Figure 7, a comparison is made between the original method and the modified method in the scenario with background traffic. Both algorithms were tested with and without the presence of the guard-time. Without the guard-time, the algorithms acted almost in the same way, and the mean packet delay is almost equal. This is due to the low proportion of packets in the secondary path comparing to the primary. With the guard-time implemented, the modified method exhibited lower delays in the highest utilization levels. The trend of secondary paths could now be calculated more precisely because of the decrease in the HB interval during guard-time, giving the modified algorithm a slight advantage. This behavior assisted this latter algorithm to distribute the transmissions evenly among the paths.

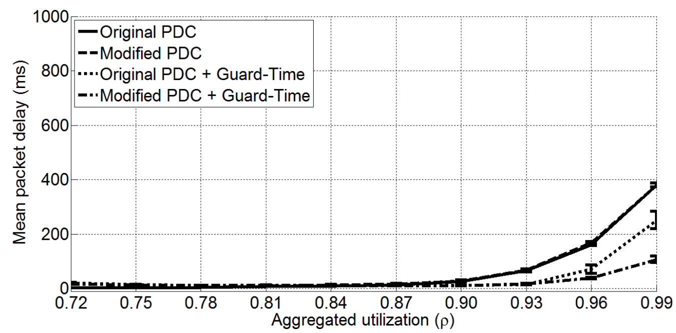

Fig. 7. Comparison between predictive delay-centric methods.

D. Algorithms' Behavior with a third path

The results of the tests with two paths may apply to most of the scenarios, since few end systems have access to more than two simultaneous links to the Internet. However, the analysis of the algorithms in the presence of an additional path is relevant. All methods presented in this paper were tested with 3 paths and background traffic. The proportion of this background traffic was 40% of total traffic, the same as in previous tests. Figure 8 stands for the simulation of the RDC and RDC with Guard-Time, while Figure 9 represents the PDC and PDC with Guard-Time. In both figures, the results are presented with their equivalent result in the 2 paths simulation.

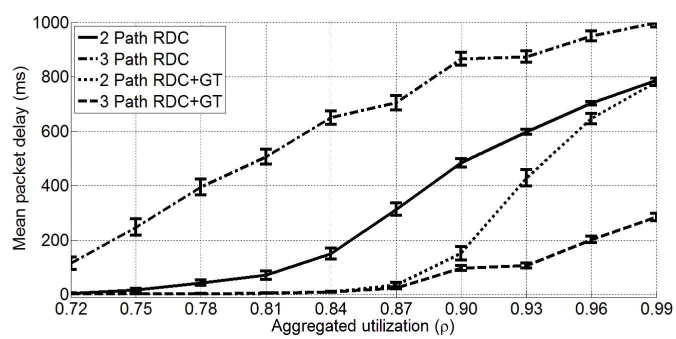

Fig. 8. RDC and RDC+Guard-Time with 3 paths.

The pure delay-centric (RDC) proved once more to be unable to keep low delay for the transmissions. Moreover, the instability of this method increased even more with the additional path, showing its incapacity in dynamic conditions. Unlike the pure RDC, the RDC with guard-time had a better response with the third path, denoting the high efficiency of the guard-time in aiding the algorithm to distribute the transmission along the paths.

In Figure 9, it is possible to see that the differences in the pure PDC method were more discrete, but the addition of the third path raised the overall delay too, as it did to the pure

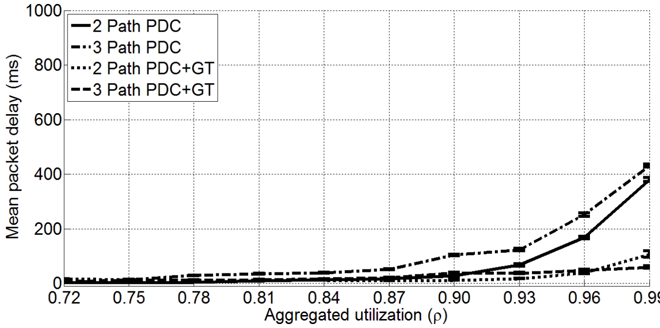

Fig. 9. PDC and PDC+Guard-Time with 3 paths.

RDC. Allied to the guard-time algorithm, the PDC had almost the same results, perhaps because of the already very low delay. The only notable difference is when $\rho = 0.99$. In that case, the third path helped lower delay for all transmissions.

V. Conclusion

Low-delay communication is a desired condition for multimedia transmission, such as VoIP calls and video streaming, especially in real-time scenarios. Although delay-centric mechanism for multihomed communication has been efficient to reduce packet delay by selecting the path with smaller SRTT, some instabilities may occur during high utilization when many sources use the same mechanism. This could be a serious problem if the mechanism becomes standard in any multihomed protocol. We confirmed the existence of such instabilities in transmission between two computers with two and three interface cards.

One cause of instability is the poor sampling of actual smoothed round trip time of the alternate path. Modification of delay-centric with the introduction of a guard-time and frequent updates of the SRTT in the alternate path significantly reduced these instabilities and consequently the overall mean packet delay.

The predictive delay-centric, using trend comparison of current and alternate path also proved to be efficient in reducing the instabilities. The PDC with HB interval reduction during guard-time was able to eliminate almost every instability in the handover mechanism. A comparison between the original PDC [19] and the proposed PDC in this paper was also evaluated. Even though the methods are not very different, the modified PDC reacted better in the presence of the guard-time.

Scenarios with background traffic and the addition of a third path were also investigated. The PDC method with guard-time performed better in all cases as well. Transmissions could be performed at very high utilization with lower mean delay compared to the other methods.

References

[1] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen, and P. Conrad, "Stream Control Transmission Protocol (SCTP)," *RFC 4960*, 2007.

[2] A. Ford, C. Raiciu, M. Handley, U. C. London, and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses," *RFC 6824*, pp. 1–65, 2013.

[3] M. Zekri, B. Jouaber, and D. Zeghlache, "A review on mobility management and vertical handover solutions over heterogeneous wireless networks," *Computer Communications*, vol. 35, no. 17, pp. 2055–2068, 2012.

[4] G. Cunningham, L. Murphy, S. Murphy, and P. Perry, "Seamless Handover of Streamed Video over UDP between Wireless LANs," in *IEEE conference*, 2004, pp. 1–6.

[5] C. Casetti and W. Gaiotto, "Westwood SCTP: load balancing over multipaths using bandwidth-aware source scheduling," in *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, vol. 4, 2004, pp. 3025–3029.

[6] G. Ye, T. N. Saadawi, and M. Lee, "Independent per Path Congestion Control for Reliable Data Transmission between Multi-homed Hosts," in *IEEE/Sarnoff Symposium on Advances in Wired and Wireless Communication*, 2004.

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

[8] S. J. Koh, M. J. Chang, and M. Lee, "mSCTP for soft handover in transport layer," *Communications Letters, IEEE*, vol. 8, no. 3, pp. 189–191, 2004.

[9] L. Ma, F. Yu, V. C. M. Leung, and T. S. Randhawa, "A new method to support UMTS/WLAN vertical handover using SCTP," *IEEE Wireless Communications*, vol. 11, no. 4, pp. 44–51, 2004.

[10] A. Kelly, G. Muntean, P. Perry, and J. Murphy, "Delay-Centric Handover in SCTP over WLAN," *Transactions on AUTOMATIC CONTROL and COMPUTER SCIENCE*, vol. 49, no. 63, 2004.

[11] S. Kashihara, T. Nishiyama, K. Iida, H. Koga, Y. Kadobayashi, and S. Yamaguchi, "Path selection using active measurement in multi-homed wireless networks," in *Proceedings of the 2004 International Symposium on Applications and the Internet (SAINT'04)*, 2004, pp. 273–276.

[12] J. Noonan, P. Perry, S. Murphy, and J. Murphy, "Simulations Of Multimedia Traffic Over SCTP Modified For Delay-Centric Handover," *World Wireless Congress*, 2004.

[13] J. Fitzpatrick, S. Murphy, M. Atiquzzaman, and J. Murphy, "Evaluation of VoIP in a mobile environment using an end-to-end handoff mechanism," *Mobile and Wireless Communications Summit, 2007. 16th IST*, pp. 1–5, July 2007.

[14] I. Gavriloff and E. P. Ribeiro, "Influence of background traffic on delay-centric path selection algorithm for multihomed SCTP." in *Proceedings of the International Workshop on Telecommunications – IWT'09*, 2009.

[15] R. Runcos and E. P. Ribeiro, "Analysis of multi-homed SCTP and delay-centric transmission for VoIP traffic in lossy networks," in *7th International Telecommunications Symposium*, 2010.

[16] I. Gavriloff, "Analise de aspectos envolvidos no mecanismo de selecao de caminho baseado em atraso para sistemas multiabrigados utilizando SCTP," *M.Sc. Dissertation – UFPR*, 2009.

[17] J. Postel, "Transmission Control Protocol," *RFC 793*, 1981.

[18] V. Paxson, M. Allman, J. Chu, and M. Sargent, "Computing TCP's Retransmission Timer," *RFC 6298*, pp. 1–11, 2011.

[19] A. Torres, "Method for quality improvement in video transmission over SCTP protocol," *M.Sc. Dissertation – UFPR*, 2014.

[20] J. Leung, V. C. M., Ribeiro, E. P., Wagner, A., Iyengard, Ed., *Multihomed Communication with SCTP (Stream Control Transmission Protocol)*. CRC Press, 2012.