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 an ACK of a transmitted packet is received. In the secondary paths, the SRTT is updated only after a heartbeat-ack (HB-ACK) of a sent heartbeat (HB) is received. This does not give these paths a proper estimation of its latency, as the standard interval 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 method occurs when the difference between the SRTT of the current path and the SRTT of an alternate path becomes greater than zero, or greater than a threshold, called hysteresis.

A variation of the delay-centric method includes the utilization of a guard-time [20]. This is the period 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. A random value is selected for each new period to prevent synchronous operation of the sources and to promote better distribution of the handover decision over the time. In this paper, we propose the reduction of the heartbeat interval to 20 ms during guard-time, in order to improve the estimate of the delay over 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 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 use a modified version of predictive delay-centric. 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 an indication 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 will occur if the alternate path has more favorable conditions in terms of latency trend than the primary path, and if conditions (ii) and (iii) are also true.

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 Ethernet link.

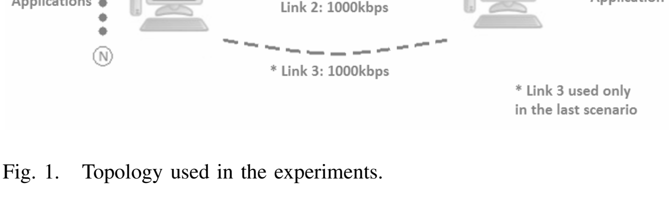

Fig. 1. Topology used in the experiments.

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 (User Datagram Protocol) 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 for different path utilizations. The definition of path utilization was extended to consider a multipath scenario. The aggregated utilization is calculated as

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

where *t* is the number of active transmissions and *A* is the transmission traffic bit rate. The number of active paths is *n* and *B* is the background traffic bit rate. The individual path capacity is *C*. In the experiments, the number of active transmissions (*t*) was 6 and the number of paths (*n*) was 2, except in the last experiment that had a third path. The link capacity was 1 Mbps for each path. The background traffic was zero in the initial test, but then changed to a given proportion of the total traffic in the subsequent tests. In all the simulations, *ρ* was varied from 0.72 to 0.99 by altering the values of *A* and *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 during guard-time = 20 ms).
iii) Predictive delay-centric.
iv) Predictive delay-centric, guard-time = 1 – 4 s (heartbeat interval during guard-time = 20 ms).

Heartbeat interval on all mechanisms were set to a random value between 0.5 s and 1.5 s generated after each heartbeat, except when in guard-time period, when it is changed to 20 ms. Because the bandwidth was limited in the sender machine, the queue delay was originated in the sender side only. The delay on the return path was negligible. To compensate that, we adjusted 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 among all mechanisms in a different scenario.

IV. Results

A. Algorithms' Behavior

The first round of experiments was executed without any background traffic. For a particular aggregated utilization, the average delay was evaluated after 200 repetitions. The obtained histogram of empirical end-to-end delay did not exhibit a single mode, but two modes. One of them is close to zero milliseconds (minimum delay) and the other mode is between zero and the maximum delay. The interpretation of this result is that regarding some initial conditions, the mechanisms were able to distribute all flows over the available paths and achieve low delay for all flows. But other initial conditions led to non-stable behavior and the transmissions kept changing paths. This resulted in large average delays for all flows. Figure 2 displays the average delay as a function of the utilization. All analyzed handover methods are presented in Figure 2 with their respective average packet delay. For each curve, confidence intervals (95%) are also plotted.

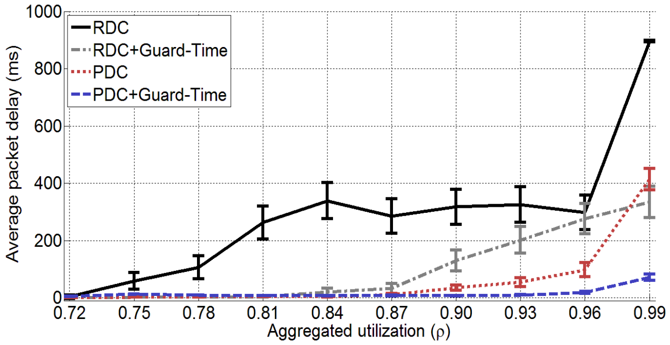

Fig. 2. Algorithms' behavior without background traffic.

For low utilizations ($\rho < 0.75$) the pure reactive delay-centric (RDC) mechanism was able to promote an even distribution of the transmissions over the two paths. As expected in higher utilizations ($\rho > 0.75$), instabilities may occur and the excessive path changes increase the overall mean delay. This instability problem is mainly caused by poor latency estimation of the secondary path, due to fact that the HB intervals (1 s) are much longer than the data packet interval (approximately 20 ms). Shortening these intervals too much may not be a good solution, since the HBs would consume unnecessary bandwidth. The reduction of HB interval during guard-time only proved to be a good compromise yielding improved latency estimation with low overhead. The tests in Figure 2 showed that RDC with guard-time resulted in lower packet delay compared to the pure RDC. It may have given the transmissions a better estimative of the SRTTs, leading to a improved handover decision.

The predictive delay-centric (PDC) method was also evaluated: both the pure predictive delay-centric, and the predictive delay-centric with guard-time and HB interval reduction. The pure PDC algorithm resulted in lower packet delay than the pure RDC, which indicates its ability to avoid instabilities by detecting the trends of SRTTs. Following the behavior of RDC, the guard-time with HB interval reduction implemented in PDC helped to lower the overall delay even more, eliminating almost every case of instability. With this mechanism, even in high utilization (ρ = 0.99), the average delay was not high enough to greatly affect the QoS (Quality of Service) of a VoIP (Voice over IP) call, for example.

B. Background Traffic Scenario

Additional experiments were conducted with background traffic to investigate the performance of the handover mechanisms in more realistic scenario. We noted that the addition of background traffic resulted in an unimodal distribution for all mechanisms.

Figure 3 presents the comparison between the handover methods. In this test, all the sources represent 60% of the total traffic and background traffic represents 40% of total traffic. The mean delay and confidence intervals (95%) for each method were plotted. The predictive delay-centric with guard-time also 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.

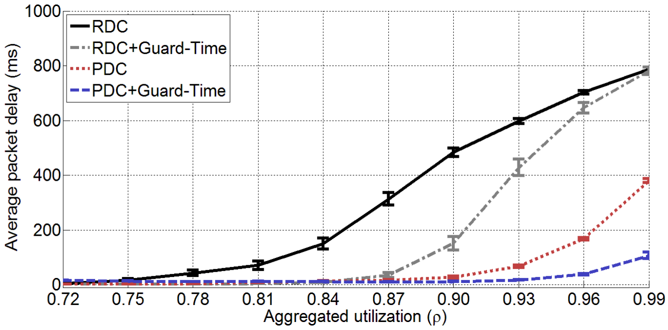

Fig. 3. Scenario with random background traffic.

C. Comparison of Predictive delay-centric methods

A comparison between the original PDC method and the modified PDC method was performed and results are displayed in Figure 4. Without the guard-time, the algorithms acted almost in the same way, and yielded similar mean packet delay values. With the guard-time implemented, the modified PDC method exhibited lower delays at higher utilization levels. The trend of secondary paths could now be calculated more accurately because of the decrease in the HB interval during guard-time, giving the modified algorithm more stability.

D. Algorithms' Behavior with a third path

All methods were tested with 3 paths to verify how the handover algorithm scale with the number of paths. The proportion of background traffic was 40% of total traffic, the same as in previous tests. Figure 5 shows the results for RDC and RDC with Guard-Time, and Figure 6 shows the results for PDC and PDC with Guard-Time. In both figures, the results obtained with 2 paths are also plotted for comparison.

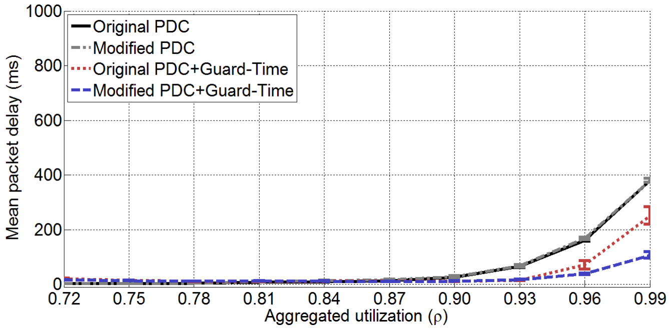

Fig. 4. Comparison between predictive delay-centric methods.

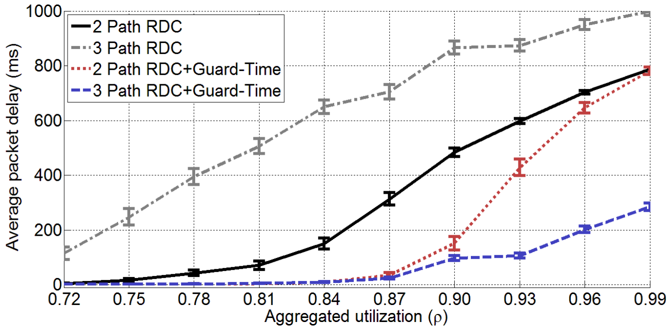

Fig. 5. RDC and RDC+Guard-Time with 3 paths.

The pure reactive delay-centric was unable to keep low delay for the transmissions. Moreover, the instability of this method increased even more with the addition of the third path. Unlike the pure RDC, the RDC with guard-time had a better result with the three paths, promoting good distribution of the transmission among the paths.

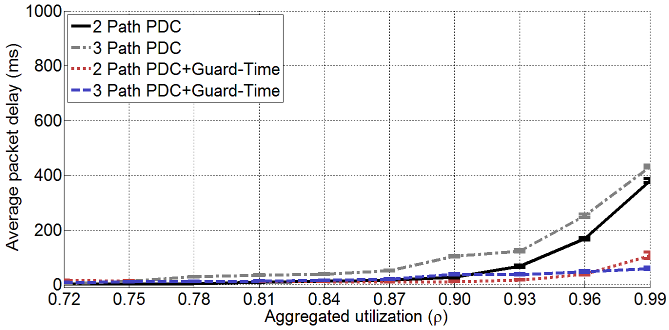

Fig. 6. PDC and PDC+Guard-Time with 3 paths.

In Figure 6, it is possible to see that the differences in the pure PDC method were more discrete, but the addition of the third path also raised the mean delay. The PDC with no guard-time displayed similar result for two and three paths, with low mean delay even for high utilization. The only notable difference was for $\rho = 0.99$. In that case, the third path enabled better performance of the method resulting in lower mean delay.

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 the delay in 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 of the 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.

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.