LEDBAT-MP – On the Application of "Lower-than-Best-Effort" for Concurrent Multipath Transfer

Hakim Adhari*, Sebastian Werner*, Thomas Dreibholz§, Erwin Paul Rathgeb*

*University of Duisburg-Essen, Institute for Experimental Mathematics
Ellernstraße 29, 45326 Essen, Germany
{hakim.adhari, sebastian.werner, erwin.rathgeb}@iem.uni-due.de

§Simula Research Laboratory, Network Systems Group
Martin Linges vei 17, 1364 Fornebu, Norway
dreibh@simula.no

***Abstract*—The Internet is based on best effort communication, i.e. it tries to deliver packets but does not provide any guarantees. A transport protocol can make use of this best effort service to provide a suitable service to its applications. Also, its congestion control is responsible for a fair distribution of the resources within the Internet. However, background data transfer applications (like file sharing or update fetching) do not require "best effort"; they in fact could use a "lower-than-best-effort" service to leave resources to more important applications if needed. For this purpose, the Low Extra Delay Background Transport (LEDBAT) algorithm has been standardized by the IETF.**

Nowadays, multi-homing is becoming increasingly common in modern networks and several approaches to exploit this feature (e.g. CMT-SCTP, MPTCP) have evolved that are able to combine resources of multiple paths. For background traffic oriented algorithms like LEDBAT, this feature could be of great use, too, i.e. by increasing the overall bandwidth while shifting the transmission away from paths which are used by other flows. This could be particularly useful for non-critical bulk transfers in data centres. In this paper, we introduce our approach LEDBAT for Multi-Path – denoted as LEDBAT-MP – and analyze its performance by simulations. With this paper, we want to highlight some generic design questions and start a discussion on how a solid universal background multi-path congestion control strategy should behave.

Keywords: Multi-Path Transfer, Congestion Control, Background Traffic, CMT-SCTP, MPTCP

I. INTRODUCTION

In the last years, communication devices such as laptops and smart phones have become more and more common. One of the important features of these devices is the availability of more than one network interface (e.g. WLAN and UMTS), providing multiple network carriers and access technologies to be part of the communication. This feature of having multiple IPs in a host is denoted as *multi-homing*. Multi-homing implies several possible benefits, such as providing mobility or increasing availability. Furthermore, it makes aggregating bandwidths to achieve throughput benefits possible. This is denoted as *load sharing* or *multi-path transport*. Actually under discussion are end-to-end Transport Layer protocols and protocol extensions, such as Multi-Path TCP (MPTCP [1]) and Concurrent Multipath Transfer for SCTP (CMT-SCTP [2]) that are in an advanced stage of the standardization in the IETF.

One of the issues related with load sharing on the Transport Layer is the congestion control (CC) mechanism used. First approaches applied well-known mechanisms originally designed for single-path TCP [3] or SCTP [4] for multiple paths [2] and led to fairness problems while dealing with bottleneck links [5]. In order to deal with this issue, new CC mechanisms were designed with the inherent ability to manage multi-path scenarios. Strategies based on coupling all the subflows belonging to a flow, such as CMT/RP CC [6] for CMT-SCTP and MPTCP CC [7] for MPTCP were under discussion. [8] proved that the MPTCP CC, also called Linked-Increases Algorithm (LIA), was not pareto-optimal and has severe performance issues and proposed OLIA as a possible solution. [9] discussed generic challenges of fairness for CC in multi-homed systems.

However, all aforementioned CC mechanisms base on the same loss-based strategy. Operation of loss-based CC results in a periodical filling and emptying of router buffers with an associated fluctuation of the queuing delays. Here, limiting the end-to-end delay was not a design goal and the consequence of this behaviour could be buffer bloat [10]. An alternative to the loss-based strategies are the delay-based CC mechanisms. This approach is the base for most algorithms designed to be used in high-speed networks. Some of these proposals, like Compound TCP [11] and TCP Illinois [12], are hybrid loss- and delay-based mechanisms. On the other side, multiple other mechanisms are only delay-based, such as Vegas [13] or variants of it such as Fast TCP [14] or Code TCP [15]. Vegas for example increases or decreases the sending rate, based on the difference on the expected and the actual throughput. The throughput is calculated based on the window size and the measured minimum round trip time (RTT).

Particularly, lower-than-best-effort (LBE) transport protocols are based on delay measurements. An LBE service is defined as service which results in smaller bandwidth and/or delay impact on standard TCP than standard TCP itself when sharing a bottleneck [16]. TCP-Nice [17] and TCP-LP [18] are LBE-based protocols and make use of the RTT or the one-way delays (OWD) in order to be able to use only the bandwidth that is not used by other flows (such as ordinary TCP flows, for example). Another delay-based background transport approach is *Low Extra Delay Background Transport* (LEDBAT) [19]. It also assumes that the increase of the queuing delay is an indicator for congestion and adapts the transmission rate based on the delay variation. To sum up, delay-based mechanisms can be an alternative to loss-based mechanisms, especially for background traffic. Multiple approaches have been discussed and deployed for single-path transfer. However, this is not the case for multi-path transfer. Here, only minor efforts have been performed in the research community despite of everyone agreeing on the benefits of utilizing multiple paths. [20] proposed a delay-based CC for MPTCP based on TCP Vegas CC [13], but no efforts have been made to design a LBE algorithm for multi-path transport.

In this paper, we discuss the benefits and the challenges related to the application of LEDBAT for multi-path transfer. This paper is structured as follows: first, we introduce the basic design of LEDBAT CC and discuss the adaptations required to use this mechanism for multi-path transfer. After that, the considered simulation topologies as well as the used parameters are described. This is followed by the evaluation of the CC mechanisms behaviour in relation with the chosen topologies. Finally, we conclude this paper by mentioning the points where we want to highlight an urgent need for discussion and we describe our future work in this area.

II. THE LEDBAT-MP CONGESTION CONTROL

In this paper, we introduce the LEDBAT-Mult-Path (LEDBAT-MP) CC. LEDBAT-MP is based on the single-path LEDBAT CC.

A. *LEDBAT for Single-Path Transport*

The delay-based CC algorithm LEDBAT [19] was originally designed for background applications and assumes that the increase of the queuing delay is a sign of congestion. This algorithm performs OWD measurements in order to estimate the queuing delay. The minimum delay is traced over a time period and considered as the base delay. The queuing delay is admitted as the difference between the base delay and the instantaneous delay measured. LEDBAT responds to the increase of the queuing delay by decreasing its sending rate and thereby avoiding packet loss. In this way, it makes it possible to maintain an inflicted queuing delay of a predetermined value called *Target*. On concurrency with flows using loss-based CC, LEDBAT is designed to yield quickly to the other flows.

1) Basic Design: LEDBAT uses timestamps of the data messages in order to calculate the OWD. These messages are denoted as *DATA chunks* for SCTP. After the reception of a DATA chunk, the receiver calculates the OWD as the difference between the local time stamp and the remote time stamp and sends the result to the sender. On the sender side and after the reception of a new OWD, the steps performed can be simplified in the following points:

1) current_delay = OWD
2) base_delay = min(base_delay, current_delay)
3) queuing_delay = current_delay - base_delay
4) off_target = (Target - queuing_delay) / Target
5) cwnd += $\frac{\text{GAIN}*\text{off\_target}*\text{bytes\_newly\_acked}*\text{MSS}}{\text{cwnd}}$

In this case, Target is the maximum queuing delay that the algorithm may cause in the network; off_target is a normalized value that makes the congestion window (cwnd) increase or decrease proportionally to the difference between the current queuing delay and the Target. GAIN determines the rate at which the cwnd responds to changes in the delay and is here set to 1. Bytes_newly_acked is the amount of data that just has been acknowledged. The maximum segment size (MSS) describes the size of the largest segment that can be transmitted. For multiple OWDs received with an acknowledgement, step 1 to 3 are repeated, the delays are then stored in an adequate data structure. For the current delays, a fixed number of current delays is maintained. With every new OWD, the oldest one is deleted. Concerning the base delays, a base history is also maintained with n elements. In the history, every element represents the minimum delay measured over 60 seconds.

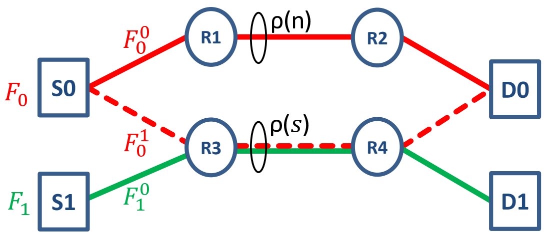
2) Adaptation to SCTP: For our evaluation, we use the SCTP protocol [4]. Therefore, we adapted the LEDBAT CC mechanism to SCTP first: instead of the MSS, as mentioned in the draft [19], the path maximum transmission unit is used (see [21, Section 3.8] for details). In addition to it, since an SCTP packet may include multiple DATA chunks (so-called bundling), only the first DATA chunk will be considered for each packet and only one OWD will be maintained even if multiple DATA chunks are included. With every acknowledgement sent from the receiver (in form of a Selective Acknowledgement chunk, see [4]) to acknowledge data, an additional chunk, denoted as *LEDBAT-CONTROL chunk* is included in the packet. This contains the delays measured and the corresponding path. Furthermore, in order to have as many delay measurements as possible and with that accordingly a better view on the network, delayed acknowledgements [22] have been deactivated.

B. *LEDBAT for Multi-Path Transport*

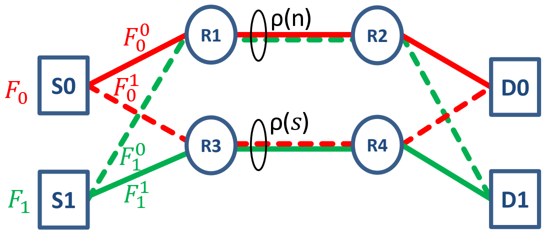
Early MPCC strategies focused on the application of single-path loss-based CC algorithms on every path independently. This led to severe fairness issues on common bottlenecks [5]. In order to deal with this issue, *resource pooling* (RP) [23], which means that multiple resources (here: paths) should behave like a single, pooled resource, have been introduced. It couples the per-path CC mechanisms in order to shift traffic from more congested to less congested paths. Releasing resources on a highly utilized or congested path decreases the loss rate and improves the stability of the whole network. [24] sets three design goals on RP-based multi-path CCs for a TCP-friendly Internet deployment, which have been commonly adopted as fixed rules:

1) *Improve throughput*: a multi-path flow should perform at least as well as a single-path flow on the best path.
2) *Do not harm*: a multi-path flow should not take more capacity on any one of its paths than a single-path flow using only that path.
3) *Balance congestion*: a multi-path flow should move as much traffic as possible off its most congested paths.

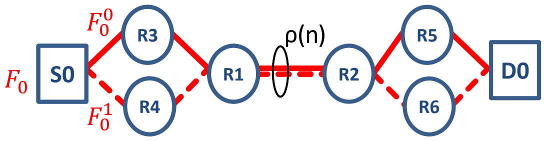
Loss-based MPCC algorithms such as CMT/RP CC [6] or LIA and OLIA [8] are solutions which are approaching the realization of these three rules. Here, resource pooling proved to be a possible solution that could make these MPCC algorithms approach the compliance with these rules. However, it is still not clear whether these rules are also valid and sufficient for LBE MPCC algorithms. Obviously, *improve throughput* should be maintained here. LEDBAT-MP is conform to this rule as it does perform better on multiple paths than on a single path with concurrent flows. However, "performing as well as a



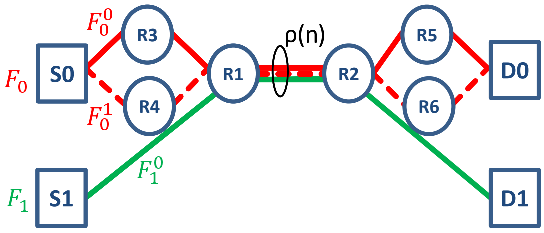
(a) Concurrency on One Path



(b) Concurrency on Two Paths



(c) Shared Bottleneck



(d) Concurrency on Shared Bottleneck

Figure 1. The Considered Scenarios

single flow on the best path" should be understood differently for LBE flows, since they are supposed to underperform on multiplexed paths, hence every gain over the smallest share of bandwidth is an improvement. In addition to it, background CC algorithms are designed to yield to standard TCP-like traffic, even if a non-coupled CC is used. From the design point of view, LEDBAT-MP automatically restrains itself in case of sharing capacity with a loss-based single path flow. This makes it conform to the second rule *do not harm*. Concerning the third rule *balance congestion*, background CC algorithms should be conform to these rules out of the box, since they are designed to not only move as much possible traffic as possible off their most congested paths but to avoid congestion in general. To sum up, LEDBAT-MP can be used as a non-coupled MPCC and is actually from the design point of view conform to the three design goals on RP-based MPCCs.

III. SIMULATION SETUP

For the evaluation, we considered the scenarios that are presented in Figure 1. Let us consider for example the scenario shown in Subfigure 1(b). Here, four communication partners are transferring data through two disjoint paths. The complete capacity of the link *North* is denoted as $\rho(n)$; $\rho(s)$ is the capacity of the link *South*. The flow between S0 and D0 is denoted as F_0 and is composed of two sub-flows (F_0^0 and F_0^1). The bandwidth occupied by F_0^0 is denoted as B_0^0 and the bandwidth occupied by F_0 as B_0.

For our evaluation, we have utilized the OMNET++-based INET framework. The CC mechanisms considered in this paper have been implemented in our CMT-SCTP simulation model [6], [21], [25]. In this work, and in order to avoid buffer blocking issues, the send and the receive buffer sizes on the endpoints have been set to 5,000,000 bytes and buffer splitting [26], [27] as well as NR-SACKs [28] have been used. Unless otherwise specified, the following parameters have been configured: both paths, North and South, have a capacity of 5 Mbit/s. The delay of each independent path has been set to 10 ms. In addition to it, FIFO queues with a maximum size of 100 packets have been configured on the routers. The sender has been saturated (i.e. it has tried to transmit as much data as possible); the message size has been set to 1,452 bytes at an MTU of 1,500 bytes (i.e. full-sized packets). The LEDBAT Target is set to 100 ms, as in [19].

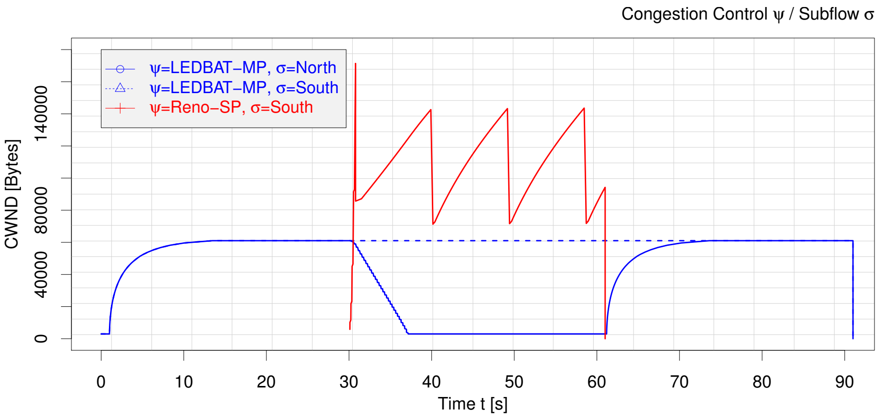
IV. EVALUATION

A. Benefits of LEDBAT-MP

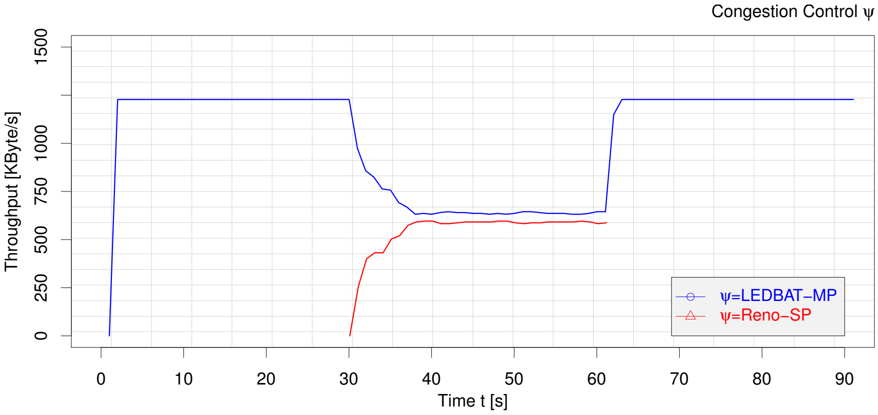
Our first experiment uses the setup depicted in Subfigure 1(a). It can be divided into three steps: in the first 30 seconds, a CMT-SCTP flow using the LEDBAT-MP CC is started alone on both links. We denote this flow in the following as *LEDBAT-MP* flow. In the second step, at t=30 s, a second standard SCTP (i.e. non-CMT) flow using the loss-based Reno CC [4] on the southern path is also scheduled. Hence we denote it as *Reno-SP* flow. The results of this experiment are shown in Figure 2.

The Congestion Window (cwnd) grows quickly and stabilizes after a while (see Figure 2(a)). The sender is able to send enough data to saturate both links and reaches twice the throughput that could be reached by a single-path flow on only one of the paths. The second flow is started at t=30 s. Here, LEDBAT-MP behaves as expected and the cwnd on the southern path is reduced. After a transient phase, the Reno-SP flow is able to eclipse subflow F_0^1 almost completely while subflow F_0^0 is keeping its cwnd and continues to use the northern path efficiently. The dissimilarity caused in this case is handled well due to the large send and receive buffer space available on the endpoints (not to be confused with the buffer space configured on the routers) in addition to SCTP-specific features such as buffer splitting [26], [27] as well as NR-SACKs [28]. At T=60 s, the Reno-SP flow F_1 stops sending and subflow F_0^1 increases its cwnd again and resumes to utilize the southern path efficiently.

In the next scenario (see Subfigure 1(b)), two multi-path flows are considered. Again, the LEDBAT-MP flow is started first. At t=30 s a second multi-path loss-based SCTP flow is started. This flow is denoted as LIA flow, because it is using the LIA CC [8]. Similar to the first scenario, LEDBAT-MP behaves as expected and drives its cwnd down on both paths. At t=50 s, the LEDBAT-MP flow is stopped and – as shown in the throughput graph (see Subfigure 3(b)) for flow F_1 – the throughput reduction reached in the presence of the LEDBAT-MP flow is negligible. That is, the LIA flow F_1 can almost



(a) Congestion Window



(b) Throughput

Figure 2. Simulation Results for Scenario 1(a)

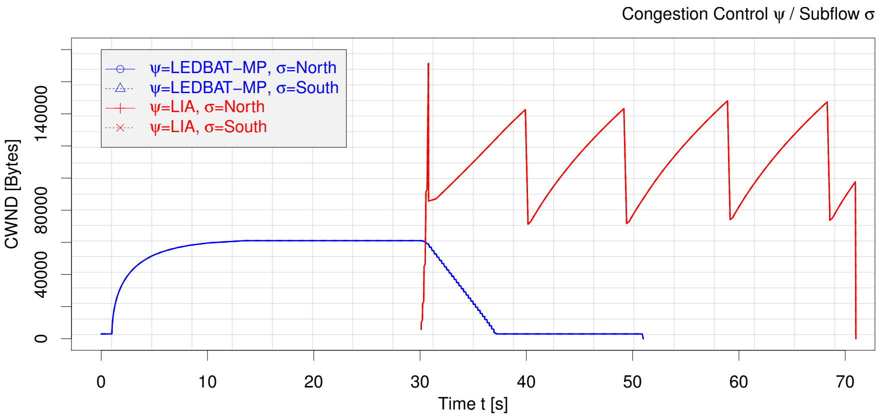
utilize the full capacities of both paths. This is the desired behaviour of a background CC like LEDBAT-MP.

B. Issues Related to LEDBAT-MP

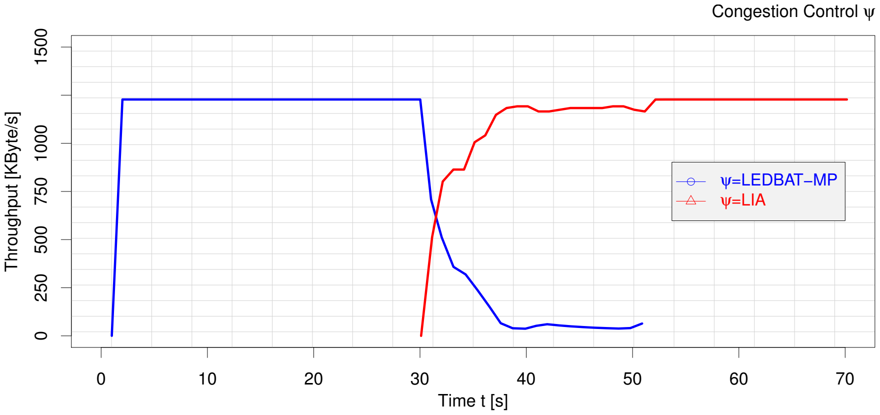
In the second part of our analysis, we want to address some issues with the LEDBAT-MP CC. The first experiment we considered uses the setup in Subfigure 1(c)). In this scenario, we analyzed how LEDBAT-MP handles shared bottlenecks. For this purpose, only one LEDBAT-MP flow with two subflows is used. Each subflow first takes a separate path but they are rejoined at the bottleneck. The results of this experiment are presented in Figure 4. Subfigure 4(a) shows the variation of the cwnd for the duration of 2000 seconds, while Subfigure 4(b) shows the RTT reached for both subflows. Here, we can observe two separate issues:

1) The first issue can be seen in Subfigure 4(a) between t=0 s and t=150 s. One subflow (F_0^1) drives off the other subflow (F_0^0) completely over the course of this interval. Since both subflows belong to the same flow, this has no effect on the resulting overall bandwidth at the bottleneck and is therefore only a minor issue. However, if these subflows would belong to different flows, this would imply serious inter-protocol fairness issues. Furthermore, the load balancing between the first two path segments is practically inexistent. This issue is caused by the detached LEDBAT-MP operation for both subflows: both try to achieve the same target delay, but minor differences in their start timing result in different base delay measurements. These differences are fairly minor with about 2 ms to 5 ms, but the effects are pretty severe. When the OWD for F_0^1 approaches the target delay, F_0^0 has already reached it, since it observed a smaller base delay. F_0^1 continues to increase its cwnd to reach the target delay and therefore pushes the delay of F_0^0 over the threshold. This forces F_0^0 to decrease its cwnd accordingly. At t=150 s, F_0^0 has reached the minimum value of cwnd and the combined throughput stabilizes.

2) Another issue we want to show is related to a periodic reset of the base delay measurements used by LEDBAT-MP for the cwnd calculation. Again in Figure 4, it can be observed that the RTT of both flows suddenly doubles at t=360 s. This effect is repeated at t=720 s, but at this point the available resources are exhausted and instead loss – and therefore a cwnd reduction – occurs. This leads to an unstable state of both LEDBAT-MP subflows, where both flows repeatedly increase their cwnd in order to raise the delay to the newest target delay, which is above the maximum delay for this path. This periodically leads to further loss events. This effect was mentioned by [29] for single-path transfer and – as we see here – it is an even a worse issue for a multi-path flow. This behaviour is caused by the way the base delay is estimated by the LEDBAT-MP algorithm. In fact, the minimum delays are held for a pre-configured time interval (here: 60 seconds). A



(a) Congestion Window



(b) Throughput

Figure 3. Simulation Results for Scenario 1(b)

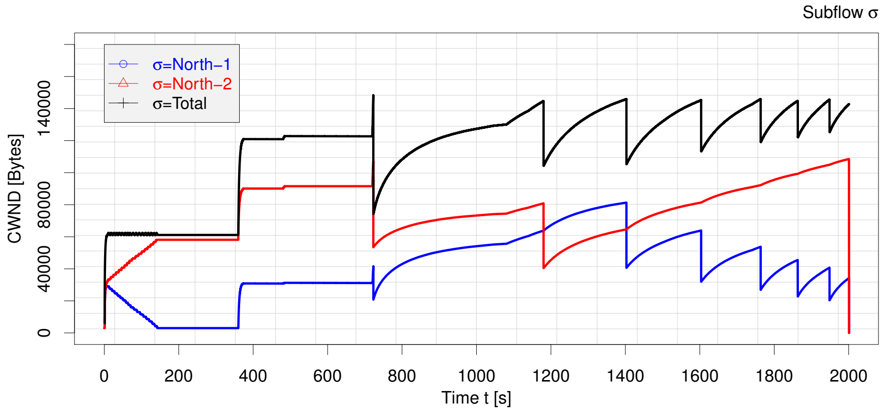
history of these minima is maintained in a sliding window of a pre-configured length (here: 6). These values are maintained for every path separately. After 360 s, i.e. 60 s×6, the sender is supposed to forget the old values which are now assumed to be irrelevant for the transmission.

With the first packets transmitted, both subflows – F_0^0 and F_0^1 – have already measured the most realistic value of the base delay, which is about 10 ms. It then adds the target delay of 100 ms to the path delay since it occupies a small amount of buffer space at the bottleneck. After 360 s, when the algorithm forgets about the delays measured during the first minute, the algorithm now considers the delay of around 110 ms as the new base delay and again tries to add the target delay of 100 ms. When this behaviour is repeated at t=720 s, the algorithm will add another share of the buffer space worth 100 ms in additional delay. In this case, it is more than the buffer can handle. From this point on, loss events cause LEDBAT-MP to drastically decrease the cwnds. Since the maximum path delay for the given bottleneck is lower than the current target delay + base delay, the algorithm never stabilizes again. Curve *Total* in Subfigure 4(a), which indicates the sum of both subflows' cwnds, shows how both subflows adopt a Reno-like behaviour.

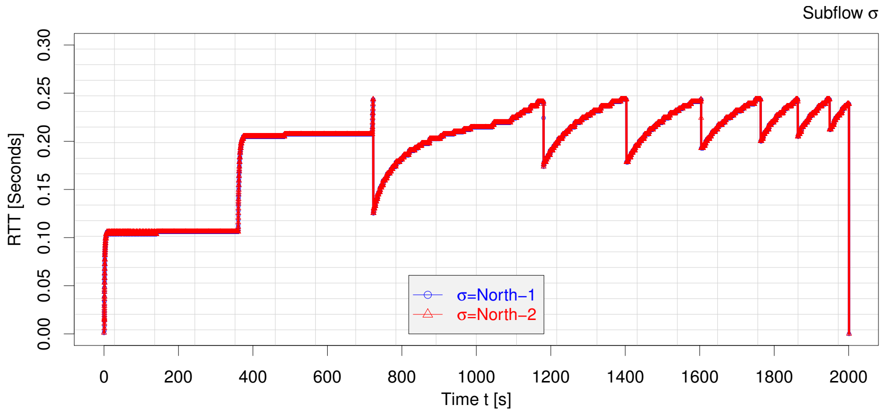
In order to show the difficulties caused by this issue, we extended this experiment to the topology shown in Subfigure 1(d). Here, a new single-path Reno flow is started at t=1400 s. In contradiction to the scenario considered in Subsection IV-A, we see that the impact of LEDBAT-MP proves to be negative. In fact, the LEDBAT-MP flow not only behaves like a background flow (because of the assumptions on the base delay). In addition, there is also a fairness issue: LEDBAT-MP acquires more bandwidth than the single-path flow on the common bottleneck. This is against the three rules mentioned in Section II-B. The second rule *do not harm* says that a multi-path flow should not take more capacity on any one of its paths than a single-path flow using only that path. The results of the experiment shown in Figure 5 show that this is not the case here.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced LEDBAT-MP, an adaptation of the delay-based LEDBAT congestion control strategy for background traffic to multi-path transfer. We furthermore also discussed some design issues that can generally be adopted for future MPCCs. In our evaluation, we have demonstrated the ability of LEDBAT-MP to provide a benefit for multi-path background traffic, at least for a certain time period. However, there is still room for improvement: while LEDBAT-MP already works in well-defined topologies like e.g. for data centre networks (which, of course, is an important use case



(a) Congestion Window



(b) Round-Trip Time

Figure 4. Simulation Results for Scenario 1(c)

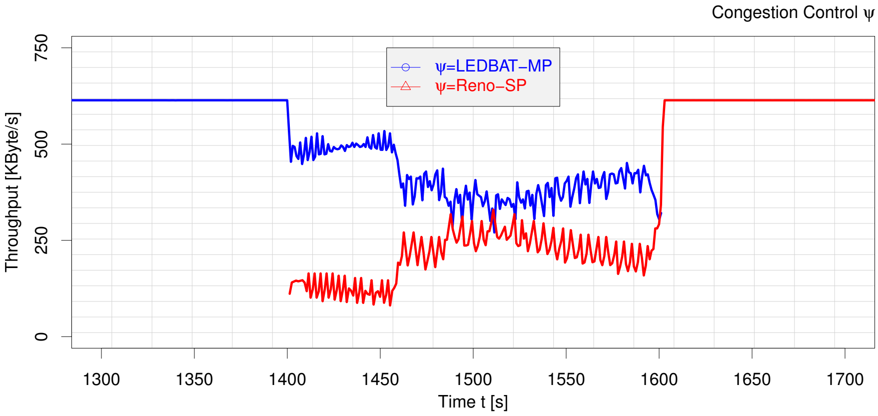


Figure 5. Simulation Results for Scenario 1(d) – Throughput

for this kind of traffic), issues may occur when it is applied in arbitrary setups. That is, a "normal" end-user cannot easily make use of it – e.g. for downloading system updates during a video phone call – at the moment.

The goal of our ongoing work in progress is therefore to make LEDBAT-MP robust for end-user application, by tackling its current shortcomings that we have identified as part of this paper: the fixed addition of a 100 ms target delay on a path is unreasonable, particularly when it comes to highly dynamic 3G/4G networks [30]. In addition, we are also working on a relaxation of the rigid sliding window length and duration for tracing of the base delay over the time, in order to be able to handle long-running flows as well as delay variation due to path changes. Furthermore, we are also working on a theoretical basis that extends the multi-path transport design rules specified by [24] with general conceptual work of multi-path congestion control and fairness [9] to background traffic handling. However, also a practical evaluation of the results in real Internet setups is crucial. Therefore, we are also going to analyze them in reality in the NORNET testbed [31]–[33], a large-scale distributed research platform for multi-homed systems in the Internet. Such a practical analysis is finally also necessary in order to contribute our research results into the ongoing IETF standardization process of the MPTCP and CMT-SCTP protocol extensions, in order to transfer our research to application.

REFERENCES

[1] A. Ford, C. Raiciu, M. Handley, S. Barré, and J. R. Iyengar, "Architectural Guidelines for Multipath TCP Development," IETF, Informational RFC 6182, Mar. 2011, ISSN 2070-1721.

[2] J. R. Iyengar, P. D. 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, pp. 951–964, Oct. 2006, ISSN 1063-6692.

[3] J. B. Postel, "Transmission Control Protocol," IETF, Standards Track RFC 793, Sept. 1981, ISSN 2070-1721.

[4] R. R. Stewart, "Stream Control Transmission Protocol," IETF, Standards Track RFC 4960, Sept. 2007, ISSN 2070-1721.

[5] T. Dreibholz, M. Becke, J. Pulinthanath, and E. P. Rathgeb, "Applying TCP-Friendly Congestion Control to Concurrent Multipath Transfer," in *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, Perth, Western Australia/Australia, Apr. 2010, pp. 312–319, ISBN 978-0-7695-4018-4.

[6] T. Dreibholz, M. Becke, H. Adhari, and E. P. Rathgeb, "On the Impact of Congestion Control for Concurrent Multipath Transfer on the Transport Layer," in *Proceedings of the 11th IEEE International Conference on Telecommunications (ConTEL)*, Graz, Steiermark/Austria, June 2011, pp. 397–404, ISBN 978-953-184-152-8.

[7] C. Raiciu, M. Handly, and D. Wischik, "Coupled Congestion Control for Multipath Transport Protocols," IETF, Standards Track RFC 6356, Oct. 2011, ISSN 2070-1721.

[8] R. Khalili, N. Gast, M. Popovic, and J.-Y. L. Boudec, "MPTCP is not Pareto-Optimal: Performance Issues and a Possible Solution," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1651–1665, Oct. 2013, ISSN 1063-6692.

[9] M. Becke, T. Dreibholz, H. Adhari, and E. P. Rathgeb, "On the Fairness of Transport Protocols in a Multi-Path Environment," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Ottawa, Ontario/Canada, June 2012, pp. 2666–2672, ISBN 978-1-4577-2052-9.

[10] J. Gettys, "What is Bufferbloat, Anyway?" 2011.

[11] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "A Compound TCP Approach for High-speed and Long Distance Networks," in *Proceedings of the 25th IEEE INFOCOM*, Apr. 2006, pp. 1–12, ISSN 0743-166X.

[12] S. Liu, T. Başar, and R. Srikant, "TCP-Illinois: A Loss and Delay-Based Congestion Control Algorithm for High-Speed Networks," in *Proceedings of the 1st International Conference on Performance Evaluation Methodolgies and Tools (VALUETOOLS)*, Pisa/Italy, 2006, ISBN 1-59593-504-5.

[13] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: New Techniques for Congestion Detection and Avoidance," in *Proceedings of the Conference on Communications Architectures, Protocols and Applications (SIGCOMM)*, London/United Kingdom, Aug. 1994, pp. 24–35, ISBN 0-89791-682-4.

[14] C. Jin, D. X. Wei, and S. H. Low, "FAST TCP: Motivation, Architecture, Algorithms, and Performance," in *Proceedings of the 23rd IEEE INFOCOM*, St. Maarten/Netherlands Antilles, Mar. 2004, pp. 2490–2501, ISBN 978-0780383555.

[15] Y.-C. Chan, C.-L. Lin, C.-T. Chan, and C.-Y. Ho, "CODE TCP: A Competitive Delay-Based TCP," *Computer Communications*, vol. 33, no. 9, pp. 1013–1029, June 2010, ISSN 0140-3664.

[16] D. Ros and M. Welzl, "Less-than-Best-Effort Service: A Survey of End-to-End Approaches," *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 898–908, May 2013, ISSN 1553-877X.

[17] A. Venkataramani, R. Kokku, and M. Dahlin, "TCP Nice: A Mechanism for Background Transfers," *ACM SIGOPS Operating Systems Review*, vol. 36, pp. 329–343, Dec. 2002, ISSN 0163-5980.

[18] A. Kuzmanovic and E. W.Knightly, "TCP-LP: Low-Priority Service via End-Point Congestion Control," *IEEE/ACM Transactions on Networking*, vol. 14, no. 4, pp. 739–752, Aug. 2006, ISSN 1063-6692.

[19] S. Shalunov, G. Hazel, J. Iyengar, and M. Kühlewind, "Low Extra Delay Background Transport (LEDBAT)," IETF, Standards Track RFC 6817, Dec. 2012, ISSN 2070-1721.

[20] Y. Cao, M. Xu, and X. Fu, "Delay-based Congestion Control for Multipath TCP," in *Proceedings of the 20th IEEE International Conference on Network Protocols (ICNP)*, Austin, Texas/U.S.A., Nov. 2012, pp. 1–10, ISBN 978-1-4673-2445-8.

[21] T. Dreibholz, "Evaluation and Optimisation of Multi-Path Transport using the Stream Control Transmission Protocol," Habilitation Treatise, University of Duisburg-Essen, Faculty of Economics, Institute for Computer Science and Business Information Systems, Mar. 2012.

[22] I. Rungeler, M. Tüxen, and E. P. Rathgeb, "Improving the Acknowledgment Handling of SCTP," in *Proceedings of the 4th IEEE International Conference on Digital Society (ICDS)*, St. Maarten/Netherlands Antilles, Feb. 2010, pp. 41–46, ISBN 978-1-4244-5805-9.

[23] D. Wischik, M. Handley, and M. B. Braun, "The Resource Pooling Principle," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 5, pp. 47–52, Oct. 2008, ISSN 0146-4833.

[24] C. Raiciu, D. Wischik, and M. Handley, "Practical Congestion Control for Multipath Transport Protocols," University College London, London/United Kingdom, Tech. Rep., Nov. 2009.

[25] T. Dreibholz, H. Adhari, M. Becke, and E. P. Rathgeb, "Simulation and Experimental Evaluation of Multipath Congestion Control Strategies," in *Proceedings of the 2nd International Workshop on Protocols and Applications with Multi-Homing Support (PAMS)*, Fukuoka/Japan, Mar. 2012, ISBN 978-0-7695-4652-0.

[26] T. Dreibholz, M. Becke, E. P. Rathgeb, and M. Tüxen, "On the Use of Concurrent Multipath Transfer over Asymmetric Paths," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, Miami, Florida/U.S.A., Dec. 2010, ISBN 978-1-4244-5637-6.

[27] H. Adhari, T. Dreibholz, M. Becke, E. P. Rathgeb, and M. Tüxen, "Evaluation of Concurrent Multipath Transfer over Dissimilar Paths," in *Proceedings of the 1st International Workshop on Protocols and Applications with Multi-Homing Support (PAMS)*, Singapore, Mar. 2011, pp. 708–714, ISBN 978-0-7695-4338-3.

[28] P. Natarajan, N. Ekiz, E. Yilmaz, P. D. Amer, and J. R. Iyengar, "Non-Renegable Selective Acknowledgments (NR-SACKs) for SCTP," in *Proceedings of the 16th IEEE International Conference on Network Protocols (ICNP)*, Orlando, Florida/U.S.A., Oct. 2008, pp. 187–196, ISBN 978-1-4244-2506-8.

[29] D. Ros and M. Welzl, "Assessing LEDBAT's Delay Impact," *IEEE Communications Letters*, vol. 17, no. 5, pp. 1044–1047, 2013, ISSN 1089-7798.

[30] S. Ferlin-Oliveira, T. Dreibholz, Özgü Alay, and A. Kvalbein, "Measuring the QoS Characteristics of Operational 3G Mobile Broadband Networks," in *Proceedings of the 4th International Workshop on Protocols and Applications with Multi-Homing Support (PAMS)*, Victoria, British Columbia/Canada, May 2014.

[31] E. G. Gran, T. Dreibholz, and A. Kvalbein, "NorNet Core – A Multi-Homed Research Testbed," *Computer Networks, Special Issue on Future Internet Testbeds*, 2014, ISSN 1389-1286.

[32] T. Dreibholz, "NorNet – An Open, Large-Scale Testbed for Multi-Homed Systems," Invited Talk at Swinburne University, Melbourne, Victoria/Australia, Jan. 2014.

[33] T. Dreibholz and E. G. Gran, "Design and Implementation of the NorNet Core Research Testbed for Multi-Homed Systems," in *Proceedings of the 3nd International Workshop on Protocols and Applications with Multi-Homing Support (PAMS)*, Barcelona, Catalonia/Spain, Mar. 2013, pp. 1094–1100, ISBN 978-0-7695-4952-1.