2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)

Evaluating and Modeling 5G MPTCP Performance

Johan Garcia*†, Per Hurtig*, Jonas Hammar*‡

*Karlstad University, Karlstad, Sweden
Emails: {johan.garcia, per.hurtig}@kau.se

†Icomera AB, Gothenburg, Sweden

‡Prevas AB, Karlstad, Sweden
Emails: {jonas.hammar}@prevas.se

Abstract—**Multipath connectivity and aggregation of multiple communication links is actively being researched with the aim to achieve higher throughput and lower latency. In this work we perform an emulation-based evaluation of the relative goodput of MPTCP and TCP in a 5G usage context. A large range of path capacity and delay conditions is explored, for both the primary and secondary paths, with over 2000 different configurations evaluated. Evaluations are performed over eight combinations of MPTCP schedulers and congestion controls. The results show that MPTCP running over two links provide lower goodput than TCP over a single link for the majority of cases. Asymmetry in link conditions is in many cases a major complication for the MPTCP scheduler. To examine the predictability of poor performance, and to obtain further insight on the structure of this phenomena, we perform regression modeling of the relative goodput. In addition to the traditional approaches of Linear Regression and Random Forest, we also employ Symbolic Regression to obtain mathematical expressions capable of providing insight on the path conditions most contributing to poor MPTCP performance. Such regression expressions can be informative when evaluating different schedulers or link aggregation approaches.**

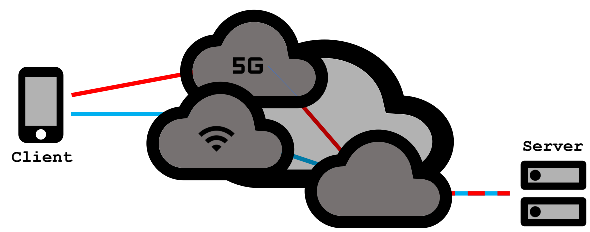

Figure 1: Typical 5G usecase and our experimental setup.

I. Introduction

5G networks are the next generation of mobile Internet connectivity, offering faster speeds and more reliable connections on smartphones and other devices than ever before. Different from the previous generations (e.g., 3G and 4G) of mobile Internet, the 5G standard advocates the use of multi-connectivity to fulfill some of its requirements. For instance, if both a 5G mmWave cellular network interface and a WiFi interface can be employed at the same time (as sketched in Figure 1), it would be possible to aggregate the transmission capacity of the two access technologies, possibly increasing overall capacity and lower the experienced latency. The concurrent use of multiple access networks in 5G is defined by the Access Traffic Steering, Switching, and Splitting (ATSSS) [1] architecture, which mandates the use of Multipath TCP (MPTCP) [2] to deliver multi-connectivity.

The main benefits of using a multipath protocol, such as MPTCP, are typically those hinted earlier; increasing performance by aggregating the capacity of multiple network paths [3], provide seamless handover between network interfaces to enable user mobility, and be more resilient to link failures [4]. However, current protocols that support the use of multiple network interfaces are not necessarily designed to sufficiently consider the quality of the available paths. Instead, many multipath protocols will try to use all available network interfaces for communication. This approach can, however, often be an inappropriate strategy. The poor use of multiple network paths has been identified by previous research [5]–[7], all confirming that slow network paths are utilized too frequently by multipath transports, effectively limiting application performance to that of the poorest path. The upcoming 5G standard will feature a set of radio technologies that are highly asymmetric [8], requiring multipath transports to be more careful when utilizing heterogeneous paths. Within this context, we

- report MPTCP relative performance results over wide-ranging path conditions and transfer sizes,
- construct and evaluate the predictive performance of regressors for MPTCP relative performance,
- employ symbolic regression to generate algebraic expressions to provide insights on underlying relations.

The rest of the paper is structured as follows. Section II gives an overview of related work. Section III details the experiments and the environment used. Section IV describes the results from our experimental campaign, while Section V describes regression modelling on this data. Section VI discuss the interpretability of these models, followed by the conclusions.

978-1-6654-6975-3/22/$31.00

II. Related Work

When transporting data over multiple paths, there are several important aspects to consider. A recent survey of multipath in a 5G context is provided in [9]. The work in [10] tries to identify the most important of multipath transport, and offer a list of benefits including features such as load balancing and capacity aggregation. To assess the usefulness of such features, several studies have evaluated MPTCP's performance in real networks [5], [6], [11]–[13] and found that MPTCP indeed can provide higher goodput than, e.g., TCP.

However, when MPTCP uses network paths that are asymmetric, in terms of capacity and/or delay, the performance benefits starts to fade. MPTCP needs to deliver data in-order to the application, but transporting data over asymmetric paths will result in data arriving out-of-order, resulting in increased latency and possibly reduced goodput due to a number of side effects [12]. This problem has mostly been addressed in relation to applications that require low latency [5], [6], where it has been found that MPTCP sometimes performs worse than a single-path protocol such as TCP. The blame for such poor performance has been put on MPTCP congestion control [14], but even more so on the MPTCP scheduler whose job is to distribute data among the available paths. The standard scheduler of MPTCP [15] is rather simplistic in its design, and not able to fully account for severe path asymmetry [15]. A number of MPTCP schedulers have been proposed to address this deficiency [12], [15]–[18]. However, almost no evaluation have considered a wide range of transfer sizes, as well as asymmetric connections on a scale that represents novel 5G deployments, where both the capacity and delay of the available paths might differ by orders of magnitude. In this work we address the lack of knowledge of MPTCP performance in wide-ranging settings, and also evaluate the ability to predict such performance.

III. Experimental setup

Nine different 5G primary path configurations are considered, constructed as three end-to-end path capacities and three round-trip time (RTT) delays, i.e., 3x3=9 configurations. On the secondary path, four variations that roughly corresponds to different access technologies (3G, 4G, 5G, WLAN) are used, although with some modifications to generate spread in the evaluated conditions. Each access technology is represented in the same 3x3 fashion, thus leading to 4x(3x3)=36 secondary path configurations. The performance is evaluated in terms of goodput of a single transfer. Seven different transfer sizes are considered, ranging from 1 kbyte up to 1 Gbyte in 10x increments.

Data was collected through experiments running on a Linux-based network emulation setup consisting of five hosts. The hosts were connected as illustrated in Figure 1, to emulate a topology with: a client, two wireless access points (primary: 5G, secondary: 3G/4G/5G/WLAN), a server access router and a server. To realize this setup, the mininet network emulator [19] was used together with version 0.95 of the Linux kernel MPTCP reference implementation [20] which implements two MPTCP schedulers and four multipath congestion controls (CCs). Except for some configuration parameters to enable independent and repeatable experiments, e.g., disabling of the TCP metric cache, we consider the default parameterization of both TCP and MPTCP. For setup and script details see [21].

For all experiments, the client connects to the server (over the primary path) and requests a certain amount of data. The server then responds accordingly and closes the connection. During the run-time of an experiment the client will try to open a subflow over the secondary path, over which data can also be transmitted, thus aggregating the capacity of the two paths. The goodput for each experiment was calculated as transfer size divided by the time elapsed from the clients' request to the reception of the last byte of data.

To create an initial TCP baseline, a set of measurements were performed over the range of primary path conditions and transfer sizes, resulting in 9x7=63 TCP baseline goodput values. The main experimental campaign then considered MPTCP connections over these 63 parameter combinations, but where also an additional secondary path is available according to each of the 36 secondary path configurations. In total, 63x36=2268 MPTCP measurements were collected for each of the 2x4 scheduler and CC combinations.

IV. MPTCP Performance

We are interested to examine the MPTCP goodput relative to the goodput achieved by a TCP connection utilizing only the primary path. The link capacity and transfer size parameters span over several orders of magnitude, so a relative performance factor metric z is used as a metric for evaluating MPTCP performance. Here, $z = T_{MPTCP}/T_{TCP}$ where T is the goodput of the respective protocol. Hence, when z is 1 MPTCP and TCP perform identically, and z values of 0.25 and 4 correspondingly mean a fourfold decrease or increase in goodput. To ensure correct model fitting we symmetrize z as $z' = 1 + \log_{10}(z)$ for the regression modeling and evaluation.

A snapshot of results from the experimental campaign is provided in Figure 2 which shows z, expressed as a percentage for clarity in the figure (z' is also shown). On the y-axis the nine combinations of the primary path conditions are given, and here a single transfer size of 10 MB is displayed. On the x-axis, the four blocks correspond to the four considered access technologies with configuration details given in the x-axis labels. The results in Figure 2 are for the default MPTCP scheduler, i.e. Shortest RTT first (SRTT), and congestion control, i.e. LIA. As can be seen, using MPTCP with two paths in many cases result in worse performance than using TCP on the primary path. When 5G configurations are used on the primary and secondary paths, MPTCP over two paths in most cases perform better or similar to TCP over a single path. If another access technology is configured on the secondary path, MPTCP in general performs worse than TCP. A more complete overview of the results for this scheduler and CC is provided in the upper left of Figure 3 which shows the results across five transfer sizes, with layout across the y-axis blocks and x-axis similar to Figure 2. For transfer sizes of 100 kB or

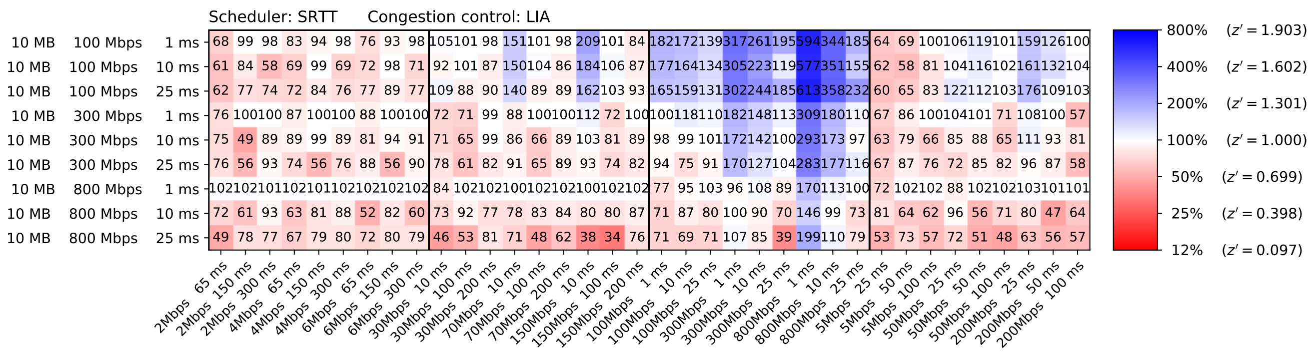

Figure 2: Relative goodput of default MPTCP, expressed as percentage of goodput obtained by TCP, for a 10 MB transfer size. Y-axis labels: transfer size, primary path configuration. X-axis labels: secondary path configuration.

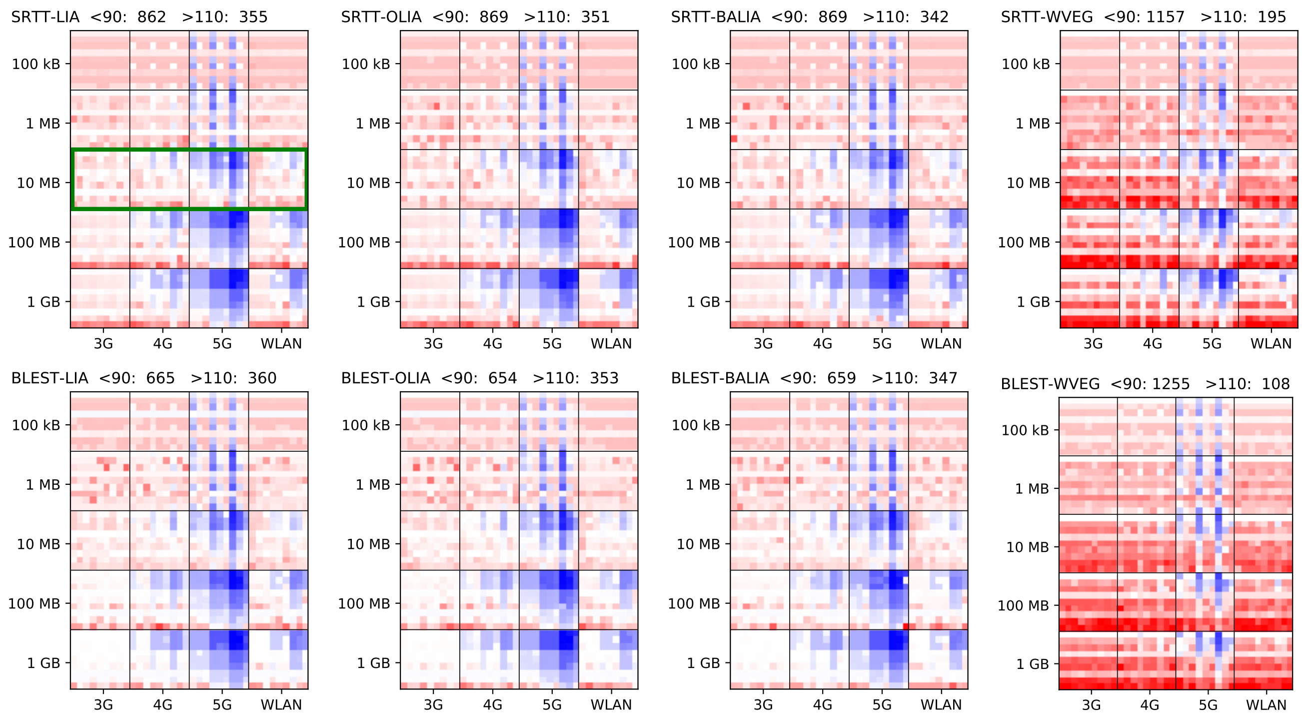

Figure 3: Relative MPTCP vs TCP goodput for different combinations of schedulers and congestion controls. Throughput and delay values on X and Y axes correspond to those given in Figure 2, whose data here is encircled in green. <90: Number of configurations where MPTCP achieves less than 90% of TCP goodput. >110: Same for >110% of TCP.

1 MB, the relative MPTCP performance is worse than for the 10 MB case, only providing improved goodput when RTT on the secondary path is very low (1ms). For these transfer sizes one can observe the entries in the diagonals of the 5G blocks, where the path characteristics are symmetrical, noting that the use of MPTCP here often provides inferior performance even on symmetric paths.

For larger transfer sizes MPTCP generally improves, and it achieves >100% in a few more cases when considering all configuration combinations. Although not shown in the figures, experiments with transfer sizes of 1 kB and 10 kB were also performed. However, the initial congestion window size of TCP/MPTCP is sufficiently large to allow the whole transfer to complete within the first RTT on the primary path. In such circumstances, the C and RTT on the secondary path have no impact and as a results for transfer sizes of 1 kB and 10 kB all results were around 100% and consequently left out from the graphs.

In general, if a lower performance 5G path is used to establish the connection, the addition of a fast secondary path improves the performance. A bit unexpected, however, is the magnitude of MPTCP degradation when asymmetric paths are used. Although previous work have shown MPTCP to perform rather poorly using asymmetric paths, our results indicate extremely bad performance when asymmetry is high. For instance, there are path combinations that result in MPTCP

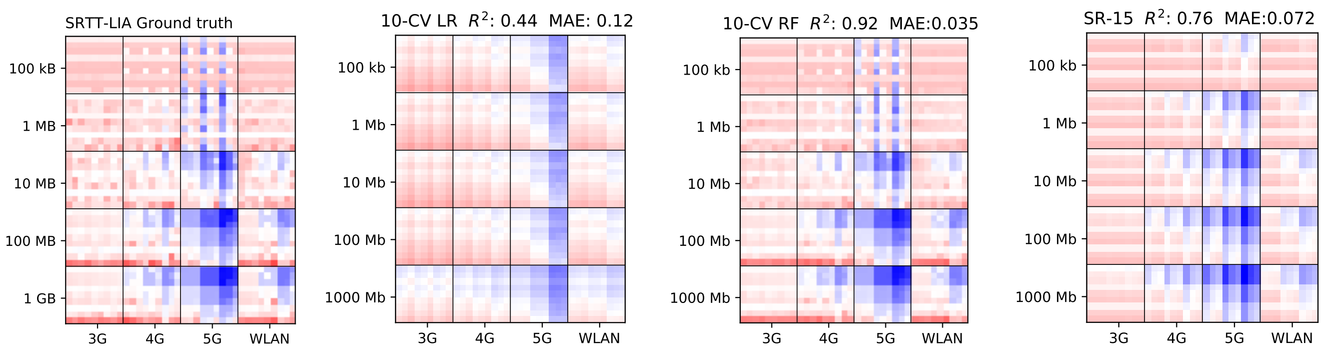

Figure 4: Predictions of relative MPTCP performance for three approaches. Left to right: Ground truth from experiments, Multiple linear regression, Random Forest, Symbolic regression. R^2: Coefficient of determination, MAE: mean absolute error.

only getting 25% of TCP's goodput. To better understand to which extent the relatively poor MPTCP performance is tied to the specific default scheduler and congestion control (CC), a visual comparison can be made between the eight scheduler-CC combinations shown in Figure 3. Such comparison suggests that there are little performance difference between LIA, OLIA and BALIA for the test setup employed here. Weighted Vegas performes markedly worse. The alternative BLEST scheduler shown in the second row improves the performance somewhat, notably by not giving as many instances of poor performance for the large transfer sizes of 100 Mb and 1 GB. For WVEGAS, negative interaction yields even worse performance of WVEGAS with BLEST than the already poor performance provided by WVEGAS under SRTT.

V. Regression modeling

Given that in many circumstances using MPTCP is detrimental for performance in relation to using TCP, we want to consider to what extent a model can be constructed that can predict the performance impact of using MPTCP. Thus, the task at hand is a regression task $z' = f(\mathbf{x}, \theta)$ where function $f(\mathbf{x})$ with parameters θ predict z' based on path conditions and transfer sizes expressed in vector $\mathbf{x}$. We model the defult SRTT-LIA case, and consider three different modeling approaches.

Multiple linear regression (LR) [22] is an extension of ordinary least-squares regression to consider several independent variables to predict the dependent variable. For LR, $f(\mathbf{X}) \equiv \mathbf{X}\mathbf{a}+b$ where $\mathbf{X}$ here is a 1620x5 matrix of values for the independent variables (i.e. configuration values). We have $\mathbf{z}'$ as a vector holding the 1620 dependent variable values. By minimizing, $\mathrm{argmin}_{\mathbf{a},b} ||\mathbf{X}\mathbf{a} - \mathbf{y} + b||_2$ we obtain the LR coefficient vector, $\mathbf{a}$, and intercept b, which for LR jointly constitute θ. While severely limited in expressive power, this type of straightforward linear model provides a baseline for regression performance.

Random forest regression (RF) [23] is a machine learning approach using an ensemble of decision trees. This allows for considerably more expressive models at the cost of interpretability and some associated potential for model overfitting. RF is advantageous in that it is robust to hyperparameter selection, and has been shown to be among the best general machine learning approaches [24] for the type of problem at hand here.

Symbolic regression (SR) [25] is an approach that not only search to estimate θ as in LR, but rather searches for both $f(\cdot)$ and the associated θ. The SR approach provides more interpretability than RF but has less ability to model complex interactions. SR has a low risk of overfitting since the model is a mathematical expression subject to suitable complexity constraints, with few free parameters.

We create LR and RF models using SciKit-learn [26], and the SR model using PySR [27]. The LR and SR models are constructed using 10-fold cross-validation (CV) such that in each fold 90% of the data is used for training, and the unseen 10% is predicted. The combined test data from all folds are shown for each model type in Figure 4, where the leftmost subfigure shows the SRTT-LIA ground truth of Figure 3. Also provided is coefficient of determination (R^2) [28] and mean absolute error (MAE). The R^2 reflect the proportion of variation of the dependent variable that is explained by the model. For a perfect regressor R^2 is 1, and R^2 goes to 0 when the model has no predictive capability.

While the LR modeling approach is widespread, the results indicate poor performance of LR for this problem. By visual inspection it is clear that the LR model manages to capture some of the global characteristics of the underlying system behavior, but the performance figures highlight the poor quality of the model. To explore if the LR performance could be improved, we created additional log transformed features and explored alternate regression approaches. The best linear model was obtained when using Lasso regression with BIC for model selection, yielding a R^2 of 0.59 and z' MAE of 0.1.

A visual comparison between the RF model and the GT highlights the likeness of the model output to the GT, and the performance metrics corroborate this. Finally, the SR model is able to capture considerably more of the system characteristics than LR, but is performing worse than RF from a regression performance metrics perspective. However, the SR approach has considerable advantages in terms of model interpretability.

Table I: Model complexity and interpretability versus model performance

	Random Forest			Symbolic Regression			
#	RF features	z' R^2	z' MAE	SR model	z' R^2	z' MAE	SR Params θ
1	C_2, Size	0.42	0.114	$\sqrt[4]{a + \frac{C_2}{C_1}}$	0.38	0.119	a=0.5588
2	C_2, Size, C_1	0.65	0.0911	$\sqrt[4]{a + \frac{bC_2 \sqrt[4]{Size}}{C_1}}$	0.63	0.0932	a=0.5864, b=0.006430,
3	C_2, Size, C_1, RTT_2	0.66	0.0865	$a + \frac{b}{\frac{C1}{\sqrt{C_2}\sqrt[4]{\frac{Size}{RTT_2}}} + c}$	0.70	0.0851	a=0.8537, b=0.06658, c=0.05354
4	C_2, Size, C_1, RTT_2, RTT_1	0.92	0.0348	$\sqrt[8]{\frac{a}{\sqrt{RTT_1}} + \frac{(C_2+\sqrt{RTT_2})\left(\sqrt[8]{Size}-b\right)}{(C_1-c)\sqrt{RTT_2}}}$	0.76	0.0722	a=0.6729, b=5.236, c= 54.15

VI. Model interpretability

In addition to employing a model for prediction, statistical or machine learning modeling can be employed to obtain useful knowledge regarding the underlying behavior of the modeled system. The level of interpretability is by necessity heavily dependent on the particular model construction method, and we now consider the interpretability of the regression models shown in Figure 4. While it is straightforward to get the coefficients from the LR model, it is of questionable value to consider them given the poor model performance.

The RF approach provides excellent regression performance, but given the ensemble nature of the model it is hard to obtain deeper model understanding. For RF, we instead investigate the relative importance of the features affecting the MPTCP performance, to provide additional clues for the mechanism underlying the observed results, and to provide a baseline to compare the SR results. For RF models, feature importance information can be extracted from the trained model. By analyzing the out-of-bag reduction of the node split criterion, a feature ranking can be obtained from the model itself. We use the determined feature importance to create RF models with varying number of features, and provide the resulting regression performance metrics in the left side of Table I. The increased model expressiveness of RF over LR allows it to perform similarly to basic LR even when considering only two features. Adding more features improves performance, and a final step improvement is obtained for RF #4 which employs the full feature set.

The mathematical expression provided by SR gives this approach excellent interpretability and the potential for creating useful insights regarding the structure of the underlying system. The SR search procedure tries to jointly minimize a loss metric, and a model complexity metric. The loss metric computes the error distance between the predicted $\hat{z}'$ and the true z'. For our SR invocations, we utilized a tailored loss function of $|\hat{z}' - z'|^{1.5}$. Model complexity is measured as the number of operators and operands in the mathematical expression. When searching for a single best expression, a trade off exists between decreasing loss and decreasing complexity. To perform this trade off, scoring functions can be used as discussed in [29]. The joint loss-complexity optimization thus fundamentally creates a Pareto front rather than a single best solution. In our case, we select informative expressions from the Pareto front of several SR invocations, and provide these in the right side of Table 1. A zero SR model a, i.e. consisting only of the average, will have $a = 0.9550$ and provides $z'R^2 = -0.01$ and z' MAE$= 0.142$. For some intuition regarding the magnitude of z' MAE refer to the colorbar in Figure 2. We note that the SR fitting, i.e. joint search of $f(\cdot)$ and θ, for computational reasons was done on the whole data set without using CV. This may raise potential concerns of overfitting. However, separate tests performed using 10-CV indicated CV-stability for locating $f(\cdot)$, and when estimating θ for a specific identified $f(\cdot)$ the range of the extreme values between all folds were within a few percent. This suggests low risk of parameter overfitting, but further evaluation of this is planned for future work.

SR model #1 provides a hint of what performance a very basic model can provide, and illustrates that the SR search at this complexity level identified path capacity asymmetry as the most informative pair of features. The second SR model adds transfer size as a feature and one free parameter, while retaining the capacity asymmetry influence. Model #3 adds the RTT_2 feature and one additional free model parameter. Here it can be noted that the parameters in many cases can provide meaningful interpretations. For example, different schedulers can be compared and their generated experimental data fitted to model #3. For that model, it is desirable for parameter a to be as large as possible since a gives a lower bound on the fitted models predicted z' values. The model fits to a higher a when the data used for fitting reflect higher z', i.e. the scheduler generating the data has better relative MPTCP performance. Similarly, the b and c parameter can be interpreted as reflecting the amount of variability due path asymmetries the model fits to, and a corresponding argument can be made for them. Finally, model #4 provides the best regression performance. Here it becomes somewhat harder to disentangle the contributions of the different features, illustrating the modeling trade off between performance versus complexity.

We note that the SR expressions identified here are not modeling the exact mechanisms of the underlying system in the sense that the SR expressions discovered in [30] does. A reflection of this is that the SR expressions, with the exception of SR #1, are not consistent with the expected unit. Here, the expectation is a unit-less number as z and z' represent a ratio. The SR expressions can rather be viewed as fitted approximations of the observed behavior, and useful for the understanding they nevertheless can convey.

In summary, we believe that the use of SR opens up for exploring behavioral comparisons of MPTCP variations in an exciting manner. Different schedulers, congestion controls (CCs), link aggregation approaches and parameterizations can be examined. By considering an appropriate set of SR expressions, comparisons of model fit and θ parameter values can be made, thus allowing for insights to be formed regarding the relative performance and sensitivity to capacity or RTT asymmetries.

VII. Conclusions and Future work

We have performed a comprehensive experimental campaign evaluating Linux MPTCP versus TCP. Our results highlight that MPTCP provides lower goodput than TCP in the majority of cases as asymmetry in the path conditions hinders proper utilization of both paths. Previous work have noted similar phenomena in smaller experiments, but we evaluate all MPTCP schedulers and congestion controls currently included in the kernel, cover a considerably wider parameter space, and provide a unified visual presentation of the 12000+ measured values.

In addition, we examine the corresponding MPTCP relative performance regression task, and find that a random forest regressor provides strong efficacy with an R^2 of 0.92. Further, we utilize symbolic regression modeling to derive analytical models that can provide insights on the factors influencing MPTCP relative performance, and their structural relationships. Future work includes studying additional environmental factors such as queue depth, competing traffic, varying link capacity, alternative link aggregation techniques, and more. From a data perspective a more comprehensive data set covering an even larger parameters space, potentially with a finer resolution, could be used. Additional work could also be spent on guided search strategies for symbolic regression.

Acknowledgments

Parts of this work was supported by the Swedish Internet Foundation, and the Swedish Knowledge Foundation.

References

[1] M. Boucadair *et al.*, "3gpp access traffic steering switching and splitting (atsss)-overview for ietf participants," *Internet-Draft draft-bonaventure-quic-atsss-overview-00, IETF*, 2020.

[2] A. Ford, C. Raiciu, M. J. Handley, O. Bonaventure, and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses," RFC 8684, Mar. 2020.

[3] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "Design, implementation and evaluation of congestion control for multipath TCP," in *NSDI*, 2011, pp. 99–112.

[4] C. Paasch and O. Bonaventure, "Multipath TCP," *Communications of the ACM*, vol. 57, no. 4, pp. 51–57, Apr. 2014.

[5] S. Ferlin, T. Dreibholz, and Ö. Alay, "Multi-path transport over heterogeneous wireless networks: Does it really pay off?" in *IEEE Global Communications Conference*, 2014, pp. 4807–4813.

[6] B. Han, F. Qian, S. Hao, and L. Ji, "An anatomy of mobile web performance over multipath TCP," in *CoNEXT*, Dec. 2015.

[7] K. Yedugundla, S. Ferlin, T. Dreibholz, Ö. Alay, N. Kuhn, P. Hurtig, and A. Brunstrom, "Is multi-path transport suitable for latency sensitive traffic?" *Computer Networks*, vol. 105, pp. 1–21, 2016.

[8] NGMN Alliance, "NGMN 5G white paper," 2015, https://www.ngmn.org/5g-white-paper/.

[9] H. Wu, S. Ferlin, G. Caso, Ö. Alay, and A. Brunstrom, "A survey on multipath transport protocols towards 5g access traffic steering, switching and splitting," *IEEE Access*, vol. 9, 2021.

[10] S. Habib, J. Qadir, A. Ali, D. Habib, M. Li, and A. Sathiaseelan, "The past, present, and future of transport-layer multipath," *CoRR*, vol. abs/1601.06043, 2016.

[11] Y.-C. Chen, Y.-s. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley, "A measurement-based study of MultiPath TCP performance over wireless networks," in *IMC*. ACM, 2013, p. 455–468.

[12] P. Hurtig, K.-J. Grinnemo, A. Brunstrom, S. Ferlin, Ö. Alay, and N. Kuhn, "Low-latency scheduling in MPTCP," *IEEE/ACM Transactions on Networking*, vol. 27, no. 1, pp. 302–315, 2019.

[13] Q. De Coninck, M. Baerts, B. Hesmans, and O. Bonaventure, "Observing real smartphone applications over multipath TCP," *IEEE Communications Magazine*, vol. 54, no. 3, pp. 88–93, 2016.

[14] T. Gilad, N. Rozen-Schiff, P. Godfrey, C. Raiciu, and M. Schapira, "MPCC: online learning multipath transport," in *CoNEXT*, 2020.

[15] C. Paasch, S. Ferlin, Ö. Alay, and O. Bonaventure, "Experimental evaluation of multipath TCP schedulers," in *2014 ACM SIGCOMM Wshop on Capacity Sharing Workshop*, ser. CSWS '14, 2014, p. 27–32.

[16] N. Kuhn, E. Lochin, A. Mifdaoui, G. Sarwar, O. Mehani, and R. Boreli, "DAPS: Intelligent delay-aware packet scheduling for multipath transport," in *IEEE Int Conf on Comm (ICC)*, Jun. 2014, pp. 1222–1227.

[17] F. Yang, Q. Wang, and P. D. Amer, "Out-of-order transmission for in-order arrival scheduling for multipath TCP," in *28th Intnl Conf on Advanced Information Networking and Applications Workshops*, May 2014, pp. 749–752.

[18] Y.-s. Lim, E. M. Nahum, D. Towsley, and R. J. Gibbens, "ECF: An MPTCP path scheduler to manage heterogeneous paths," in *CoNEXT*. ACM, 2017, p. 147–159.

[19] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks," in *9th ACM SIGCOMM Workshop on Hot Topics in Networks*, ser. Hotnets-IX, 2010.

[20] C. Paasch, S. Barre *et al.* Multipath TCP in the Linux Kernel. [Online]. Available: https://www.multipath-tcp.org

[21] J. Hammar, "Multipath packet scheduling for 5g systems," Master's thesis, Karlstad University, 2022.

[22] C. L. Lawson and R. J. Hanson, *Solving least squares problems*. SIAM, 1995.

[23] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[24] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *JMLR*, vol. 15, no. 1, pp. 3133–3181, 2014.

[25] J. R. Koza, "Genetic programming as a means for programming computers by natural selection," *Statistics and computing*, vol. 4, no. 2, pp. 87–112, 1994.

[26] F. Pedregosa, G. Varoquaux *et al.*, "Scikit-learn: Machine learning in python," *JMLR*, vol. 12, pp. 2825–2830, 2011.

[27] M. Cranmer, "Pysr: Fast & parallelized symbolic regression in python/julia," Sep. 2020. [Online]. Available: http://doi.org/10.5281/zenodo.4041459

[28] S. Wright, "Correlation and causation," *Journal of Agricultural Research. 1921;XX(7):557–585*, vol. XX, no. 7, pp. 557—-585, 1921.

[29] M. Cranmer, A. Sanchez-Gonzalez, P. Battaglia, R. Xu, K. Cranmer, D. Spergel, and S. Ho, "Discovering symbolic models from deep learning with inductive biases," *NeurIPS 2020*, 2020.

[30] S.-M. Udrescu and M. Tegmark, "Ai feynman: A physics-inspired method for symbolic regression," *Science Advances*, vol. 6, no. 16, 2020.