

TCP ROCCET Performance Report

lukas.prause@ikt.uni-hannover.de

I. INTRODUCTION

Since the publication of TCP ROCCET [1], we further developed the congestion control. In this document, we present a short performance analysis of the current implementation of TCP ROCCET available in our Git repository [2]. During the Linux kernel review process, we improved the implementation and added new functionalities to the ROCCET algorithm. These are probing for a new minimum RTT, packet pacing, and ECN handling. The remainder of this document is structured as follows: in Sec. II we describe the min RTT probing; in Sec. III, the use of packet pacing. In Sec. IV, we show measurement results in stationary 5G NR scenarios, and in Sec. V, we present the bandwidth share of TCP ROCCET with other congestion control. The last section summarize and conclude this report.

II. MINIMUM RTT PROBING

TCP ROCCET probes for a new minimum RTT if none has been found within the last 5 seconds. To this end, TCP ROCCET enters *RTT_PROBE* and halves its congestion window for a calculated probing interval. The probing interval has a duration of one RTT, but at least 200 ms. After the probing duration, TCP ROCCET enters *RTT_PROBE_REFILL* and the congestion window is set to its previous value. In addition to refilling the pipe carefully, the congestion window remains unchanged for the duration of the probing interval.

If a packet loss occurs during *RTT_PROBE*, TCP ROCCET stays in its current state but reduces the saved previous congestion window to set it after the end of *RTT_PROBE_REFILL*. If a packet loss occurs during *RTT_PROBE_REFILL*, the current phase ends and the congestion window gets reduced.

III. PACKET PACING

Besides ROCCET and BBR, all congestion control variants in the Linux kernel use the default packet pacing ratio. After each congestion window update, the new sending rate is calculated. The pacing rate depends on the phase of the congestion control. In slow start it is 200 % of the sending rate, and in congestion avoidance, it is 120 % of the sending rate. ROCCET uses 200 % in LAUNCH, like the default for slow start, and 100 % in the congestion avoidance phase ORBITER.

IV. STATIONARY 5G NR MEASUREMENT RESULTS

To show the performance of TCP ROCCET, we conducted 4 measurement series with CUBIC, ROCCET, and BBRv3. For each series, we performed 30 measurements per congestion control; each transmission is 30 sec long. The measurements are conducted in the downlink of a major real-world 5G NR and NSA-NR network.

In Fig. 1, the results for the NSA-NR measurement series with good and bad channel conditions are shown. For good conditions (Fig. 1a), it is observable that ROCCET, CUBIC, and BBRv3 achieve similar performance regarding latency and goodput. For bad channel conditions (Fig. 1b), we observe the highest latency with CUBIC, but also the highest mean goodput. When we compare BBRv3 with ROCCET we can observe similar goodput (ROCCET has ~ 7 Mbps less in mean than BBRv3), but lower latency. Here, nearly 75 % of ROCCET's latency is lower than 50 % of BBRv3's latency. In mean the difference is 19 ms.

In Fig. 2, we show the results for our SA-NR measurement series with good and bad channel conditions. For good channel conditions, our results are similar to those in our SA-NR measurement series. For good channel conditions (Fig. 2a), CUBIC, ROCCET, and BBRv3 achieve similar goodput, but CUBIC causes higher latency. For bad channel conditions (Fig. 2b), we observe the highest latency for CUBIC. When we compare ROCCET with BBRv3, we observe lower latency with ROCCET by maintaining ~ 6 Mbps less in mean than BBRv3. On the other hand, the latency is in mean 17 ms less.

V. TCP BANDWIDTH SHARE

TCP ROCCET is primarily developed for use in cellular links. In cellular networks, a fair bandwidth share is usually achieved by the base station's scheduler. In addition, TCP flows are separated and do not share a buffer with other TCP flows. Since a TCP sender can not determine the network architecture or the used technology at the bottleneck link, it is likely that ROCCET has to face competing TCP flows. Therefore, we conducted bandwidth share experiments for the link characteristics of $50 \text{ Mbps} \times 30 \text{ ms}$ and $10 \text{ Mbps} \times 40 \text{ ms}$. In these experiments, 1 to 32 ROCCET flows compete with a single RENO, CUBIC, BBRv1, BBRv3, and ROCCET flow with bottleneck link buffer sizes of 1 to 64 BDP. For each combination of parameters, we conduct 5×2 min long measurements. The results are shown in Fig. 3.

For a link of $50 \text{ Mbps} \times 30 \text{ ms}$ (Fig. 3a), it is observable that ROCCET can compete with RENO and CUBIC, but not with BBRv1 and BBRv3. It is also interesting that the bandwidth share increases with BBRv1 and decreases with BBRv3 by increasing the size of the buffer. It is also observable that with increasing the number of ROCCET flows, the bandwidth share increases.

For a link with $10 \text{ Mbps} \times 40 \text{ ms}$ (Fig. 3b), in addition to the previous results, it is observable that the bandwidth share with RENO and CUBIC decreases if the bottleneck link buffer is larger than 4 BDP.

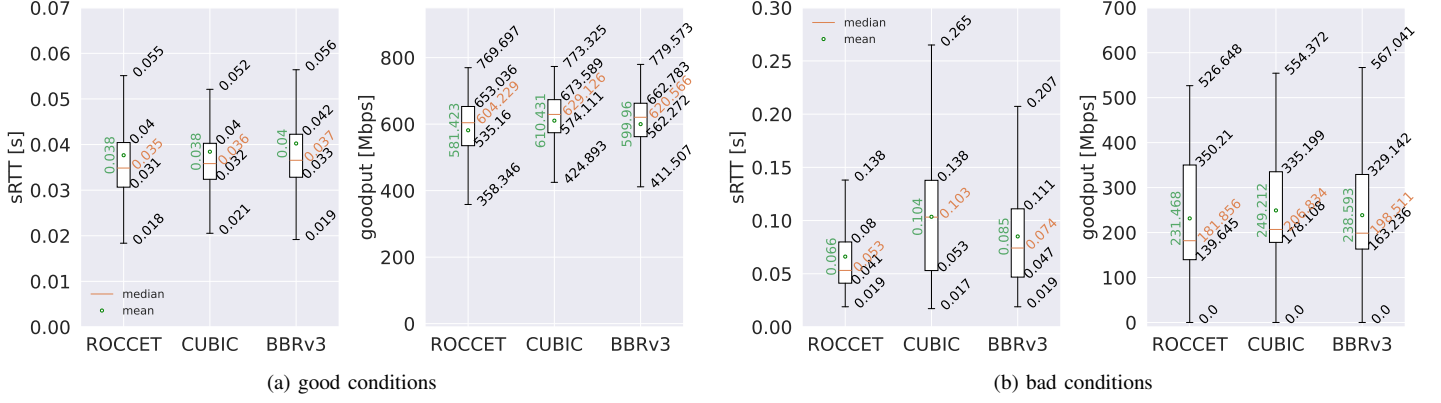


Fig. 1: Goodput and latency for ROCCET, CUBIC and BBRv3, measured in a major real-world **SA-NR** network. Measurement series were conducted for good and bad channel conditions.

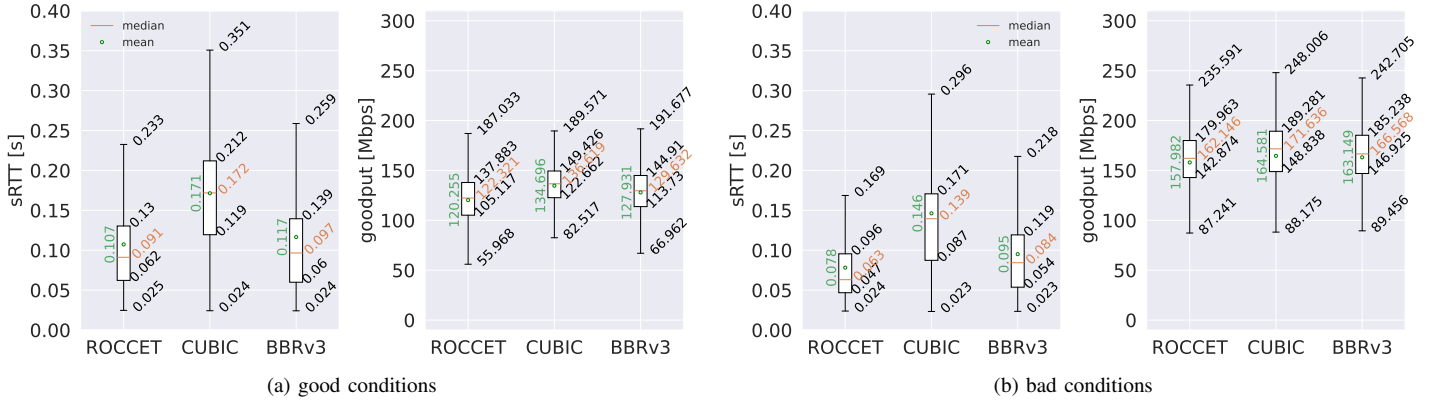


Fig. 2: Goodput and latency for ROCCET, CUBIC and BBRv3, measured in a major real-world **NSA-NR** network. Measurement series were conducted for good and bad channel conditions.

For both link characteristics, we can observe a nearly equal bandwidth share if ROCCET competes with other ROCCET flows, with a few exceptions.

VI. SUMMARY AND CONCLUSION

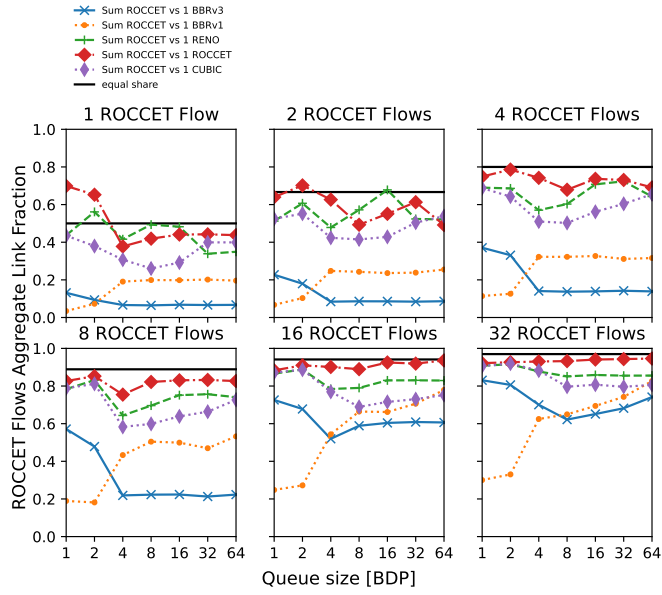
In this report, we provide an overview of the performance of the current TCP ROCCET implementation in our Git repository [2]. We describe the mechanisms added since the first release and provide measurement results in two different real-world major NR networks. In addition to these measurements, we analyzed the bandwidth share of TCP ROCCET with other common congestion control algorithms.

Our results show that TCP ROCCET can achieve a similar goodput and latency to BBRv3, while the network channel is in a good condition. If the channel conditions are worse, TCP ROCCET achieves a similar goodput to BBRv3, by maintaining lower latency.

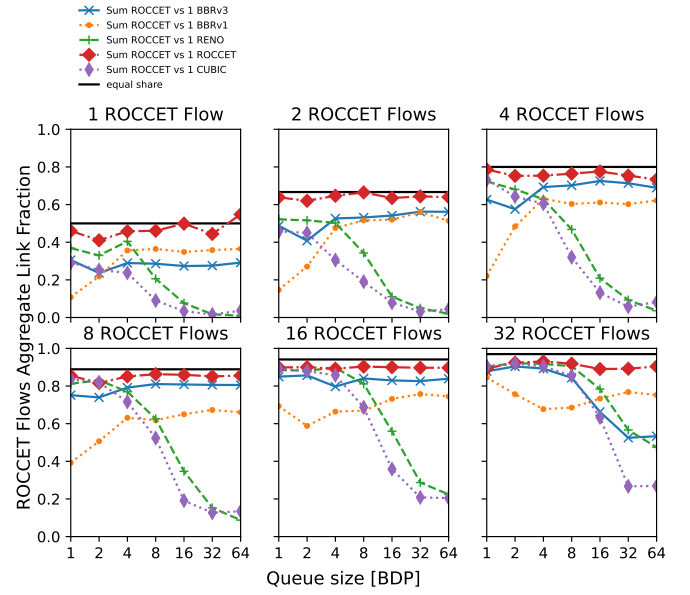
In addition to these results, our bandwidth share experiments show that TCP ROCCET competes with CUBIC and RENO at a bottleneck link buffer, with a small penalty.

REFERENCES

- [1] L. Prause and M. Akselrod, "TCP ROCCET: An RTT-Oriented CUBIC Congestion Control Extension for 5G and Beyond Networks," 2026 21st Wireless On-Demand Network Systems and Services Conference (WONS), Crans Montana, Switzerland, 2026, pp. 105-112, doi: 10.23919/WONS68803.2026.11501781.
- [2] L. Prause and T. Füchsel, "TCP ROCCET Kernel Module Git Repository" <https://gitlab.uni-hannover.de/lukas.prause/tcp-roccet-kernel-module>



(a) $50 \text{ Mbps} \times 30 \text{ ms}$



(b) $10 \text{ Mbps} \times 40 \text{ ms}$

Fig. 3: Bandwidth share of 1 to 32 ROCCET flows competing with a single BBRv3, BBRv1, RENO, CUBIC, or ROCCET flow. Link characteristics are $10 \text{ Mbps} \times 40 \text{ ms}$ and $50 \text{ Mbps} \times 30 \text{ ms}$ with a bottleneck link buffer size of 1 to 32 BDP.