

2 Motivation

Legacy congestion control protocols such as TCP and its variants are known to perform inefficiently over cellular networks due to highly variable channel capacities over short time scales and unpredictable changes in channel latency [WB11, WSB⁺13]. These channel properties lead to two considerable shortcomings of legacy congestion control protocols: First, packet loss is not linked to congestion in the network or access link. Due to the nature of cellular channels, (e.g., hand-over, changing channel capacities, etc.) packets may be discarded at any point in time without experiencing an actual congestion in the network. That is, most legacy congestion control protocols treat lost packets as an indication of a congested network and hence react (incorrectly) to the cause of the packet loss by lowering the sending rate. Second, due to the design of legacy congestion control protocols, channel changes (either improvement or degradation) are mostly detected too late and the sender does not react to these changes in time. In case of a channel degradation, the sender pushes too many packets into the network without being aware of the current (degraded) network conditions. In turn, this leads to the circumstance of filling up network buffers and hence results in increased packet delays. In the literature, this effect is often referred to as self-inflicted queuing delays or bufferbloats. During the past years, congestion control was an extensively studied topic and a variety of TCP variants have been proposed.

For example, TCP Cubic [HRX08] is one of the most recent and widely deployed congestion control protocol today. Most Linux distributions leverage TCP Cubic and use it as its default transport protocol for Web browsing and file transfers. However, TCP Cubic and many other variants of TCP highly (and only) depend on packet losses to detect congestion or changes in the network conditions. That is, TCP necessarily needs to fill up the network buffers until a packet loss occurs in order to adapt its sending rate. In addition, most applications are not using TCP in its traditional sense. That is, current Web browser open multiple TCP connections to load a single web page and hence also overwhelm the network buffer. Due to this circumstance, network operators often have to increase the buffer sizes of their network components to suit the high bandwidth demand of TCP [JCD⁺13, JPD03].

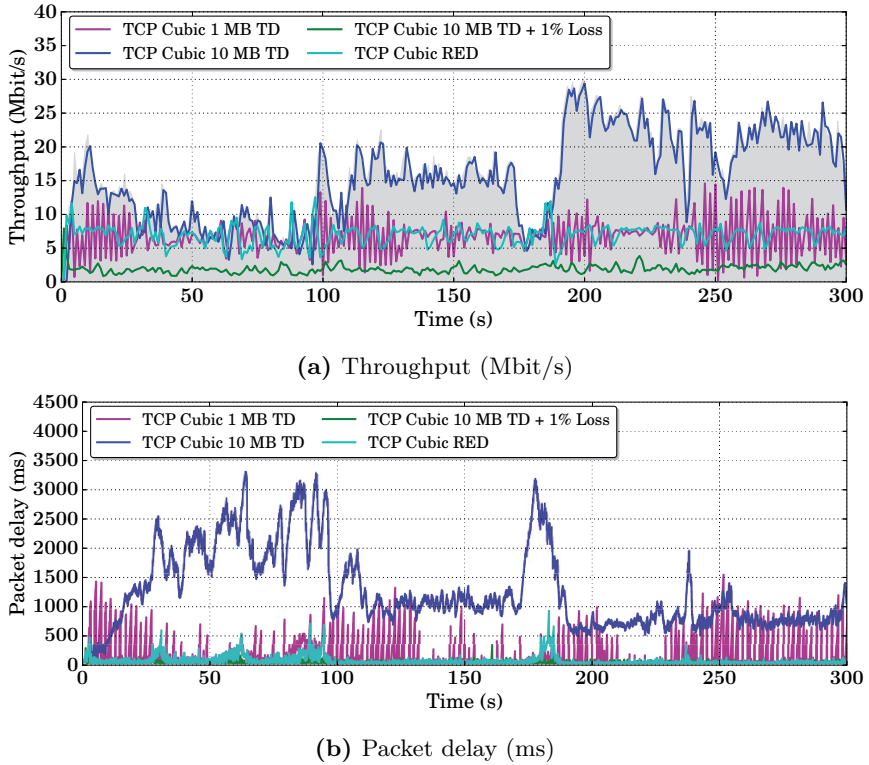


Figure 2.1: Throughput and packet delay over time of TCP Cubic under different settings of queuing disciplines

Figure 2.1 depicts the throughput and packet delay over time of TCP Cubic with different buffer settings on rapidly changing channel conditions. The channel trace that was used here was recorded using a full buffer User Datagram Protocol (UDP) transmission that completely saturated the link. Further, the channel trace was recorded while driving on a highway and replayed in a network emulator to emulate a rapidly changing network of a real cellular network. Hence, the gray shaded area in Figure 2.1a represents the maximum available bandwidth of the network. In order to highlight the performance of TCP Cubic under different settings, the buffer at the intermediate router in the network emulator was set to 1 and 10 MByte with a tail drop queuing discipline (magenta and blue) including one scenario

with an additional 1% packet loss rate (green). Further, another scenario uses a Random Early Detection (RED) queuing discipline to show a more widely used queuing discipline (cyan). RED is an active queue management mechanism that drops packets based on a statistical probability before the buffer is completely filled. That is, the more packets are in the queue, the higher the probability of dropping incoming packets. The parameters for the RED queuing discipline are a packet drop probability of 0.1% and a minimum and maximum average queue length of 3 Mbit and 9 Mbit, respectively. All four experiments were run for 300 s and with the default parameters of TCP Cubic under a current Linux distribution.

It can be seen from the figure that TCP Cubic with a buffer size of 1 MByte is not able to fully exploit the available channel capacity and misses most of the peaks in the channel trace and shows packet delays of up to 1500 ms. As mentioned above, the literature often argues that TCP requires large buffers in the network to show reasonable TCP performance. Thus, the buffer size is increased to 10 MByte and the experiment is repeated under this setting. In this scenario, TCP Cubic shows significantly higher throughput and is able to track precisely all peaks of the channel trace. However, this comes at the expense of extraordinary high packet delays of up to several seconds. That is, the packet delay under these settings are not suitable for real-time applications that require small packet delays. In the next scenario, a 1% packet loss rate¹ is introduced (green) and causes TCP Cubic to achieve only a fraction of the available throughput – despite the large buffer size of 10 MByte. This indicates that packet loss significantly degrades the throughput performance of TCP Cubic. Further, another scenario (cyan) uses a RED queuing discipline that is widely used in today's network routers. Also with this queuing discipline, TCP Cubic only achieves similar throughput as in the tail drop buffer discipline with 1 MByte buffer size but with lower packet delays. However, TCP Cubic is not able to fully exploit the available bandwidth and performs inefficiently.

In conclusion it can be said that none of the above scenarios show a reasonable performance of TCP Cubic on rapidly changing networks. That is, TCP Cubic and also other loss-based variants of TCP are not efficient on these kind of networks and perform inefficiently especially when packet losses are introduced. Even in scenarios where the buffer size of the routers is increased to allow larger buffering of packets, the packet delays increase either significantly or the throughput is degraded.

¹According to [CNGT12], packet loss rates in 3G and 4G cellular networks are below 1%. However, this scenario represents slightly worse channel conditions to demonstrate the effect of packet loss.

Future Mobile Transport Protocols
Adaptive Congestion Control for Unpredictable Cellular
Networks

Pötsch, Th.

2016, XXIX, 195 p. 64 illus., 19 illus. in color., Softcover

ISBN: 978-3-658-14814-0