Poster: Experimental Evaluation of TCP Congestion Control over 60GHz WLAN

Xiao Zhu
University of Michigan

Yuqi Jin
University of Michigan

Feng Qian
University of Minnesota

Z. Morley Mao
University of Michigan

1 INTRODUCTION

The multi-Gbps bandwidth offered by 60GHz millimeter-wave (mmWave) links has enabled a wide range of applications such as untethered virtual reality, fast file synchronization, and uncompressed video streaming. To provide these applications with high network utilization while preventing overload, TCP congestion control (CC) determines the data sending rate to achieve high throughput and/or low delay.

CC over highly fluctuating 60GHz networks is extremely challenging – the high directionality and small wavelength of mmWave make it vulnerable to mobility and blockage. Despite numerous studies on CC in WiFi/LTE, CC in 60GHz remains underexplored. To the best of our knowledge, the only related work on CC in mmWave is [1]. However, it's simulation-based and focuses on 28GHz mmWave cellular. Also, it only looks at loss-based CC algorithms and limited scenarios. We close this gap with a first experimental study that comprehensively evaluates the CC performance over commercial 60GHz WLAN using COTS devices. Our study aims to characterize the behavior and performance of different CC algorithms over 60GHz in a wide range of real-world mobility and blockage scenarios. We believe this is a critical step towards designing mmWave-friendly CC algorithms.

Experimental Methodology. Our testbed consists of an Acer Travelmate laptop client, a Dell PowerEdge server, as well as a Netgear Nighthawk router. The router and laptop are connected with 802.11ad 60GHz. The server and router are connected using two configurations: (1) a 10GbE SFP+ cable, (2) a 1Gbps cable. The two different configurations help us study the end-to-end (E2E) performance with different backbone capacity. Our machines run Linux kernel 4.10 that implements a wide range of loss-based CC (*e.g.,* Cubic, Reno), delay-based CC (*e.g.,* Vegas), and the latest BBR. We run `iperf3` on the server and client to generate three traffic patterns: (1) upload, (2) download, and (3) concurrent upload and download. We use `tc netem` to add delays to the E2E path to emulate different backbone RTT. To prevent the side effect of `netem` with 10GbE, we run `tc` on the laptop for the `egress` to delay the outgoing data (ACK) packets for upload (download), this "asymmetry" does not affect the results as none of our studied CC uses one-way delay measurements. We do not explicitly control network bandwidth with `tc`. Instead, we set up different distance, blockage, and mobility patterns for the 60GHz link to examine its behavior and impact on the transport layer performance. We build *mmNetAnalyer*, a cross-layer tool to analyze the CC performance over 60GHz mmWave networks. It measures the throughput, delay, and packet loss and collects protocol information from multiple layers on different devices. Specifically, it leverages `tcpprobe` to obtain TCP state information (*e.g.,* `cwnd`, `rwnd`) from Linux kernel to track the CC behavior, and queries the physical layer 802.11ad information (*e.g.,* Tx/Rx MCS, SQI) using the Linux `iw` command-line tool.

Preliminary Results. We present preliminary results for download with configuration (1). Even under stationary LoS scenarios when the base RTT is low (1ms), Cubic does not always fully utilize the 60GHz link, *e.g.,* we observe throughput drop of 40% for up to 2s before recovery. Under this setting, Vegas achieves even worse network utilization (5x lower overall throughput than Cubic) although its E2E latency is reduced by 8x. When the base RTT is increased to 50ms, the default TCP send and receive buffer setting on the 60GHz laptop significantly limits the achievable throughput. After we increased the buffer size to be enough, Cubic achieves fair throughput under LoS, but it reacts very slowly after temporary blockage: it can take more than 20s for its `cwnd` to grow to re-saturate the 60GHz link. For long outage that lasts several seconds, Cubic's `cwnd` stays very low for a long time even if MCS quickly recovers after the outage. For BBR, we observe a much faster reaction to different blockage scenarios. However, during its probeRTT phase, the throughput is largely degraded, this is especially harmful to bandwidth-hungry interactive applications that continuously require high throughput and low delay. We are currently investigating their causes.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

S3'19, October 21, 2019, Los Cabos, Mexico

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6929-9/19/10.
https://doi.org/10.1145/3349621.3355733

REFERENCES

[1] Zhang et al. 2016. Transport Layer Performance in 5G mmWave Cellular. In *INFOCOM WKSHPS*. IEEE.