# Congestion Control by Leveraging LTE PHY-Layer Information

Can Carlak (ccarlak@umich.edu), Zhensheng Jiang (jzsheng@umich.edu)

EECS 589 Advanced Networks 2019


## Abstract

Last hop cellular link is often the bottleneck in end-to-end communication, commonly known as the last-mile problem. Mainly because of the lack of correlation between wireless link layer and transport layer, TCP often under utilize the available bandwidth. In fact, TCP receive window throttles approximately the half of all downlink TCP flows. Having access to wireless physical layer channel quality information can potentially improve congestion control in the end hosts. The main objective of this study is to increase the bandwidth utilization of transport layer protocols, and to enhance the throughput for application layers.


## Background

### Wireless link

- Frequent change in achievable link-layer data rate
  - Path-loss
  - Slow-fading
- Large variations in packet delay
  - Automatic Repeat Request (ARQ) protocol
  - Error correction

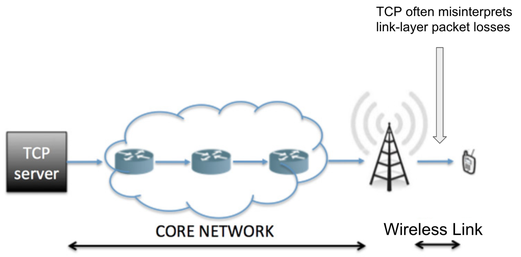


Figure 1: Cellular network.

## Methodology

### Phase I

- Collect LTE parameters at UE
  - RSRP
  - RSRQ
- Measure throughput
- Record RSRP/RSRQ -> throughput relations

### Phase II

- UE sends its current RSRP/RSRQ values via HTTP headers
- Server does a table look up to set initial congestion window size dynamically for each connection.
- UE measures throughput

## Challenges and Future Work

- Finding useful physical layer information that are closely related with the performance at network/transport/application layers. Performance measures should be effectively correlated across different network protocol layers.
- Adaptation of the congestion control to leverage physical/link layer visibility of network is hard. Existing congestion control algorithm should be correctly updated with the additional cross-layer information.
- User equipment measurements can only have a local view of the wireless channel. Instead, cross-layer correlations should be initiated at base stations (eNB) which possess a broader scope for the link. This requires to conduct experiments on a test-bed with an eNB that we can control.
- Make use of learning algorithms that uses PHY-layer parameters as features.

## Results

Figure 2 and Figure 3 show the experiment results of HTTP requests throughput and time-per-request under different $RSRQ$ values. Throughput and time-per-request values are average of 50 concurrent HTTP requests. In our experiment settings, when $RSRQ = -9$, the dynamic initial CWND is set to 10, which is also the default initial CWND in Linux. The points of default and dynamic initial CWND in both figures where $RSRQ = -9$ are very close to each other as expected, because they share the same initial CWND and should have similar traffic patterns.

Although we are not able to collect data for default initial CWND when the channel quality is bad($RSRQ \leqslant -10$), we notice that the performance of dynamic initial CWND when the channel quality is bad is even marginally better than that of default initial CWND when the channel quality is good($RSRQ > -9$). This is because the last-hop is the bottleneck when the channel quality is bad and the default initial CWND is too large for this case. We argue that when the CWND is set too large in the condition of bad channel quality, it may result in a bufferbloat in the base station, which can lead to high latency and even packet loss. We leave it for future work.

When the channel quality is good, we dynamically set the initial CWND to a greater value, which result in better throughput and time-per-request as expected.

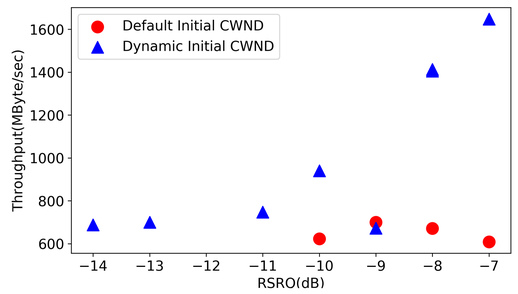


Figure 2: Throughput under different RSRQ.

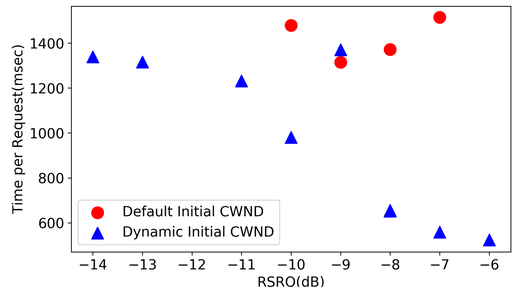


Figure 3: Time-per-request under different RSRQ.

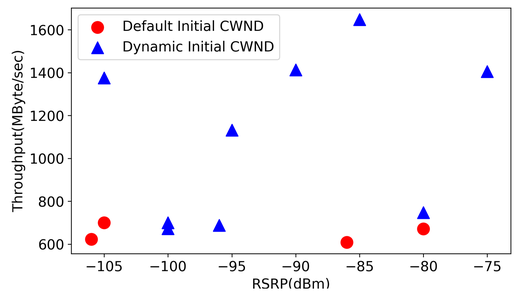


Figure 4: Throughput under different RSRP.