Chrome Case 1 Study

Sihan Min (Mint)

Outline

- Motivation
- Experiment
- RRC Connection Latency Measurement
- Three Types of Web Browsing
- RRC Connection Request Summary
- RRC Connection Release Pattern
- Conclusion

Motivation

Setting up LTE RRC Connection with BS often has a latency. This latency may slow down web page loading in Chrome.

We want to know how long is the RRC connection latency, how often would the latency occur at user clicks, and how can we prevent the latency.

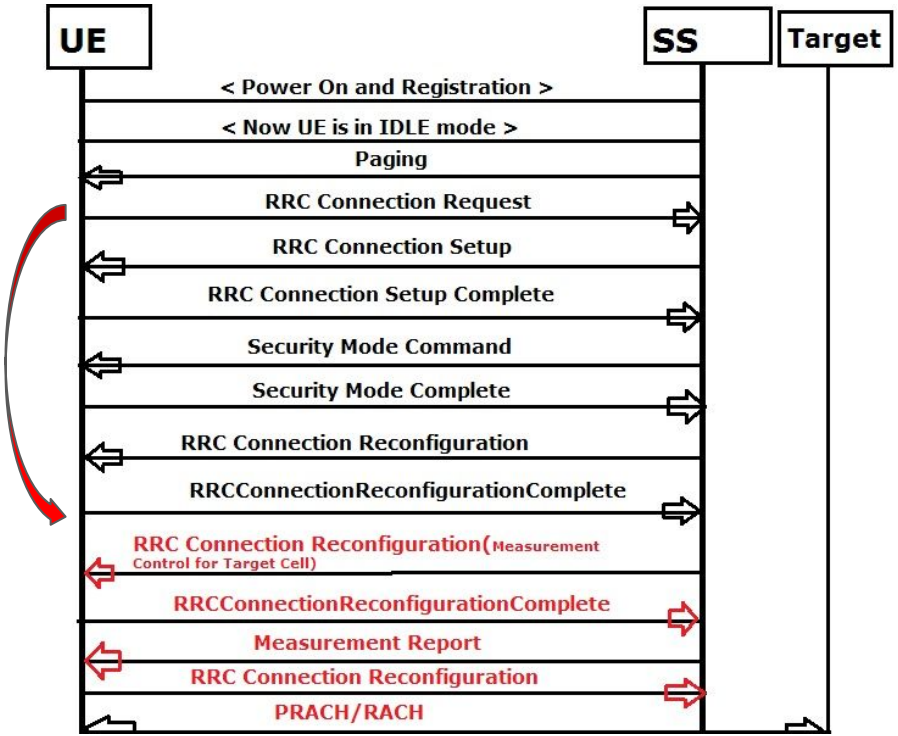
This case study focuses on the above three questions and analyzes how much users can benefit from solving the latency problem.

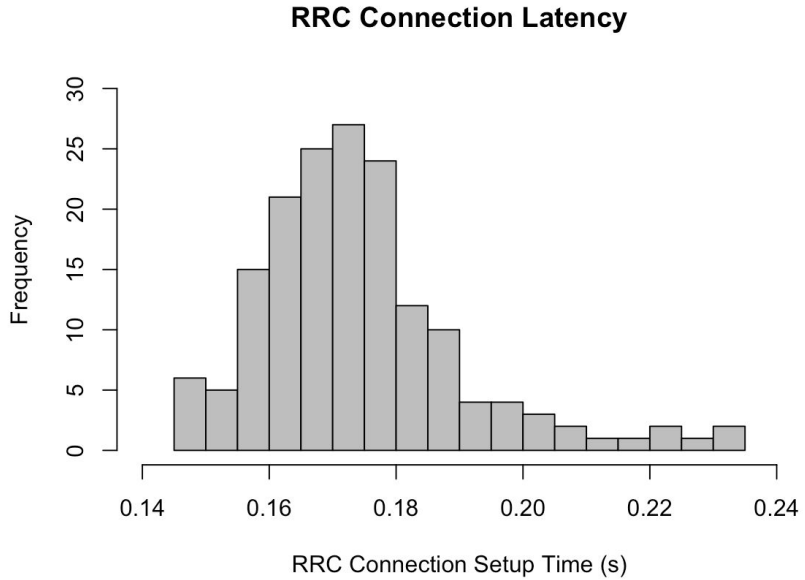
Experiment

- Settings:
  - Google Pixel XL with Mobile Insight 3.3 installed
  - Simulation of daily web browsing in Chrome
  - Three times of experiment, each more than 20 minutes
- Data Measurements:
  - Backstage MI recording LTE Control / Data plane packages
  - Total 105 minutes of records
  - Total 57 times of web page request (clicks) counted
- Data Analysis:
  - LTE_RRC_OTA_PACKET : extracting timestamp of rrcConnectionRequest, rrcConnectionReconfigurationComplete, and rrcConnectionRelease
  - LTE_MAC_DL_Transport_Block : extracting timestamp and total downlink bytes of each packet, aggregated by second

RRC Connection Latency Measurement

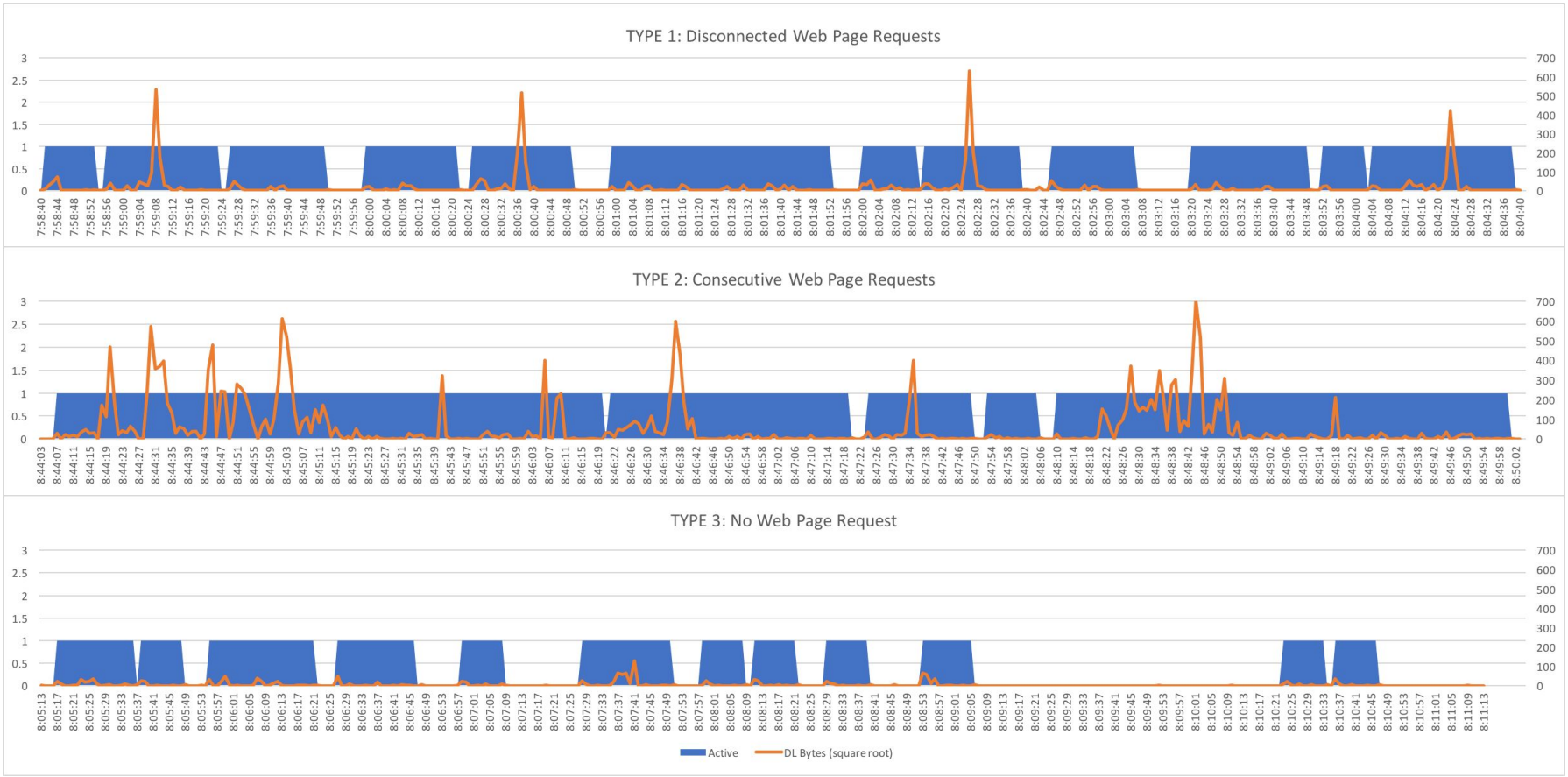
RRC Connection Request TO
the 1st RRC Connection Reconfiguration Complete





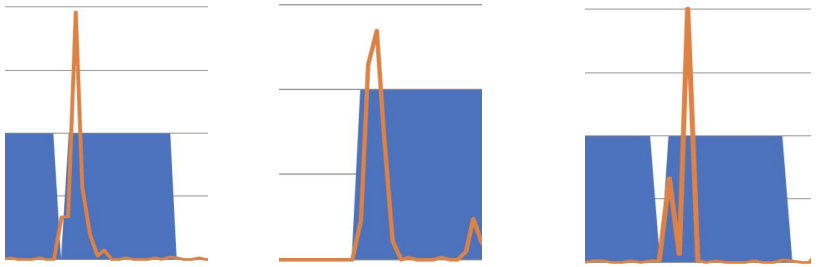
1st Qu.	Mean	3rd Qu.
164 ms	174.6 ms	181 ms

Three Types of Web Browsing

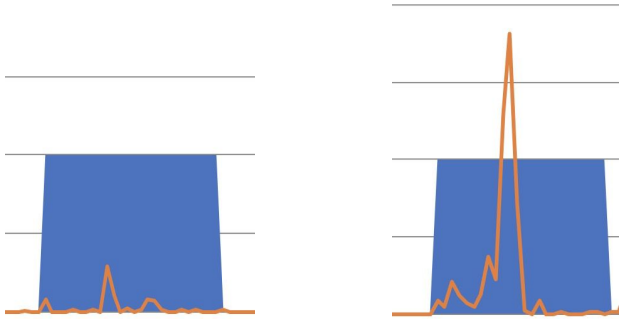


RRC Connection Request Summary

RRC Connection triggered by web page requests:



RRC Connection triggered by other network IOs:



Total RRC Connection Requests: 168 times

Total Web Page Requests (Clicks): 57 times

Among all RRC Connection Requests, 3 times are likely triggered by web page requests.

Clicks Triggered RRC Request / Total Clicks = 5.26%

RRC Connection Request Summary

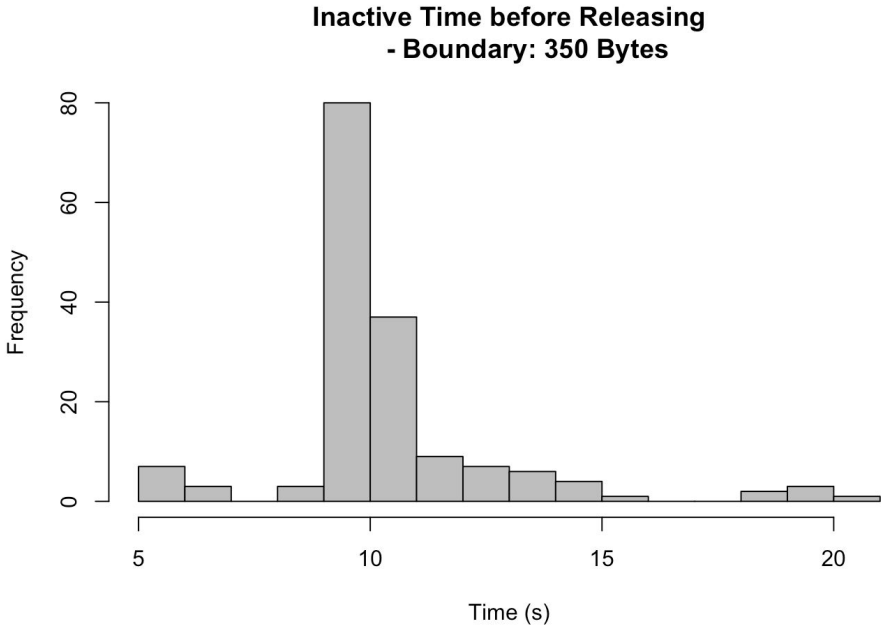
- Most of the RRC connections are not triggered by user clicks.
  - Backstage applications and OS updates
  - Chrome searching autofills
- When the user starts a new search, small network IO is likely to setup the connection before he finishes typing keywords or URL.
- Connections are not likely to break at consecutive clicks.
- When no user clicks, small packet requests from the existing web page and the backstage apps also trigger or maintain RRC Active.
- **Question: After how long of UE inactive would BS release RRC connection?**

Active to Idle Measurement
- RRC Connection Release Pattern

Inactive time: the time with no throughput exceeding boundary, counted from connection release backward.

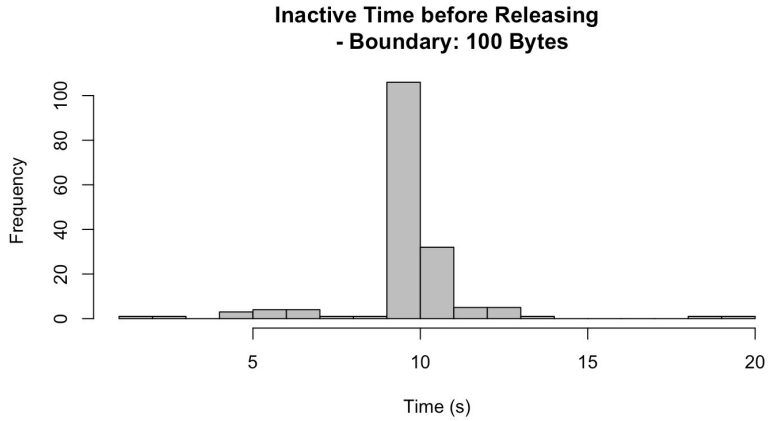
Throughput boundary per second: 350 Bytes (the smallest throughput measured at activation)

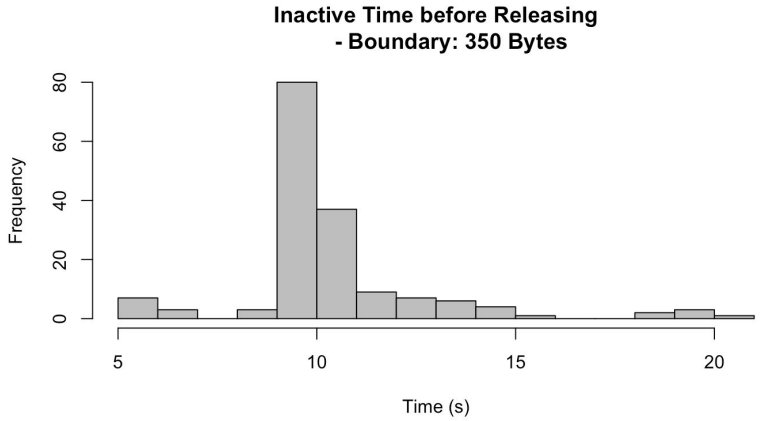
Result: mostly 9 or 10 seconds

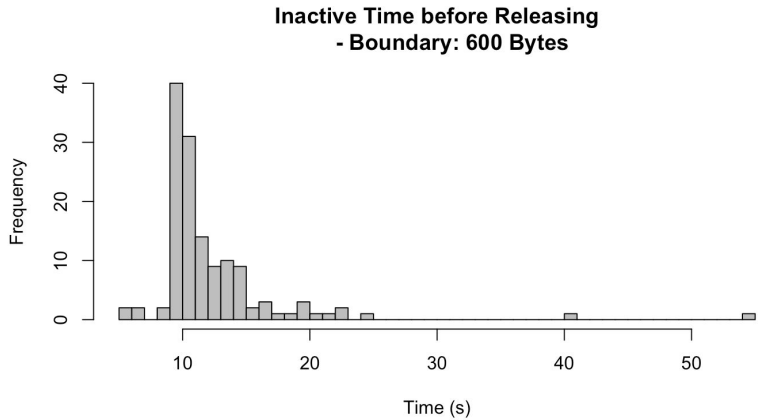


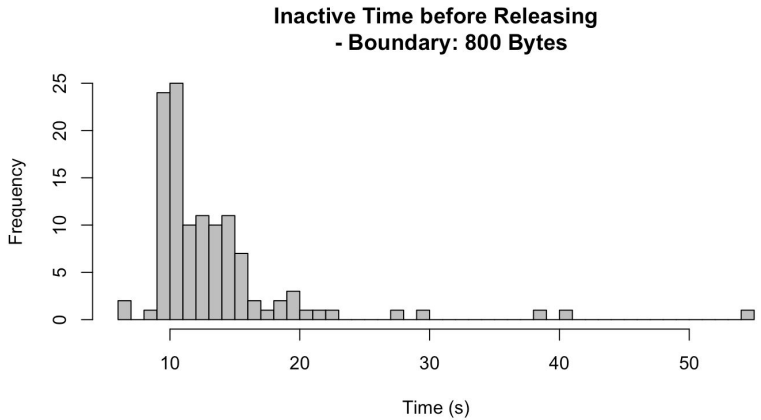
For every connection, the total downlink throughput in the last 9 seconds is less than 1500 Bytes.

RRC Connection Release Pattern









Conclusion

- Experiment results:
  - RRC Connection Latency: 150 - 200 ms, on average 174.6 ms
  - Clicks Triggered RRC Request / Total Clicks = 5.26%
  - Inactive Time before Release: 9 - 14 s, mostly 9 or 10 seconds
- Possible ways to save RRC connection time:
  - Send small packet request to ping website every 8 seconds
  - Trigger connection setup once the user puts Chrome in frontstage
- However, since for most of the time connection already exists when the user clicks new web page, and the connection setup time is at most 0.2s, the performance improvement from C1 would not be noticeable to users.