Yingcong Li, S1616245
Demystifying and Puncturing the Inflated Delay in Smartphone-based WiFi Network Measurement

---

Measuring wifi network performance using apps gained popularity these years. But these apps usually focus on user-level performance, they cannot be used to test the real network performance in network-level. Also, previous research shows that the network Round-trip Time (RTT) measured by apps are all inflated delay overhead in different degrees. Although parts of the delay inflation can be mitigated, but there is still a negligible delay overhead which is not understood. Li et al. performed a test to find the last missing piece for this delay.

Same concept--*delay overhead*--used to measure the delay. Specifically, *User-kernel overhead (*$\Delta d_{u\text{-}k}$*)* indicating the delay from application layer to kernel, similar way to *Kernel-driver overhead (*$\Delta d_{k\text{-}v}$*)* and *Driver-phy overhead* $\Delta(d_{v\text{-}n})$*.* In particular, the overhead incurred by DVM (i.e., $\Delta d_{u\text{-}k}$) can be mitigated by bypassing the DVM. Besides, the PSM can inflate the nRTT significantly since AP will not send packet to smartphone immediately.

Testbed consists of a measurement server, a wireless AP, a load generator and a load server to generate cross traffic, including three sniffer computers to compute the correct timestamps for hardware layer. Both Google Nexus 4 and 5 are tested under the condition of 30ms nRTT and 50ms nRTT with two packet sending intervals, a small interval of 10ms and larger default of 1s.

Results show that both two phones report a small $\Delta d_{k\text{-}n}$ (*Kernel-phy overhead*), which are smaller than 4 ms when the sending interval is small. While Nexus 5's $\Delta d_{k\text{-}n}$ (~18ms) is nearly two times more than Nexus 4 when sending interval is 1s. On the other hand, $\Delta d_{u\text{-}k}$ is close to 0 in both two phones. Some delay underestimates in this part are due to the low resolution of *ping* results.

Looking into the source code of the WNIC driver in Nexus 5, the cause of the delay can be found in the SDIO (Security Digital Input Output) bus.The driver puts the SDIO bus to sleeping mode when the data rate is not high, so when a packet is ready to be sent or received, an additional time is needed to this driver to bring the SDIO bus to wake up. When the sleep mode is disabled, delay overheads drop closely to 1ms in both sending intervals. In a similar way, the PSM (Power Saving Mode, PSM) also could inflate the nRTT, for example, if d_n (the actual nRTT) is longer than the PSM timeout, a smartphone will turn into PSM mode before receiving the reply message, this will introduce additional delay. So both SDIO bus sleep and the PSM will inflate the delay. Particularly, the SDIO bus sleep affects the delay internally, while the PSM externally, between the smartphone and the AP.

Finally, they implemented their approach based on AcuteMon, using background traffic thread (BT) to keep the SDIO bus keep active during the measurement time and measurement thread (MT) will send measurement probes to measure the nRTT between phone and server. MT is a pre-compiled C program to mitigate the delay caused by DVM. As a result, the overall median delay overheads are controlled within 3ms using their AcuteMon. This is the most accurate delay that can achieved so far on Android platform without hacking the system. In the future, more works can be done by conducting experiments on other phones or operating system such as iOS and Windows Phone. This accuracy measurement also can be extended to cellular network. What's more, it is worth exploring the reason for the last 3 ms delay overheads, or performing a calibration to get the true value.