Who slowed down my app? A Measurement Study of Response Times of Smartphone Apps in India

Abhivandan Pandey
2017007

Prince Sachdeva
2017080

BTP report submitted in partial fulfillment of the requirements
for the Degree of B.Tech. in Computer Science & Engineering

on June 2, 2021

BTP Track: Research

BTP Advisors
Dr. Mukulika Maity
Dr. Arani Bhattacharya

Indraprastha Institute of Information Technology
New Delhi

Student's Declaration

I hereby declare that the work presented in the report entitled **Who slowed down my app? A Measurement Study of Response Times of Smartphone Apps in India** submitted by us for the partial fulfillment of the requirements for the degree of *Bachelor of Technology* in *Computer Science & Engineering* at Indraprastha Institute of Information Technology, Delhi, is an authentic record of our work carried out under guidance of **Dr. Mukulika Maity** and **Dr. Arani Bhattacharya**. Due acknowledgements have been given in the report to all material used. This work has not been submitted anywhere else for the reward of any other degree.

.... **Place & Date: ....**
Prince Sachdeva

.... **Place & Date: ....**
Abhivandan Pandey

Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

.... **Place & Date: ....**
Dr. Mukulika Maity

.... **Place & Date: ....**
Dr. Arani Bhattcharya

Abstract

India has seen a rapid rise in the adoption of smartphones, with a majority of Indians accessing the Internet through smartphones. Smartphones have become one of the most important mediums of delivering essential services such as news, entertainment, and even payment services in India. However, it is not clearly understood how the quality of experience of individual users varies, especially considering the huge diversity of the smartphone models, network conditions, and regions from which they are used. We quantify the *Quality of Experience* of using the android apps using response time i.e., the time needed to reflect UI changes corresponding to a user's action. This can, for example, the time needed to reflect an item has been added to cart for *Add product to cart* action on Amazon app. In this work, we design a tool called EvalApp which uses automation to record the response times of a total of 30 actions for 12 apps popular in India. We then crowdsource this desktop app to a total of 41 users working from home from across north and central India and perform a causal analysis of the factors that affect the response times of actions. We find that in most cases, the response time is very strongly correlated with the ping round-trip time to the nearest Google server, thus indicating that the network is the most common bottleneck. We further identify that the distance from a major city is the other factor that strongly affects latency. This is counter-intuitive in view of recent works on the mobile web in the US, which identified smartphone hardware as the primary bottleneck. We also do a few controlled experiments to observe that having a higher app version does not always ensure better response times. Our observations are likely to help designers of apps to better identify techniques of improving the Quality of Experience for users.

Keywords: Mobile Application Testing, Internet measurement, causality analysis

Acknowledgments

We would like to thank our advisers Dr. Mukulika Maity and Dr. Arani Bhattacharya for guiding us, and providing us with the insights and expertise and giving us the opportunity to pursue this project.

Contents

Chapter 1

Introduction

1.1 Motivation

Over the last five years, there has been an increasing adoption of smartphones in India. For example, the number of smartphones sold in India increased by over 2 times from 2014 to 2019, with over 150 million smartphones being sold in 2019 [16]. Smartphone services are increasingly being used for payment, messaging, social networking, reading news as well as entertainment like watching videos and playing games. The number of apps downloaded in India is the highest in the world [2]. Thus, it is vital for such app developers to understand the quality of experience (QoE) observed by users in India.

The quality of experience of smartphone apps is primarily governed by the response time observed by the users on interaction [6, 7]. For example, a user watching videos over YouTube would have a worse experience if the video takes longer to load. A key concern in countries like India is that there is a wide divergence in the response times observed by different users. This concern about response time is especially important in the context of India because the vast majority of the population of India accesses the Internet through smartphones [17]. Since this divergence in latency can lead to poorer access to essential services among users, it is essential to quantify the amount of divergence seen by actual users while interacting with the apps. A second question that is currently not understood is the reason behind the divergence in response times. The divergence in response times can be due to multiple reasons, such as the type of smartphone used, location of the users, the version of the apps, nature of backhaul network as well as the type of network used. However, which of these causes dominates is currently not understood.

In this work, we conduct a large-scale crowdsourced measurement of the response time for different user actions on the 12 most common apps. For each of these apps, we choose the most common types of user actions and use an automation tool Appium [10] to activate specific actions automatically. We then wait until the UI element shows that the action has been completed, and compute the time elapsed between the change in the UI and the time of activation. Using this technique, we compute the response time of each application. We then crowdsource this automated tool to over 40 users after suitable Institutional Review Board (IRB) approval, and request them to run the automated tool by connecting it to a personal computer. Apart from logging the response times observed, we also send ping requests to a few servers from the smartphone to measure the network condition. We collect over 16000 such response time values, from across north and central India. We also request the users to specify their phone model and Android version using a separate form.

We then perform a causal analysis to identify the underlying reasons behind the high response times. We compute the correlations between the other parameters collected, such as the Android version, the type of network used, the location of the user, and the smartphone's hardware characteristics, with the response times. We find that in general, the ping round trip times (RTT's) to the nearest Google server are much more strongly correlated with the response times than any other factors with a correlation coefficient of 0.7 across all users, thereby indicating that it is the network that is primarily responsible for most users. Furthermore, since ping RTT's themselves depend on multiple other parameters, we then remove it as a feature to find out the correlations with the other metrics. We find that the geodesic distance from the city with a data center is only moderately correlated with response times. None of the other factors, including time of day, better quality of smartphone or type of network used (WiFi/cellular) has a significant affect on the response times. This further implies that buying better smartphones can lead to very limited improvement in response times for most users. Additionally, the app developers should consider reducing data consumption to improve response times.
We summarize our contributions as follows:

- We create an automated tool EvalApp to compute the response times of 12 common Android applications for a total of 30 common interactions (§2.1.1).
- We crowdsource EvalApp to over 40 users spread across north and central India. We further collect a total of over 16000 data points with each user performing an IRB-approved experiment for over 5 days (§2.1.2).
- We compare the response times observed (§3.1.2) and perform a causal analysis to identify that the quality of the network is the main bottleneck for most users, as opposed to the quality of smartphones. We find that the response time is strongly correlated with the ping RTT times to a few standard servers. Furthermore, we find that a campus network with low ping RTT values can reduce the response times of some actions by over 10 times. This is contrary to the results seen in recent experiments on the mobile web in the US, where the smartphone hardware is identified as the primary bottleneck [21, 22]. Interestingly, distance from a major city was only moderately correlated with the response times for most apps[1] (§3.1.2).
- We perform a controlled experiment to check how app versions affect the response times. We show that in general, having a higher app version does not always ensure better response times (§3.1.3).

[1]We identify cities with data centers [1] as a "major city" in this work.

Chapter 2

Design & Implementation

2.1 Design & Implementation

We first discuss the design of EvalApp. We then discuss our technique of crowdsourcing and the collected dataset.

2.1.1 Design of EvalApp

The goal of EvalApp is to create a user-friendly framework that can be utilized by non-technical users to perform tests on Android apps to measure the response time. It uses a GUI-based desktop application written in Java programming language, with the GUI written using JavaFX library [11]. The test code for each app is written using Appium library [10], which is an open-source automation tool to test apps.

The design of EvalApp is shown in Figure 2.1. Appium starts a server on a personal computer (PC) and requires the smartphone to be connected to it via a USB port. The automation scripts for each individual app first need to be installed on the PC. When the user connects their smartphone to the PC using a USB cable and launches the tests from EvalApp, these automation scripts activate some specific user actions on the apps. This script is designed in such a way that it waits for a specific change on the User Interface (UI) to occur, and then records the time elapsed between the request sent and the change on the UI. The exact set of UI actions used for each app is shown in Table 2.1. The data collected is sent to the Appium server, which in turn forwards it to a remote MongoDB server for storage. Simultaneously, we collect ping data from the smartphone running *adb ping* commands in the connected mobile device to `google.com`, `mobikwik.com` and `amazon.com` once every five seconds throughout the experiment to collect at least 10 ping RTT values per server.

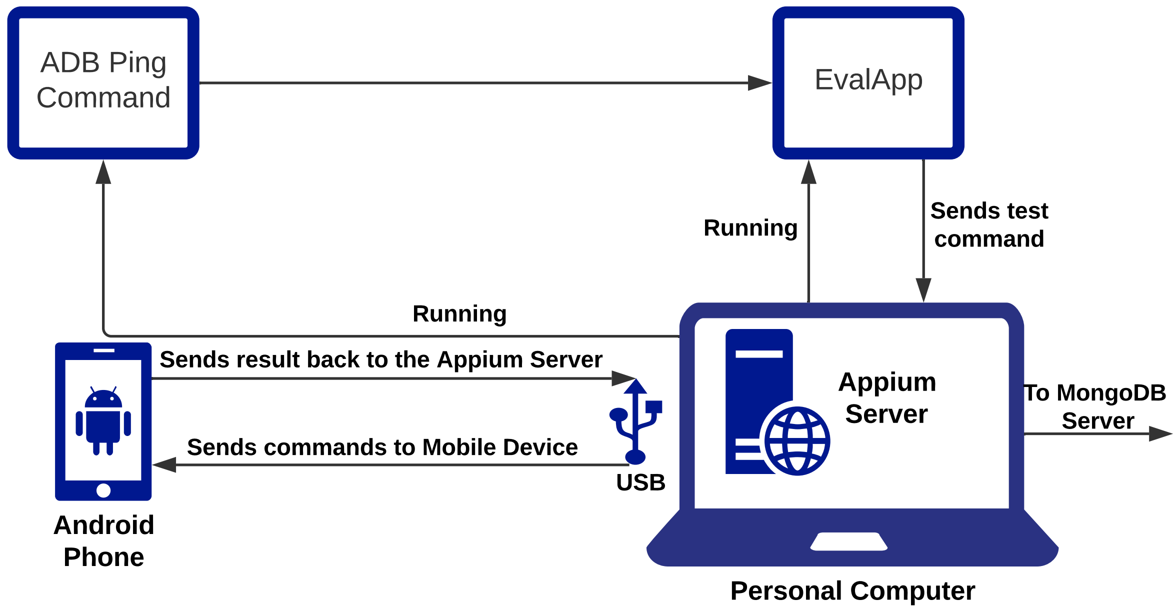


Figure 2.1: Design Flow Chart of EvalApp

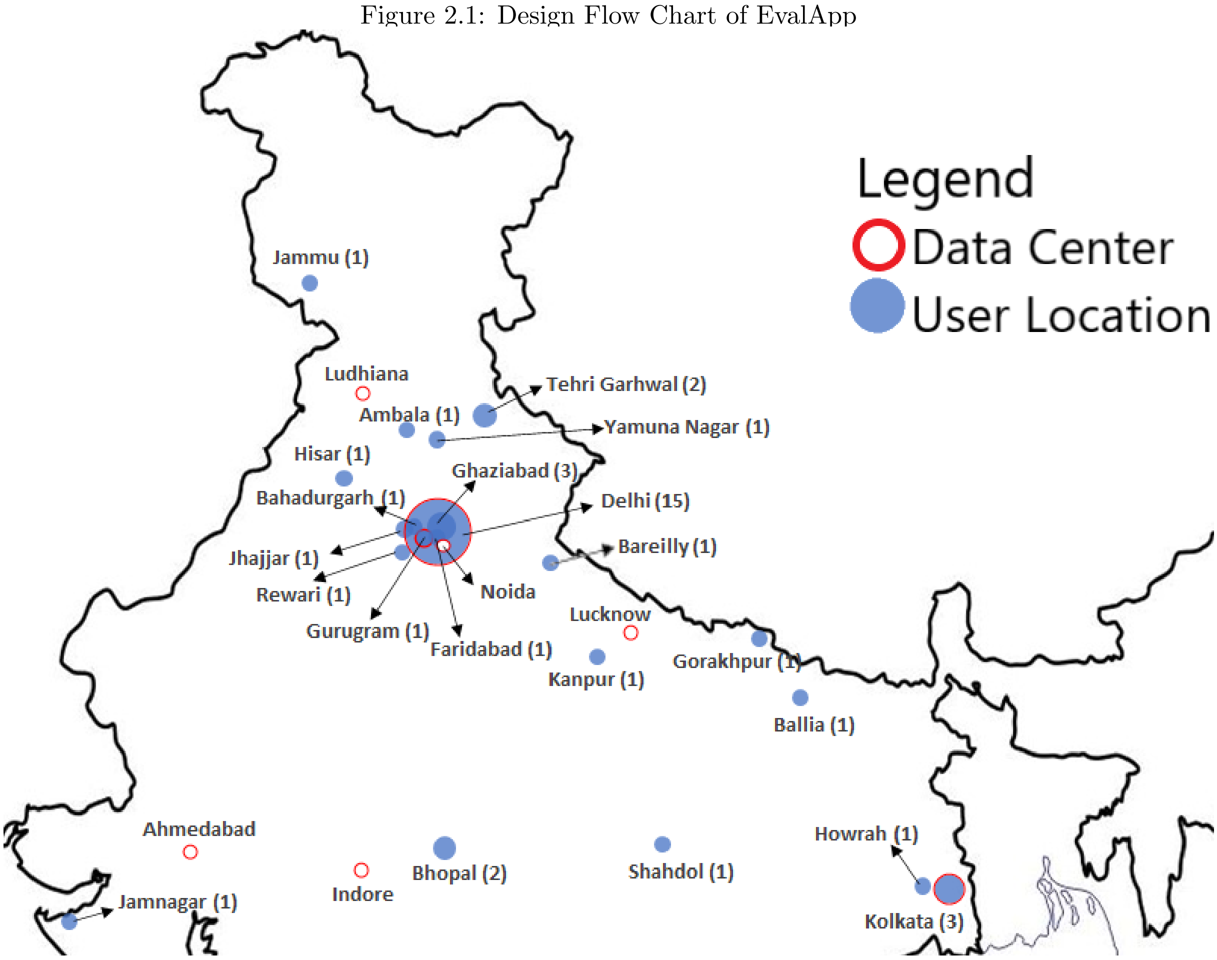


Figure 2.2: Locations of the volunteers who run EvalApp and cities with data centers (major cities). The number in parenthesis denotes the number of users in each location.

Table 2.1: A list of 12 apps and 30 actions for which we have measured the response times.

	App	App Type	Action	UI Response
1	YouTube	Streaming	Search Channel	Search results displayed
			Open Channel Page	Channel logo shown
			Search Video	Search results displayed
			Play Video	Video player box appeared
2	Hotstar	Streaming	Search video	Search results displayed
			Play video page	Video title displayed
			Open trending page	Video player box appeared
3	LinkedIn	Social	View profile	Profile picture shown
			Check connections	Connections page appeared
			Search Person	Search result displayed
			Open person profile page	Profile picture shown
4	Facebook	Social	Post a message	Message box appeared
			Search Person	Search results displayed
			Open person profile page	Profile picture shown
5	Google News	News	Search News	News page title displayed
6	DailyHunt	News	Search News	News box appeared
7	Amazon	Shopping	Search a product	Search results displayed
			Open product page	Product rating stars shown
			Add product to cart	Cart items increased
			Go to cart	Cart items shown
			Remove product from cart	Product removed shown
8	Flipkart	Shopping	Search a product	Search results displayed
			Open product page	Product rating stars shown
			Add product to cart	Cart items increased
			Go to cart	Cart items shown
			Remove product from cart	cart items decreased
9	WhatsApp	Messaging	Send message	Ticks shown
10	Telegram	Messaging	Send message	Ticks shown
11	Paytm	Payment	Send ₹1 payment	Payment success shown
12	Google Maps	Navigation	Search a location	Street Thumbnail displayed

2.1.2 Our Dataset

We requested a total of around 41 students of our department to serve as volunteers. The volunteers had to install and login to these 12 apps. Due to the COVID pandemic, most of the students attended their classes from home. Thus, the students were spread over north and central India, with a small majority of students coming from the National Capital Region (see Figure 2.2). We requested them to run the experiments twice a day for at least five days to reduce the chances of outliers affecting our conclusions. The volunteers used either WiFi, cellular network, and/or WiFi hotspot depending on their convenience. The parameter ranges seen in our experiments are shown in Table 2.2. We collected a total of over 16,000 data points from these volunteers. Collecting this dataset required approval from the Institutional Review Board (IRB), informing the volunteers about the ethical issues involved and getting their consent. We discuss in detail the ethical considerations involved in the Appendix.

Furthermore, we performed controlled lab experiments for analyzing the app response times across app version upgrades for two different android versions. For that, we took two Lenovo phones with model number *A6020a40* that has a RAM of 2GB. One of them runs *Lollipop* Android *5.1.1* and another one runs *Oreo* Android *8.0.0*. We also developed another app to measure metrics such as the bytes transmitted (Tx) and received (Rx). We analyze the response time of doing the actions across the three most recent versions of Google Maps, YouTube, Flipkart, and Telegram. We could not run this analysis for other apps as 1) the forced update for some apps did not allow us to run their older versions, and 2) authentic APKs of older versions were not available for some of the apps.

Table 2.2: Range of parameters of the experiments. Many of these parameters depend on the individual settings used by the users.

Parameter	Value/Range
Number of Users	41
Range of Android versions	7 - 11
Range of Android RAM	2 GB - 8 GB
Maximum physical distance from major city	397 km
Times of Day (Morning)	9:00 - 15:00
Times of Day (Evening)	17:00 - 23:00
Number of identical actions repeated by a single user	5-10
Ping probes sent per app experiment	10

Chapter 3

Response Times and its cause

3.1 Response Times and its Cause

3.1.1 Distribution of Response Times

We first plot the distribution of response times for each action in Figure 3.3. We first note that the response times have appreciable divergence, with the 25^{th} and 75^{th} percentiles (shown using the boxes) having a significant difference. For example, Amazon's "Add to Cart" action has a response time of $1.8s$ and $3.9s$ respectively at 25^{th} and 75^{th} percentiles. Similarly, we find that "Play Video" in Youtube, "Search Results" in Facebook and LinkedIn, and "Sent Payment" on PayTM have high divergences, with the 25^{th} percentiles and 75^{th} percentiles being equal to $2.5s$ and $3.7s$, $2.8s$ and $4s$, and $100ms$ and $700ms$ respectively. In most of these cases, the differences are higher than $1s$, and in some cases, higher than $2s$. Such perceptibly high divergence in response times is known to hurt the quality of experience [19]. We analyze the reasons behind such divergence of response times further later on in this section.

We further compare the response times across apps from Figure 3.3. We note that for most actions the two shopping apps Amazon and Flipkart have statistically very similar response times. Similarly, LinkedIn and Facebook's common action "Search Person" as well as YouTube and Hotstar's "Play Video" action also have similar response times. WhatsApp has a significantly slower response than Telegram, whereas Google News and DailyHunt have similar response times.

3.1.2 Causal Analysis

Methodology: For each user, we collect the data about their Android version, the amount of RAM on their smartphone, and their distance from a major city. We specifically consider distance from a major city because India is known to have a spatial digital divergence in network infrastructure, where the smaller towns have higher network latency than the major cities. For each experiment, we also log the type of network used (WiFi/cellular), time of day, and the

Round Trip Time (RTT) of the ping probes. These parameters act as potential causal factors and we observe their correlation with the response times.

Our dataset consists of both discrete features, such as type of network, time of day, Android version, amount of RAM available as well as continuous features, like RTT of ping probes. Thus, any correlation between the response time, a continuous metric and these discrete features is likely to be non-linear and the standard Pearson correlation metric would not give us a good understanding of the actual correlation. To find the correlation between the features and the response times, we first apply k-means clustering to classify all the response times into an optimum number of clusters. We choose the optimum by applying silhouette analysis [24] on the response times. In general, we get 2-4 clusters for all the apps. Since the ping RTT values are likely to have a high degree of outliers, we only utilize the median RTT value. We then determine which of these features best help us predict the cluster of response time on a decision tree for the actions. The decision has a median F1 score of 0.726, which indicates the accuracy of classification using our decision tree.

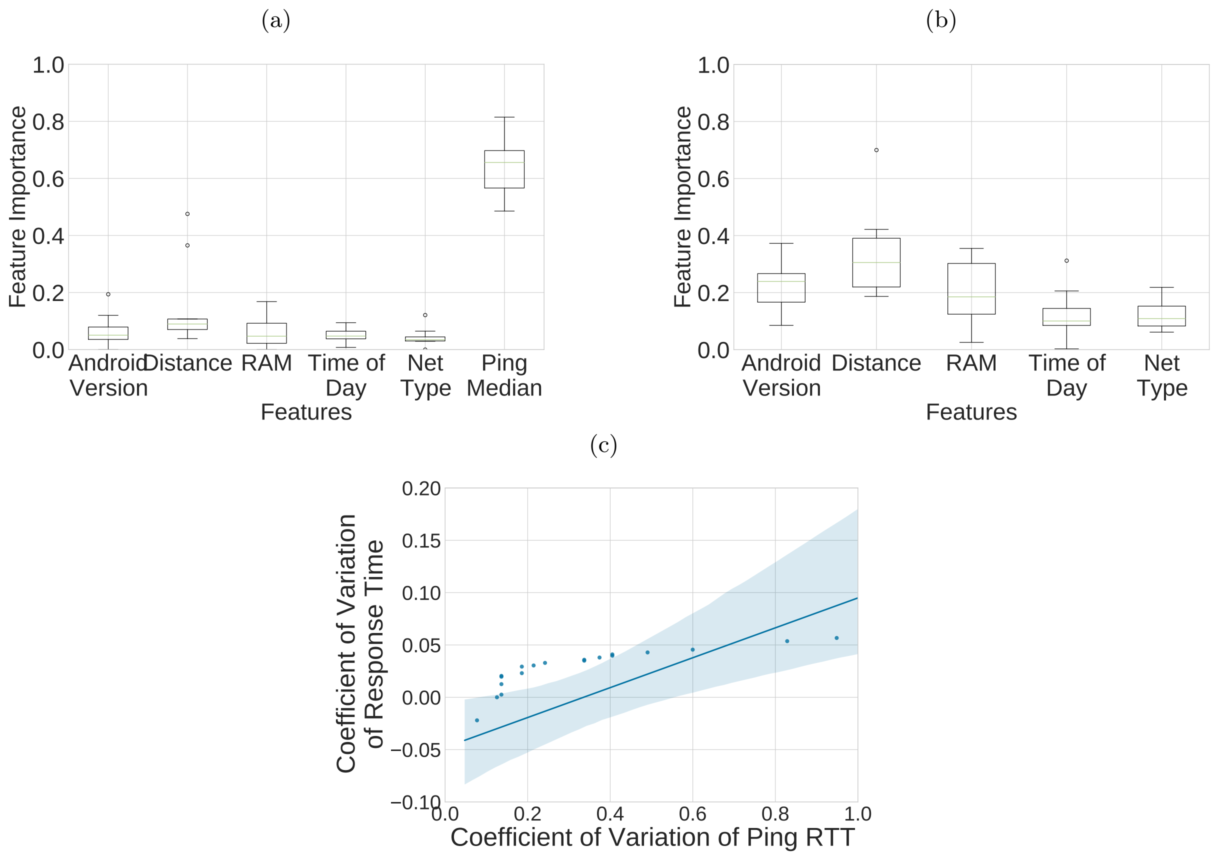
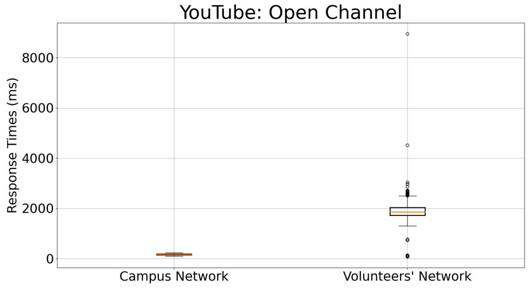


Figure 3.1: Causal analysis of latency with respect to different features. In (a), we plot the box plots of feature importance with respect to all features, whereas in (b), we plot them after excluding the ping RTT. In (c), we show the correlation between coefficient of variation (COV) of ping mean and COV of response time.

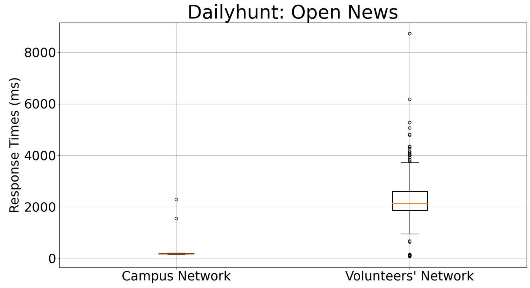
Observations: To study the performance in detail, we perform three different sets of analyses. First, we study the importance of all the features (Figure 3.1(a)). We find that the ping median (to Google server) dominates in all cases, with its importance coming between 0.57 and 0.72.

This shows that the network is the primary bottleneck for almost all users for all actions.

(a)



(b)



(c)

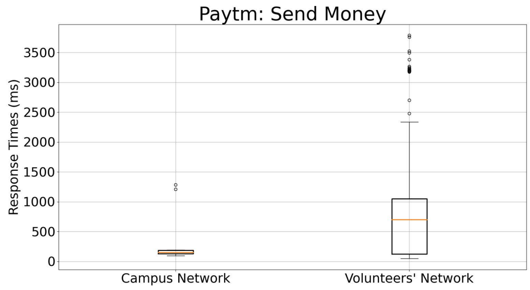


Figure 3.2: Comparison of response times obtained when run on the Campus network, NKN and the Volunteer's network for three applications a) YouTube, b) Dailyhunt and c) Paytm

We confirm this observation using a controlled experiment on our campus situated in a major city using a Samsung A50S smartphone having 4GB of RAM during a period of low congestion. Our campus is connected to the National Knowledge Network (NKN) via a Gigabit link, run by the Government exclusively for educational institutions. The network latency of NKN is known to be significantly smaller than the commercial home networks, with the ping RTT to Google server having a median of just $9.1ms$ compared to $67.6ms$ seen by our volunteers. We repeat these experiments 20 times on the phone and observe a significant reduction in response times for most of the actions. For example, the median response time of "Search Channel" on YouTube (Figure 3.2(a)), "Open News" on DailyHunt (Figure 3.2(b)), and "Send Money" on PayTm (Figure 3.2(c)) reduces by $10.32x$, $11.43x$, and $4.58x$ times respectively. Only a limited number of actions, such as "Remove From Cart" of Amazon, "Send Payment" of Payment, and "Open News" on Google News do not see a significant difference. This further confirms that network latency is the primary bottleneck for Indian users.

We then note that the ping median among the features is itself a function of the distance from a major city, time of day as well as the network type. Therefore, we now remove ping median as a feature, and perform the analysis again (Figure 3.1(b)). We note that distance from a major city as a factor dominates in most cases. Distance is the most significant factor for Flipkart, Facebook, PayTM, and Hotstar, with feature importance scores of 0.52, 0.72, 0.51, and 0.43. This suggests that there is a large divergence across regions in quality of experience depending on the location of the users.

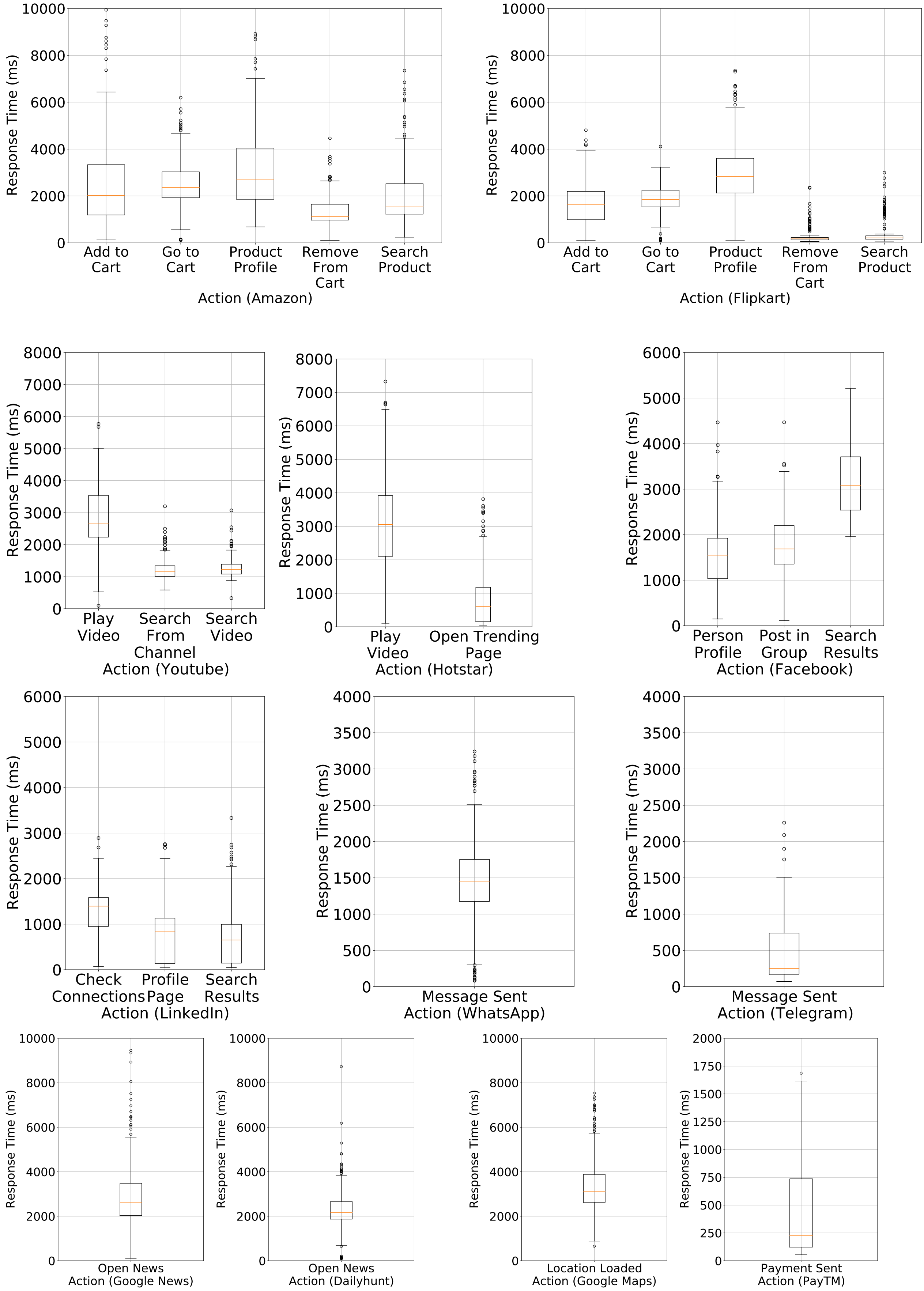


Figure 3.3: Distribution of Response Times for each action of the 12 different apps. Note that apps of similar types are plotted closer beside one another and have identical scales for the sake of comparison.

We further observe that the response time is not governed by the distance for all the apps. For example, the response time of Google Maps is most strongly governed by the RAM available, with a feature importance score of 0.38. The response times of Amazon and Google News are most strongly affected by the time of day, thus suggesting that the server response time is most important. Youtube is most strongly affected by the type of network (WiFi/cellular) with a feature importance score of 0.24, although distance still plays a major role as it has a feature importance score of 0.21. Surprisingly, we further note that the type of network does not play a significant role for other apps. This implies that users utilizing cellular networks compared to WiFi do not experience a significant reduction in response times.

Divergence of Response Times: We compare the coefficient of variation (COV) or relative standard deviation of the response times across all users with the root mean square of the standard deviations taken for each user (Figure 3.1(c)). Note that we avoid comparing the standard deviations since standard deviations can only be interpreted in conjunction with the mean values. We find that the COV's have very similar values, with a median difference of only equal to 0.035. This indicates that the divergence of response times across users is close to that of individual users. We now further look into the root cause of this divergence, by computing the COV's for the ping RTT values as well. We then compute the correlation between the COV's of the ping RTT's (to Google server) and the difference of COV's for the response times. We find a Pearson correlation coefficient of 0.64, which shows that the divergence of ping is strongly correlated with the divergence of the response times. Interestingly, no other factor, including the region or time of day had a significant impact on the divergence of response times. This strongly indicates that the backhaul network is the key factor that increases the divergence of response times of most users. This further suggests that app designers should reduce data consumption to improve response time.

3.1.3 Effect of App Version Upgrades

We further perform controlled lab experiments to analyze the changes in response times with version upgrades across two different android versions i.e. Android *5.1.1* and Android *8.0.0*. We present the results for four such apps Google Maps, Telegram, YouTube and Flipkart.

Android 5 Experiments

For Google Maps 3.4a), we can see from the graphs that the median latency has increased from 4s to about 8.5s with version update from *V 9.67.1* to *V 10.8.1*. In *V 10.32.2*, the median latency drops to about 2.5s. The Rx and Tx plots reveal that a similar trend was visible, wherein the data received and transmitted by *V 10.8.1* was the highest.

In the case of Flipkart 3.4b), the response time values are similar for *V 6.10*, *V 6.15*, while the values are significantly higher for *V 7.15*. For example, the median latency has increased from 0.5s to around 2.8s. Similarly, we see that both Tx and Rx bytes have increased significantly

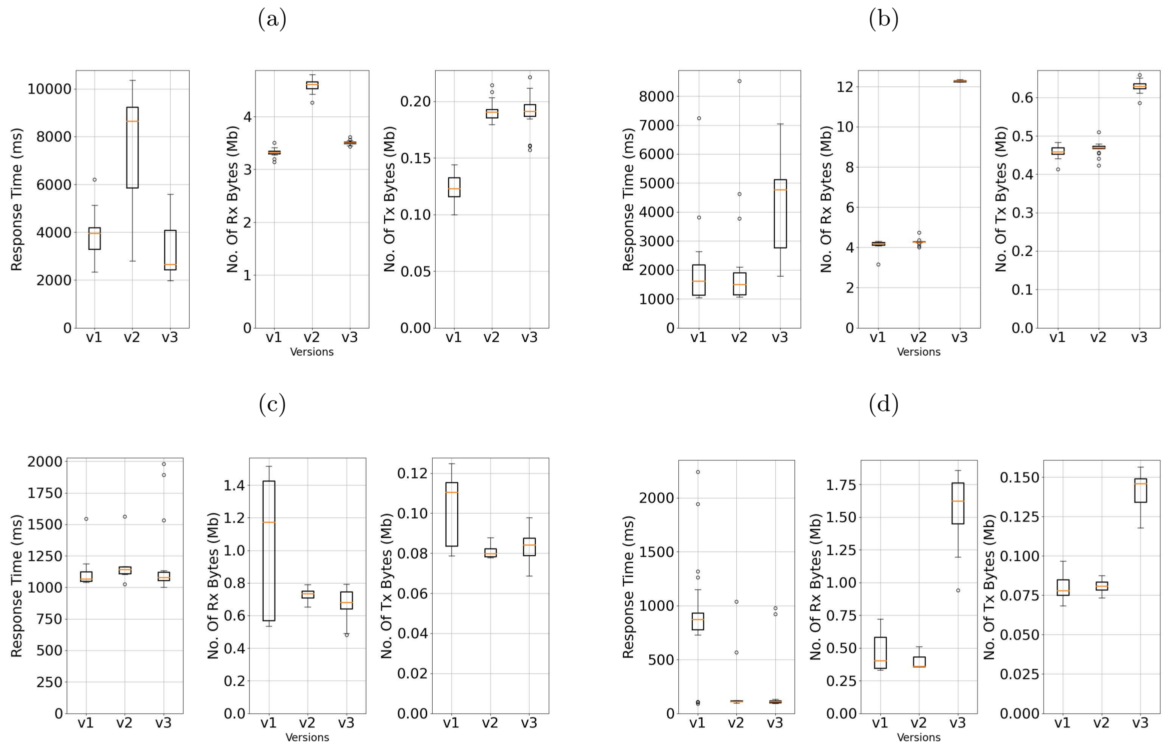


Figure 3.4: Response Times for experiments run on Android 5 a) Google Maps "Search Place" action b) Flipkart "View Profile" action c) YouTube "Search Channel" action d) Telegram "Send Message" action for three app versions

with version updates. The Rx bytes have increased from 4 MB to more than 12 MB from *V 6.10* to *V 7.15*.

For YouTube with versions *V 14.43.55*, *V 15.50.35*, *V 16.02.32* shown in Figure 3.4c), we can see that that the response times are similar for all the three versions, with *V 15.50.35* having a slightly higher response time. The median Rx bytes and Tx bytes for *V 14.43.55* are way higher than for *V 15.50.35* and *V 16.02.32*.

For Telegram with versions *V 5.15*, *V 6.3.0*, *V 7.6.0* shown in Figure 3.4d), the latency for *V 5.15* is higher than both *V 6.3.0* and *V 7.6.0*, whereas the Rx bytes and the Tx bytes are higher for *V 7.6.0*. In this case, latency, Rx and Tx bytes all follow the same pattern of variation wherein the values first see a decrease from *V 5.15* to *V 6.3.0* and then increase from *V 6.3.0* to *V 7.6.0*.

Android 8 Experiments

Figure 3.5a) plots response times of searching a place on Google Maps versions *V 9.67.1*, *V 10.8.1* and *V 10.32.2*. We see that both median and 75^{th} percentiles response times have increased significantly from around 3.8s to 6s, with version update from *V 9.67.1* to *V 10.8.1*. With *V 10.32.2*, it is around 4s. To identify the cause, we looked at the transmitted (Tx) and received bytes (Rx) across the versions. We see that both Tx and Rx bytes have increased significantly

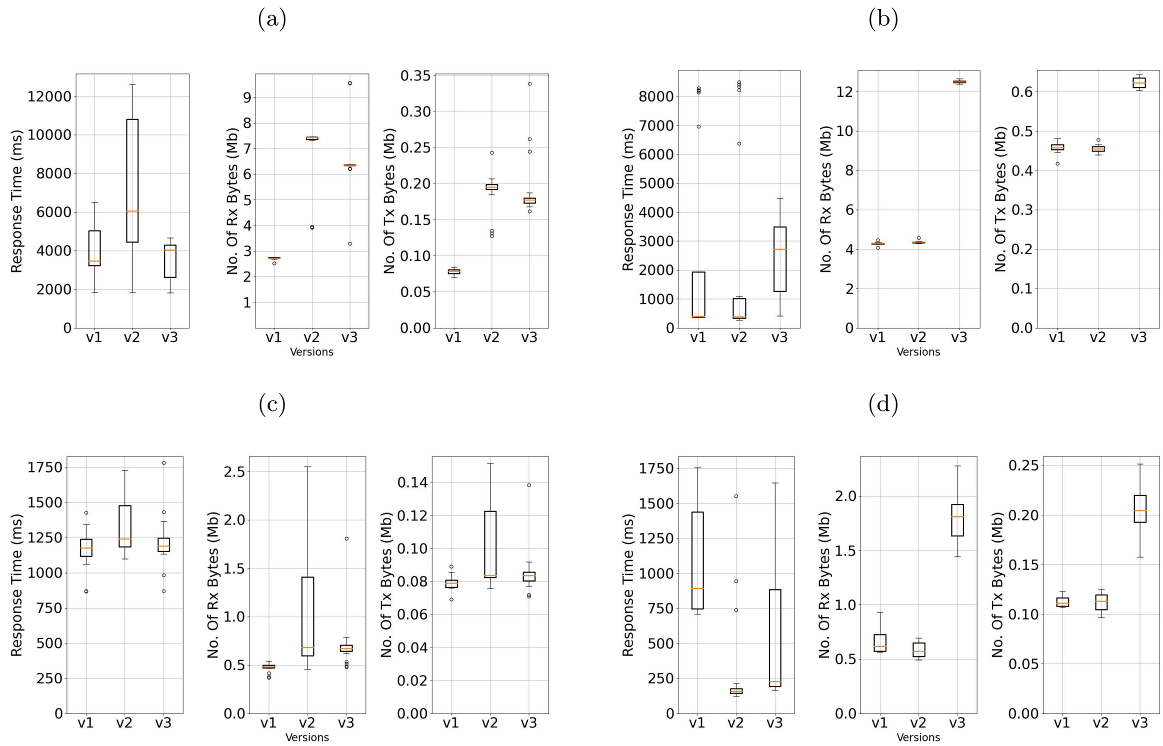


Figure 3.5: Response Times for experiments run on Android 8 a) Google Maps "Search Place" action b) Flipkart "View Profile" action c) YouTube "Search Channel" action d) Telegram "Send Message" action for three app versions

with version updates. The Rx bytes have increased from 2.8MB to around 7.4MB and Tx bytes from 80 KB to 200KB.

Figure 3.5b) plots response times for viewing the details of a product on Flipkart for versions *V 6.10*, *V 6.15* and *V 7.15*. We see that even though the response time values are similar for *V 6.10*, *V 6.15*, the values are significantly higher for *V 7.15*. For example, the median latency has increased from 0.5s to around 2.8s. Similarly, we see that both Tx and Rx bytes have increased significantly with version updates. The Rx bytes have increased from 4MB to more than 12MB from V6.10 to V7.15. For Telegram and YouTube, we have observed a similar relationship between the response times and the amount of data usage across app versions i.e. response times have increased with Tx and Rx bytes across version upgrades. Hence, we conclude that having a higher app version does not always ensure lower response times.

For YouTube with versions *V 14.43.55*, *V 15.50.35*, *V 16.02.32* shown in Figure 3.5c), we can see that that the response times are similar for all the three versions, with *V 15.50.35* having a slightly higher response time. The median Rx bytes and Tx bytes for *V 15.50.35* and *V 16.02.32* are similar.While *V 14.43.55* has the lowest median value of Rx and Tx bytes.

For Telegram with versions *V 5.15*, *V 6.3.0*, *V 7.6.0* shown in Figure 3.5d), the latency for *V 5.15* is higher than both *V 6.3.0* and *V 7.6.0*, whereas the Rx bytes and the Tx bytes are higher for *V 7.6.0*. In Android 8 as well, latency, Rx and Tx bytes all follow the same pattern of variation wherein the values first see a decrease from *V 5.15* to *V 6.3.0* and then increase from *V 6.3.0* to *V 7.6.0*.

Hence, for both the android OS versions, we conclude that having a higher app version does not always ensure lower response times. Also in most cases we have observed that there exists a relationship between the response times and the amount of data usage across app versions i.e. response times have increased with Tx and Rx bytes across version upgrades.

Chapter 4

Related Works

The first category of related works optimize and measure the latency in mobile web and apps, whereas the second category studies the disparities in network latency across time and region. The third category deals with the performance of smartphones and networks in developing countries.

Latency Optimization: A number of studies aim to measure and/or optimize the latency of mobile web and apps [6,15,21,23,26,28]. AppInsight [23] and WProf [21] identify the critical path in the execution of mobile apps and web respectively. The work [18] compares the differences in latency between mobile apps and web apps if they provide the same functionality, and shows that mobile apps tend to have lower latency than web apps in most cases. QoE-Doctor [6], PerfProbe [15] and AppInsight [23] use UI automator provided by the Android framework for analyzing the latency of different apps. Such an approach requires modification of the APK, which makes the users uncomfortable in performing the experiments as they are unaware of the time period of data logged. QoE Doctor and PerfProbe also do not crowdsource data collection, thus obviating their need to convince users to volunteer for the experiments. Finally, SIF [12] and PUMA [13] design automated techniques of analyzing the app properties and testing the user interface similar to EvalApp, but do not collect crowdsourced data.

Disparities in Global Network Latency: M-Lab [14] performs active tests to identify regional and longitudinal variations in Internet latency across the world. The work [5] identifies the geographical and protocol-level differences that lead to large diversities in the end-user Internet latency. The work [8] quantifies the latency observed from different regions while playing videos over the streaming platform YouTube. Also, [4] identifies the disparities in last-mile latency depending on the time of day, ISP used, and geographical location. We do not focus on network latency directly, but on the response times observed by app users.

Network Characterization in Developing Regions: The works [9] and [27] look at the web latencies within Africa and specifically within Ghana respectively. Sharma et al. [25] study the latency observed by Indian cellular networks before the deployment of LTE. WebMedic [20] characterizes the webpages visited by users in developing countries and the smartphones used, and identified memory as the bottleneck. Another work [3] characterizes the smartphone hardware used in a specific district in Pakistan. Our work is orthogonal as we focus on apps, with the same goal of identifying the problems of smartphone users in developing countries.

Chapter 5

Conclusions

In this paper, we analyzed the response time of popularly used apps in India. We designed a tool EvalApp to automate the collection of response times. We collected response time of a total 30 actions on 12 popularly used apps from 41 volunteers. We observed that in a majority of the cases, ping round-trip-time to a Google server is strongly correlated with the response times. This suggests that the network is the most contributing factor in terms of the response time. We found that the next major cause is the distance from the major cities where the data centers are situated. Counter-intuitively, we also found that the response times do not change significantly with the type of network (WiFi/cellular) used or with a better smartphone. This suggests that app developers should focus on reducing the data consumption to reduce response times. We also observed that the response time of apps do not always improve with version upgrades.

The volunteers in our study share the same demography, drawn from our university's students, living in urban areas. Furthermore, they enjoy good Internet connectivity having attended online classes for over a year. We have also excluded the south and north-east of India in this study, due to our limited ability to gather volunteers from these regions. Conducting this study across different demography and other regions including remote regions to remove potential bias is an immediate future work.

Chapter 6

Future Work

The volunteers in our study share the same demography, drawn from our university's students, living in urban areas. Furthermore, they enjoy good Internet connectivity having attended online classes for over a year. We have also excluded the south and north-east of India in this study, due to our limited ability to gather volunteers from these regions. Conducting this study across different demography and other regions including remote regions to remove potential bias is an immediate future work. A detailed analysis to figure out why network affects the app latencies the most is also a potential future work.

Bibliography

[1] Data center map, 2021.

[2] 42matters. India app market statistics in 2021 for android, May 25, 2021.

[3] Ahmad, S., Haamid, A. L., Qazi, Z. A., Zhou, Z., Benson, T., and Qazi, I. A. A view from the other side: Understanding mobile phone characteristics in the developing world. In *Proceedings of the 2016 Internet Measurement Conference* (New York, NY, USA, 2016), IMC '16, Association for Computing Machinery, p. 319–325.

[4] Bajpai, V., Eravuchira, S. J., and Schönwälder, J. Dissecting last-mile latency characteristics. *SIGCOMM Comput. Commun. Rev. 47*, 5 (Oct. 2017), 25–34.

[5] Bozkurt, I. N., Aguirre, A., Chandrasekaran, B., Godfrey, P. B., Laughlin, G., Maggs, B., and Singla, A. Why is the internet so slow?! In *Passive and Active Measurement* (Cham, 2017), M. A. Kaafar, S. Uhlig, and J. Amann, Eds., Springer International Publishing, pp. 173–187.

[6] Chen, Q. A., Luo, H., Rosen, S., Mao, Z. M., Iyer, K., Hui, J., Sontineni, K., and Lau, K. Qoe doctor: Diagnosing mobile app qoe with automated ui control and cross-layer analysis. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (2014), IMC '14, p. 151–164.

[7] Choi, B., Kim, J., Cho, D., Kim, S., and Han, D. Appx: An automated app acceleration framework for low latency mobile app. In *Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies* (New York, NY, USA, 2018), CoNEXT '18, Association for Computing Machinery, p. 27–40.

[8] Doan, T. V., Pajevic, L., Bajpai, V., and Ott, J. Tracing the path to youtube: A quantification of path lengths and latencies toward content caches. *IEEE Communications Magazine 57*, 1 (2019), 80–86.

[9] Fanou, R., Tyson, G., Fernandes, E. L., Francois, P., Valera, F., and Sathiaseelan, A. Exploring and analysing the african web ecosystem. *ACM Trans. Web 12*, 4 (Sept. 2018).

[10] Foundation, O. Appium mobile automation, 2021.

[11] Foundation, O. Javafx library, 2021.

[12] Hao, S., Li, D., Halfond, W. G., and Govindan, R. Sif: A selective instrumentation framework for mobile applications. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services* (New York, NY, USA, 2013), MobiSys '13, Association for Computing Machinery, p. 167–180.

[13] Hao, S., Liu, B., Nath, S., Halfond, W. G., and Govindan, R. Puma: Programmable ui-automation for large-scale dynamic analysis of mobile apps. In *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services* (New York, NY, USA, 2014), MobiSys '14, Association for Computing Machinery, p. 204–217.

[14] Høiland-Jørgensen, T., Ahlgren, B., Hurtig, P., and Brunstrom, A. Measuring latency variation in the internet. In *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies* (2016), CoNEXT '16, p. 473–480.

[15] Hong, D. K., Nikravesh, A., Mao, Z. M., Ketkar, M., and Kishinevsky, M. Perfprobe: A systematic, cross-layer performance diagnosis framework for mobile platforms. In *2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems (MOBILESoft)* (2019), pp. 50–61.

[16] (IDC), I. D. C. India smartphone market registered 42% yoy growth, record october shipments of 21 million units, says idc india, 17 Dec 2020.

[17] Keelery, S. Mobile phone internet users in india 2015-2023, Oct 16, 2020.

[18] Ma, Y., Liu, X., Liu, Y., Liu, Y., and Huang, G. A tale of two fashions: An empirical study on the performance of native apps and web apps on android. *IEEE Transactions on Mobile Computing 17*, 5 (2018), 990–1003.

[19] Maity, M., Raman, B., and Vutukuru, M. Tcp download performance in dense wifi scenarios: Analysis and solution. *IEEE Transactions on Mobile Computing 16*, 1 (2017), 213–227.

[20] Naseer, U., Benson, T. A., and Netravali, R. Webmedic: Disentangling the memory-functionality tension for the next billion mobile web users. In *Proceedings of the 22nd International Workshop on Mobile Computing Systems and Applications* (New York, NY, USA, 2021), HotMobile '21, Association for Computing Machinery, p. 71–77.

[21] Nejati, J., and Balasubramanian, A. An in-depth study of mobile browser performance. In *Proceedings of the 25th International Conference on World Wide Web* (Republic and Canton of Geneva, CHE, 2016), WWW '16, International World Wide Web Conferences Steering Committee, p. 1305–1315.

[22] Nejati, J., Luo, M., Nikiforakis, N., and Balasubramanian, A. Need for mobile speed: A historical analysis of mobile web performance, 2020.

[23] Ravindranath, L., Padhye, J., Agarwal, S., Mahajan, R., Obermiller, I., and Shayandeh, S. Appinsight: Mobile app performance monitoring in the wild. In *10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12)* (Hollywood, CA, Oct. 2012), USENIX Association, pp. 107–120.

[24] Rousseeuw, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics 20* (1987), 53–65.

[25] Sharma, A., Kaur, M., Koradia, Z., Nishant, R., Pandit, S., Raman, A., and Seth, A. Revisiting the state of cellular data connectivity in india. In *Proceedings of the 2015 Annual Symposium on Computing for Development* (New York, NY, USA, 2015), DEV '15, Association for Computing Machinery, p. 149–157.

[26] Wu, D., Chang, R. K. C., Li, W., Cheng, E. K. T., and Gao, D. Mopeye: Opportunistic monitoring of per-app mobile network performance. In *2017 USENIX Annual Technical Conference (USENIX ATC 17)* (Santa Clara, CA, July 2017), USENIX Association, pp. 445–457.

[27] Zaki, Y., Chen, J., Pötsch, T., Ahmad, T., and Subramanian, L. Dissecting web latency in ghana. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (New York, NY, USA, 2014), IMC '14, Association for Computing Machinery, p. 241–248.

[28] Zhang, S., Li, W., Wu, D., Jin, B., Chang, R. K., Gao, D., Wang, Y., and Mok, R. K. P. An empirical study of mobile network behavior and application performance in the wild. In *2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS)* (2019), pp. 1–10.