

TATTLE – ‘HERE’S HOW I SEE IT’
CROWD-SOURCED MONITORING AND ESTIMATION OF CELLULAR
PERFORMANCE THROUGH LOCAL-AREA MEASUREMENT EXCHANGE

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Electrical & Computer Engineering

Huiguang Liang

B.Eng. (1st Class Honors), Computer Engineering, National University of Singapore

M.S., Electrical and Computer Engineering, Carnegie Mellon University

Carnegie Mellon University

Pittsburgh, PA, United States of America

May, 2015

ABSTRACT

The operating environment of cellular networks can be in a constant state of change due to variations and evolutions of technology, subscriber load, and physical infrastructure. One cellular operator, which we interviewed, described two key difficulties. Firstly, they are unable to monitor the performance of their network in a scalable and fine-grained manner. Secondly, they find difficulty in monitoring the service quality experienced by each user equipment (UE). Consequently, they are unable to effectively diagnose performance impairments on a per-UE basis. They currently expend considerable manual efforts to monitor their network through controlled, small-scale drive-testing. If this is not performed satisfactorily, they risk losing subscribers, and also possible penalties from regulators.

In this dissertation, we propose Tattle¹, a distributed, low-cost participatory sensing framework for the collection and processing of UE measurements. Tattle is designed to solve three problems, namely coverage monitoring (**CM**), service quality monitoring (**QM**) and, per-device service quality estimation and classification (**QEC**).

In Tattle, co-located UEs exchange uncertain location information and measurements using local-area broadcasts. This preserves the context of co-location of these measurements. It allows us to develop U-CURE, as well as its delay-adjusted variant, to discard erroneously-localized samples, and reduce localization errors respectively. It allows operators to generate timely, high-resolution and accurate monitoring maps. Operators

¹ The meaning of *Tattle*, as a verb, is to gossip idly. By letting devices communicate their observations with one another, we explore the kinds of insights that can be elicited based on this peer-to-peer exchange.

can then make informed, expedient network management decisions, such as adjusting base-station parameters, to making long-term infrastructure investment.

We propose a comprehensive statistical framework that also allows an individual UE to estimate and classify its own network performance. In our approach, each UE monitors its recent measurements, together with those reported by co-located UEs. Then, through our framework, UEs can automatically determine if any observed impairment is endemic amongst other co-located devices. Subscribers that experience isolated impairments can then take limited remedy steps, such as rebooting their devices.

We demonstrate Tattle's effectiveness by presenting key results, using up to millions of real-world measurements. These were collected systematically using current generations of commercial-off-the-shelf (COTS) mobile devices.

For **CM**, we show that in urban built-up areas, GPS locations reported by UEs may have significant uncertainties and can sometimes be several kilometers away from their true locations. We describe how U-CURE can take into account reported location uncertainty and the knowledge of measurement co-location to remove erroneously-localized readings. This allows us to retain measurements with very high location accuracy, and in turn derive accurate, fine-grained coverage information. Operators can then react and respond to specific areas with coverage issues in a timely manner. Using our approach, we showcase high-resolution results of actual coverage conditions in selected areas of Singapore.

For **QM**, we show that localization performance in COTS devices may exhibit non-negligible correlation with network round-trip delay. This can result in localization errors of up to 605.32m per 1,000ms of delay. Naïve approaches that blindly accepts measurements with their reported locations will therefore result in grossly mis-localized data points. This affects the fidelity of any geo-spatial monitoring information derived

from these data sets. We demonstrate that using the popular localization approach of combining Global-Positioning System together with Network-Assisted Localization, may result in a median root-mean-square (rms) error increase of over 60%. This is in comparison to simply using the Global-Positioning System on its own. We propose a network-delay-adjusted variant of U-CURE, to cooperatively improve the localization performance of COTS devices. We show improvements of up to 70% in terms of median rms location errors, even while subjected to uncertain real-world network delay conditions, with just 3 participating UEs. This allows us to refine the purported locations of delay measurements, and as a result, derive accurate, fine-grained and actionable cellular quality information. Using this approach, we present accurate cellular network delay maps that are of much higher spatial-resolution, as compared to those naively derived using raw data.

For **QEC**, we report on the characteristics of the delay performance of co-located devices subscribed to 2 particular cellular network operators in Singapore. We describe the results of applying our proposed approach to addressing the **QEC** problem, on real-world measurements of over 443,500 data points. We illustrate examples where “normal” and “abnormal” performances occur in real networks, and report instances where a device can experience complete outage, while none of its neighbors are affected. We give quantitative results on how well our algorithm can detect an “abnormal” time series, with increasing effectiveness as the number of co-located UEs increases. With just 3 UEs, we are able to achieve a median detection accuracy of just under 70%. With 7 UEs, we can achieve a median detection rate of just under 90%.

ACKNOWLEDGEMENTS

I would firstly like to express my utmost gratitude to Professor Hyong S. Kim, my thesis advisor and mentor, for his patience, belief and support in this journey. Under his tutelage, I learnt both professional and life skills and lessons which have profoundly shaped me as a person and a researcher. In every step along the way, Professor Kim was my *Master Yoda*, and amongst his innumerable teachings, the foremost, devilishly-intricate and deceptively-simple question he taught me to ask and answer was, “*who cares?*” Upon this reflection, I would argue that a meaningful journey to obtain a Ph.D. perhaps necessarily features episodes of being in struggle and mired in self-doubt. Under the guidance of Professor Kim, seeking and finding the answer to “*who cares?*” became the key to overcoming the difficulties of those periods. Then, as it is sung, “*At the end of the storm, there’s a golden sky*”. Thank you Professor Kim for teaching me how to evaluate a problem, and determining if it is *really* worthwhile solving. This may be the most important skill I’ve ever learnt in research.

Next, I want to thank Dr. Hwee-Pink Tan, who leads the ‘Sense and Sense-abilities’ (S&S) Program at the Institute for Infocomm Research (I²R), Singapore. He is first my unwaveringly-supportive friend, then my unit head and thesis co-advisor. His even-temperedness, patience, belief and gentle guidance gave me the boost I needed whenever the going got tough. I am also especially thankful that he hired me for a full-time position when my scholarship funding reached its full term, and as such allowed me to both contribute back to S&S, and at the same time carry on with this research with financial support. Having such a chance of working full-time in an established research unit, under

the leadership of Dr. Tan, was vital in my professional development, as he is the very embodiment of a good research leader that is fair and just, has clarity of thought, foresight, vision, inter-personal and motivational skills, and also the grit and determination to pull the unit through periods of difficulty.

Next, I am thankful for my friend and third co-advisor, Dr. Wai-Leong Yeow, whom I hold tremendous respect for his technical competency and for his project leadership skills. Dr. Yeow is always able to offer sharp insights and advice on the technical matters involving this research. He is always my go-to person whenever I need critical evaluation of my work, and whose advice I always value.

I am indebted to my close colleague and friend, Dr. Hwee-Xian Tan, who is my biggest champion, supporter, and believer of my work, often even more than I believe in myself. In periods of self-doubt and difficulty, she never fails to offer the encouragement and motivation I need to continue. I thank her for freely sharing her struggles during her time as a graduate student, which always gave me comfort that I'm not alone in my experience.

I am glad and honored to have Professor Peter Steenkiste and Professor Seshan Srinivasan as part of my thesis evaluation committee, and providing invaluable, insightful comments to improve my research.

I thank my colleagues and fellow students at both Carnegie Mellon University and S&S, who have contributed in their own ways to make my work possible. Thank you for your patience and your insights given from your various professional points-of-views. I specifically owe gratitude to my collaborator, Dr. Ido Nevat, for introducing me to the quantile regression, a tool which helped me realize a core part of my work on cellular quality prediction.

To my wife, Fiona Li, and to my family, I owe them much for their years of unwavering patience, support, understanding and belief. This thesis is dedicated to them.

To the Agency of Science, Technology and Research (A*STAR), and in turn the Singaporean taxpayer, I humbly say ‘Thank You’ with utmost gratitude, for giving me the A*STAR Graduate Scholarship to pursue my Ph.D. at Carnegie Mellon University. I will never forget my charge to contribute, in whatever capacity my professional training allows, back to the economy of Singapore.

GLOSSARY

1. **RSCP**: Received Signal Code Power, the received power on one code measured on the Primary Common Pilot Channel (CPICH) [1].
2. **RSRP**: Reference Signal Received Power, defined as the linear average over the power contributions (in watts) of the resource elements that carry cell-specific reference signals within the considered measurement frequency bandwidth [1].
3. **UE**: User Equipment, in particular smart devices such as smart phones and tablets.
4. **Down-link**: Direction of data flow, from the cellular base-station down to the UE.
5. **Up-link**: Direction of data flow, from the UE up to the cellular base-station.
6. **UMTS**: Universal Mobile Telecommunications System, a term for the 3G (third generation) radio technologies developed by the 3GPP [2].
7. **HSPA**: High Speed Packet Access, an enhanced 3G (third generation) radio technology which allows Universal Mobile Telecommunications System (UMTS) cellular networks to feature higher data transfer speeds and capacity through the implementation of adaptive modulation and coding (AMC), hybrid automatic repeat requests (HARQ), and greatly reducing the transmission time interval (TTI) [3][4].
8. **LTE**: Long-term Evolution, a new, 4G (fourth generation) cellular network technology that features evolutions in the radio access and core network systems to support a wholly packet-switched, Internet Protocol (IP) based access network [5].

TABLE OF CONTENTS

Abstract	i
Acknowledgements	iv
Glossary	vii
Table of Contents	viii
List of Tables	xiii
List of Figures	xiv
1. Introduction	1
1.1 Background and Motivation	1
1.1.1 Coverage Monitoring	3
1.1.2 Quality of Service Monitoring	4
1.1.3 Quality of Service Estimation and Classification	6
1.2 Thesis Objectives	7
1.3 Thesis Contributions	7
1.4 Thesis Organization	10
2. Background Survey	11
2.1 Crowd-Sourcing and Participatory Sensing	11
2.2 Cellular Network Performance Estimation and Assertion	12
2.2.1 Coverage Assertion: An Empirical Approach	13

2.2.2	Other approaches to cellular network performance monitoring	16
2.3	UE Localization Techniques	17
2.4	Network- and Device-Level Fault Monitoring and Management	20
2.5	Summary of Related Work	21
3.	Tattle - Cellular Coverage Monitoring Through UE Participatory Sensing	24
3.1	Coverage Monitoring – Problem Overview	25
3.2	Contributions	26
3.3	Chapter Organization	27
3.4	Tattle – Monitoring Through Participation	27
3.4.1	Benefits of Low-Power Measurement Exchange	29
3.4.2	The Tattle App – Description and Operation	31
3.4.3	Motivation for Users	33
3.4.4	Tattle – of Power and Location Uncertainty	35
3.4.5	A Note on Security, Privacy and Trust	41
3.5	Hierarchical Clustering With Uncertainty	43
3.5.1	U-CURE: Extending the Original CURE Algorithm	45
3.5.2	Considering Uncertainty in U-CURE	46
3.6	Measurement Collection, Processing, and Signal Coverage Representation with U-CURE	48
3.6.1	Mean Coverage Visualization	49
3.6.2	Region-of-Interest Based CDF Derivation	56

3.6.3	Limitations of Our Current Prototype	59
3.7	Summary	60
4.	Quality Monitoring of Cellular Network Delay With Tattle	61
4.1	Quality Monitoring – Problem Overview	61
4.2	Contributions	64
4.3	Chapter Organization	65
4.4	Localization in Current Generations of Mobile Devices	66
4.5	Experimental Methodology and Localization Performance	68
4.5.1	Experimental Devices	68
4.5.2	Mobility	69
4.5.3	Network Delay	71
4.5.4	Distance to Ground Truth Computation	72
4.5.5	Effects of Device Heterogeneity	72
4.5.6	Effects of Combining GPS with Network Localization vs. Using GPS Only	76
4.6	Tattle – Cooperative Localization through Delay-Adjusted Clustering	80
4.6.1	Delay-Adjusted Clustering using U-CURE	81
4.6.2	Experimental Evaluation of Delay-Adjusted U-CURE	83
4.6.3	Putting It Together – Tattle, Delay-Adjusted U-CURE, and Building Crowd-Sourced Quality Monitoring Maps	85
4.7	Summary	92

5.	Estimating and Classifying UE-Specific Network Performance through Tattle	94
5.1	Quality Estimation and Classification – Problem Overview	94
5.2	Contributions	96
5.3	Chapter Organization	97
5.4	Tattle – Cooperative Estimation and Classification of Network Delay	97
5.5	Optimal Classification Based On Robust Estimation	100
5.5.1	Algorithm Description	101
5.5.2	Addressing the QEC problem	107
5.6	Experimental Setup and Results	107
5.6.1	Temporal Correlation of Network Delay	109
5.6.2	Choice of Probed Servers	111
5.6.3	Performance Differences between Cellular Providers	112
5.6.4	Performance Differences between Device Models	112
5.6.5	“Normal” vs. “Abnormal” Performance	113
5.6.6	Outlier Detection Performance	116
5.6.7	Effectiveness of the Overall Algorithm	117
5.6.8	False Negatives vs. False Positives Performance	119
5.7	Summary	120
6.	Conclusions and Future Directions	122
6.1	Summary of Research Contributions	122
6.2	Future Directions	125

6.2.1	Applicability to Indoor Positioning Techniques	125
6.2.2	Automatic Coverage Adaptation through Network Reconfiguration	126
6.2.3	Heterogeneous Networks	126
6.2.4	Validation-as-a-Service for Other Mobile Applications	127
	References	128

LIST OF TABLES

Table 1: Summary of pertinent differences between Coverage Monitoring and Quality Monitoring.....	63
Table 2: The devices that were used in our experiments. For units of the same make and model, we updated all devices to their latest official stable firmware, with no other after-market apps installed, except for the location-collection app.	69
Table 3: Gradient of least-squares fit (rms error per second of network round-trip delay)	76
Table 4: The devices that were used in our experiments. For units of the same make and model, we updated all devices to their latest official firmwares, with no other after-market apps installed, except for the Tattle app.....	107
Table 5: The Ljung-Box Q-Test for temporal autocorrelation. The null hypothesis \mathcal{H} null is constructed such that the autocorrelation coefficients $\rho_1, \rho_2, \dots, \rho_x$ considering the first x lags are jointly zero (that is, completely uncorrelated).....	110

LIST OF FIGURES

Figure 1: Newly-constructed high-rise buildings often suffer from poor coverage, especially at higher floors. The existence of such poorly-covered areas are mainly discovered through subscriber complaints. 2

Figure 2: The scope of this dissertation, addressing two general aspects of network performance monitoring, namely the challenge of monitoring coverage and quality on a large, aggregated scale, as well as monitoring performance on a per-UE basis. 9

Figure 3: An overview of how the areas surveyed are related to the work presented in this dissertation.11

Figure 4: A diagrammatic description of Tattle, a distributed, low-cost and comprehensive cellular network measurement collection and processing framework. In this chapter, we exemplify Tattle by leveraging on participating UEs to report on network coverage in real-time.28

Figure 5: Participating users form local-area communication networks using interfaces such as WiFi Direct, to exchange signal strength readings. Elected Group Owners forward overheard readings to the network.30

Figure 6: Screen capture of the Tattle app in operation. Nearby devices will be invited to join the network, and periodically exchange GPS location and signal readings.....32

Figure 7: Battery drain test demonstrating the battery consumption of the naïve reporting method vs. the Tattle prototype round-robin approach.35

Figure 8: CDF plot of the root-mean-square distances of 18,191 static sample points from a known true location.36

Figure 9: CDF plot of the pair-wise root-mean-square distance of sample locations known to be co-located from local-communication exchange.37

Figure 10: A snapshot of device reports taken from our urban outdoor experiments. All 6 devices were actually co-located in a fitted box, each spaced apart by Styrofoam paddings. The spread of reported locations demonstrate the problem of inaccurate location reports. We will further investigate deviations from location ground truths in Chapter 4.38

Figure 11: A plot of 1,628 Received Signal Code Power measurements and their corresponding localization errors in a challenging high-speed mobility condition, collected by 3 sets of GT-P3100s. The Pearson correlation coefficient between RSCP and localization error is -0.0441, suggesting little to no correlation between the two variables.39

Figure 12: A plot of 9,226 location measurements and their corresponding localization errors, collected by 4 sets of SM-T325s. The Pearson correlation coefficient between the reported location uncertainties and actual error distances between the reported location means, and the actual ground truth, is 0.6344, suggesting a strong correlation between the two variables.40

Figure 13: The U-CURE algorithm.44

Figure 14: Mean coverage map obtained using naïve reporting, with 39,947 sample points in the region (1.307, 103.791) to (1.320, 103.763), in given as latitude and longitude coordinates. The size of the area is approximately 1.5 km by 3.2 km. The scatter plot of sample points' mean positions is given at the top. The bottom represents the mean signal map (where orange to red regions represent areas of barely- to unacceptable coverage, i.e. -90 dBm to -110 dBm, and blue to yellow regions represent regions of excellent- to barely acceptable coverage -50 dBm to -90 dBm). Areas of interest are marked out with rectangles.....51

Figure 15: Mean coverage map obtained using the standard CURE algorithm, with 39,947 sample points in the region (1.307, 103.791) to (1.320, 103.763), in given as latitude and longitude coordinates. The size of the area is approximately 1.5 km by 3.2 km. Contrast this with Figure 14 and Figure 16.52

Figure 16: Mean coverage map obtained using the U-CURE algorithm, with 39,947 sample points in the region (1.307, 103.791) to (1.320, 103.763), in given as latitude and longitude coordinates. The size of the area is approximately 1.5 km by 3.2 km. Contrast this with Figure 14 and Figure 15.53

Figure 17: Mean coverage map of a high-rise residential area obtained using naïve reporting, derived from 30,352 samples in the region (1.309, 103.772) to (1.312, 103.768). The size of the area is approximately 350 m by 450 m. This sub-region was chosen from the above region to highlight the importance of removing wrongly-localized sample points.

The $1-\sigma$ uncertainty radii were plotted for points with location uncertainty exceeding 90 m.54

Figure 18: Mean coverage map of a high-rise residential area obtained using the U-CURE algorithm, derived from 30,352 samples in the region (1.309, 103.768) to (1.312, 103.772). The size of the area is approximately 350 m by 450 m. Contrast this with Figure 17. Here, no samples with $1-\sigma$ uncertainty exceeding 90 m remains.....55

Figure 19: The spatial plot of differences in the mean RSCP maps observed in Figure 17 and Figure 18.56

Figure 20: The figures above represent the cumulative probability distribution of (a) the $1-\sigma$ location uncertainty of sample points before and after processing, (b) RSCP for GT-P3100s in the same area, and (c) RSCP for Asus Nexus 7s in the area, constructed from 34,837 sample points in the region-of-interest from (1.332, 103.741) to (1.335, 103.743).57

Figure 21: Indoor residential area, where Tattle’s effectiveness is tapered.59

Figure 22: The localization technique selection screen presented to the user, captured from a Samsung Galaxy TabPRO SM-T325. All Android mobile devices display similar variants of this menu to their users.....67

Figure 23: An outdoor pedestrian path measuring around 940 meters in length.70

Figure 24: An outdoor high-speed train route measuring approximately 8 kilometers in length.70

Figure 25: (a) Nexus 7 / (b) GT-P3100 / (c) SM-T325 - Network localization performance under pedestrian mobility.....	73
Figure 26: (a) Nexus 7 / (b) GT-P3100 / (c) SM-T325 - Network localization performance under high-speed mobility.	74
Figure 27: (SM-T325) - (a) GPS+NLP vs. (b) GPS-Only under pedestrian mobility....	77
Figure 28: (SM-T325) - (a) GPS+NLP vs. (b) GPS-Only under high-speed mobility....	77
Figure 29: (SMT-325) Performance of (a) GPS+NLP vs. (b) GPS-Only, without artificially-introduced network delay, in a high-speed mobility environment.	78
Figure 30: (a) Cumulative distribution function of GPS-Only vs. GPS+NLP, and (b) the corresponding quantile-based percentage improvement, without artificially-introduced network delay.....	78
Figure 31: A real-world example of using Delay-Adjusted U-CURE to improve the individual devices' reported positions by shifting them towards the primary cluster centroid by a function of their network round-trip delays.....	82
Figure 32: (a) Cumulative distribution function of using GPS+NLP after delay-adjusted U-CURE processing, and (b) the corresponding quantile-based percentage improvement, across different number of neighbors.	83

Figure 33: Heat-map of crowd-sourced network round-trip delay using GPS+NLP, before (top), and after (middle) delay-adjusted U-CURE processing, vs. the ground-truth (bottom).....87

Figure 34: Quantitative evaluation of the heat-map derivation accuracy, based on (a) the rms error of measured delay, and (b) the amount of ‘smudge’ generated by mis-localized data points, using GPS+NLP localization.....89

Figure 35: Heat-map of crowd-sourced network round-trip delay using GPS-Only localization, before (top), and after (middle) delay-adjusted U-CURE processing, vs. the ground-truth (bottom).....90

Figure 36: Quantitative evaluation of the heat-map derivation accuracy, based on (a) the rms error of measured delay, and (b) the amount of ‘smudge’ generated by mis-localized data points, using just GPS localization.....91

Figure 37: A representative plot of real HTTP fetch delay measurements conducted by 8 co-located devices. Each device fetches a small HTML resource from `google.com.sg` every 5 seconds and measures the delay taken for the fetch to be completed. It is noteworthy that even while co-located, some devices can experience very large variability in network delay, while others see consistently low delays.102

Figure 38: The train station platform, as well as the individual spots where data collection was performed, are illustrated in this figure.108

Figure 39: The distribution of autocorrelation coefficients for 2,680 time-series of network delay. Each time-step is 5 seconds in length. Data were collected with the GT-P3100's, Nexus 7's, and SM-T325's on Provider #1's network.109

Figure 40: (a) A comprehensive characterization of the HTTP fetch delays for `google.com.sg`, experienced by each device in our entire data-set of over 220,000 data points. Provider #2's network performs considerably and observably worse than Provider #1's network in largely every experiment. (b) The same plot is given, but these devices instead perform HTTP fetches on our self-managed UNISENSE server, located in Singapore.....109

Figure 41: An illustration of two sets of representative time-series, representing the HTTP fetch delay for `google.com.sg`. The devices were co-located at the same place, and are the exact same make and model, but 3 were connected to Provider #1's network, while the others were connected to Provider #2's network. The interval between each time step is 5 seconds.....111

Figure 42: An illustration of three sets of representative time-series, representing the HTTP fetch delay for `google.com.sg`. The devices were co-located at the same place, using the same provider, but are arranged according to their makes and models. Periods of high delay can affect many devices in the same area, as shown.113

Figure 43: An illustration of three sets of representative time-series, representing the HTTP fetch delay for `google.com.sg`. The devices were co-located at the same place, using the same provider, but are arranged according to their makes and models.

Oftentimes, a device may experience severe outage of network service, but most other devices are unaffected.114

Figure 44: An illustration of two sets of time-series. Here, a time-series of HTTP fetch delays for `google.com.sg`, belonging to a device connected to Provider #2’s network, is mixed into that of the data set of time-series of fetch delays for devices on Provider #1’s network. We chose this foreign time-series from Provider #2 because it offers many points in time in which the delay experienced by the foreign device is radically different from those on Provider #1’s network. The goal of our algorithm is then to detect those points in time where the foreign time-series is deemed to be behaving “abnormally” from others. Those detected points are then marked out as shown.115

Figure 45: A complete characterization of the misdetection rates of a foreign time-series of HTTP fetch delays, belonging to Provider #2’s network, when mixed into the time-series of delays experienced by devices on Provider #1’s network. As the number of neighbors considered by the algorithm increases, the detection performance of our algorithm becomes better.118

Figure 46: An illustration of the false negative rates of our algorithm. The false positive rate can be inferred by flipping the line-series about the (0,0)-(100,1) diagonal. Of all the times that a misdetection happens, having more neighbors helps to suppress the false negative rate, in favor of the false positive.118

1. INTRODUCTION

We are becoming increasingly-dependent on our smart phones and smart devices to assist us in our day-to-day activities. These can range from booking cabs, searching for nearby restaurants, navigation, to couponing for retail discounts. Pervasive applications such as these have helped to fuel an explosive growth of mobile data services, with more than half a billion smart devices and connections added worldwide in 2013. These spurred 2013's year-on-year growth of global mobile traffic to 81% [6]. Up to 102 billion mobile apps were downloaded [7] in 2013 alone.

Robust and resilient cellular networks are necessary to support such growth, both in the number of mobile devices as well as the amount of data that each device generates. However, the operating environment of these cellular networks can be in a constant state of change. Over the course of hours and days, cell loads can change due to diurnal cycles, or events such as spectator sports. Over the course of weeks and months, network topology can change due to the addition of new cells, and resizing of existing cells. Over the course of months and years, changes in the physical operating environment can come about when new physical infrastructure, like buildings, is constructed. An example of such a situation is given in **Figure 1**. Due to evolution of the operating environment, operators may therefore have to regularly tune their networks so that cell service is not degraded.

1.1 Background and Motivation

Singapore's urban landscape, like in many developing cities, is always rapidly evolving. We had a series of interactions with a local network operator to better understand their operating challenges and difficulties. One of the key issues they intimated was that the

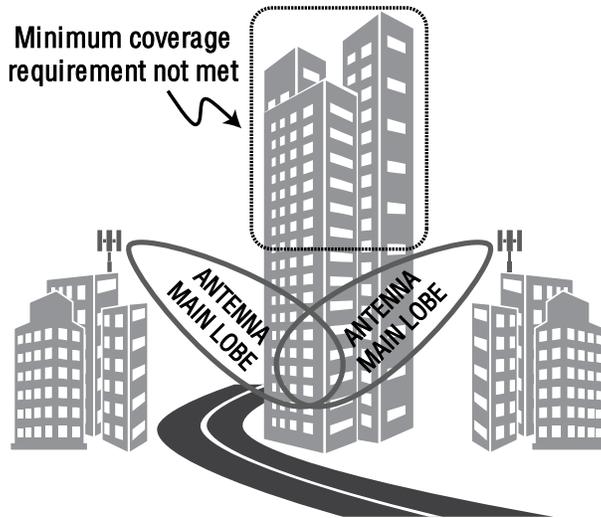


Figure 1: Newly-constructed high-rise buildings often suffer from poor coverage, especially at higher floors. The existence of such poorly-covered areas are mainly discovered through subscriber complaints.

pace of urban development often exceeds the operator's ability to keep up with network reconfiguration and infrastructure investment. Just as in many cities that grapple with space constraints, low-lying buildings constantly make way for increasingly taller ones that routinely exceed 40 floors in height. These new and tall structures cause heavy shadowing and coverage holes. The operator has to rely on subscriber complaints to discover areas that are poorly served. The lead time between discovering the need for a new base-station, to its installation and commission is around 6 months. In the meantime, the operator has to reconfigure existing base-stations to alleviate coverage problems, and yet minimize any impact to areas that are previously well-served.

However, besides just making phone calls and sending short text messages like in the past, many subscribers are already heavily-dependent on cellular data services for various day-to-day aspects of their lives. Any form of impairment, both structural (e.g. insufficient base-stations, antennas, or back-haul capacity) and temporal (e.g. surge of demand due to diurnal travel patterns) will quickly erode subscriber satisfaction and confidence if not

tackled, or at least acknowledged in an expedient manner. Otherwise, subscribers that are chronically-affected by these issues (with little avenues of redress) may switch operators. This dissertation addresses three areas related to the monitoring of cellular networks, namely, (1) building quasi-real-time, accurate cellular coverage maps on large geographical scales to identify coverage holes to meet regulatory requirements, (2) constructing accurate, large-scale cellular quality maps that are robust towards user mobility for quasi-real-time monitoring of subscriber experience, and (3) estimating and classifying local cellular network performance on a per-UE² basis to determine whether a device's observed poor performance is an isolated incident (possibly caused by a device's hardware or software, or simply due to unavoidable physical phenomena such as fading [8]), or endemic to the area it is in.

In the following sections, we briefly discuss the backgrounds and motivations for addressing each of the aforementioned areas.

1.1.1 Coverage Monitoring

Due to evolution of the operating environment, dynamic parameters (such as radio, antenna, handover and load-balancing) may therefore have to be regularly reconfigured so that cell service is not degraded. One of the key problems which the interviewed network operator put to us is coverage monitoring (**CM**): *how can they efficiently verify that minimum coverage is met for an area of interest?* Minimum coverage is defined by the local regulatory authority to be a threshold percentage of Received Signal Code Power

² UE, or *User Equipment*, is a common terminology in 3rd Generation Partnership Project (3GPP) standards that refers to subscriber devices that consume the network provider's cellular services. In this dissertation, we use the term *device*, *smart device*, and *UE* interchangeably to refer to current generations of mobile devices such as smart phones and tables.

(RSCP) samples that must exceed -100 dBm. The RSCP is the received code power from the down-link Common Pilot Channel (CPICH), broadcast as a “beacon” from cellular base-stations [1]. In an outdoor environment, at least 99% of all sampled RSCP measurements must exceed -100 dBm. In an indoor environment, at least 85% of measurements must exceed that same threshold. These requirements are invariant to other spatial-temporal evolutions, such as time-of-day, month-of-day, or how populated or crowded an area is.

The operator currently follows a manual approach. Areas with poor coverage are first identified through subscriber feedback. Coverage is then appraised by manual walk-tests. If minimum coverage is not attained, radio and antenna configuration parameters (such as pan, tilt and power) are iteratively tweaked. After each iteration, walk-tests will be conducted to re-appraise coverage. This is repeated until minimum coverage is obtained. Furthermore, areas that were previously amply-covered should not be significantly impacted after this tuning. Such an approach is labor-intensive, expensive and limited in scale. It provides a non-real-time and non-continuous snap-shot of coverage. The process has to be repeated whenever the operating environment evolves, especially over long periods due to aforementioned reasons.

1.1.2 Quality of Service Monitoring

The next problem that was put to us is that of quality of service monitoring (**QM**). The operator does not have a method or system that is currently able to effectively monitor the quality of data service experienced by each UE, in a scalable manner. The operator intimates that subscribers often complain about the inability to access remote content using their subscribed data service (e.g. 3G HSPA, LTE). The complaints are also often vague because most complainants lack the technical knowledge to describe the impairment

in a comprehensive manner. Most complaints can be as vague as “*Help, I cannot access the Internet*”, “*I can’t watch videos on YouTube.com*”, or “*My websites are loading slowly*”. The helpline operator then responds with a sequence of pre-determined questions and instructions to try and diagnose the problem. These often includes questions such as “*Where is your current location?*”, “*Are you able to access other websites, such as FaceBook.com?*”, and “*Can you try and restart your device?*”

The customer service center is also often a separate division from that of the network management center. Hence, the complaint generally has to be translated into a trouble ticket, with a description of the technical issue faced by the complainant. If the network is suspected to be the cause of the issue, this trouble ticket is then passed on to the network management and operations team to be further investigated. This approach is tedious and labor-intensive, both for the subscriber and the operator. Besides submitting the complaint, the subscriber generally has no other avenues of support until the operator investigates and addresses the specific issue, or until the problem goes away on its own. The latter often happens when the problem is caused by temporary, or diurnal temporal-spatial congestion, e.g. rush hour in a busy subway station.

Unfortunately, the complainant is left with a poor service experience that cannot be remedied or explained in a timely manner. The operator also loses the ability to gather spatial-temporal context about the complaint, after the fact. Important details such as “*were UEs that were in the same area facing the same impairment?*” are difficult to glean or ascertain once the complaint has been filed.

1.1.3 Quality of Service Estimation and Classification

The general focus in the previous two sections is on monitoring cellular coverage and quality on a large, aggregated scale. Here, we turn our attention towards addressing real-time issues with the quality of experiences for *individual* UEs.

Oftentimes, when subscribers experience frustratingly high network delays and timeouts, they would like to know whether their experiences are shared by other users nearby. The question that is often asked is essentially this: “*is it just me, or do others around me face the same problem?*”

As traffic load increases, cellular networks naturally become increasingly congested. Subscribers tend to experience high network delays, slow speeds, and possibly service outages in cells and areas that are overloaded. This can be a common occurrence and a frustrating experience, especially during periods of congestion, such as rush hours and spectator events.

Whenever a subscriber experiences periods of unsatisfactory network performance, the subscriber may often blame the cellular operator for providing inadequate coverage and resource for the cellular plan they are paying. On the other hand, the cellular operator may deflect the cause of the performance issue back to the subscriber’s device hardware or software. It is then difficult to derive any conclusion to the network performance issue after the fact. In cellular networks, systemic problems manifest themselves in a *fate-sharing* manner (i.e. users in the same area tend to collectively suffer from network-related problems). It is then instructive to estimate and classify network quality of service (**QEC**) on a per-UE basis, as being ‘normal’ or ‘abnormal’, in the context of its neighbors’ observed performances.

1.2 Thesis Objectives

The objectives of this thesis is therefore summarized as follows.

1. Develop an overall framework for cellular network monitoring, which is robust, scalable, timely, works on a large geographical scale with good measurement location fidelity, and requires minimal involvement of the subscribers, as well as minimizes manual efforts by the operators.
2. This framework must be beneficial to both the operators (so that they can efficiently monitor and manage their cellular networks) and the subscribers (so that they can derive meaningful insights and explanations to their own observed performances, and take limited mitigation steps whenever possible). This increases the possibility of adoption.

1.3 Thesis Contributions

The main contributions of this thesis is summarized as follows.

1. The design of Tattle [9], a wireless service monitoring framework that is distributed, low-cost and exploits peer-to-peer measurement-exchange to preserve the co-locality of readings and conserve power. These key features of Tattle allow operators to generate high-resolution, fine-grained, accurate and large-scale monitoring maps. Tattle also allows devices to determine if their service qualities are comparable to those of their co-located neighbors'. To the best of our knowledge, we are the first to implement and demonstrate the achievable, real-world benefits of such a framework when applied to cellular networks. Our findings are derived from an actual proof-of-concept which we developed and tested. The findings are validated through millions

of real-world measurements, collected using commercially-available commodity handsets in actual production cellular networks.

2. The design of an algorithm, U-CURE (extended from the original CURE algorithm [10]), and its delay-adjusted variant. These are exemplary clustering algorithms that the Tattle framework admits, in order to improve the location accuracy of measurements collected using commodity handsets. These algorithms are designed to be key enablers for operators to derive very fine-grained, large-scale, accurate, and actionable coverage and quality information about their networks. Our algorithms result in *much more* accurate geographic representations of the measurements, as compared to naively accepting all measurements and their purported locations in good faith. For a pair of co-located devices that are physically less than 1 meter apart, U-CURE can reduce the maximum positioning distance error from as high as 3 kilometers to less than 40 meters. For a group of just 6 co-located devices, Delay-Adjusted U-CURE can reduce median positioning errors of measurements by over 60%. These are validated through hundreds of thousands of real-world measurements, collected from a production cellular network, using commodity mobile devices. To demonstrate the efficacy of these algorithms, we present actual network quality maps that reveal problematic areas accurate *to the order of meters*. This is in contrast to quality maps generated using existing or naïve approaches, which are often aggregated to the order of hundreds or thousands of meters.

3. The design of a delay estimation and classification framework that uses Tattle and its peer-to-peer measurement-exchange mechanism. It allows a UE to collect measurements from its co-located neighbors, and determine if it is performing normally or abnormally through quantile regression. To the best of our knowledge, we are the

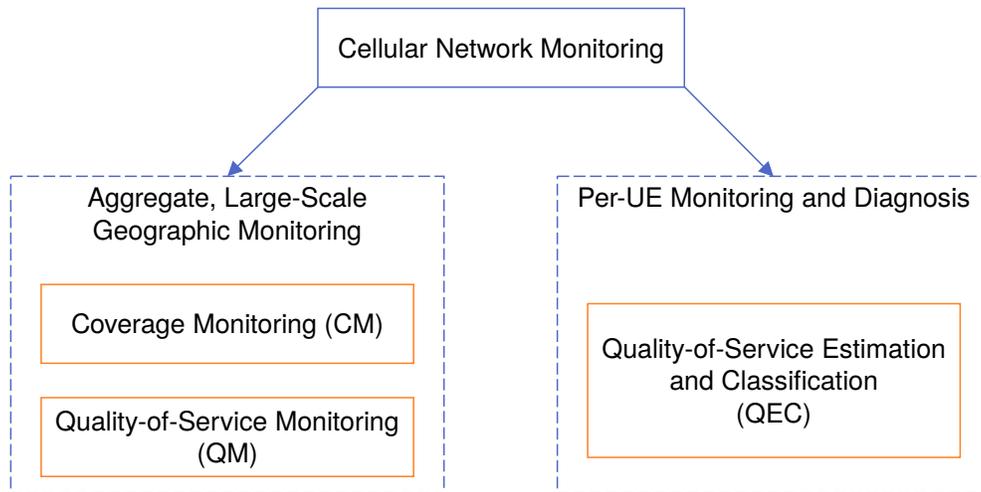


Figure 2: The scope of this dissertation, addressing two general aspects of network performance monitoring, namely the challenge of monitoring coverage and quality on a large, aggregated scale, as well as monitoring performance on a per-UE basis.

first to propose, describe and implement such a peer-to-peer mobile network fault detection application. We validate our approach with hundreds of thousands of data points, collected through actual experiments conducted with COTS mobile devices, in actual production cellular networks. With just 6 participating devices, we are able to detect abnormal network delays with a median accuracy of up to 90%. Of those mis-detected samples, we are able to reduce frustrating false-negative occurrences to just 18%.

Figure 2 illustrates the scope of this dissertation, which addresses two general aspects of network performance monitoring. They are namely the challenge of accurately monitoring coverage and quality on a large, aggregated scale, as well as monitoring performance on a per-UE basis.

1.4 Thesis Organization

This thesis is organized in the following manner. In Chapter 2, we provide a background review of existing work in the areas of cellular network coverage estimation and assertion, UE localization techniques, as well as network- and device-level fault monitoring and management. Next, in Chapter 3, we describe the Tattle monitoring framework, and explain how it can be used, in conjunction with the U-CURE clustering algorithm, to construct large-scale, geographically-accurate cellular coverage maps for monitoring purposes. We showcase real-world results gathered from extensive data-collection and measurements from a production cellular network, using commercial off-the-shelf (COTS) devices that best reflect the reality of network coverage. Then, in Chapter 4, we discuss how the cellular quality may be impacted under different mobility conditions. We use Tattle together with a delay-adjusted variant of U-CURE to construct accurate cellular quality monitoring maps. As before, results from extensive real-world experiments and data collection will be presented to demonstrate the efficacy of our proposed approach. Next, in Chapter 5, we turn our attention to the performance of individual devices, and describe how Tattle, together with a quantile-regressed-based framework, can be used to distill the local performance context of co-located COTS devices. It automatically determines if a device is performing normally, or abnormally with reference to its neighbors. Finally, in Chapter 6, we give concluding remarks, as well as discuss future directions.

2. BACKGROUND SURVEY

This dissertation touches on several areas relating to (1) crowd-sourcing and participatory sensing, (2) cellular network performance estimation and assertion, (3) UE localization techniques, and (4) network- and device-level fault monitoring and management in wireless networks. Although we are unable to exhaustively survey all related work, we reference in this chapter some representative work in each of the above areas. **Figure 3** illustrates how these areas relate to the work presented in this dissertation.

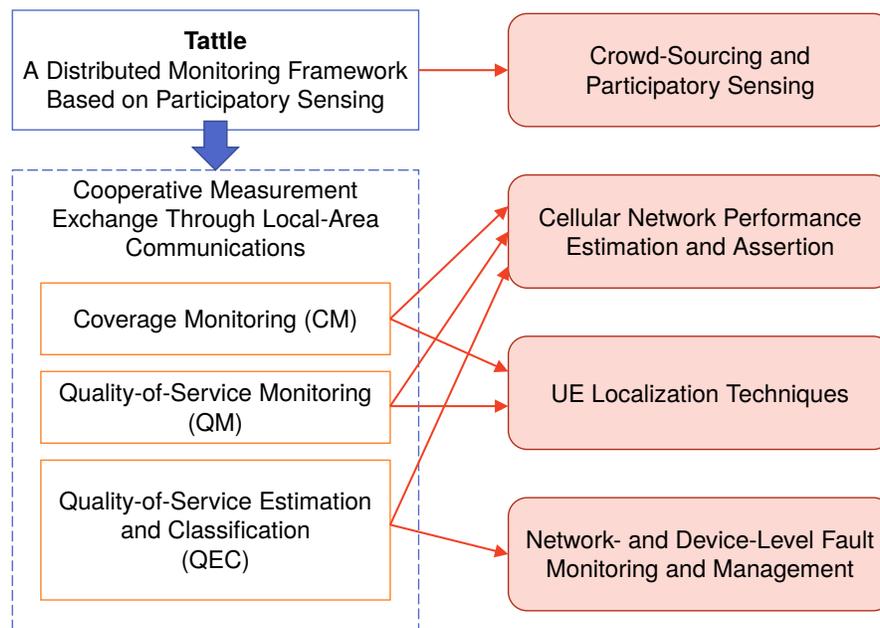


Figure 3: An overview of how the areas surveyed are related to the work presented in this dissertation.

2.1 Crowd-Sourcing and Participatory Sensing

Subscribers are the best monitors of coverage wherever they are and wherever they require service. Hence, rather than expending considerable manual efforts to selectively monitor network performance in small geographic areas, our Tattle framework uses *crowd-sourcing*

and *participatory sensing* [11] as a core approach to gather network performance measurements from subscribers themselves. The key feature of crowd-sourcing and participatory sensing is the engagement and encouragement of willing participants. They contribute individually-gathered sensory information (which, in this case, are network measurements collected by their devices wherever they are) to form a larger body of knowledge (such as a real-time geographic map of network coverage).

The proliferation of smart devices has been a key enabler for many interesting research projects in this area. Related work in participatory sensing leverages on the potential collaborative and participative nature of co-located smart devices to achieve various greater goals. In [12], the authors use commodity smartphones, mounted on vehicle dashboards, to collaboratively predict traffic signals in real-time, using the smartphones' camera and peer-to-peer 802.11 connectivity capabilities. In another study [13], its authors propose a Wi-Fi Access Point recommendation system based on gathering historical and empirical evaluations from participating clients. Other diverse participatory sensing applications include finding suitable biking routes [14], fuel-efficient driving routes [15], and estimating bus arrival times [16].

2.2 Cellular Network Performance Estimation and Assertion

The FCAPS framework (addressing fault, configuration, accounting, performance and security management) forms the core principles of any cellular management system. In [17], the objectives and requirements of cellular network management are broadly described. We begin by examining the background work addressing network coverage monitoring in Section 2.2.1. In Section 2.2.2, we briefly discuss how operators use walk-and drive-tests to monitor the general performance of their networks. We also introduce

other independent efforts currently underway to monitor cellular network performance on a distributed basis.

2.2.1 Coverage Assertion: An Empirical Approach

The *coverage assertion* problem is generally related to performance and configuration management, and is loosely related to the problem of *coverage estimation*. The latter has a rich history of background work, yet because of the complexities of real-world deployments, existing proposed approaches find limited application. These approaches can be categorized into three groups, namely *deterministic*, *stochastic* and *empirical*.

Deterministic models, such as ray-tracing [18], are used when the complete 3-D propagation environment is known. However, obtaining complete knowledge of the propagation environment is prohibitively expensive. Stochastic models are often used in analysis. They typically model coverage as a random variable or random process. A comprehensive treatment of cellular coverage, modeled as a point process, is given in [19]. While stochastic models are often analytically tractable, they have limited use for the *coverage assertion* problem because the assumed propagation models are often too generalized. Empirical estimation is based on empirical observations, such as from UE measurement reports, and drive-tests [20]. We review this in detail as it is the approach currently undertaken by operators.

The operator typically finds out about areas with poor coverage through subscriber feedback. It may dispatch technicians to those specific localities to perform walk-/drive-tests. RSCP samples are generally collected with dedicated measurement and diagnostic equipment. The operator's main complaints about the current approach are that it is too labor-intensive, time-consuming, and expensive to conduct. Subscriber feedback can also often be vague, subjective, and after the fact. Useful spatial-temporal clues (such as the

occurrence of a temporary surge in human traffic, resulting in high radio frequency signal attenuation by human bodies) may be omitted. Operating expense is high, yet a satisfactory survey of signal coverage is not guaranteed for two reasons as follows.

1. It is prohibitively expensive to collect samples from all areas where there might be subscribers (especially in pedestrian or indoor areas).
2. The readings collected by their measuring equipment may not directly reflect the coverage experienced by subscribers due to device heterogeneity.

Instead of a manual approach, the operator desires a system and framework in which quantitative measurements are collected, localized, and reported to the network by the UEs themselves, in real-time.

In the literature, the following generic approach to coverage monitoring is commonly assumed. A typical operator may collect physical layer measurement reports from UEs [20][21][22], monitor cell-wide statistics [23] and alarms [24], and perform periodic drive-tests to assert cell performance. They also depend on subscriber feedback and support ticketing to determine if a cell is providing satisfactory coverage. However, in the context of coverage assertion, our operator finds that directly relying on UEs' reported physical layer measurements has several drawbacks. We list them as follows.

1. The ability to gather measurement reports from UEs is tightly coupled with their abilities to obtain full cell service. Any impairment to the service can affect measurement reporting. For places which are very poorly-served, UEs often do not get a chance to send measurement reports.
2. Measurement reports are made and tagged with any location estimate that the UE may have. No additional effort will normally be imposed to refine location information

for power saving purposes [21][25]. Because of this, most reported measurements tend to have very coarse location information (mostly within the confines of the UEs' cells). Hence, they are only useful for cell-wide analysis. We shall further discuss the current state-of-the-art in device localization in Section 2.3.

3. Direct and frequent reporting of measurements can quickly consume UE power, as well as uplink capacity. This problem is especially important when trying to monitor poorly-served areas, because UEs in those areas have to transmit at higher powers to overcome poor channel conditions.

Due to these issues, the local operator which we interviewed does not depend on UE reported measurements. Instead, they conduct manual walk- and drive-tests where conditions (e.g. mobility, equipment used) are well-controlled. Measurements are tagged with precise GPS locations. However, these results may not be representative of subscriber experience due to device heterogeneity. The geographic extent of their tests is also limited due to resource constraints.

One related study considers overlaying estimated performance information, gleaned from UE measurements, over a geographic map. However, the authors consider all reporting UEs to be perfectly localized either using GPS or the Observed-Time-Difference-Of-Arrival (OTDOA) technique [26]. Parameterized empirical path-loss models, such as the Okumura-Hata or COST-231 model, can be synthesized from large-scale studies. They are useful for coarse-grained estimation of large-scale path-loss [27], but not suitable for coverage assertion. Other in-depth empirical studies [28][29] have investigated real-world performances of cellular networks, but these are expensive to conduct, limited in geographic extent, and difficult to scale.

In this dissertation, we will show through real-world measurements that assumptions of perfect localization may not be valid for current generations of commercial-off-the-shelf (COTS) devices. Based on this, we will propose, in Chapter 3 and Chapter 4, a distributed, scalable peer-to-peer measurement exchange framework to gather performance measurements on a large geographic scale, with good location fidelity.

2.2.2 Other approaches to cellular network performance monitoring

Besides just surveying their network's coverage, a battery of tests for network performance are also conducted during an operator's regular walk- and drive-tests [30][31]. These tests try to detect and capture common issues with cellular networks, such as interference, bit errors, handover failures, and call drops. As aforementioned, this approach, together with customer complaints and trouble-ticketing, is still the primary means of detecting and monitoring network performance today. Hence, the shortcomings of such an approach discussed earlier unfortunately continue to apply here.

Official performance surveys done by operators through these walk- and drive-tests are mostly not released to subscribers (especially if the results are not ideal). Several independent projects are now ongoing, to gather independent, real-world feedback on cellular network performance from subscribers. In [32], the regulatory authority in Singapore maintains a smart device application that allows people to passively contribute various network performance measurements, such as packet loss and voice call drop rates. However, they do not impose any localization accuracy constraint on submitted measurements. Measurements are naively accepted regardless of how egregious the location error may be. Because of this, they may only be able to glean performance information on a coarse geographic basis.

RootMetrics [33] is another popular large-scale cellular performance mapping project. Their aim is to provide independent evaluations of cellular networks through drive- and walk-tests, conducted by a small number of hired ‘scouts’. While the tests that they conduct are exhaustive, they are unable to capture small-scale spatial-temporal evolutions in the network performance as they have a limited number of scouts. Hence, locations are only surveyed twice a year (so the effects of crowd dynamics or changes in physical environments may not be sufficiently captured), and the location resolution of their reported performance maps is in the order of hundreds of meters (which may not be sufficient to capture small coverage holes).

OpenSignal [34] is a similar large-scale cellular performance mapping project. Instead of relying on hired scouts to perform exhaustive tests while walking and driving, OpenSignal makes use of subscribers to contribute network performance measurements. However, it still suffers from the same shortcoming as in [32], where a lack of any localization accuracy constraints may imply that only coarse-grained location information is available. The location resolution of its performance maps, like that of RootMetrics, is in the order of hundreds of meters.

Unlike these efforts, the framework proposed in this dissertation allows the operator to glean geographically fine-grained network performance information, *in the order of meters to tens of meters*.

2.3 UE Localization Techniques

Studies that feature crowd-sourcing and participatory sensing typically assume that smart devices are able to always geo-locate themselves accurately. This is because most devices have Global Positioning System (GPS) and/or network-assisted localization capabilities. In this dissertation, we shall show through extensive real-world measurement collection

that this assumption is not always true. Hence, we briefly discuss background work related to this area.

Localization of cellular devices has always attracted considerable attention in terms of research, and more so in recent years. This is a direct result of the growing pervasiveness of location-based services. Before the emergence of “smart” devices, localization of simple mobile terminals were required by legislation, mostly for emergency-response services to locate users in need of assistance [35].

Early work in mobile localization focused on *network-based* approaches, where the computation of a mobile terminal’s location occurs within the network. This is based on identifiable features such as the terminal’s associated Cell ID [36][37]. Subsequent coarse-grained mobile-assisted localization techniques are generally measurement-based triangulation approaches. These require mobile terminals to measure various signals from multiple base-stations. Time-of-Arrival (TOA), Time-Difference-of-Arrival (TDOA), Angle-of-Arrival information are distilled from these measurements as collected by the mobile terminal. They will then be used by the network to compute coarse-grained location [38]. This is still the current approach in use by network providers today at the network-plane level [21]. The best network-based localization technique in the 3GPP Universal Terrestrial Radio Access Network standard is the Observed-Time-Difference-of-Arrival (OTDOA) method. The main shortcomings are that it incurs high signaling costs if constantly used, and its accuracy suffers from many sources of errors, such as multipath [25].

Recent developments in sensor networks have also touched on cooperative wireless node localization for sensors in-the-wild [39]. Many of these are “anchor-node” based approaches, where one or more nodes in the network have definitive knowledge on their positions.

Other nearby nodes with unknown locations can estimate their positions based on aforementioned techniques like TOA [40]. In particular, ultra-wideband technology is a promising radio interface for node-to-node communications that enables fine-grained TOA computations. It has gained considerable attention because of its resilience to multi-path effects, and its ability to penetrate obstacles [41]. It has been demonstrated in a controlled indoor environment that cooperative UWB-based localization can resolve locations up to centimeter-level [41]. However, till date, to the best of our knowledge, no commercial COTS devices have built-in UWB radios. Also, implementation of UWB-based schemes require nanosecond level operating-system (OS) interrupts [42], while current generations of smart device OSs are only capable of interrupts in the order of milliseconds [43].

Pedestrian Dead-Reckoning (PDR) [44][45][46] is another emerging area that has gathered research attention due to the proliferation of sensor-rich smart devices. The basic idea of this approach is to fuse *intra-device* information gathered from various sensors on a smart device. Readings from sensors such as accelerometers, and even sound, light, and image sensors [47], can be fused together to directly localize a mobile phone. Otherwise, they can correct displacements or drifts from known locations, either from anchor locations or opportunistic GPS readings. A comprehensive review of this approach can be found in [48]. This approach is interesting and potentially-deployable in current generations of COTS devices. The work described in this dissertation complements the PDR approach by operating at an *inter-device* level, but implementation of aforementioned PDR schemes is out of the scope of this dissertation.

The framework and algorithms proposed in this dissertation complement these localization techniques, as well as any others that might be developed in the future, which offer a

reasonable probability that at least one or several devices in a co-located area will be well-localized.

2.4 Network- and Device-Level Fault Monitoring and Management

Fault management in wireless networks, as a core component under the FCAPS framework, is a fairly-well studied topic. Strong existing contributions address the management of Wi-Fi network faults in particular. In [49], the authors propose the use of large arrays of commodity desktop computers, equipped with Wi-Fi cards or dongles, as enterprise network sensors and monitors. In another study [50], its authors suggest that Wi-Fi clients and access points can be instrumented to become diagnostic agents. These agents can be directed to switch to promiscuous mode to help detect and relay problems.

Cellular networks have conventionally required different, more centralized approaches to fault management. This is due to their large geographical spreads, provisioned QoS, strict centrally-managed infrastructure, and until recently, ‘dumb’ clients, which cannot be easily instrumented to perform any complex tasks [17]. Existing work has a strong focus on detecting network-side faults, down to at most a cell-wide level, by examining key performance indicators (KPIs). In [51][52], the authors provide a brief description of possible network-side faults and symptoms in cellular networks. They then propose a Bayesian inference model to compute the probability of a fault based on observed KPIs. In another recent study [53], its authors propose a two-stage detection-diagnosis model where each monitored KPI is compared against normal ‘profiles’. Deviations are then matched to known root causes. These aforementioned work are useful for back-end diagnosis and after-the-fact root-cause identification. However, they do not provide immediate relief, nor timely acknowledgement of a user’s perceived troubles with his network connection.

The estimation and classification framework proposed in this dissertation is, to the best of our knowledge, the first designed specifically to cooperatively address UE-level faults. It identifies, in a timely manner, on a per-device basis, whether a user's observed performance impairment is an isolated one, or endemic to those in the same vicinity. This complements existing fault management methodologies currently practiced by network operators, by providing critical spatial-temporal context to any customer feedback and complaints. Using this framework, the network operator, as well as participants, can quickly determine whether performance impairments are a result of problems with the network, or just transient, isolated issues that require no further attention.

2.5 Summary of Related Work

We summarize the main findings of the survey as follows.

1. The proliferation of smart devices has enabled a wave of innovative participatory sensing and crowd-sourcing applications that relies on users to contribute sensory information.
2. Deterministic and stochastic tools that model cellular network performance, such as coverage, are useful for aggregate analysis. However, they find limited application in the real-world. Empirical approaches, such as manual drive-testing, is still the main technique used by operators today.
3. Independent third-party projects that aim to map out cellular performance are becoming popular, with interests from regulatory authorities, commercial companies, as well as users themselves. The main drawback of these efforts is that they are unable to capture fine-grained location resolution of measurements. This is either due to

limited manpower (as in the case of [33]), or lack of constraints on subscribers' reported location accuracy (as in the case of [32][34]).

4. Current research efforts in device localization report encouraging results, where devices are able to be localized down in centimeter-level in controlled and limited indoor laboratory settings. However, these are dependent on radio technologies, such as UWB, which to the best of our knowledge is not yet available in commercial off-the-shelf devices. They may also not be available for the foreseeable future. Without specific chipset support, running fine-grained, nanosecond-level Time-of-Arrival-based computations on current generations of devices is not possible.

5. Fault management in cellular networks conventionally differs from that of wired networks. This is because of their large geographical spreads, and, until recently, 'dumb' clients that cannot be instrumented to perform complex tasks. Hence, majority of related work on fault management in cellular networks typically focused on back-end diagnosis and root-cause identification of cell-wide problems. There is little emphasis on providing timely relief to the subscriber whenever any service impairment is perceived.

To address the gaps and drawbacks identified above, this dissertation proposes a comprehensive monitoring framework, which possesses the features listed as follows.

1. The framework relies on subscribers themselves, wherever they are, to perform and collect empirical network measurements on a fine-grained, large-scale basis. This minimizes operator expenditure on manual walk- and drive-testing efforts. Compared to existing approaches that monitors areas on the order of hundreds to thousands of meters, our framework can generate monitoring maps that are accurate *to the order of meters*.

2. This framework advocates short-range, peer-to-peer exchange of measurements using low-power, local-area radio technologies that are pervasive in current generations of smart devices. This allows the preservation of the context of co-location between measurements collected in the same vicinity. Using this context of co-location, complementary algorithms will be proposed in this dissertation to reduce and remove measurements that are poorly-localized. Accurate, high-resolution network monitoring maps can then be constructed. To the best of our knowledge, this is a unique feature that has not been discussed or implemented before, for cellular network monitoring.

3. Through local-area measurement exchange, this framework allows for a basic and immediate level of fault diagnosis for participants. This is achieved through a proposed comprehensive statistical framework. It estimates and classifies, on a per-device basis, whether any perceived network impairment is isolated to that device, or endemic to other devices in the same area. If the former is identified, participants can take remedy steps to alleviate their impairments (e.g. restart their devices). Otherwise, they attain some level of closure and comfort in knowing their impairments are likely the fault of the network, and are shared by others in the vicinity. To the best of our knowledge, we are the first to propose and implement this novel peer-to-peer collaborative application for device-level fault diagnosis in cellular networks.

3. TATTLE - CELLULAR COVERAGE MONITORING THROUGH UE PARTICIPATORY SENSING

The operating environment of cellular networks can be in a constant state of change. Over the course of hours and days, cell loads can change due to diurnal cycles, or events such as spectator sports. Over the course of weeks and months, network topology can change due to the addition of new cells, and resizing of existing cells. Over the course of months and years, changes in the physical operating environment can come about when new physical infrastructure, like buildings, is constructed. Due to this evolution, dynamic parameters (such as radio, antenna, handover and load-balancing) may therefore have to be regularly reconfigured so that cell service is not degraded. However, operators can sometimes struggle to keep pace with the regularly-changing operating landscape. For example, in many space-constrained urban areas, low-lying buildings constantly make way for increasingly taller ones. These new and tall structures cause heavy shadowing and coverage holes.

The operator has to rely on subscriber complaints to discover areas that are poorly served. For the cellular network operator that we interviewed, their lead time between discovering the need for a new base-station, to its installation and commission, is around 6 months. In the meantime, the operator has to reconfigure existing base-stations to alleviate coverage problems, and yet minimize any impact to areas that were previously well-served.

3.1 Coverage Monitoring – Problem Overview

One of the specific problems that the operator which we interviewed put to us is *coverage monitoring (CM)*: *how can they efficiently verify that minimum coverage is met for an area of interest?*

Minimum coverage at any given spatial location is defined by Singapore’s regulatory authority to be a minimum threshold percentage of Received Signal Code Power (RSCP) samples (collected at that location) which exceed -100 dBm. The RSCP is the received code power from the downlink Common Pilot Channel (CPICH), broadcast as a “beacon” from base-stations [54]. The regulatory authority in Singapore mandates that at least 85% and 99% of RSCP samples, collected within any indoor and outdoor area respectively, must exceed -100 dBm. If areas which fail these requirements are overlooked, the operator faces penalties by the regulator, and stands to lose subscribers to other competitors.

The operator currently follows a manual approach. Areas with poor coverage are first identified through subscriber feedback. Coverage is then appraised by manual walk-tests. If minimum coverage is not attained, parameters (such as pan, tilt and power) are iteratively tweaked. After each iteration, walk-tests will be conducted to re-appraise coverage. This is repeated until minimum coverage is obtained. Furthermore, areas that were previously amply-covered should not be significantly impacted after this tuning. Such an approach is labor-intensive, expensive and limited in scale. It only provides a non-real-time and non-continuous snap-shot of coverage. The process has to be repeated whenever the operating environment evolves, especially over long periods due to aforementioned reasons.

3.2 Contributions

In this chapter, we introduce Tattle, a distributed, low-cost and comprehensive RSCP monitoring framework that addresses the **CM** problem in a scalable and real-time manner.

Tattle has 3 key components, namely:

1. the local exchange of RSCP measurements between devices, and subsequently the uploading of co-located readings to the network,
2. the real-time pre-processing of co-located readings at the network side to discard erroneously-localized measurements, and,
3. the visualization of coverage based on collected RSCP measurements in specific regions-of-interest.

Tattle is designed to minimize operator expenses and labor, and yet monitor coverage in real-time on large geographical scales with good fidelity. It does this by removing erroneously-localized measurements. It enables operators to effectively appraise coverage without conducting expensive walk-/drive-tests. Operators can also proactively identify and mitigate spots with poor coverage in a timely manner, instead of reactively acting only upon subscriber complaints.

We demonstrate through experiments that current-generation commodity handsets can exhibit wild variations in terms of reported GPS location under urban conditions. A pair of devices that are physically placed 1 m apart can report erroneous locations of up to kilometers of pair-wise distance. We are the first to demonstrate that using local-area measurement exchange and our proposed U-CURE clustering algorithm, co-located mobile devices can reduce their errors of their reported pair-wise distances by up to 40%.

To the best of our knowledge, we are the first to implement, validate and demonstrate the achievable, real-world benefits of such a complete monitoring framework when applied to cellular networks. We use real-world, COTS devices to collect more than 3.78 million readings of RSCP using Tattle in an actual production cellular network. We show its efficacy in performing **CM** by presenting high-resolution results of actual coverage conditions in selected areas of Singapore.

3.3 Chapter Organization

In Section 3.4, we describe our proposed Tattle framework, and give details on the Tattle app and prototype. In Section 3.5, we discuss how readings can be pre-processed based on co-location to remove samples with location errors. In Section 3.6, we describe our measurement collection procedures, and evaluate Tattle in terms of the localization fidelity of resulting measurements. We collect over 3.78 million RSCP measurements, and present real-world mean RSCP coverage maps and RSCP cumulative distribution functions (CDFs) of various areas in Singapore. Finally, we give a summary of this chapter in Section 3.7.

3.4 Tattle – Monitoring Through Participation

We propose Tattle, a distributed monitoring framework that is scalable, minimizes manual labor and operator expense on drive-tests. It monitors real-time coverage on a large geographical scale with good measurement location fidelity, and requires minimal involvement of subscribers (other than simply running a background app on their smart devices).

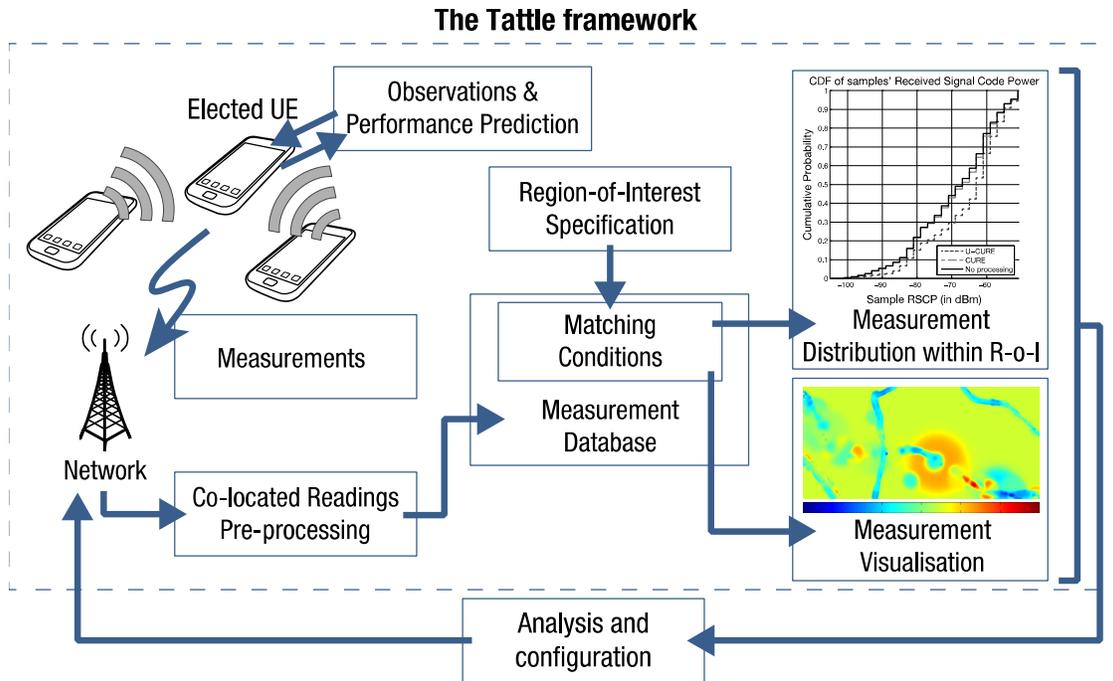


Figure 4: A diagrammatic description of Tattle, a distributed, low-cost and comprehensive cellular network measurement collection and processing framework. In this chapter, we exemplify Tattle by leveraging on participating UEs to report on network coverage in real-time.

Figure 4 illustrates the overall system architecture of the Tattle framework. In the context of network coverage monitoring, there are 3 key components, namely:

1. the local exchange of RSCP measurements between devices, and uploading of co-located readings to the network,
2. the pre-processing of co-located readings to discard erroneously-localized measurements (either at the backend or at the UE-level), and,
3. the visualization of coverage based on collected measurements in specific regions-of-interest.

We will discuss each of the above components in the rest of this chapter.

3.4.1 Benefits of Low-Power Measurement Exchange

Subscribers are the best monitors of coverage wherever they are and wherever they require service. In our framework, the UEs form distributed mobile sensor networks to monitor the RSCP experienced by each UE. Instead of purely relying on the cellular service to upload measurement reports, we advocate using low-power local communications (such as Bluetooth Low Energy [55], or WiFi Direct [56]) to exchange RSCP measurements between UEs. This preserves the co-locality of measurements and conserves UE power.

Conventionally, when a UE reports its measurement, together with its coarse location information, its measurement is taken in isolation. Two reports from separate UEs cannot be corroborated unless they have very accurate location fixes. However, if readings are exchanged through local communication, the principle of co-locality is preserved. This is because a UE that overhears a report from another UE can be sure that the transmitter is within range (depending on the interface used). In our indoor range test experiment, we find that the devices used in our experiments (specifically the Asus Nexus 7's, and the Samsung Galaxy Tab 2 7.0's) can reliably receive each other's WiFi Direct broadcasts within 30 meters.

In Section 3.5.1, we will use this communication range as an evaluation condition to determine whether an assessed reading should be admitted into the group of accepted readings. This is based on the knowledge that these readings were co-located. We can reject and discard a measurement if the reported GPS-location's distance to the admitted group is beyond local transmission range. *This is a key feature of Tattle.* Naïve reporting of measurements in isolation lacks the context of co-location, and the network has to accept all reports in good faith. This includes those that have GPS locations which are egregiously wrong. Outdoor communication range may be higher, but we chose to use the

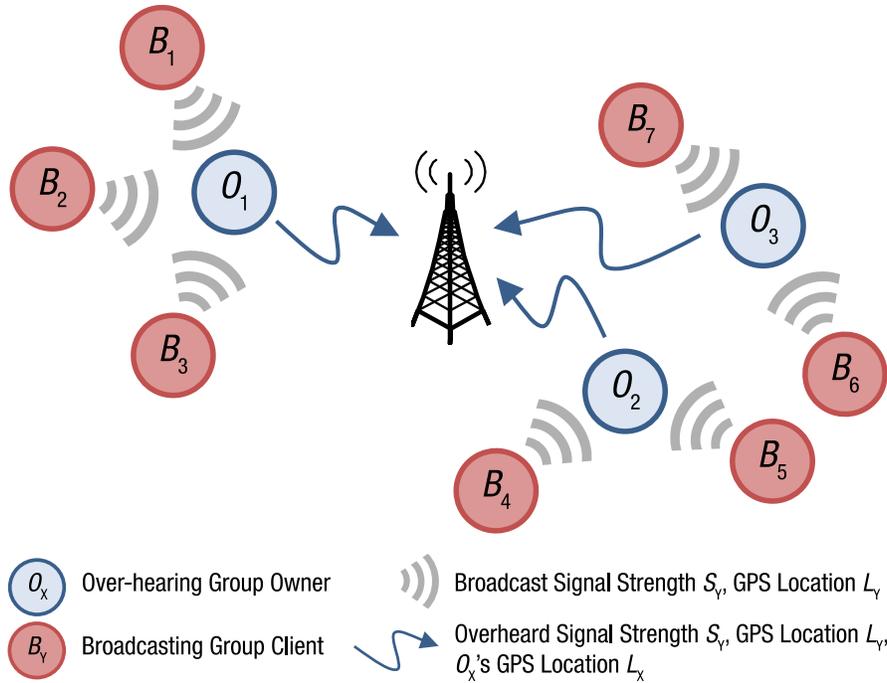


Figure 5: Participating users form local-area communication networks using interfaces such as WiFi Direct, to exchange signal strength readings. Elected Group Owners forward overheard readings to the network.

observed indoor range of 30 meters as a conservative upper limit. This keeps the group of admitted measurements tightly clustered in space. We stress that the emphasis is on *keeping measurements that are likely to be well-localized, rather than admitting as many measurements as possible.*

The aforementioned local-area radio technologies, such as Bluetooth and WiFi Direct, are pervasive in smart devices. Hence, they become ideal candidates for local-area communications [12]-[16]. The industry is also starting to see value in allowing mobile devices to directly communicate with each other through the cellular interface. Promising research and development efforts, such as in [57], are currently underway. If none of these interfaces are available, we then rely on the cellular service as a fallback. **Figure 5** shows how the measurements can be communicated from UEs back to the network.

3.4.2 The Tattle App – Description and Operation

The eponymously-named Tattle app is a simple prototype to demonstrate the efficacy of our proposed coverage monitoring framework. It exploits WiFi Direct as a local communication interface for UEs to exchange RSCP measurements. The app can be easily extended to use other local communication interfaces, such as Bluetooth. The app has the ability to scan for and form an ad-hoc network with other participating devices within WiFi Direct range. WiFi Direct is an attractive choice because it can work in tandem with a user's existing WiFi association to any WiFi Access Point. Once a device-pair creates a network, one UE will be nominated transparently by the WiFi Direct protocol to be the group owner. One limitation of WiFi Direct is that the setup time for connections is in the order of seconds. However, we believe that further revisions and improvements to WiFi Direct, and other peer-to-peer protocols, such as BLE, will in the future enable seamless peer-to-peer on-demand connectivity.

The group owner assigns an Internet Protocol (IP) address to each client that subsequently joins the network. Every time a new device joins the WiFi Direct network, the group owner broadcasts an **Address List Update** to notify all existing devices of the incoming device's IP address. Multi-hop transmission is not yet supported by WiFi Direct, but it is not essential for our purposes. This is because a client is only admitted to the WiFi Direct group when it is one hop from the group owner.

Upon joining a network, each device periodically broadcasts its GPS location, current network type (UMTS/ HSPA/HSDPA/HSDPA+), and the signal reading associated with its network type. In our prototype, we configured our broadcast period to be 1 second. In an active deployment, the sensing and broadcasting interval can be extended to automatically adapt to remaining battery power, current signal conditions, etc. It may

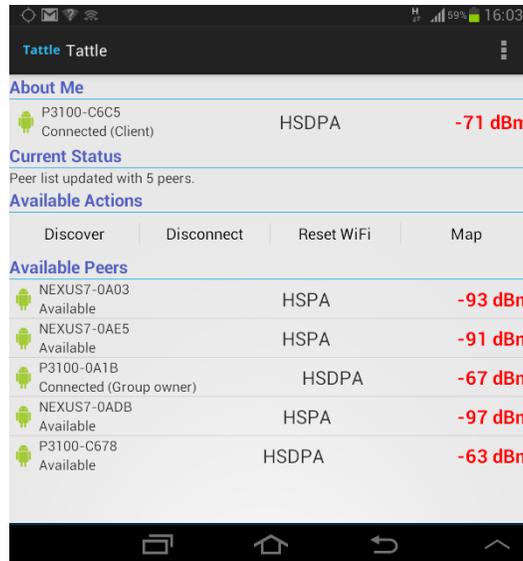


Figure 6: Screen capture of the Tattle app in operation. Nearby devices will be invited to join the network, and periodically exchange GPS location and signal readings.

also, for example, dynamically adapt to the number of participating UEs that were last overheard. So, the broadcast periodicity may be decreased whenever a large number of co-located participating UEs are overhead. Whenever a signal reading is overheard, the overhearing device will log the broadcast, and update the Tattle app GUI to reflect the reported signal of the corresponding device. A simple round-robin approach is used to determine which UE in the network reports the next batch of overheard signal readings to the measurement database. All uploads will be tagged with the timestamp of reception, and appended with the reporter's current GPS location.

We stress that our framework readily admits the use of reporting schemes (other than round-robin) which may robustly adapt to dynamic conditions, such as remaining device power, signal conditions of individual UEs, etc. However, this dissertation focuses on the Tattle framework and system, hence the investigation of other detailed reporting schemes is out of the scope of this dissertation.

Figure 6 demonstrates the Tattle app in action, with all 6 devices (as shown in the screenshot) connected to the same WiFi Direct network. In our experiments, all devices operate on one common provider's network. The Tattle framework can also be adopted by third-party data collection entities that are not affiliated with the cellular operators. In this case, devices on different cellular networks can exchange measurements of their own network's coverage. Then, the data collection entities collect uploaded reports at their backend servers, which are independent of any cellular operator. They can subsequently provide operators and subscribers with coverage information as a service. This can also boost the number of participating subscribers if the use of Tattle is not limited to subscribers of any particular operator.

3.4.3 Motivation for Users

Oftentimes, when users experience poor or no signal coverage (as evinced by the display of 'signal bars' on most phones), they often ask: *is this phenomenon observed by others?* Knowing that others are experiencing similar coverage conditions does not necessarily help a user. It however provides some level of comfort in knowing that the network, and not the user's equipment, is likely the source of the problem. On the other hand, if a user knows that most other subscribers are getting adequate signal coverage while his own signal readings are poor to non-existent, he can take some limited steps to alleviate the situation (e.g. restart his device, check his device settings, etc.). Through local-exchange of measurements, the user can gain a new level of insight into the service. The same kind of measurement reporting and retrieval can conceivably be implemented entirely based on the cellular interface instead of using local communication. However, there are several drawbacks to this approach, which we refer to as *naïve reporting*.

Firstly, transmitting measurements to the network over the long-distance cellular link consumes more power than sending them to co-located peers, using short-range local-area interfaces such as WiFi Direct. In poorly-served areas, this problem is exacerbated because devices have to transmit to the cellular base-station at higher powers to overcome poor channel conditions. Secondly, pulling real-time measurements of nearby users from the network consumes precious downlink capacity. This is made worse in congested and poorly-served areas, whereby a multitude of users who want real-time signal information actually contribute further to congestion. Thirdly, by relying on the cellular network, we lose the knowledge of co-location. We show in Section 3.4.4 that reported GPS locations of devices can sometimes be as far as several kilometers away from their actual location, which can result in numerous misleading readings.

Compared to naïve reporting, short-range exchange consumes less power and does not consume downlink capacity. It enables users to obtain real-time signal information from other users that are guaranteed to be within local communication range.

To further incentivize subscribers to participate, recent studies have focused on monetary-based reward schemes that provide payouts to participants based on various criterions. These may be their current locations [58], or their sensing contributions [59]-[62]. The operator can readily leverage on these mechanisms to instate a reward system. They can either providing monetary compensation to participating UEs (taken from a small percentage of the operating cost savings gained by avoiding the need for walk-/drive-tests), or even using subscription rebates, loyalty points, or handset discounts to incentivize participants.

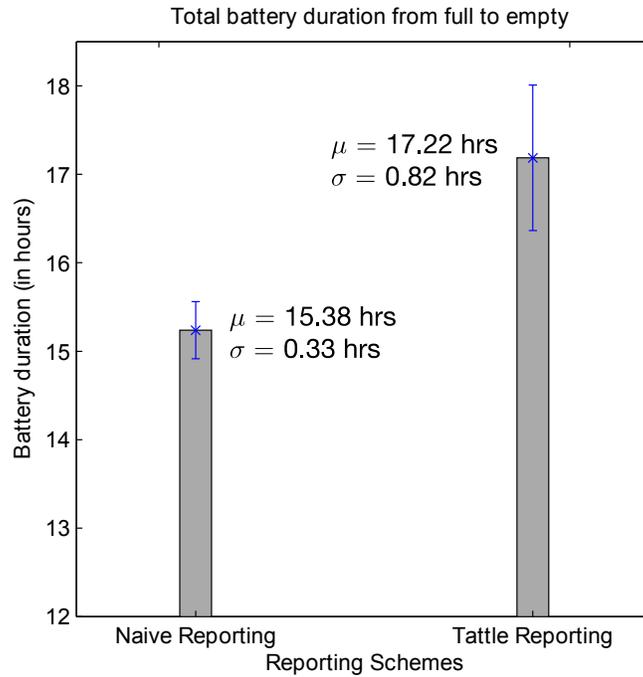


Figure 7: Battery drain test demonstrating the battery consumption of the naïve reporting method vs. the Tattle prototype round-robin approach.

3.4.4 Tattle – of Power and Location Uncertainty

In **Figure 7**, we demonstrate results of battery drain tests conducted with 3 Samsung GT-P3100s. All 3 were first set to perform naïve reporting where RSCP readings and GPS locations were uploaded every second to a remote server, in a JavaScript Object Notation format using the HTTP POST method. The GT-P3100s’ batteries were fully charged at the commencement of the experiment. The same experiment was repeated using Tattle’s RSCP local exchange and round-robin network reporting approach. Each GT-P3100 broadcasts its RSCP reading and GPS location every second. Using round-robin, each device reported all overheard measurements, including its own, for consecutively 20 seconds in every 60 second window. These two experiments were each repeated thrice. With naïve reporting, the devices lasted an average of 15.38 hours with a standard

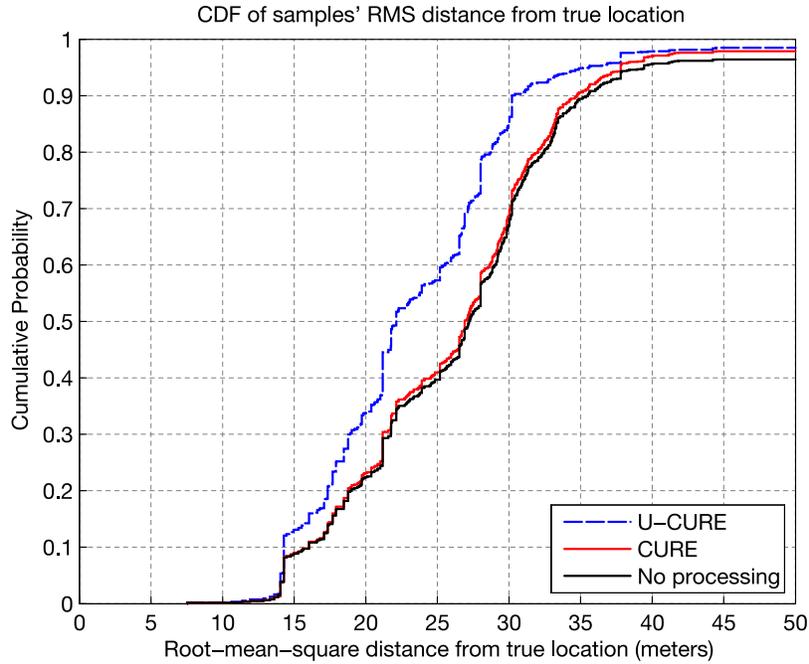


Figure 8: CDF plot of the root-mean-square distances of 18,191 static sample points from a known true location.

deviation of 0.33 hours. With Tattle, the devices drained completely after a mean of 17.22 hours with a standard deviation of 0.82 hours, lasting 12% longer on average.

We stress that further power savings are possible if more robust schemes other than round-robin are used. The choice of the latter is simply for ease of prototyping. The framework readily admits other reporting approaches.

We then conducted a ground-truth experiment at a known location to investigate the performance of GPS accuracy in an outdoor urban environment populated with low-lying buildings. We co-located 6 static devices, 3 of which are GT-P3100s and the other 3 are Asus Nexus 7s. The first several minutes of the experimental data were discarded in order to allow each device sufficient time to obtain a coherent GPS signal. In total, 18,191 samples were collected. 50% of these had mean locations more than 27.22 m off from the true location, while 2.79% had mean locations that were 2,877.69 m away from the known

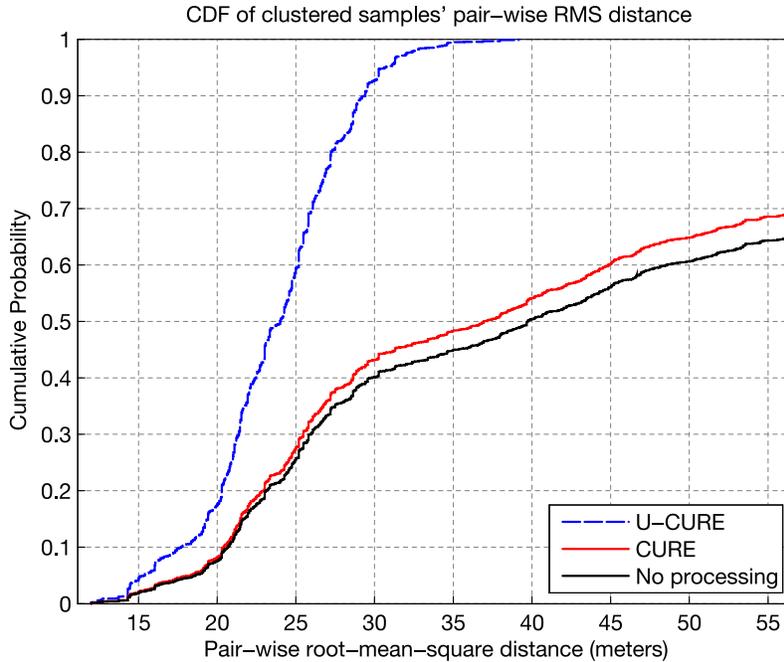


Figure 9: CDF plot of the pair-wise root-mean-square distance of sample locations known to be co-located from local-communication exchange.

spot. The maximum observed discrepancy between a sample’s mean location and the known spot was 2,996.07 m. Only 22.42% of measurements were less than 20 m distance from the true spot.

Figure 8 shows the cumulative distribution function of the root-mean-square (rms) distances between the sample points and the known spot, before and after processing. In Section 3.5, we detail how the rms distance is evaluated. We then introduce the CURE algorithm, as well as our extended U-CURE algorithm, which we used to pre-process the data in order to discard readings that are wrongly localized. After processing with U-CURE, the number of readings with rms location errors below 30 m saw a 17.85% improvement, compared to that of a naïve reporting approach, and a 16.19% improvement over CURE.

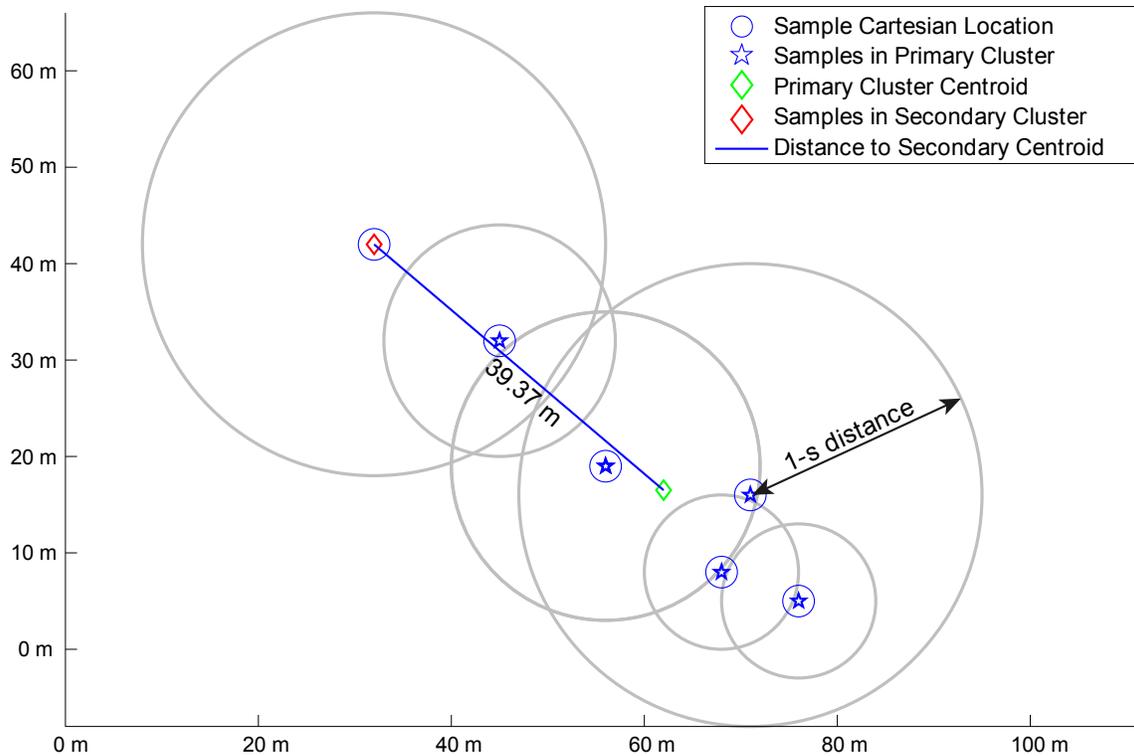


Figure 10: A snapshot of device reports taken from our urban outdoor experiments. All 6 devices were actually co-located in a fitted box, each spaced apart by Styrofoam paddings. The spread of reported locations demonstrate the problem of inaccurate location reports. We will further investigate deviations from location ground truths in Chapter 4.

Figure 9 demonstrates the cumulative distribution of pair-wise rms distance between each sample-pair in each co-located batch of readings. Using the naïve reporting method, all context of co-location is lost. However, using U-CURE, we are able to exploit the knowledge of measurement co-location to discard samples that are likely to be erroneously-localized. All samples after U-CURE processing had less than 39.15 m rms pair-wise distances. The naïve reporting method had pair-wise rms distances as high as 3,032.22 m, even though they were physically juxtaposed.

Figure 10 is an illustrative temporal snapshot of the reported spatial locations of each co-located device captured sometime into the experiment. It clearly demonstrates the

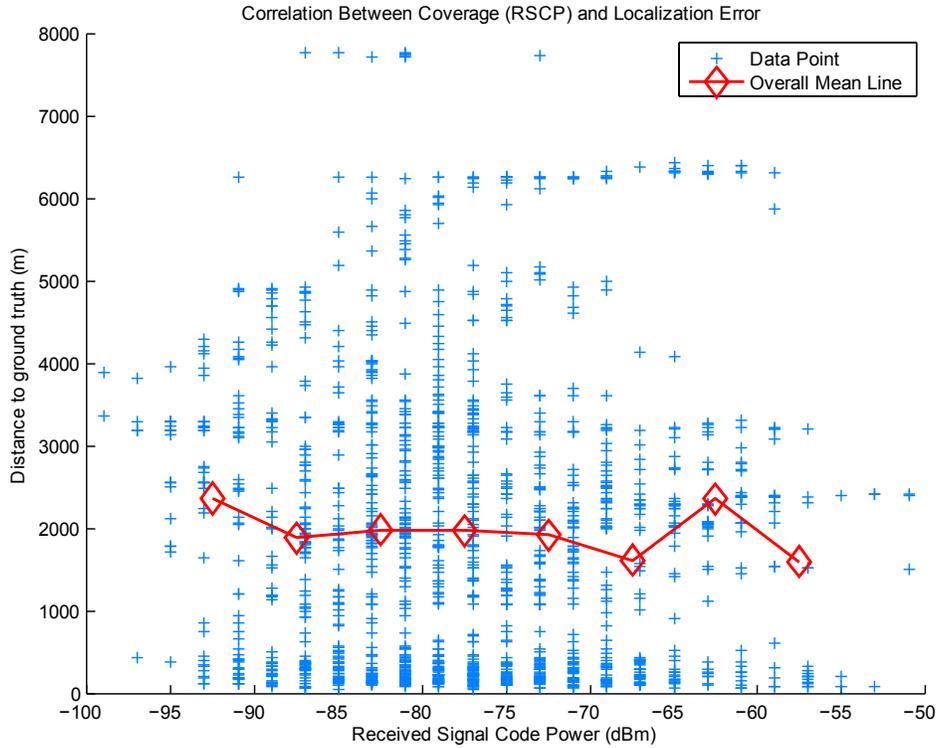


Figure 11: A plot of 1,628 Received Signal Code Power measurements and their corresponding localization errors in a challenging high-speed mobility condition, collected by 3 sets of GT-P3100s. The Pearson correlation coefficient between RSCP and localization error is -0.0441 , suggesting little to no correlation between the two variables.

problem of location uncertainty in urban environments, even when outdoors. Though the devices were physically juxtaposed, the maximum reported pair-wise distance between device location means actually exceeds 50 meters.

Next, we investigate whether there exists any meaningful correlation between coverage conditions and the magnitude of localization errors, as reported by our devices. It is important to establish this because any processing of measurements based on their reported locations must not skew the final computed result of coverage. If devices that are in areas with poor signal coverage tend to have higher localization errors, then any bias against those RSCP measurements (for example, if we were to discard these

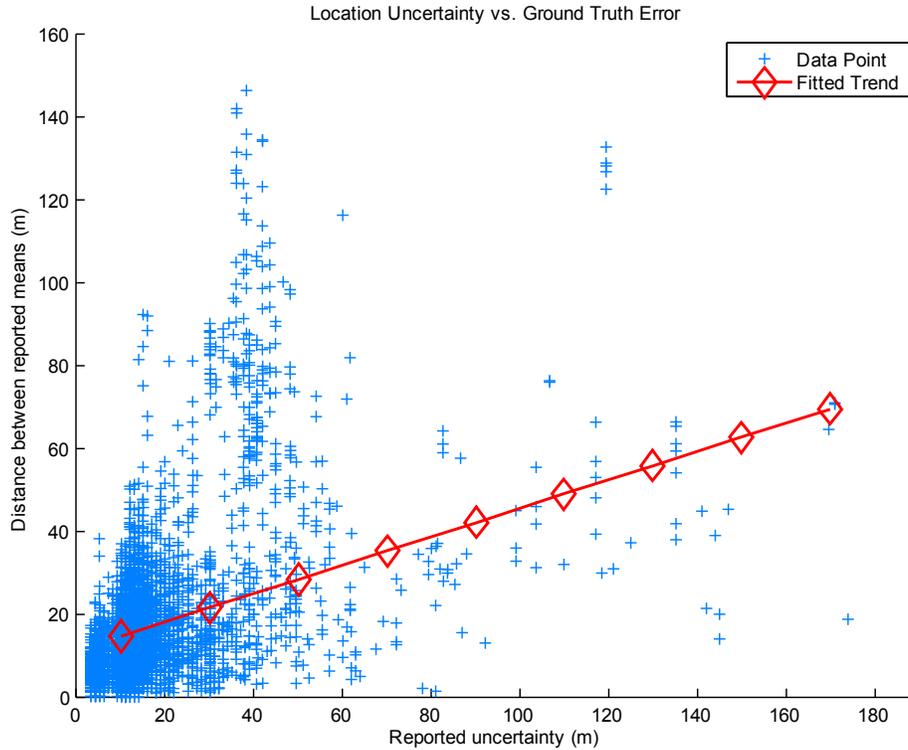


Figure 12: A plot of 9,226 location measurements and their corresponding localization errors, collected by 4 sets of SM-T325s. The Pearson correlation coefficient between the reported location uncertainties and actual error distances between the reported location means, and the actual ground truth, is 0.6344, suggesting a strong correlation between the two variables.

measurements based on their location errors) will result in an artificially-positive representation of coverage conditions. We demonstrate the results of this investigation in **Figure 11**. Using three units of GT-P3100s, we collected 1,628 RSCP measurements in a challenging, high-speed mobility setting to elicit any relationship between coverage conditions and the magnitude of localization errors. As shown in the figure, the mean trend-line stays relatively flat. The quantitative Pearson correlation coefficient turns out to be -0.0441, suggesting very little to no correlation between RSCP values and the magnitude of localization errors.

These experiments serve to establish the fact that, for commercial off-the-shelf (COTS) devices that are very closely co-located (where inter-pair distance is less than a meter), their reported locations may be upwards of tens of meters apart, and sometimes more. We will investigate in further detail the localization performances of COTS devices in Chapter 4. But now, we continue the discussion on the systems aspect of Tattle and its application to the **CM** problem in the rest of this chapter.

Finally, we investigate whether the location uncertainties reported by COTS devices directly correlates with the actual error between their reported locations and their ground truth locations. **Figure 12** illustrates the result of this investigate. Using four units of SM-T325s, we collected 9,226 location measurements reported by the devices, and their actual ground-truth locations as measured by a commercial, aviation-grade GPS. The fitted trend clearly suggests an obvious increase in the ground-truth error as the reported location uncertainty increases. The Pearson correlation coefficient turns out to be 0.6344, which further supports this observation.

3.4.5 A Note on Security, Privacy and Trust

In this dissertation, we focus on the systems aspect of our Tattle measurement collection and monitoring framework. Hence we omit, for brevity, comprehensive security, privacy and trust mechanisms in our Tattle prototype. However, we note the importance of having these mechanisms in an active, full-scale deployment, and briefly discuss some existing work in this area.

In any general participatory sensing system, security [11], privacy [63] and trust [64][65] of both the system and its participants need to be considered. For Tattle, there are two primary concerns that we note:

1. How does the operator increase the likelihood that the measurements overheard and reported to the measurement database are what was actually observed by the participating UEs, and not fabricated by malicious participants?
2. How can participating subscribers be assured of their anonymity and privacy, while retaining their trustworthiness and ability to be recognized and rewarded for their contributions?

In the context of the first concern, participants can lie about their measurement values, their purported locations, or both. In the case of fabricated measurement values, [66] suggests the possibility of using hardware-based solutions for trusted computing, such as Trusted Platform Modules (TPMs) [67].

One critical feature provided by this hardware-based solution is that of remote attestation. It serves to verify that the software (in this case, the Tattle app running on the UE, as well as the OS-level API calls to retrieve GPS and network measurements) has not been maliciously modified. While TPMs exist primarily within desktop computers, there are on-going efforts that seek to implement the mobile-equivalent of TPMs on consumer devices [68]. Mechanisms to prevent intentional physical damage to disrupt GPS, sensors, and radio functionality are beyond the scope of this discussion. Some participating UEs (whose measurements were overheard by others) may deliberately lie about their purported locations. Tattle inherently considers this by discarding their measurements should it be beyond possible overhearing range. We will explain this in Section 3.5.

As for the second concern, the authors in [69] propose a comprehensive reputation and trust framework to address the “trust without identity” problem. Using this framework in the context of Tattle, we can decouple the participant from the device (and hence from the network-level parameters, such as IP addresses, which can be used to identify the

subscriber) by allowing participants to register user accounts. Any reputation, or rewards, will be attributed to the account instead of the device. The framework also decouples the account from its actual measurements reported to the network. The system then rewards individual measurements, as opposed to accounts, by sending reward coupons which can only be redeemed by the account that sent those measurements. In this case, the identity, privacy and reputation of the participant can be preserved.

3.5 Hierarchical Clustering With Uncertainty

In a naïve reporting approach, the operator has to accept each report in good faith. However, the use of local measurement exchange guarantees that a device-pair is *surely at least within local communication range*. This context is important because it allows us to design techniques to discard patently incorrect reports that can otherwise affect the fidelity of the results. We apply pre-processing to co-located readings as the second key component of the Tattle framework.

The goal is to identify and discard samples which have GPS locations that are likely to be wrong, rather than to improve on the accuracy of localization of individual samples. The key which allows us to achieve this goal is through the use of local communications. To this end, we propose U-CURE (uncertain clustering using representative points), an extension of the CURE clustering algorithm [10]. We chose to extend the original CURE algorithm because:

1. it is robust against outliers, and,
2. it identifies clusters that are non-spherical in shape.

The latter feature is especially desirable because the spatial distribution of participating devices does not conform to any particular shape. We stress that our proposed framework

```

function u_cure(S, k, max_dist)
% S is an input of  $N \times 3$  matrix of  $N$  rows of [ $x$ ,  $y$ , uncertainty] entries
% k is the desired number of clusters
% max_dist is the maximum expected distance allowed between clusters
1. Initialize C, a sorted array of clusters, each cluster has:
    a. list of points inside the cluster,
    b. a pointer to its nearest cluster,
    c. expected distance to nearest cluster,
    d. centroid in [ $x$ ,  $y$ ], simply the mean of all  $x$ 's and all  $y$ 's of points in C,
    e. a list of representative points for this cluster
2. Sort C according to ascending distances to nearest cluster
3. Initialize r, the number of representative points that represents each cluster
4. While length(C) > k,
    if distance of cluster  $C\{1\}$  to the next nearest cluster is > max_dist,
        break;
    else
        merge top two clusters of C;
    endwhile

function merge_clusters(C, r)
1. Sort C according to ascending distances to nearest cluster
2. Merge the first two clusters of C
3. Evaluate centroid of the newly merged cluster
4. Find the representative points of the newly merged cluster
5. Re-compute expected distance between every cluster-pair and re-sort C

function find_rep_points(p, centroid, r)
% p is the complete set of points in the cluster that we want representative points for
1. If length(p) ≤ r
    return p,
else
    set the first representative point as the point furthest from centroid
2. While length(rep_points) < r,
    a. find the point from p whose min. distance to other rep_points is max
    b. add that point to rep_points and clear it from p
endwhile
return rep_points

```

Figure 13: The U-CURE algorithm.

readily admits other choices of pre-processing algorithms. The choice of U-CURE is intended to be instructive rather than exclusive. We make 2 important modifications to the CURE algorithm to handle the reported uncertainty in a measurement's location, as well as to discard samples with likely incorrect locations. We refer to the modified algorithm as U-CURE, and a sketch of the U-CURE algorithm is given in **Figure 13**.

3.5.1 U-CURE: Extending the Original CURE Algorithm

We first briefly describe the CURE algorithm. One can refer to [10] for more details. CURE considers each data point as a point source in Cartesian space. The distance between two clusters A and B is the minimum distance between all the possible representative point-pairs. It is given by:

$$d(A, B) = \min(d(r_A, r_B)) \quad \forall r_A \in R_A, r_B \in R_B, \quad (1)$$

where $d(.,.)$ is a measure of Euclidean distance between the two inputs, r_I is a representative point in the set of representative points R_I of cluster I , where $I \in (A, B)$. It works as follows:

1. Start by considering every point as a separate cluster, and each point is its own cluster's representative point.
2. Find the two clusters that are closest in distance, and merge them.
3. Find the representative points of the newly merged cluster, where the first point chosen is furthest from the centroid. Each subsequent point is chosen sequentially such that its minimum distances to all previous representative points is maximum, until a desired number of representative points is reached.
4. Repeat Step 2 and Step 3 until the desired number of clusters remain.

At the conclusion of CURE, we pick the largest cluster as the set of admitted points, and discard the rest. We extend CURE in 2 important ways to further improve clustering accuracy. First, we implement a stopping condition: when the closest two clusters have a root-mean-square (rms) distance larger than the maximum range of the local communications interface, they should not be merged and the algorithm should stop. This corresponds to the *max_dist* condition in the algorithm. Since we are interested in the cluster with the largest number of sample points most closely located together, we set the desired cluster parameter k as 2. This is equivalent to classifying samples into either the primary cluster (which we are admit), or the secondary, out-lying cluster (which we discard). The *max_dist* condition then terminates U-CURE clustering early if clusters are too far apart to be co-located. In doing this, we exploit the knowledge of co-location garnered through local measurement exchange. In the next section, we describe the second extension to the original CURE algorithm.

3.5.2 Considering Uncertainty in U-CURE

Unlike the assumption of location certainty in CURE, GPS locations are expressed as a 3-tuple of latitude, longitude, and uncertainty [70][71]. This uncertainty is expressed as a σ value in meters, and modeled as a 2D normal distribution with the mean location given by the latitude and longitude. Hence, the ‘true’ location is said to fall within a radial distance of σ meters from the mean location with 68.2% probability.

We convert reported latitudes and longitudes into Cartesian (x, y) space using the Map Grid Reference System (MGRS) [72] so that distances are easily evaluated. In order to extend CURE to take into account the uncertainty of sample locations, we consider the *expected square-distance* between uncertain location points, instead of just the distance between their means. Hence, the distance metric becomes the following:

$$d'(A, B) = \min(d'(r_A, r_B)) \quad \forall r_A \in R_A, r_B \in R_B, \quad (2)$$

where $d'(\cdot, \cdot)$ is a measure of *mean square distance* between the two uncertain location points. Because the measure of distance on the \mathbb{R}^2 Cartesian plane is real and non-negative, the order statistics after squaring a distance measure is preserved. We formalize the evaluation of the expected square-distance between two uncertain location points as follows.

Lemma: The expected square-distance between two samples $U = (X_U, Y_U)$ and $V = (X_V, Y_V)$, centered in Cartesian space (X_U, Y_U) and (X_V, Y_V) respectively, where $X_U \sim \mathcal{N}(x_U, \sigma_U^2)$, $Y_U \sim \mathcal{N}(y_U, \sigma_U^2)$, and $X_V \sim \mathcal{N}(x_V, \sigma_V^2)$ as well as $Y_V \sim \mathcal{N}(y_V, \sigma_V^2)$, is equal to $2(\sigma_U^2 + \sigma_V^2) + (x_U - x_V)^2 + (y_U - y_V)^2$.

Proof: We construct two random variables (R.V.) X and Y as follows,

$$X = X_U - X_V, Y = Y_U - Y_V. \quad (3)$$

Since the difference of Gaussian R.V.s is also normally distributed, we know that,

$$X \sim \mathcal{N}(x_U - x_V, \sigma_X^2 = \sigma_U^2 + \sigma_V^2), Y \sim \mathcal{N}(y_U - y_V, \sigma_Y^2 = \sigma_U^2 + \sigma_V^2). \quad (4)$$

Next, we construct a R.V. D representing the square-distance between samples U and V , where,

$$D = X^2 + Y^2. \quad (5)$$

Now, we can make use of the non-central χ^2 distribution and its properties to evaluate $\mathbf{E}(D)$. Let D' be a R.V. of the following,

$$D' = \left(\frac{X}{\sigma_X}\right)^2 + \left(\frac{Y}{\sigma_Y}\right)^2 = \frac{1}{\sigma_U^2 + \sigma_V^2} (X^2 + Y^2). \quad (6)$$

Since X and Y are independent and normally distributed, the sum of their squares normalized by their variances is represented by the non-central χ^2 distribution where,

$$D' \sim \chi^2 \left(k = 2, \lambda = \left(\frac{\mu_X}{\sigma_X} \right)^2 + \left(\frac{\mu_Y}{\sigma_Y} \right)^2 \right), \mathbf{E}(D') = k + \lambda. \quad (7)$$

Applying the linearity of expectation on Equation (6) yields,

$$\mathbf{E}(D') = \frac{1}{\sigma_U^2 + \sigma_V^2} \mathbf{E}(X^2 + Y^2) = \frac{1}{\sigma_U^2 + \sigma_V^2} \mathbf{E}(D). \quad (8)$$

Finally, combining Equations (7) and (8), we get,

$$\blacksquare \quad \mathbf{E}(D) = 2(\sigma_U^2 + \sigma_V^2) + (x_U - x_V)^2 + (y_U - y_V)^2. \quad (9)$$

This lemma is an important one because it allows us to easily evaluate the numerical result of the expected square-distance between any two samples without having to evaluate any computationally-expensive integrals. This makes U-CURE more efficient in terms of computation. By taking into consideration location uncertainty and measurement co-location, U-CURE enables the pre-processing of co-located readings to discard samples that are likely to be mis-localized. Finally, **Figure 10** illustrates a real-world example of a spatial-temporal snapshot of device measurement locations taken from our experiments. It demarcates the primary, accepted cluster of measurements after pre-processing it with U-CURE. The secondary, rejected measurement is also shown.

3.6 Measurement Collection, Processing, and Signal Coverage Representation with U-CURE

In this section, we describe the third key component of Tattle: the visualization of coverage based on collected RSCP measurements in specific regions-of-interest. We conducted extensive RSCP measurement collection over the course of more than 4 weeks, gathering over 3.78 million samples. 6 tablets were used in our experiments, namely 3 GT-P3100s, and 3 Asus Nexus 7s. All 6 tablets were always co-located and maintained WiFi Direct connections to one another. Collection of data points was done throughout the day, in all kinds of environments.

3.6.1 Mean Coverage Visualization

In **Figure 14**, **Figure 15** and **Figure 16**, we first illustrate the efficacy of Tattle in the mass-collection of data points, particularly on our local public rail system and roads. The plots of mean RSCP maps should be of particular interest to cellular operators, who will find difficulty in achieving this scale of sampling by performing walk-/drive-tests. Our framework enables this by allowing participating UEs to undertake the task of coverage monitoring. This map is obtained by:

1. taking every sample point and weighing its RSCP value with a two-dimensional Gaussian filter, centered at the sample's reported mean, with sigma value equal to the sample's σ uncertainty, and,
2. summing up the resulting two-dimensional matrix generated for each measurement, and,
3. computing the weighted average for every 1 meter by 1 meter bin.

The corresponding scatter plots of measurements for naïve reporting, CURE and U-CURE are juxtaposed. In reality, all of the sample points should lie on the main veins, which correspond to the high-speed rail tracks and roads. In both the naïve reporting and CURE approaches, we see the debilitating effects of stray signal points with especially large uncertainties in the highlighted areas. They pollute the overall signal map with large blobs of measurements (due to their large σ values) that extend way beyond the main veins. These cause measurements with high location accuracies (and correspondingly low σ values) to be averaged out. We term this as the 'smudging' effect. Interesting features, which are specific areas with either excellent or unacceptable coverage are difficult to spot.

This poses a problem for operators as they require a high degree of fidelity to identify specific problem areas. In contrast, U-CURE processing removed most of the mis-localized points and reveals a significant coverage hole demarcated at the bottom right. Localized features are more accentuated with much less ‘smudging’ blobs.

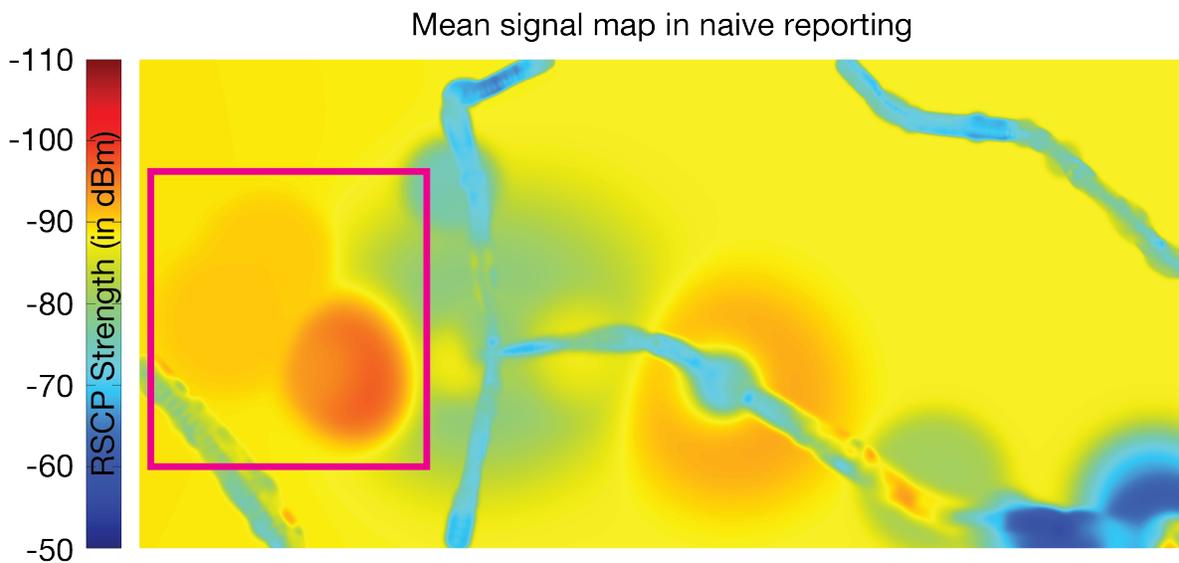
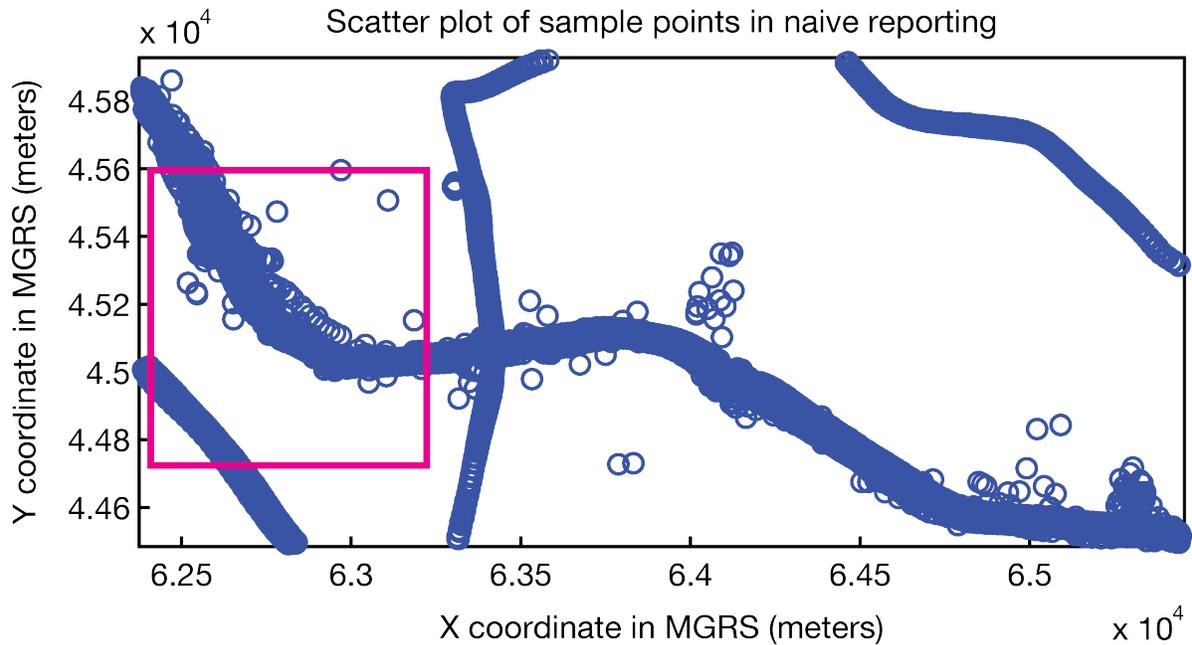


Figure 14: Mean coverage map obtained using naïve reporting, with 39,947 sample points in the region (1.307, 103.791) to (1.320, 103.763), in given as latitude and longitude coordinates. The size of the area is approximately 1.5 km by 3.2 km. The scatter plot of sample points' mean positions is given at the top. The bottom represents the mean signal map (where orange to red regions represent areas of barely- to unacceptable coverage, i.e. -90 dBm to -110 dBm, and blue to yellow regions represent regions of excellent- to barely acceptable coverage -50 dBm to -90 dBm). Areas of interest are marked out with rectangles.

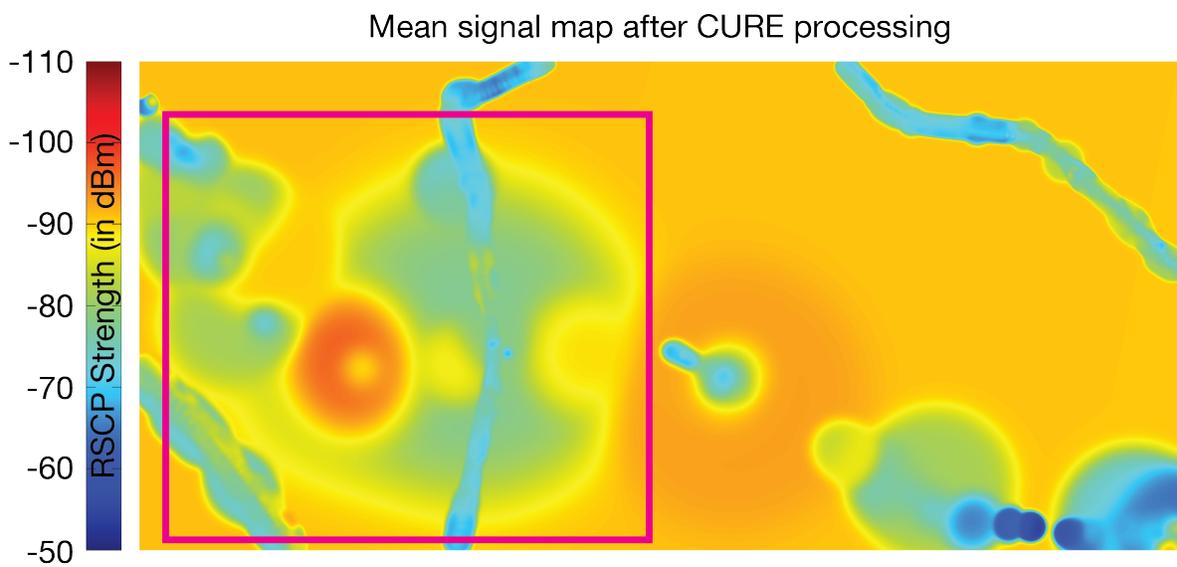
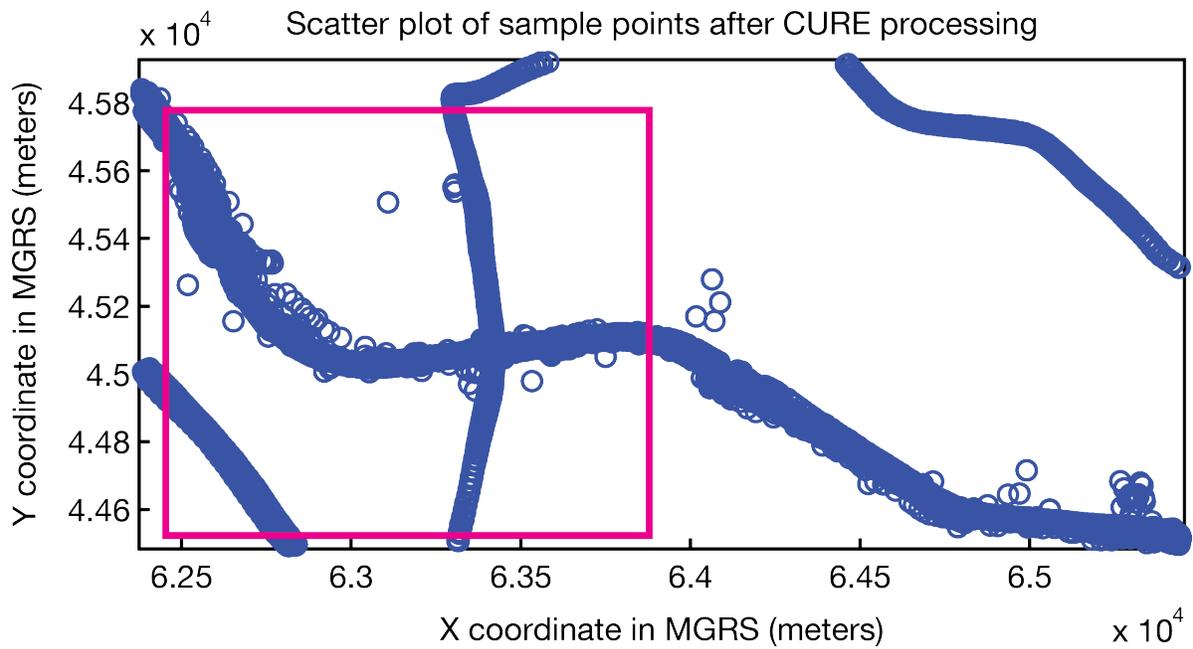


Figure 15: Mean coverage map obtained using the standard CURE algorithm, with 39,947 sample points in the region (1.307, 103.791) to (1.320, 103.763), in given as latitude and longitude coordinates. The size of the area is approximately 1.5 km by 3.2 km. Contrast this with Figure 14 and Figure 16.

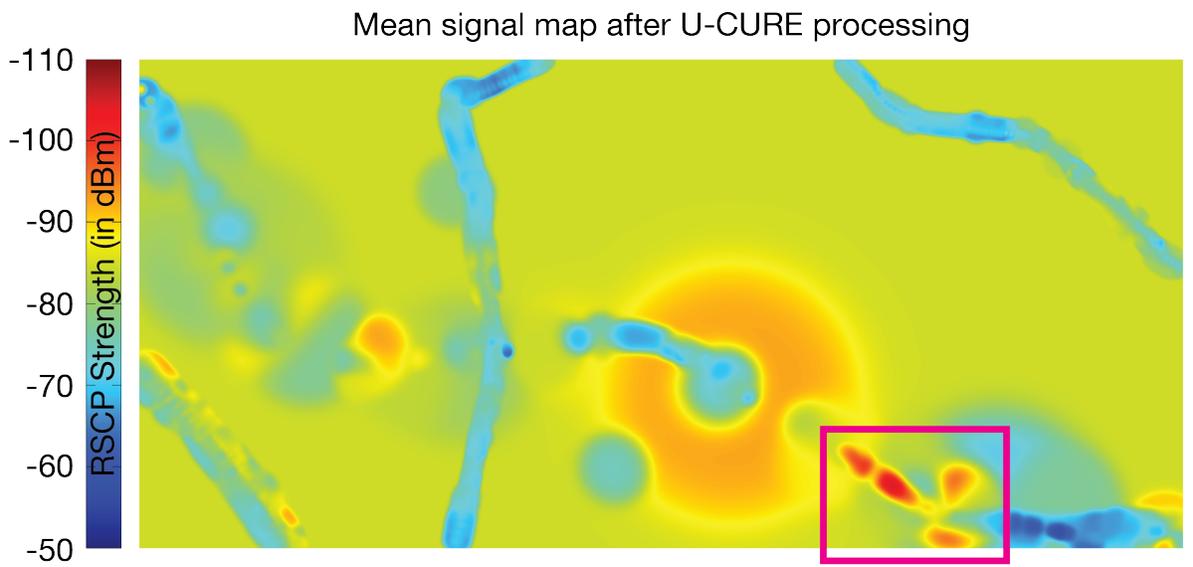
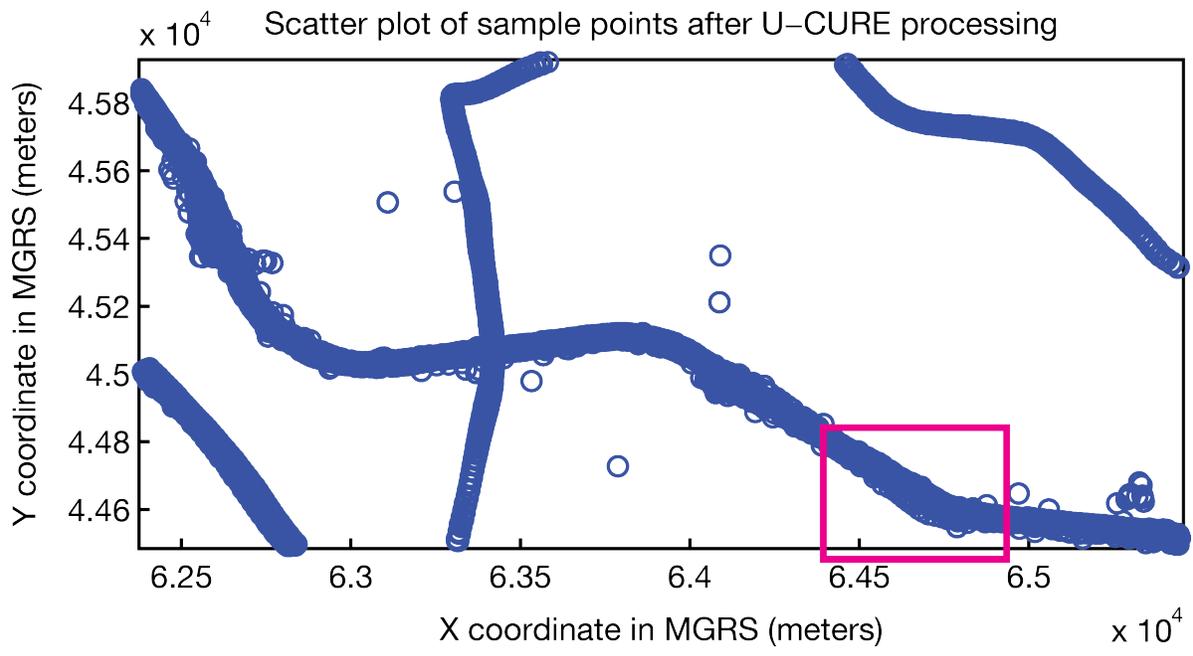


Figure 16: Mean coverage map obtained using the U-CURE algorithm, with 39,947 sample points in the region (1.307, 103.791) to (1.320, 103.763), in given as latitude and longitude coordinates. The size of the area is approximately 1.5 km by 3.2 km. Contrast this with Figure 14 and Figure 15.

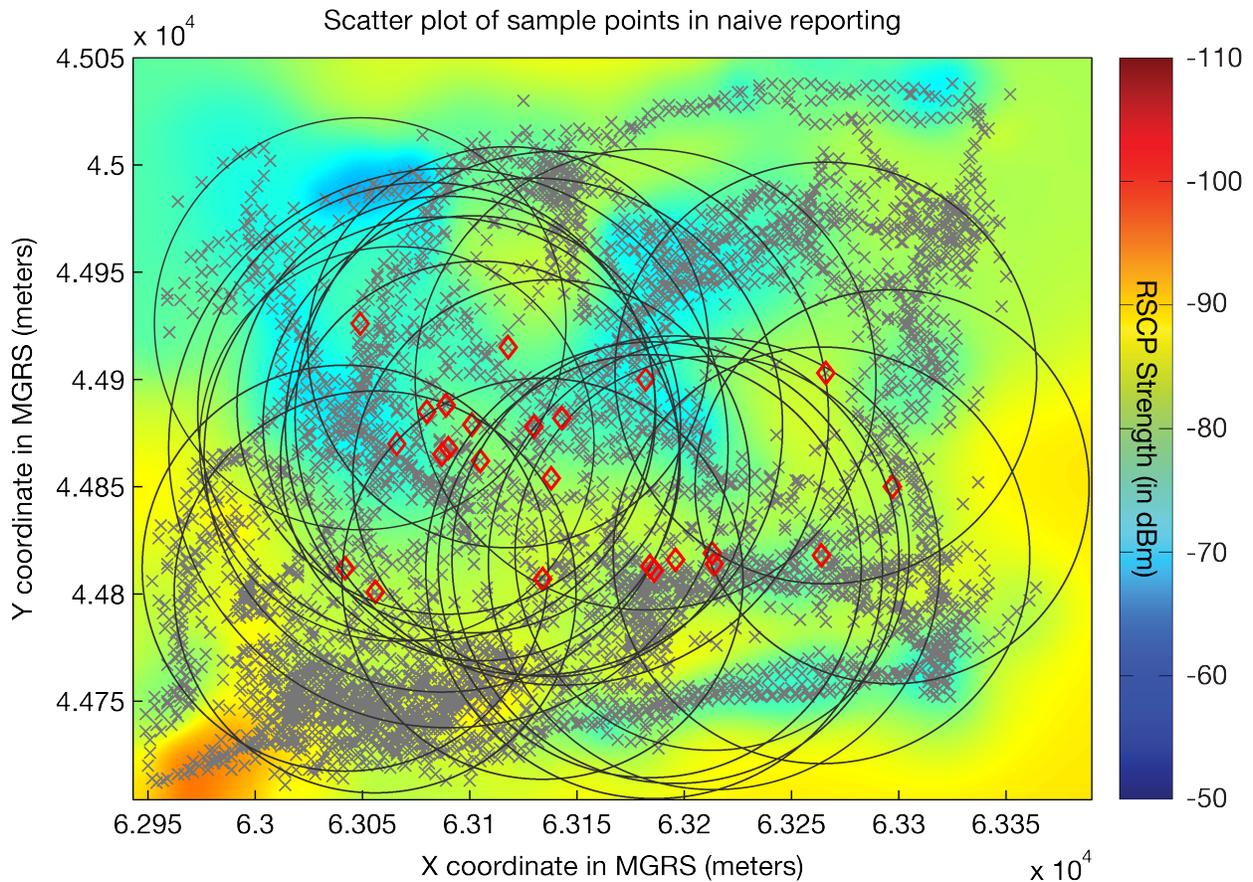


Figure 17: Mean coverage map of a high-rise residential area obtained using naive reporting, derived from 30,352 samples in the region (1.309, 103.772) to (1.312, 103.768). The size of the area is approximately 350 m by 450 m. This sub-region was chosen from the above region to highlight the importance of removing wrongly-localized sample points. The $1-\sigma$ uncertainty radii were plotted for points with location uncertainty exceeding 90 m.

In **Figure 17** and **Figure 18**, we present mean RSCP signal maps, overlaid with scatter plots for a newly-developed area which is known to have poor signal coverage. The $1-\sigma$ uncertainty radius for measurements with σ exceeding 90 m were also plotted for illustration purposes. Without processing, the signal map has indistinct features that appear ‘smudged’ and averaged out. This is evident by the existence of large numbers of points with high location uncertainty. However, after applying U-CURE to discard polluting points with large location uncertainties, the areas that had either excellent

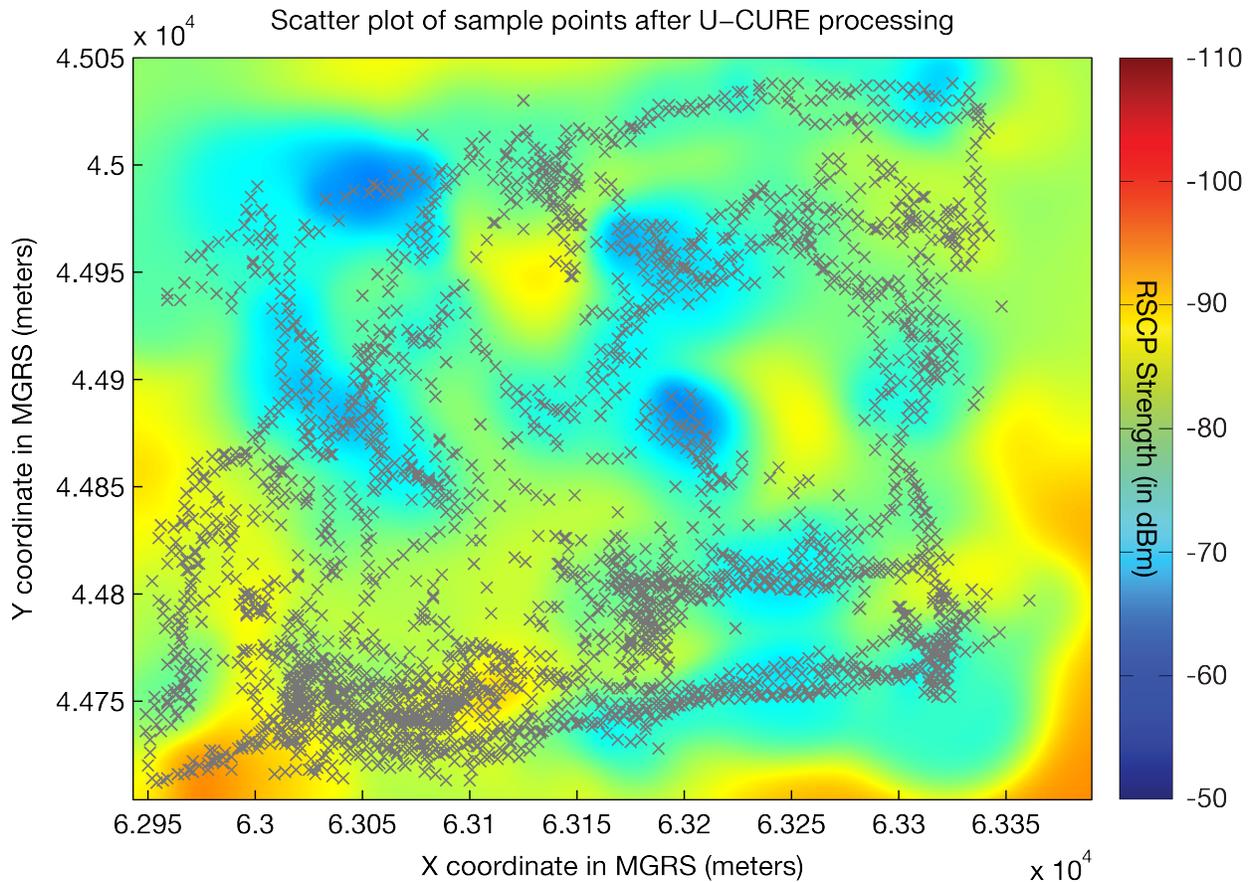


Figure 18: Mean coverage map of a high-rise residential area obtained using the U-CURE algorithm, derived from 30,352 samples in the region (1.309, 103.768) to (1.312, 103.772). The size of the area is approximately 350 m by 450 m. Contrast this with Figure 17. Here, no samples with $1-\sigma$ uncertainty exceeding 90 m remains.

coverage or poor coverage are now clearly demarcated. No remaining measurements had σ exceeding 90 m. Although no areas in this case demonstrate pointedly poor coverage (e.g. at a mean of -100 dBm and below), we remark that subscribers in those areas that have mean levels of coverage below -90 dBm will likely suffer from regular impairments to their services. These may result in call drops and poor call quality.

Figure 19 shows the differences in the mean signal map between U-CURE and naïve reporting. U-CURE reveals that some areas had worse RSCP reception by up to 5.23

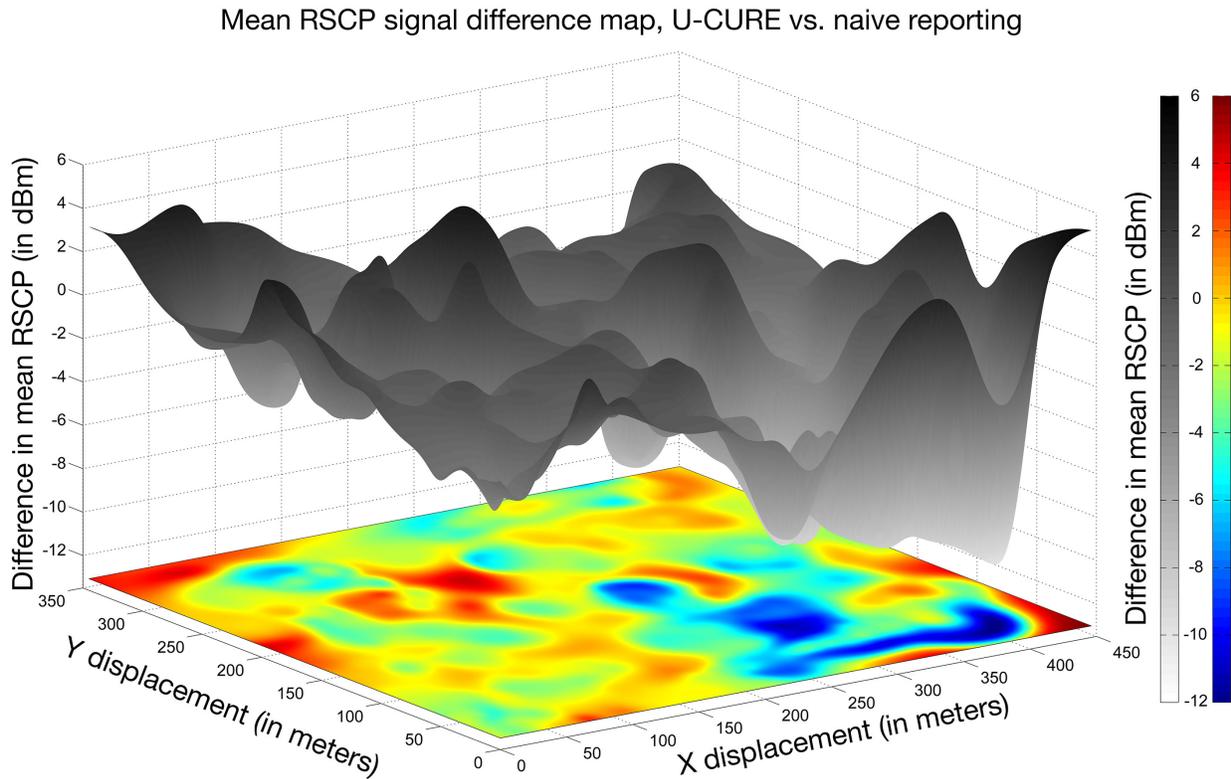
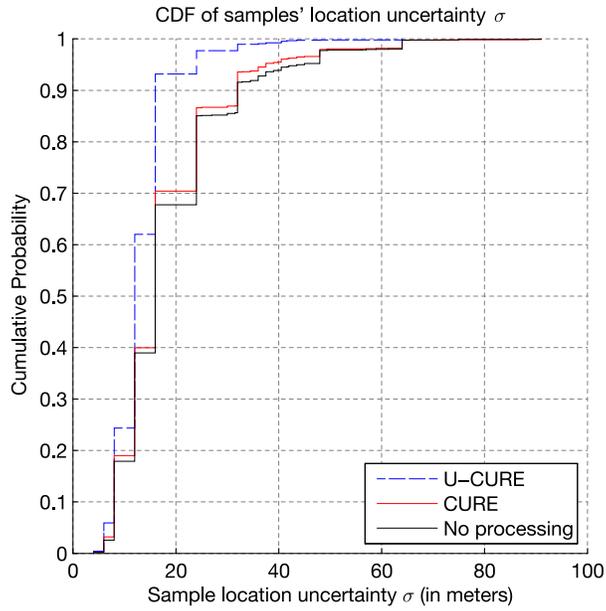


Figure 19: The spatial plot of differences in the mean RSCP maps observed in Figure 17 and Figure 18.

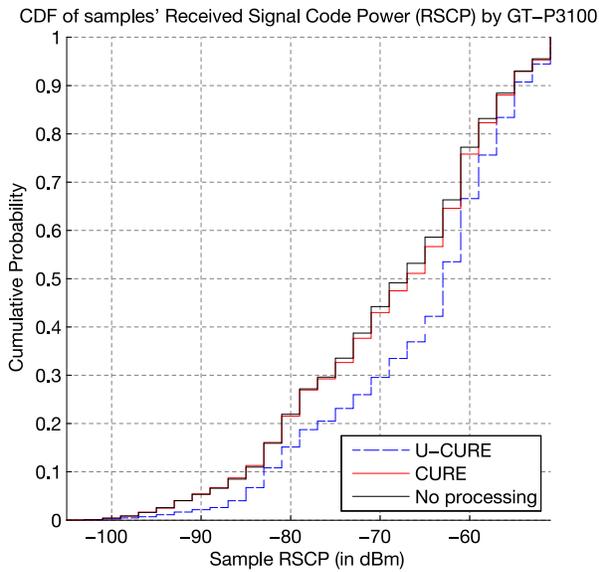
dBm compared to naïve reporting, and better reception in others by up to 10.06 dBm. These correspond to the false-positive and false-negative areas derived by naïve reporting respectively. These differences will otherwise be ‘smudged’ out by large uncertainties in naïve reporting. We remark that these are mean spatial RSCP maps. Hence, areas with only barely-acceptable mean coverage will most likely fail the minimum coverage requirement described in Section 3.1.

3.6.2 Region-of-Interest Based CDF Derivation

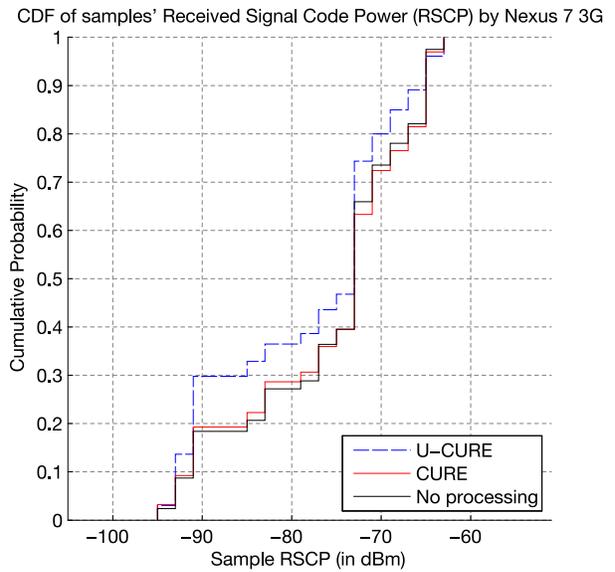
In **Figure 20**, we present another representation of signal coverage that allows cellular operators to directly solve the **CM** problem. In this figure, the cumulative distribution of



(a)



(b)



(c)

Figure 20: The figures above represent the cumulative probability distribution of (a) the $1-\sigma$ location uncertainty of sample points before and after processing, (b) RSCP for GT-P3100s in the same area, and (c) RSCP for Asus Nexus 7s in the area, constructed from 34,837 sample points in the region-of-interest from (1.332, 103.741) to (1.335, 103.743).

RSCP readings, rather than just the mean, is illustrated for an urban outdoor area. **Figure 20(a)** represents the distribution of 34,837 RSCP measurements' location uncertainty collected over the 187 m x 374 m area. U-CURE processing results in a 25% increase of points with under 20 m uncertainty. We see another dimension of analysis when we separate the data by the type of reporting device, as illustrated in **Figure 20(b)** and **Figure 20(c)**. In all of our experiments, we observed that the GT-P3100 reports RSCP measurements very responsively, and the delta of consecutive RSCP measurements is often as small as ± 1 dBm within a space of 1 second.

However, the Nexus 7s tend to be comparatively sluggish and often go tens of seconds or more without reporting any change in RSCP, even when mobile. This can either be due to a driver implementation issue (e.g. a large smoothing window) or hardware differences (in terms of antenna size, build and quality). The ability for Tattle to be extended to include device make and model in RSCP reporting is especially useful for operators, which have to support a plethora of mobile devices on their networks. Knowing that a model of UE is particularly problematic helps the operator in making either network (e.g. increase cell tower antenna coverage at places with higher number of these devices) or business (stop offering these devices to subscribers altogether) decisions.

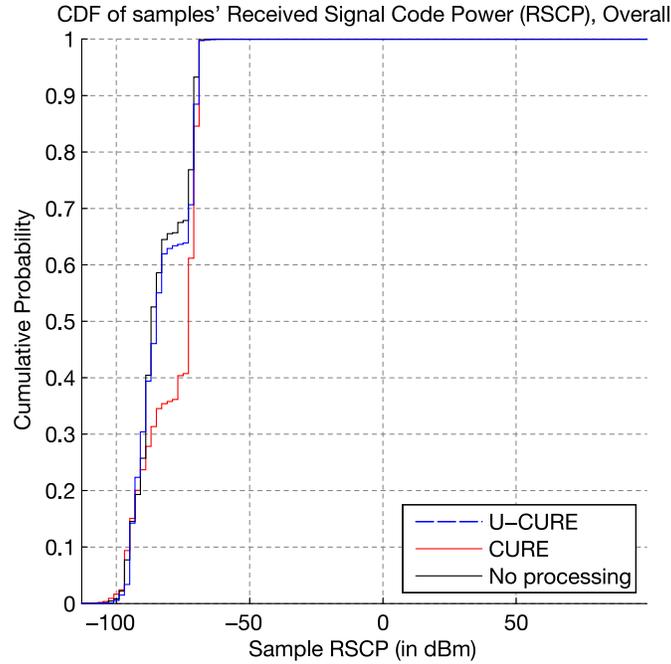


Figure 21: Indoor residential area, where Tattle’s effectiveness is tapered.

3.6.3 Limitations of Our Current Prototype

We also discovered situations where our current prototype tends towards the performance achievable by naïve reporting. This is illustrated in **Figure 21**, where 1,118,384 sample points were collected in a fully indoor environment. Every sample point had large location uncertainties. In this type of cases, there is no basis to keep or discard any particular data point, as their rms distance between one another, even when co-located, is large. The resulting CDF or mean signal map between U-CURE and naïve reporting has no significant differences. However, we remark that in terms of measurement location fidelity, naïve reporting forms the lower bound of performance. Tattle will not perform worse than naïve reporting.

3.7 Summary

In this chapter, we described Tattle, a comprehensive, large-scale cellular network monitoring framework. It leverages on participating UEs to address the **CM** problem. The Tattle framework relies predominantly on opportunistic inter-UE measurement exchange to preserve the co-location of measurements and conserve UE power.

We show through experiments that in urban built-up areas, GPS locations reported by UEs may have significant uncertainties. They can sometimes even be several kilometers away from their true locations. We describe how U-CURE can take into account reported location uncertainty and the knowledge of measurement co-location to remove erroneously-localized readings.

We then illustrate several real-world representations of signal distribution that are of interest to cellular operators. These are made possible through the Tattle monitoring framework. When deployed on a large-scale with sufficient participants, operators can minimize their operational costs of conducting manual walk-/drive-tests. They can also proactively mitigate poor coverage conditions in a timely manner. Otherwise, they have to continue to depend on subscriber complaints after the fact, which can often be vague and subjective.

4. QUALITY MONITORING OF CELLULAR NETWORK DELAY WITH TATTLE

In Chapter 3, we introduced the Tattle crowdsourcing-based monitoring system and framework. We explained how it can be used to construct high-resolution signal coverage maps to address the coverage monitoring (**CM**) problem. It gathers passive downlink-only Received Signal Code Power (RSCP) readings from participating devices. It then leverages on the knowledge of co-location to discard readings that are patently mis-localized. In this chapter, we shall further extend Tattle to address the problem of quality monitoring (**QM**) of cellular network services.

4.1 Quality Monitoring – Problem Overview

Besides just surveying their network's coverage, a battery of tests for network performance are typically conducted during an operator's regular walk- and drive-tests [30][31]. These are to ascertain the quality-of-service of their network. These tests try to detect and capture common issues with cellular networks, such as interference, bit errors, handover failures, and call drops. This approach, together with customer complaints and trouble-ticketing, is still the primary means of detecting and monitoring network performance today. Hence, the shortcomings of such an approach discussed in Chapter 2 unfortunately continue to apply here.

Aggregate coverage (as measured by the aggregate downlink received signal level at any given geographic location) is influenced primarily by the distance to the nearest cellular base-station (also known as path-loss), as well as the effects of shadowing due to obstacles

in the physical environment (such as buildings)³ [8]. However, service quality is subjected to these and many more factors, such as cell load, backhaul capacity and end-to-end latency. Having excellent coverage therefore does not necessarily imply that the quality-of-service experienced by a subscriber will be good.

Hence, instead of passively collecting measurements and discarding those that are mis-localized (as we did for **CM**), we now have to establish a consistent measurement methodology to measure quality-of-service metrics of the network.

In this dissertation, we choose to focus on *network delay* as the example quality-of-service metric to measure. This is due to its fundamental and well-studied importance to the perceived performance [74][75][76] of most of the user-centric applications that we use through our smart devices (such as surfing the web, searching for nearby restaurants, and even hailing a cab). Operators that closely monitor and optimize their networks for network delay can therefore gain huge boosts in terms of their networks' perceived quality-of-service.

We will show later in this chapter that correlations between network delay and localization error exist for COTS smart devices. Instead of simply discarding mis-localized readings (which will skew the remaining results of the measured metric), we leverage on this correlation, as well as the knowledge of co-location, to improve on a measurement's position error (instead of discarding it). We do this using a delay-adjusted extension of the U-CURE algorithm. In this way, we preserve the statistics of the measured metric,

³ The downlink signal received power is also influenced by effects of fast-fading. However, fast-fading is known to perturb the signal power rapidly in time (in the order of milliseconds [73]) around a mean typically determined by slow-fading. Since our intention is to design a system to monitor aggregate coverage on a reasonably useful time-resolution (e.g. on the order of minutes), we neglect discussions on fast-fading as its effect is assumed to be sufficiently averaged out over any reasonably-useful monitoring period.

Table 1: Summary of pertinent differences between Coverage Monitoring and Quality Monitoring

Category	Metric Measured	Correlation with Localization Error	Will Discarding Poorly-Localized Measurements Skew Results?
CM	RSCP Value	Little to None ⁴	No
QM	Network Delay	Strong ⁵	Yes

which we are inherently trying to measure, and also improve the geographic accuracy of derived aggregate result. In order to understand this correlation better, we shall investigate in detail how current generations of mobile devices make use of Global Positioning System (GPS), together with network-assisted localization (also known as the Network Location Provider (NLP) service in Android), to estimate their locations. We will evaluate the real-world performance of these localization techniques under different mobility conditions, and network delay conditions. We then quantitatively establish the relationship between the metric under measurement and localization error. This will clearly motivate the introduction of the Delay-Adjusted U-CURE algorithm.

We summarize the differences between **CM** and **QM** in **Table 1**. Just as in **CM**, we will show how these knowledge come together to allow us to build real-world, accurate, high-spatial-resolution monitoring maps of network delay. By extension, we can address the **QM** problem just using commercially-available devices and our ready-to-deploy Tattle framework. *This can be done without keeping expensive system states. We do not require any location anchors or additional instrumentation, nor any external knowledge that is not available programmatically to application designers (e.g. time-of-arrival information).*

⁴ This is discussed in detail in Section 3.4.4.

⁵ We will demonstrate this in Section 4.5.

4.2 Contributions

In this chapter, we make the following contributions.

1. We report on the performance of localization techniques, used by commercial off-the-shelf mass-market Android mobile devices, based on more than 100,000 data points gathered over 2 months.

- 1.1. We compare the localization performance of various makes and models of devices, on both pedestrian and high-speed mobility, as well as across different network round-trip delay conditions. We focus on a particular example make and model of a popular Android device (Samsung SM-T325) and show through extensive data collection (of over 37,000 data points) that its localization performance using Global Positioning System (GPS), together with Android's network-assisted Network Location Provider (NLP), may be modeled with a simple least-squares fit under high mobility. The mean reported location's root-mean-square (rms) error is non-negligibly correlated with network round-trip delay, increasing by 605.32m per 1,000ms of delay.

- 1.2. To the best of our knowledge, we are the first to systematically compare and contrast the localization results of using GPS+NLP vs GPS-Only, using over 35,000 data points collected in an actual production network. Our evidence suggests that using a combination of GPS+NLP can be detrimental to localization accuracy in both pedestrian and high-speed mobility environments, when compared with using GPS only. In the latter, the median rms error may be increased by over 60%.

2. Based on our quantitative observation of the relationship between network delay and localization error, we make use of Tattle, the cooperative distributed monitoring framework detailed in Chapter 3, and propose the novel Delay-Adjusted U-CURE

clustering algorithm. It extends U-CURE by ‘shifting’ the reported locations of high-delay, possibly mis-localized, samples towards a computed cluster center. We show through extensive real-world measurements that Delay-Adjusted U-CURE greatly improve the localization performance of both GPS-only, and GPS+NLP techniques. It does not require keeping expensive system states, nor requiring any location anchors. No additional instrumentation, nor any external knowledge that is not available programmatically to application designers (e.g. time-of-arrival information) is needed. Our results are promising, demonstrating that median rms location error can be reduced by over 30% with just 3 cooperative co-located devices. Improvements of 70% and more can be seen by having just 6 cooperative devices in a local-area.

3. Finally, we demonstrate that, through extensive real-world measurements of over 34,000 data points, accurate crowd-sourced cellular network delay maps can be derived with Tattle and the Delay-Adjusted U-CURE algorithm. It performs admirably in terms of measurement accuracy, compared to naively relying on raw data.

4.3 Chapter Organization

This chapter is organized as follows. In Section 4.4, we briefly review how Network-Assisted Localization is implemented in Android, our chosen experimental platform. In Section 4.5, we present results of localization performance using GPS+NLP, compared to using GPS only. The impact of factors such as device heterogeneity, mobility, and network round-trip delay will be investigated. Real-world, in-the-wild results of localization performance will be presented. Based on these insights, we briefly describe in Section 4.6 how we use Tattle, and the Delay-Adjusted U-CURE clustering algorithm, to greatly improve the localization performance of both GPS-only, and GPS+NLP techniques. In Section 4.6.3, we apply Tattle and Delay-Adjusted U-CURE to generate accurate, real-

time crowd-sourced network round-trip delay maps. We do this by leveraging on the insights and improvements drawn from the above to address the **QM** problem. We then conclude this chapter in Section 4.7.

4.4 Localization in Current Generations of Mobile Devices

Global Positioning System (GPS) is a well-understood and mature technology that has found its way into an over-whelming majority of smart devices. As such, studies such as [78], [79], and [80] have sought to establish the performances of GPS positional-tracking in commodity mobile handsets. These studies form a useful basis in the understanding of how GPS chipsets perform in the real-world. However, to the best of our knowledge, we are the first to conduct an extensive study of how GPS, and its use in conjunction with Android's Network Location Provider (NLP), performs across different makes and models of devices, network delay, and mobility types.

A comprehensive overview of how Assisted-GPS (A-GPS) is implemented in modern smart-phones is given in [81]. Basically, in most mobile devices, including those used in this dissertation, the device downloads some assistance data (in the form of a file, downloaded using HTTP) from the Internet to aid its searching and decoding of GPS signals. For example, the Samsung SM-T325 downloads this file from <http://xtra1.gpsonextra.net/xtra2.bin>. The assistance data contained in this file include the precise orbital information of satellites, known as the ephemeris, and the coarse orbital parameters and status information of satellites, known as the almanac. Obtaining these data directly from the satellites, which broadcast at a rate of 50 bits/s, can take upwards of minutes or longer [81]. Hence, downloading these data from the network is comparatively much faster. In this dissertation, we use the terms GPS, and A-GPS interchangeably.

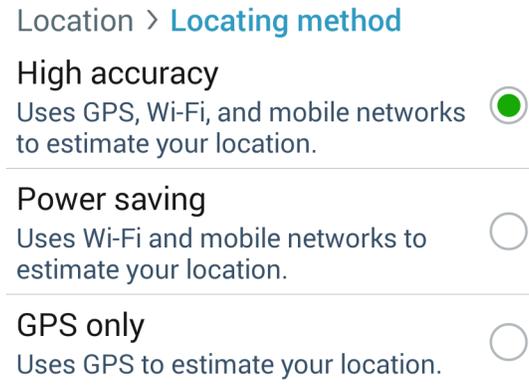


Figure 22: The localization technique selection screen presented to the user, captured from a Samsung Galaxy TabPRO SM-T325. All Android mobile devices display similar variants of this menu to their users.

Another form of network-assisted localization is through finger-printing, which is the basis of how Android's NLP (and Apple's equivalent location service) operates. It works based on vast databases of known cellular towers and Wi-Fi Access Point locations collected through war-driving or clandestine crowd-sourcing using consumers' own devices running on Android [82][83] or iOS [84]. Google's or Apple's location service can then estimate a mobile phone's position based on its current associated cell tower ID, and a list of Wi-Fi Access Points it currently 'hears' [85]. In Android's case, we found that our devices were submitting these information vectors to a single domain name using HTTPS at `google.com`, and getting estimated locations in return. This was verified by blocking the IP address resolution of `google.com` at the kernel-level, which disabled the NLP, and then re-enabling it, which restored the network-assisted localization function.

In typical Android devices, users are presented with a choice of localization techniques, as shown in **Figure 22**. GPS+NLP is used when the first option is chosen, and NLP- or GPS-only is used when the second or third option is selected respectively.

4.5 Experimental Methodology and Localization Performance

We first detail our experimental methodology, and explain our choice of devices, as well as mobility and network conditions under which we conducted our experiments.

4.5.1 Experimental Devices

In our experimental setup, we used the following devices, all connected to the same cellular network service provider, as listed in **Table 2**. We focus on Android devices due to the following reasons:

1. The base Android OS source code is freely available for usage and inspection [86]. For example, the Java code detailing how the Android OS handles location requests from both GPS, and NLP is given in the path `/frameworks/base/services/java/com/android/server/location/`. This can be seen in its entirety in [87]. This gives us the opportunity to interpret our results with more context.
2. Android is still the overwhelmingly dominant smartphone OS, powering 84.4% of smartphones in Q3 2014 [88]. Amongst handset manufacturers, Samsung has the largest market share, owning 23.7% [89]. Hence, we focus our attention on the Samsung SM-T325's from Section 4.5.6 onwards, in order to generate as large a data-set as possible during data collection. It also allows us to control for device variability. However, this should not detract from our main insights presented throughout the chapter as we will show that the results in Section 4.5 are consistent across the investigated devices.

Table 2: The devices that were used in our experiments. For units of the same make and model, we updated all devices to their latest official stable firmware, with no other after-market apps installed, except for the location-collection app.

Manufacturer	Model	# Units	Android
Samsung	GT-P3100	03	v4.1.2
Asus	Nexus 7 3G	03	v4.4.4
Samsung	SM-T325	09	v4.4.2

Although we do not focus on Apple’s popular range of mobile devices, our location improvement scheme proposed in Section 4.6 is *independent of device make and model*, and hence *equally applicable* to Apple’s mobile products.

To determine the ground-truth location, we log our corresponding positions using a commercial aviation-grade Bluetooth GPS device, the Garmin GLO™. To improve accuracy, this device can also receive position information from Russian-based Global Navigation Satellite System (GLONASS) satellites, and it refreshes position information at 10 times per second.

4.5.2 Mobility

We defined two controlled paths in the island-country of Singapore, which we follow in order to investigate the effects of mobility on localization accuracy.

1. *Pedestrian path*: This is an outdoor path measuring around 940 meters in length, and the devices were carried within a backpack while traversing this path at pedestrian speed, that is, around 4.5 km/h. We occasionally had to stop for traffic. In this path, the line-of-sight to the open sky is always unblocked as buildings on both sides of the path are not tall. The path is shown in **Figure 23**.
2. *High-speed path*: This is a public-transport class train route measuring approximately 8 km in length. While moving, trains travel at a mean speed of around



Figure 23: An outdoor pedestrian path measuring around 940 meters in length.

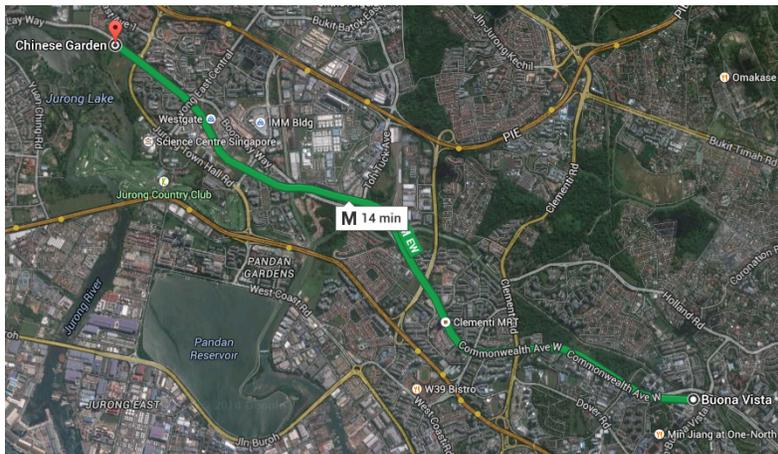


Figure 24: An outdoor high-speed train route measuring approximately 8 kilometers in length.

15 km/h, and up to 80 km/h at top speed. Between the start- and end-point stations, there are also three designated stations, where the train always stops to allow passengers to board and alight. Along this path, there are occasionally tall buildings shadowing either side of the track, thus GPS fixes are not always consistent along the path. This track is illustrated in **Figure 24**.

From this point henceforth, we shall use the term ‘*pedestrian mobility*’ to mean travelling on the pedestrian path, and ‘*high-speed mobility*’ to mean travelling on the high-speed train track. We chose two specific paths to focus on, in order to generate large datasets for analysis while subjected to experimental resource constraints, instead of evaluating many trace routes with small datasets.

4.5.3 Network Delay

As a reference, in a period of over 4 days, we collected over 35,000 samples of network round-trip delay on high-speed mobility. We found that the median delay was 250 ms, and the 5th, 25th, 75th and 95th percentile delay was 144 ms, 198 ms, 558 ms and 1,243 ms respectively. For reference, the maximum timeout value is set at 10,000 ms, and attempted measurements that exceed this timeout value will be set at the nominal value of 10,000 ms. Hence, at the higher ranges of network delays (e.g. those above 1,000 ms), we have much less available data corresponding to location errors at those delays. As such, we setup all our devices to connect to an Amazon EC2 instance hosted in Singapore, operating as an OpenVPN server, and tunnel all packet traffic to and from the devices through that instance. We then use the Linux program, NetEm, to introduce controlled delays ranging from 100 ms to 500 ms per direction, in intervals of 100 ms, at the VPN tunnel interfaces to emulate various degrees of network round-trip delays. This allows us to collect sufficient location data, corresponding to higher network delays (up to around 1,600 ms), to draw meaningful insights. From the application-level, and even to Android’s NLP, this method of emulating network delays is not any different from experiencing real network delays, as they all operate at the packet-based user-plane level.

To measure network round-trip delay at a packet-level, each device issues very small packet probes in the form of HTTP HEAD messages to our Amazon EC2 instance, and

measures the time it takes for the HTTP response to be returned. Each probe results in less than 2000 bytes transferred in total on the uplink and downlink.

4.5.4 Distance to Ground Truth Computation

At every 5 second interval, each device records its location obtained (at the application-level, which developers see) from Android’s `LocationManager` class. The class is instantiated as faithfully described in [90]. It allows the recommended logic to decide which location fix (provided either by the GPS, or Android’s NLP) is the best. In this way, an estimated location, in terms of latitude and longitude, is returned to an application developer. We convert each estimate to a location sample $U = (X_U, Y_U)$ in Cartesian coordinates. Every estimate has an associated uncertainty, expressed as a σ value in meters, and modeled as a two-dimensional normal distribution with the mean location given by the latitude and longitude [91]. In other words, $X_U \sim \mathcal{N}(x_U, \sigma_U^2)$ and $Y_U \sim \mathcal{N}(y_U, \sigma_U^2)$. The ground-truth estimate taken from the GLO is also taken as such. Hence from this point onwards, we compute the root-mean-square distance $D_{S \leftrightarrow G}$ for any sample U_S to the ground truth location U_G as $D_{S \leftrightarrow G} = \sqrt{2(\sigma_S^2 + \sigma_G^2) + (x_S - x_G)^2 + (y_S - y_G)^2}$. This follows directly from our proof given in Equation (9).

4.5.5 Effects of Device Heterogeneity

First, we present the results of using GPS+NLP as the localization approach in a pedestrian mobility setting. This involves three different models of devices, collecting over 3,600 data points per model of device over 3 hours. The results are illustrated in **Figure 25**. Minor variability in the mean rms error distance can be observed across a range of network delays, where the mean is taken for data points within bin widths of 100 ms, from those below 100 ms, up to those within 1500 ms to 1600 ms. This range effectively covers just over 95% of all measured delays described in Section 4.5.3.

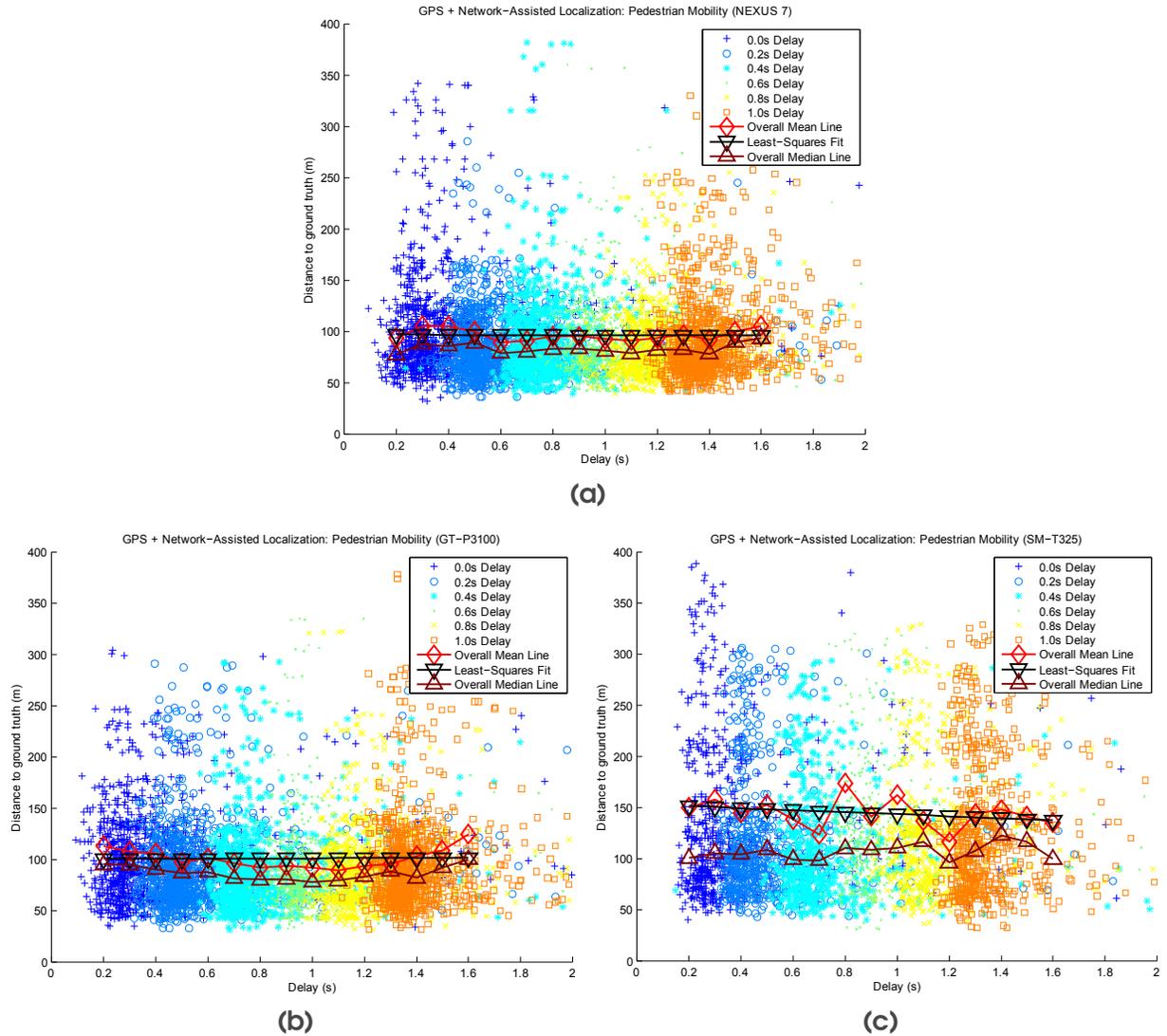


Figure 25: (a) Nexus 7 / (b) GT-P3100 / (c) SM-T325 - Network localization performance under pedestrian mobility.

An immediate takeaway from these set of results is that the rms location errors for each device model vary little across network round-trip delays. Each of the least-squares fit demonstrates little correlation between localization error and network delay. We also observe that the device make and model can introduce fairly different localization performance, echoing the findings given in [79]. Somewhat surprisingly, even under ideal conditions with constant unblocked LOS skywards, rms errors upwards of several

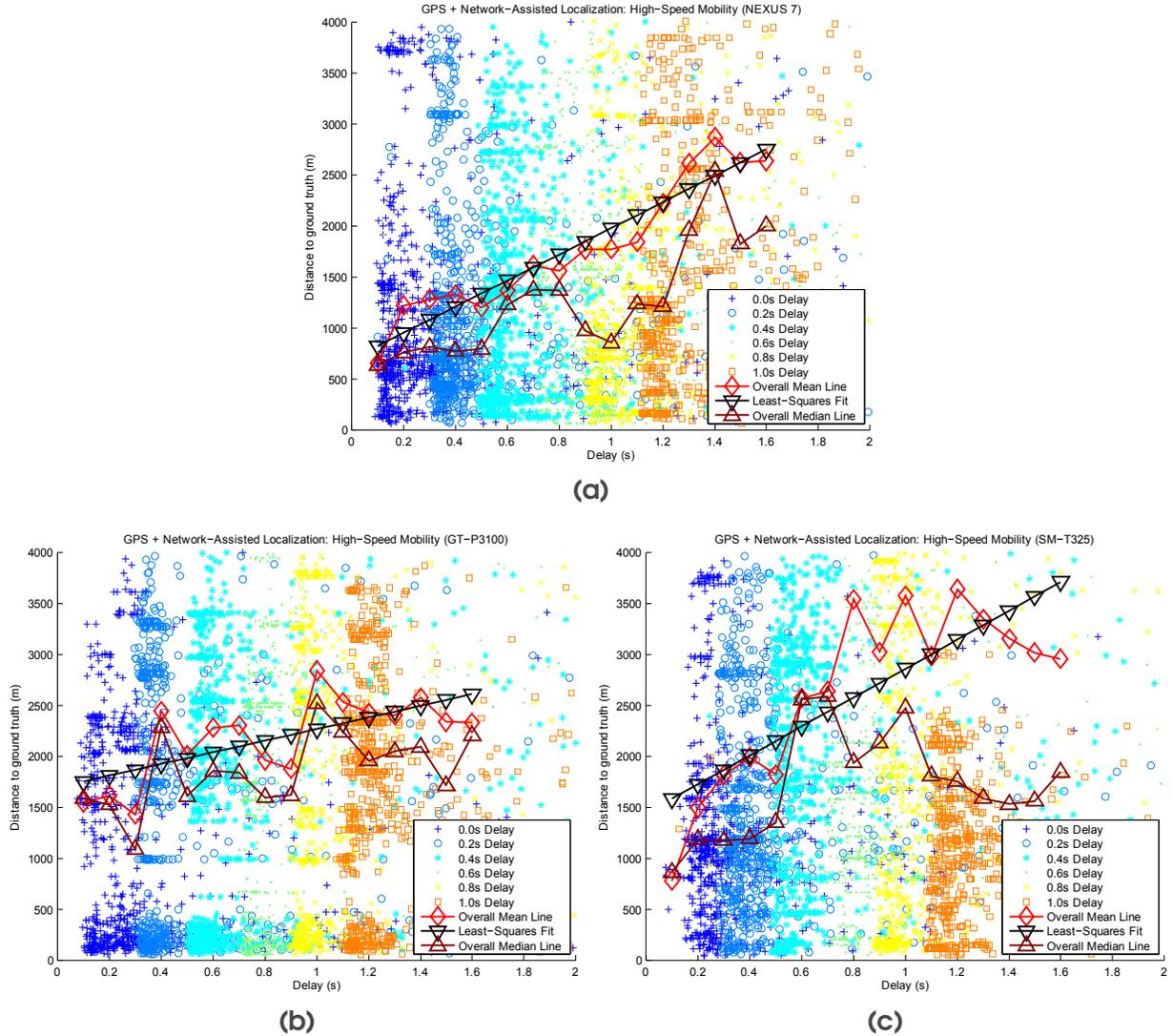


Figure 26: (a) Nexus 7 / (b) GT-P3100 / (c) SM-T325 - Network localization performance under high-speed mobility.

hundreds of meters can be frequently observed. We observe that in our experiments, these correspond to points in time where the reported location “jumps” across large distances, or when the uncertainty of the location (corresponding to σ_U^2) suddenly grows very large. Next, we conducted the same experiment using GPS+NLP, but this time over high-speed mobility. The results are given in **Figure 26**. Under this mobility, the correlation between

localization error and network delay is now obvious. This strong correlation is seen across all three models of devices, albeit to varying degrees.

In **Table 3**, we give the slopes of the least-squares fit to each of the models' localization error behavior, in relation to the network round-trip delay. In this particular experiment, the SM-T325s demonstrate positive rms error correlations with network delays as high as 1416 m/s. This is in stark contrast with its performance with pedestrian mobility.

In the high-speed mobility scenario, we often observe that the devices' GPS receptions may drop out for up to tens of seconds, owing to the occasional shadowing conditions along the mobility path. Hence, each device's location estimate is supplanted by NLP readings which are 'newer', but may be significantly way off from the ground truth as the network delay grows. There are three possible aggravating factors that result in the large correlation between network delay and localization error. Firstly, as discussed in Section 4.4, NLP works by transmitting a vector of information, such as a mobile phone's associated Cell ID and overheard Wi-Fi Access Points, to `www.google.com`. It then awaits the server to return an estimated location. In periods of high network-delay and high-speed mobility, the mobile terminal may have already moved large distances before the estimated location is returned.

Secondly, due to its the dependence on overheard Wi-Fi Access Points and Cell IDs, NLP tend to give very bad results in areas where Wi-Fi Access Points are sparse or non-existent, and cell sizes are large. This is the typical circumstance in the high speed mobility path which we evaluated. In such cases, NLP has very little data to triangulate upon. As a result, NLP may only return an approximate location that is near or at the cell tower which the device is associated with. With large cell sizes, which can be upwards of

Table 3: Gradient of least-squares fit (rms error per second of network round-trip delay)

Manufacturer	Model	Pedestrian	High-Speed
Samsung	GT-P3100	0.8542 m/s	572.1040 m/s
Asus	Nexus 7 3G	-0.5403 m/s	1282.3936 m/s
Samsung	SM-T325	-9.8075 m/s	1416.6122 m/s

kilometers in radius [92], NLP will return locations with errors of those orders of distances as well.

Thirdly, under high speed conditions, the signal strengths of Wi-Fi APs and cellular towers also tend to be subject to large Doppler spreads as a result of fading [8]. Hence, at each point where a snapshot of overheard WiFi APs and Cell IDs is taken, their resulting RF strength readings can be significantly different than those registered within Google's databases. Using these snapshots for Radio Frequency matching and triangulation therefore may result in considerable errors in the location estimated by NLP.

Summary: Across the device models investigated, network round-trip delay has a negligible effect on GPS+NLP performance when mobility is low. However, at high-mobility, all devices demonstrate a high margin of localization error as network round-trip delay increases.

4.5.6 Effects of Combining GPS with Network Localization vs. Using GPS Only

We have established the effects of network round-trip delay on the performance of GPS+NLP localization for high-speed mobile terminals. Next, we investigate the impact of a user's choice of GPS+NLP, or GPS-only localization on an application's localization performance.

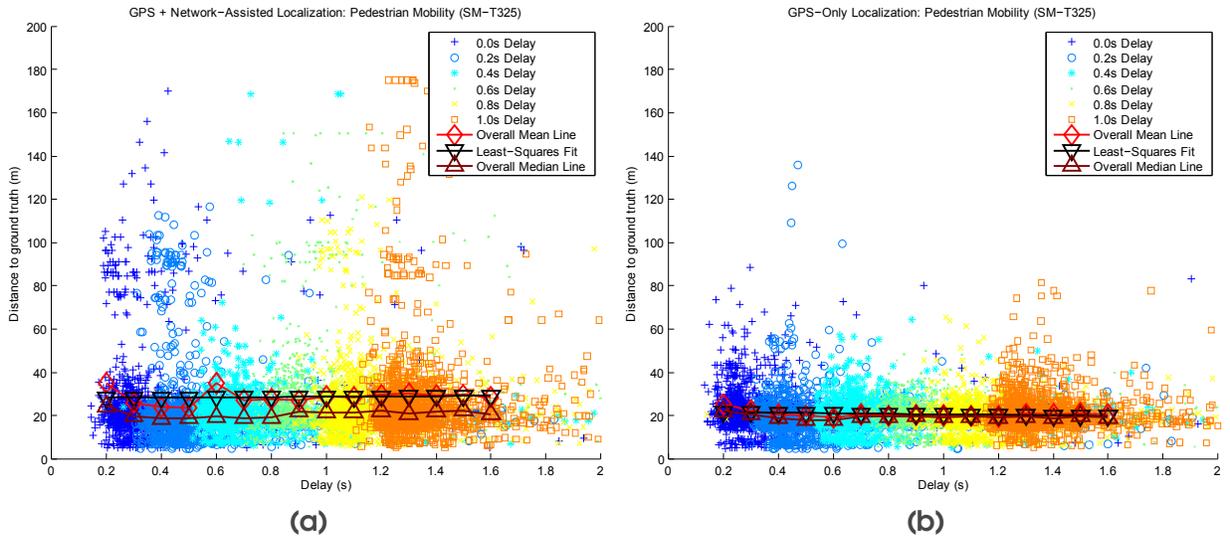


Figure 27: (SM-T325) - (a) GPS+NLP vs. (b) GPS-Only under pedestrian mobility.

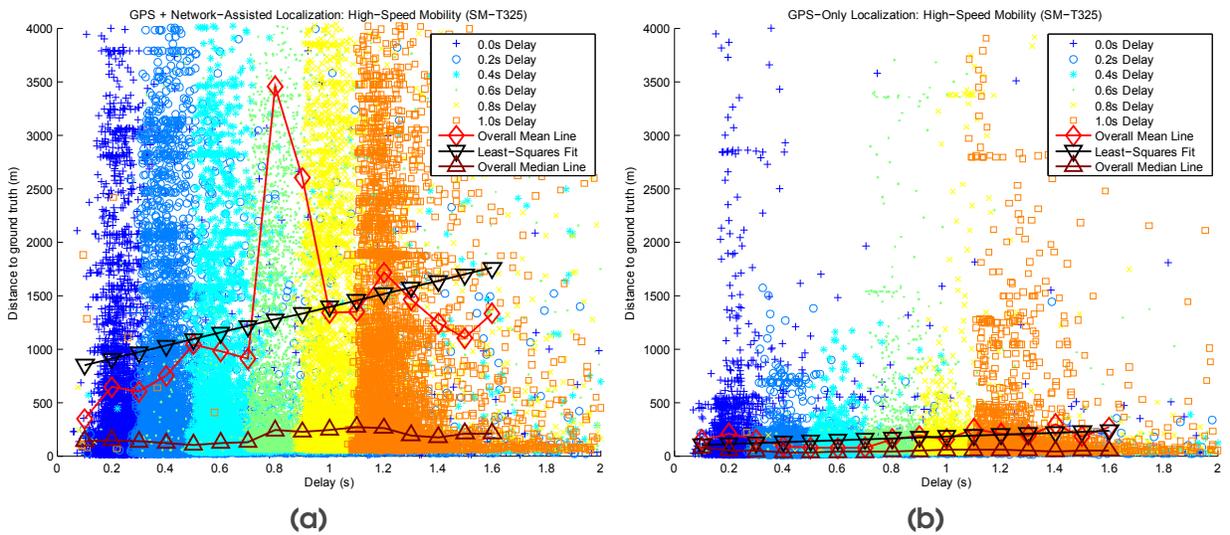


Figure 28: (SM-T325) - (a) GPS+NLP vs. (b) GPS-Only under high-speed mobility.

We now focus on results from the SM-T325s for brevity. However, the overall trends gathered and observed herein are seen in both the GT-P3100's and the Nexus 7's as well.

In **Figure 27**, we demonstrate the results of another data set, collected while on pedestrian mobility, with eight sets of SM-T325s. Four were configured to obtain locations using GPS+NLP, and the other four set to use only locations given by the GPS receiver. This

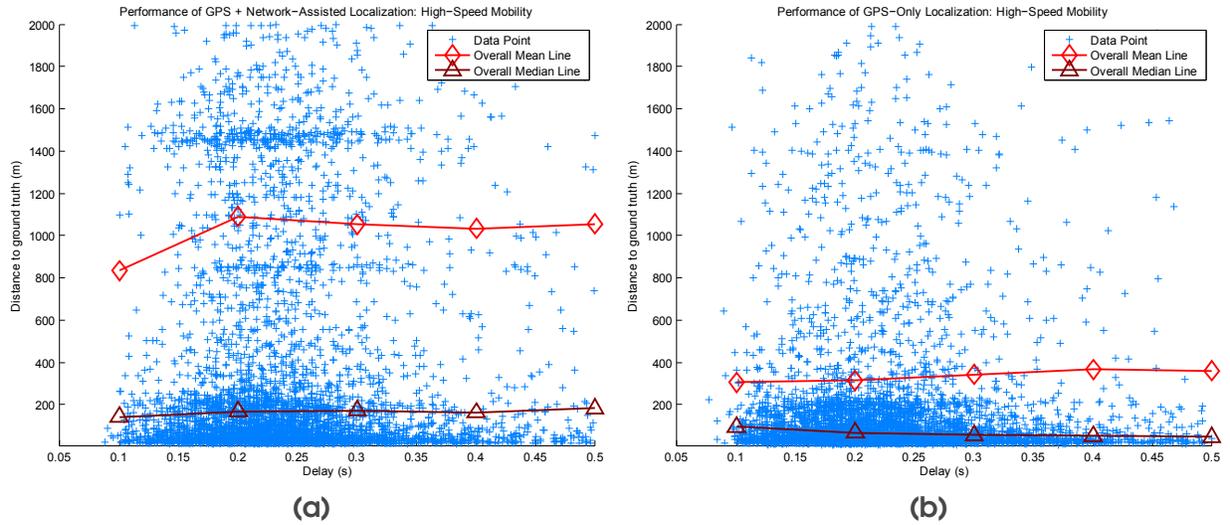


Figure 29: (SMT-325) Performance of (a) GPS+NLP vs. (b) GPS-Only, without artificially-introduced network delay, in a high-speed mobility environment.

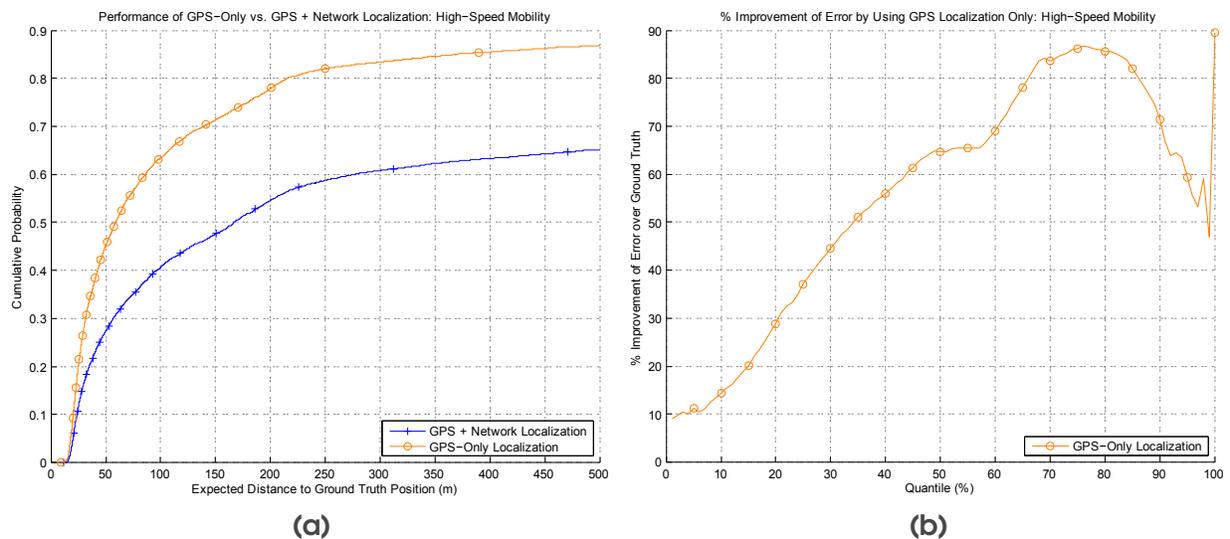


Figure 30: (a) Cumulative distribution function of GPS-Only vs. GPS+NLP, and (b) the corresponding quantile-based percentage improvement, without artificially-introduced network delay.

collection yielded almost 14,700 data points in total. This allows us to remove the temporal variability (e.g. data sets collected on different days may give different results due to weather difference) and directly compare the results.

The overall rms errors look well-controlled despite any increasing network delays. However, it is immediately apparent from the corresponding trend lines that the ‘high accuracy’ mode (see **Figure 22**) does peculiarly worse than simply using the GPS. In fact, the mean rms error of using GPS-only is 20.55 m, with a standard deviation of 8.36 m. However, GPS+NLP yields a mean of 28.09 m, with a standard deviation of 42.84 m.

Next, in **Figure 28**, we illustrate the results of repeating this setup while on high-speed mobility. The two configurations yielded over 18,700 data points each. The stark difference in terms of performance between using GPS+NLP and GPS-only is immediately apparent, and undoubtedly interesting. In this experiment, the former’s fitted mean rms error increases by 605.32 m per second of round-trip network delay, while the latter demonstrated only a very slight positive correlation, at an error-increase rate of just 87.60 m/s.

Then, to reinforce our observations, we repeated the experiments with the SM-T325s under high-speed mobility conditions, but without introducing any network delays. These results are illustrated in **Figure 29**. Furthermore, we present detailed rms error distribution functions yielded by the two different configurations in **Figure 30**. While the mean trend line obviously demonstrates much better results if only GPS localization is used, we see better the vast improvements yielded when comparing across quantiles. At the median, an app developer can possibly *improve the accuracy of localization by close to 65%, just by ignoring results given by Android’s NLP*. Note that the correlation between delay and rms location error is not as apparent in this case, as a vast majority of samples experienced less than 400 ms of round-trip delay, unlike that shown in **Figure 28**. In the latter, we can draw more representative results with a much more uniform spread of network delays.

Summary: The results in this section suggest that *using the ‘high accuracy’ mode of GPS+NLP in fact yields worse, and sometimes much worse, results than simply relying on GPS-only.* This is especially apparent in the high-speed scenario, where mean rms location errors grow as a function of increasing network round-trip delay. However, NLP still has important roles to play for some applications. It provides coarse location estimates in GPS-denied environments, where getting GPS locations are near impossible (e.g. at some sections of the high-speed track, where tall buildings shadow either side of the track). In such cases, perhaps a coarse location estimate may be more desirable than having no location estimate. So, in order to mitigate the shortcomings of GPS+NLP, we propose a location refinement technique described in the following section.

4.6 Tattle – Cooperative Localization through Delay-Adjusted Clustering

Tattle is a cooperative local measurement-exchange system that we detailed in Chapter 3. It comprises of a distributed monitoring framework that is scalable, is designed to monitor real-time network performance on large geographic areas with good measurement location fidelity. It requires minimal instrumentation of smartphone hardware and involvement of subscribers (other than to run a passive background app, as seen in **Figure 6**).

Just as we used Tattle to address the **CM** problem, we propose to use the framework in a similar manner to address the **QM** problem. This allows users in urban areas to simply broadcast their purported locations (gathered at the application-level as described earlier) periodically, at synchronized intervals. This is possible since mobile smartphones can already be time-synchronized to sub-second accuracy using the NITZ protocol [93], or the Network Time Protocol [94]. In this way, each user gathers periodically a vector of $\langle \text{Timestamp, Estimated Location} \rangle$ from all the neighbors that it overhears, together with its

own. The range of overhearing depends on the type of local-area interface used. Because our implementation relies on the WiFi-Direct interface for peer-to-peer exchange, we define the overhearing range to be 30 m, as described in Section 3.4.4.

4.6.1 Delay-Adjusted Clustering using U-CURE

We have established in Section 4.5 that smartphones' estimated locations, especially in high-speed mobility scenarios, can sometimes be egregiously far from their true locations (see **Figure 29**). This error can be influenced by the smartphones' experienced round-trip delay (see **Figure 28**). We shall then use these two observations, together with an extension of the U-CURE clustering algorithm, to improve a mobile phone's own location estimate. This is done without keeping expensive system states, nor requiring any location anchors or additional instrumentation. It does not require any external knowledge that is not available programmatically to application designers (e.g. time-of-arrival information).

We have also given details on the U-CURE algorithm in Section 3.5. In normal circumstances where this algorithm is executed at some backend server (e.g. for crowd-sourcing of some aggregate information), those locations that cannot be merged into the primary, largest cluster are deemed to be outliers. This is because they claim to be too far away from the 'majority' of their peers. However, based on our observations made in Section 4.5.6, we can leverage on this algorithm to *improve on the location estimates* of those previously deemed to be outliers. This is done simply through the wisdom of the crowds, *without needing any anchors or ground truths*.

We briefly illustrate an example of this in **Figure 31**. This snapshot is taken from a real, high-speed mobility scenario from our experiments with nine SM-T325s. We see that the individual reported locations of each device (represented by the diamond-shaped markers) can be upwards of hundreds of meters away from the ground-truth (as measured by the

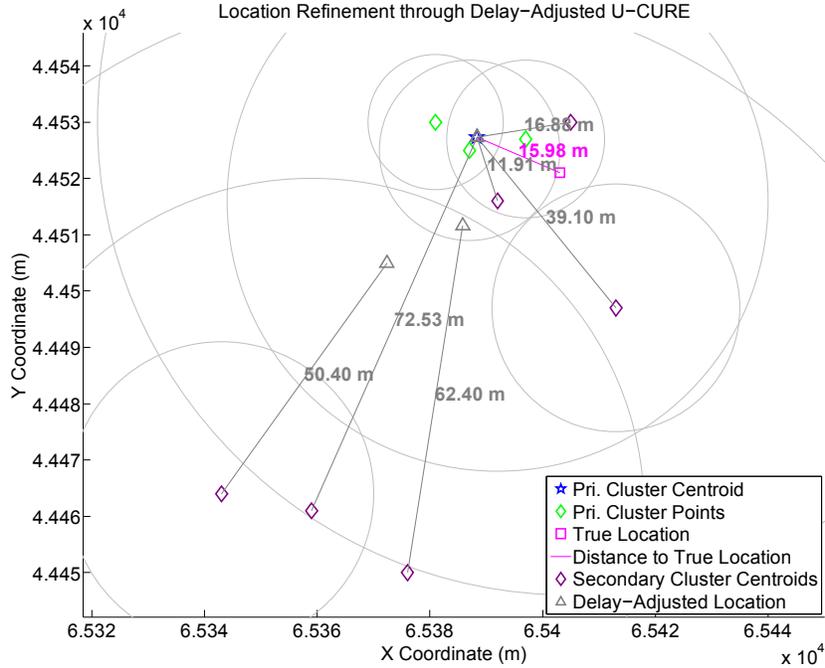


Figure 31: A real-world example of using Delay-Adjusted U-CURE to improve the individual devices' reported positions by shifting them towards the primary cluster centroid by a function of their network round-trip delays.

Garmin GLO, represented by the square-shaped marker). We remark that the knowledge of ground-truth locations is generally unobtainable in real deployments. The beauty of using the U-CURE algorithm lies in its ability to estimate the primary cluster centroid (as represented by the star-shaped marker, and then taken as the estimated ground truth), in the presence of outliers, and without having to conform to any cluster shapes [10] (since the shape of crowds is determined by geographic limitations).

We learnt previously that network round-trip delays tend to affect localization performance, especially for GPS+NLP in high-speed scenarios. So, *we can shift the estimated locations of all those points outside of the primary cluster towards the primary cluster centroid, by a factor of their experienced network round-trip delays* (605.32m per second of delay, as described in Section 4.5.6). This is in contrast to our approach of simply discarding mis-localized measurements, as taken for **CM**. That approach was

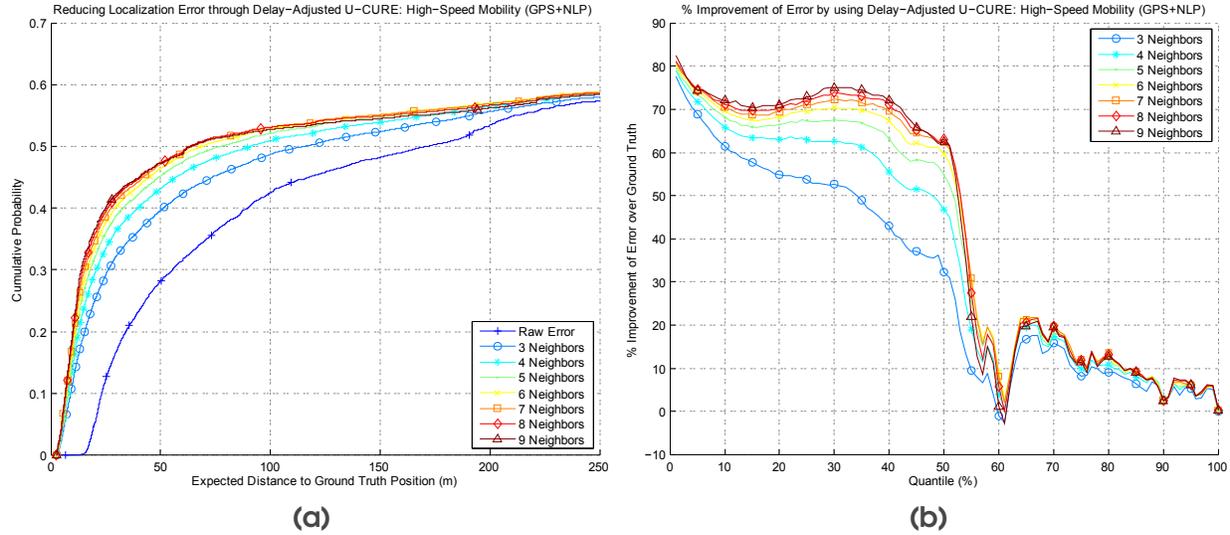


Figure 32: (a) Cumulative distribution function of using GPS+NLP after delay-adjusted U-CURE processing, and (b) the corresponding quantile-based percentage improvement, across different number of neighbors.

justified as our experiments showed little to no correlation between coverage conditions and localization errors, as detailed in Section 3.4.4. However, we cannot discard measurements for **QM** as the quality metric under investigation can be intricately correlated with localization errors.

In **Figure 31**, the displacements of the secondary location points towards the primary cluster centroid are represented as gray lines, and their displaced distances are annotated accordingly.

4.6.2 Experimental Evaluation of Delay-Adjusted U-CURE

Using the procedure explained in Section 4.6.1, we evaluate the localization performance through another set of experiments, first with nine SM-T325s. They are configured to use GPS+NLP localization. The devices were then brought on the high-speed mobility path for a duration of close to three hours, collecting over 17,000 data points in total. Each device broadcasts its own location every 5 seconds, with all the devices' clocks

synchronized to within at most a few hundred milliseconds using NTP. Therefore, at each time-step, a snapshot is taken as a vector of nine estimated locations (obtained through GPS+NLP), corresponding to each of the nine devices. In this way, over 1,900 snapshots were generated. The Delay-Adjusted U-CURE algorithm, using the error-per-delay value corresponding to the SM-T325 as illustrated in **Table 3**, is then applied to every snapshot. Every possible combination of number of devices is taken into consideration (i.e. choosing $i \in \{3, \dots, 9\}$ out of 9 devices).

In **Figure 32**, the cumulative distribution function of the aggregated rms errors before and after applying Delay-Adjusted U-CURE is shown. It is immediately obvious that applying the algorithm yields very tangible improvements in terms of reducing rms location error. With just *three* co-located devices, using the wisdom of crowds and our proposed algorithm can yield an impressive 32.27% improvement in the median rms location error, from 171 m down to 115.6 m. Though a small amount of regression (around -2.7%) is seen between the 60th and 61st quantile, considerable gains are made more than half the time, and also from the 62nd quantile onwards.

Interestingly, the improvements of having just *six* co-located devices and beyond will hit diminishing returns. At the median, upwards of *60% reduction* in rms error can be achieved. With nine co-located devices, even a 75% improvement is possible in the 30th percentile. Taken as a whole, these results are very promising for a relatively-simple approach such as this. We also believe that in populated urban areas, having upwards of 3 or more co-located devices in many areas is often possible.

Summary: We have proposed a simple, *ready-to-deploy* algorithm that involves very little complexity in terms of implementation, but yields very promising results in reducing localization errors in a high-speed mobility scenario while using GPS+NLP. Our proposed

approach involves nothing more than devices exchanging their locations periodically. *No expensive system states, nor any location anchors, nor additional instrumentation are required. There is no dependency on any external, pseudo-oracular knowledge that is not available programmatically to application designers (e.g. nano-second level information such as time-of-arrivals).*

4.6.3 Putting It Together – Tattle, Delay-Adjusted U-CURE, and Building Crowd-Sourced Quality Monitoring Maps

We established the localization performances of devices under various factors such as heterogeneity, network delays, and mobility in Section 4.5. Then, we proposed a simple, ready-to-deploy location refinement method of devices travelling at high-speed scenarios in Section 4.6.1, and subsequently demonstrated its effectiveness in Section 4.6.2. We are now motivated to put together an example **QM** application to demonstrate the usefulness of our work discussed in this chapter.

Here, we propose a crowd-sourcing **QM** application whereby participating users periodically contribute network round-trip delay information. This allows us to construct near real-time aggregate mean network delay heat-maps of places where users frequent. Such applications are finding increasing attention [32][33][34], as irate users in congested places often have no avenues nor means to corroborate their displeasure at the network service. However, unlike these existing applications, our goal is to build very high location-resolution network delay maps that are in the order of meters, just as we did for **CM**. This is unlike those afore-referenced approaches, which only collect network metrics that are location-accurate to the order of hundreds of meters.

Users run a passive background app, the eponymously-named *Tattle*, on their Android devices. It collects network delay information, as well as constantly tracks their

whereabouts, using GPS+NLP or GPS only. Each participating device attempts to measure network delay by probing a standard set of servers. Each probe is a HTTP HEAD request for each server, which includes `google.com.sg`, `youtube.com`, and our Amazon EC2 instance hosted in Singapore. Each probing attempt generates a very small amount of data transferred on the cellular uplink and downlink. For instance, we found that each fetch to `google.com.sg` involves only 484 bytes transferred on the uplink, and 1079 bytes downloaded, fully inclusive of HTTP and TCP/IP overheads. These are based on interface packet captures.

At the start of every five second interval, in accordance with the Tattle framework, each device initiates concurrent probes to the standard set of servers. It logs its estimated location at the start of the probing attempt. At the same time, it broadcasts its location to any nearby neighbors. It will then log any overheard locations together with those broadcasters' IDs, as well as all the delays of the returning probes. This becomes the data vector for that interval, and one vector is thus generated per device per interval. In an actual deployment, the device can immediately upload each vector to a backend for real-time processing. However, in our implementation, we perform computations off-line.

We conduct trials of this application over the course of 2 days, each session lasting more than 3 hours, and collecting over 17,000 data points for the GPS+NLP case, and the same amount in the second day for the GPS-only case.

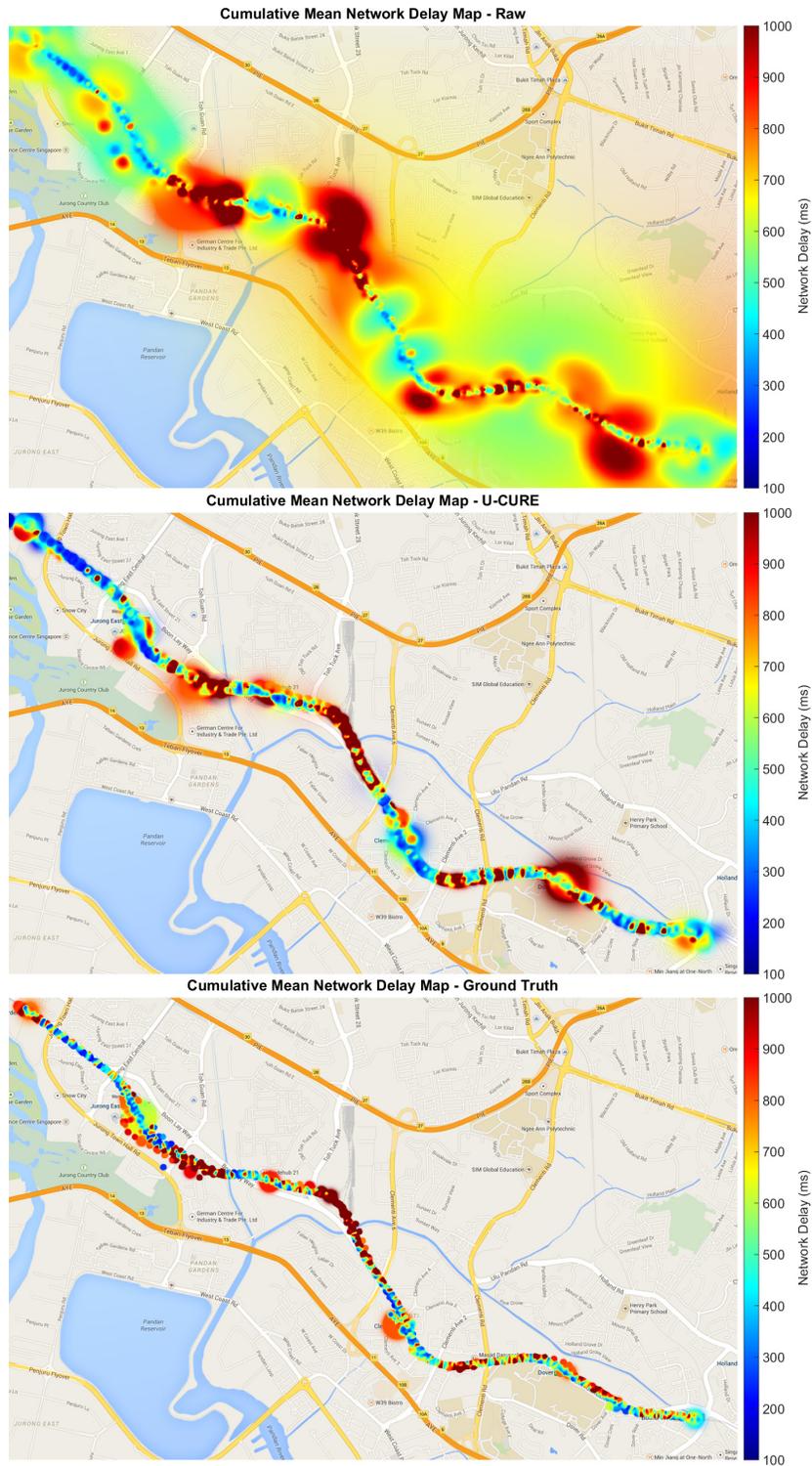


Figure 33: Heat-map of crowd-sourced network round-trip delay using GPS+NLP, before (top), and after (middle) delay-adjusted U-CURE processing, vs. the ground-truth (bottom).

We present visualization results of the reconstructed network delay map for the GPS+NLP case in **Figure 33**. We illustrate the raw, unprocessed results of crowd-sourcing, compared with the one after processing with our proposed Delay-Adjusted U-CURE. We also provide the ground-truth version of the mean network delay map, where location ground-truths were established using our aviation-grade Garmin GLO. Visually, it is immediately apparent that simply and naively taking the raw data in good faith yields a comparatively very poor representation of the network delay (**Figure 33, top**). This is in comparison with the actual ground-truth (**Figure 33, bottom**). However, a much more visually-pleasing mean delay map can be derived just by using our proposed Delay-Adjusted U-CURE together with the Tattle framework (**Figure 33, middle**).

In order to quantitatively evaluate how well the map was reconstructed, we use two measures. Firstly, we define the rms delay error as follows,

$$e_{rms}(\mathbf{M}_{XY}, \mathbf{T}_{XY}) = \sqrt{\frac{\sum_x^X \sum_y^Y [\mathbf{T}_{XY}(x,y) - \mathbf{M}_{XY}(x,y)]^2}{\sum_x^X \sum_y^Y \mathbf{1}_{Z^+}(\mathbf{T}_{XY}(x,y))}}, \quad (10)$$

where \mathbf{M}_{XY} and \mathbf{T}_{XY} are the estimated and ground-truth network delay 2D matrix of width X meters and height Y meters respectively. $\mathbf{1}_{Z^+}(t)$ represents the indicator function that returns 1 if $t > 0$, and 0 otherwise. The e_{rms} is computed at each snapshot (of 5 seconds interval). More than 1,900 rms errors were collected.

Next, we define the ‘smudge’ error s as follows,

$$f(\mathbf{W}_M, \mathbf{W}_T) = \mathbf{1}_{Z^+}(\mathbf{W}_M - \mathbf{W}_T), \quad (11)$$

$$s(\mathbf{W}_M, \mathbf{W}_T) = \sum_x^X \sum_y^Y (f(\mathbf{W}_M, \mathbf{W}_T) \times d_T(x,y) \times \mathbf{W}_T), \quad (12)$$

where \mathbf{W}_M and \mathbf{W}_T are the estimated and the ground-truth 2D matrix (of width X meters and height Y meters) respectively of the ‘weight’ contained in each element. The

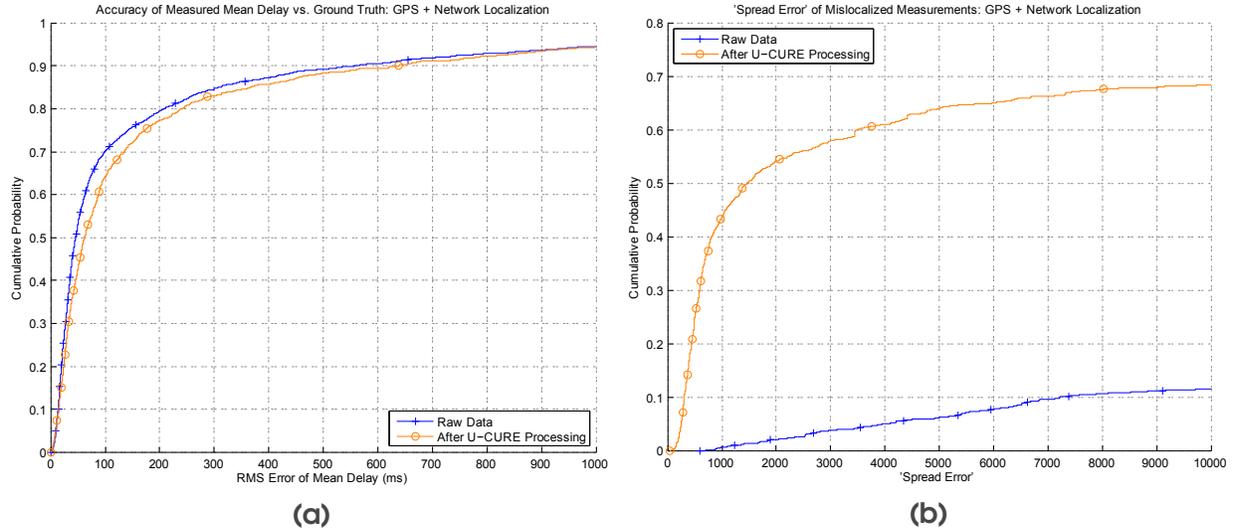


Figure 34: Quantitative evaluation of the heat-map derivation accuracy, based on (a) the rms error of measured delay, and (b) the amount of ‘smudge’ generated by mislocalized data points, using GPS+NLP localization.

weight matrix is simply a sum of all the normally-distributed location probability masks contained in that snapshot, as explained in Section 4.5.4. $d_{\mathbf{T}}(x, y)$ is a 2-D matrix where every element at (x, y) contains a numeric distance from (x, y) to the actual ground-truth location measured by the Garmin GLO at that snapshot.

Simply put, the ‘smudge’ error measures *how much artificial ‘spread’ of the measurements has occurred. These are due to the mis-localized, far-away points that are patently not where they estimate themselves to be, and also those measurements tagged with large location uncertainties.* We present the distribution functions of the rms errors and the ‘smudge’ errors in **Figure 34**. We find that after correcting location errors by a function of network delay, we increased the median e_{rms} slightly, from 45.49 ms to 61.81 ms, and introduced a minor increase in delay error across the 10th to 90th quantile. However, we argue that the slight error increase (in the order of tens of milliseconds, which is almost unperceivable by the average user) is a valuable trade-off considering that we are able to reduce the ‘smudge’ error tremendously, as shown in **Figure 34(b)**.

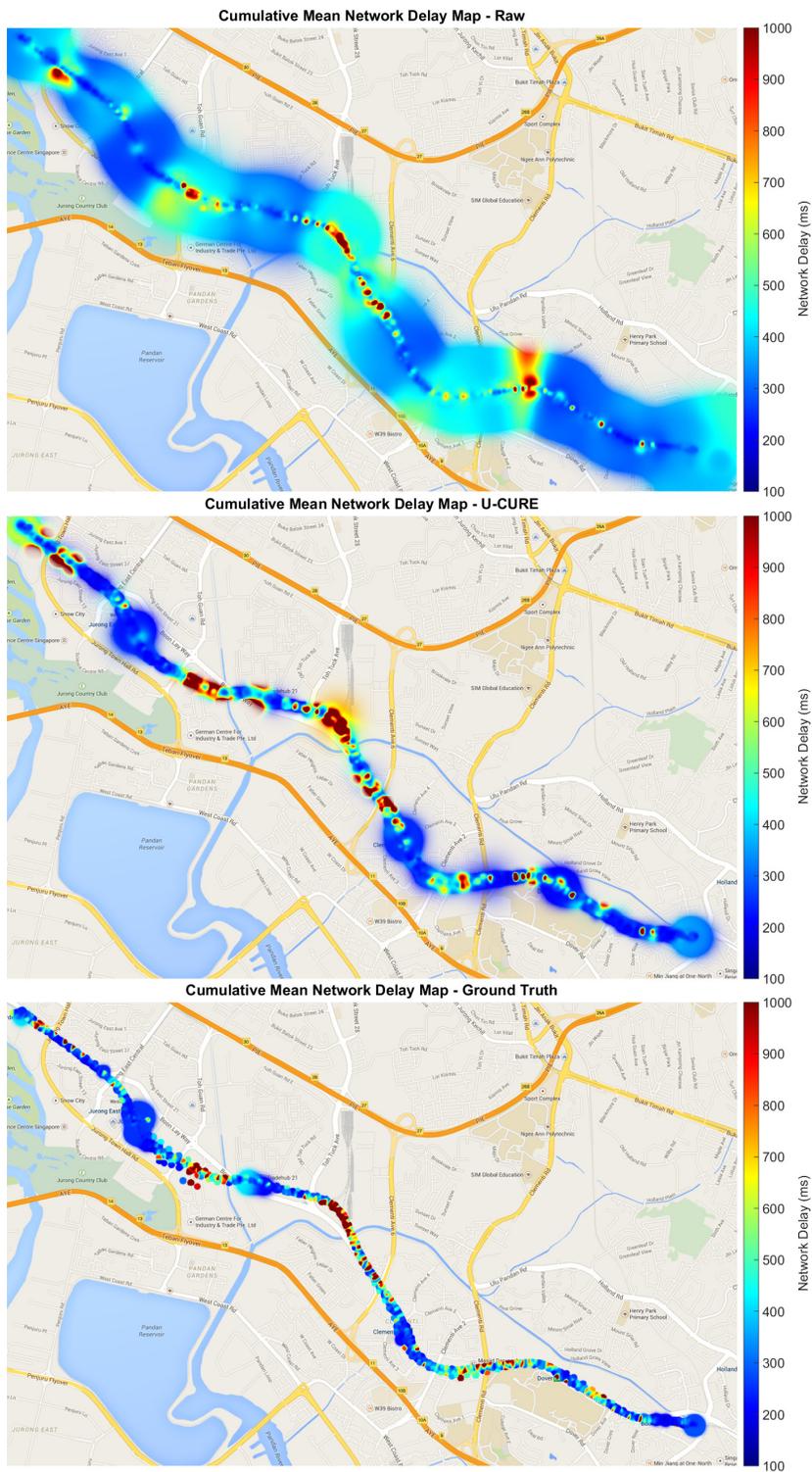


Figure 35: Heat-map of crowd-sourced network round-trip delay using GPS-Only localization, before (top), and after (middle) delay-adjusted U-CURE processing, vs. the ground-truth (bottom).

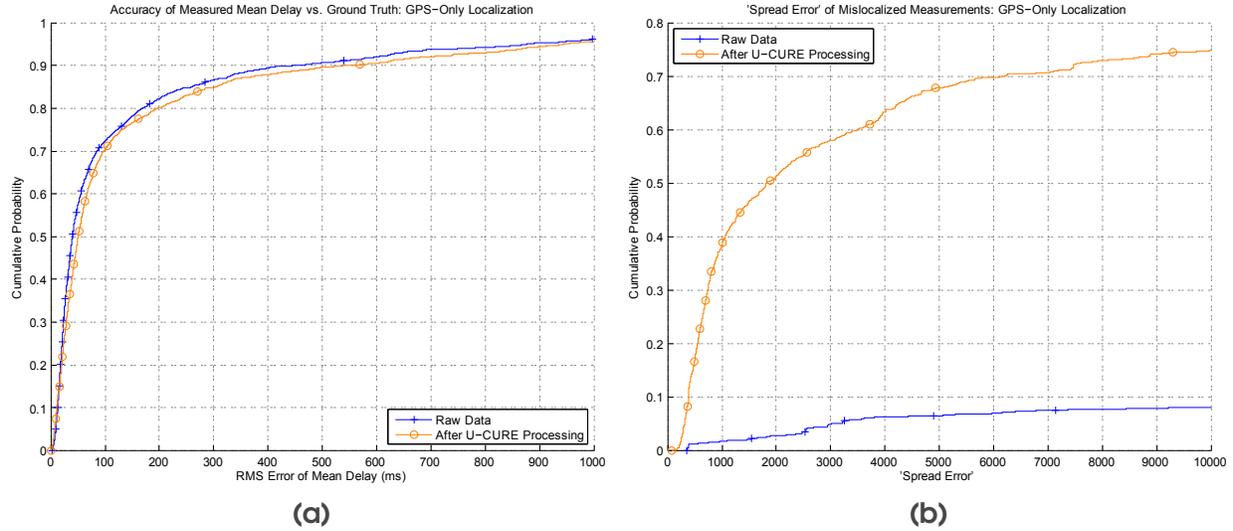


Figure 36: Quantitative evaluation of the heat-map derivation accuracy, based on (a) the rms error of measured delay, and (b) the amount of ‘smudge’ generated by mislocalized data points, using just GPS localization.

From an application developer’s stand-point, greatly reducing ‘smudge’ in exchange for a slight increase in measurement error can be preferable, resulting in a greater visual match as illustrated in **Figure 33**.

We are also interested in examining whether Delay-Adjusted U-CURE can help in a setting where only GPS localization is used. We illustrate the results of this experiment in **Figure 35**. First, we make the observation that from the first day (where the GPS+NLP experiment was conducted) to the second (where the GPS-only experiment was conducted), the network delay situation changes drastically. This is even though both were weekdays, and the experiments conducted at the same times of the day. This motivates the use case of building up-to-date cellular service quality heat maps to track poorly-serviced areas.

Next, we also observe that even though the raw unprocessed map appears slightly less ‘smudged’ than the corresponding one using GPS+NLP, the application of Delay-Adjusted

U-CURE again reduces the ‘smudges’ greatly. We see this quantitatively in **Figure 36**, where, with a very slight, minor increase in e_{rms} across the quantiles, we are able to again suppress the effects of mis-localized points. These points tend to have very large σ_U^2 uncertainties, which can be upwards of hundreds of meters under shadowing conditions that occur as part of the high-speed route.

Based on the above experiments, we can see how Tattle, and the Delay-Adjusted U-CURE algorithm, can help us to build real-world, accurate, high-spatial-resolution monitoring maps of network delay. By extension, we can address the **QM** problem by just using commercially-available devices and our ready-to-deploy Tattle framework. We believe that with these high-resolution monitoring maps, operators can mitigate small blind spots using cost-effective measures, such as deploying femto- and micro-cells. They do not have to resort to always installing large cell-towers, which are expensive and time-consuming to deploy. By remaining agile and fixing small blind spots quickly, operators can also gain favor with subscribers easily.

4.7 Summary

In this chapter, we detailed a comprehensive approach that builds upon our work in Chapter 3, to address the **QM** problem. We first study extensively the achievable, in-the-wild localization performance of commodity retail smart devices. We investigate the effects of mobility and network delay on localization performance. Based on this, we observe that across three types of evaluated devices, using GPS+NLP (as recommended by Android as the ‘High accuracy’ mode) can result in very poor location estimates in high-speed scenarios. Over 600 meters of rms location error per second of network round-trip delay may be introduced in the case of the SM-T325s. Under pedestrian mobility, however, localization performance of these retail devices are much more robust to network

delays. Surprisingly, using GPS+NLP in this type of mobility still yields worse results than relying on GPS alone.

We leverage on the Tattle cooperative framework detailed in Chapter 3 and propose Delay-Adjusted U-CURE. It is a novel clustering algorithm that greatly improves the localization performance of both GPS-only, and GPS+NLP techniques. It does this without keeping expensive system states, nor requiring any location anchors. Additional instrumentation is not needed, nor any external knowledge that is not available programmatically to application designers. Our results are promising, demonstrating that median location accuracy improvements of over 30% is achievable with just 3 co-located devices. Close to 60% improvement can be observed with just 6 co-located devices. Finally, we demonstrate that Delay-Adjusted U-CURE can accurately reconstruct visually-comparable heat-maps that are close to the ground-truth. This is in stark contrast to maps constructed using raw, unprocessed samples. This is demonstrated for both GPS-only and GPS+NLP localization.

5. ESTIMATING AND CLASSIFYING UE-SPECIFIC NETWORK PERFORMANCE THROUGH TATTLE

As traffic load increases, cellular networks naturally become increasingly congested. Subscribers tend to experience high network delays, slow speeds, and possibly service outages in cells and areas that are overloaded. This can be a common occurrence and a frustrating experience, especially during periods of congestion, such as rush hours and spectator events.

5.1 Quality Estimation and Classification – Problem Overview

Whenever a subscriber experiences periods of unsatisfactory network performance, the subscriber may often blame the cellular operator for providing inadequate coverage and resource for the cellular plan they are paying. On the other hand, the cellular operator may point the performance issue back to the device or the software. Indeed, there are websites such as PC Advisor [95], which compares web performance among the many smart phones available in the market today. It is then hard to derive any conclusion to the network performance issue. Network-side rectification is also often unlikely since the root cause cannot be easily identified.

In Chapter 3 and Chapter 4, we detailed how **CM** and **QM** can be performed on an accurate, aggregate, pseudo real-time and large geographic-scale basis using the Tattle framework. These will allow operators to detect coverage and quality impairments in a timely manner. However, when subscribers themselves experience problems with network service quality,

they immediately, and understandably, want to understand whether the perceived fault is network-wide, or one that is just isolated to their devices. Two devices on the same cellular network may experience differing levels of service quality when considered in isolation. However, interesting insights may be drawn when the service qualities of three or more co-located devices on the same cellular network are compared and contrasted.

We believe that one natural way to assist the diagnosis of the root cause of the service quality impairment, from the subscriber's perspective, is to ask: "*am I the only one suffering from this, or is this also experienced by others?*" The subscriber's intention (often without consciously realizing) is to determine if the fault lies with his own device (possibly as a result of software/hardware issues), or with the network. If the subscriber is able to determine that the former is true, he can take some limited mitigation steps. These can include rebooting his device, or closing errant apps that are hogging network resources. If the fault lies with the network, then the subscriber gains closure in knowing that his device is likely to be working fine. He may also find comfort in knowing that other co-located subscribers of the same network in the same area are experiencing the same kinds of performance impairments.

For ease of reference, we shall refer to this as the *Quality Estimation and Classification (QEC)* problem, where two key general questions, listed below, need to be addressed.

1. What is level of service quality one can 'expect' at this moment, wherever one is?
2. Given some level of service quality one can 'expect' to have, does the actual observed service quality suggest any fault that is isolated to one's device?

In this chapter, we will turn our attention from monitoring coverage and quality on large aggregate scales, to estimating and classifying performances of mobile devices on a per-

device basis. We continue to leverage on the Tattle local-area measurement exchange framework to preserve the context of co-location. This allow devices to draw insights about their own performances as compared to those of their immediate neighbors in the same vicinity. We apply quantile regression, a robust statistical technique to estimate a median level of network performance that can be expected at any location with participating devices. We then design a classifier based on robust outlier detection to determine if a device is performing normally, or abnormally. Taken together, these will allow us to directly address the **QEC** problem.

5.2 Contributions

The contributions of this chapter are as follows:

1. We develop a complete and flexible statistical framework, based on *quantile regression* and *outlier classification*, to analyze and systematically identify points in time where a device is not performing as well as its co-located neighbors. This approach allows a device to determine if it is performing “normally”, or “abnormally”. To the best of our knowledge, we are the first to propose, implement, and validate such a peer-to-peer mobile network fault detection application.
2. We report on the characteristics of the delay performance of co-located devices subscribed to two particular cellular network operators in Singapore, and describe the results of applying our proposed approach to addressing the **QEC** problem. These are derived from real-world measurements of over 7,300 time-series of network delay, consisting of over 443,500 data points.

With these contributions, this chapter provides a framework to comprehensively and systematically answer the key **QEC** question. A device can determine whether its observed

poor network performances is an isolated problem, or an endemic condition that affects most other devices in its area. The latter can possibly be caused by poor signal coverage, frequent congestions, or other network- or location-related issues. Through this framework, upwards of 90% accuracy can be observed in the median case in the detection of “normal” and “abnormal” performance with just 6 co-located neighbors.

5.3 Chapter Organization

This chapter is organized as follows. In Section 5.4, we describe how we use the Tattle framework. Smart devices will leverage on low-power, short-range peer-to-peer information exchange to glean performance measurements from other co-located participants. In Section 5.5, we introduce the accompanying statistical framework that we will leverage on in order to address the **QEC** problem. In Section 5.6, we describe our measurement collection methodology, and present some key features of the 7,300 real-world network delay time-series that was collected. We then apply our proposed algorithm on the data, and report on the results. Finally, we conclude this chapter in Section 5.7.

5.4 Tattle – Cooperative Estimation and Classification of Network Delay

Tattle, as detailed in Chapter 3, is a distributed monitoring framework that is scalable, and monitors real-time network performance on large geographic areas with good measurement location fidelity. It requires minimal involvement of participants (other than to allow their devices to participate in monitoring by running a background app on their smart device). In the context of answering the **QEC** problem, we focus on the peer-to-peer front-end component of the framework. As explained in Section 4.1, we choose to measure network delay as a representative metric of network quality because of its great impact on perceived user experience [74][75][76].

In our prototype, each participating device attempts to measure network delay by probing a standard set of servers. We probe several servers so as to determine if the delay trends that are measured are mostly due to the access link (in which case, the general delay trends should be consistent regardless of the choice of probed server), or influenced mostly by the choice of server (where probing different servers return considerably different delay observations). We will show in Section 5.6.2 that most of the delay variability observed is not dependent on the choice of server, so long as the servers are hosted in the same wide-area network (WAN).

Each probe is a `HTTP HEAD` request for each server, such as `google.com.sg`, `youtube.com`, and our self-managed physical server, which we shall refer to as `UNISENSE`. Each probing attempt generates a very small amount of data transferred on the cellular uplink and downlink. For example, based on interface packet captures, we found that each fetch to `google.com.sg` involves only 484 bytes transferred on the uplink, and 1079 bytes downloaded. These are fully inclusive of `HTTP` and `TCP/IP` overheads. We designed our prototype such that each device attempts to measure the network delay every five seconds. This interval is chosen so as to avoid excessive and unnecessary traffic, and to conserve power. More importantly, it avoids the cellular Buffer Bloat problem [96] which can cause an undesirable skew in measured delay. Probing each server in this manner generates just a little over 1 megabyte of data in total per hour. Simple `PING` probes are not used because many networks and servers actively blocks `ICMP` traffic. `PING` behaviors also do not fully reflect the overheads incurred at the transport and application layer. In a production system, this probing interval may be extended to reduce load on the network.

The devices' times are synchronized to the order of milliseconds using the Network Identity and Time Zone (NITZ) protocol [93]. Every time a network probe is complete, each device will broadcast the result of that probe as a tuple of **<Probe Timestamp, Hashed Device ID, Measured Server ID, Measured Delay>**. For probes that experience time-outs (where the HTTP connection did not receive a reply after more than 10 seconds), a nominal value of 20 seconds is used. This broadcast is done on the Wi-Fi Direct interface, and any device that overhears this broadcast simply retains it in memory. If a peer's broadcast is not received by its co-located neighbors due to transmission loss, then the peer's contribution will simply not be considered. That is, in a given area, broadcast losses simply mean that devices will see less neighbors, and will not otherwise adversely impact our proposed system.

This loop of probing and exchanging of measurements, in a production system, is meant to be performed on-demand. For example, a user experiencing impairments to his network service can broadcast a "help" beacon through the local-area wireless interface. Nearby willing participants can then begin the process of limited probing and exchanging of measurements to assist in diagnosis.

So, in order to address the **QEC** problem, the goal is to collect these measurements from other nearby devices, as well as a participant's own measurements. This is done for a small window of time. Subsequently, we apply a statistical algorithm that essentially determines the following:

1. In the given window of time, at which periods were a given device performing considerably worse than others?
2. Given those periods that a device was said to be performing worse than others, is the device performing "normally", or "abnormally" on the whole?

If the device is deemed to be performing “normally”, then any service impairment will also clearly be shared by other participating devices in the same vicinity. If the device is determined to be performing “abnormally”, then any observed service impairment is only isolated to one or a few neighboring devices, but not suffered by others. This algorithm can then be repeated as desired using a sliding window approach, or simply at fixed intervals. It can be implemented either on the individual devices, or at the back-end. In the latter case, each device transmits the results of its own measurement window, as well as that of those that it overheard, to the back-end. The results of the algorithm is then sent back to the device. The choice of either approach will depend on the energy cost of the computation vs. the energy cost of transmission over the cellular interface, as well as whether the device is facing total cellular outage. For our prototype, computation is performed at the back-end.

5.5 Optimal Classification Based On Robust Estimation

In this section, we will detail how we develop our statistical framework to perform **QEC**, given a set of network delay observations of a device, as well as those of its neighbors’. Hence, our overall goal is to design a classifier, such that given a frame of L observations of measured network delay at a selected device, denoted by $\mathbf{Y} := \{Y_i\}_{i=1}^L \in \mathbb{R}^L$, as well as those of its K neighboring devices, denoted by $\mathbf{X} := \{\mathbf{X}_i\}_{i=1}^L \in \mathbb{R}^{L \times K}$, the classifier will detect if the device under investigation performs “normally”, or “abnormally”.

To do so, we perform the following steps:

1. Fit a regression model to explain the relationship between the observations \mathbf{Y} from the device under investigation, and the explanatory variables \mathbf{X} , which are the realizations from the stochastic processes measured by the neighboring devices.

2. Perform outlier detection using quantile regression residual analysis, based on the regression coefficients.
3. Count the number of outliers n in a given window. We assume that n is a realization of the random variable $N \sim F_m(n; \theta_m)$ that is an outcome of one of two models, namely “normal”, or “abnormal”, where m is the model indicator.
4. Perform hypothesis testing to decide if the number of outliers in a given window is consistent with a “normal” behavior (the null hypothesis, \mathcal{H}_0) or an “abnormal” behavior (the alternative hypothesis, \mathcal{H}_1).

5.5.1 Algorithm Description

We now detail each of the steps in our algorithm.

5.5.1.1 Quantile Regression Model Fitting

The most standard regression model structure is *mean regression*, in which one would assume for instance a linear relationship between the observed process and the p explanatory variables $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})$, which include the measurements of the other devices, given by,

$$Y_i = \alpha_0 + \sum_{j=1}^p \alpha_j \mathbf{x}_{ij} + \epsilon_i. \quad (13)$$

ϵ_i is a random variable representing the residual error, which accounts for the fact that the regression model does not capture all variation in the observed process. In the case that the random variables ϵ_i were all i.i.d. with a symmetric zero mean distribution, then this would be equivalent to modelling the conditional mean of the process given a linear function:

$$\mathbb{E}[Y_i | \mathbf{x}_i] = \alpha_0 + \sum_{j=1}^p \alpha_j \mathbf{x}_{ij}. \quad (14)$$

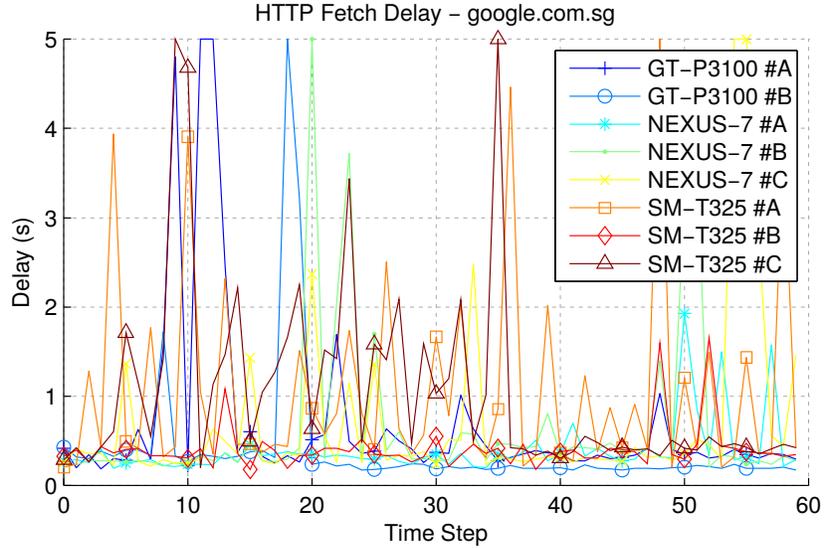


Figure 37: A representative plot of real HTTP fetch delay measurements conducted by 8 co-located devices. Each device fetches a small HTML resource from `google.com.sg` every 5 seconds and measures the delay taken for the fetch to be completed. It is noteworthy that even while co-located, some devices can experience very large variability in network delay, while others see consistently low delays.

The estimation can then be solved via maximum likelihood if the distribution of the errors is assumed, or simply using least-squares. The coefficients are given by:

$$(\hat{\alpha}_0, \dots, \hat{\alpha}_p) = \arg \min_{\alpha} \sum_{i=1}^L (y_i - \alpha_0 + \sum_{j=1}^p \alpha_j \mathbf{x}_{ij})^2. \quad (15)$$

The main limitation of linear regression relates to its high sensitivity (and hence, lack of robustness) to outliers in the observed data. Cellular network delay is one such type of data where outliers occur frequently. This is illustrated in **Figure 37**, which is a representative plot of actual measured delays experienced by eight co-located devices when attempting to fetch a common HTTP resource. These types of outlier occurrences are pervasive throughout our experimental data set, consisting of over 7,300 time-series of actual measured network delay. With data-sets such as these, mean regression will

unbiasedly adapt to the numerous, random ‘spikes’ in the delay by minimizing the overall square error. This behavior is the reason why mean regression is not robust to outliers.

In contrast to mean regression, quantile regression models are capable of dealing with the presence of outliers in the data. Our approach is hence based on *Quantile Regression* (QR), which has the following properties:

1. QR is a robust framework that can identify outliers (i.e., spikes in network delay that are not observed by other neighboring devices), and heavy-tail realizations from the observed process, which in this case is represented by the measurement of network delay.
2. QR is flexible and allows us to introduce any relevant covariates (independent regressor variables) to make inference on the properties of the observed process.
3. QR reconstructs the desired level sets, such as a certain percentile, for the physical process in a consistent and model-based approach, which does not rely upon simplifying assumptions on the distribution of the underlying process properties. This is exceedingly ideal for our approach as we do not need to assume that the statistical distribution of network delay follows any particular class of distributions.

Under a parametric approach, we assume that $Y_i^* \sim F(y^*|\boldsymbol{\theta})$, where $F(y^*|\boldsymbol{\theta})$ is the conditional cumulative distribution function and $\boldsymbol{\theta} \in \Theta$ is a vector of model parameters, all unknown coefficient parameters, and distributional parameters. The quantile function for the conditional distribution of Y_i^* given \boldsymbol{x}_i at a quantile level $u \in (0,1)$ is:

$$Q_{Y^*}(u|\boldsymbol{x}_i) \equiv \inf\{y^* : F(y^*|\boldsymbol{\theta}) \geq u\} = \arg \min_{\boldsymbol{\theta} \in \Theta} \mathbb{E}[\rho_u(\epsilon_i)], \quad (16)$$

where the loss function in the expectation is given by:

$$\rho_u(\epsilon) = y(u - \mathbb{I}[y > 0]), \quad (17)$$

Under this formulation, the conditional quantile function in Equation (16) is given by,

$$Q_{Y^*}(u|\mathbf{x}_i) = \mu_i + Q_\epsilon(u)\sigma_i, \quad (18)$$

where $Q_\epsilon(u) = F_{Z^*}^{-1}(u)$ is the inverse cumulative distribution function for the standardized variable $Z_i^* = \frac{Y_i^* - \mu_i}{\sigma_i}$, and:

$$\begin{aligned} \text{location: } \mu_i &= \alpha_0 + \sum_{k=1}^m \alpha_k x_{ik}, \\ \text{scale: } \sigma_i^2 &= e^{(\beta_0 + \sum_{k=1}^v \beta_k s_{ik})}. \end{aligned} \quad (19)$$

This optimization problem can be solved efficiently using the approach given in [97].

5.5.1.2 Calculating Residuals and Classifying Outliers

To decide if the i^{th} sample in Y_i is an outlier/inlier, we perform a residual-based outlier detection, as detailed in [98]. To achieve that, we calculate the residual for the i^{th} sample as:

$$r_i = y_i - Q(0.50|\mathbf{x}_i), \quad (20)$$

where $Q(0.50|\mathbf{x}_i)$ is the 50th conditional quantile for the i^{th} observation. This corresponds to a median regression instead of a mean regression. We perform the following hypothesis test:

$$\mathcal{H}_{\text{inlier}} : r_i < k_r \hat{\sigma}, \text{ (inlier)}$$

$$\mathcal{H}_{\text{outlier}} : r_i \geq k_r \hat{\sigma}, \text{ (outlier)}$$

where k_r is a resistant parameter that controls the cut-off rate, and $\hat{\sigma}$ is the corrected median of the absolute residuals:

$$\hat{\sigma} = \text{median} \left(\frac{|r_i|}{\hat{\beta}_0}, i = 1, \dots, L \right), \quad (21)$$

where $\hat{\beta}_0 := \Phi^{-1}(p)$ is the inverse cumulative distribution function (CDF) of Gaussian density with the p^{th} quantile. Note that the formulation of Equation (20) distinguishes between positive residuals (where the actual delay is higher than the regressed quantile) and negative residuals. The latter case, where actual delay y_i is better (lower) than the regressed quantile $Q(0.50|\mathbf{x}_i)$, is actually desirable. This corresponds to the case where a device experiences better network delay compared to the median derived from its co-located neighbors, and hence will always be considered an inlier.

5.5.1.3 Counting the Number of Outliers in a Given Frame

We define the following random variable N to represent the number of outliers identified in a given window, where,

$$N = \sum_{i=1}^L \mathbf{1}(r_i \text{ declared as } \mathcal{H}_{\text{outlier}}), \quad (22)$$

$\mathbf{1}(\cdot)$ is the indicator function, and L is the frame length. We used extensive analysis of real data from both “normal” and “abnormal” representations to model N as a realization from a Geometric distribution with different success probabilities p , as follows:

$$\mathcal{H}_0 : N \sim \text{Geo}(p_{\text{normal}}),$$

$$\mathcal{H}_1 : N \sim \text{Geo}(p_{\text{abnormal}}),$$

where $\text{Geo}(p)$ is the Geometric distribution defined as $\mathbf{P}(N = n; p) = (1 - p)^{n-1}p$. To find the values p_{normal} and p_{abnormal} , we used 384 time-series as a training data set, based on which we performed a Maximum Likelihood estimation (MLE), as presented next. The likelihood function is given by:

$$\mathbf{P}(N_1 = n_1, \dots, N_L = n_L) = \prod_{l=1}^L (1 - p)^{n_l} p. \quad (23)$$

By taking the derivative of the logarithmic transform, setting it to zero and solving, we obtain the MLE estimate as follows:

$$\hat{p} = \frac{1}{\sum_{l=1}^L N_{l+1}}, \quad (24)$$

where L is the number of training examples for each model.

5.5.1.4 Device Classification via Likelihood Ratio Test

The final step of the algorithm involves a second hypothesis test to classify the behavior of the device under investigation. This step tests whether the number of outliers detected is best explained by a “normal” behavior (the null hypothesis, denoted by \mathcal{H}_0) or an “abnormal” behavior (the alternative hypothesis, denoted by \mathcal{H}_1). To achieve this, we derive the Likelihood Ratio Test (LRT), given by:

$$\Lambda(Y_{1:L}) := \frac{\mathbf{P}(N = n | \mathcal{H}_0)}{\mathbf{P}(N = n | \mathcal{H}_1)} \underset{\mathcal{H}_1}{\overset{\mathcal{H}_0}{\geq}} \gamma, \quad (25)$$

where the threshold γ can be set to assure a fixed system false-alarm rate under the Neyman-Pearson approach, or can be chosen to minimize the overall probability of error under the Bayesian approach [99]. Since the marginal distribution under both hypotheses follows a geometric distribution with different probabilities p , the test statistic is given by:

$$\Lambda(Y_{1:L}) = \frac{(1-p_{\text{normal}})^n p_{\text{normal}}}{(1-p_{\text{abnormal}})^n p_{\text{abnormal}}}. \quad (26)$$

It is important to note that there is no objective way to choose “optimal” values of k_r and γ , as it reflects a trade-off between detection rate and false-alarm rate. The values of both parameters should be chosen by the policy makers to reflect the Quality-of-Service (QoS) that is acceptable in a particular network, and may change as a function of time and location. In our experiments, we chose the values $k_r = 5, \gamma = 1$ as these provided good detection performance with low false alarm rate.

Table 4: The devices that were used in our experiments. For units of the same make and model, we updated all devices to their latest official firmwares, with no other after-market apps installed, except for the Tattle app.

Manufacturer	Model	No. of Units	Cell Provider
Samsung	GT-P3100	2	#1
Asus	Nexus 7 3G	3	#1
Samsung	SM-T325	3	#1
Samsung	SM-T325	3	#2

5.5.2 Addressing the QEC problem

Once hypothesis testing is complete, we can then answer the **QEC** problem for each investigated device, given its observed network delay, together with those of its co-located neighbors’. If the null hypothesis \mathcal{H}_0 is accepted, the device is classified as experiencing “normal” network delays as compared to its neighbors. Hence, any impairments experienced by one’s device can also be said to be shared by other co-located devices. If the alternative hypothesis \mathcal{H}_1 is accepted, then there is evidence to suggest that the device is experiencing “abnormal” network delays that are not observed by others. The participant should then proceed to perform checks on his device.

We also note that our approach will inherently adapt to situations where most devices experience very high network delay (in the order of seconds and above). In such cases, the threshold expressed in Equation (21) will be elevated accordingly, such that only extreme deviations will be labeled as outliers. In this case, the algorithm will consider devices that jointly experience very high network delays as performing “normally”, which is a result we expect.

5.6 Experimental Setup and Results

In our experimental setup, we used the devices listed in **Table 4**. For each of these devices, we collected HTTP HEAD delay measurements every 5 seconds for `google.com.sg`, and

UNISENSE, for a window of 5 minutes, at each of the 24 positions, located on a busy outdoor train station platform, as illustrated in **Figure 38**.

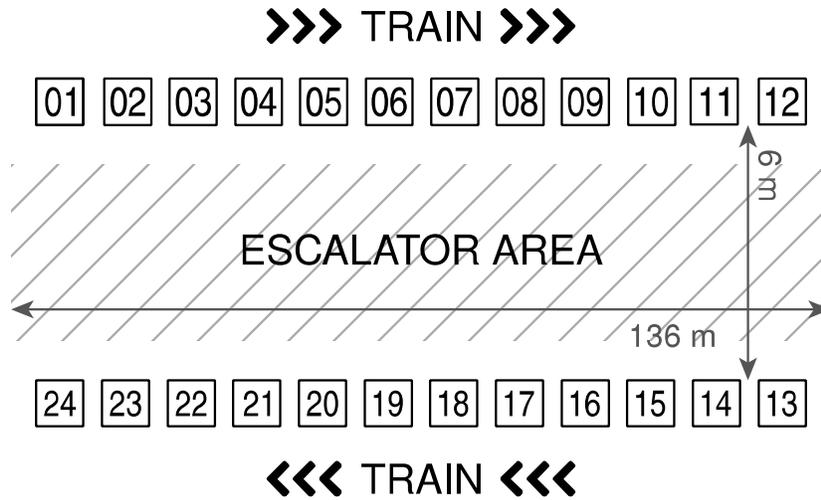


Figure 38: The train station platform, as well as the individual spots where data collection was performed, are illustrated in this figure.

Each set of collection lasts for two hours, and is repeated twice a day, over a stretch of 7 consecutive days. Over 7,300 time-series of five minutes each are collected in this way, comprising a total of 443,500 delay measurements.

This train platform was chosen for its consistent crowdedness, which is typical of the type of congested urban areas that are pervasive in Singapore. Our proposed approach is inherently applicable in such scenarios, where there are large crowds, and unreliable cellular network connectivity.

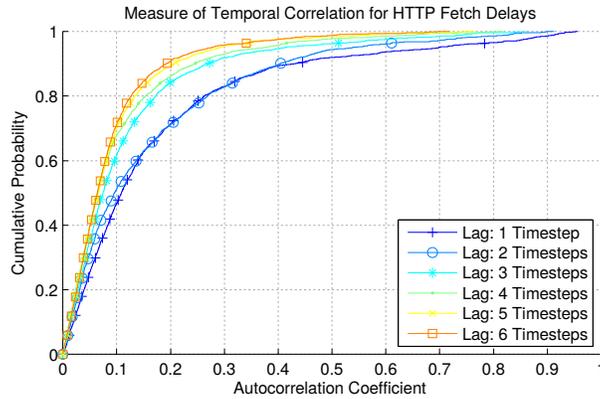


Figure 39: The distribution of autocorrelation coefficients for 2,680 time-series of network delay. Each time-step is 5 seconds in length. Data were collected with the GT-P3100's, Nexus 7's, and SM-T325's on Provider #1's network.

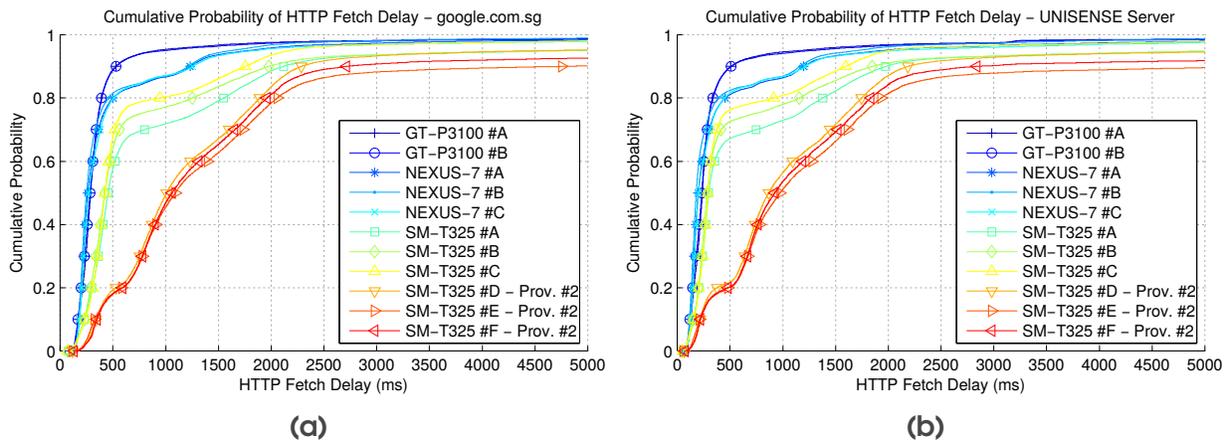


Figure 40: (a) A comprehensive characterization of the HTTP fetch delays for google.com.sg, experienced by each device in our entire data-set of over 220,000 data points. Provider #2's network performs considerably and observably worse than Provider #1's network in largely every experiment. (b) The same plot is given, but these devices instead perform HTTP fetches on our self-managed UNISENSE server, located in Singapore.

5.6.1 Temporal Correlation of Network Delay

In regression analysis of time-series data, one of the core assumptions is that the observations come from an i.i.d. process. In order to validate this assumption, we examine

Table 5: The Ljung-Box Q-Test for temporal autocorrelation. The null hypothesis $\mathcal{H}_{\text{null}}$ is constructed such that the autocorrelation coefficients $\rho_1, \rho_2, \dots, \rho_x$ considering the first x lags are jointly zero (that is, completely uncorrelated).

Null Hypothesis, $\mathcal{H}_{\text{null}}$	Ratio of non-rejection
$\rho_1 = 0$	78.02%
$\rho_1 = \rho_2 = 0$	71.79%
$\rho_1 = \rho_2 = \rho_3 = 0$	72.05%
$\rho_1 = \rho_2 = \dots = \rho_4 = 0$	73.02%
$\rho_1 = \rho_2 = \dots = \rho_5 = 0$	74.78%
$\rho_1 = \rho_2 = \dots = \rho_6 = 0$	75.90%

the temporal autocorrelation of each time-series collected by devices on Provider #1's network.

Figure 39 illustrates the distribution of the autocorrelations computed for 2,680 time-series (collected for `google.com.sg`, using the GT-P3100s, Nexus 7s and SM-T325s on Provider #1's network), over a range of lags. The interval between each time-step is five seconds. The key result here is that even at a lag of one time-step, close to 80% of the time-series demonstrated very little autocorrelation (coefficient ≤ 0.25).

To further validate the assumption, we next apply the parameterized Ljung-Box Q-Test to the data. It tests each time-series under the strict null hypothesis that the autocorrelations under the first x lags are *jointly* zero. The results are given in **Table 5**. The ratio of non-rejection indicates the percentage of those 2,680 time-series tested that does not reject the null hypothesis at a significance level α of 0.05. Even at a lag of $x = 1$, there is no evidence to reject the null hypothesis, more than 78% of the time.

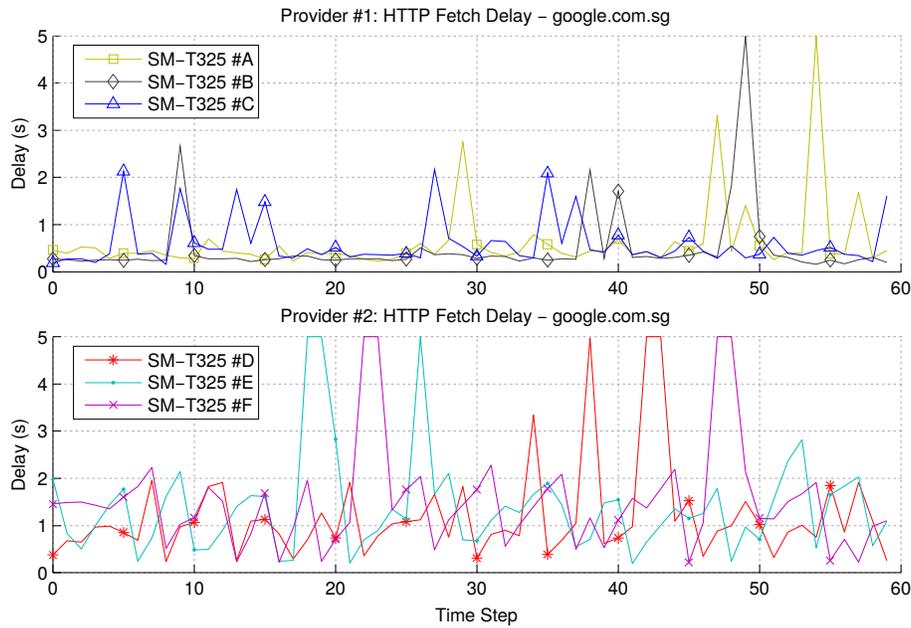


Figure 41: An illustration of two sets of representative time-series, representing the HTTP fetch delay for `google.com.sg`. The devices were co-located at the same place, and are the exact same make and model, but 3 were connected to Provider #1’s network, while the others were connected to Provider #2’s network. The interval between each time step is 5 seconds.

5.6.2 Choice of Probed Servers

The observed variability in delay is mostly due to the performance of the cellular access portion of the network, and not influenced by the choice of probed servers. To demonstrate this, we provide the cumulative distribution functions of the HTTP HEAD fetch delay for both `google.com.sg` and UNISENSE in **Figure 40**. Clearly, the overall trends in performance are almost identical. This is also patently observed on Provider #2’s network. Hence, without loss of generality, we focus on the delays measured for `google.com.sg` henceforth.

5.6.3 Performance Differences between Cellular Providers

It can be seen in **Figure 40** that the difference between the delay performances of Provider #1's and Provider #2's network is drastic, regardless of the choice of probed servers. The median fetch delay for both `google.com.sg` and `UNISENSE` on Provider #2's network is almost 300% longer than that of Provider #1's. **Figure 41** shows an example of this difference in the temporal domain. Delays on Provider #1's network were subjected to much less variability compared to Provider #2's network, and this is true in most of the time-series that we collected. Low delay variability, in general, is desirable for applications that are sensitive to delay jitters, such as video conferencing.

5.6.4 Performance Differences between Device Models

Figure 40 also reveals some interesting relationships between the delays observed and the make and model of the device performing the measurement. The Samsung GT-P3100, introduced in 2012, performs very stably, with around 90% of its measurements coming in below 500 ms for both servers probed. In contrast, Samsung's latest flagship tablet, the SM-T325, performs poorly, with 90% of their measurements coming in between 1500 to 2000 ms, on the same network. This however, does not invalidate our approach of taking all co-located participants' measurements into consideration, regardless of make and model, to determine if a device is doing worse than its neighbors. This exactly is in line with the basic intention of **QEC**. However, if there are other co-located devices in the same area with the same make and model, further extensions may be easily made to consider only same device types. This is however beyond the scope of this dissertation.

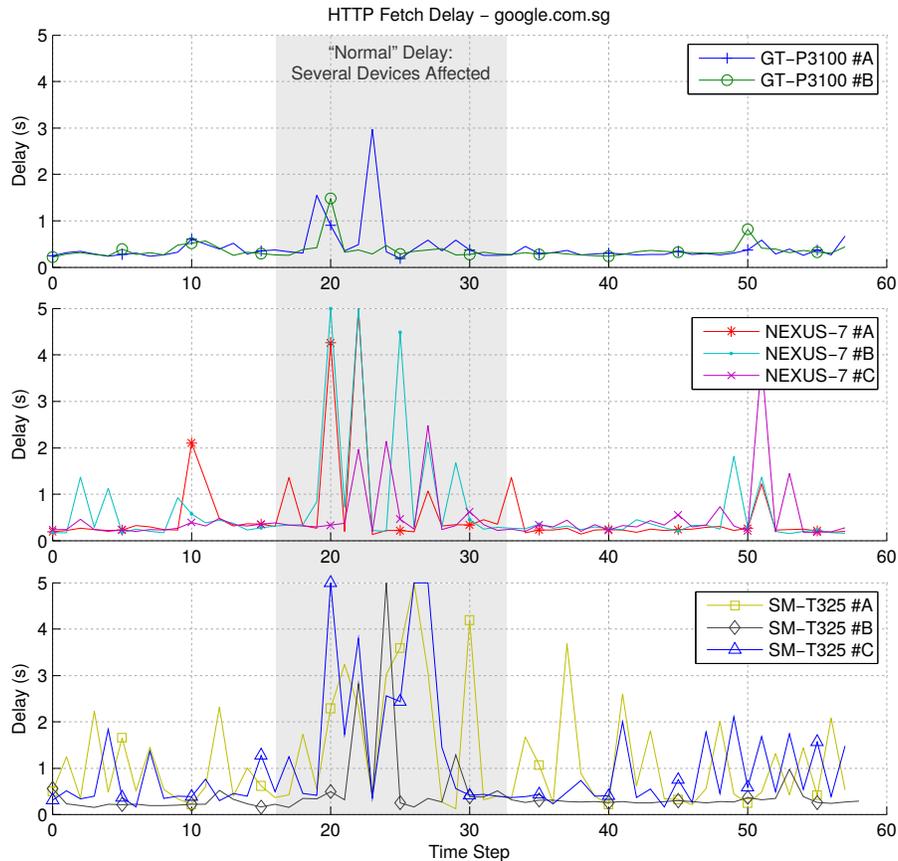


Figure 42: An illustration of three sets of representative time-series, representing the HTTP fetch delay for `google.com.sg`. The devices were co-located at the same place, using the same provider, but are arranged according to their makes and models. Periods of high delay can affect many devices in the same area, as shown.

5.6.5 “Normal” vs. “Abnormal” Performance

There are numerous periods in the course of our experiments where sudden, persistent spikes in delay are observed by many devices. These will result in windows that could stretch over several minutes where many devices observe very high delays, or complete outage of data service.

This is illustrated in **Figure 42**, where within a window stretching over more than one minute, devices experience abrupt spikes in their fetch delays. At least 6 out of the 8

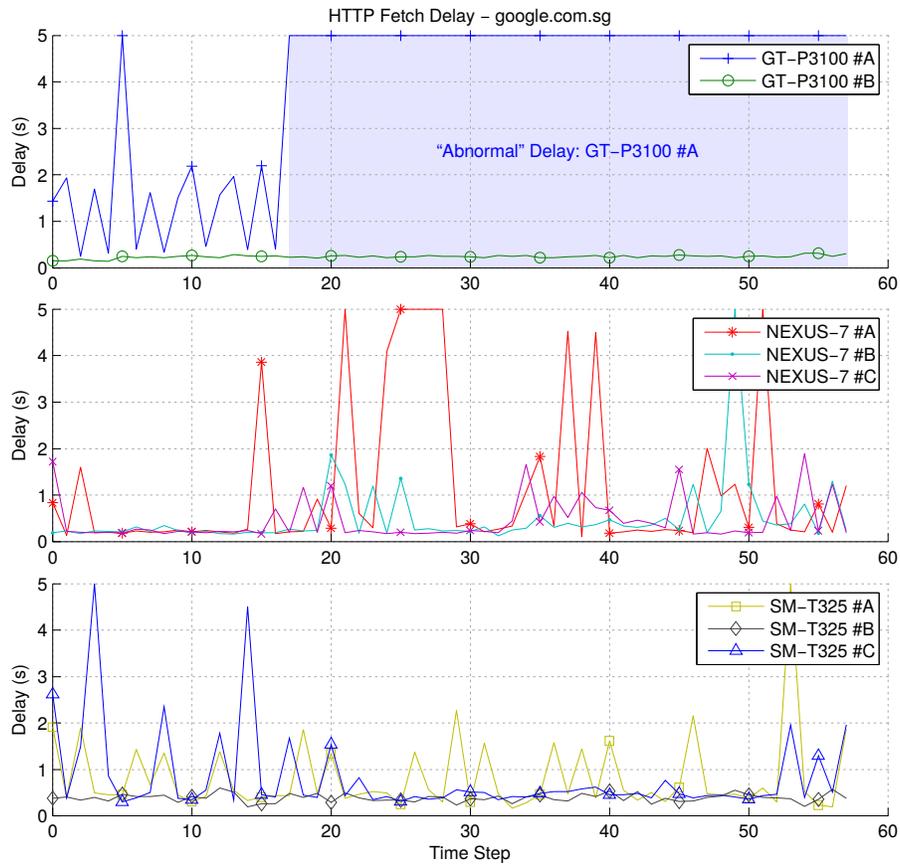


Figure 43: An illustration of three sets of representative time-series, representing the HTTP fetch delay for `google.com.sg`. The devices were co-located at the same place, using the same provider, but are arranged according to their makes and models. Oftentimes, a device may experience severe outage of network service, but most other devices are unaffected.

devices tested were affected. For applications that are somewhat tolerant of delay, such as web-surfing, this may not appear to be wholly debilitating. However, this could have frustrating consequences for users that are streaming videos from `youtube.com`, or are having live conversations on Skype. Nevertheless, in situations such as these, many co-located devices indeed suffer together (although to varying degrees). Hence, each participant’s device that are faced with such delays can be said to be performing “normally”.

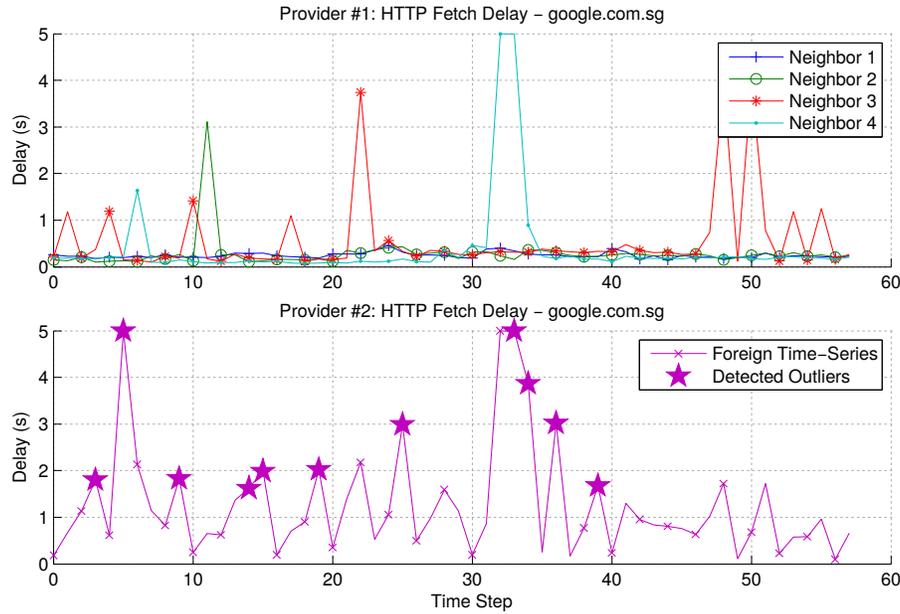


Figure 44: An illustration of two sets of time-series. Here, a time-series of HTTP fetch delays for `google.com.sg`, belonging to a device connected to Provider #2’s network, is mixed into that of the data set of time-series of fetch delays for devices on Provider #1’s network. We chose this foreign time-series from Provider #2 because it offers many points in time in which the delay experienced by the foreign device is radically different from those on Provider #1’s network. The goal of our algorithm is then to detect those points in time where the foreign time-series is deemed to be behaving “abnormally” from others. Those detected points are then marked out as shown.

There are also significant stretches during our experiments where one or two of our devices experience severe outages that are not observed by other devices. **Figure 43** illustrates this phenomenon exactly. In this scenario, the device `GT-P3100 #A` experiences a sudden stretch of high delays and timeouts that lasted more than 3 minutes. Besides `NEXUS-7 #A`, which also saw some spikes and a short outage of less than 30 seconds, the rest of the devices experienced delays that were fairly normal. In fact, the worst-ever time-series that we observed in a 60 sample window (lasting 5 minutes) had 58 fetch attempts which experienced time-outs. This is despite its 7 other neighbors observing 0, 0, 2, 2, 0, 3 and 0 time-outs respectively. In these types of scenarios, addressing the **QEC** question is

obviously useful. When the device is deemed to be performing “abnormally”, the participant can proceed to perform limited diagnostic checks, and possibly reboot his device to try and mitigate the outage.

In the real world, there can be many reasons why such outages happen. For example, the device could be physically damaged, or experiencing a rare and very unfortunate prolonged period of deep fading. Other reasons may include bugs in the firmware that manifest themselves after prolonged use (e.g. software aging), having errant malware that are hogging system resources, or simply having left a Virtual Private Network (VPN) connection switched on in the background. Identifying the root cause of such outages is beyond the scope of this dissertation.

5.6.6 Outlier Detection Performance

In the second step of our statistical framework introduced in Section 5.5, we consider a given delay time-series measured by a device, as well as those of the device’s co-located neighbors’. We identify points in time where the device is doing considerably worse than its neighbors, using QR. We term those points as outliers. The result of this approach can be seen in **Figure 44**. We took a time-series of delay measurements collected on Provider #2’s network, and used four other time-series collected on Provider #1’s network as those of its neighbors, and performed the regression analysis and outlier detection. The outliers that were detected by the algorithm are clearly marked out in **Figure 44**.

The algorithm patently detected points in time where delay spikes seen by the foreign device were not observed by its neighbors. More importantly, during periods where spikes *were* observed by its artificially-introduced neighbors, the algorithm adapts, and correctly classifies those measurements as inliers. This is evident in time-steps 22, 48, and 51.

5.6.7 Effectiveness of the Overall Algorithm

In the previous subsections, we illustrated results for selected sets of time-series that highlight the effects discussed above. Here, we setup the following experiment to test the effectiveness of the algorithm as a whole, on how accurate it is in detecting time-series that are known to be “abnormal”. In order to do this, for each of the set of 8 time-series (belonging to 8 devices) collected per position (over 24 positions), per experiment (over 2 experiments), per day (over 7 consecutive days) on Provider #1’s network, we introduce a foreign time-series that were collected at the same corresponding times by a device on Provider #2’s network. This resulted in 336 sets of 9 time-series (with 8 devices using Provider #1, and another SM-T325 using Provider #2). Each of the 336 foreign time-series was manually and individually inspected, and labeled as “normal” or “abnormal”. This is based on whether its median delay is closer to those of the SM-T325s’ on Provider #1’s network (labeled as “normal”), or those of SM-T325s’ on Provider #2’s network (labeled as “abnormal”), with reference to **Figure 40**.

Using a window length of 60 samples (collected in 5 minutes), we apply our 4-step algorithm on each set of 9 time-series. We vary the number of neighbors used in the regression analysis to also examine the effects of increasing the number of co-located participants on detection accuracy. The goal is then to check if the algorithm labels the foreign time-series correctly, or incorrectly (which we term as a mis-detection). **Figure 45** shows the results of this experiment. Using our approach, the median mis-detection rate is around 30% when only 2 neighbors are considered in the regression analysis. As more neighbors are introduced, the detection rate improves steadily. When 6 neighbors are used, the median mis-detection rate drops to only around 10%. In populated urban areas, we believe that having 6 or more co-located devices in many areas is often possible.

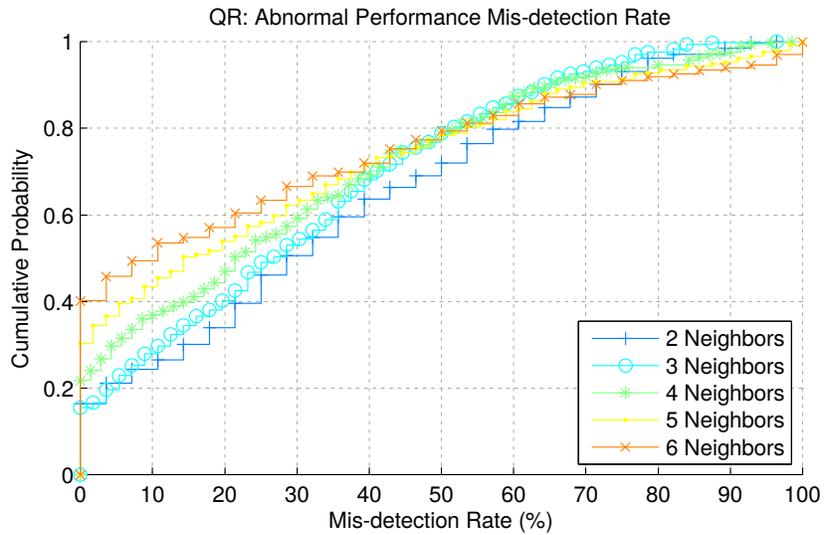


Figure 45: A complete characterization of the misdetection rates of a foreign time-series of HTTP fetch delays, belonging to Provider #2’s network, when mixed into the time-series of delays experienced by devices on Provider #1’s network. As the number of neighbors considered by the algorithm increases, the detection performance of our algorithm becomes better.

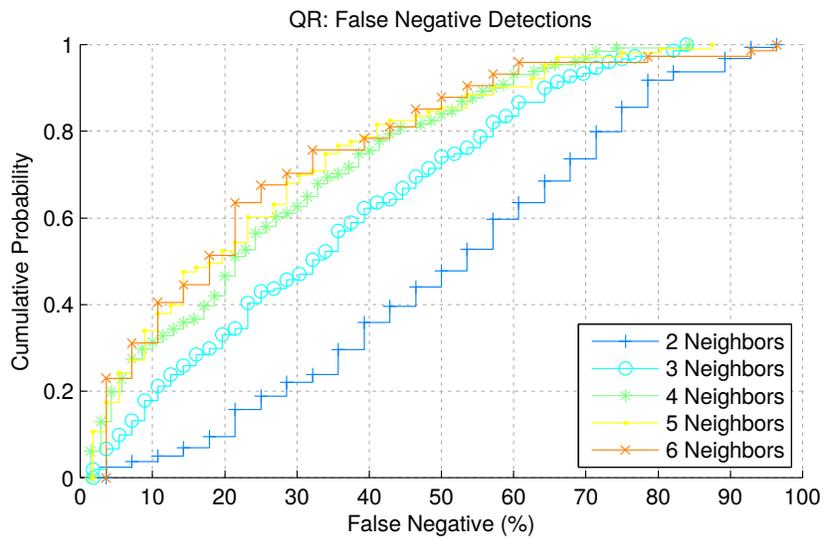


Figure 46: An illustration of the false negative rates of our algorithm. The false positive rate can be inferred by flipping the line-series about the (0,0)-(100,1) diagonal. Of all the times that a misdetection happens, having more neighbors helps to suppress the false negative rate, in favor of the false positive.

We do not show the results of considering 1, 7 or 8 neighbors because these combinations only result in 336 readings, which are insufficient to obtain a smooth cumulative distribution. However, with 7 neighbors, our algorithm can already reduce the median mis-detection rate to zero. With 8 neighbors, we can in fact detect the foreign time-series with 100% accuracy, in more than 80% of the time-series.

5.6.8 False Negatives vs. False Positives Performance

False negative mis-detection happens when a participant experiences very poor network delays compared to its neighbors, but the algorithm returns a “normal” verdict. This can be an exasperating experience because the user’s frustrations are not validated. So, we argue that false positives (where the network delay seen by the participant is actually comparable to that of its neighbors’, but the algorithm returns an “abnormal” verdict) are preferable in the context of **QEC**. **Figure 46** illustrates comprehensively the false negative rates demonstrated by our algorithm, for the same set of experiments. Having just 2 neighbors results in a median false negative rate of over 53%. This improves steadily as more neighbors are added into the regression analysis. With 6 neighbors, the median false negative rate drops to around 18%, which reduces the potential frustrations of participants.

Hence, taking the results presented in Section 5.6.7 and Section 5.6.8 together, we can see that with just 6 co-located participating neighbors, we can very effectively identify abnormal delays experienced by one or just a few devices with up to 90% accuracy in the median case. Of those 10% mis-detected delays, we can suppress the frustrating false negative detections to just 18%. These are very promising results, which suggest that our approach is viable for **QEC**.

5.7 Summary

In this chapter, we first describe the **QEC** question that should be asked by cellular service subscribers when they experience periods of poor network performance. This will assist in diagnosing the root cause of the performance problem from a users' perspective rather than relying solely on cellular operators. We describe how we use *Tattle*, a comprehensive, large-scale cellular network monitoring framework, to allow participating devices to glean networking performance information from one other in real-time. They do this by leveraging on their capabilities to opportunistically exchange network measurements, to preserve the context of co-location and conserve device power.

We then introduce and develop a robust statistical framework, based on Quantile Regression, which cumulates into a 4-step algorithm. It first derives a regression model based on a device and its neighbors' network delay measurements. Then, it identifies points in time where a device performs poorly compared to its neighbors. Next, it counts n , the number of these identified outliers in a small finite window. Finally, it performs hypothesis testing by deciding which one of two estimated geometric distributions (from which "normal", and "abnormal", numbers of outliers are drawn) that n most likely belongs to. Through this approach, we can directly perform **QEC**.

We validate our approach based on real-world measurements of network delay, collected using several devices of assorted makes and models. In total, we collected over 7,300 time-series of measurements, comprising over 443,500 samples. This includes measurements on 2 different providers' networks for comparison. We first show that at a sampling interval of 5 seconds, close to 80% of all the time-series demonstrated very little autocorrelation, so as to fulfill the necessary prerequisite of i.i.d in order to apply regression.

We then illustrate examples where “normal” and “abnormal” performances occur in real networks. We report that there are numerous real-world instances where a device can experience complete outage, while none of its neighbors are affected. We give quantitative results on how well our algorithm can detect an “abnormal” time series, with increasing effectiveness as the number of neighbors increases. With just 2 neighbors, we are able to achieve a median detection rate of just under 70%. With 6 co-located devices, we can achieve a median detection rate of just under 90%.

We also characterize the false negative and false positive tendencies of our algorithm. We demonstrate that our algorithm favors false positives, which is more desirable than frustrating false negatives in the context of **QEC**.

6. CONCLUSIONS AND FUTURE DIRECTIONS

In this dissertation, we investigate three broad areas of cellular monitoring, namely aggregate coverage monitoring (**CM**), aggregate quality monitoring (**QM**), and per-device quality estimation and classification (**QEC**), and describe a unifying framework that relies on peer-to-peer measurement exchange to address these three areas, with promising results. We now conclude with a summary of our main research contributions, and discuss some brief areas for future investigations.

6.1 Summary of Research Contributions

In Chapter 3, we propose Tattle, a comprehensive unifying framework that relies on participating subscribers to report on the network coverage conditions wherever they are. Tattle relies predominantly on opportunistic inter-device measurement exchange to preserve the co-location of measurements, as well as conserve device power, to build high-resolution aggregate coverage maps. Through this framework, operators can minimize expenditure on laborious walk- and drive-test regimes in order to ascertain coverage of their networks. They can quickly identify and react to micro-scale coverage issues (e.g. coverage holes in the order of meters or tens of meters) in a timely manner.

We show through real-life experiments that in urban built-up areas, the GPS locations reported by devices may have significant uncertainties, as well as errors that can sometimes be upwards of several kilometers from their true locations. We describe the U-CURE algorithm, and how it takes into account reported location uncertainties, as well as the knowledge of measurement co-location, to reduce and remove erroneously-localized readings. In the median case, using U-CURE can reduce pair-wise rms distance errors, for

a pair of devices physically less than 1 meter apart, from 40 meters, to less than 25 meters. Large extreme inter-pair errors that spans up to 3 kilometers can be brought down significantly, to less than 40 meters, by using U-CURE. We then demonstrate exactly how Tattle can be used for **CM** by using our prototype app, running on COTS devices. Through this, we collect more than 3.78 million RSCP measurements, and illustrated detailed, high-resolution coverage maps constructed from these data.

In Chapter 4, we move on from monitoring coverage, to monitoring the quality of network service (**QM**), using network delay as a key example metric of network quality of service. We report in detail the performance of localization techniques used by COTS Android mobile devices, based on more than 100,000 real-world data points. We establish that a strong correlation exists between network delay and localization error when the purported ‘High Accuracy’ mode of localization is used. This correlation manifests especially when a device undergoes high-speed mobility. For the Samsung SM-T325, this correlation results in increases of over 600 meters per 1,000 ms of network delay. This correlation motivates our introduction of the Delay-Adjusted U-CURE algorithm. Instead of safely discarding mis-localized data points, we leverage on the knowledge of correlation to reduce the localization errors of mis-localized data points. This approach does not require any location anchors or additional instrumentation. Our results are very promising, demonstrating that improvements, in terms of median rms location error reduction, of over 30% can be achieved with just 3 co-located participants. Upwards of 60% reduction in median rms location error can be observed with 6 co-located neighbors.

We then demonstrate how **QM** can be achieved with very good location fidelity and measurement accuracy. We present accurate and high-resolution aggregate network delay

maps constructed using over 34,000 real-world data points, collected using our COTS devices, with the Tattle framework and Delay-Adjusted U-CURE.

Then, in Chapter 5, we take a different approach to monitoring, and shift our focus from aggregate network monitoring, to considering observed network performance on a per-device basis. By using the Tattle framework, we empower devices to collect network quality measurements (in the form of network round-trip delay) from their co-located neighbors. Through this, they can then derive context about their own observed performances. We develop a complete and flexible statistical framework based on quantile regression and outlier classification. It systematically estimates and classifies a device's observed network delay (**QEC**) as “normal” or “abnormal”. This allows an affected subscriber to very quickly determine if his or her observed impairment is an isolated one, or also suffered by others in the same area. In this way, if the former is determined, then the subscriber can take immediate remedy steps (such as rebooting his or her device) to try and alleviate the impairment. If it is determined that others also suffer from long network delays, then subscribers can quickly attain some level of closure and validation to their negative experiences. They can also gain some comfort in knowing that their devices are working fine.

We validate our approach by using real-world measurements of over 7,300 network delay time-series, comprising over 443,500 samples. Then, using our statistical framework, we can very effectively identify abnormal delays with up to 90% accuracy in the median case with just 6 co-located participants. Of those 10% mis-detected delays, we can suppress frustrating false negative detection rates to just 18%. These are very promising results for **QEC**.

6.2 Future Directions

The technology behind smart devices continues to advance at an exhilarating pace. By the time this dissertation is published, there is no doubt that manufacturers would have already announced the next generation of their smart devices that boast faster processors, more memory and new features. However, Tattle's principle of relying upon inter-device communication to improve location estimates, as well as derive context of localized network performance, will continue to be applicable for the foreseeable future. This remains until realistic and affordable localization techniques with very tight guaranteed bounds, as well as advanced fault monitoring and management modules, are available in COTS devices. However, there are several directions that can be further explored in relation to the work presented in this dissertation.

6.2.1 Applicability to Indoor Positioning Techniques

Thus far, we have seen how well Tattle can work outdoors, where there is a good chance where some or at least one device can be accurately-localized with small location uncertainty. However, current generations of smart devices still lack the capability to be localized indoors in a reliable fashion. When indoor positioning systems with reasonable levels of error bounds become commercially feasible, we will be eager to apply the Tattle framework to these systems to see how much gains can be achieved to further reduce location uncertainty. In such cases, fine-grained indoor coverage monitoring becomes possible. It will allow operators to become even more agile in addressing network issues. Small-scale cellular network technologies (such as distributed antennas, micro-cells, pico-cells, and femto-cells) can be rapidly deployed in a complementary fashion wherever Tattle detects coverage holes and areas with endemic network quality problems.

6.2.2 Automatic Coverage Adaptation through Network Reconfiguration

The goal of this dissertation is to design a framework for cellular network monitoring, and hence we proposed and implemented Tattle to achieve this objective. Another important area of network management that was not considered in this thesis is that of network configuration. When **CM** and **QM** become sufficiently adopted, dynamic and self-adapting reconfigurations of network parameters, such as antenna pan, antenna tilt, and transmit power, may happen in real-time. This is in response to the timely insights gathered through **CM** and **QM**. If adaptive self-configuration, and subsequently self-optimization of cellular networks is realized, the potential impact on the operating expenditure of cellular operators is tremendous. We believe that a fine-grained monitoring framework such as Tattle serves as an important first step to realizing the ultimate vision of self-optimizing cellular networks.

6.2.3 Heterogeneous Networks

Recently, the concept of heterogeneous networks is piquing considerable interest from academia, governments, regulatory authorities, network operators, service providers, and vendors. The vision of a heterogeneous network is one where multiple radio technologies and interfaces of various differing capabilities, spanning across different providers and infrastructure management domains, all inter-operate to provide subscribers with a single, logical and coherent wireless link, regardless of wherever they are. To realize such a vision, we believe that the Tattle framework can be extended to monitor the performances of all these radio technologies. It can then provide a consistent, timely, and accurate basis in order to support rapid and predictive radio switching techniques. Thus, it may ensure seamless connectivity, even for subscribers under high-speed mobility.

6.2.4 Validation-as-a-Service for Other Mobile Applications

Many mobile applications require some form of validation for the data that is generated by the subscriber. For example, loyalty-reward tracking applications may offer subscribers perks, rewards and coupons based on their spending and shopping trends, locations and patterns. However, some unscrupulous subscribers may engage in deceitful behavior to earn these perks by falsifying their reported locations (e.g. using apps that falsify GPS locations). Tattle inherently handles these types of location-spoofing by validating a user's reported location with others in his or her vicinity. If the location is patently wrong, application developers can detect these errant users and suspend their privileges. This will work unless a majority of users engage in such behaviors.

REFERENCES

- [1] 3rd Generation Partnership Project, “*3GPP TS 25.215 V12.0.0 (2014-09); Physical layer; Measurements (FDD) (Release 12)*”, 3GPP, 2014.
- [2] 3rd Generation Partnership Project, “*Keywords & Acronyms – UMTS*”, 2014. <http://www.3gpp.org/technologies/keywords-acronyms/103-umts>
- [3] 3rd Generation Partnership Project, “*3GPP TS 25.308 V12.1.0 (2014-09); High Speed Downlink Packet Access (HSDPA); Overall description; Stage 2 (Release 12)*”, 3GPP, 2014.
- [4] 3rd Generation Partnership Project, “*Keywords & Acronyms – HSPA*”, 2014. <http://www.3gpp.org/technologies/keywords-acronyms/99-hspa>
- [5] 3rd Generation Partnership Project, “*Keywords & Acronyms – LTE*”, 2014. <http://www.3gpp.org/technologies/keywords-acronyms/98-lte>
- [6] “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018”, Cisco, 2014.
- [7] “Gartner Says Mobile App Stores Will See Annual Downloads Reach 102 Billion in 2013”, Gartner, 2013. <http://www.gartner.com/newsroom/id/2592315>
- [8] T. S. Rappaport, “*Wireless Communications: Principles and Practice*”, 2nd edition, Prentice Hall, 2012.
- [9] H. Liang, H.S. Kim, H.P. Tan, W.L. Yeow, “*I've heard you have problems: Cellular signal monitoring through UE participatory sensing*”, GLOBECOM '14, 2014.
- [10] S. Guha, R. Rastogi, K. Shim, “*CURE: an efficient clustering algorithm for large databases*”, ACM SIGMOD '98, 1998.
- [11] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, M. B. Srivastava, “*Participatory Sensing*”, ACM WSW '06, 2006.
- [12] E. Koukoumidis, M. Martonosi and L.-S. Peh, “*Leveraging Smartphone Cameras for Collaborative Road Advisories*”, IEEE Transactions on Mobile Computing, vol. 11, no. 5, pp. 707-723, 2012.
- [13] J. Pang, B. Greenstein, M. Kaminsky, D. McCoy, S. Seshan, “*Wifi-Reports: Improving Wireless Network Selection with Collaboration*”, ACM MobiSys '09, 2009.

- [14] S. Reddy, K. Shilton, G. Denisov, C. Cenizal, D. Estrin, M. Srivastava, “*Biketastic: Sensing and Mapping for Better Biking*”, ACM CHI ‘10, 2010.
- [15] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, T. F. Abdelzaher, “*GreenGPS: A Participatory Sensing Fuel-Efficient Maps Application*”, ACM MobiSys ‘10, 2010.
- [16] P. F. Zhou, Y. Zheng, M. Li, “*How Long to Wait? Predicting Bus Arrival Time with Mobile Phone based Participatory Sensing*”, ACM MobiSys ‘12, 2012.
- [17] 3rd Generation Partnership Project, “*3GPP TS 32.101 V12.0.0 (2014-09); Telecommunication management; Principles and high level requirements (Release 12)*”, 3GPP, 2014.
- [18] G. Koutitas, A. Karousos, L. Tassiulas, “*Deployment Strategies and Energy Efficiency of Cellular Networks*”, IEEE Transactions on Wireless Communications, vol. 11, no. 7, pp. 2552-2563, July 2012.
- [19] F. Baccelli, B. Błaszczyszyn, “*Stochastic Geometry and Wireless Networks*”, Foundations and Trends in Networking, vol. 3, no. 3-4, pp. 249-449, 2009.
- [20] 3rd Generation Partnership Project, “*3GPP TS 37.320 v12.2.0 (2014-09); Radio measurement collection for Minimization of Drive Tests (MDT) (Release 12)*”, 3GPP, 2014.
- [21] 3rd Generation Partnership Project, “*3GPP TS 36.331 v12.3.0 (2014-09); Radio Resource Control (RRC); Protocol specification (Release 12)*”, 3GPP, 2014.
- [22] 3rd Generation Partnership Project, “*3GPP TS 36.314 v12.0.0 (2014-09); Layer 2 – Measurements (Release 12)*”, 3GPP, 2014.
- [23] 3rd Generation Partnership Project, “*3GPP TS 32.450 v12.0.0 (2014-10); Key Performance Indicators (KPI) for E-UTRAN: Definitions (Release 12)*”, 3GPP, 2014.
- [24] 3rd Generation Partnership Project, “*3GPP TS 32.111-1 v12.0.0 (2013-06); Fault Management; Part 1: 3G fault management requirements (Release 12)*”, 3GPP, 2013.
- [25] 3rd Generation Partnership Project, “*3GPP TS 36.305 v12.1.0 (2014-07); Stage 2 functional specification of User Equipment (UE) positioning in E-UTRAN (Release 12)*”, 3GPP, 2014.

- [26] T. Kürner et al., “*Final Report on Self-Organisation and its Implications in Wireless Access Networks*”, FP7 Self-Optimisation and Self-Configuration in Wireless Networks Project, Deliverable 5.9, 2010.
- [27] International Telecommunication Union, “*Propagation data and prediction methods for the planning of indoor radiocommunication systems and radio local area networks in the frequency range 900 MHz to 100 GHz*”, ITU-R P.1238-4, 2005.
- [28] X. Liu, M. Seshadri, A. Sridharan, H. Zang, S. Machiraju, “*Experiences in a 3G network: interplay between the wireless channel and applications*”, ACM MobiCom ‘08, 2008.
- [29] W. L. Tan, F. Lam, W. C. Lau, “*An Empirical Study on the Capacity and Performance of 3G Networks*”, IEEE Transactions on Mobile Computing, vol. 7, no. 6, pp. 737-750, 2008.
- [30] S. Goksel, “*Optimization and Log File Analysis in GSM*”, 2003. Retrieved Online.
- [31] Agilent Technologies, “*Optimizing Your GSM Network Today and Tomorrow – Using Drive-Testing to Troubleshoot Coverage, Interference, Handover Margin and Neighbor Lists, Application Note 1344*”, 2000. Retrieved Online.
- [32] Infocomm Development Authority of Singapore, “*MyConnection SG - Crowdsourcing Consumers' Mobile Broadband Experience*”, 2014. <http://www.ida.gov.sg/Individuals-and-Community/Infocomm-You/MyConnection-SG>
- [33] RootMetrics, 2014. <http://www.rootmetrics.com/us/product>
- [34] OpenSignal, 2014. <http://opensignal.com/>
- [35] “*Emergency Communications*”, Federal Communications Commission, <http://www.fcc.gov/guides/emergencycommunications>
- [36] “*Cellular network-based location system*”, United States Patent US 5519760 A, May 1996.
- [37] J.J. Caffery, G.L. Stuber, “*Overview of radiolocation in CDMA cellular systems*”, IEEE Communications Magazine, vol. 36, no. 4, pp. 38–45, April 1998.
- [38] F. Gustafsson, F. Gunnarsson, “*Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements*”, IEEE Signal Processing Magazine, vol. 22, no. 4, pp. 41-53, July 2005.

- [39] M. Z. Win, A. Conti, S. Mazuelas, Y. Shen, W.M. Gifford, D. Dardari, M. Chiani, “*Network Localization and Navigation via Cooperation*”, IEEE Communications Magazine, vol. 49, no. 5, pp. 56 - 62, May 2011.
- [40] H. Wymeersch, J. Lien, M. Z. Win, “*Cooperative localization in wireless networks*”, Proceedings of the IEEE, vol. 97, no. 2, pp. 427–450, February 2009.
- [41] A. Conti, M. Guerra, N. Decarli, M. Z. Win, “*Network Experimentation for Cooperative Localization*”, IEEE Journal on Selected Areas in Communications, vol. 30, no. 2, pp. 467 – 475, February 2012.
- [42] S. Gezici, Z. Tian, G.B. Biannakis, H. Kobayashi, A.F. Molisch, H.V. Poor, Z. Sahinoglu, “*Localization via Ultra-Wideband Radios: A look at positioning aspects of future sensor networks*”, IEEE Signal Processing Magazine, vol. 22, no. 4, pp. 70-84, July 2005.
- [43] B. Cole, “*Real-time Android: real possibility, really really hard to do - or just plain impossible?*” <http://www.embedded.com/electronics-blogs/cole-bin/43728702/Realtime-Android--real-possibility--really-really-hard-to-do---or-just-plain-impossible-->
- [44] X. Zhu, Q. Li, G. Chen, “*APT: Accurate outdoor pedestrian tracking with smartphones*”, IEEE INFOCOM ‘13, 2013.
- [45] M.B. Kjærgaard, S. Bhattacharya, H. Blunck, P. Nurmi, “*Energy-efficient trajectory tracking for mobile devices*”, ACM MobiSys ‘11, 2011.
- [46] M.B. Kjærgaard, J. Langdal, T. Godsk, T. Toftkjær, “*EnTracked: Energy-Efficient Robust Position Tracking for Mobile Devices*”, ACM MobiSys ‘09, 2009.
- [47] M. Azizyan, I. Constandache, R.R. Choudhury, “*SurroundSense: Mobile Phone Localization via Ambience Fingerprinting*”, ACM MobiCom ‘09, 2009.
- [48] R. Harle, “*A Survey of Indoor Inertial Positioning Systems for Pedestrians*”, IEEE Communications Surveys & Tutorials, vol. 15, no. 3, pp. 1281-1293, 2013.
- [49] P. Bahl, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, B. Zill, “*DAIR: A Framework for Managing Enterprise Wireless Networks Using Desktop Infrastructure*”, ACM HotNets ‘05, 2005.
- [50] A. Adya, P. Bahl, R. Chandra, L. Qiu, “*Architecture and Techniques for Diagnosing Faults in IEEE 802.11 Infrastructure Networks*”, ACM MobiCom ‘04, 2004.

- [51] R. Barco, P. Lazaro, L. Diez, V. Willie, “*Continuous versus Discrete Model in Autodiagnosis Systems for Wireless Networks*”, IEEE Transactions on Mobile Computing, vol. 7, no. 06, June 2008.
- [52] R. Barco, V. Wille, L. Diez, M. Toril, “*Learning of model parameters for fault diagnosis in wireless networks*”, Wireless Networks, vol. 16, no. 1, January 2010.
- [53] P. Szilagyi, S. Novaczki, “*An Automatic Detection and Diagnosis Framework for Mobile Communication Systems*”, IEEE Transactions on Network and Service Management, vol. 09, no. 02, June 2012.
- [54] 3rd Generation Partnership Project, “*3GPP TS 25.133 v12.5.0 (2014-10); Requirements for support of radio resource management (FDD) (Release 12)*”, 3GPP, 2014.
- [55] “*About Bluetooth® Low Energy Technology*”, <http://www.bluetooth200b.com/Pages/low-energy-tech-info.aspx>, Bluetooth SIG, Inc., 2014.
- [56] “*Wi-Fi Direct*”, <http://www.wi-fi.org/discover-wi-fi/wi-fi-direct>, Wi-Fi Alliance, 2014.
- [57] “*LTE Direct*”, <http://www.qualcomm.com/research/projects/lte-direct>, Qualcomm, 2014.
- [58] L. G. Jaimes, I. J. Vergara-Laurens, M. A. Labrador, “*A location-based incentive mechanism for participatory sensing systems with budget constraints*”, IEEE PerCom ‘12, 2013.
- [59] T. Luo, H. P. Tan, “*Profit-Maximizing Incentive for Participatory Sensing*”, IEEE INFOCOM ‘14, 2014.
- [60] J. S. Lee, B. Hoh, “*Sell your experiences: A market mechanism based incentive for participatory sensing*”, IEEE PerCom ‘10, 2010.
- [61] D. Yang, G. Xue, X. Fang, J. Tang, “*Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing*”, ACM MobiCom ‘12, 2012.
- [62] I. Koutsopoulos, “*Optimal incentive-driven design of participatory sensing systems*”, IEEE INFOCOM ‘13, 2013.
- [63] E. De Cristofaro, C. Soriente, “*Participatory privacy: Enabling privacy in participatory sensing*”, IEEE Network, vol. 27, no. 1, pp. 32-36, 2013.
- [64] P. Gilbert, L. P. Cox, J. Jung, D. Wetherall, “*Toward trustworthy mobile sensing*”, ACM HotMobile ‘10, 2010.

- [65] A. Dua, N. Bulusu, W. C. Feng, W. Hu, “*Towards trustworthy participatory sensing*”, USENIX HotSec ‘09, 2009.
- [66] S. Saroiu, A. Wolman, “*I Am a Sensor, and I Approve This Message*”, ACM HotMobile ‘10, 2010.
- [67] Trusted Computing Group, “*Trusted Platform Module Library Specification, Family "2.0", Level 00, Revision 01.16 - October 2014*”, http://www.trustedcomputinggroup.org/resources/tpm_library_specification, 2014.
- [68] Trusted Computing Group, “*Mobile Trusted Module 2.0 Use Cases*”, http://www.trustedcomputinggroup.org/resources/mobile_trusted_module_20_use_cases, 2013.
- [69] X. Wang, W. Cheng, P. Mohapatra, T. Abdelzaher, “*ARTSense: Anonymous Reputation and Trust in Participatory Sensing*”, IEEE INFOCOM ‘13, 2013.
- [70] National Marine Electronics Association, “*NMEA 0183 Standard*”, http://www.nmea.org/content/nmea_standards/nmea_0183_v_410.asp, 2014.
- [71] Official Android Developers’ Guide, “*Location Strategies*”, <http://developer.android.com/guide/topics/location/strategies.html>, 2014.
- [72] National Geospatial-Intelligence Agency, “*Grids and Reference Systems*”, <http://earth-info.nga.mil/GandG/coordsys/grids/referencesys.html>, 2013.
- [73] A. Kumar, “*Mobile Broadcasting with WiMAX: Principles, Technology, and Applications*”, 1st edition, Focal Press, 2008.
- [74] I. Grigorik, “*High Performance Browser Networking*”, 1st edition, O’Reilly Media, 2013.
- [75] J. Nielsen, “*Usability Engineering*”, 1st edition, Morgan Kaufmann, 1993.
- [76] S. C. Seow, “*Designing and Engineering Time: The Psychology of Time Perception in Software*”, 1st edition, Addison-Wesley Professional, 2008.
- [77] I. Rish, M. Brodie, N. Odintsova, S. Ma, G. Grabarnik, “*Real-time Problem Determination in Distributed Systems using Active Probing*”, IEEE/IFIP NOMS ‘04, 2005.
- [78] S. von Watzdorf, F. Michahelles, “*Accuracy of Positioning Data on Smartphones*”, ACM LocWeb ‘10, 2010.
- [79] B. Hess, A.Z. Farahani, F. Tschirschnitz, F. von Reischach, “*Evaluation of Fine-Granular GPS Tracking on Smartphones*”, ACM MobiGIS ‘12, 2012.

- [80] C. Bauer, “*On the (In-)Accuracy of GPS Measures of Smartphones: A Study of Running Tracking Applications*”, ACM MoMM ‘13, 2013.
- [81] N. Vallina-Rodriguez, J. Crowcroft, A. Finamore, Y. Grunenberger, K. Papagiannaki, “*When assistance becomes dependence: characterizing the costs and inefficiencies of A-GPS*”, ACM SIGMOBILE Mobile Computing and Communications Review archive, vol. 17, no. 4, pp. 03-14, October 2013.
- [82] R.S. King, M. Gröndahl, “*How Google Collected Data From Wi-Fi Networks*”, The New York Times, 23 May 2012. <http://www.nytimes.com/interactive/2012/05/23/business/How-Google-Collected-Data-From-Wi-Fi-Networks.html>
- [83] S.J. Vaughan-Nichols, “*How Google--and everyone else--gets Wi-Fi location data*”, ZDNet, 16 November 2011. <http://www.zdnet.com/blog/networking/how-google-and-everyone-else-gets-wi-fi-location-data/1664>
- [84] G. Fleishman, “*How the iPhone knows where you are*”, Macworld, 28 April 2011. http://www.macworld.com/article/1159528/how_iphone_location_works.html
- [85] A. Konstantinidis, G. Chatzimiloudis, C. Laoudias, S. Nicolaou, D. Zeinalipour-Yazti, “*Towards planet-scale localization on smartphones with a partial radiomap*”, ACM HotPlanet ‘12, 2012.
- [86] The Android Open Source Project, 2014. <https://source.android.com/>
- [87] AndroidXRef, 2014. http://androidxref.com/4.4.4_r1/xref/frameworks/base/services/java/com/android/server/location/
- [88] IDC, “*Smartphone OS Market Share, Q3 2014*”, 2014. <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
- [89] IDC, “*Smartphone Vendor Market Share, Q3 2014*”, 2014. <http://www.idc.com/prodserv/smartphone-market-share.jsp>
- [90] Official Android Developers’ Guide, “*Location Strategies*”, 2014. <http://developer.android.com/guide/topics/location/strategies.html>
- [91] Official Android Developers’ Guide, “*Location*”, 2014. [http://developer.android.com/reference/android/location/Location.html#getAccuracy\(\)](http://developer.android.com/reference/android/location/Location.html#getAccuracy())
- [92] J. Korhonen, “*Introduction to 3G Mobile Communications*”, 2nd edition, Artech House Publishers, 2003.

- [93] European Telecommunications Standards Institute, TS 101 626 V7.0.0, “*Digital cellular telecommunications system (Phase 2+); Network Identity and Timezone (NITZ); Service description, Stage 1*”, 1999.
- [94] “*NTP: The Network Time Protocol*”, <http://www.ntp.org/>
- [95] “*What's the fastest smartphone 2014: processor, graphics, web performance comparison*”, PC Advisor UK, 30 July 2014, <http://www.pcadvisor.co.uk/test-centre/mobile-phone/3515533/whats-fastest-smartphone-2014/?pn=2>
- [96] H. Jiang, Z. Liu, Y. Wang, K. Lee, I. Rhee, “*Understanding bufferbloat in cellular networks*”, ACM SIGCOMM CellNet '12, 2012.
- [97] R. Koenker, K. Hallock, “*Quantile Regression: An Introduction*”, Journal of Economic Perspectives, Vol. 15, No. 4, 2001.
- [98] A. Nardi, M. Schemper, “*New residuals for Cox regression and their application to outlier screening*”, Biometrics, Vol. 55, No. 2, Jun 1999.
- [99] S. Kay, “*Fundamentals of Statistical Signal Processing, Volume II: Detection Theory*”, Prentice Hall, 1998.