ENHANCING MOBILITY SUPPORT IN CELLULAR NETWORKS WITH DEVICE-SIDE INTELLIGENCE

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Haotian Deng

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2020

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Chunyi Peng, Chair
Department of Computer Science
Dr. Bharat K. Bhargava
Department of Computer Science
Dr. Sonia A. Fahmy
Department of Computer Science
Dr. Kihong Park
Department of Computer Science

Approved by:

Dr. Clifton W. Bingham
Head of the School Graduate Program

To my parents.

ACKNOWLEDGMENTS

I would like to express my deep gratitude to my advisor, Prof. Chunyi Peng, for her never-ending supervision, support and patience and trust throughout my PhD study. I have learned a lot from her to do great research. I am very grateful for her high standard for research and hands-on training for me to proceed with new research ideas, develop intellectual contributions and deliver intermediate progress firmly and steadily, write high-quality papers and disseminate our work to the community.

I would like to thank my PhD advisory committee members, Prof. Bharat Bhargava, Prof. Sonia Fahmy and Prof. Kihong Park for their constructive feedback and help. I also thank Prof. Y. Charlie Hu for his valuable advices and feedback on my preliminary study.

Special thanks goes to Prof. Dimitrios Koutsonikolas (University at Buffalo) who opened my door to academic research adventure. I also would like to thank Dr. Xin Wu for his kind mentoring during my research internship at Alibaba Group.

Throughout my graduate study, I was also fortunate to work with excellent people from University at Buffalo, Ohio State University, University of California, Los Angeles and Purdue University: Dr. Li Sun, Jie Zhao, Dr. Yuanjie Li, Jiayao Li, Zengwen Yuan, Qianru Li, Ans Fida, Jiayi Meng, Weicheng Wang, Jiawei Lyu, Kai Ling and Jingqi Huang. I have learned a lot from our collaboration and teamwork.

My deepest gratitude goes to my family for their unconditional love and support. Thank Xingya Zhao for accompanying me in this journey. Thank my parents for their love and support throughout my life. I hope both of you are proud of me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

ABBREVIATIONS

CA	Carrier Aggregation
CSFB	Circuit Switch Fallback
EARFCN	E-UTRA Absolute Radio Frequency Channel Number
GSM	Global System for Mobile Communications
HSPA	High Speed Packet Access
LTE	Long-Term Evolution
MIMO	Multiple-Input and Multiple-Ontput
MPTCP	Multipath Transmission Control Protocol
NAS	Non-access Stratum
RAT	Radio Access Technology
RRC	Radio Resource Control
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
SDN	Software Defined Networking
SIB	System Information Block
SNR	Signal-To-Noise Ratio
SON	Self-Organizing Network
UARFCN	UTRA Absolute Radio Frequency Channel Number
UMTS	Universal Mobile Telecommunications Service
USRP	Universal Software Radio Peripheral
WCDMA	Wideband Code Division Multiple Access

ABSTRACT

Deng, Haotian Ph.D., Purdue University, December 2020. Enhancing Mobility Support in Cellular Networks with Device-Side Intelligence. Major Professor: Chunyi Peng.

Internet goes mobile as billions of users are accessing the Internet through their smartphones. Cellular networks play an essential role in providing "anytime, anywhere" network access as the only large-scale wireless network infrastructure in operation. Mobility support is the salient feature indispensable to ensure seamless Internet connectivity to mobile devices wherever the devices go or are. Cellular network operators deploy a huge number of cell towers over geographical areas each with limited radio coverage. When the device moves out of the radio coverage of its serving cell(s), mobility support is performed to hand over its serving cell(s) to another, thereby ensuring uninterrupted network access.

Despite a large success at most places, we uncover that state-of-the-practice mobility support in operational cellular networks suffers from a variety of issues which result in unnecessary performance degradation to mobile devices. In this thesis, we dive into these issues in today's mobility support and explore possible solutions with no or small changes to the existing network infrastructure.

We take a new perspective to study and enhance mobility support. We directly examine, troubleshoot and enhance the underlying procedure of mobility support, instead of higher-layer (application/transport) exploration and optimization in other existing studies. Rather than clean slate network-side solutions, we focus on device-side solutions which are compatible with 3GPP standards and operational network infrastructure, promising immediate benefits without requiring any changes on network side.

In particular, we address three technical questions by leveraging the power of the devices. *First, how is mobility support performed in reality?* We leverage device-side observation to monitor the handoff procedures that happen between the network and the device. We unveil that operator-specific configurations and policies play a decisive role under the standard mechanism and conduct a large-scale measurement study to characterize the extremely complex and diverse handoff configurations used by global operators over the world. *Second, what is wrong with the existing mobility support?* We conduct model-based reasoning and empirical study to examine network performance issues (*e.g.*, handoff instability and unreachability, missed performance) which are caused by improper handoffs. *Finally, how to enhance mobility support?* We turn passive devices to proactive devices to enhance mobility support. Specifically, we make a showcase solution which exploits device-side inputs to intervene the default handoff procedure and thus indirectly influence the cell selection decision, thereby improving data speed to mobile devices. All the results in this thesis have been validated or evaluated in reality (over top-tier US carriers like AT&T, Verizon, T-Mobile, some even in global carrier networks).

1 INTRODUCTION

Mobile networks have become one critical infrastructure to provide Internet access as a utility service, like water, gas and electricity, indispensable to our daily life. It provides ubiquitous Internet services to support a wide variety of applications on mobile phones and empower more thrilling usage scenarios such as vehicle to everything communication (V2X), virtual/augmented/mixed reality, internet access on high-speed railway, cellular connected drones and to name many.

The success of mobile networks must give credit to its capability of supporting anytime and anywhere network access, no matter where the users are and whether they are static, walking, driving or even flying. Wide-area mobility support is the key enabler. The radio signal coverage of single cell is limited, *e.g.*, the radius of a typical LTE cell in urban area is less than 1 km. In order to enable seamless network access during move, the mobile device needs to switch the serving cell from one to another when user moves out of the coverage of original serving cell. The mechanism used to switch between serving cells is called handoff. In this dissertation, we focus on this salient feature and seek to enhance mobility support in operational cellular networks. Our target is to provide better mobility support experiences using a device-side approach which is compatible with existing cellular networks instead of making a clean slate design.

This research is driven by findings that end users could suffer from poor network service caused by improper mobility support. For instance, users may find themselves encounter the following awkward situations at specific moments: 1) user device may lose cellular network connection from time to time even when user is in a location with acceptable radio signal strength; 2) a user may find the cell phone browser cannot open a web page as smooth as usual, sometimes it happens when the cell phone gets stuck on the previous generation of mobile network like 3G network and

is limited by network capability, and sometimes it is already connected to the latest LTE network but still has the same observation; 3) when a user is travelling in a vehicle on highway and watching videos, YouTube suddenly picks the lowest video quality at some sections, which is not what we want. We blame these issues on state-of-the-practice mobility support mainly for two reasons. First, the mobility support is implemented in a distributed way by design which has many benefits including easier to deploy and lower management overhead in the first place. However, without a centric controller, it leaves possibility for uncoordinated handoff behaviors which may contradict the original goal of mobility support. Second, the mobility support mechanism is designed for providing seamless network access without optimizing for data performance. As a result, the network capability could be under utilized in reality.

Driven by the fact that academic community has very limited access to industrial operational cellular networks, we take a device-centric approach to study mobility support and propose device-side solutions which are completely compatible with existing cellular network systems without requiring any changes at network side. We look for solutions which can bring benefits to end users immediately. Furthermore, today's mobile devices have become more and more powerful. It also makes it possible for us to utilize device-side intelligence and make mobile device play a more important role in network systems.

To provide better mobility support experiences of today's cellular network, we raise the following three technical questions:

1) How does mobility support perform in reality?

2) Does mobility support go wrong? If yes, how and why?

3) How can we improve the mobility support?

We follow the roadmap as illustrated in Fig. 1.1 to study state-of-the-practice mobility support and give answers to these questions.

First, we conduct an in-depth study on cellular mobility support and examine how a handoff is performed. We give background of a basic handoff procedure in §2.1. We

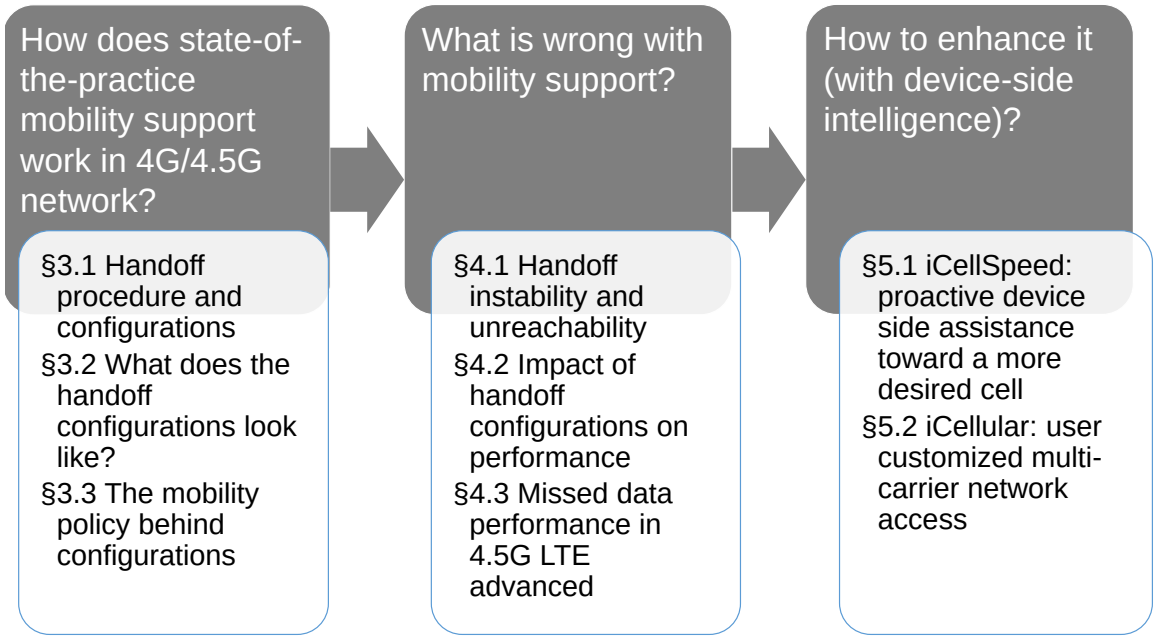


Figure 1.1.: Roadmap of this dissertation.

look into the persistent and structural factors, various handoff configurations, which determine every single step in a handoff procedure as regulated by 3GPP standard in §2.2. We unveil that real-world handoff configurations are extremely complex and diverse and allow micro-level mobility management in the wild. Furthermore, we unveil the secrets of carriers in managing mobility support (how to deploy/update handoff configurations) in their networks by inferring the majority of handoff configuration logic and algorithms in §2.3. We show that mobility management is a dynamic and adaptive process over time.

Second, we examine the performance of mobility support from three perspectives. We first check the function correctness of handoff and identify two issues: handoff instability and handoff unreachability in §3.1. By conducting a systematic study on handoff configurations, we derive and conduct formal analysis on handoff instability/unreachability conditions, and further validate the existence of undesired handoff behaviors and their impacts to end users. Next, we examine the impacts of handoff configurations on data performance in moving in §3.2. We find that these configu-

rations affect radio signal and performance as expected by design, but not all the impacts are intuitively "positive". Finally, we examine the data performance of end users in §3.3 and expose that a significant portion of performance gain is missed in reality and it is not uncommon. Our measurement study shows that, instead of transient factors (e.g., radio and network loads), the set of serving cells plays a persistent role and dominates data performance. We show that the current practice by mobile operators on selecting serving cell(s) should be held accountable.

Third, we seek to improve data performance by taking actions at device side to influence the default mobility support behaviors by utilizing the intelligence of today's smart mobile devices. To address the network underutilization issue caused by sub-optimal serving cells selection, we propose iCellSpeed which takes a "device-assisted, infrastructure-decided" approach and seeks to reach the performance-oriented global optimum cell selection from all candidate cells in §4.1. We implement iCellSpeed on commodity smartphones, and confirm its effectiveness with large speed gains. iCellSpeed increases user access speed by more than 10 Mbps at 79.2% of test locations ($>$ 25 Mbps at 29.2% of locations, up to 80.6 Mbps). In addition, we unveil a similar issue in multi-carrier network that the full benefits of multi-carrier network access are constrained by design in §4.2. To solve it, we propose iCellular, a client-side service to let mobile devices customize their own cellular network access. Complementing the design of Google Fi, iCellular further leverages low-level, runtime cellular information at the device side and online performance learning to select the best mobile carrier and achieve 3.74x throughput improvement and 1.9x latency reduction on average.

2 HOW DOES MOBILITY SUPPORT PERFORM IN REALITY?

In this chapter, we study how does mobility support in operational cellular network perform. We first introduce necessary background of handoff, the key mechanism to implement mobility support in cellular network and elaborate how mobility configurations play an important role to determine each step in a handoff procedure. We next study the deployed mobility configurations through a global-scale measurement study and show how complexity and diversity they are. We finally unveil the mobility policy of each operator by exploring the logic behind dynamic updating of mobility configurations in handoff.

2.1 Policy-Based Handoff Primer

To maintain network access in mobile state, a mobile device needs to switch the connected cell from one cell tower (also known as base station) to another. Such cell switch is achieved through a mechanism called handoff. Although the industry standard 3GPP regulates the general handoff procedure, there are tuning space left for mobile operators to implement policy-based handoff with operator specific strategy. As a result, the mobility configurations deployed by mobile operators play an important role to determine the handoff action taken by mobile devices and network together. In this section, we give basic background of policy-based handoff.

Cellular networks deploy many overlapping cells across geographic areas in reality. Multiple cells are accommodated by a cell tower with directional antennas and each cell has a specific radio coverage area. At a given location, a mobile device is served by at least one cell but covered by multiple cells in proximity. A cell is a logical unit that communicate with mobile devices over a contiguous radio spectrum frequency block with specific size. Such contiguous radio spectrum frequency block is also referred

as a radio frequency channel (or component carrier in context of carrier aggregation) and the radio spectrum size is referred as channel bandwidth. The cells may use distinct radio access technologies (RATs) including 2G, 3G, 4G and 5G (see [1] for a complete list of channels used by different RATs). Depending on the used frequencies and RATs, handoff can happen over: intra-freq (between two cells using the same RAT and frequency channel), inter-freq (between two cells using the same RAT but different channels), and inter-RAT (between two cells using different RATs).

2.1.1 Handoff Procedures

Considering whether an active radio resource control (RRC) connection is established when handoff happens, handoffs are generally classified into two categories: idle-state handoff (also referred as cell selection/reselection) and active-state handoff (also referred as handover), depending on whether the device is at the idle/active state without/with user traffic (an active RRC connection).

Active/Idle State Handoff. The idle-state handoff is performed by the device. It selects an appropriate cell for future access. The active-state handoff is initiated by the network. The serving cell migrates the device to another target cell to retain radio access.

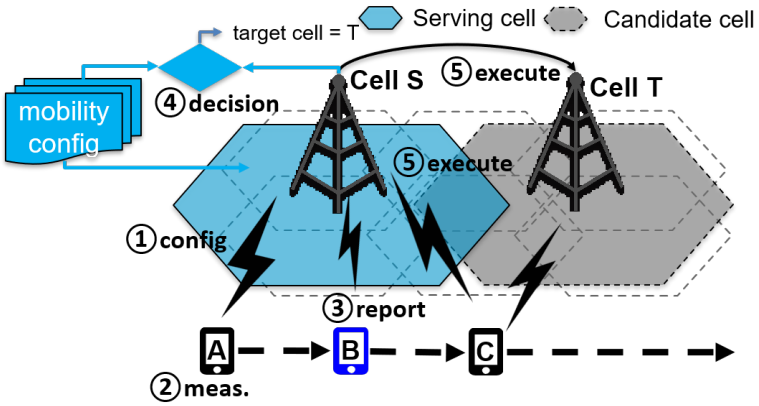


Figure 2.1.: One basic handoff procedure.

Fig. 2.1 depicts a basic handoff procedure. It typically consists of four to five steps: configuration, measurement, reporting (only for active-state handoff), decision and execution. Initially, the device is served by cell S, it receives handoff configuration parameters broadcast by cell S (①) and learns the criteria to trigger, decide and perform a handoff, including whether and when to invoke measurement, which radio frequency channel to measure, when to send measurement report back and how to determine the next target cell, to name a few. The subsequent steps (②-④) will be invoked when the criteria configured by cell S's handoff parameters are satisfied at runtime.

Active-state and idle-state handoffs differ at step ② and ③. In an active-state handoff, the device receives measurement object (which radio frequency channel to measure) and reporting criteria within RRC messages from serving cell S and keep conducting measurements as long as the measurement objects are not removed. The device reports its measurement results back when any one of the reporting criteria is met (*e.g.*, , one candidate cell's radio signal strength is offset stronger than S's). Cell S then decides whether to switch to a new cell and to which cell to go (④). In an idle-state handoff, device conducts measurement in step ② when measurement criterion is met and step ③ is skipped. The device makes a decision locally and use the decision criteria pre-configured by the serving cell. Eventually, the cell switches from cell S to cell T under network-device cooperation (⑤). Once this round completes, the device is served by cell T and is ready to repeat the above procedure.

2.1.2 Policy-Based Configurations

Cellular networks use policy-based handoffs: each step in a handoff is controlled by various pre-configured criteria in the format of mobility configurations. There are many factors to be considered during a handoff procedure, including cell priorities, radio link quality, list of measurement report events of interest, eligible candidate cells, *etc.* Here, we use 4G LTE to illustrate the main parameters (Tab. 2.1) and their

Table 2.1. Main configuration parameters standardized for handoff at 4G LTE cells.

Category	Parameter	Remark
Cell priority	P_S	Priority of the serving cell, ranging from 0-7 with 7 being the most preferred
	P_C	Priority of candidate cells in neighborhood, associated with its frequency channel, *i.e.*, P_{freq}
Radio signal evaluation	Θ_{intra}	Threshold of radio signal strength level for intra-freq measurement ($\Theta_{intra,rsrp}$, $\Theta_{intra,rsrq}$)
	$\Theta_{nonintra}$	Threshold of radio signal strength level for non intra-freq measurement ($\Theta_{nonintra,rsrp}$, $\Theta_{nonintra,rsrq}$)
	Δ_{min}	minimum required signaling strength for handoff ($\Delta_{min,rsrp}$, $\Delta_{min,rsrq}$)
	H_e, Θ_e, Δ_e	Hysteresis, threshold(s) and offset(s) used for the reporting event e (A1–A5, B1–B2)
	H_s	Hysteresis value added to the serving cell's radio signal strength
	$\Theta^{(c)}_{higher}$	Threshold of radio signal evaluation for a higher-priority candidate cell
	$\Theta^{(c)}_{lower}, \Theta^{(s)}_{lower}$	Thresholds for a lower-priority candidate cell and the higher-priority serving one
	Δ_{equal}	Offset of radio signal comparison for equal-priority cells, $\Delta_{s,n}$, Δ_{freq}, Δ_{cell}
Timer	$T_{reselect}$	Time required to fulfil switching condition
	$T_{reportTrigger}$	Time to trigger when measurement report triggering criterion is always fulfilled
	$T_{reportInterval}$	Interval for sending measurement report
	$T_{decision}$	Time to trigger when the radio signal evaluation criterion is always fulfilled
Misc	$Freq_{interest}$	List of the frequency channels of interests
	$List_{forbid}$	List of forbidden candidate cells (due to access control)
	$meas_{bandwidth}$	maximum bandwidth allowed for performing measurement

usage. Complementary to the general handoff procedure described previously, we use decision trees to exemplify how each step is determined by handoff configurations in three stages: measurement, reporting and decision for active/idle-state handoffs. We elaborate these three steps one by one and explain some representative configurations as examples. Note, in these steps, there are also some less important parameters such as timers and counters for various purposes being used.

Mobility configurations for idle-state handoffs. Fig. 2.2 depicts representative configurations used in an idle-state handoff procedure. In idle-state handoff, mobile devices follow the guide from connected serving cell to do measurement (step 2) and make handoff decision locally (step 4).

For energy efficiency, the measurements over candidate cells (except serving cell) are conducted as efficient as possible instead of running all the time. LTE runs two types of measurements: (M1) intra-freq and (M2) non intra-freq (aka, inter-freq and inter-RAT) [2]. If

$$\gamma_S \leq \Theta_{intra}(\Theta_{nonintra}), \text{ (actually, } \gamma_S - \Delta_{min} \leq \Theta) \tag{2.1}$$

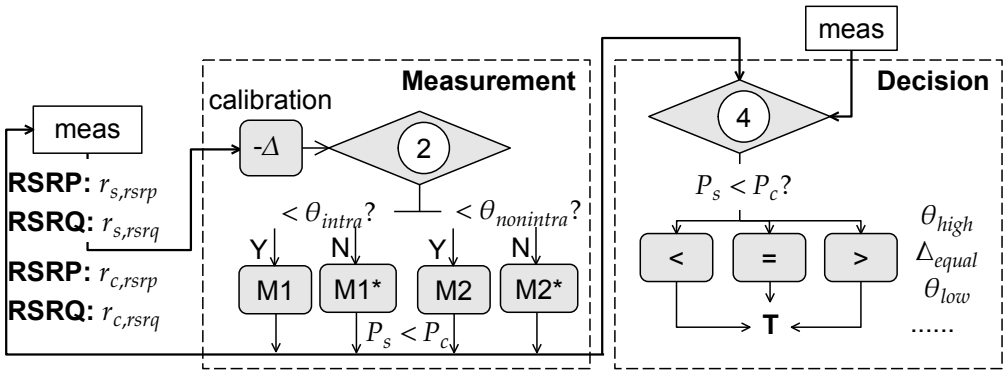


Figure 2.2.: Flow of idle-state handoff steps in a decision tree.

intra-freq (non intra-freq) measurement is triggered, otherwise only the measurement for those higher priority cells ($\{C|P_C > P_S\}$) is performed periodically (every $T_{higherMeas}$ seconds). Mobile device measures the received radio signal quality using multiple metrics like RSRP (reference signal received power) and RSRQ (reference signal received quality) for 4G LTE. They take different values: RSRP (-140 dBm, -44 dBm), RSRQ (-19.5 dB, -3 dB), and thus use distinct configuration parameters. Without loss of generality, we use RSRP hereafter unless specified. Calibration is used to compensate for different transmission power and ensure fair radio signal comparison. It converts the actual measurement $\dot{\gamma}_S$ into a level of radio signal quality $\gamma_S = \dot{\gamma}_S(\text{actual}) - \Delta_{min}$, where Δ_{min} is another pre-configured parameter indicating the minimum accepted radio signal strength.

The idle-state handoff decision is made by comparing radio signal strengths of the serving and candidate cells, given their cell priorities. It considers three cases: higher-priority, equal-priority and lower-priority. The ranking of a candidate cell is higher ($rank_c > rank_s$) when one of the following criteria is satisfied,

$$\begin{cases} (1) \text{ if } P_c > P_s, & \gamma_c > \Theta^{(c)}_{higher} \\ (2) \text{ if } P_c = P_s, & \gamma_c > \gamma_s + \Delta_{equal} \\ (3) \text{ if } P_c < P_s, & \gamma_c > \Theta^{(c)}_{lower}, \gamma_s < \Theta^{(s)}_{lower} \end{cases} \tag{2.2}$$

The decision is made until the above requirements have been fulfilled for $T_{decision}$ to avoid frequent handoffs caused by measurement dynamics. Δ_{equal} (> 0 expected) implies the favor towards the serving cell. Other rules count on the threshold settings to customize the criteria at a higher or lower priority.

Mobility configurations for active-state handoffs. Fig. 2.3 depicts representative configurations used in an active-state handoff procedure. In active-state handoff, mobile devices conduct measurements on request (step 2), send measurement report when needed (step 3) to assist serving cell to make handoff decision (step 4).

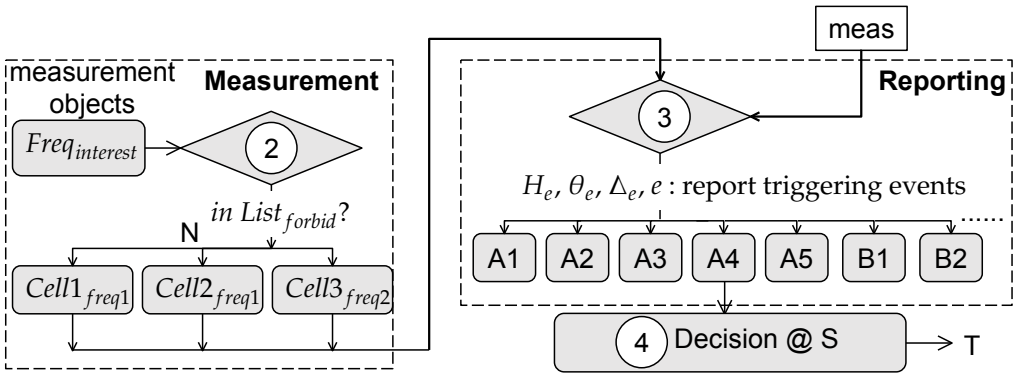


Figure 2.3.: Flow of active-state handoff steps in a decision tree.

The measurement objects required for an active-state handoff are directly assigned by serving cell, and device conducts measurement accordingly. When the measurement object is on inter-freq, mobile device switches to the target cell, performs measurement and switches back in measurement gap agreed with current serving cell. The measurement gap is a small period of time agreed between mobile device and network during which no UL/DL transmission happens.

During an active-state handoff, mobile device needs to assist serving cell to make handoff decision by sending measurement reports notifying changes of radio environment. LTE uses a set of reporting events to determine whether, what and when a user device should report its measurement results. LTE defines ten events (A1-A6, B1,B2, C1,C2) [2], but our measurement study shows that not all the events are used (A1-A5, B1 and B2 observed). Each event targets one specific condition and has its own

configuration set (thresholds Θ_e, hypothesis H_e and offsets Δ_e). A1 and A2 indicate that the serving cell's radio signal strength γ_S is better or worse than a threshold. A3 and A4 indicate that one neighboring (candidate) cell is better than the serving one plus an offset (A3) or a threshold (A4). A5 indicates that the neighboring cell is larger than a certain threshold while the serving one is weaker than another threshold. Events B1 and B2 indicate the existence of a decent inter-RAT neighboring cell (B1: larger than one threshold, B2: larger than another threshold and the serving one weaker than certain threshold). We use A3 to illustrate the event form:

$$\begin{cases} \textit{A3. Reporting condition:} & \gamma_c > \gamma_s + \Delta_{A3} + H_{A3} \\ \textit{A3. Stopping condition:} & \gamma_c < \gamma_s + \Delta_{A3} - H_{A3} \end{cases} \tag{2.3}$$

Δ_{A3} is a positive offset and indicates a stronger candidate cell. H_{A3} is a hysteresis to adapt the start and stop conditions. It is expected to be positive.

As for the final handoff decision in active-state handoff, it is made at network side. The radio signal evaluation (through measurement reports) is treated as a necessary but not a sufficient condition. How network makes handoff decision based on the input depends on mobile operator's proprietary mobility policy.

2.2 What Does The Handoff Configurations Look Like In Reality?

Following the common mechanism standardized in 3GPP specifications, mobile operators have freedom to deploy tunable handoff configurations to manage handoffs at each cell and different locations. These configurations further control how handoff is performed in the wild and enable micro-level mobility management for mobile operators. Despite the importance, there is little study on practical handoff configurations facing two main challenges. First, there are no handoff traces public available for study. For privacy issue, mobile operators are reluctant to release their data sets related to handoff operations. Furthermore, it is nontrivial for the operators to collect and archive handoff operations, given that handoffs are executed at each cell for each

mobile device in the distributed manner over geo-distributed areas. Second, handoff configurations take many parameters and are distributed at all cells for a mobile carrier network. According to the standard specifications [2–6], there are 66 parameters for a single 4G LTE cell and 91 parameters for 3G/2G RATs being used. It is not an easy task to monitor, manage and tune those configurations deployed on every single cell at network side even for mobile operators.

To address the above challenges, we take a device-centric, rather than network-centric, approach to conduct measurement study on handoffs and collect handoff configurations. Our study exhibits two main points regarding handoff configurations. 1) Operators deploy extremely complex and diverse configurations to control how handoff is performed in the wild. 2) The settings of handoff configurations determine the handoff behavior and affect data performance rationally.

2.2.1 Measurement Methodology

MobileInsight and trace collection. To collect handoff traces and configurations on mobile device, we develop and use MobileInsight [7], a software tool runs on commercial off-the-shelf mobile devices and achieve access to various low-level protocol operations in 2G/3G/4G networks from device side. By utilizing a side-channel interface of cellular hardware chipset on mobile device exposed through enabling diagnostic mode, MobileInsight interacts with cellular chipset directly through DIAG commands (*e.g.*, Qualcomm Proprietary Protocols for Qualcomm chipset) and achieve diagnostic messages of low-level cellular protocol operations in binary stream. MobileInsight further parse the achieved messages to understand the network operations. It uses Wireshark dissector for standard messages (*e.g.*, LTE RRC and NAS layer messages) and reverse-engineering approach for decoding chipset specific diagnostic messages. MobileInsight overcomes the long-lasting research barrier of closed cellular networks, where operators do not release data traces collected at network side.

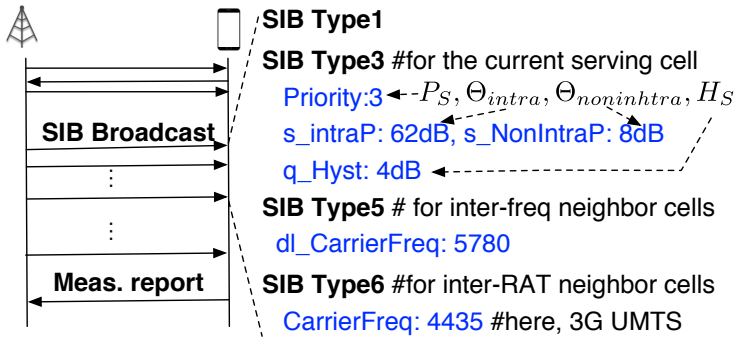


Figure 2.4.: An example trace collected by MobileInsight.

Fig. 2.4 shows an example trace collected by MobileInsight at RRC layer in AT&T LTE network which contains deployed handoff configurations. Once a mobile device gets connection with a serving cell, it receives handoff-related configurations from the serving cell conveyed in RRC layer messages. Those handoff configurations can be included in multiple types of System Information Block (SIB) (*e.g.*, type 1, 3, 5 and 6) mainly for idle-state handoff and RRC Connection Reconfiguration message for active-state handoff. In this example, the AT&T serving cell notifies the mobile device that the priority of current frequency channel is 3 (ranging from 0-7 and 7 is the most preferred), and also indicates the specifies conditions for intra-frequency and non intra-frequency measurement and two inter-frequency channels for interest including a 3G UMTS channel. These handoff configurations are used for idle-state handoff. Note the last measurement report sent from mobile device is used for active-state handoff. It indicates that a measurement report triggering event is triggered in device side and mobile device sends a measurement report including measurement results of both current serving cell and a specific measurement object. It leaves to the network to determine whether an active-state handoff is needed or not. By keep monitoring the RRC layer messages exchanged between serving cells and a mobile device including RRC Connection Reconfiguration messages along with handover request commands (used for active-state handoff) and SIBs with serving cell ID changes (used for idle-

state handoff), we successfully achieve the whole picture of handoff configurations and executions from the view of device side without any assistance from operators.

MI-Lab and experiment design. Furthermore, we develop and use the experiment framework MI-Lab [8] to run experiment under specific network context as needed and expand the measurement study to global scale. MI-Lab uses client/server architecture to distribute measurement tasks to researchers at different locations to conduct designed experiments on larger scale. At server side, we publish designed experiment task (here MMLabv2 in this project), receive collected experiment traces uploaded from each joined client device and conduct further data analysis. At client side, researchers download the latest experiment task (a designed Android APP) and install it on their client devices through MI-Lab client APP, conduct experiment by running the APP while connecting to local cellular networks and upload traces to server. When the measurement task is conducting at client device, MI-Lab client APP itself keeps running as a background service to collect MobileInsight logs from radio chipset. The MMLabv2 task APP is developed to manipulate the network environment so that we can collect mobility configurations with much higher efficiency. MMLabv2 disconnects the connected cellular network regularly to receive SIBs for idle-state handoff (a serving cell always sends SIBs after a new mobile device gets connect to it) and sends PING traffic in background to trigger active-state handoff and receiving related configurations. It also asks the mobile device to switch network RATs among 4G LTE, 3G and 2G networks periodically to collect mobility configurations of not only 4G LTE but also other RATs.

2.2.2 First Look At LTE Handoff Configurations

We first look into real-world handoff configurations by assessing observed handoffs using AT&T and T-Mobile LTE networks in three US cities (Chicago, IL, Indianapolis, IN and Lafayette, IN) and highways in between.

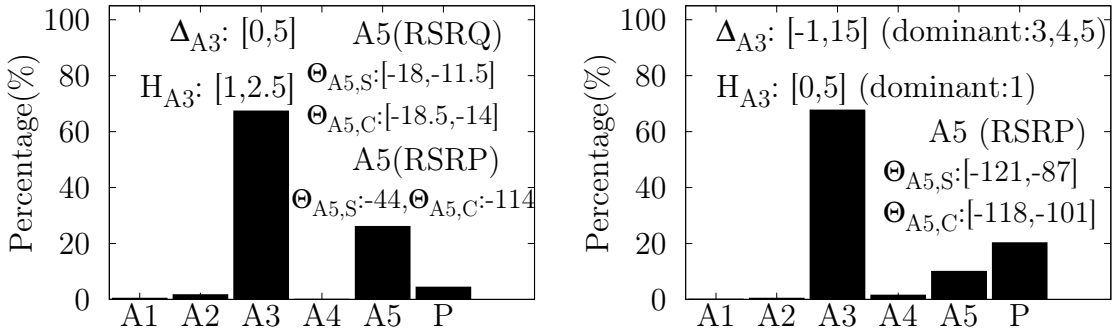
1) Active-state Handoff Configurations: reporting configurations vary and two event types are dominant. For active-state handoff, we check which measurement report triggering events are dominant and what do their parameters look like.

We observe that active-state handoffs are triggered by different types of reporting events with distinct configuration values. We observe that all active-state handoffs (99.6%) come after multiple reporting events (*e.g.*, one or multiple A2/A3/A5/P events) and they always end with one of the following events: A3, A5 and periodic event (periodically triggered measurement report for strongest cells) triggered reporting of neighboring cells' radio signal quality.

We gauge that the last event is likely to be decisive because all the handoffs happen immediately (within 80-230 ms) once the last measurement report is sent to the serving cell. It is not hard to understand that other events are not enough to invoke a handoff. Specifically, event A2 (the current serving cell is weaker than one threshold) should not trigger a handoff alone unless there is a strong candidate cell (A3, A5, P) being reported. Periodic reporting and other events can be triggered when the reporting criteria is satisfied but the reported radio signal quality is not sufficient to make a handoff decision until the last one comes.

Fig. 2.5 plots the distribution of the last observed triggered events before an active-state handoff, along with the range of their main parameter values in AT&T and T-Mobile. We clearly see uneven usage of distinct events and carrier-specific configurations. This conclusion is also applicable to other carriers. Operators may use different radio signal metrics. AT&T uses RSRP for A3 and RSRP and RSRQ almost equally for A5, whereas T-Mobile uses RSRP in most cases. Although RSRP and RSRQ are conceptually interchangeable, there is no 1:1 mapping between them. As such, we observe more A5 options in AT&T. Specifically, we observe that AT&T primarily uses A3 (67.4%) and A5 (26.1%) while P (4.4%) and A2 (1.7%) are occasionally observed. In T-Mobile, most active-state handoffs come after A3 (67.7%), P(20.2%) and A5 (10.0%). In both carriers, A1 and A4 are rarely observed ($< 0.5\%$)

and other events like A6, B1, B2, C1 and C2, are never observed. Moreover, parameter values are quite different. For example, Δ_{A3}, the offset value in event A3, ranges in [-1 dB, 15 dB] in T-Mobile, but [0 dB, 5 dB] in AT&T (dominated by 3 dB). Parameters for A5 are even more dispersed.



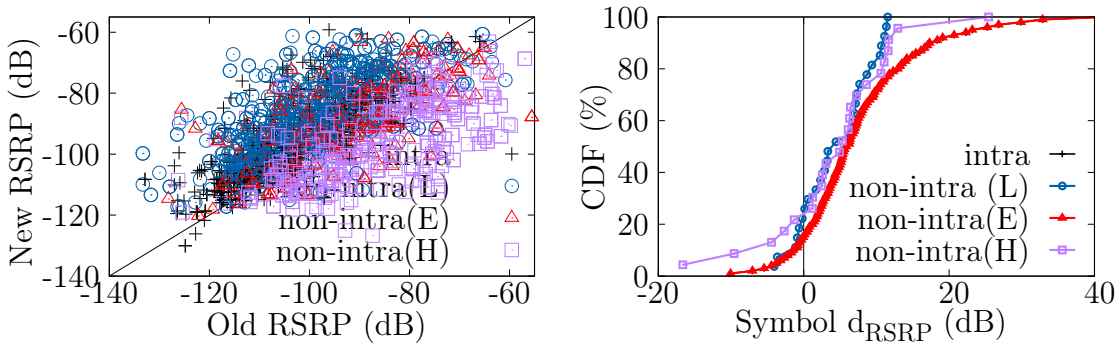
(a) Percentage of events (AT&T) (b) Percentage of events (T-Mobile)

Figure 2.5.: The last LTE Measurement report triggering events observed before active-state handoffs.

Implications: A3 and A5 events dominate the practice by mobile operators. A3 is the most popular one for its simplicity. It uses a *relative* comparison to mandate that the new cell is better (usually, $\Delta_{A3} > 0$) than the serving one. A5 is the most flexible one. It can make the same comparison as A3 (*e.g.*, offset = the gap in two thresholds). Note that, it differs from A3 because A5 has additional requirements on the absolute radio signal quality (the serving one $< \Theta_{A5,S}$ and the candidate one $> \Theta_{A5,C}$). A5 can substitute other events with particular parameter settings, such as A2 (when $\Theta_{A5,C}$ = -140 dB, the worst RSRP) and A4 (when $\Theta_{A5,S}$ = -44 dB, the best RSRP). In fact, we do observe the latter one in A5 frequently. This may explain why other events are rarely observed.

2) Idle-state Handoff Configurations: handoff decision configurations prefer candidate cells with better radio signal except for cell with higher priority, measurement configurations prefer intra-freq measurements over non intra-freq measurements. Idle-state handoff configurations take two main

responsibilities: 1) determine when to perform an idle-state handoff and 2) determine when to perform intra/inter frequency measurement, and two sets of configurations are used to make these decision. We examine the basic implications observed from these configurations.



(a) RSRP before and after handoffs

(b) CDFs of δ_{RSRP}

Figure 2.6.: RSRP improves in idle-state handoffs except for non-intra target cell with higher priority.

Idle-state handoffs use priority and radio signal thresholds together to tune their policies on radio signal evaluation. The observed configurations indicate an implication that radio signals should "enhance" after idle-state handoffs except when switch to higher priority target cells. Fig. 2.6 examines the RSRP change before and after an idle-state handoff to a target cell with higher, equal or lower priority. The results are consistent across different carriers. Almost all the handoffs (except higher-priority non-intra freq handoffs) go to stronger cells as required by the configurations. We observe that most configuration follow the common expectations: $\Delta_{equal} > 0$ determines that it will choose a stronger cell when both have equal priorities, $\Theta^{(c)}_{lower} > \Theta^{(s)}_{lower}$ implies that the chosen cell is better than the previously serving one, when the new cell has a lower priority. Only in the higher-priority cases, handoffs occur as long as the candidate cell is better than an *absolute* value $\Theta^{(c)}_{higher}$, regardless of the serving one. It is possible that it switches to a weaker cell (20% observed). *Implications: Higher-priority cells may be preferred for better performance even with lower radio*

signal quality (4G vs 3G/2G) or other non-performance reasons (e.g., operators favor some newly deployed cells and acquired bands).

Intuitively, non intra-freq measurements should take less frequently compared with intra-freq measurements considering intra-freq measurements take less time and non intra-freq ones must measure other frequency bands with larger overhead. To satisfy this assumption, Θ_{intra} should be no smaller than $\Theta_{nonintra}$ so that $\Theta_{nonintra} \leq \Theta_{intra}$ always holds true. We plot the CDF of $\Theta_{intra} - \Theta_{nonintra}$ in Fig. 2.7 along with all value pairs observed. Clearly, it holds true (≥ 0) in these tested areas although the value pairs are diverse. We also note that it becomes zero ($\Theta_{intra} = \Theta_{nonintra}$) in 5% cases: both measurements use the same criteria and will be invoked at the same time. However, some counterexamples are found in our larger-scale study though very rare (only observed from two carriers in specific areas) In those cases, non-intra freq measurements are performed even more often than intra-freq ones. *Implications: intra-freq measurements is likely to be preferred over non intra-freq measurements for energy efficiency.*

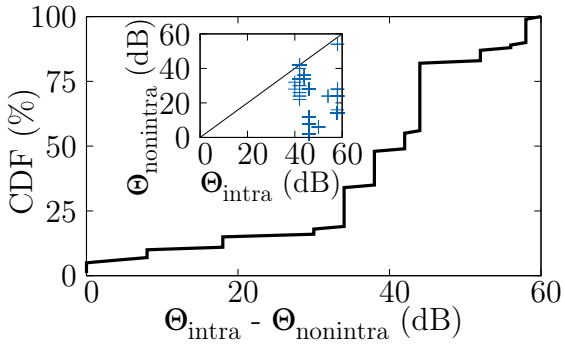


Figure 2.7.: CDFs of radio signal thresholds used for measurement in idle-state handoffs.

2.2.3 Rich Diversity Of Configurations On Larger-Scale

With the help of 35+ volunteers around the world, we are able to collect mobility configurations in a global scale from Nov 2017 to April 2018. Our dataset covers

Table 2.2. Main carriers measured on large-scale and their acronyms.

Country/Region		**Carriers**
USA (US)	4	**A**(T&T), **T**(-mobile), **V**(erizon), **S**(print)
China (CN)	3	**C**(hina)**M**(obile), **C**(hina)**U**(nicom), **C**(hina)**T**(elecom)
Korea (KR)	2	**K**(orea)**T**(elecom), **SK**(Telecom)
Singapore(SG)	3	**ST**(arhub), **SI**(ngTel), **MO**(bileone)
Hong Kong (HK)	2	**TH**(ree), **C**(hinamobile)**H**(ongKong)
Taiwan(TW)	2	**C**(hung)**W**(haTelecom), **T**(aiwan)**C**(ellular)
Norway(NO)	1	**N**(et)**C**(om)
Others	13	*e.g.*, **O**range (France), **D**eutsche**T**elekom (Germany), **VO**dafone (Spain), **M**ovi**S**tar (Mexico), $\cdots$

7,996,149 configuration samples from 32,033 unique cells locating in 30 carriers over 15 countries and regions (main carriers shown in Tab. 2.2). Fig. 2.8 shows the breakdown per carrier. Most data is collected in USA, China and several countries/regions in Asia, and a few cells (< 100) are in France, Germany, Spain and Mexico, etc. Note that we count the number of unique cells only, and we treat each parameter set observed as one sample. As a result, the number of cells is relatively small in small regions like Singapore, Hong Kong, Taiwan and Korea (Seoul only). The number of samples is much larger because some cells have been observed at multiple rounds and each round collects a set of configuration parameters as one sample. This dataset covers all five RATs (LTE, UMTS/GSM, EVDO/CDMA1x) and 4G LTE contributes to 72% cells (Tab. 2.3).

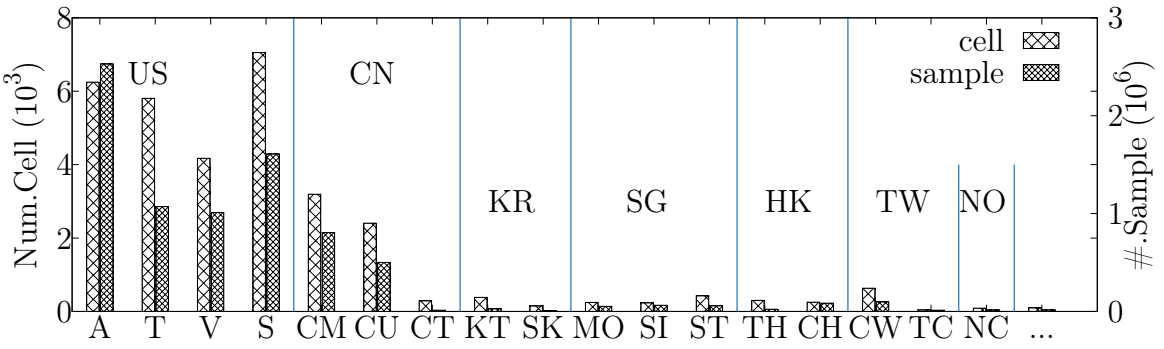
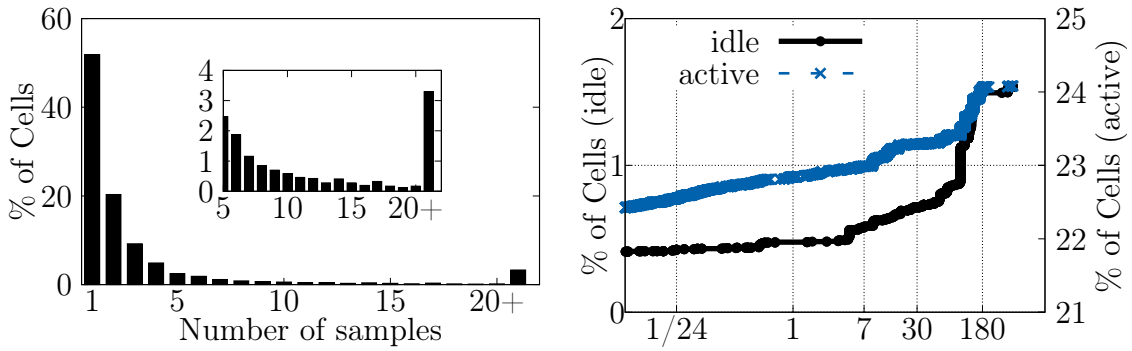


Figure 2.8.: Number of cells and samples per carrier.

Table 2.3. Handoff Parameters Breakdown per RAT.

	4G LTE	3G UMTS	GSM	3G EVDO	CDMA1x
#. parameter	66	64	9	14	4
cell-level (%)	72%	14%	5%	5%	4%

1) Mobility Configurations Have Low Temporal Dynamics. We first examine whether (and how) handoff configurations change over time. This is also critical to our dataset quality and data cleaning. Note that most data is collected by volunteers beyond our control and data collection does not run at all times. As such, not all the updates, if existing, are captured in this dataset. Therefore, actual temporal dynamics may be underestimated. However, our following analysis shows that configurations do not vary over time frequently, and thus one-time collection is sufficient.



(a) # Samples per cell (b) Temporal dynamics

Figure 2.9.: Temporal dynamics in configurations.

We first confirm that we have enough samples to study temporal dynamics. Fig. 2.9(a) shows the number of samples across all the cells for given serving cell configuration parameters (in SIB3), and almost half of the cells (48.1%) have multiple samples. This indicates that at least 48.1% of cells have > 1 samples for certain idle-state handoff configuration parameter and our dataset suffices to examine temporal dynamics. We find that active-state handoff configuration parameters have higher temporal dynamics compared with idle-state handoff configurations. We plot the percentage of LTE cells with distinct samples observed over time in Fig. 2.9(b) (two

y-axes). If the cell is observed with multiple samples for the same parameter in one round, it will be counted into the t=0 case. We see that configuration updates over time are really rare. Those idle-state handoff parameters are updated even less frequently. Both idle-state and active-state do not vary too much over time (idle: 0.4% to 1.6%, active: 21.2% to 24.1%, up to 2 years, mostly in 6 months). Active-state handoffs are updated more frequently.

Implications: Given low temporal dynamics, our data collection even with only one-time observation is enough. In our following study, we consider unique samples, so as not to tip distributions in favor of cells with many same samples.

2) Complex and Diverse Configurations in One US Carrier. We first use one US carrier (AT&T) to characterize handoff configurations in reality and then extend to other carriers later. We find that configurations are quite complex and diverse in all carrier networks. We characterize such complexity and diversity in terms of three measures: the number of unique values, the distribution and the dispersion over the value range.

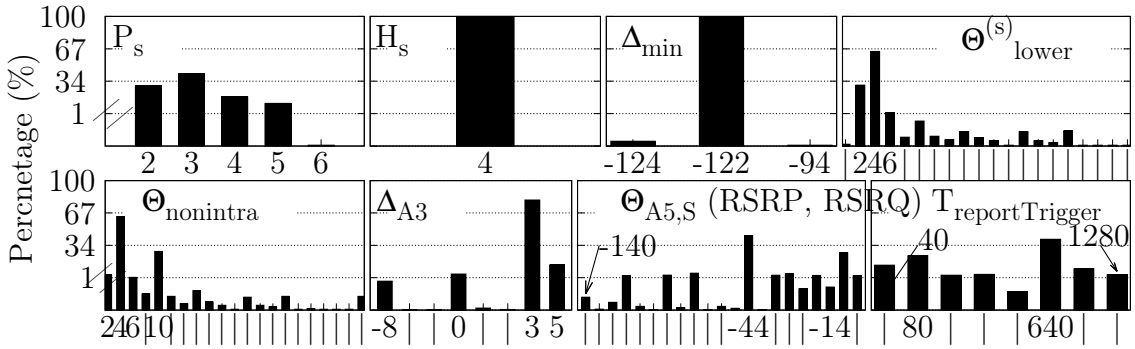


Figure 2.10.: The value distribution of eight representative handoff parameters in AT&T.

Fig. 2.10 plots the distribution of eight representative parameters selected from Table 2.1. They cover all three handoff steps: two for measurement (P_s and $\Theta_{nonintra}$), three for reporting (Δ_{A3}, Θ_{A5} and $T_{reportTrigger}$) and three for decision (Δ_{min}, H_s and $\Theta^{(s)}_{lower}$). They also cover three configuration categories of cell priority, radio evaluation and timer. The y-axis uses two scales and the one of [0%, 1%] has been amplified for

better readability. '|' represents skipped parameter value. We have three observations. First, there are multiple distinct values for most parameters, except the hysteresis for the serving cell's radio signal evaluation H_s ($4dB$). On the extreme end, some parameters such as $\Theta^{(s)}_{lower}$, $\Theta_{nonintra}$ and $\Theta_{A5,S}$ have around 20+ options. Second, their distributions vary a lot as well. Some have a skewed distribution with one or few dominant values (*e.g.*, Δ_{min}, the measurement calibration threshold mainly set as -122 dB). Others have a relatively even distribution across most values (*e.g.*, the priority of the serving cell P_s as 0-7 for LTE cells). This indicates that AT&T does not treat all 4G LTE cells equally by using finer-grained priority settings. However, it may induce inconsistent priority settings and problematic handoffs are disclosed in our later study. Third, rich diversity does not only exist in their distribution but also in their value range. Some parameters disperse in a broad range of values (*e.g.*, [-140 dB, -8 dB] for $\Theta_{A5,S}$ (both RSRP and RSRQ supported), and [40ms, 1280ms] for the $T_{reportTrigger}$ timer). Such wide dispersion implies that those parameters probably affect handoff quality more.

To quantify such diversity, we apply two popular metrics: Simpson index of diversity [9] and coefficient of variation [10]. Simpson index is to quantify the diversity in distribution. It is better than the naive measure of the number of unique values (richness) because it takes into account the relative abundance of each value. Coefficient of variation is a well-defined, statistical measure to quantify the diversity in the value range. This complements Simpson index for measuring relative variability. They are given by

$$D = 1 - \sum_{i=1}^{m} (n_i)^2/N^2, \quad C_v = \frac{\sqrt{Var[X]}}{E[X]} \tag{2.4}$$

Where m is the number of unique values, n_i is the count of a single value x_i, and N is the total counts of all values $N = \sum_{i=1}^{m} n_i$. $E[X]$ and $Var[X]$ are the expectation and variance of the data X ($X_j, j = 1 \cdots N$). Simpson index D ranges from $[0, 1]$

and a lower value indicates less diversity. A lower coefficient in C_v indicates lesser dispersion in value.

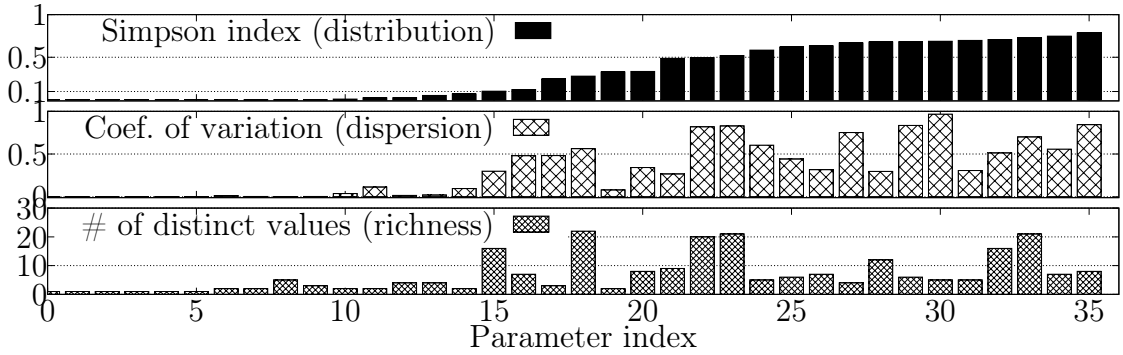


Figure 2.11.: Diversity measures of LTE handoff parameters in AT&T.

Fig. 2.11 shows the diversity measures of all handoff configuration parameters observed in AT&T, sorted in the increasing order of Simpson Index. We only observe a subset of configuration parameters because AT&T does not support 3G EVDO and 2G CDMA technologies (some parameters not applicable). Some events are not observed (say, B1, B2, A6) or rarely observed (say A1, A4). Those parameters are omitted as well. The eight representative parameters in Fig. 2.10 are included and the Simpson index and coefficient of variation: P_s(index:31, 0.69,0.3), H_s(index:1,0,0), Δ_{min} (index:9,0.003,0.003), $\Theta^{(s)}_{lower}$(index:22,0.49,0.81), $\Theta_{nonintra}$(index:23,0.52,0.82), Δ_{A3}(index:20,0.33,0.34), $\Theta_{A5,S}$ (index:32,0.72,0.69), $T_{reportTrigger}$(index:35,0.78,0.84). We can see that each configuration parameter has its unique diversity pattern. The only exception is those parameters with no/low diversity (index $\leq$ 16 or 8). In fact, the first 8 parameters are single valued and No.9-16 are dominated by a single value. We find that these parameters do not exhibit rich diversity because they are primarily used for calibration or are associated with other varying parameters (*e.g.*, Event A3 uses both an offset and a hysteresis. The hysteresis remains fixed as the offset varies). This way, carriers are still armed with sufficient power for fine-grained handoff management. Among those parameters with distinct values, diversity is multi-faceted with consistent or divergent patterns among their distribution, dispersion and richness. For

instance, $\Theta_{A5,S}$ (index:32), $\Theta^{(s)}_{lower}$ (index:22) and $\Theta_{nonintra}$ (index:23), are consistently diverse, but the serving priority P_s (index: 31) is diverse in the distribution but not in dispersion and richness. In contrast, $\Theta^{(c)}_{lower}$ (index:15) and $\Theta^{(c)}_{Higher}$ (index:18) have high richness and dispersion but medium (lower) distribution diversity because one or two values are dominant in use.

Implications: operators have power to realize fine-grained handoff managements with diverse configurations.

3) From One To Many Carriers. We extend the above study to other carriers. Unsurprisingly, rich diversity is observed in all other carriers. Here, we mainly present interesting results on carrier-specific diversity. We consider all four US carriers and other representative carriers each from China (China Mobile), Korea (SK Telecom), Singapore (MobileOne), Hong Kong (China Mobile Hong Kong) and Taiwan (Taiwan Cellular). The conclusions are applicable to other carriers. We select four representative parameters with different-level diversity observed in AT&T to exemplify their distributions in those carrier networks in Fig. 2.12. We show diversity measures of the same eight parameters across the chosen carriers in Fig. 2.13.

We clearly see that each parameter configuration is carrier specific. This gives several implications. First, parameters are likely configured by carriers, not by telecom equipment vendors (default values not in use). Second, we observe that diversity across multiple parameters is consistent for certain carriers. For example, SK Telecom (Korea) exhibits the lowest diversity for almost all the parameters. All four representative parameters (priority, radio signal evaluation thresholds/offsets) are single-valued. In contrast, all other carriers except Mobileone (Singapore) use highly diverse configurations for all the parameters. This implies that carriers adopt distinct (likely proprietary) configurations and policies. There might be no single answer given different goals of interests (performance, operational cost, robustness, etc.) But it might be a concern without thorough investigation on whether the current one is a winner and how far away, if not.

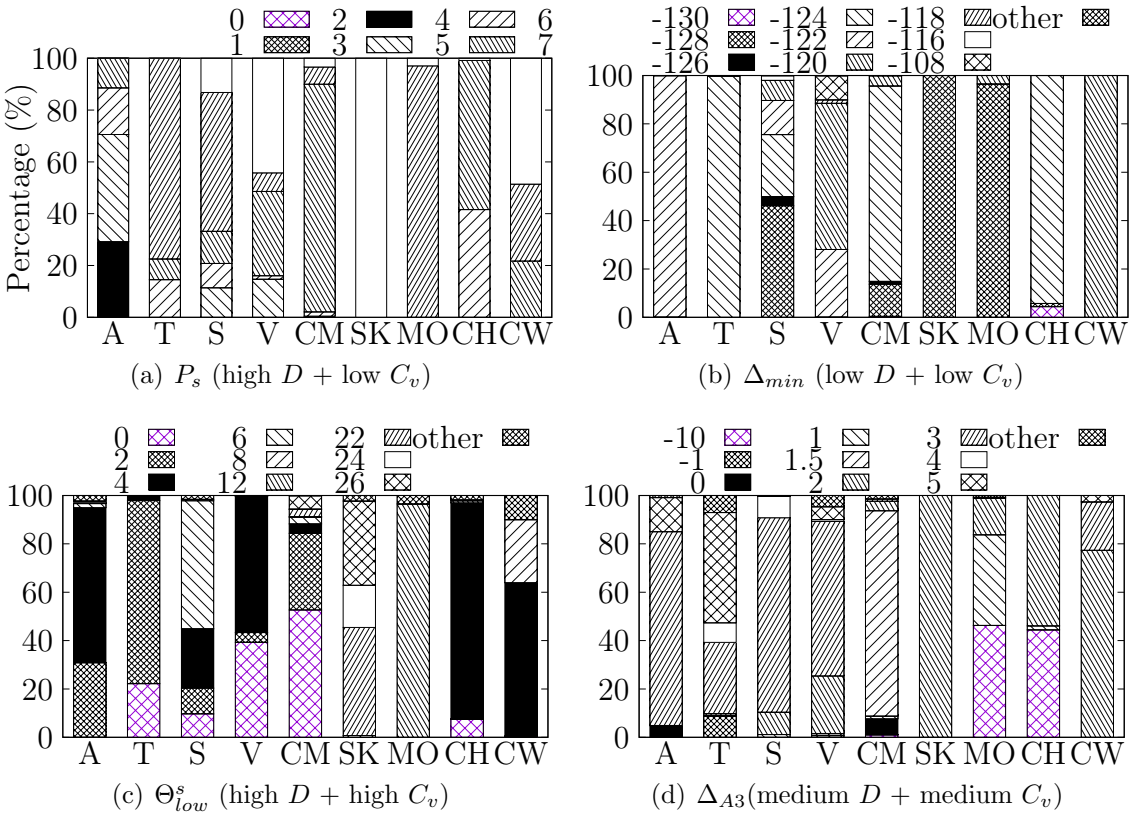


Figure 2.12.: Illustrative distributions of four representative parameters across carriers.

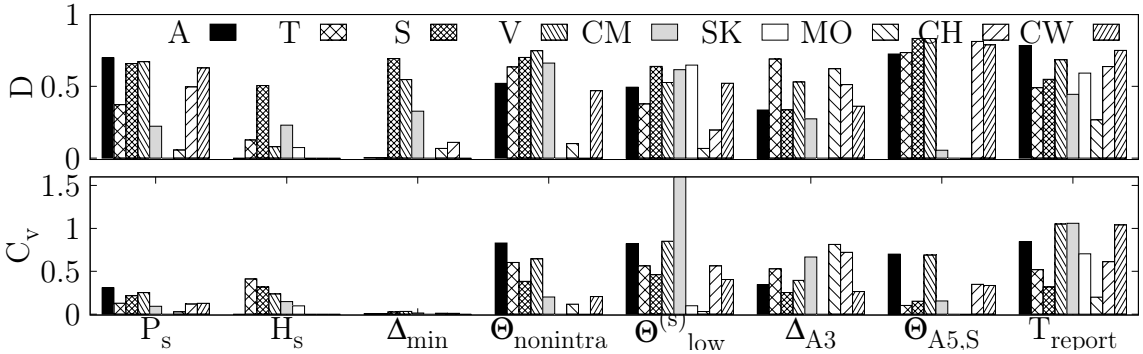


Figure 2.13.: Diversity measures of eight representative parameters across across carriers.

Implications: Carrier-specific configurations raise an interesting question: which configuration (policy) runs better? There might be no single answer given different

goals of interests (performance, operational cost, robustness, etc.) It might be a concern if handoff configurations are not well managed and verified before their use.

2.2.4 Understanding Of Configuration Diversity

We next delve into a closer look at why they are configured so. We attempt to unveil what attributes contribute to current configuration diversity and how. We consider three factors: cell frequency channel, RAT, and location. We choose them because intuitively, operators may customize their policies per cell for finest-grained management (low temporal dynamics validated in §2.2.3). These three factors decide the cell type (what the cell is) and location (where the cell is), which are the most important cell properties visible to us.

1) Frequency Channel. We first select P_S and P_C, the priorities of the serving and candidate LTE cells for frequency dependence analysis. Intuitively, they should be frequency-dependent. Fig. 2.14 plots their breakdown per frequency channel in AT&T. AT&T uses 24 distinct channels, and the operating frequency for the serving cell primarily over the channels numbered as 850, 1975, 2000, 5110, 5780 and 9820, which matches with its 4G band usage [11]. The channel number is called EARFCN (LTE Absolute Radio Frequency Channel Number), and their mappings to frequency spectrum bands are regulated by [12] and can be found online, *e.g.*, via [13]. We find actually all carriers with multiple values (except SK and MO with low diversity) observe similar frequency-dependent patterns.

We see that each frequency channel is mostly associated with one single/dominant value and the use of multiple frequency channels is the primary contributor to current priority diversity (exceptions explained later). There are several interesting findings. First, AT&T uses a lower priority (here, 2) for LTE-exclusive bands (called main bands [11], bands 12 and 17 around 700MHz), including 5110/5145 (band 12) and 5780 (band 17). Channel 1975 (band 4, AWS-1) is an exception. A higher priority (5 or 4) is mainly assigned to the 9820 channel (band 30, 2300 MHz WCS), which was

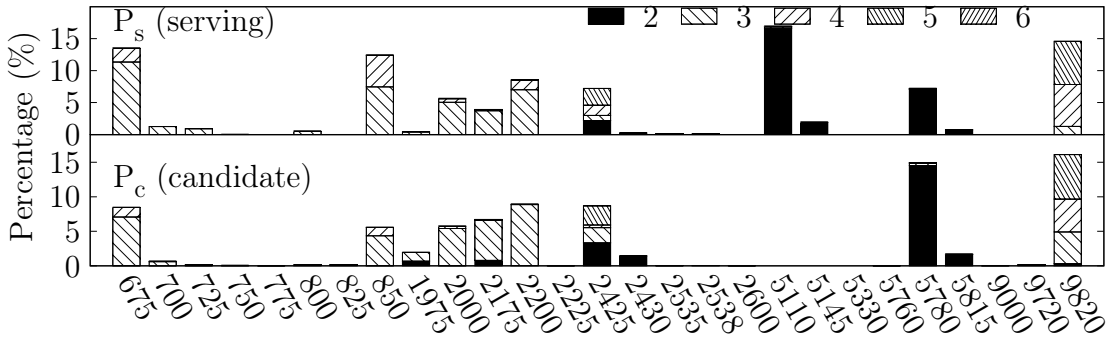


Figure 2.14.: The breakdown of the serving and candidate cell priorities over frequency channel in AT&T.

recently acquired to provide additional bandwidth. Such priority setting implies that AT&T prefers the additional bands to the LTE-exclusive (main) bands and tends to use 3G-coexisting bands for LTE as much as possible. This sounds like a good upgrade strategy which facilities and accelerates wide adoption of a new RAT. Second, some frequencies use multiple values which are prone to conflicts. For example, AT&T assigns two or three values over the channels of 1975, 2000, 2425, 5870 and 9820. This is observed at 6.3% of AT&T cells in our measurement study. Such multiple-value priority settings are also observed in other carriers. However, such inconsistent priority settings might make trouble. Consider a case where two cells believe the other has a higher priority. It is prone to a handoff loop. Our large-scale study shows that this problem exists in many carriers and unfortunately it is not as rare as we anticipated before. Third, our study helps troubleshoot the problematic practice. We notice that updating priorities with new channel preference can be problematic in practice. [14] reports one recent user complaint that AT&T breaks the 4G service for the phones that do not support band 30 (here, channel 9820). But no technical cause has been reported. Now, we know why. AT&T sets the highest priority to band 30 and thus the handoff decision step likely chooses the cell over it, even when 4G LTE cells over different channels are available (otherwise, the phone could not work well before). Given that not all the phones support band 30 (*e.g.*, a Verizon iPhone 6S+ [14]), they are unable to switch to the target cell all along and thus the 4G service

is disrupted. Actually, conflicts or inconsistent configurations between base stations and mobile devices are not rare. There are more instances observed in study [15].

We further quantify such frequency dependence, using a generic measure of parameter θ's dependence on a factor F:

$$\zeta_{\mathcal{M},\theta|F} = E\left[|(\mathcal{M}(\theta|F = F_j) - \mathcal{M}(\theta)|\right] \tag{2.5}$$

where $\mathcal{M}(\theta)$ is the θ's diversity measure (here, D or C_v). We compare it with the expectation of the conditional ones $\{\mathcal{M}(\theta|F = F_j)\}$. We plot $\zeta_{\mathcal{D},\theta|freq}$ and $\zeta_{\mathcal{C}_v,\theta|freq}$ for all the parameters observed in AT&T in Fig. 2.15. We indeed observe that frequency dependence per parameter is also carrier-specific. However, it holds true to all the carriers that not all highly diverse parameters (here, No.$\geq$ 17) are frequency-dependent. Interestingly, we find that some reporting events are frequency-dependent like A2 (index: 32) and A5 (index: 33 and 34) but some not, such as A1 (index: 21) and A3 (index: 21). This helps to infer the carrier's handoff policies. Here, we can see that there is a universal standard for a good cell (A2) and relative comparison (A3) but the standard for a poor cell (A2) and the absolute value setting (A5) are frequency-dependent. We also observe that some other parameters like $T_{reportTrigger}$ (index 35) and hysteresis (index: 27) are frequency-independent, which matches with their usage purpose.

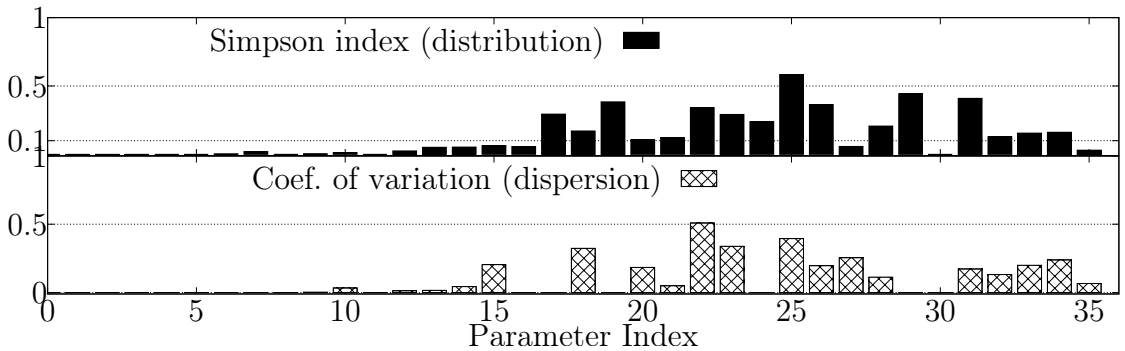


Figure 2.15.: Measures of frequency dependence: $\zeta_{\mathcal{D},\theta|freq}$ and $\zeta_{\mathcal{C}_v,\theta|freq}$ across all the parameters in the same order of Fig. 2.11 in AT&T.

2) Spatial Location. We quantify the impacts of location at the macro-level (city) and micro-level (proximity). We aim to answer two questions: (1) Do operators customize their configurations in cities? (2) Will diversity disappear (or greatly decline) among nearby cells?

We first check city-level by studying US cities only. We divide our dataset based on the cities where the configurations are collected and we present the results for top-5 cities (total number of cells in four US carriers): C1(Chicago: 4671), C2 (LA: 2982), C3 (Indianapolis: 2348), C4 (Columbus: 1268), C5 (Lafayette: 745). We choose P_S and normalize its distribution in each city. Fig. 2.16 plots the results. We observe that carriers may configure cells at different geographical locations slightly differently. In C1 (Chicago), their configurations obviously differ from those in other cities. This is understood. Operators usually divide their network domain (one nation) into multiple market areas and they may run incremental deployment and configurations over time. The bands used may differ as well. We also check other parameters and observe location-dependent diversity.

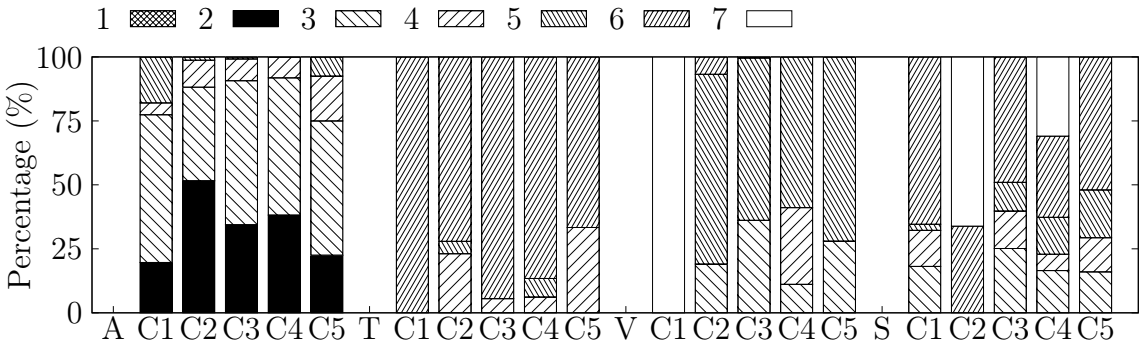


Figure 2.16.: City-level priority distributions in five US cities.

Implications: Operators deploy different configuration values at different locations(cities). This also explains some problematic configurations are observed only at specific areas.

We further consider those cells in close proximity. Handoffs are distributed. After mobile device switches to a new serving cell, the configurations associated with the

new cell take effects. So a handoff is affected by configurations at co-located cells. As our data collection is dependent upon user movement, we observe that the cells covered in our dataset are sparse except those cells collected carefully by us. So we use a subset of dataset which is collected in a more controlled manner by us. In particular, we drive along the main roads separated every 500m–1Km and cover the whole city to get a more dense collection. We have done so in C3, C4 and C5, partially in C1 and C2. We apply Eq. (2.5) to define a measure of spatial diversity as $\zeta_{\mathcal{M},\theta|R}$, where R is the radius of one neighborhood, $\mathcal{M}$ is the diversity metric and θ is the parameter to study. For any cell c, we obtain the cluster of cells located in a circle of radius R km and obtain $\zeta_{\mathcal{M},\theta|R}[c]$. To illustrate its spatial diversity, Fig. 2.17 shows the boxplot of $\zeta_{\mathcal{M},\theta|R}[c]$ for all the cells in C3. We select various radii to gauge the change in configurations. We only show the results for AT&T, Sprint and Verizon. We observe that carriers indeed use varying values for cells located closely to each other. This indicates that even in a very small geographical area ($r < 0.5$), carriers prefer to fine tune different parameters. However, this is not the case for all the carriers. In T-Mobile, we observe that spatial diversity in close proximity is extremely small (almost zero). That is, spatial diversity does exist across small geographical areas but is also carrier dependent.

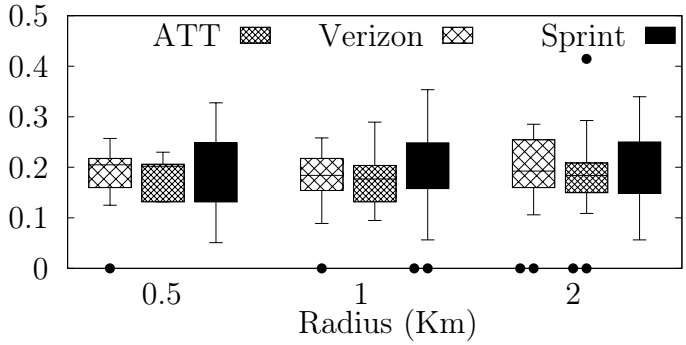


Figure 2.17.: Spatial diversity for P_s under various Radii in Indianapolis (C3).

Implications: Lower dispersion is observed in a smaller range (like a city or a neighborhood). Location-dependency is likely caused by real-world deployment (the

network deployment and upgrade do not happen at the same time using the same equipments). It can be also attributed to the carrier's configuration over a geographic area.

3) RATs. We finally study the configuration patterns under other RATs and learn how they have evolved. Because different RATs use different sets of parameters, it is hard to compare each parameter across RATs. We thus calculate the diversity metric (here, Simpson index) for all the parameters and show their boxplots in Fig. 2.18. We see that handoff configurations are becoming more and more diverse along the RAT evolution. In particular, LTE heavily inherits from UMTS and thus they have a large number of parameters in common. CDMA2000 and CDMA1x are used by Sprint and Verizon and configured differently from LTE. They use a smaller number of handoff parameters. Most of the parameters are observed to have a single dominant value and relatively static configurations. Similarly, GSM is also observed to have an almost static configuration scheme. The average diversity of their parameters is significantly smaller than those of LTE and WCDMA, indicating single dominant values. Thus, the evolution of RATs over time has also made the cell handover procedure more convoluted and complicated where more numerous parameters with varying and diverse configurations are used.

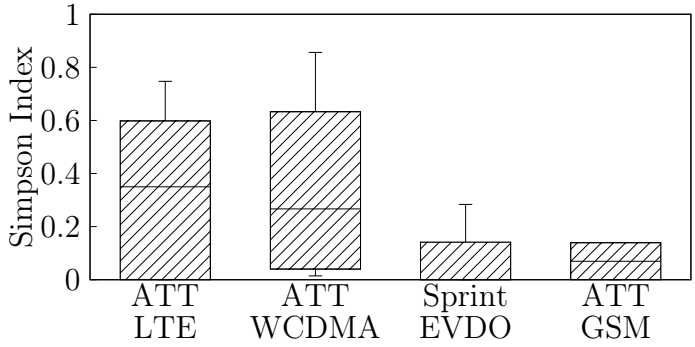


Figure 2.18.: Boxplots of diversity metrics of all parameters used by different RATs.

Implications: Increasing diversity may continue in the coming 5G, especially with hybrid and more radio access options. This measurement study likely helps understand mobility support in 5G as well.

2.3 The Mobility Policy Behind Configurations Of Mobile Operators

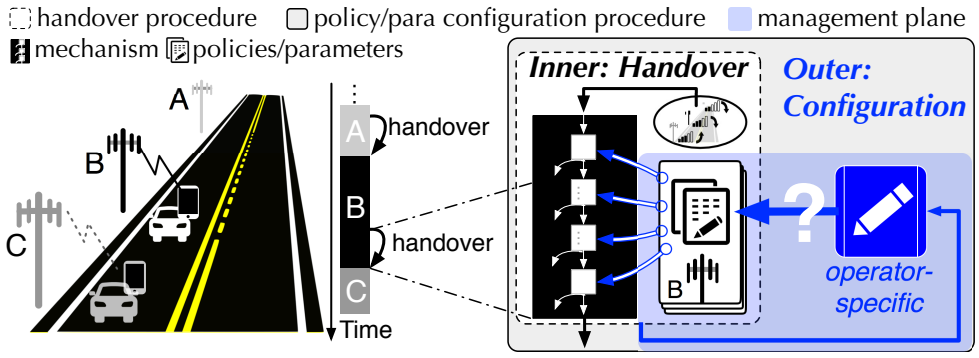


Figure 2.19.: Mobility support is via handover, a standard inner-procedure managed by per-cell policies which are configured by the operator via an outer-procedure.

Section 2.2 reveals what mobility configurations look like and shows the complexity and diversity of observed carrier specific configuration values. However, the operators' mobility management plane remains unclear as we have no idea for the following questions. Why do operators deploy their mobility configurations in this way? What is the rules operators follow to deploy configurations and make active-state handoff decision at network side? In this section, we step further deep to address a subtle yet fundamental question: *What is operators' mobility management policy?* Fig. 2.19 illustrates the important role played by operator-specific mobility management policy. We have already known that each step in handover (the inner procedure) is determined by mobility configurations. Over that, there is an outer procedure used to follow operator's mobility policy and configure those parameters. Our goal is to understand the outer procedure and eventually delve into the whole management

loop (in blue) to demystify how mobility management is realized over the standard handover procedure.

In this section, we focus on active-state handoff configurations and first expose unexplored rationale behind high diversity by showing the fact that the outer procedure of configuring parameters and policies is not a one-time attempt, but a dynamic process which interacts with and adapts to the inner handover procedure in multiple iterations. Second, we perform a large scale driving measurement study in the US using 4 top-tier LTE networks to collect the sequence of configuration parameters per handover. Third, we uncover the mobility management policy as a casual inference problem and expose the configuration logic adopted by US carriers.

2.3.1 Mobility Management Is Not A One-time Attempt, But A Dynamic Process

Our previous work exhibits what mobility configurations look like but fails to disclose why operators deploy them in that way. Diverse event types and parameter values are deployed with unknown logic. For example, event A3 and A5 are observed to be decisive to trigger active-state handoff, but what is the purpose for event A1 and A2 if they are not used to suggest an active-state handoff? Without understanding the rationale behind deploying these configurations, the mobility management plane remains unclear. We find the key is to study handover configurations in *a dynamic process*, not in one static snapshot. This is because the actual handover procedure follows an updated flow. Handover may not occur given one-time measurement report. Instead, the serving cell updates the events of interests (likely adapts to the measurement results) in multiple iterations, till it finds a good candidate cell or finds no need for a handover. Here, we use two real-world examples to demonstrate how handover configurations change over time and then describe the problems to address.

Fig. 2.20 presents a real-world instance of T-Mobile with an inter-freq handover from cell 293 to cell 259. Initially, the serving cell 293 is on frequency channel 1051 (band 2, 1975.1MHz). In the next 4 seconds from 23:23:01.40 to 23:23:05.25, we

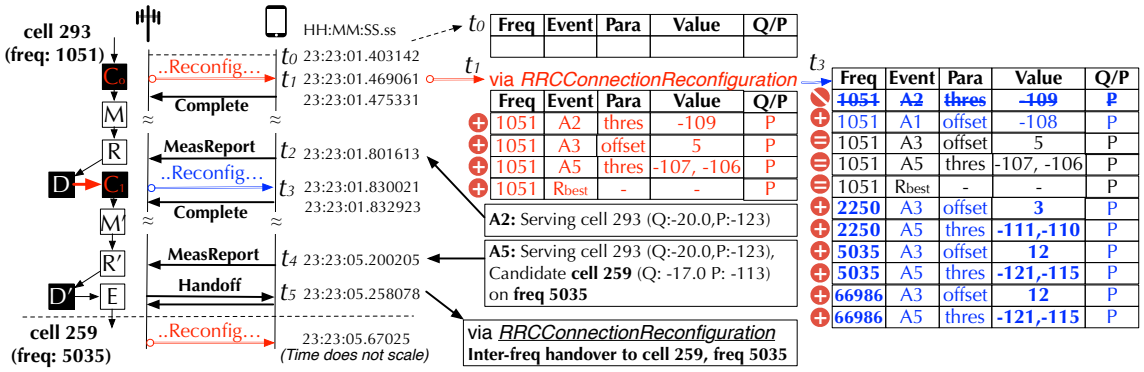


Figure 2.20.: A real-world T-Mobile handover instance with initial and updated configurations.

observe two configuration rounds, both via *RRCConnectionReconfiguration* which is a standard signaling message to modify an established RRC connection (here, setup and modify measurement and reporting events for handover) [16]. The initial round, at t_1, configures the triggering events for intra-freq measurement only (on serving cell frequency channel): A2 (γ_s < -109 dBm), A3 ($\gamma_c > \gamma_s + 5$ dBm), A5 (γ_s <- 107 dBm and γ_c > -106 dBm) and R_{best}. All use RSRP. Afterwards, the device reports the measured results that satisfy at least one of these events of interests: A2 and R_{best} (omitted in the plot). This indicates that the serving cell has a weak radio strength (-123 dBm) but it is still the best cell out of all the observed candidate cells on the same channel of 1051. No intra-freq candidate cells are observed with radio strength stronger than -118 dBm (A3: -123 dBm + 5 dBm). Otherwise, A3 would have been reported. Upon receiving the measurement reports, the serving cell sends new handover configurations which adjust the existing ones and add new items. There are four observations and implications. First, the received reports are insufficient to make a handover decision (no good candidate cell observed) but is enough to invoke more measurements (from intra-freq only to considering inter-freq candidate cells). This is not hard to understand that handover configuration should not be a one-time attempt. Second, it updates the intra-freq events by removing A2 but adding A1 (γ_s > -108 dBm). We gauge that it is used to monitor whether

the serving cell becomes good back. Note that this is possible because radio signal quality can be highly dynamic over arbitrary mobility beyond the network's control. This implies that new configurations likely depend on the previous ones as well as measurement results. Third, the obvious and also important change is to add inter-freq measurement events on the following channels: 2250 (band 4), 5035 (band 12) and 66986 (band 66). These channels are known or learned as a prior as the serving cell should be able to know the frequency channels of its neighboring cells. For each channel, it adds both A3 and A5 but with different parameter settings. Fourth, we would like to emphasize that multiple measurement reports are possible because the configured events are not exclusive. It is possible to have multiple inter-freq candidate cells better than the serving one although only one cell is observed in this example. This leaves room for the serving cell to make a decision. In this example, the device reports A5 for one candidate cell (cell 259 on channel 5035) and eventually invokes an inter-freq handover. Note that γ_c = -113 dBm and γ_s = -123 dBm, and only A5 is satisfied but not A3 (the offset of 12 dBm is big).

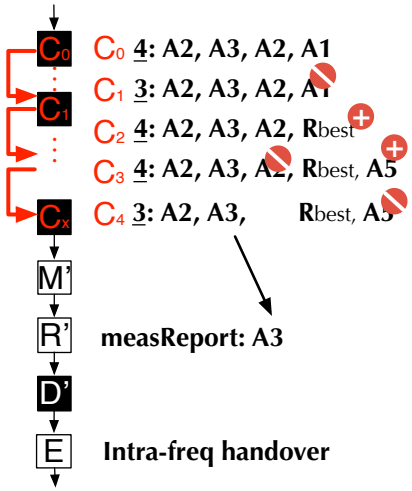


Figure 2.21.: A real-world AT&T handover instance.

This is not the only example. In fact, similar results are common in all four US carriers: **A**T&T, **T**-Mobile, **V**erizon and **S**print. All these operators change their configurations over time. In another example of an intra-freq handover in AT&T

(Fig. 2.21), we have observed five rounds and many more re-configuration options which add and then remove certain events, like A1 from C_0 to C_1, A2 from C_2 to C_3 and A5 from C_3 to C_4 . It implies that a trial-and-error approach is in use. As a result, we believe that our previous measurement study is limited as it simply characterizes the distribution of all the parameter values observed, *e.g.*, -107, -111, -121 for A5 events' threshold for the serving cell (Θ^s_{eA5}), regardless of their dynamics and adaptive purposes at different rounds. Such approach fails to capture the essence of a handover process. It is not enough to understand and explain operators' handover policies without considering dynamics in mobility management.

Problem Statement. We seek to understand and explain mobility management by examining its dynamics. The above examples show:

1. One handover may require multiple-round configurations which vary to control measurement and reporting.

2. Each round may define multiple events of the same or distinct types and parameter settings.

3. The current-round configurations seem to be the consequence of the previous ones and measurement results, but this logic behind is unclear.

We believe that the operators must run a set of algorithms to calibrate, adapt and adjust handover configurations in the mobility management plan (the blue loop in Fig. 2.19). It is likely true for the operators not to adopt the default configurations set by their vendors (otherwise, the parameter values should be the same or similar). It is also hard to believe that the operators configure or reconfigure these parameters in an ad-hoc or manual manner, given their infrastructure scale (*e.g.*, millions of cells or more for any top-tier carrier in the US). Here, we validate our hypothesis and attempt to infer the logic behind handover configurations. Although the configuration logic and algorithms are operator-specific and likely proprietary, we demonstrate that the majority is learnable out of the sequence of configured parameters and device-side observations. Given the exposed management logic, we give a better understanding of configuration heterogeneity and diversity.

2.3.2 Experiment Methodology And Dataset: MMReal

Figure 2.22.: Driving Route of MMReal.

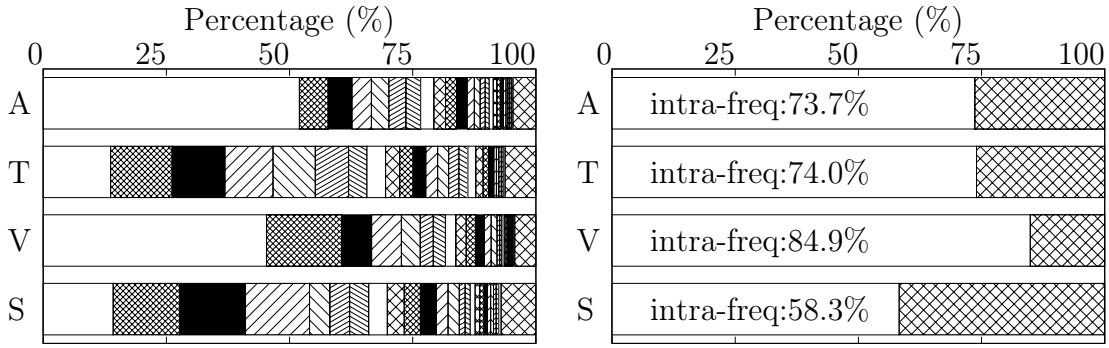
In order to capture the dynamics of mobility configurations in run-time, we develop task MMReal over MI-Lab to collect all RRC messages while keep sending PING (Google) every second. PING traffic keeps the device active all the time if no loss of connectivity. Although ping traffic is different with data traffic in real usage and brings bias to the serving cell load, RSRQ measurement result and consequently network handover behavior, our local field experiment with both elephant and mouse traffic flows show that the mobility management logic behind configuration iterations remains consistent. We run MMReal to collect all the signaling messages relevant to handovers when we drive along local and highway routes (see the map in Fig. 2.22). We collect data sporadically in Feb 05 – April 15, 2019 and heavily in April 16 - 30, 2019 using four top US carriers. In this study, we consider 4G LTE only because our test routes are almost fully covered by 4G in all four carrier networks.

Tab. 2.4 shows our dataset statistics. We drive around 4,320 miles and collect 33,647 handovers over 20,140 cells in four US carriers. For each handover, we extract the sequence of configured events and measurement reports out of the signaling messages (in *RRCConnectionReconfiguration* and *MeasurementReport*), as illustrated

Table 2.4. Statistics of MMReal dataset.

	Hours	Miles	Cells	HOs	Rounds	Events
A	114 hr	3,594 mi	4,070	5,543	27,088	35,296
T	136 hr	4,190 mi	5,925	10,222	35,224	80,339
V	143 hr	4,320 mi	5,634	10,005	24,855	52,890
S	102 hr	3,215 mi	4,511	7,867	16,307	69,735
Total			20,140	33,647	103,474	238,260

in Example 1 (Fig. 2.20). *RRCConnectionReconfiguration* also carry active handoff commands and all these messages carry the serving cell ID. We track the serving cell ID over time and detect a handover via its change. It is an active handover when it is realized via *RRCConnectionReconfiguration*. We also record the serving cell's frequency channel and learn whether it is an inter-freq or intra-freq handover. For any *RRCConnectionReconfiguration* messages before a handover, we extract all the configured events if applicable. We mark it as a new configuration round if there is any change. In total, we observe 103,474 configuration rounds and 238,260 events. We show dataset breakdown in Fig. 2.23 and Fig. 2.24.



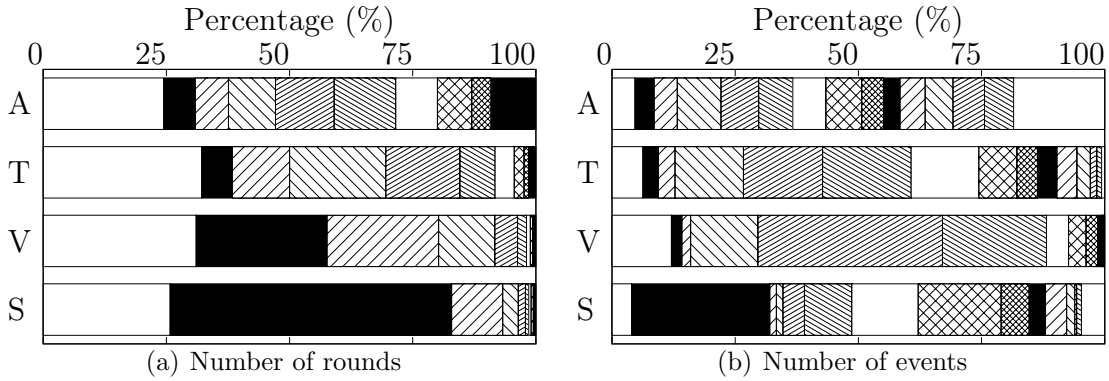
(a) Cells over channels (b) Intra-freq vs. inter-freq

Figure 2.23.: Dataset breakdown on cell channels and handover types.

Cells over a variety of frequency channels. All the four carriers use a number of frequency channels (A: 76, T: 90, V: 85, S: 90 observed in our dataset). Fig. 2.23(a) plots the percentage of cells over top-25 frequency channels in descending order. We have two observations. First, the use of each frequency channel is not equal. 75%

of cells use a small number of (*i.e.*, 5-10) channels. In AT&T and Verizon, there is one single channel that is dominant and contributes to almost half of their cells: channel 5110 (band 12, 739 MHz, 52%) for AT&T and channel 5230 (band 13, 751 MHz, 45.3%) for Verizon. Second, the use of frequency channels are carrier-specific. In contrast, T-Mobile and Sprint use top-4 channels more evenly than AT&T and Verizon. This is partly dependent on their cell deployment and channel assignment, and also partly related to their handover management (revealed later).

Handover: intra-freq vs inter-freq. We plot the breakdown of intra-freq and inter-freq handovers per carrier in Fig. 2.23(b). We see that most handovers are intra-freq (A: 73.7%, T: 74%, V: 84.9% and S: 58.3%). This implies that the current dense cell deployment provides same-freq candidate cells in proximity and intra-freq handovers are likely preferred (also validated in our logic inference).



(a) Number of rounds (b) Number of events

Figure 2.24.: Dataset breakdown on number of configuration rounds and events.

Configuration rounds. Fig. 2.24(a) shows the handover breakdown in terms of the number of configuration rounds (from 1 to 9 and > 9). Not surprisingly, most handovers complete within four rounds except AT&T, but we also see that some handovers go up to 10 rounds or even more. The average numbers for AT&T, T-Mobile, Verizon and Sprint are 4.9, 3.4, 2.5 and 2.1 respectively. Later we will show that this is because of carrier-specific configuration schemes.

Configured events. Fig. 2.24(b) shows the number of the configured events uniquely observed among all the handovers. Note that some configured events may remain the same at the next round (see Example 1 with those over channel 1051 in the second round). The average numbers of observed events are 6.4, 7.9, 5.3 and 8.9 for AT&T, T-Mobile, Verizon and Sprint, respectively. Most have more than 3 events in use, except that Sprint uses only 2 events in 28% cases.

In a nutshell, we can see that handovers along with their configurations are quite dynamic and carrier-dependent.

2.3.3 Understanding Dynamics In Mobility Management

In this section, we first characterize how handover configurations vary and then infer the logic behind.

1) Characterizing Dynamics. The management plane controls and manages a handover by configuring the triggering events of measurement reporting. Each event e is represented by its type and a list of parameters, namely, $e(type, @params)$ where $@params$ is a vector of $(name, value)$ pairs. As illustrated in Fig. 2.20, the key parameters include the frequency channel to measure (Freq), radio quality metric (P or Q), and radio quality thresholds or offsets which are event-specific. A sequence of all configurations for one handover can be represented as follows, where $c_k[i]$ represents the i-th event defined at round k, where n_k is the number of configured events at round k. $c_k[i] \in \{e_x\}$, where e_x denotes any eligible event which might be configured by the operator. $C_k = \{c_k[i], i = 1, 2, \cdots, n_k\}$ is the set of all configured events at round k, $k = 1, 2, \cdots K$, where K is the number of rounds. Therefore, a sequence of all configurations for one handover can be represented as $[C_1, \cdots C_K]$. Clearly, it is high-dimensional and hard to characterize all the varying parameters together.

$$\begin{aligned}
\texttt{Round 1}\ (C_1): &\quad c_1[1], c_1[2], c_1[\cdots], c_1[n_1] \\
\cdots &\quad \cdots \\
\texttt{Round k}\ (C_k): &\quad c_k[1], c_k[2], c_k[\cdots], c_k[n_k] \\
\cdots &\quad \cdots \\
\texttt{Round K}\ (C_K): &\quad c_K[1], c_K[2], c_K[\cdots], c_k[n_K]
\end{aligned}$$

To understand its dynamics and challenges for dynamics characterization, we examine the events at each round. In Fig. 2.24(a) and Fig. 2.24(b), we see the distribution of K and the number of unique events in all the rounds over handovers in the wild. For each US carrier, more than 70% handovers require the second round. For simplicity, we examine the event type first, regardless of its parameter values. For any type e out of all the six event types (A1-A5 and R_{best}), we assess its contribution at round k by its occurrence. That is,

$$\lambda_k[e] = \frac{\text{Number of handovers which use event e at round k}}{\text{Number of handovers which have round k}}$$

Fig. 2.25 plots the percentage of event A1-A5 and R_{best} adopted at the first several rounds, compared with its adoption in one handover in any rounds. We have three observations.

First, the use of event types are dynamic. Some events are not used initially but are invoked later, such as A5. Its usage increases significantly at round 2 and later comparing their values at round 1 and all rounds. Note that the number of handovers with round 1 is the same as the one with all rounds. Other events are used at the start, but not so much later, such as A1 in AT&T and T-Mobile. However, for Verizon and Sprint, the use of A1 grows after round 1.

Second, the use of event types are carrier-specific. In addition to A1 (which is not commonly observed in Verizon), we also observe carrier-specific usage in A4, A5 and

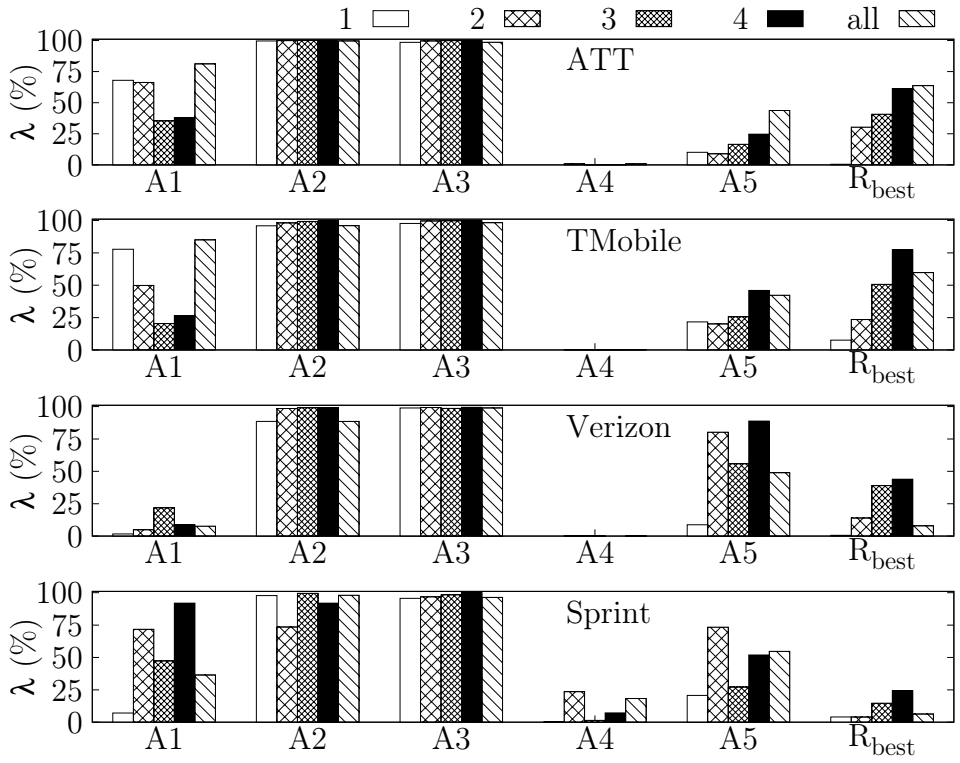


Figure 2.25.: Occurrence of event types at the first four rounds and all rounds before an active-state handoff.

R_{best}. For example, Sprint uses A4 more than other three carriers but R_{best} much less. We later show that this is determined by the information carried by these events and carrier-specific implementation.

Third, we observe that some events are popular all the time, like A2 and A3, for all the carriers. However, it does not mean that they vary less overtime or their use is common among all the carriers. Instead, this is because the occurrence metric is insufficient to capture all their dynamics. We observe that they change the frequency channel to measure or update the parameter values over these rounds. However, characterizing high-dimensional changes is challenging. We next define a new model to directly track dynamics.

2) Modeling Dynamics. In order to better characterize its changes and infer the logic behind, we use a state machine to model configuration dynamics (Fig. 2.26).

Each state is the set of configured events at round k. A state transition is triggered by certain inputs (*e.g.*, measurement reports from devices, no measurement reports within a time window) and performs certain actions (*i.e.*, change current applied configurations or make a handover). For round k to round k+1, there are four types of configuration changes made by three atomic operations: *null (doing nothing), deletion,* and *creation*, as shown in Fig. 2.27. For any particular event at round k, it may disappear, remain the same or get value updated at round $k+1$. They are realized by *deletion*, *doing nothing* and *deletion* plus *creation*. We use *deletion* plus *creation* to represent an update in order to reduce complexity when we infer the configuration logic. We observe that there are many forms of updates: (1) change the frequency channel to measure, (2) change the value of a threshold or an offset in radio quality comparison, (3) change the radio evaluation metric (P or Q). We also observe that one event can be split to multiple events. For example, two A3 events are observed at round k, one in RSRP and the other in RSRQ. Tracking all these updates needs a huge state machine. By using two atomic operations, we are able to represent all the updates and make it easy to decouple complexity. For those events which do not appear at round k but at round $k+1$, they all are added by the *creation* operation.

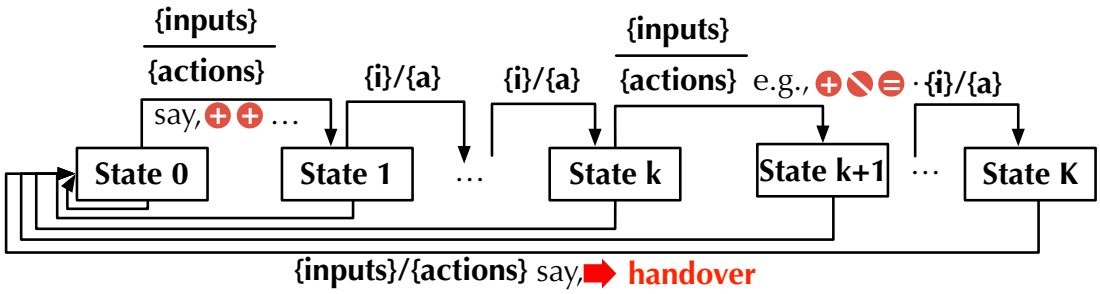


Figure 2.26.: Modeled state machine for handover reconfiguration.

Clearly, the initial state is a null set and only the *creation* operations are performed from round 0 to round 1. Once a handover occurs, everything is reset and the configuration procedure goes back to round 0 for a new serving cell. The state

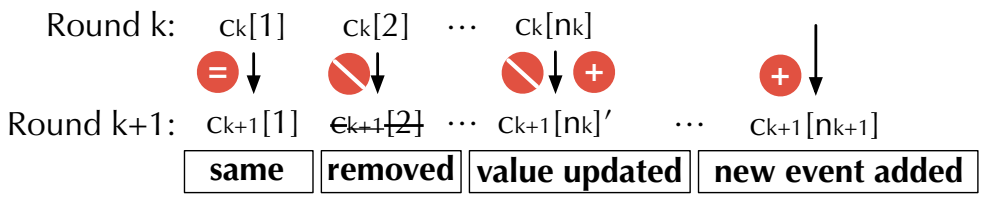


Figure 2.27.: Three reconfiguration operations.

dynamics is associated with three actions: *deletion*, *creation* and *handover*). Consequently, we can convert the delta in handover configurations into a sequence of these three actions which finally completes the state machine in the end.

3) Decoupling Complexity. Consider the sequence of mobility configurations before an active-state handover happens, if we track every single event related to all frequency channels and count any unique snapshot during this procedure as a state, then the total number of states could easily increase to a large number after multiple observations of handovers from multiple cells. This makes it extremely hard to infer the whole logic behind. Hence, we apply two heuristics to decompose the whole state machine inference into several small ones.

(a) The reconfiguration logic is independent of round k except $k = 0$. That is, the whole state machine can be decoupled into two parts: initialization and reconfiguration (Fig. 2.28). We believe that the reconfiguration logic is the same regardless of k ($k > 0$). The rationale here is that the number of rounds depends on varying environment and mobility, which is beyond the cell's control. Take an example of ping-pong mobility. The user first moves to one new spot where the serving cell is weaker and the candidate cell is stronger but not enough to invoke a handover. Upon receiving such reports, the cell may change the configuration and monitor whether the candidate cell becomes stronger or the serving cell becomes good back. If the user moves back to the old spot where the serving cell is stronger again, the cell changes the configuration back to monitor whether the serving cell becomes weak. In this example, the number of rounds depends on the user movement history and reconfiguration is independent of round number. More importantly, this makes simpler for a

cell to manage handovers for all the devices served by itself. For each device, the cell does not need to track all the configuration and measurement history but the latest ones. These heuristics are validated in our inference results.

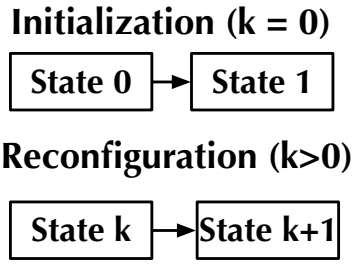


Figure 2.28.: Decompositions of the (re)configuration state machine over rounds.

This way, we can divide training samples to infer the logic for initialization ($k = 0$) and reconfiguration ($k > 0$). Each training sample for reconfiguration inference consists the serving cell information, the inputs at k and the current state k, with the output label as state $k + 1$. For initialization, each training sample contains the serving cell information with state 1 as the output.

(b) Each event has its own state machine. For one single event, its state is in use (1) or not (0). Its state machine is much simpler as illustrated in Fig. 2.29. The transition from state k to $k+1$ can be regarded as n_k state machines for each existing event and several state machines for newly added events. Through this decomposition, we do not need to track the configuration state as a whole which has an explosion problem. Instead, each event has two states and four state transitions. Among them, only two transitions are crucial: *creation* and *deletion*. That is, we treat the problem as inferring the logic to create an event and remove an event in the configuration state. We notice that there is one special case for handover. It is not an event but its occurrence terminates the configuration procedure. The key of this state machine is to learn when a handover is performed.

We would like to highlight the rationale behind such decomposition. It makes sense to manage each event configuration individually. Although some events seem relevant, they are primarily determined by runtime measurement results. Take Example 1

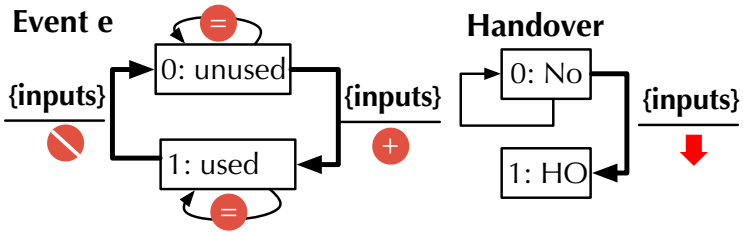


Figure 2.29.: Decompositions of the (re)configuration state machine by events.

(Fig. 2.20), the serving cell adds A3 and A5 events at three frequency channels upon receiving one measurement report (A2). The creation of A3 and A5 is indeed triggered by the same input. When we put all the inputs observed at round k as the input for one single event's state machine, we are able to decouple the creation for these two events. In this example, we add one transition in both A3 and A5's state machines. We observe that the number of inputs (measurement reports) is small and thus we can reduce complexity by inferring many small state machines rather than a huge one. To infer the state machine for each single event, we track its change (creation and deletion) across all the rounds to obtain the training samples.

(c) Pre-processing on frequency channel and parameter value. Before we infer the logic using these training samples extracted through the above two decompositions, we need some pre-processing for the frequency channel. Each event is associated with one frequency channel and multiple frequency channels are in use across the cells. However, what really matters is not the absolute channel number, but whether the frequency channel to measure is the same as the serving one or not. We observe that all the four US carriers use carrier aggregation (CA), an LTE-advanced radio technology [12] to use two types of cells: primary cell (PCell) and secondary cell (SCell) to serve the device. CA is used to aggregate frequency carriers and increase bandwidth, and thereby increase the data rate. PCell and SCell may use contiguous and non-contiguous frequency spectrum while most are non-contiguous in our dataset. As a result, there are three frequency channel types: the same as PCell, the same

as SCell and others. Hence, we convert an absolute channel number into such a three-value flag.

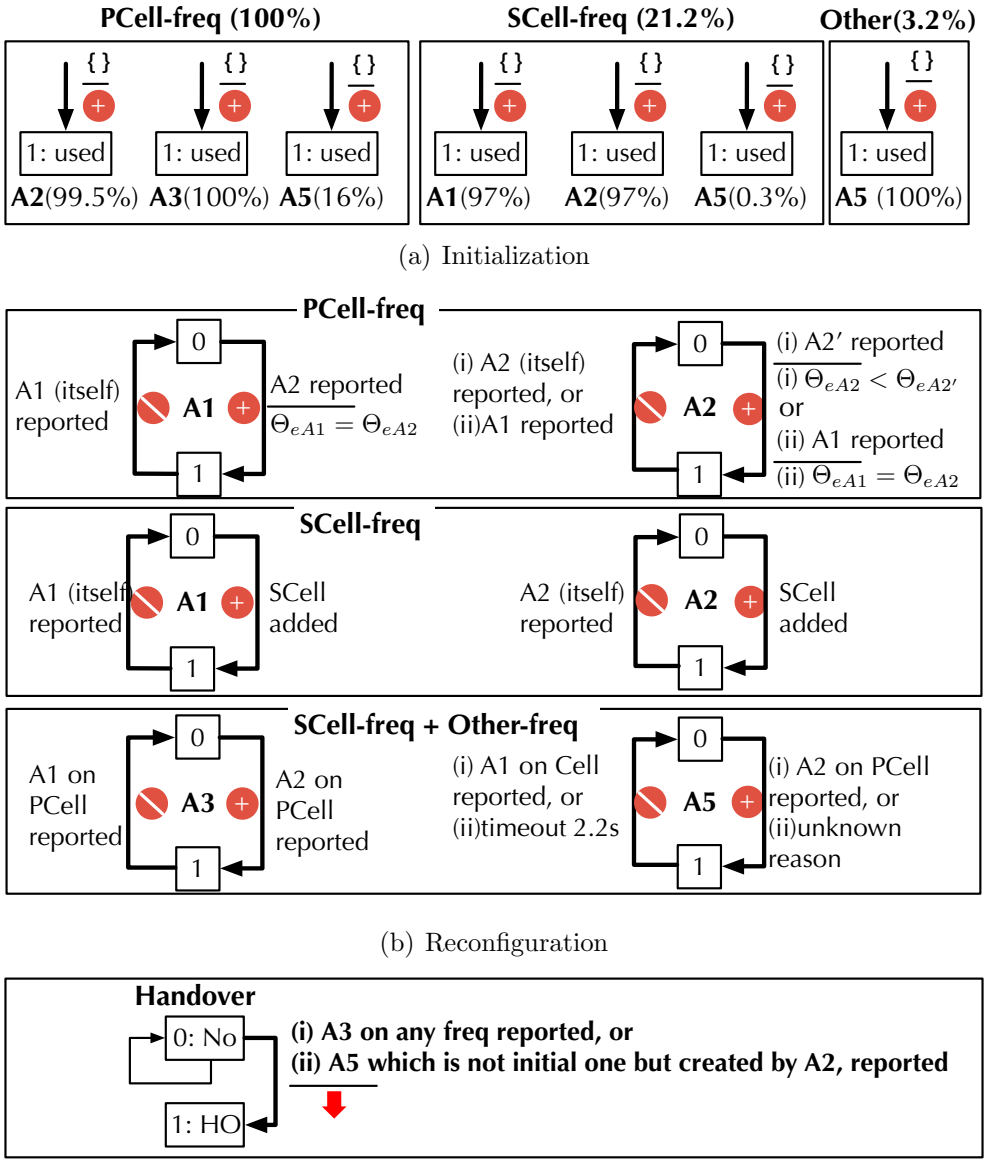
We also observe that many parameter values are in use and these values are cell-dependent. To make use of samples across the cells, we extract the features using the delta of the parameter values and thus there are three changes: being equal, larger or smaller.

2.3.4 Inferring The Configuration Logic Of AT&T

Now, we present our inference for the logic behind initialization (configuration), reconfiguration and handover. We use AT&T to illustrate our approach which is generic to all the carriers. The results for other carriers are presented in next section.

1) Inferring the Logic Behind Initialization

We extract all the events that appear at round 1 and learn their creation logic during the initialization. We find that the number of events is very small regardless of their parameter values. We gauge that the parameter values are cell-specific and thus use the event type to infer the common logic. Fig. 2.30(a) presents the results. We find that the initial configurations are determined by the frequency channels of PCell and SCell if applicable. Almost all the cells set A2 and A3 events over the PCell's frequency channel. In particular, A2 occurs with one instance (53%), two instances (46.5%) and none (0.5%) while A3 occurs with one instance (95%) and two instances (5%). We look into these cells and try to classify the cells with one or two instances based on the observations on the device. Unfortunately, they are not classifiable because the information observed on the device is limited. While it is unclear what cells use one or two events, it is clear that AT&T assigns A2 and A3 events over the same primary frequency channel at the start. We also observe that one A5 event is assigned in 16% cases. We can only see that this is cell-specific because the limited information on the device cannot distinguish the cells with A5 and those without A5.



(a) Initialization

(b) Reconfiguration

(c) Handover

Figure 2.30.: Inferred logic for AT&T.

We find that AT&T configure A1 and A2 events on the frequency channel of the SCell if existing. Currently, at most two SCells are observed. Its existence is detected in the *RRCConnectionConfig* message which explicitly adds one or two SCells. Once a SCell exists, the pair of A1 and A2 events with the same threshold value is assigned (97%). This is interesting because A1 and A2 are complementary and mutually

exclusive. This implies that one event will be triggered immediately. We also observe that A5 is rarely configured at the start (0.3%) for some unknown reason. We find that most initial configurations are restricted to the serving cell's frequency channels. The events over other frequency channels are configured only in 3.2% cases. The only configured event is A5.

2) Inferring the Logic Behind Reconfiguration

We track the changes for one event after the initialization stage. Across the rounds, we focus on those with either creation or deletion operation and ignore those without changes. We pair each operation with its triggering inputs, *i.e.*, measurement reports just before this reconfiguration operation (creation or deletion). Our measurement shows that most reconfigurations happen within 200 ms after the measurement report. So we believe that the reconfiguration is resulted by this report if it is within 200 ms. If the reconfiguration happens without any measurement reports, we believe that it is associated with no measurement report. Here, we would like to emphasize that no measurement report also delivers the surrounding information to the serving cell. It acts as an NACK and notifies the serving cell that the device does not have any measurements which are qualified for reporting. By applying the decompositions over round and per event, we derive a large number of training samples per event, regardless of its round number. Each sample is associated with either creation or deletion, with the measurement reporting (including no reporting) as the inputs. Fig. 2.30(b) plots the inferred logic for main events.

We observe the state transitions for four events: A1, A2, A5 and R_{best} (periodic measurement to report the strongest cell) on the PCell's frequency channel. For A1, its creation is caused by A2 event being reported, namely, the PCell finds its self weaker than certain threshold. The newly created A1 event has the same threshold equal to the one of the reported A2 event. A1 will be deleted once itself is triggered. For A2, it can be created under either of two conditions. First, it is created when another A2 event being triggered and the newly created A2 event is assigned with one threshold lower than the original one. This actually reflects how A2 updates its

threshold accordingly. Once the serving cell is below one threshold, it continues to monitor whether it will be even weaker. Second, it can be created with A1 event being reported and the assigned threshold of the A2 event equals to the one used in A1. A2 can be removed under one of two conditions: this A2 event itself being triggered or one A1 event being triggered. Actually, the life circle of A1 and A2 events together implies a clear logic behind reconfiguration. A1 and A2 complement each other and helps to monitor whether the serving cell is in an acceptable radio quality range. More are presented when we combine all the findings together. For A5 and R_{best} over the PCell's frequency channel, we do not identify any inputs associated with its creation. The logic can be similar to the creation of A5 during the initialization which is unclear. Once they are created, they keep alive till next handover or the RRC connection release.

On the SCell's frequency channel if any SCell is being used, we observe the state transition of A1, A2, A3 and A5 events. A1 and A2 both are created when a new SCell is being added. Note that this creation is the same as the one observed at the initialization. We gauge that the same logic is being used. Both events will be removed once itself is being reported. Moreover, we also find that the reporting of A2 leads to dropping the SCell. This is easy to understand. Once the SCell is not good enough, there is no need to retain it as SCell. We observe that these two events are only associated with SCell's activation and release. When we merge them together, we derive the life circle of a SCell (see Fig. 2.31(b)).

We also derive the state machine for the creations and deletions of A3 and A5 on non-PCell frequency channels (namely, SCell's frequency or other frequency channels). All are determined by A1 or A2 events on the PCell frequency channel. Specially, A3 or A5 is created when an A2 event on the PCell's frequency channel is reported, and gets removed when an A1 event is reported. Note that, A1 and A2 are to report the serving cell's radio quality measurement, and they measure PCell's or SCell's frequency channels only. In our study, we also notice a special class of A5 event. It is configured with -44 dBm (the maximum P value) for Θ^c_{eA5}. This indicates that its

reporting does not require anything from the serving cell (because it is always weaker than the maximum value). The creation of such A5 event is not related to any measurement report (for some reason which is unknown and invisible to the device side). But its deletion always occurs after 2.2s when no handover happens even if it is reported. This is a special A5 event to be discussed later. It indicates that such A5 event is associated with a timer and is removed upon a timeout.

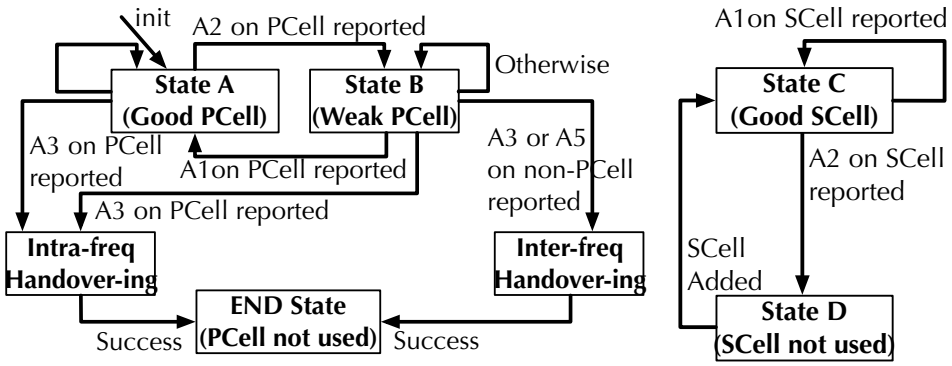
3) Inferring the Logic Behind Handover

We next infer when a handover is triggered. We find that the reporting of A3 or A5, not other events, may result in a handover. For A3, its reporting is surely associated with a handover, regardless of its parameter values, frequency channels or creation at the start or later. This indicates that AT&T uses the A3 reporting (the candidate cell's radio quality is offset stronger than the serving one) as a decisive criterion.

In contrast, we find not every A5 reporting leads to a handover. We thus look into these examples and learn the exception rules. We find that the A5 event created at the start will not lead to a handover. For the A5 event which is created afterwards, we find that the one created out of A2 reporting will surely lead to a handover. For the other A5 event which could be added not out of A2 reporting at any moment over any non-PCell frequency, it keeps alive for about 2.2 seconds and eventually triggers a handover or gets removed after timeout. We fail to identify a clear rule for the reporting of such A5 event as it may or may not trigger an inter-freq handover. Both consequences are commonly observed in our dataset. An interesting finding is that such A5 event is always assigned with the maximum threshold value for Θ^s_{eA5}, *i.e.*, -44 dBm in RSRP. This implies that such A5 event does not take into account the serving cell's radio quality. As long as there is a neighboring cell with radio quality better than another threshold Θ^c_{eA5}, A5 will be reported. This makes it easier for the serving cell to receive measurement reports from the device.

4) Putting them together

We now merge the logic fragments previously inferred into a complete state machine. We combine these inferred logic with the same triggering conditions and find that there are two main state machines running for PCell and SCell as shown in Fig. 2.31.



(a) PCell (b) SCell

Figure 2.31.: Handover management in AT&T.

(a) PCell. We find there are generally two main states when a PCell is in use: one state (A) means the PCell offers good radio quality and only considers intra-freq neighboring cells as candidates, and the other state (B) means the PCell's radio quality is weak and it considers both intra-freq and inter-freq neighboring cells. A1 and A2 events over the PCell frequency channel are configured at these two states respectively (A2 at state A and A1 at state B). They are used to control the transitions between these two states.

State A is the initial state. The serving cell always starts form state A. It configures to primarily monitor intra-freq A2 and intra-freq A3 events. It is deemed efficient given the overhead to monitor inter-freq neighboring cells. There are three transitions in this state. First, if the PCell's radio quality is so weak that invokes A3, it decides an intra-freq handover immediately and moves to the END state where this PCell is not in use any longer after a successful handover. Second, when the PCell's radio quality is not enough to invoke A3 but trigger A2, it switches to State B. This is because that it is inadequate to consider intra-freq neighboring cells only when its

radio signal becomes weak and there are no good intra-freq candidates. A2 event is exactly used to notify the PCell for such change and aim to balance effectiveness (not missing a potential handover) and efficiency (low measurement load).

Once entering into state B, the serving cell configures an intra-freq A1 event and adds inter-freq A3 and/or A5 events on each candidate frequency channel. Here the A1 event is set with the same threshold with previous A2 event. It allows the serving cell to be notified when the radio signal turns back to the previous good level so that it can recover to state A. The A3/A5 event is used to monitor inter-freq neighboring cells for a potential inter-freq handover. For AT&T, we observe that it may initialize one or two A2 events at the beginning. For each A2 event, its reporting brings the configuration of more inter-freq A3 and/or A5 events. This is because AT&T uses two sets of events each with one radio quality metric, namely, RSRP (P) or RSRQ (Q). This strategy provides the serving cell with a mutually referenced view regarding the change of radio signal quality. In state B, any reporting of these new added A3 or A5 event right away triggers an inter-freq handover. Once it succeeds, the serving cell is not used any longer. If one A1 event is reported, it leads to adding the previous A2 event back and removing the set of all inter-freq A3 and/or A5 events associated with this A2 event. If two A1 events are used and both are reported, no inter-freq A3/A5 events remain and the serving cell switches back to state A.

(b) SCell. We find a much simpler and standalone state machine to manage the SCell. There are only two states: in use (State C) and not used (State D). Once a SCell is added, it enters into state C with the pair of A1 and A2 events being created to monitor the radio signal of this SCell. Both events are set with the same threshold value. In State C (a good SCell), the reporting of the A1 event will delete itself but not influence the state of this SCell. It will retain the A2 events to monitor if SCell becomes weaker. Only when the A2 event is reported, it means the radio signal of SCell is below the bottom level and will be dropped immediately. This logic works independently with the one at PCell. It is aligned with the standard specification on SCell activation/release procedure [12] and also matches with our common sense.

5) Inference Accuracy

We admit that there is no ground truth from the operator to verify the inferred algorithm. We evaluate the accuracy in two means: (1) 5-fold cross-validation widely used to evaluate classification accuracy. (2) explainable results which can be easily understood and agreed by humans.

Table 2.5. Inference accuracy for AT&T.

PCell Transitions	**Accuracy**	**SCell Transitions**	**Accuracy**
A2 → State B	97.57%	A2 → Drop	96.6%
A1 → State A	99.7%	A1 → Keep	99.67%

	Intra-Freq	**Inter-Freq**
A3 → Handover	98.37%	98.55%
	Inter-Freq	**Special A5***
A5 → Handover	100.0%	79.1%

In the cross-validation evaluation, we randomly break the dataset into five folds and use 80% samples for training and the rest 20% for inference. We repeat it for five times. We evaluate the inference accuracy by checking whether the expected actions are taken when the triggering conditions in the inferred logic are satisfied. We uncover the reconfiguration state machine through a small dataset collected at local area and find it is generally followed in our large real-world dataset. Table 2.5 shows the inference accuracy for each main transition in the inferred state machine. The inferred logic can accurately predict the real-world reconfiguration instances with more than 96.6% (the exception is the A5 special event explained next).

Recall Example 2 (Fig. 2.21), now we can explain completely what happens in such a five-round configuration procedure. Each round matches with our inferred logic. Initially user device connects with PCell 453 on channel 5110 (band 17, 739MHz) and a SCell 54 on channel 700 (band 60, 1940MHz). Network configures A2 and A3 events with RSRQ metric over PCell frequency, A1 and A2 events over SCell frequency with the same RSRQ threshold (-19.5 dB). After that, the serving cell receives reporting of the A1 event, the configuration of A1 is deleted at round 1. After a while, the serving cell receives the reporting of A2 over the SCell channel. As expected, the SCell is

dropped at round 3. In this process, our inferred logic fails to predict the occurrence of R_{best} added over PCell channel at round 2 without any inputs from the device. We gauge that network still runs some logic which is not exposed to the device side. After dropping the SCell, a special A5 event is added on the channel of 700 (the same as the original SCell) at round 3. This is used to monitor another inter-freq channel. Note, it is not created by the reporting of A2 over the PCell channel and its deletion is controlled by a timer (it is removed finally after 2.2 seconds). Eventually, this example ends with an intra-freq A3 reporting which triggers an intra-freq handover to another cell over channel 5110. Our inferred logic accurately predicts the occurrence of each reconfiguration except R_{best} that using standalone logic at network. Example 2 illustrates a reconfiguration-handover instance where the radio quality of PCell and SCell becomes weak on the go and SCell is dropped first and then an intra-freq cell is found to replace this PCell. Given the inferred logic, almost all the instances can be predicated and explained with rationale.

2.3.5 Carrier-specific Configuration Logic

We infer the logic for another three carries (T, V, S) and find that all four US carriers take similar state machines in their mobility management but with carrier-specific configurations. We highlight their distinctions and discuss their implications in this section. There are four main differences in (1) the transitions of State A and State B, (2) the SCell's state machine, (3) Handover and special A5 events and (4) miscellaneous.

1) State A $\leftrightarrow$ State B. One of the primary distinction lies in how to configure A1/A2 events to control the state transitions between A and B. Tab. 2.6 lists the typical configurations from each carrier to illustrate carrier-specific choices.

◦ *AT&T.* AT&T uses A1/A2 to monitor whether PCell is good enough or too weak. It may configure one or two events. By default, it configures the first event in terms of the Q (RSRQ) metric. In 46% cases, it uses two events and the other set

Table 2.6. Typical configurations in state B by four US carriers.

(a) AT&T

Freq Ch.	Event	Para	Value	P/Q
PCell	A1	thres	-15.0	Q
PCell	A1	thres	-122	P
PCell	A2	thres	-19.5	Q
PCell	A2	thres	-130	P
PCell	A3	offset	3.0	Q
PCell	R_{best}	-	-	P
non-PCell	A3	offset	5.0	Q
non-PCell	A3	offset	8.0	P

(b) T-Mobile

Freq Ch.	Event	Para	Value	P/Q
PCell	A1	thres	-108	P
PCell	A2	thres	-123	P
PCell	A3	offset	5.0	P
PCell	A5	thres	-112,-111	P
PCell	R_{best}	-	-	P
non-PCell	A3	offset	3.0	P
non-PCell	A5	thres	-111,-110	P

(c) Verizon

Freq Ch.	Event	Para	Value	P/Q
PCell	A1	thres	-111	P
PCell	A1	thres	-136	P
PCell	A2	thres	-116	P
PCell	A2	thres	-140	P
PCell	A3	offset	3.0	P
PCell	A5	thres	-13,-18	Q
non-PCell	A5	thres	-120,-118	P

(d) Sprint

Freq Ch.	Event	Para	Value	P/Q
PCell	A1	thres	-75	P
PCell	A3	offset	3.0	P
8465	A4	thres	-113	P
8665	A5	thres	-117,-116	P
8763	A5	thres	-105,-117	P
40270	A4	thres	-110	P
40978	A4	thres	-113	P
41176	A4	thres	-113	P

uses the P (RSQP) metric. So AT&T counts on RSRQ more than RSRP. If two sets of events are configured, they work independently. Tab. 2.6(a) gives an example in state B after both two sets of A2 events are reported. Here we can see that AT&T configures two A3 events for each non-PCell frequency channel. Each A3 is associated with each A2 event reports separately. In contrast, other carriers use one set of A3/A5 events only.

- *T-Mobile.* T-Mobile prefers the P metric and uses this single set in 74% cases. Tab. 2.6(b) shows a typical snapshot in which only one set of A1/A2 is assigned. Both inter-freq A3 and A5 events are controlled by this single set.

- *Verizon.* For Verizon, 92% of cells indeed use two sets of A1/A2 events, but most of them (87%) use the P metric in both sets. When two sets of A1/A2 events in the same metric are used, they co-work together to determine the PCell state and invoke the one single set of inter-freq events of A3 and/or A5. Tab. 2.6(c) gives a

typical example. In this example, only when A1 event with higher threshold (-111 dBm) is triggered, the PCell switches back to state A.

◦ *Sprint.* Sprint is significantly different from other carriers. Most cells (83%) use one set of A1/A2 events in the P metric. But the A2 event usually get removed once it switches to state B. Other three carriers do create another A2 events with lower thresholds to monitor the PCell. Another difference is that Sprint's configuration is frequency-specific and it uses A4 events. Other carriers likely use the same configuration for each non-PCell frequency channel. Here, an A4 event is equivalent to a special A5 event (Θ^s_{eA5} = -44 dBm, $\Theta^c_{eA5} = \Theta^c_{eA4}$). Its reporting leads to an inter-freq handover like A3 or A5 specified before. Tab. 2.6(d) gives an example.

2) SCell's state machine. Four US carriers apply the same logic and similar strategy except Verizon. All the other three carriers configure a pair of A1 and A2 events but Verizon configure one A2 event only for a new added SCell without A1 events. Actually, it makes sense since the carrier should care only when the used SCell becomes weaker, which can be handled by A2 only.

3) Handover and special A5 events. We observe that all the four carriers treat A3 in the same way. Namely, a reporting of any A3 event invokes a handover. They differ in handling A5 reporting. AT&T and Verizon rarely invoke an handover upon an A5 reporting unless the A5 event is added by an A2 event being triggered (except those special A5 events). In T-Mobile and Sprint, the intra-freq A5 events assigned at the initialization round can also lead to an intra-freq handover. It indicates that both carriers use A3 and A5 for intra-freq handovers while the other carriers use A3 only.

Recall the special A5 events with Θ^s_{eA5} = -44 dBm (P), namely A4. AT&T adds them for some unknown reason and remove them after 2.2 seconds if no handovers occur even with being reported. This is also observed in T-Mobile and Verizon. Handover upon the reporting of this special A5 event is also not guaranteed (32.83% for T-Mobile and 69.73% for Verizon). Given the observation that they are only used on one specific non-PCell channel (e.g., the one used by the dropped SCell), not over

all the non-PCell channels in a short period, we gauge that this is a mechanism used by the carriers to dynamically control the trade-off between intra-freq and inter-freq (special) handovers. It seems to load balance between the serving cell and one specific inter-freq neighboring cell.

4) Miscellaneous. There are some carrier-specific configurations which are uncommon. For example, we find 1.8% Sprint cells deploy more than one A3 events at the initialization round and the extra A3 events have a timer of 0ms. Once the measurement reporting condition is satisfied, it generates a batch of measurement reports, but no handover happens thereafter. It is unclear what is the rationale behind such configuration. We also find some Verizon cells use two A1 events with exactly the same threshold value (*i.e.*, , -136 dBm RSRP) in state B. Such values cause 3.8% instances of A1 measurement report being sent duplicately, which is unnecessary.

Table 2.7. Inference accuracy for T-Mobile, Verizon and Sprint.

State Transitions	**T-Mobile**	**Verizon**	**Sprint**
A2 → State B (PCell)	99.3%	99.0%	99.8%
A1 → State A (PCell)	92.9%	91.9%	100%
A2 → Drop (SCell)	98.5%	100%	100%
A1 → Keep (SCell)	99.4%	N/A	98.2%
A3 → Handover (intra-freq)	99.7%	99.8%	99.8%
A3 → Handover (inter-freq)	99.1%	N/A	N/A
A5 → Handover (inter-freq, not special)	99.7%	100%	100%
A5 → Handover (inter-freq, special)	32.8%	69.7%	50%
A4 → Handover (inter-freq)	N/A	N/A	100%

Tab. 2.7 shows our inference accuracy for other three carriers. Except the special A5 event, we have achieved high accuracy in predicting most state transitions. Verizon and Sprint use A5 instead of A3 to invoke an inter-freq handover. A4 is used by Sprint only. In a nutshell, each carrier has its own strategy to determine whether it is necessary to monitor inter-freq neighboring cells or not and how. Because these strategies are distinct, an interesting question is which works best in reality? We believe that understanding their management plane in different carriers may help us to learn from each other to design a better management.

2.4 Summary

In this chapter, we answer how does mobility support perform in operational cellular network. We first give an introduction about basic handoff procedure and exhibit how mobile network carriers are feasible to tune the conduction of handoff at fine-grained level by deploying policy-based mobility configurations at each cell.

Next, we conduct a sizable measurement study on policy based handoff configurations from 30 mobile carriers in the US and globally. To this end, we design a new device-centric tool, MMLab, which collects runtime handoff configurations without the assistance from operators. Through our analysis, we exhibit that extremely complex and diverse configurations are deployed by operators in reality.

Furthermore, we present the first work to expose mobile network carriers' mobility management policy, which is confidential and even proprietary. We propose a generic device-centric approach to characterize, model and track configuration dynamics and infer the policy behind. Through this closer look, we have a deeper understanding of how four US carriers dynamically manage their mobility support in the wild.

Given these efforts, we are capable to explain the behaviors at both client and network side during a handoff procedure in operational cellular network and the rationale behind.

3 WHAT IS WRONG WITH STATE-OF-THE-PRACTICE MOBILITY SUPPORT?

Cellular network relays on mobility support to ensure "always connected" network access. However, it is often insufficient to accommodate nice user experiences. The advance from "always connected" to "always well connected" concerns the quality of mobility management, especially when today's cellular networks become heterogeneous and dense deployed so that they can offer many access choices at one location. In this chapter, we study the quality of mobility management and examine why the current practice and design cannot guarantee "always well connected" experiences. We first start from the handoff function correctness and identify misbehaviours in handoff procedure that leads to two undesired handoff results: handoff instability and unreachability. Secondly, we study how diverse handoff configurations affect data performance and lead to unexpected compound effect to performance and efficiency. Last but not least, we unveil the data performance gap in today's 4.5G LTE advanced cellular network which prevents user devices to fully utilize the network performance potentials.

3.1 Handoff Instability and Unreachability

Handoff process is iterative and distributed in nature. There is no central point which collects all the information and makes decision. Instead, each decision is made locally at a cell or by the mobile device. The target cell is selected by each handoff decision made by the current serving cell or the mobile device. However, such design leaves potential flaws in handoff decision and executions.

In this section, we first show that uncoordinated configurations among cells can lead to handoff instability: the mobile device oscillates between a set of cells covering

the same area, even when no radio-link or location change is detected. Surprisingly, we find load balancing is not the main concern and there are other key drivers to handoff instability. Our modeling of handoff instability generally follows a discrete-event style. Each handoff is abstracted as a transition from serving cell s to target cell t among multiple candidate cells C. The handoff execution for $s \rightarrow t$ is acted on the serving cell s and the mobile device. So after handoff to a new cell, the execution would change. Consecutive handoffs may occur even with the same observations (*e.g.*, no location/radio condition change). Stability is ensured, if for any invariant observation, a device initially associated with any cell s will always converge to the target t but not move to other cells, *i.e.*,

$$\mathbf{s} \rightarrow c_1 \rightarrow c_2 \rightarrow \cdots \rightarrow c_k \rightarrow \mathbf{t}, \qquad c_x, t \in C. \tag{3.1}$$

If this property is violated, a persistent loop can happen between a set of cells even for some unchanged measurements (or measurements fluctuating within a small range). User would experience data/voice performance degradation. Carrier cannot achieve the designated handoff goal, and may suffer from excessive signaling overhead. Note, a handoff is assumed to always succeed without failures (*e.g.*, we ignore radio-link outage). In this work, we classify the handoff instability cases based on the causes of the configuration conflicts. Broadly speaking, there are two classes: parameter misconfiguration, and loop-prone decision logic. We derive the instability condition for each case.

We next (in)validate the existence of each handoff loop and quantify its negative impact by conducting experiments in two metropolitan areas from both west and east coasts over two top-tier US carriers: AT&T and T-Mobile. The validation takes two steps. First, we develop a loop detection tool over *MobileInsight* [7] to check the carriers' handoff policies. The detection algorithm is based on our analytical (in)stability results. It reports the stability violations and its condition for runtime observations. Second, for each stability violation, we conduct validation experiments

to test its existence, and quantify its negative impact if it exists. We run both outdoor and indoor experiments. The outdoor experiments cover 63 different locations over 240 km^2 in the west coast and 260 km^2 in the east coast. Each location is selected by at least 2 km apart, in order to obtain different cell coverage. We also collect information on indoor experiments at 50 spots in two 8-floor office buildings and one apartment. In the indoor settings, we mainly collect the radio quality observations at various spots, since most cells, as well as their configurations, are similar across locations. We also deploy four 3G femtocells in office and at home for indoor tests. We use four Android phone models: Samsung Galaxy S4, S5 and Note 3, and LG Optimus G.

In addition to handoff instability, we also discuss the handoff unreachability problem by examining how well a handoff process converges (handoff instability is the non-convergence case of handoff process). We concern whether the handoff settles down at the most preferred target cell (in other words, whether the optimal cell is reachable through the actual handoff process). The most preferred target cell is defined as the one that maximizes the utility function (*e.g.*, yields the best performance) by the operator or the user, given certain network conditions. It satisfies that

$$t_{opt} = \arg \max_{c \in Candidates} \Psi(c), \tag{3.2}$$

where $\Psi(\cdot)$ represents the utility function of our interest. For example, it aims to maximize the available throughput to the users. This represents a globally optimal choice regardless of whether it is reachable through the distributed, iterative handoff process. The violation against the optimization occurs when the handoff fails to settle down at t_{opt}, namely, the handoff walks over iterations and converges to another choice.

$$\mathbf{s} \rightarrow \cdots c_i \rightarrow \cdots \mathbf{t}, \qquad t \neq t_{opt}, c_x, t \in C. \tag{3.3}$$

In our study, we select the optimization target of desired cell simply based on common wisdom, *i.e.*, 4G > 3G > 2G unless the preferred cell has weak radio signal.

In principle, there are two classes of violations: convergence split and premature convergence.

Finally, we discuss the implications we learned and potential fix.

3.1.1 Instability By Uncoordinated Parameter Configuration

In this category, the instability is observed when cells' tunable parameters are not well coordinated. We have discovered three cases of uncoordinated parameter configurations as illustrated in Fig. 3.1, Fig. 3.5 and Fig. 3.6.

1) Inconsistent Preference Values. In this case, each cell locally configures its preference, but these preferences are not globally coordinated. Figure 3.1 illustrates a simple two-cell case. In this setting, c_1 configures c_2 to be more preferred to c_1 itself, but c_2 assigns equal preference to both cells. The persistent loop happens if the signal strength satisfies $\gamma_2 > c_1.\Theta^{c_2}_{higher}(-108$ dBm), and $\gamma_1 > \gamma_2 + c_2.\Delta_{equl}(3$ dBm). Note that, this loop can occur for any threshold settings (in the achievable range).

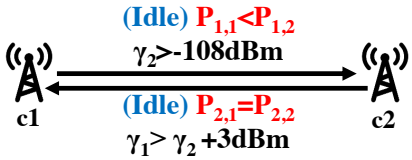


Figure 3.1.: Instability with uncoordinated parameter configuration: preference inconsistency.

We first derive the instability conditions for this category. Recall that in an idle-state handoff, when associated with cell s, the mobile device evaluates each candidate c with the pre-configured preference $P_{s,c}$, and its runtime signal strength γ_c. Each cell c is compared with serving cell s, and would be selected if

(1) it is more preferred than the serving cell, and its signal strength is higher than a threshold ($\gamma_c > \Theta^c_{higher}$),

(2) it is equally preferred, and its signal strength is offset higher than the serving cell's ($\gamma_c > \gamma_s + \Delta_{equal}$), or

(3) it is less preferred, but the serving cell's strength is weak ($\gamma_s < \Theta^s_{lower}$), *and* the target cell's signal strength is satisfying ($\gamma_c > \Theta^c_{lower}$).

If more than one cell satisfies above condition, the one with the highest preference could be chosen.

The following shows that, persistent loop can be caused by improper configurations of preference values. The good news is that, such persistent loops can be eliminated, when the derived preference conditions are avoided:

Proposition 1. *Consider n cells $c_1, c_2, ..., c_n$ that use idle-state handoffs only. A loop $c_1 \rightarrow c_2 \rightarrow ... \rightarrow c_n \rightarrow c_1$ can always happen, if their preference settings satisfy: (1) at least one cell c_i sets $P_{i,i} < P_{i,i+1}$, and (2) every cell c_j sets $P_{j,j} \leq P_{j,j+1}$, and $P_{n,n} \leq P_{n,1}$.* □

Two results follow from Proposition 1. First, some preference settings would always trigger persistent loops with some runtime observations. For stability, they should always be avoided. Second, with consistent preference configuration, the idle-state decision logic can further ensure stability for a device with other proper configurations. This serves as the foundation for stability analysis on other forms of handoff logic. As we will see later, there exists pairwise coordination methods for loop freedom so that enumerating all possible loops is not needed.

We next conduct empirical validation for this case. We have been able to identify 17 instances that can cause loops in AT&T. These configuration conflicts can happen at the same areas (with different runtime observations), and are reported in all locations. Fig. 3.2 summarizes these loops. The smallest loop involves 3 cells, while the largest one includes 7 cells. Among these loops, 16 out of 17 conflicts would occur with femtocell deployed, while the remaining one can occur without femtocell. Our outdoor tests first show that all 2G/3G/4G macrocells have the problematic configurations, and 61 out of 63 locations (96.8%) have all macrocells deployed. This implies that a potential loop would exist if a femtocell were deployed at the spot. We further deploy a femtocell in a campus building, and conduct indoor experiments in all viable

locations. In that floor, 25% of the testing locations satisfy both configuration and signal strength conditions, thus triggering loops. For T-Mobile, we do not observe loops in this category. Based on the causes of the preference conflicts, the loops found in AT&T can be further classified in three categories.

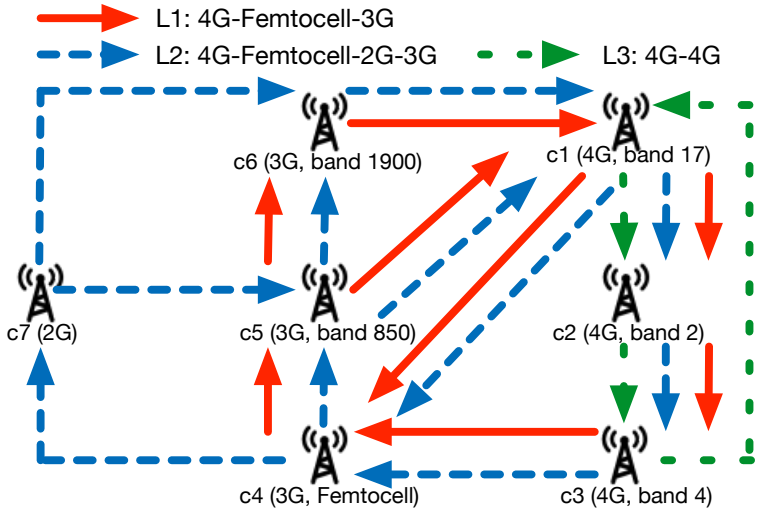


Figure 3.2.: Idle-state persistent handoff loops detected in AT&T.

◦ L1: uncoordinated handoff goals. In this category, 8 variants of loops are reported, all happening between 4G macrocell, femtocell and 3G macrocells. These loops are caused by preference settings for conflicting goals. The 4G macrocells intend to offload user to his/her private femtocells, so it assigns femtocell with the highest preference (over 4G/3G macrocells). The 3G femtocell has equal preference to all 3G/4G cells. But 3G macrocells prefers to move the user to high-speed 4G network, so it assigns 3G femtocell lower preference to 4G. This violates Proposition 1.

Now, we quantify the negative impacts. We observe that the loop frequently occurs. Fig. 3.3 plots a two-hour log of serving cells in the 40-hour test in one 4G-femtocell-3G example. Our test further shows that, more than 90% loops happen every < 200 seconds. With such high frequency, the carrier can neither offload the users to femtocell, nor offer high-speed 4G service. Such frequent handoff loops incur large signaling overhead between the phone and the network. Fig. 3.4(a) shows that, compared with the 4G case, the current loop increases its signaling to the cell and to

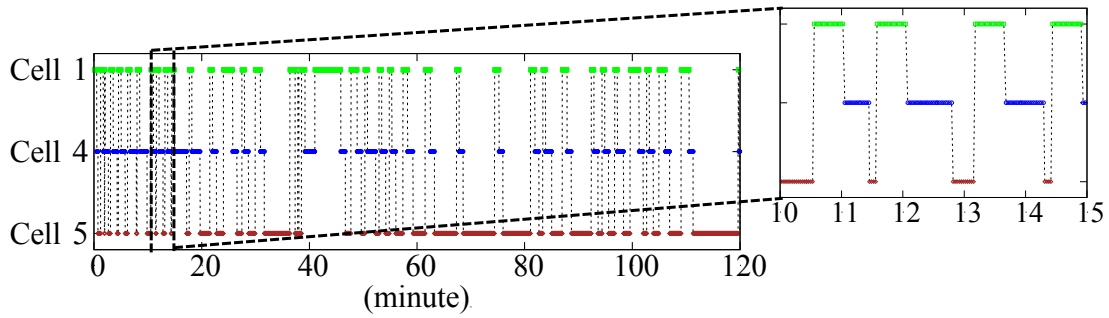
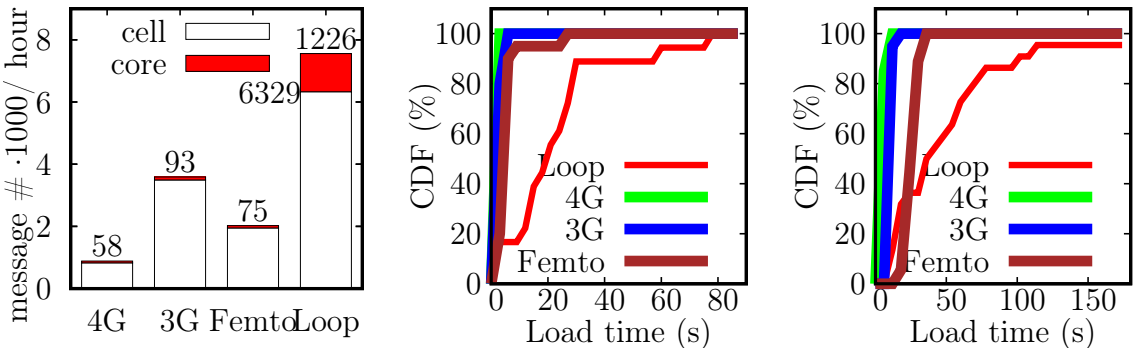


Figure 3.3.: A two-hour log of associated cells at one static phone. The loop is observed despite varying loop cycles.

the core network by 7.6x (6329 : 827) and 23.5x (1226 : 58), respectively. These signaling messages are triggered by location update [17], and include messages related to radio resource allocation, data forwarding path reconfigurations and authentications.



(a) Signaling overhead

(b) Web browsing

(c) Music downloading and playing

Figure 3.4.: Impacts of loops of L1.

These frequent loops can also degrade data performance. To evaluate it, we load a webpage (www.cnn.com) and a music file (about 5 MB) every five minutes using Firefox, and record the loading time with/without loop at the same location in Fig. 3.4(b) and Fig. 3.4(c). The loop slows down the webpage downloading by 11x than 4G (33x fold in the worst case). In the worst case, it takes 1.5 minute to download this webpage, whereas it would take 3 seconds using 4G. We observe similar performance slump in the music case with delay increase by 10x (median) and 14.5x (the worst

case). The music file can be stably downloaded within 12 seconds using 4G, whereas it takes up to 180 seconds in our test. Such performance degrade is mainly caused by repetitive location update. The location update may trigger data path forwarding reconfiguration and re-authentication, during which the incoming/outgoing traffic would be delayed or dropped. Each 3G and femtocell location update typically takes 3 to 6 seconds. In the worst case, the femtocell location update can be further delayed to up to 30s. Besides data performance, such idle-state persistent loops can also cause call drops. In the same experiment setting, we launch voice calls every five minutes using phone's default dialler, and record the failures of voice call setup. We find that the call drop rate is 9.6%. The reason is that, when the call is initiated in presence of loops, the network cannot locate the user to the specific cell, thus unable to establish the voice call session.

◦ L2: device-side preference misconfiguration. Our loop detector further reports 8 variants of loops between 4G macrocells, femtocell, 2G and 3G macrocells. Compared with previous category, when leaving the femtocell, the mobile device handoffs to 2G first, then handoffs to 3G macrocells. This happens when the femtocell's signal strength is weak (< -115 dBm) but still higher than 4G's high-preference handoff threshold (-116 dBm in this scenario). It turns out that, this extra handoff is caused by improper preference configuration on the mobile device. With low signal strength, the device may temporarily lose association to femtocell. Based on the 3GPP standard, the mobile device resumes the service by scanning all the cells, and associates to the first available one [5]. The order of the scanning is based on a preference list in the phone's SIM card. For some phones, the 2G is listed as highest preference, so the phone moves to 2G instead of 3G macrocells. Once associated with 2G, the device would immediately handoff to 3G macrocells, because the 2G cell assigns 3G cells higher preference. This way, the persistent loop continues.

For loops in this category, all negative impacts in 4G-femtocell-3G loops also retain here. Besides, the device may further lose voice services in 2G cells. The reason is

that, some 2G cells cannot support voice and data service concurrently. When the device is transmitting data, the voice service would be disabled.

◦ L3: incremental 4G infrastructure upgrade. The last variant is a 4G-only loop that appears upon some infrastructure upgrading. We observe that AT&T is deploying cells under a new frequency band (c_2 in Fig. 3.2). Before the upgrade, existing 4G cells (c_1 and c_3 in Fig. 3.2) assign equal preferences to each other. AT&T intends to migrate users to the new cells, which offers higher bandwidth. To achieve it, some old cells (c_1) assign higher preference to new cells. However, not all cells' preferences are updated timely: equal preference still exists on some cells (c_2). Such partial update cannot migrate user to the new cells: it violates Proposition 1, and incurs loops between cells. This loop has no direct impact on users, because all cells belong to the same location area. But this incurs larger 4G-femtocell-3G and 4G-femtocell-2G-3G loops, and indirectly amplifies their negative impacts.

2) Inconsistent Threshold. In this category of instability, handoffs may oscillate among cells with uncoordinated thresholds. This may occur even when the preference values are globally consistent. It can be exemplified in Figure 3.5. In the setting, both cells c_1 and c_2 agree that c_1 is preferred, but they apply different rules. Specifically, c_1 uses high-preference rule ($\gamma_2 > c_1.\Theta^{c_2}_{higher}$ (-108 dBm)), and c_2 uses equal-preference rule ($\gamma_1 > \gamma_2 + c_2.\Delta^{c_1}_{qual}$ (3 dBm)). Therefore, the loop exists as long as the received signal strength meets the above condition. Similar misconfigurations can also occur in active-state handoff.

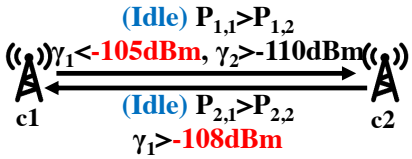


Figure 3.5.: Instability with uncoordinated parameter configuration: threshold inconsistency.

We next derive the instability condition with respect to the radio threshold. We assume the preference settings are globally consistent, *i.e.*, they all see the non-

conflicting ordering on cell preference values. The following result shows the necessary and sufficient threshold configurations for any loop-free handoff:

Proposition 2. *Consider n cells that use the idle-state handoffs only, and configure consistent preferences. The handoff stability is guaranteed iff. the radio thresholds are coordinated as follows: for every two cells c_i and c_j, (1) $c_i.\Theta^{c_j}_{higher} \geq c_j.\Theta^{c_j}_{lower}$ if $P_i < P_j$, (2) $c_j.\Theta^{c_i}_{higher} \geq c_i.\Theta^{c_i}_{lower}$ if $P_i > P_j$, (3) $c_i.\Delta^{j}_{eual} + c_j.\Delta^{i}_{equal} \geq 0$ if $P_i = P_j$.* □

Compared with Proposition 1, Proposition 2 offers a *pairwise* configuration between any two cells. With consistent parameter configurations, two-cell loop avoidance also implies larger loop avoidance. Furthermore, the instability detection is still polynomial even with inconsistent preference. Given n cells at a location, the complexity of finding persistent loops is $O(mn)$, where m is the number of idle-state handoff rules from all cells.

For this category of idle-state threshold misconfigurations, our experiments report no conflicts in AT&T/T-Mobile. The traces show that, both carriers impose stricter conditions over the idle-state thresholds than required (by Proposition 2). The real threshold settings are fully decoupled from the preferences: no matter how preferences are configured, the high-preference threshold Θ_{higher} (AT&T: [-114 dBm, -110 dBm], T-Mobile: [-114 dBm, -111 dBm]) is always higher than the serving threshold Θ^{s}_{lower} (AT&T/II: [-120 dBm,-116 dBm]) between any two cells. This signifies prudent engineering practice, contributing to good operations by both carriers in reality most of the time.

3) Active-idle Misconfiguration. Instability may also be observed when the idle-state handoff is used in some cells but the active-state handoff is adopted in others. For instance, this could occur when the device exchanges highly bursty traffic, e.g., during Web browsing or instant messaging. The device thus stays active with traffic for a while, but then remains idle without traffic. This active/idle state switching is driven by the setup/release of radio resource control connections, and is regulated by

the Radio Resource Control (RRC) protocol [16]. For this scenario, uncoordinated configurations between idle and active state handoff may incur instability.

Figure 3.6 illustrates such an example of two-cell loop. In the setting, c_1's active-state handoff policy evaluates c_1 and c_2's signal strength with two thresholds. But it does not coordinate with c_2's idle-state handoff. So the persistent loops between them can happen when -111 dBm$< \gamma_2 < \gamma_1 - 3$ dBm$<$-105 dBm.

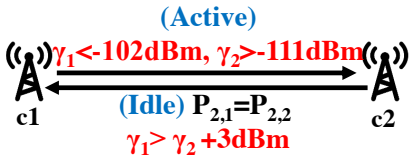


Figure 3.6.: Instability with uncoordinated parameter configuration: active-idle misconfiguration.

We next derive the stability conditions when active-state handoff is involved. Different from idle-state handoff, the active-state handoff logic is customizable by applying various measurement report triggering events. It can thus decide whether to access cells' radio qualities in decision. We first make assumption for all cells' active-state handoffs evaluate radio conditions (the remaining cases will be handled by Proposition 5 of next section). To this end, we assume that the active-state handoff adopts the following radio criteria.

Assumption 1. *For any active-state handoff policy $c_i \rightarrow c_j$, it evaluates cell radio in the decision logic, and takes one of the following forms: (a)* ***absolute comparison****: $\gamma_j > c_i.\Theta_e$ (e.g., A4, B1 event), (b)* ***indirect relative comparison****: $\gamma_i < c_i.\Theta_e 1, \gamma_j > c_i.\Theta_e 2$ (e.g., A5, B2 event), (c)* ***direct relative comparison****: $\gamma_j > \gamma_i + c_i.\Delta_e$ (e.g., A3 event).*

The following configurations ensure stability for active-idle misconfiguration:

Proposition 3. *Consider n cells $c_1, c_2, ..., c_n$ that satisfy Assumption 1. The stability is guaranteed, if all cells' active and idle-state handoffs' radio thresholds are coordinated as follows. For every two cells c_i and c_j, consider c_i's idle-state and c_j's active-state parameters: (1) $c_i.\Theta^s_{lower} \leq c_j.\Theta_e$ if $P_i > P_j$ and c_j uses absolute comparison, (2) $c_i.\Theta^{c_j}_{higher} \geq c_j.\Theta_e 1$ if $P_i < P_j$ and c_j uses indirect relative comparison, (3) $c_i.\Delta_{equal} + c_j.\Delta_e \geq 0$ if $P_i = P_j$ and c_j uses direct relative comparison.* □

Assumption 1 holds in real mobile networks as we have already unveiled the active-state handoff algorithm and policy in previous section 2.3. Given Assumption 1, any active-state handoff can be split into two parts: an equivalent "idle-state handoff" and decisions over other observations. Indeed, when only active-state handoffs are used, Proposition 3 is sufficient but not always necessary. Stability can be ensured through other means (*e.g.*, coordinating other parameters). The merit of Proposition 3 stems from its support for idle-state handoffs: it ensures stability within active-state handoffs only, *and* between idle and active state handoffs.

In this subcategory, our detection tool has found one instance in AT&T (L4). We observe threshold incoordination between 3G macrocell's idle-state handoff and femtocell's active-state handoff for voice. The scenario is similar to that of Figure 3.6. The device oscillates between 3G macrocell (c_1) and femtocell (c_2) when $\gamma_1 < -102dBm, \gamma_2 > -111dBm, \gamma_1 > \gamma_2 + 3dBm$. Interestingly, though no threshold misconfiguration is observed at the idle state, it occurs between idle and active-state handoffs.

This threshold incoordination is not shown without reasons. The femtocell tends to move the active-state user to the macrocell, even when the macrocell's signal strength is weaker than the femtocell's. This is because the femtocell is deployed by users in an unplanned and isolated fashion. Its radio coverage is smaller than that of the macrocell's. So the device has a higher chance to leave the femtocell coverage. To avoid the potential voice call disruption, the femtocell proactively switches the

device to the macrocell earlier than needed. Unfortunately, this configuration violates Proposition 3, and fails to achieve the expected goal. The device may handoff back to the femtocell at the idle state under the same observations.

We test this loop at all viable indoor locations. At each spot, we launch a 24-hour test, and periodically load the webpage with Firefox every five minutes. We count the total number of connections, and how many instances of looped transition between two cells. The active-state handoff condition is satisfied with probability 9.4% (see Table 3.1). However, once satisfied, the loop always occurs in our test. We run the same webpage loading test to assess its impact on user traffic. As shown in Figure 3.9, this loop incurs extra delay about 40-90 seconds.

3.1.2 Instability By Loop-Prone Decision Logic

The instability also occurs when different cells apply conflicting decision logics. No matter how well parameters are fine-tuned, conflicts always exist between decision logics. The fundamental reason is that, the active-state logic is customizable at each cell and the current standards do not mandate the same decision algorithm. We further discover two categories of instability in this class, one is between active-state logic, and the other between active and idle-state engines.

1) Active-active logic conflicts

Figure 3.7 illustrates a two-cell-loop example. The active-state handoff decision logic at each cell adopts a simple rule: both agree to switch to the other if the signal strength at the neighboring cell is good enough (*e.g.*, >-106 dBm). However, if both cells satisfy the signal strength condition, the loop would occur. Note that regardless of the radio threshold to be set, the signal strength that satisfies the loop condition always exists.

We now derive the stability conditions for the active-state decision logic. When all active-state decision engines assess radio quality and satisfy Assumption 1, loop-prone logic can be eliminated as follows:

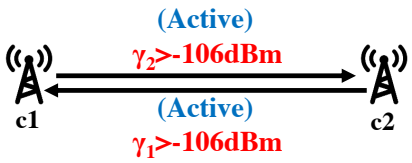


Figure 3.7.: Instability with loop-prone decision logic: active-state logic conflict

Proposition 4. *Consider n cells $c_1, c_2, ..., c_n$ that satisfy Assumption 1. The stability is guaranteed, if all cells' active and idle-state handoffs' radio thresholds are coordinated as follows. For every two cells c_i and c_j, consider c_i's idle-state and c_j's active-state parameters: (1) $c_i.\Theta_e 1 \leq c_j.\Theta_e$ if c_i uses indirect relative comparison, and c_j uses absolute comparison, (2) $c_i.\Theta_e \geq c_j.\Theta 1_e$ if c_i uses absolute comparison, and c_j uses indirect relative comparison, (3) $c_i.\Delta_e + c_j.\Delta_e \geq 0$ if c_i and c_j both use direct relative comparison.* □

Proposition 4 specifies a sufficient condition for loop-free, active handoffs. It is applicable to any handoff decision logic among cells, as long as it assesses radio quality. By coordinating the radio evaluation portion, no observations on the radio quality can lead to a loop. There are two benefits to this approach. First, it does not require coordination of other unknown parameters, thus being flexible and extensible. Second, it is also backward compatible with the idle-state handoff. This helps to prevent persistent loops between active-state handoffs, *and* between active and idle-state handoffs.

Our tool also detects one instance (L5) between two 4G cells in AT&T at one location (Figure 3.7). Both cells try to offload users to each other, when the other's signal strength is higher than certain threshold. However, such load-balancing decisions are not coordinated. A user thus oscillates between these two cells. Fortunately, this loop is not commonly observed. Among all 4G cells we collect, 67% of them use the same policy for the active-state handoff, but its neighboring cells are not observed to use the same rule except at one location. At this location, we conduct 6-hour ping

tests and observe 8 loops (every 45 minutes on average) and the minimum one lasts only 43 seconds.

We further discover that, active-state loop-prone handoff engines are less observed due to prudent engineering practice. In AT&T/T-Mobile, most macrocells' active-state decision logic is more conservative than their idle-state handoffs' counterpart. The radio quality measurements, which may trigger idle-state handoffs, cannot always trigger active-state handoffs in the same cell. The RRC measurement configuration for active handoffs uses stricter thresholds and report criteria than their idle-state handoff counterpart. The reason is that, active-state handoff is usually activated with data traffic. It tends to be more conservative to ensure the seamless data/voice service. Since the idle-state handoffs' thresholds follow the conservative loop-free setting, it is not surprising to see fewer active-state-only handoff loops.

Another related finding is that, the active-state loop-prone handoff logics are more common in 4G. In both AT&T and T-Mobile's 3G macrocells, we didn't observe loop-prone handoff logics in active state. Our experiments show that, both operators' 3G active-state handoff logics are even more conservative than 4G's active-state counterparts: only handoffs between cells under the same frequency (intra-frequency handoff) are used, whose triggering condition is based on the direct relative comparison between serving and neighboring cells' radio qualities (Assumption 1). The reason is that, different from 4G LTE, 3G UMTS supports soft handover between cells under the same frequency. Because of this, intra-frequency handoff offers more seamless data service in mobility, thus more preferred by network operators. This further limits the available candidate cells, thus less prone to loops than 4G.

2) Active-idle logic conflicts

This category of instability occurs when some cells apply idle-state handoffs, while others use active-state handoffs but without assessing radio quality. This can be illustrated by the example of Figure 3.8. In this case, c_1's active-state decision never considers the signal strength used by the c_2's idle-state handoff. Instead, c_1 uses load-balancing, regardless of the radio quality. Consequently, the serving cell oscillates

between c_1 and c_2 once the switch conditions are satisfied in both handoff iterations. It shows that incoordination between decision logic functions is responsible for this unnecessary loop.

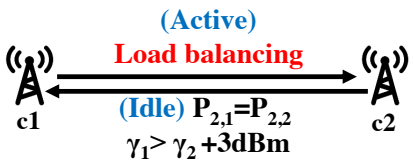


Figure 3.8.: Instability with loop-prone decision logic: active-idle logic conflict

If the active-state handoff does not assess radio quality, we show that idle-active loops would always occur:

Proposition 5. *Two-cell persistent loop between c_i and c_j always exists, if c_i's active-state logic to c_j does not evaluate radio quality, while c_j applies idle-state decision logic to c_i.* □

In practice, the idle-state decision logic is available at all devices. Proposition 5 implies that, if a cell c_i's active-state handoff logic does not assess radio quality, the only possibility to avoid loops is that all its neighboring cells do not allow handoffs to c_i. However, this cell would consequently become isolated from others.

We have found two instances in this category, one (L6) in AT&T and T-Mobile, and the other (L7) in AT&T only. L6 happens between two 3G macrocells, whereas L7 happens between a 3G macrocell and a femtocell. The handoff decision logic is shown in Figure 3.8, with c_1 = 3G and c_2 = 3G/femtocell. This setting violates Proposition 3. Our study shows that, both instances are caused by a design defect in 3G Radio Resource Control (RRC) protocol. The 3G RRC defines an offloading mechanism during connection setup. When a device attempts to set up a radio connection, the cell can reject the device's request, and redirects the device to a nearby cell [18]. However, this redirection cannot take cells' radio quality into consideration. Without a connection, the device cannot report the observations to the cell. If the

current cell's radio quality is better than neighbors', the offloading cannot succeed, because the device would shift back in idle state.

We quantify both loops' impact under similar experiment settings to those for US1-L4. We find that L6 and L7 are not commonly observed even when the loop condition is satisfied. L6 and L7 occur at the probability of 2.15% and 0.49% in our observation at one location (Table 3.1). The serving-cell congestion probability largely determines the loop occurrence. As shown in Figure 3.9, both loops incur delay about 20-53 seconds (median), up to two minutes. We observe that some phones do not always follow the cell's handoff command. Instead, they seek to reconnect to the serving cell. This is why the user device may still not suffer from it even when the loop condition is met.

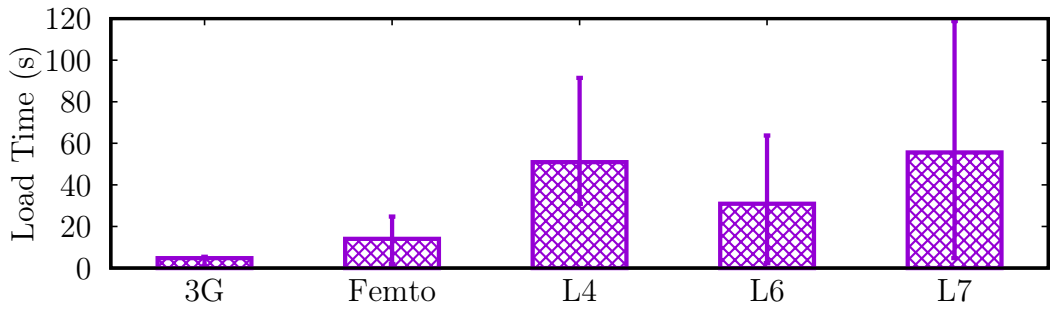


Figure 3.9.: Impacts of loops caused by active-idle misconfiguration logic conflicts.

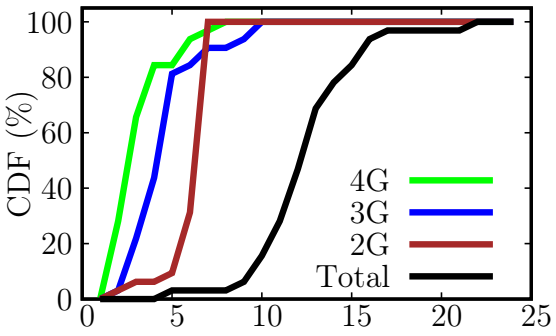
3.1.3 Observations of handoff instability on operational networks

Figure 3.10 summarizes our outdoor and indoor experiment settings. The cell distribution at different outdoor locations confirms that today's deployment is quite dense and hybrid. At most locations, there are about 8-16 cells. On average, there are about 11 cells in AT&T and 10 cells in T-Mobile. The number of unique cells, excluding those observed at multiple locations, are 275 (4G: 120, 3G: 97, 2G: 58) in AT&T and 222 (4G: 92, 3G: 66, 2G: 64) in T-Mobile. It confirms that 4G cells have smaller coverage and denser deployment whereas the 2G coverage is much larger. The

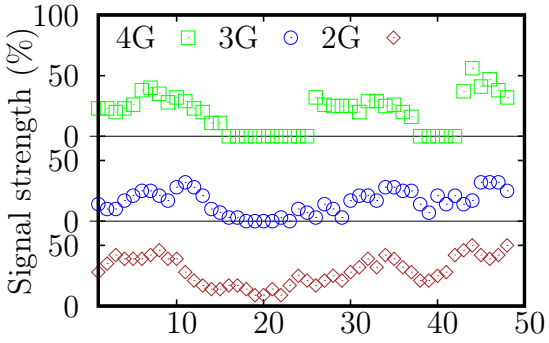
indoor setting has similar cell density as the outdoor one. Figure 3.10(c) plots the median radio signal strength measured at 50 indoor spots in AT&T networks. For 2G/3G/4G comparisons, we use normalized percentages obtained from OpenSignal, a popular network monitor app, where 0% indicates no coverage and 100% indicates the strongest signal strength (Note, 4G uses different signal strength metrics from 3G/2G. The minimal strength observed in 3G/2G is -113 dBm whereas it is around -125 dBm in 4G). It implies that despite higher speed, 4G suffers worse coverage than 3G and 2G in indoor scenarios.

	Avg. cell#/spot		Unique cell#	
	AT&T	**T-Mobile**	**AT&T**	**T-Mobile**
#4G	2.6	2.1	120	92
#3G	3.4	2.4	97	66
#2G	5.4	5.6	58	64
#All	11.4	10.1	275	222

(a) Statistics of outdoor cell deployment



(b) Outdoor cell density in AT&T



(c) Indoor radio signal strength at 50 spots in AT&T

Figure 3.10.: Summary of outdoor and indoor deployment.

With our tool, we have found 21 instances of potential misconfigurations and/or loop-prone logics, which are further classified into 7 categories as elaborated before. For each category, we further run indoor experiments to validate its existence, and estimate its occurrence probability. For each indoor spot, we run a 24-hour test and record the looped handoffs between cells. Table 3.1 lists the occurrence probability

of problematic configurations (left column), and the occurrence probability of loops (right column) observed at one specific location. Other locations have similar results.

Table 3.1. Loop occurrence probability in AT&T.

	#Scenario instances	Occurrence of Misconfigurations or Loop-prone logic	Loop occurrence (parameter+logic +observation)
L1: 4G-Femto-3G	8	96.8%	25.0%
L2: 4G-Femto-2G-3G	8	96.8%	0.49%
L3: 4G-4G	1	2.2%	2.2%
L4: 3G-Femto	1	96.8%	9.4%
L5: 4G-4G	1	1.6%	1.6%
L6: 3G-3G	1	63.4%	2.15%
L7: 3G-Femto	1	96.8%	0.49%

It shows that, instabilities occur in 2G, 3G and 4G networks, with varying occurrence probabilities. From these instances, we show that, loops with both the uncoordinated configurations and loop-prone decision logic indeed exist. Although carriers have applied at least two prudent rules to mitigate loops, configuration conflicts still exist for various reasons, such as diverse handoff goals, the incremental and/or unplanned cell deployment, the device misconfiguration, and the design defects for the connection control mechanism. Loops incur negative impacts upon both the user and the network. We notice a big distinction between both columns, which reflect the gaps of the root causes and the actual impact. The reason is that, the occurrence of actual loops (right column) is also affected by another runtime observations, which may not always be satisfied. It has two implications. First, misconfigurations or loop-prone logics that may trigger persistent loops are not rare in reality. Most settings are problematic once femtocells are deployed. It indicates that the operator's network infrastructure is not fully upgraded to handle small cells which can be deployed by users. Second, although the misconfigurations occur with high probability, the satisfying signal strength that triggers loops do not always occur. For example, only L1 (25%) is relatively common and other loops like L2, L3, L5, L6 and L7 are rarely

observed (below 2%). This is attributed to good practice and satisfactory coverage in radio planning and cell deployment.

3.1.4 Handoff Unreachability

In this section, we uncover 7 instances of handoff unreachability from two categories identified in AT&T and T-mobile networks deployed at Columbus, OH.

1) On convergence split. In the first category, the sequence of handoffs for the given devices starts from a given initial cell and does converge but settles down at a cell other than the desired target because there is no path to it.

◦ *C1: Fail to reach 4G in case of 2G and 4G only.* We find that the 2G cells do not configure a local handoff rule to 4G, but only has a handoff rule to 3G. As a result, in no presence of a 3G cell, the 2G cell cannot hand over the device to the 4G cell. Figure 3.11 shows this instance validated in the real trace. The device initially stays in an area with only 2G coverage, but later moves into a new spot with both 2G and 4G coverage. However, the device does not move to 4G as expected. Despite strong radio coverage from 4G, the device gets stuck in 2G.

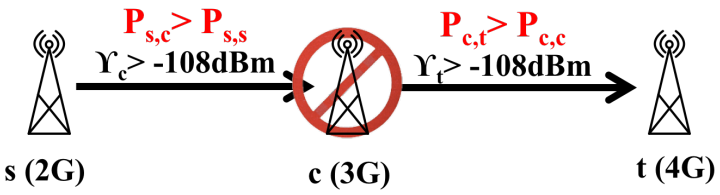


Figure 3.11.: C1: 3G relay cell inaccessible

Clearly, the root cause is that the 2G cell lacks proper handoff configurations for the 4G cell. This issue arises not without rationale. In practice, 2G has been phasing out and the operator mainly focuses on deploying new 3G or 4G cells. When new cells (here, 4G cells) are in operation, the carriers do not update the configurations at old 2G cells. Instead, they count on the presence of a 3G cell to move the handoff from 2G to 3G and finally to 4G. However, in reality, the intermediate 3G cell can be

inaccessible for various reasons. For example, 3G may fail to cover some spots, the device's received signal strength from the 3G cell is too weak, or the device has radio compatibility issue to access the 3G cell (*e.g.*, , it only supports certain 3G technology such as TD-SCDMA, but not other 3G technologies).

◦ *C2: No 4G in case of 3G femtocell and 4G only.* We observe another similar instance caused by missing configurations but among femtocell, 3G and 4G cells. It holds the same problem: the 3G femtocell has no configuration rule to the 4G cell, but only has the rule to a 3G public cell. When a 3G cell is not accessible (here, 3G is extremely weak), the migration to 4G (via the intermediate 3G cell) is infeasible. We observe this problem when a phone leaves the femtocell coverage and moves to an area with 4G only (no/weak 3G). In Figure 3.12, the device becomes out of service once moving outside the 3G femtocell coverage, despite the existence of a 4G cell. The device is thus stuck at out-of-service in this case.

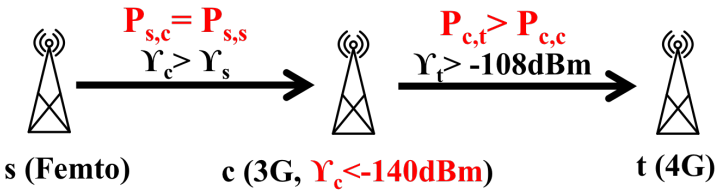


Figure 3.12.: C2: Weak/no 3G relay cell

In addition to the lessons from C1, one more thing worth noting is that it may be commonly observed when the femtocell deployment follows the common guideline that suggests the femtocell to be deployed with weak macrocell coverage. Deploying a femtocell in the area with no or weak 3G coverage may not be rare. However, it may induce problematic (though temporary) service interruption when it moves toward 4G.

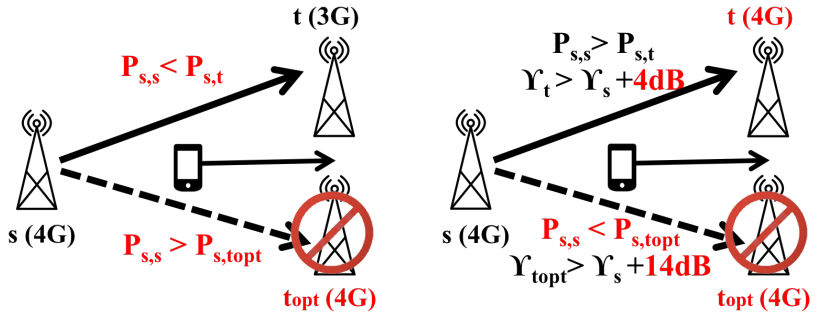
C1 and C2 both reveal a practical challenge that mobile networks are facing. Not all the cells have a direct path to any other cells and the reachability from s to e has to depend on the intermediate cell (here, 3G). However, the existence of

intermediate cells is not guaranteed. The unpleasant consequence is that the big investment on advanced technology (here, 4G) goes futile due to 2G's configuration glitch. The blame can be that 2G or 3G femtocells lack proper configurations to 4G. However, it is not without rationale. There was no 4G when 2G was deployed and the 2G infrastructure is likely not updated to date due to heavy cost (possibly retire soon). Femtocells may be configured so under the premise that 3G has been largely deployed. With versatile access technologies and rich options (different frequency bands and small cells), it is not guaranteed that each cell has a direct path to all possible cells. Mobile networks should be painstaking on their decision procedure or rigorous on their infrastructure deployment or both.

2) On premature convergence. In the second category, the actual handoff process starts from any initial cell while given path to desired target cell, however it is either unable to reach the desired target or stops early before it reaches the target.

◦ *C3: Select 3G, not 4G due to improper preference configurations.* We notice that not all the cells follow the common preference settings: 4G $\geq$ 3G $\geq$ 2G. Figure 3.13(a) shows an example validated in reality. The device is initially served by a 4G cell, but later eventually switches to 3G after moving into a new spot covered by both 3G and 4G. The reason is that at the original serving 4G cell s, the preference for 3G is even higher than the one for 4G. In particular, we find that the preference for all 3G cells using UARFCN (UTRA Absolute Radio Frequency Channel Number) 9663 is set as 5, which is higher than all 4G LTE cells (2-4) and other 3G cells (1) at other frequency bands. The higher preference values, the more preferable. UARFCN is a unique number given to each radio channel within the frequency bands used by the UMTS (3G technology), UARFCN 9663 is one band over 1900 PCS spectrum. We observe such preference settings at all the relating cells in AT&T. As a result, the handoff settles down at 3G instead of 4G where both have good radio quality.

The carriers may argue that the handoff is not designed to select the best cell in this case or 4G does not necessarily work better than 3G in all the cases. There are other non-performance factors, for example, the carriers intend to promote the



(a) C3: improper preferences (b) C4: improper thresholds

Figure 3.13.: Handoff unreachability instances C3-C4 caused by misconfigurations.

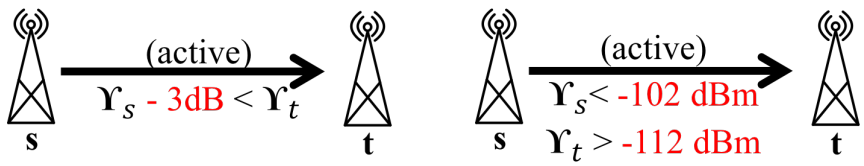
use of 3G cells over a certain channel because of low investment and maintenance cost. However, we concern whether such preference configurations are made for a short-term interest and will be corrected later to maximize the interest of the carrier and/or the user. At least, we should stay cautious when we set the preference against common sense.

◦ *C4: No desirable handoff due to inconsistent preference-threshold settings.* Following C3, we unveil a counter-intuitive handoff case even where the threshold settings contradict with the preference settings. We identify inconsistent handoff decisions under the same condition and thus infer that at least one handoff decision is not desirable. In particular, we find that a more preferable handoff is not performed while the less preferable handoff is performed. Figure 3.13(b) shows an example validated in reality. There are three 4G LTE cells, s, t and t_{opt} with different preference values as 3, 2, 4, respectively. However, the threshold for a higher preference cell $s.\Theta^{t_{opt}}_{higher} = \gamma_s + 14$ dB is larger than the one for a lower preference cell $s.\Theta^{t}_{lower} = \gamma_s + 4$ dB. As a result, when the device moves away from s, the handoff to a lower-priority cell (here, t) is performed when t is 4 dB stronger than s. In contrast, the handoff to a higher-priority cell (here, t_{opt}) is not performed unless t_{opt} is 14 dB stronger than s. Clearly, the handoff to a more preferable cell less likely occurs, which contradicts with the cell preference setting. Moreover, we can identify more similar instances

by checking whether the handoff thresholds satisfy $s.\Theta^c_{higher} > s.\Theta^c_{lower}$. Although it does not incur any service interruption, we still believe such settings conflict with our desire to obtain "always well connected" and hurt the user experience.

We gauge that the root cause of such undesired handoff lies in imprudent handoff configurations. Intuitively, the handoff toward a higher priority cell is more expected than the one toward a lower priority one, under the same radio conditions. However, with $s.\Theta^c_{higher} > s.\Theta^c_{lower}$, the former handoff is harder to get triggered. Similarly, the handoff out of a lower priority serving cell are more expected but actually harder to make. In fact, C3 and C4 together reveal unreasonable handoff configurations, at least from the user perspective. It indicates operational slips in practice and calls for the attention and actions on the operator side.

◦ *C5: Undesired handoff without better signal quality.* We identify another opposite issue in active-state handoff. The handoff is triggered without obvious benefits on radio quality improvement. Figure 3.14(a) and 3.14(b) present two configurations observed in reality. In Figure 3.14(a), we find that the measurement report is triggered when signal quality of neighbor cell becomes offset better than serving cell. However here, the real value of the "offset" is negative (-3 dB) which could lead to an undesired handoff. In other word, the handoff will be performed from a better cell s to a worse cell t. In Figure 3.14(b), intra-RAT handoff is performed when the serving cell is worse than threshold1 (here, -102 dBm) and the candidate cell is better than threshold2 (here, -112 dBm). Clearly, when the signal strength of cell t is within [-112, -102] dBm, the handoff may move to a target cell with poorer signal quality.



(a) C5: negative threshold (b) C5: inconsistent thresholds

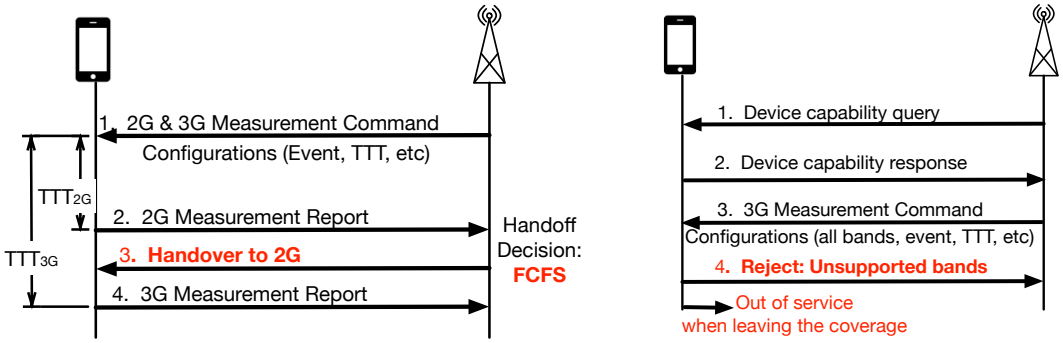
Figure 3.14.: Handoff unreachability instances C5 caused by misconfigurations.

This problem is a special (opposite) case of premature convergence. It reveals that the handoff may move to a worse choice than the original serving one. The rational behind such configurations may expect to initiate the measurement and report early so that the serving cell is aware of qualified neighbor cells around and avoid a too-late handoff. It helps to ensure "Always Connected" for ongoing data/voice service. However, this may somehow hurt the quality of handoff. It actually reveals a trade-off between "always connected" and "always well connected".

We next present another two instances caused by non-decision factors.

◦ *C6: 3G blocked by 2G with missing 3G measurement.* Figure 3.15(a) shows such a real-world scenario. The user device is at the active state and about to leave its 4G serving cell. The new location has both 2G and 3G cells, but these cells cannot reach each other. To initiate the handoff decision, the serving cell asks the device to measure and report signal strengths from both 2G and 3G cells. For each candidate cell, the 4G serving cell configures the device with (1) the report criteria, (2) the measurement duration `TTT` (TimeToTrigger) to ensure stable measurements. The problem arises when both 2G and 3G signal strengths are good. If the serving cell uses the first-come-first-serve (FCFS) strategy and the device reports 2G first, the serving cell may immediately hand over the device to 2G, without waiting the device to finish its 3G measurement. Given the good radio quality from the 2G cell, handoff to 2G is activated. A premature convergence to 2G occurs, thus ruling out the desired handoff to the 3G cell. In this example, the 3G cell is more preferable which is ideally reachable if the decision is made based on the complete measurement information. The handoff decision to 3G is never made by the early measurement report from 2G cells and the FCFS response to a good measurement report. However, once the undesired cell (here, 2G) is chosen first, the chance to selecting another desired one is blocked.

The root cause lies in improper coordination between the network and the device. The network acts as the master to control the device (the slave) to conduct measurements for the handoff. However, its FCFS response to the device reports does



(a) C6: blocked decision with missing measurement (b) C7: problematic coordination between device and network

Figure 3.15.: Two premature convergence instances caused by non-decision factors .

not work well with the device which has freedom to conduct its measurements of candidate cells in any order. In this case, both the user and the network have their valid reasons. The serving cell wants to expedite the handoff decision to minimize the handover latency, whereas the device decides its own order for measurements since it does not know the decision logic at the serving cell. However, it turns out that both get penalized. In fact, FCFS at the serving cell intends to reduce/dismiss any interruption to ensure "always connected". This instance also unveils another design trade-off between "always connected" and "always well connected".

- *C7: Out of service due to problematic device-network coordination.* We also uncover that premature convergence can be caused by problematic coordination between the network and the device. Figure 3.15(b) shows a real scenario. The 3G cell supports multiple frequency bands, but the device supports only one of them (a common case since many phone models cannot support all). Without taking into account the device's capability, the serving cell requests the user device to monitor all 3G frequency bands. Upon this request, the device rejects this command, even though it can still access some bands. No measurements would be conducted by the device thereafter. The serving cell could not initiate any handoff without measurement reports. If the user also leaves the current serving cell, the device loses its network access.

This instance reveals the importance of coordination between the network and the device. Both are developed and configured separately and it is hard, if not impossible, to verify that they are fully compatible. However, additional mechanism should be provided to identify the potential conflicts (*e.g.*, the device should be able to detect and report this issue). To ensure "always well connected", both devices and network should try their best efforts to maintain seamless connectivity and select the best as they can.

3.1.5 Potential Fix

We now discuss how to fix the improper configuration and their resulting handoff instability and unreachability issues. Given that persistent loops and unoptimized serving cell selection hurt both the user performance and the network's operation, we envision that both carriers and users have incentives to address the identified configuration issues. We next propose solutions to both sides.

Network side coordination and examination We recommend two fixes to the network. First, the network deploys a centralized controller, which collects and coordinates the handoff decision functions and configurations among cells. This is a long-term solution which is aligned with 5G trends. Second, the network applies our proposed rules to exam and correct common local misconfigurations identified in our work.

Client side approach The user device can act as an implicit controller for three functions. First, it runs self checking like what our detection tool does. It thus verifies whether handoff configuration for each cell satisfies the desired stability and reachability. If not, the device may elect to not honor such configurations from the serving cell, thus avoiding undesired handoff execution. Second, it can record the historical handoff execution in the recent past. When handoff instability is detected in history, it blocks specific handoff behavior to avoid such handoff loops. When the serving cell is not the desired one, it can probe more candidate cell on its own actively

to jump out of unwise cell selection. Third, the device can leverage crowd-sourcing to retrieve problematic areas and suggested serving cells reported by others. Our following work takes the third approach to intervene handoff decision at device side and improve network data performance significantly.

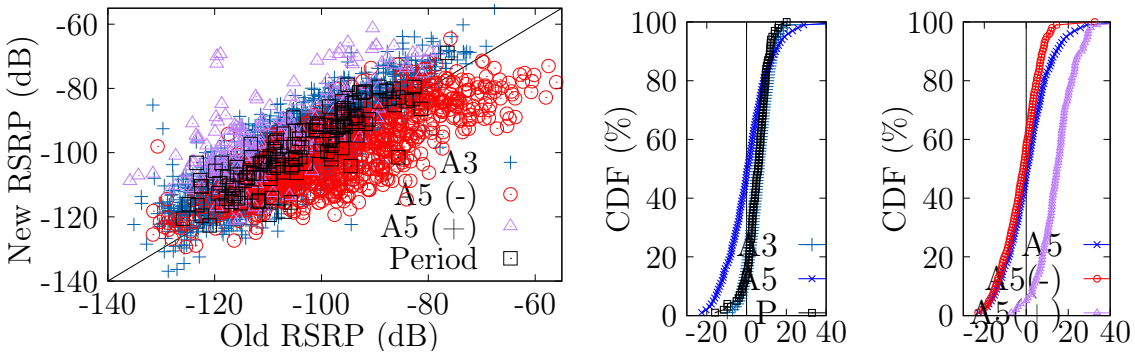
3.2 Impact of Handoff Configurations on Data Performance When Moving

In previous sections, we have seen that mobile carriers deploy handoff configurations with various carrier-specific values in real-world. Such observations raise an interesting question: which configuration (policy) runs better? In this section, we treat data performance as the goal of interests and exam the impact of diverse handoff configurations on data performance while a handoff is performing. We focus on active-state handoff only as it is more directly related to data performance of users in moving state. We study two issues here. Considering the nature of existing handoff execution criteria is radio based, we first study how do the handoff configurations affect radio signal quality before and after an active-state handoff. And then, we examine their relation to data performance and study how do they affect data performance on the go.

We conduct measurement in three US cities (Chicago, IL; Indianapolis, IN; Lafayette, IN) at limited scale and highways in between to asses data performance during active-state handoffs. We utilize the MI-Lab framework to conduct the performance assessment by running designated data services while driving locally (¡50 km/h) and on highways (90–120 km/h). Every run tests with continuous speediest as data service. We run four-week experiments intermediately in the period of Jan-April, 2018, over 16 rooted Android phones (three models: Pixel 2/XL and Nexus 6P) with two top US carriers: AT&T and T-Mobile. We use *tcpdump* to log data packets and *MobileInsight* [7] to collect cellular signaling messages that convey handoff configuration parameters. We study 4G $\rightarrow$ 4G active-state handoffs only and collect 14,510 instances (around 8,000 km in total).

3.2.1 Handoff to cells with better radio signal is not always true

We find that not all handoffs go to a cell with stronger radio signals, and this choice depends on handoff configurations. Fig. 3.16(a) shows RSRPs before and after handoffs under three decisive reporting events for active-state handoffs in AT&T (similar for other carriers). For comparisons, we also plot the cumulative distribution functions (CDFs) for the RSRP changes ($\delta_{RSRP} = RSRP_{new} - RSRP_{old}$) in Fig. 3.16(b). We see that, for A5, only 52% of handoffs get better in terms of RSRP (62% for RSRQ). In contrast, A3 and periodic reporting largely ensure a better radio signal quality: 87% of handoffs have $\Delta_{A3} > 0$ and the ratio goes up to 94% given that 3 dB measurement dynamics is common. This is because A5 reports two independent conditions: the serving cell is weaker than one threshold ($\Theta_{A5,S}$) and the candidate cell is stronger than another one ($\Theta_{A5,C}$). Given two parameter configurations, it is not ensured that the new cell is stronger.



(a) RSRP before and after handoffs (b) δ_{RSRP} (c) A5 only

Figure 3.16.: RSRP changes in active handoffs in AT&T.

For RSRQ cases, We commonly observe that $\Theta_{A5,S,rsrq} > \Theta_{A5,C,rsrq}$ (*e.g.*, , -11.5 dB vs. -14 dB). In RSRP cases, the dominant setting is $\Theta_{A5,S,rsrp}$ = -44 dB (no requirement) and $\Theta_{A5,C,rsrp}$ = -114 dB. This is exactly the special A5 events we identified previously which do not take into account the serving cell's radio signal strength using RSRP in AT&T. They are responsible for the cells after handoffs with weaker radio signal coverage. While this finding differs from expectations, it matches

the consequences of such configurations well. We further divide the A5 configurations into positive (+) ($\Theta_{A5,C,rsrq} > \Theta_{A5,S,rsrq}$) and negative (-) cases. Fig. 3.16(c) shows that weaker radio signal is caused by negative configurations. Such result of event A5 further confirms our finding: Radio quality during handoffs changes as configured, but radio signal quality is not always enhanced after handoffs.

3.2.2 Data performance and "questionable" configurations

We next show that data performance during handoffs are also affected by such configurations.

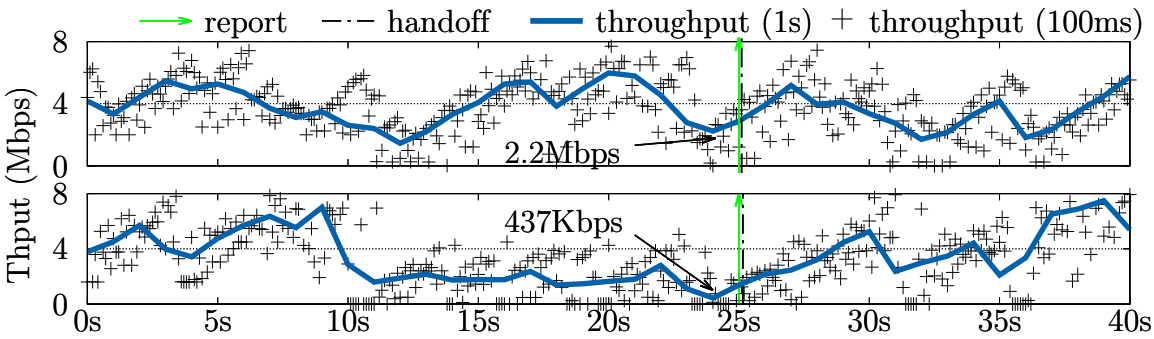
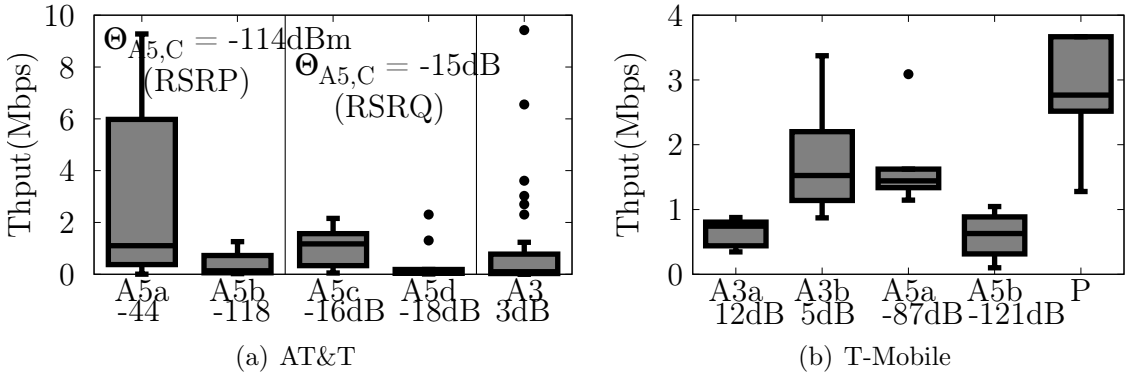


Figure 3.17.: Throughput of two handoff examples using distinct event A3 offsets Δ_{A3}: 5 dB and 12 dB.

We first present two handoff examples, which are both triggered by A3 but with different offset values: $\Delta_{A3} = 5$ dB (top) and 12 dB (bottom). We align both routes with the Measurement Report message (t = 25 s) and handoffs are performed right away after the reporting (within 180 ms). Fig. 3.17 shows the average throughput in two time bins (1 s and 100 ms) while we run a continuous speed test in T-Mobile. We see that data throughput decreases down to 2.2 Mbps (top) and 437 Kbps (bottom) before handoffs. Performance is much worse in the bottom case because Δ_{A3} is 12 dB, much higher than 5 dB (top), which invokes the handoff very late after data throughput has already severely fallen down. The handoff occurs only when one

candidate cell must be much stronger than the serving one. The minimum throughput before handoffs declines by 80.1% (5× gap).

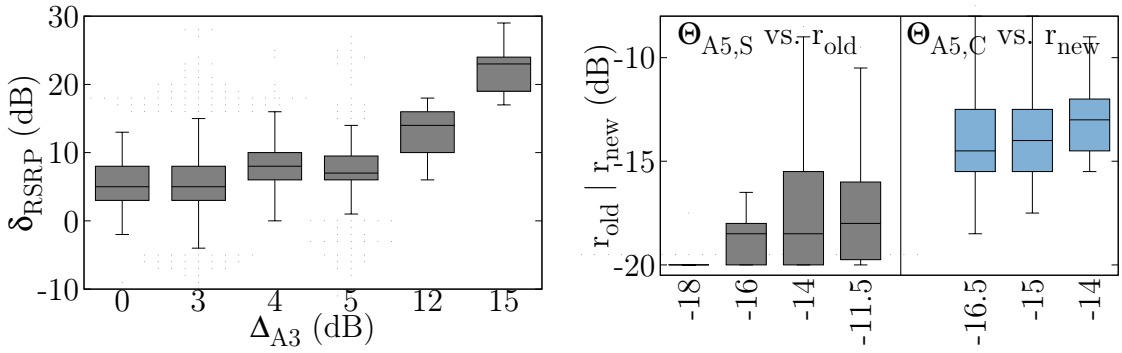


(a) AT&T (b) T-Mobile

Figure 3.18.: Impacts of A3 and A5 reporting event configurations on data performance.

We use the minimum throughput before handoffs (reporting) to assess performance impacts of reporting configurations. Fig. 3.18 compares performance under representative configurations in AT&T and T-Mobile. It shows that data performance impacts match with the anticipated consequences of such configurations. In T-Mobile, A3a (12 dB) and A5b (-121 dB) tend to defer or prevent handoffs to new cells, compared with A3b (5 dB) and A5a (-87 dB). A5 considers the serving cell's threshold $\Theta_{A5,S}$ (RSRP: -87 dB and -121 dB) only. Consequently, they result in lower throughput and worse handoff quality. This is consistent with observations in AT&T. A5a ($\Theta_{S,RSRP}$: -44 dB) outperforms A5b (-118 dB) given the same $\Theta_{C,RSRP}$ (-114 dB). It is similar in the A5c/A5d cases which use RSRQ, but the gap is much smaller as two thresholds are quite close. Such performance impacts can be somehow derived from their impacts on radio signal quality. Fig. 3.19 shows the box-plots of three pairwise relations: Δ_{A3} versus δ_{RSRP}, $\Theta_{A5,S}$ versus r_{old} (RSRQ before handoff) and $\Theta_{A5,C}$ versus r_{new} (RSRQ after handoff). We choose these three pairs by the purposes of those parameters. We can see that handoffs are performed as configured. In A3, radio quality improves more with a larger offset Δ_{A3}. In A5, we consider RSRQ only as the impact of RSRP (-44 dB) has been explained before. A smaller

$\Theta_{A5,S}$ implies that handoffs should be performed when the old cell is weaker, which reduces the likelihood of a handoff and results in poor data performance. A smaller $\Theta_{A5,C}$ implies that handoffs should be performed when the new cell is stronger, which reduces the likelihood of a handoff but usually implies better data performance once the handoff occurs.



(a) A3 (Δ_{A3}, RSRP) (b) A5 ($\Theta_{A5,S}$, $\Theta_{A5,C}$, RSRQ)

Figure 3.19.: Radio signal impacts of configurations in A3 and A5.

We discover that radio signal quality enhancement may not be the key to better handoff performance. "Better" configurations should invoke the handoffs in time, well before the performance degrades or is about to degrade.

Recall, when $\Theta_{A5,S}$= -44 dB (RSRP), A5 performs the best but stronger radio signal quality is not guaranteed. This choice relaxes the requirement on the serving cell and creates a larger chance to obtain measurement reports earlier, thus making early handoff possible. Comparatively, other strict configurations like $\Theta_{A5,S}$= -118d B (RSRP) indeed ensures a handoff only when the current cell is really poor (depending on the value of $\Theta_{A5,S}$) and avoids some handoffs (*e.g.*, where the serving cell is stronger than -118 dB while a neighboring one is even better (*e.g.*, >-100 dB)).

This illustrates two different policies for handoff management. The former is more performance driven while the latter also takes into account handoff overhead and seeks to reduces handoff frequency. It is hard to argue which one is better. As the cellular network infrastructure has been evolving with long-lasting deployment and

upgrades (radio signal coverage likely enhanced and overhead for frequent handoffs not likely a big concern), it may be the time for us to update handoff policies and their configurations. We need a more rigorous way to configure parameters as it seems the existing configuration values is more likely to be set based on experience in reality.

3.3 Missed Data Performance In 4.5G LTE Advanced

In recent years, mobile carriers have been heavily upgrading their network infrastructures to boost the raw system capabilities (*e.g.*, carrier aggregation in 4.5G, and new radio in 5G for higher data rate). In this section, we argue that, it is equally important to better exploit the available and deployed capabilities to reach their full potentials for mobile users. We thus study an important, yet largely overlooked performance problem: What is the performance gap between what a mobile device actually gets in reality and what it could have possibly gotten at best, given the same operational network? If this gap is small, the user gets what (s)he expects and deserves with the upgraded infrastructure. Specifically, when radio access technology is elevated from 4G LTE to 4.5G, the device should enjoy the enhanced data rate (say, up to 10s of Mbps), as long as the user device connects to the appropriate serving cell(s).

Unfortunately, our measurements show that the gap is not small, at least not always small in operational mobile networks. We find that, infrastructure upgrade might not always lead to enhanced performance for end users. The enhanced raw system capabilities are not fully exploited, and remain to be underutilized. The fundamental problem lies in the performance gap between the actual performance a mobile device receives and the achievable performance it could have obtained at best. This gap is deemed as the missed performance, since the device fails to achieve the higher, feasible performance given the same infrastructure. We notice that the operators never promise the best performance to mobile users, and they may even intend to offer sub-optimal performance. For example, the operator selects a cell

that has lighter load but afford poorer performance to serve the user device for the sake of load balancing. In this study, we take the user-centric perspective and aim to explore the upper bound of performance achievable on the device side. We delve into its cause analysis, and confirm that the better performance, as indicated by the gap, can indeed be reached for the mobile device, with simple, device-based operations without any infrastructure change.

First, we start with a real-world instance to expose that a significant portion of performance gain is indeed missed. Second, we conduct a city-scale measurement study to quantify the missed performance in the wild. Third, we further delve into the root cause analysis for such under-utilization.

3.3.1 An illustrative instance showing the missed data performance

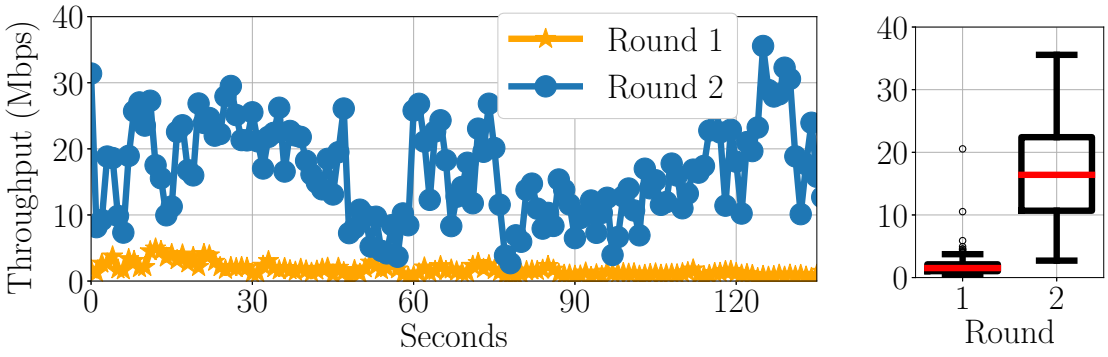


Figure 3.20.: An illustrative example with > 10-fold, missed throughput.

Fig. 3.20 shows an illustrative example, where the phone receives 1.5 Mbps from AT&T on average, whereas 16.9 Mbps is feasible at the same spot. We run experiments in two rounds, where the phone downloads the same large file from our server. Round 1 uses the default network operations. It starts once the user walks to the spot along a pre-determined route. Round 2 runs immediately after Round 1 with resetting its mobile network (*i.e.*, turning on the airplane mode and then turning off). Fig. 3.20 plots the downlink speed in one experimental run, with at least 10-fold ((16.9-1.5)/1.5) throughput reduction. Such large speed gaps are repeatedly observed

at this spot in 20+ experimental runs, over different days and across different hours of the day. Two implications immediately follow. First, the device indeed misses a large portion of feasible downlink speed. The missed throughput can be even bigger because the "best" one is no smaller than that observed in Round 2. Second, the large gap is persistently observed. It is unlikely to be incurred by transient factors (*e.g.*, varying radio signal quality or network congestion).

By examining the serving cells used in these two rounds, we identify that different serving cell sets are being selected. In round 1, we find the following cells are used to serve the device: cell 253 on frequency layer 5145, cell 52 on frequency layer 2000 and cell 55 on frequency layer 2175. While in round 2, three completely different cells are used: cell 102 on frequency layer 2425, cell 426 on frequency layer 2175 and cell 166 on frequency layer 850. Here, one core technology, carrier aggregation (CA), is used to advance from LTE (4G) to LTE-Advanced (4.5G).

Carrier Aggregation.

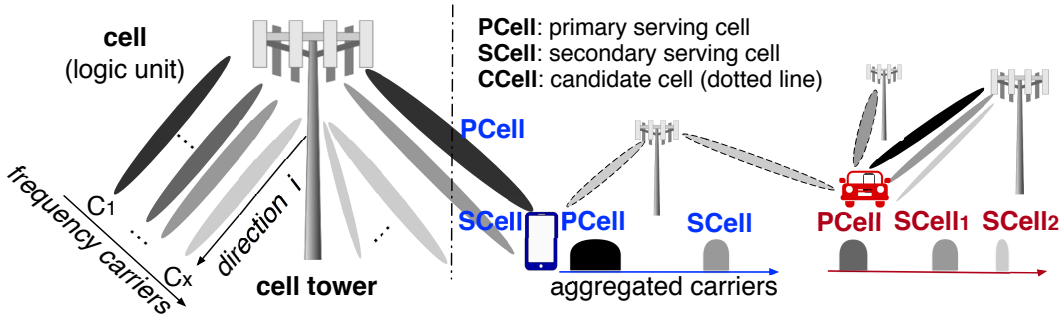


Figure 3.21.: Radio access with carrier aggregation.

Intuitively, larger bandwidth (wider spectrum) promises higher data speed. In order to boost network speed, carrier aggregation technology is proposed as a key feature in 4.5G LTE Advanced. As showed in Fig. 3.21, CA allows more than one serving cells to offer simultaneous radio access, thus increasing bandwidth on an aggregated carrier (over multiple individual carriers with each being used by a cell). The set of serving cells consists of a primary serving cell (PCell/PSC) operated on primary component carrier (PCC) and several secondary serving cells (SCells/SSCs)

operated on secondary component carriers (SCCs). PCell is mandatory and needed for data transmission and connection management. SCells are optional and used for data transmission only. For mobility support, only the selection of PCell is managed by handoff procedure directly, the selection of SCell(s) is determined by PCell afterwards. Since LTE release 10, CA supports up to 100 MHz by aggregating maximum five 20MHz carriers.

We gauge that the selection of serving cell(s) plays an important role in determining the data performance and may be the reason behind such big performance gaps. We validate and refine our findings later.

3.3.2 Measurement methodology and dataset

In this work, we conduct a city-scale measurement in West Lafayette, IN (a 6 km $\times$ 6.7 km area marked in Fig. 3.22). We focus on data performance in terms of downlink throughput while downloading a big file (500 MB) from our lab server. We notice that there are many other performance metrics like uplink throughput, latency or application-specific metrics (*e.g.*, jitters and stalls in video streaming), and leave them as our future work. We use eight Google Pixel phones (Pixel 2, 2 XL and 3) for four US carriers (AT&T, Verizon, T-Mobile and Sprint). No device-specific results are observed and the data from different devices are combined. We use *tcpdump* to collect packets captured on the phones. To analyze and understand the root cause of missed performance, we use *MobileInsight* [7] to collect signaling messages exchanged between the phone and the network (including RRC messages and physical layer messages on radio signal/quality measurement and reporting). These messages are used to monitor the set of serving cells (PCell, SCells) and other neighboring cells being measured and learn how handoff is performed.

We first run experiments at certain locations to get a glimpse of missed performance in reality and then run driving tests for a larger scale measurement. In static tests, we randomly choose 14 locations in three representative zones: campus (aka,

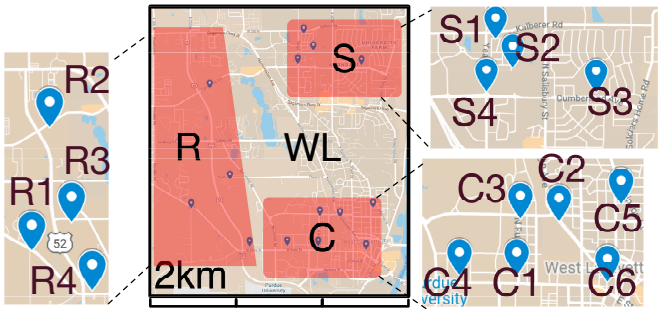


Figure 3.22.: Map of test areas.

urban), suburban (residence) and rural, as marked in Fig. 3.22. At each test location, we first run static tests under default network operations at different hours of the day and at different days. In addition to tests under default network operations, we exploit extra actions allowed at the device side (*e.g.*, turning on/off airplane modes, configuring the preferred network or frequency band) to disturb the default operations. We find that this makes it possible and fast for us to observe data performance distinct from (sometimes significantly outperforming) what the device gets by default. We retrieve performance samples (here, downlink throughput per second) and then cluster them according to their serving cell sets. We observe significant performance missed in our static tests. We observe that more than one serving cell sets are used to serve the device at the same location and the perceived performance varies with distinct serving cell sets. We cluster data performance by its serving cell set and later show that missed performance is associated with its serving cell set.

Afterwards, we run driving tests to check missed performance at a larger scale, across the whole city. We drive along **every** road in West Lafayette, and divide these roads into small grids (each approximately 55 m x 42 m) and retrieve missed performance per grid. We measure the performance under current network operations, but we find performance at each location varies in every run, partly due to varying radio signaling strength, partly due to the distinct set of serving cells which are decided by different network operations in each run. We drive along a variety of routes to cover each road (grid) multiple times (main roads: $\geq$ 20, almost all local

roads: ≥ 5). We admit that the missed performance might be under-estimated given a limited number of samples. We are unable to measure the best and worst performance at each grid in our limited measurement. However, even given such a conservative measurement, the significant performance miss is indeed observed in our study, and quite frequently. Finally, we figure out that the set of serving cells plays an essential role in missed performance. We further run more driving tests, not with heavy file downloading, but with mice traffic (ping Google every 50 ms) in the background for the cause analysis. We conducted this measurement study from July to Oct. 2019, primarily in Aug 25 – Oct 15. Our dataset covers experiments for about 740 hours (in both static and driving tests) and over 8,756 Km (in driving tests only) (Tab. 3.2). It is available at our website [19].

Table 3.2. Dataset statistics.

	AT&T	Verizon	T-Mobile	Sprint	**Total**
Time (hr)	166.3	290.4	186.9	96.3	739.9
Distance (Km)	2,182	2,734	2,378	1,461	8,756

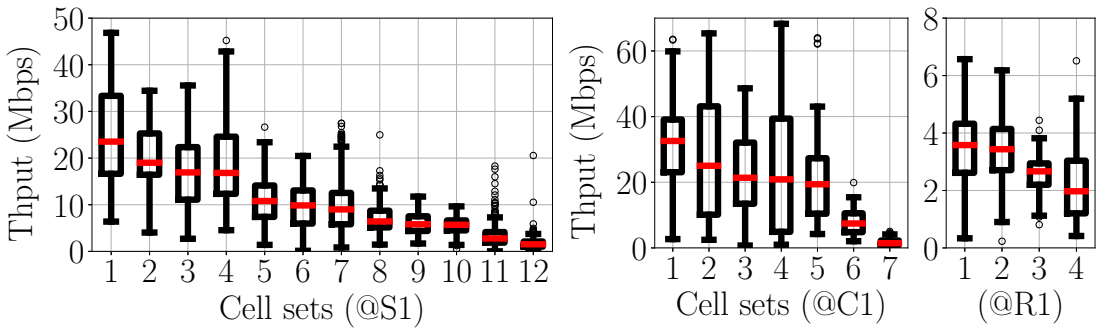
3.3.3 How vastly the missed performance is observed

We first use static tests in AT&T to examine how large the missed performance can be at different locations. Similar results are observed for other three carriers unless specified. We find that missed performance is not rare. Significant miss in performance is observed at many test places.

1) How large can the missed performance be?

We attempt to answer this question first through the tests at three selected locations in three typical zones: S1 (suburban), C1 (campus) and R1 (rural). We record downlink throughput (per second) and group these samples according to their serving cell sets in the boxplots of Fig. 3.23. The serving cell sets are given in the right table. Each cell is marked by its short ID and frequency channel number. The mapping from a channel number to its frequency spectrum band is specified in [12] and can be

found online (*e.g.*, via [13]). We exclude those rarely observed sets (with the sample size < 60). We see that throughput fluctuates (in the range of these boxplots). This matches with our common expectation because data is collected at many runs at different days or at different hours of the day, affected by varying radio signal quality and network conditions.



@S1	PCell (freq)	$SCell_1$ (freq)	$SCell_2$ (freq)
1	244 (2000)	102 (2425)	166 (850)
2	166 (850)	102 (2425)	244 (2000)
3	102 (2425)	426 (2175)	166 (850)
4	102 (2425)	244 (2000)	166 (850)
5	45 (9820)	426 (2175)	102 (2425)
6	447 (5145)	426 (2175)	244 (2000)
7	166 (850)	447 (5145)	45 (9820)
8	426 (2175)	102 (2425)	166 (850)
9	197 (850)	104 (2425)	396 (2175)
10	426 (66661)	-	-
11	166 (850)	447 (5145)	-
12	253 (5145)	52 (2000)	55 (2175)

@C1	PCell (freq)	$SCell_1$ (freq)	$SCell_2$ (freq)
1	412 (850)	424 (2425)	43 (2000)
2	422 (5145)	398 (850)	413 (1150)
3	363 (2425)	398 (850)	59 (9820)
4	363 (2425)	398 (850)	44 (2000)
5	43 (2000)	424 (2425)	412 (850)
6	363 (2425)	59 (9820)	413 (1150)
7	422 (5145)	-	-
@R1			
1	394 (5145)	394 (2175)	40 (2000)
2	394 (5145)	40 (2000)	394 (2175)
3	394 (5145)	-	-
4	394 (5145)	394 (2175)	-

Figure 3.23.: Downlink throughput as well as their sets of serving cells at 3 locations: S1 (suburban), C1 (campus) and R1 (rural) in AT&T.

In spite of these fluctuations, we still clearly see that the set of serving cells plays a decisive role to the performance perceived at locations S1 and C1, despite a less important role at R1. Both S1 and C1 have dense cell deployment: 13 unique serving cells in 12 sets are observed at S1 and 9 cells in 7 sets at C1. It is not hard to understand. Thanks to heavy infrastructure upgrades and more spectrum recently acquired, operators have deployed many more cells than before. If the first two (five) sets of serving cells are used at S1 (C1), data speed can be fast, say, $\geq$ 20Mbps (median). Unless specified, we use the median value afterwards. The highest speed

via set 1 can go up to 23 Mbps at S1 and 32.6 Mbps at C1. However, the good sets may not be always chosen in practice. The devices at these locations can be served by the "worse" choices. As a result, the median data speed shrinks to 1.5 Mbps (via set 12) or below 7 Mbps (via sets 8 -12) at S1. At C1, it reduces to 1.5 Mbps (via set 7), missing a much higher speed ($>$ 30 Mbps).

We want to point out that the performance comparison at S1 in Fig. 3.23 shows a larger performance gap than the one observed in our first example in Fig. 3.20, where round 1 is one run via set 12 and round 2 via set 3. That is, at S1, the missed performance (set 1 vs set 12) is even larger. In round 1, the median data speed decreases from 23 Mbps to 1.5 Mbps, missing 14.3-fold instead of 10-fold (from 16.9 Mbps to 1.5 Mbps).

In the rural areas (here, at R1), we observe that the performance gap is much smaller. This matches with our expectation. Many fewer cells are deployed in the countryside. At R1, there are only three cells available. As a result, there is much fewer sets of serving cells. The fewer choices, the less likelihood to miss the good cells to serve the devices. In this instance at R1, cell 394 is always chosen as the primary cell with slightly different SCell combinations.

2) How common and how large is missed performance in the wild?

We next extend our study to all 14 test locations to see that missed performance is not the corner case. We define two metrics to quantify the missed performance at each location. We first locate the best and worse set of serving cells based on their median performance. We further compare their performance at the same percentile using the absolute and relative gaps:

$$\delta_\rho = P_\rho^{best} - P_\rho^{worst}, \gamma_\rho = (P_\rho^{best} - P_\rho^{worst})/P_\rho^{worst}, \tag{3.4}$$

where ρ is the percentile from 0 (min) to 100 (max), P_ρ^{best} (P_ρ^{worst}) is the ρ-percentile of the data throughput using the best (worst) serving cell set. We notice that data performance fluctuates and the way to determine the best/worse sets using a single

value (here, the median) is not reliable. It may be questionable to locate the best and worst sets without statistically significant difference at some locations (*e.g.*, at R1). This is why we introduce relative comparisons at all the same percentiles to make such comparison robust. Note that, if we fail to locate the best and worst sets, the calculated performance gap underestimates the missed performance in reality.

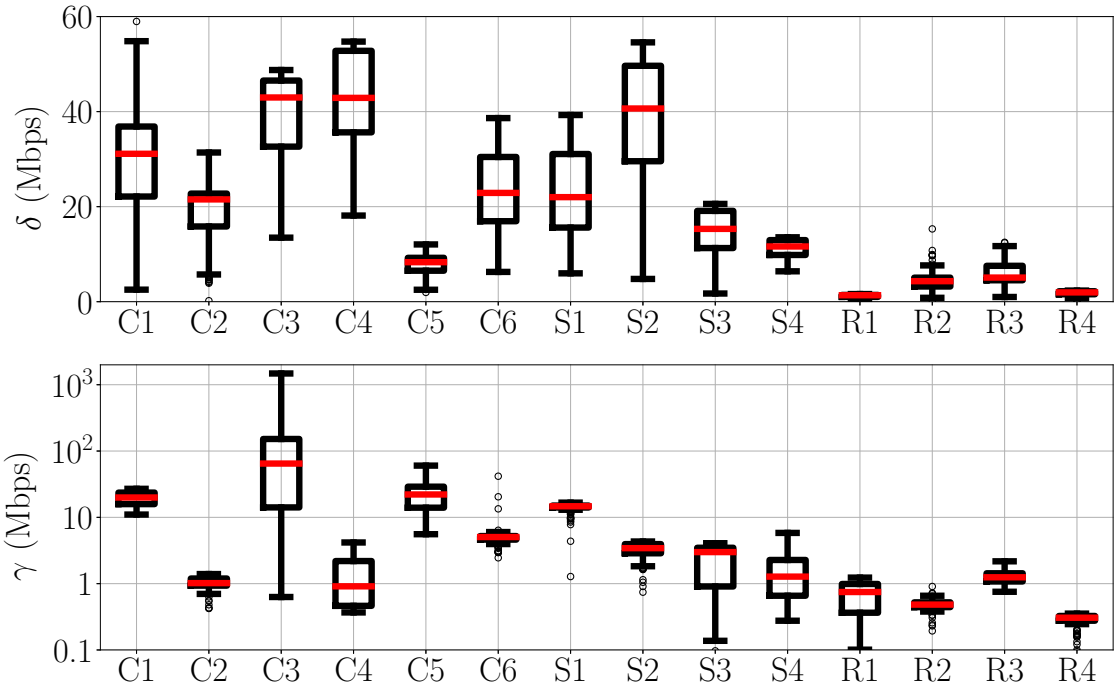
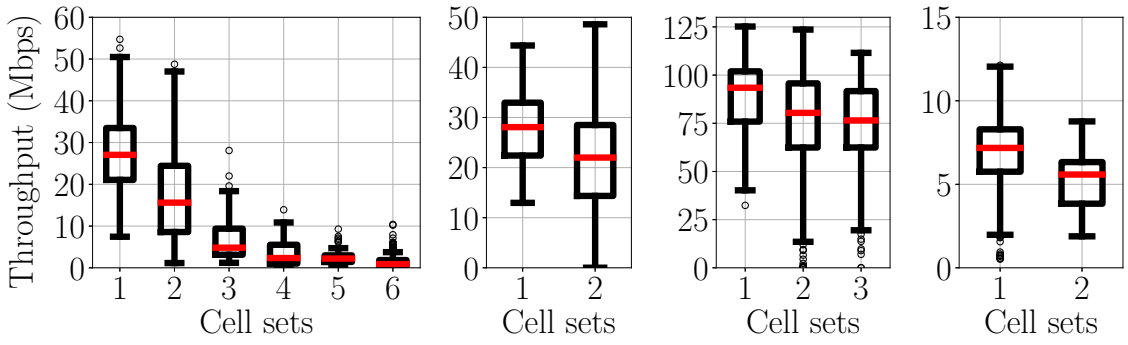


Figure 3.24.: Missed performance at 14 locations in AT&T.

Fig. 3.24 plots the distributions of δ_ρ and γ_ρ at 14 locations. It confirms our above findings at more places. Big performance gaps are widely observed in campus and suburban areas whereas the gaps are much smaller at rural areas. Specifically, δ_ρ goes up to 42 Mbps at C3 and C4. It is more than 20Mbps at 7 out of 10 locations (70%) in campus and suburban areas. The gaps are slightly larger in campus than in suburban. In rural areas, δ_ρ is more than 5 Mbps at two out of four locations. This is consistently observed in terms of the metric γ_ρ. For example, γ_ρ is more than 60 at C3. We take a closer look and find that this is because the worst performance at C3 is awful (best vs worst: 47 Mbps vs.750 Kbps). From the user perspective, such low data



(a) S4 (V) (b) S2 (T) (c) S2 (S) (d) R4 (S)

Figure 3.25.: Throughput at selected locations in Verizon, T-Mobile and Sprint carriers.

speed hurts user experience, especially when 47 Mbps is actually available but missed by the carrier's network operation. We also note that these two metrics are not fully correlated. For example, small δ but large γ (22x) is observed at C5, which implies that the absolute throughput in the worst case is low (370 Kbps). Namely, significant improvement is possible even when the best performance observed is modestly good (8.9 Mbps). In contrast, we observe large δ_ρ but small γ_ρ (γ_{50} =1x) at C2. This implies that the throughput in the worst case is not that small but the absolute improvement value is still significant (19.5 Mbps vs. 40 Mbps). We would like to highlight that $\gamma_{50} > 1$ (namely, the worst $< 50\%$ of the best) implies considerable improvement potential. We observe $\gamma > 1$ at all the test locations in campus and suburban areas. Even in rural areas, γ is larger than or almost close to 1 at two out of four locations (R1 and R3).

3) Carrier-specific findings.

We observe three carrier-specific results in our study. Here, we illustrate the key observations using four instances in Verizon, T-Mobile and Sprint (Fig. 3.25). The findings are confirmed in our larger-scale measurement described next. First, we find that the number of unique serving cells per location is much smaller in T-Mobile and Sprint, while it is comparable in Verizon. This is because T and S deploy fewer cells

than other two carriers. Specifically, only 2 (3) sets are observed at S2 in T-Mobile (Sprint). Similar results are observed at most locations in campus and suburban. Second, the measured performance gap between the best and worst case is much smaller in T and S because of fewer choices of serving cells. Compared to A and V, T and S have much smaller γ_{50} (at S2, T: 27% = (28-22)/22, S: 22% = (93.4-76.5)/76.5)). We note that δ_{50} in Sprint is not that small (16.9 Mbps at S2). This is because Sprint has much larger data speed than other three carriers. We check all other locations and find that the maximal speed in Sprint goes up to 160 Mbps, which is the fastest in four carriers. Third, all the results in rural areas are consistent across all the carriers. Gaps are much smaller, even in terms of δ_ρ for Sprint. This is because its data speed in rural areas is not fast (*e.g.*, 7.2 Mbps at R4).

3.3.4 How frequently the missed performance is observed

We further check how widely missed performance exists at a larger scale, say, at a city scale. Certainly, the above static tests do not scale. We thus run driving tests many times across every road and learn performance gaps per grid (roughly, 55 m x 42 m). For each grid, we perform the above analysis to learn the best and worst serving cells sets, and use δ_ρ and γ_ρ to quantify the missed performance. We plot the cumulative distribution function (CDF) of δ_{50} and γ_{50} (the median of estimated gaps) across all the grids in Fig. 3.26.

Before we present our findings, we would like to emphasize that such driving measurements may largely underestimate the missed performance in reality. Compared with static tests, we do not take any extra actions on the device to affect network operations. Consequently, the observed sets of serving cells are only induced by mobility and the optimal performance at each grid is highly likely undervalued. The "best" set of serving cells may not be chosen under the current and default operations, no matter how many times we drive across these grids. That is, our quantification in the driving tests are likely more modest, compared to those in static tests.

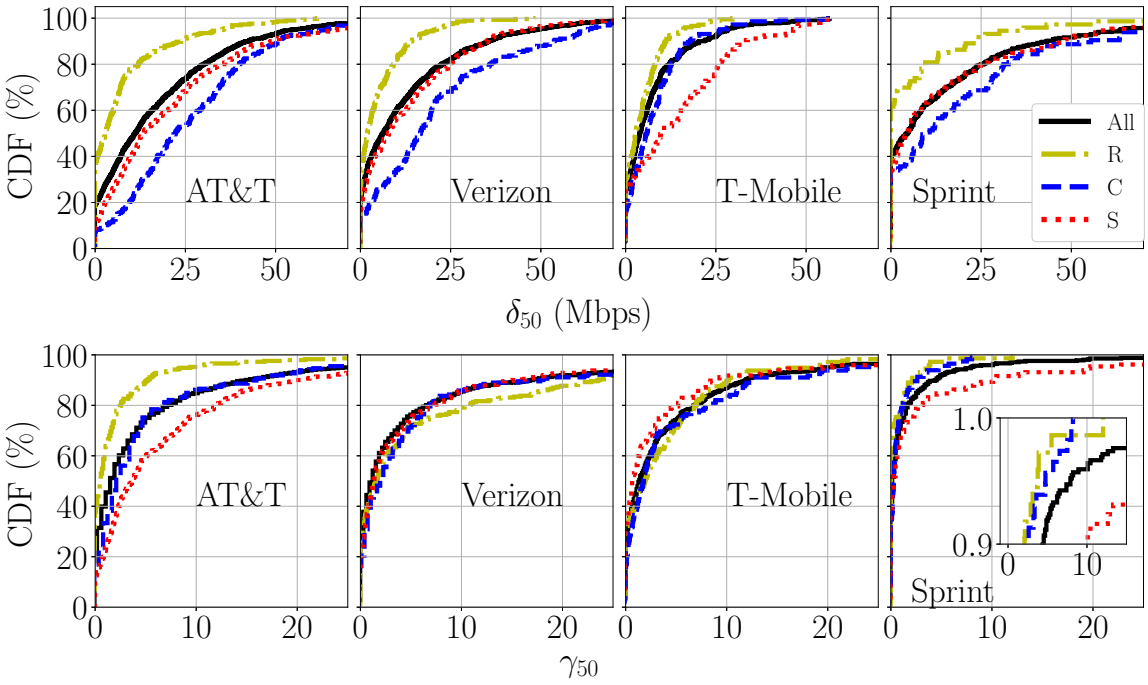


Figure 3.26.: CDF of δ_{50} and γ_{50} over all the grids in driving tests.

Even so, we see that significant performance gaps between the 'best' and the 'worst' cases frequently occur in reality, in all four carriers, despite possible underestimations. Specifically, in AT&T, we see that $\delta_{50} > 25$ Mbps at more than 43%, 29%, 9% and 26% grids in campus, suburban, rural and all areas. In other three carriers, we observe $\delta_{50} > 25$ Mbps at more than 17%, 6% and 20% places out of all the grids, in V, T, S, respectively. More than 20% grids in all the zones observe δ_{50} over 30 Mbps (7.4x), 22.6 Mbps (6.8x), 12.1 Mbps (6.9x) and 25 Mbps (1.7x) in A, V, T, S, respectively. These results are consistent with our previous findings in §3.3.3. T-Mobile, compared to other three carriers, has a relative smaller δ_{50} because its absolute speed is the smallest at more places. However, in terms of γ_{50}, the relative gap is not small in T-Mobile. More than 10% grids have γ_{50} larger than 15.3x, 17x, 12x and 4.6x in A, V, T, S, respectively. At more than 50% grids, γ_{50} is larger than 1.8x, 1.2x, 1.4x and 27% in A, V, T, S, respectively. γ_{50} is relative small in Sprint

because the absolute throughput is large in campus and residence areas but the gap is not.

We also notice two new findings. First, no gaps are observed at 20%, 20%, 25% and 39% locations for A, V, T, S, respectively. This is because we observe that the phones are always served by the same serving cells at these locations. This indicates that more than one serving cell sets are observed at other places. It depends on each carrier's cell deployment. This is also confirmed in our following study on the serving cell dynamics (Fig. 3.27). Second, the missed performance generally decreases from campus to suburban and then to rural. The only exception is T-Mobile. We find that the best case performance is comparable in both regions but the worst one is much worse in the suburban areas.

3.3.5 Root cause analysis

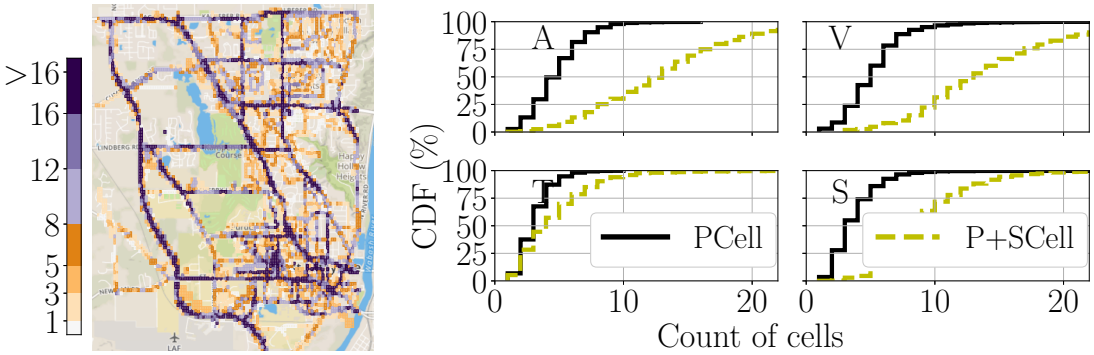
In this section, we present our preliminary efforts to learn why behind the missed performance. We focus on one core question: *why is the 'best' or 'reasonably good' set of serving cells NOT chosen in those poorly performed runs?* We find that the current network operations in serving cell selection should take the blame. In particular, when carrier aggregation is enabled, the selection of serving cells is done in two steps: PCell selection and SCell selection which is decided by the selected PCell. The first step is managed by the network with the assistance from the mobile device, via a standard procedure as regulated in [16] and studied in our prior work: The network first configures parameters for measurement, reporting and cell selection, and then the device performs measurement and reports measured results to the network as configured. The network decides whether to change or keep the currently PCell and executes its decision. The second step is determined by PCell and no standard procedure or policies have been disclosed. We identify two issues in network operations:

1. *These good choices of PCells are missed because current network operations are largely designed for seamless connectivity instead of the best data performance.*

2. *These good choices of SCells are missed because PCells can not choose SCells out of all available choices. Instead, they are restricted to a smaller pre-configured group.*

1) Missing the best or good PCell(s).

We first show that operators have deployed dense cells at most places. Fig. 3.27 shows the CDF of the number of unique serving cells (PCell only, and PCell + SCell) observed per grid in our driving tests. We use the map of observed PCell counts in AT&T (Fig. 3.27(a)) to illustrate its geographic distribution. Similar results are observed in other carriers. We see more than 5 PCells (13 P+SCells) at more than 50% grids in A and V. T and V have less dense deployment, and we still see more than 3 PCells (4 P+SCells) in T and 3 PCells (8 P+SCells) in S at more than 50% places.



(a) Counts of P+SCells (A)

(b) Serving cells per grid

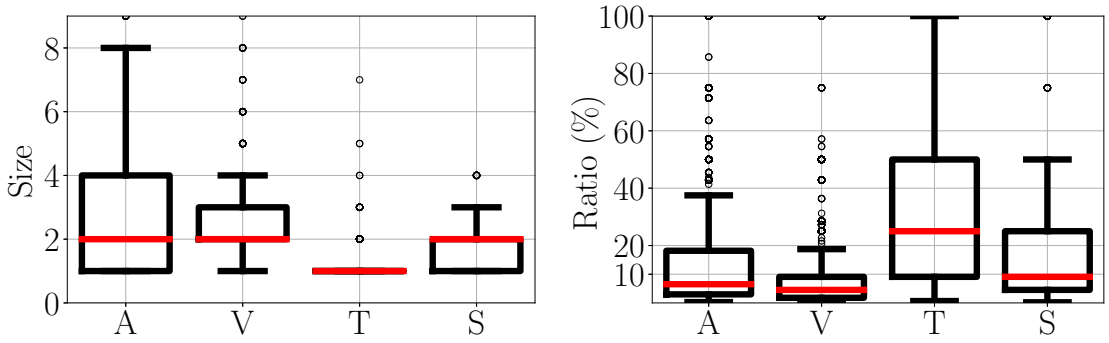
Figure 3.27.: Number of unique serving cells observed in our study.

However, such merit of denser cell deployment does not yield better performance in every case, despite more choices enabled. We observe that PCell selection is primarily determined by its radio quality evaluation, as disclosed in our previous studies. Take S1 (AT&T) as an example (Fig. 3.23). We check top four sets of serving cells at S1, excluding those observed in poor cases (sets 5-12). Apparently two cells are among top choices for PCells at this location: 244 (2000) and 102 (2425). We exclude cell 166 (850) because the performance using set 11 and set 7 are not so good. We gauge

that performance at set 2 is likely contributed by SCells (by cells 244 and 102). We also notice that the measured radio quality for each cell is consistent (with small fluctuations) across almost all the runs, unless specified. Specifically, we see that signal strength/quality (RSRP/RSRQ) of cell 244 at S1 is measured in range of (-114 dBm, -117 dBm) (RSRP) and (-14 dB, -16 dB) (RSRQ). Those runs without selecting cell 244 (*i.e.*, sets 5 and 7–12) is because the initially chosen PCell has stronger radio strength/quality and/or its measurement/reporting does not lead to a cell re-selection. For example, the RSRP (RSRQ) of cell 253 (selected as PCell in set 12) is -105 dBm (-14 dB), stronger than the one of cell 244. As a result, cell 244 is not considered. In another example, the RSRP (RSRQ) of cell 45 (selected as PCell in set 5) is measured to be -116 dB (-18 dB), which is slightly weaker than the one at cell 244. However, its measurement reporting is configured to be triggered only when the candidate cell is 5 dB stronger (RSRQ), which is still not satisfied in this case. This indicates that the current operation based on the radio signal strength/quality evaluation may fail to choose the cells that offer good performance. We guess that the reason behind such radio quality evaluation is simple to implement. After all, the located cell at least ensures seamless radio coverage, which is critical at the early phase of cellular networks when full coverage was a big concern of all the operators. We observe similar results at other locations. PCell selection depends on initial choice and its subsequent radio evaluation. Unfortunately, the cell with the best performance may not be chosen in this process.

2) Missing the best or good SCell(s). We also notice the power of carrier aggregation is not fully utilized. It is expected that PCell should select those best surrounding cells (or those with the strongest radio quality) as SCells, but it does not. Instead, for a given PCell, we find that SCells are selected from a very-limited subset which seems pre-configured. In the above example at S1, 13 serving cells are observed. This indicates that they at least pass the radio evaluation check. Theoretically, there are $C_{n-1}^2 + C_{n-1}^1 + C_{n-1}^0$ options of SCell combinations, where n is the number of serving cells observed at one location (here, $n = 13$). However, we do not observe

such a large number of SCell combinations. For example, PCell 244 has only one SCell combination (here, cells 102 and 106) out of 79 options. Fig. 3.28 plots the CDF of the size of SCell combinations (across all the grids) per PCell (left), as well as the CDF of the ratio of the observed size over all the possible options per location grid (right). We see that PCells are restricted to a much smaller group of SCells: 50% of PCells have no more than 2 SCells combinations in all four US carriers (1 in T-Mobile). The ratio is smaller 10% in all carriers except T-Mobile. As a result, it fails to make full use of the second chance to chase for better performance.



(a) Size of SCells combinations per cell

(b) Ratio of observed per grid

Figure 3.28.: Limited SCells are selected per PCell in 4 carriers.

3.4 Summary

In this section, we diagnose the State-Of-The-Practice mobility management in operational cellular networks from three perspectives.

First, we examine the functionality of mobility management. Mobile carriers are allowed for flexible configurations on their micro-mobility support scheme to address policy concerns. We find such flexibility makes it possible for two kinds of unexpected handoff behaviors. We identify the handoff instability issue (*i.e.*, the persistent handoff loop), as well as their triggering conditions, and partially validate it in operational networks. We also unveil the handoff unreachability issue (*i.e.*, mobile device fails to reach the desired cell) caused by improper configurations, inconsistencies/conflicts

between cells or between the cell and the device. The presented analysis also brings insight to avoid such handoff misbehaviours. Given the incurred damage, in terms of signaling overhead and performance degradation, such problematic issues should be addressed as we seek to build a more dependable, high-performance, mobile network infrastructure.

Second, we study the impact of diverse mobility configurations on data performance when moving. Our measurement study shows that configurations affect radio signal and performance as expected by design, but not all the impacts are intuitively "positive". We find that the serving cell's radio signal quality in handoffs changes as configured, and the data performance impacts also match with expected consequences of reasoning. However, through the comparison between different configured values by mobile carriers, we find improving radio signal quality is not the sole key to better data performance. In most cases, timing of handoffs is more crucial. Current configurations for active-state handoffs are "questionable" in terms of performance and efficiency.

Third, we present arguably the first study to unveil missed performance in operational cellular networks. Our measurement over 4 US carriers shows that missed performance indeed happens and happens a lot. We investigate missed performance potentials from the user perspective. We aim to uncover that user-perceived performance is sacrificed and such degradation is sometimes unnecessary. We hope to call for attention for making full use of those deployed infrastructure resources for better performance. We believe that it is eventually aligned with both operators' and users' interests. After examining the serving cells being used, we pinpoint the root causes into today's network operations on selecting cells. We note that performance variance caused by transient factors (e.g., radio and network loads) is not completely tackled in the missed performance evaluation. However, our measurement study shows that the serving cells is playing a persistent role and dominates data performance, despite variance per serving cell set. In addition, there are some real-world instances demonstrate that missed performance can be easily avoided through certain client-

side intervention (resetting the mobile network in our study). However, it is not all the case. We observe that resetting the network may not always select better cells where these cells are missed and more efforts are warranted to new design solutions. We see that the device is able to learn and profile data performance per serving cell set and take certain actions to get the desired cell set (and performance). It implies proactive device-side actions open up a promising approach.

4 IMPROVE MOBILITY SUPPORT WITH CLIENT-SIDE INTELLIGENCE

Increasing user access speed has been a main driver for mobile network evolution. Operators expect to deliver faster broadband experience through continuous infrastructure upgrade. They acquire wider radio spectrum, deploy denser cells, and migrate to advanced technologies (say, from 4G LTE to 4.5G LTE-Advanced to 5G New Radio). They enhance raw system capabilities to offer users higher speed (say, from tens of Mbps to a few Gbps). In this chapter, we explore potential improvements over network performance by making full use of the available network capacities with better mobility management. We boost data access performance to user devices by taking intervention to the default network handoff behaviors with client-side intelligence. In one hand, for single-carrier network, we design iCellSpeed to facilitate network-controlled cell selection with proactive device-side assistance towards more desirable cells. In another hand, for multi-carrier network, we propose iCellular to improve network access by doing proactive and adaptive multi-carrier selection. Both solutions achieve throughput improvement for user devices significantly.

4.1 Proactive Device-side Assistance Toward A More Desired Cell

As being previously unveiled in our preliminary study (section 3.3), there are significant performance gaps between what a device could get and what it actually gets in various settings. We continue to extend our measurement study and observe significant performance potentials missed of all four top-tier US operators (AT&T, Verizon, T-Mobile and Sprint). We further dive into why such low-utilization cases arise. It turns out that, the state-of-the-practice cell selection is held accountable. Without proper selection of those cells that yield higher access speed, the current

scheme under-utilizes the available infrastructure resources and misses fast wireless access to user devices.

Fundamentally, the problem is rooted in the network-centric cell selection scheme, which is designed for seamless connectivity but not for superior data performance. Note that, dense cell deployment is the norm, rather than exception, in most US regions. A user device can be served by multiple candidate cells (up to tens of cells) in principle. Despite abundant choices, few are considered in practice, and the poor ones are further chosen not without technical rationale. The practice favors those cells with good radio signal, but ignores other non-radio factors (e.g., channel bandwidth) that may have bigger impact on user-perceived data performance. The handoff procedure states that, as long as the radio quality is not too bad, current cell would remain effective despite bad data performance. It thus discourages leaving the current cell. When searching for new cells, the procedure stops once a candidate cell with acceptable radio quality is found. It thus often stops at a local sub-optimum out of a subset of candidates constrained by the current cell. Consequently, it misses the cells with the best performance out of all choices.

In this work, we propose iCellSpeed to address the identified underutilization issue. iCellSpeed has three design requirements. First, it seeks to reach the performance-oriented, *"global"* optimum from all candidates, rather than radio-based, *"local"* optimum. Second, it is compatible with the 3GPP standards without any infrastructure change. Third, iCellSpeed is designed as an on-device software solution to boosting mobile data access speed for the device. In brief, the design of iCellSpeed transforms a mobile device from a telecom-based *dumb, passive* terminal to an *intelligent, proactive* machine. Instead of passively following the decision made by the network, the device learns what is best for itself and supplies its own favored choices for the final selection made by the network.

There are two main technical challenges. First, device capabilities at the software space are quite limited. Even though the device learns the desired choice, it cannot directly select it. Second, it must work with the current selection mechanism that

makes the final decision at the infrastructure side. To address both issues, iCellSpeed takes an approach of the *"device-assisted, infrastructure-decided"* selection. The device takes runtime measurements and maintains historical performance profile to challenge whether it should accept the default decision made by the network. It performs online learning to infer missed potentials and determines its corresponding action (device-side customization). It further monitors the outcome to ensure better performance than the default choice.

We implement iCellSpeed on commodity smartphones, and confirm its effectiveness with large speed gains in both AT&T and Verizon. It increases access speed by at least 10 Mbps at 79.2% of locations out of 50 runs per location. It more than doubles the speed at 62.5% of locations, and achieves the gain up to 8.11x at one location (up to 28.4x in a single run). We want to highlight that iCellSpeed has one limitation that it would disrupt ongoing traffic for about 2 seconds on the application layer. As a result, it may not benefit mice traffic despite its improvement for bandwidth-intensive heavy traffic flows like video streaming, conferencing and file downloading. This limitation is rooted in its implementation constraint, as the device is incapable of taking action swiftly in the *software* space without corresponding support from the chipset. We note that iCellSpeed is designed to exploit under-utilized potentials, not to increase raw system potentials. There is no speed benefit on the network side in case full potentials have been used up by other devices. In a word, iCellSpeed is not perfect but offers a promising on-device solution to increase data speed.

In summary, iCellSpeed offers arguably the first on-device, pure software solution that guides infrastructure-centric cell selection for higher access speed. We make three main contributions:

1. We conduct extensive measurements to identify and quantify the missed performance potentials in today's mobile networks.

2. We unveil the limitations of current cell selection for missed potentials.

3. We design, implement, and evaluate iCellSpeed on commodity smartphones with instant speed gains.

4.1.1 An Motivating Example of Missed Performance Potentials

We first use an example to illustrate two new findings:

F1. In today's mobile networks, the device is not always served by the cells that can offer the highest access speed (Fig. 4.1).

F2. Higher speed can be achieved when the device takes extra actions without changes to the current network infrastructure (Fig. 4.1).

Fig. 4.1 plots the downlink throughput in two paired runs, where the phone keeps on downloading the same large file (500 MB) at a fixed location using AT&T. The first run (A) is performed under the default network operations. It starts after the user arrives at the test location along a fixed walking route. The second run (B) is performed at the same spot right after run A, while taking the device-side action (disabling band 5). In this case, the phone obtains 4.6 Mbps on average by default (A), while 49.3 Mbps is available (B). It is clear that the default network operations fail to select those serving cells that are used in run B and available in run A, thus miss higher speed (907% miss) (A).

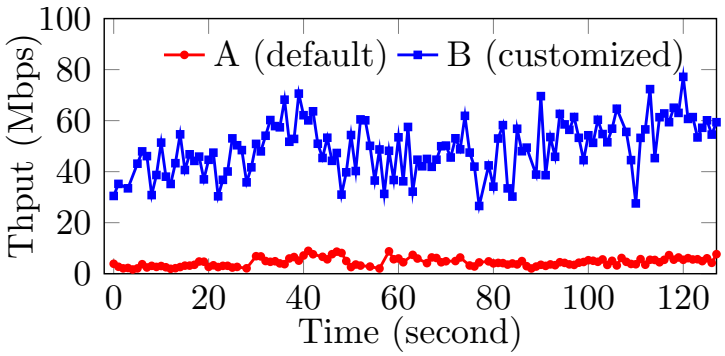
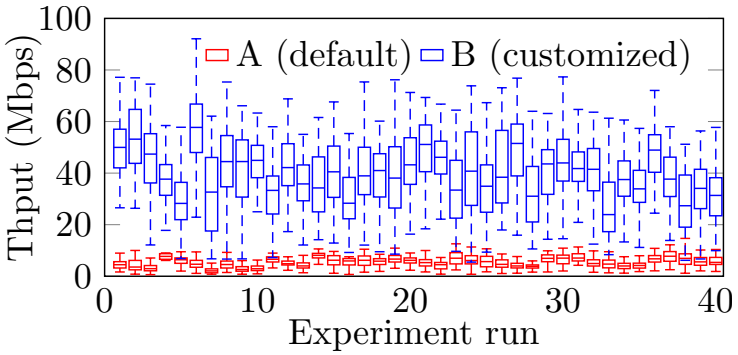


Figure 4.1.: One instance of missed performance potential at location L1.

We repeat the above experiment with more runs, and observe the same finding. Specifically, we repeat the paired runs at different hours of the day (from 9AM to 22

PM) and at different days of two weeks (in Dec 2019 and Feb 2020). Each run (B) with device customization (blocking band 5) immediately follows a default run (A). Fig. 4.2(a) compares downlink throughput (0, 25, 50, 75 and 100 percentile) in 40 test pairs. We treat the gap in each pair to be the what-if speed gain if B's choice were selected to serve the device in reality. This is a reasonable estimate because the achievable potential is no smaller than the achieved one, and we run what-if experiments almost simultaneously (under the same condition). We make two more observations.



(a) Comparison

	$\Delta^{(50)}$	$\Gamma^{(50)}$
min	29.3	351%
25th	35.9	459%
med	42.2	600%
75th	47.3	837%
max	60.6	1506%

(b) Speed gains

Figure 4.2.: 40 run pairs at location L1.

F3. The significant gaps persistently occur. Poor performance in reality can be largely avoided through device-side action.

F4. Persistent gaps imply that they are unlikely caused by transient factors like dynamic loads and time-varying radio signal quality. Instead, they are due to the poor cell selection in the current network operations.

In all our tests, device customization (blocking band 5) brings significant speed gains. By default, the phone constantly gets poor performance (medium speed $<$ 8.5 Mbps, maximum $<$ 12.4 Mbps). To quantify speed gain, we define two metrics,

$$\Delta_k^{(\rho)} = \psi_{k,B}^{(\rho)} - \psi_{k,A}^{(\rho)}, \quad \Gamma_k^{(\rho)} = (\psi_{k,B}^{(\rho)} - \psi_{k,A}^{(\rho)})/\psi_{k,A}^{(\rho)}. \tag{4.1}$$

ρ is the percentile from 0 (min) to 100 (max). $\psi_{k,A}^{(\rho)}$ and $\psi_{k,B}^{(\rho)}$ are the ρ-percentile performance (here, throughput) in the default (A) and customized (B) run at the k-th pair. At this location, we observe that the median speed increases by at least 29.3 Mbps and up to 60.6 Mbps. As showed in Fig. 4.2(b), in more than 50% tests, the median rate grows by at least 42.2 Mbps. In terms of $\Gamma^{(50)}$, the simple device customization scheme brings up to 15-fold gain with the minimal growth of 351%.

Data speed fluctuates over a small window of time (Fig. 4.1 and 4.2(a)). Such fluctuations are partly induced by dynamic resource allocation under changing traffic load and partly caused by varying radio channels. These transient factors contribute to speed variance in each run, but not the significant gaps observed in all tests. Instead, they are primarily attributed to the quality of the serving cells.

Table 4.1. Serving cell sets in use at Location 1.

	Main sets of serving cells	**(%)**
A	S19: {2425 (363) - none - none}	100%
B	S1: {5145 (16) - 850 (16) - none}	47.3%
	S3: {850 (16) - 5145 (16) - none}	39.2%
	S4: {2000(103) - 5145(16) - 2175(103)}	10.4%
	others	3.1%

We examine the set of serving cells in these 40 tests (Tab. 4.1). Each set is represented as {PCell - SCell_1 - SCell_2}. In the default runs (A), only one serving cell is used (the potential of CA is wasted). Each cell is represented by its channel number(cell ID) here, *e.g.*, 2425(363). Channel 2425 is centered at 871.5 MHz on band 5, with 5 MHz channel bandwidth. In the device-customized runs (B), this poorly-performed cell is filtered out by disabling band 5 at the device. Data speed rises with distinct serving cells. We see that CA takes effect and more serving cells are being used. It leads to three main choices and the first two use identical cells. The occurrence frequency is computed in terms of the observation duration (but not the number of runs), because more than one serving cell sets are used in some runs.

The total channel bandwidth does matter. All these new sets have larger aggregated bandwidth (25/25/20 MHz) than the default one (5MHz). We will confirm the impact of channel bandwidth later in §4.1.2. We would like to highlight that the speed variances in the default runs are relatively small. This implies that such poor performance is primarily determined by the selected serving cell(s), despite small variations due to transient factors.

F5. The above case is not rare. Significant potential miss is frequently observed among all four operators.

We conduct a measurement study across a small city and several regions of three other cities in the US. We find that significant potential miss happens frequently in all four US carriers (§4.1.5). Missed potentials are caused by the poorly-performed cell selection.

4.1.2 Why Poorly-performed Cells Are Selected?

We next examine why. It turns out that,

F6. The legacy design for seamless radio access should be held accountable. Today's cell selection is largely designed for seamless *connectivity, but not for best connectivity and data performance, thus missing good candidates for serving cells.*

In fact, current selection practice favors the cells with stronger radio signal quality over those offering higher speed. Such practice is not without rationale. However, in today's mobile networks where dense cell deployment is the norm, it is likely to underutilize high speed potentials empowered by rich cell choices at most places.

We use the above example at L1 (AT&T) to illustrate its limitations. Same/similar conclusions hold at many other spots for all four US operators in our large-scale reality check (detailed in §4.1.5). At L1, we run an extensive study. Different from §4.1.1, we perform only the default runs without taking any extra device-side action. To observe as many candidate cells as possible, we take diverse walking or driving routes to reach L1. We run 192 tests sporadically from Dec 2019 to Feb 2020. Each run lasts for 2 to

10 minutes and the total time observed at L1 is about 10.5 hours. We see 57 cell sets out of 31 cells over 6 bands (2,4,5,12,30,66). Among them, 19 sets out of 21 cells are used for more than 60 seconds. Fig. 4.3 plots their data performance and occurrence frequency, in descending order of median speed. We make two observations at L1.

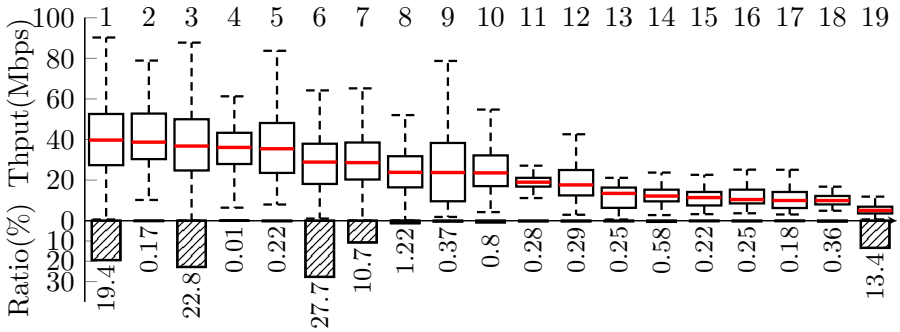
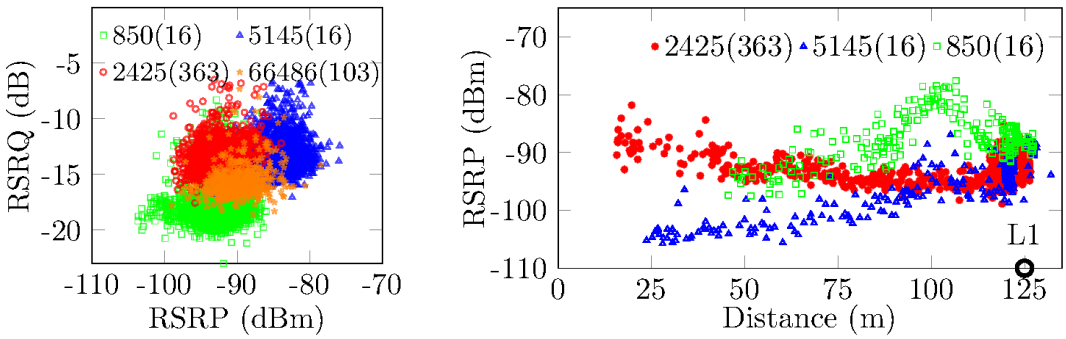


Figure 4.3.: Performance and occurrence frequency of main serving cell sets at L1 in AT&T during a 3-month reality check.

F7, abundant choices bring rich diversity in cell selection. There is no single winner at a location. We see that five sets (1, 3, 6, 7 and 19) are frequently selected at L1. S1 (short for set 1), S3 and S19 refer to Tab. 4.1. S6 is {2425(363)-1150(413)-none} that enables CA compared to S19. S7 is {66486(103)-none-none} which uses a PCell on band 66 (a superset of band 4). Fig. 4.4(a) plots radio quality of four common PCells at L1 in all the runs. RSRP/RSRQ (reference signal received power/quality) are two key measures of radio evaluation. We see that all four cells have comparable radio quality, stronger than most other cells at L1. This is why they are often selected.

F8, abundant choices cannot guarantee the selection quality. The choice with poor data performance is regularly or even repeatedly selected in reality. At L1, S19 is frequently selected but the offered speed is 8× slower than the best (via S1). We observe that S19 is always selected when the device moves to L1 along one direction (§4.1.1). Fig. 4.4(b) plots RSRPs of these three cells in such a walk instance towards L1.

We further use this instance to illustrate how current selection practice misses good choices in four aspects.



(a) Four cells' RSRP/RSRQ at L1.

(b) RSRP of three selected cells in one walking instance towards L1.

Figure 4.4.: Cells' radio signal at L1.

F9, Cell selection is primarily radio-based and ignores channel bandwidth. All steps of measurement, reporting and decision to perform a cell selection count on radio quality evaluation. For instance, whether to measure candidate cells depends on how weak the current one is. Which cell to report depends on whether its radio signal quality is strong enough. Which cell to select depends on who has the strongest radio quality out of those reported ones.

Such radio-based cell selection ignores non-radio factors which impact data performance, such as channel bandwidth. Cell 2425(363) has only 5 MHz and offers the lowest data speed at L1. 850(16) uses 20 MHz and offers higher speed. But 850(16) is not favored at locations where its radio signal is slightly weaker than 2425(363) (say, < 50 m). Note that, along the route towards L1 (Fig. 4.4(b)), 2425(363) is stronger than both good choices of 850(16) and 5145(16) at < 50 m locations (more than 75 m away from L1). This is why 2425(363) often wins at these places. Selecting the cells regardless of their bandwidths results in high likelihood of selecting poorly-performed cells.

F10, Radio-based cell selection discourages leaving the current cell despite poor performance. We find that no cell selection will be triggered as long as the current cell's radio signal quality is tolerable. In this walk instance, we see that cell 850(16) is even stronger than 2425(263) when we move closer to L1, say at [50 m, 125 m].

However, 850(16) is not considered in this instance and rarely considered in all the instances with cell 2425(263) as PCell. This is because 2425(363) has RSRP around -95 dBm and its RSRQ stays above -15 dB, which does not meet the criterion to measure other cells. Hence, cell 850(16) and other cells (*e.g.*, cell 5145(16)) are missed despite their far better performance.

F11, Current selection practice fails to make full power of CA. CA expects to leverage SCells to increase channel bandwidth and data performance. However, its power is hampered. CA is feasible at L1 when 2425(363) is a PCell, as S6 {2425(363)-1150(413)-none} is commonly observed. However, in the walk instance, no CA is enabled. It loses the second chance (via SCells) to make up the missed potentials when radio-driven selection picks a poorly-performed PCell. We further find that CA does not explore all possible SCells. The 3GPP standard [16] requires that PCell and SCells must co-locate on the same cell tower. This physical constraint is necessary to implement CA within the same radio protocol stack. However, not all the cells on the same tower are open to CA. We check all the cell sets in the neighborhood of L1 and group the cells as long as they appear in one cell set. We see 6 cells on the tower of cell 850 (16) and all of them have acceptable signal strength. Theoretically, there are $C_{6-1}^2 + C_{6-1}^1 + C_{6-1}^0 = 18$ CA options. However, we only observe 2 cell sets in reality: S3 {850(16) - 5145(16) - None} and S12 {850(16) - None - None}. It seems that the PCell - SCell combination is constrained out of a few pre-set choices. This is partly validated in our large-scale study (Fig. 4.10). We gauge that it is constrained by cell deployment and CA which is managed by the operator.

F12, Last but not least, current selection practice seeks for local optimum, not for global optimum. We find that its initial cell choice profoundly restricts the subsequent operations and consequences. It does not only discourage a cell selection, but also limits the scope of candidate cells considered for a selection. Take radio measurement as an example to illustrate this local constraint. There are intra-freq and inter-freq measurements [16]. The former measures the cells over the same channel and the latter measures more cells over distinct channels. What cells to measure depends on

the current one and its radio quality. In the walk instance (Fig. 4.4(b)), cell 850(16) or cell 5145(16) is not measured because no inter-measurement is invoked. We want to point out that the current serving cell is unaware of this biased choice at runtime when it determines the next target cell out of a small subset, not out of a whole set of all candidate cells. This way, there is no escape to better cells once the current cell gets stuck at a local stable choice. Multiple stable choices correspond to multiple winners at L1.

We would like to emphasize that we do not intend to blame the selection practice. It is indeed effective and efficient to ensure seamless radio coverage while retaining reasonable overhead (*e.g.*, avoiding handoff ping-pong effects). It switches to new cells only when the current ones are about to fail. At the early phase when full coverage was still a big concern, good radio was highly correlated to good performance. However, good radio $\neq$ good performance in today's mobile networks with abundant choices at place. We argue that it is right time to revisit cell selection and solve its underutilization problem that ends with unnecessarily poor performance.

4.1.3 iCellSpeed Design And Implementation

We propose iCellSpeed to increase data speed by tackling the underutilization problem that stems from the existing cell selection scheme. iCellSpeed seeks to achieve higher speed gains by selecting better cells that better utilize available capabilities rather than through enhancing raw capabilities. The ideal and straightforward solution is to directly implement a clean-state selection scheme for performance optimality out of all choices. However, it is not practical (at least in the near future) because of big changes to network infrastructure. We thus propose iCellSpeed, a device-side solution which exploits software power available at commodity smartphones and works in concert with the current cell selection practice. It is compatible with standard mechanisms and operational network infrastructure, with no need to change the network

side. If successful, iCellSpeed promises to bring immediate benefits for the device and by the device.

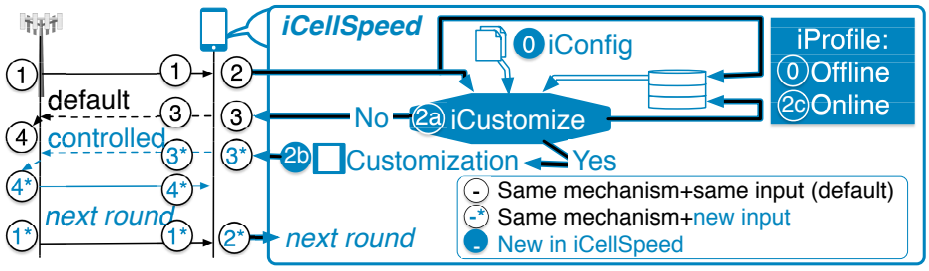


Figure 4.5.: iCellSpeed's overview and main operation flows.

1) Overview of iCellSpeed. Fig. 4.5 depicts iCellSpeed's main components and operation flows. The center is to use performance profiling as the main instrument and enable proactive device-customization atop of the existing network-centric cell selection procedure. Profiling leverages historical measurements to accumulate global knowledge and thus offers the ability to challenge and correct the improper decision out of partial information at runtime. The device switches its role from a dumb, reactive terminal to an intelligent, proactive one. Instead of simply following the commands from the network and executing rigid reactions pre-implemented in the chipset, the device proactively overrides the default reactions to influence the consequence of cell selection. But the serving cell is still decided by the network eventually.

Two core enabling modules are iCustomize (2a) and iProfile (0 and 2c). There are two main operation flows.

A. Device-side customization at runtime. iCellSpeed makes one change at the device side, incorporating with the existing handoff procedure. It adds the iCustomize step to break a direct chain between measurement (②) and reporting (③) which are configured by the network and executed at the device. Instead of passively following the commands from the network, the device is empowered to question and challenge the default selection operations, because the device stays alert of the identified limitations in §4.1.2 and the resulted potential miss. Intuitively, iCustomize is to compare real-time performance and current choice with the profiles

learned in advance. It then determines whether better cells are missed at runtime, and which action from device side is feasible to prompt better cells to be selected. If iCustomize confirms no need of further device actions, no change will be made and the default procedure proceeds (see the branch ②-2a-No-③-④). Otherwise, it executes device-side customization to override the default reaction and thus influence the selected cells (see the branch ②-2a-Yes-2b-3*-4*). Note that iCellSpeed does not alter the existing mechanism or network-side functions (say, ①, ③ and ④), which are also beyond control. Instead, it leverages just device power to change the *input* of these functions so as to indirectly affect the *consequence* of selected cells, towards a more desirable choice. iCellSpeed uses more intelligent device assistance to comply and complement the cell selection practice, while improving the chance of selecting cells with better performance.

B. Device-side profiling at runtime and offline. iProfile is to leverage measurements in the past and accumulate *global* knowledge regarding performance at every place so as to combat the limitations of local views used for cell selection at runtime. It gathers what cells are available, how they are selected, and how they perform in reality. Then it aggregates observed samples into the profiles regarding availability of choices (cell deployment), frequency of choices (cell selection), and performance of choices. It supports both offline and online modes. The former creates initial profiles through offline training or crowdsourced measurements. The latter updates the profiles with measurements over time.

2) iCustomize. It is not easy to decide a device-side action because its impact is not deterministic. As illustrated in Fig. 4.5, the final decision is still made by the network. Take the example at L1 (Fig. 4.1) to illustrate control uncertainty. Blocking band 5 results in three possible consequences, not a single deterministic one. Fortunately, all three new choices are positive, offering much higher speed than the default one (which is the worst set observed at that location). However, it is not always true. At other places, multiple new possibilities can be positive and negative. The key of iCustomize

is to predict both gains and risks (negative gains) and tame control uncertainty to determine the proper device-side action.

Specifically, iCustomize addresses three technical questions:

Q1. Should the device accept the default cell set?

Q2. If no, what action should be taken?

Q3. Should the device accept the previous action?

A naive idea. The straightforward solution is to predict the gain associated with each possible action and then maximize the predicated gain to determine whether and how to take actions (Q1 and Q2). It can be formulated into a classic Expectation-Maximization (EM) problem [20]. Let us assume that the iProfile module has sufficient samples in the past and thus gathers reliable knowledge for a given location. Let $\mathcal{S}$ be all possible choices at the given location, i.e. $\mathcal{S} = \{S_u | u = 1, 2, \cdots, U\}$. X_{S_u} is a random variable for data performance of cell set S_u. The resulted performance upon action π is a weighted sum of multiple random variables as $X_\pi = \sum_{u=1}^{U} P(S_u|\pi) \cdot X_{S_u}$, where $P(S_u|\pi)$ is the likelihood of selecting cell set S_u upon action π. Consequently, the decision is made by maximizing performance expectation,

$$\pi_* = \arg\max_{\pi \in \Pi} E[X_\pi]. \tag{4.2}$$

No action is taken when $E[X_{\pi_*}]$ is no better than the default one. Otherwise, π_* is taken. Q3 is used to address control uncertainty in case the new action brings a negative gain. To answer it, it monitors performance under action π_*. If the resulted new cell set is worse than the default one at the new round, the previous action is withdrawn and excluded from the action set Π. At next round, a new action is derived out of the updated action set the Π to maximize Eq. (4.2). Otherwise, it stops.

Our practical solution. However, the storage overhead of the above computation is too huge. We thus work on a practical solution that *approximately* predicts the gain and looks for a *reasonably good* action (customization) so that we probably boost data speed efficiently. This solution trade-off is feasible according to the famous PAC

(probably approximately correct) learning theory [21]. In our context, we enable a fuzzy logic that incorporates non-deterministic profile models and deterministic domain knowledge. Note the consequence of our customization is approximately, not precisely accurate. Once the previous action is incorrect, we take just-in-time reinforcement learning and correct our previous "mistake" with up-to-date feedback.

Fig. 4.6 shows the core logic of iCustomize. At the first level, it goes to NO-branch when no device customization is performed previously. Otherwise, it goes to the YES-branch at the next round to check whether the speed is worse. If no, no more action is needed. Otherwise, it excludes the previously-used actions and runs iCustomize* (the NO-branch).

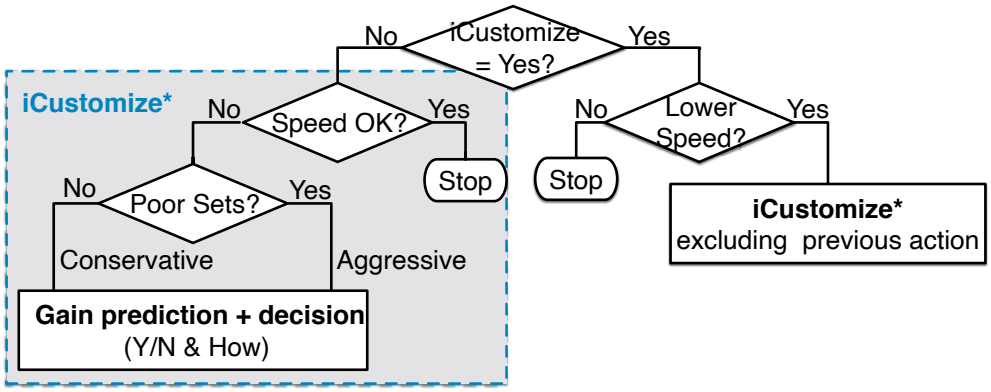


Figure 4.6.: The core logic of iCustomize.

iCustomize* uses a two-level decision sub tree. To decide whether to challenge the default choice, it takes the following factors into account: real-time performance of the current cell set, historical performance of the current cell set, and potential gains of device-side actions. On the first level (speed-OK trigger), we examine if the current speed is satisfactory. We devise intuitive rules such as current performance no worse than a portion of the best (*e.g.*, above 70% of the 50-percentile performance of the best set), or current performance above a certain percentile (*e.g.*, the midpoint of the performance range). Note that the used information (*e.g.*, the best/worst performance) can be easily learned by iProfile. We test with several rules in our evaluation and shows that these intuitive rules work robustly to differentiate good

runs from runs with missed potentials. The reason is simple. At places with significant performance potential miss (see the example of S19 at L1 in Fig. 4.3), there exists huge room to design speed-OK rules to differentiate good and poor performance. At places with small or overlapping performance gaps, low accuracy is tolerable since the likelihood to realize missed potential is low as well. As a result, intuitive rules suffice in iCellSpeed. On the next level, we check if the current cell set is poor based on iProfile. It decides whether iCellSpeed takes aggressive or conservative actions.

We next predict the gain and make a decision. We exploit domain knowledge to approximate Eq. (4.2) and simplify gain prediction.

We find that device-side actions are very limited. The ideal customization is to allow the device to directly lock the target serving cell(s) which offers desirable performance. However, such explicit cell locking is not available at almost all commodity phones (the exceptions [22, 23] can only work on specific phone). Software interfaces such as API [24], `AT` commands [25] and secret codes [26] can not lock cells. Constrained by available software power, three actions are considered:

A1: lock one frequency band,

A2: block one or multiple bands,

A3: reset, particularly via turning off and on mobile data (or the flight mode).

A1 is one special case of A2, when blocking all the other possible bands, except the one to be locked. A1 and A2 will rule out all the cells over certain bands. A3 is to give equal chances to all candidate cells by clearing the impact of the current choice. We note that all these actions have one downside in practice. Because they change frequency bands/channels or reset radio access, these device-side actions require disrupting radio resource control (RRC) and thus suspend data connection for a while. We evaluate the impact of the disruption time in §4.1.6. It is about two seconds at the application (APP) layer and several hundreds of milliseconds at RRC. Note that the disruption at RRC is mandatory but the extra disruption at the APP layer is avoidable with better mobile OS support. The disruption time is tolerable for elephant traffic flows which last long and require huge throughput. We thus consider

bulk file downloading in this work and leave its extension to other applications as future work.

We exploit the above knowledge to estimate $P(S_u|\pi)$. For A1/A2, $P(S_u|\pi)$ is zero as long as one cell in set S_u is explicitly ruled out. We further estimate $P(S_u|\pi)$ in proportion to $P(S_u)$ only for those eligible choices allowed by action π. A winner in the default runs is likely a winner under action π, because of the same network selection function. In reality, they often have relatively stronger radio quality. For A3, it is to reset radio access and the likelihood is the same as $P(S_u)$.

Given $P(S_u|\pi)$, we look for the action that maximize $E[X_\pi]$. We start with blocking one band. We iteratively expand the actions to be considered. We stop when there is no more chance to find an action with gain larger than the current maximum. One action's gain is estimated as the weighted sum of all eligible choices. We also notice that a big gain change may occur when one action is linked with a mixture of good and poor sets. So we continue only when the gain of the good choice is larger than the current maximum. Regarding performance, we use runtime measurement for the current choice and profiles for other choices. There is a slight difference when a good set is in use. Its runtime performance does not match its profile. We need to remove all high-rank actions which allow the current set and re-calculate the action with the maximal gain based on runtime performance measurement. If blocking any band(s) does not bring performance gain and current performance is way below the expectation, we can consider A3. Otherwise, no action is feasible. Its actual calculation is simple because we see that there are only 4-5 popular sets in most cases. Blocking one band is enough at most cases. We block at most 2 bands in this study.

3) iProfile. The iProfile module maintains and updates three types of profiles to support device-assisted cell selection.

(I) Availability of choices. This records working cell sets at each location. This provides global knowledge regarding cell deployment which remains stable for a long period (months or years). Its update is straightforward. Whenever a new serving cell set or cell is observed, it is inserted into its set of choices at the given location. This

helps us to efficiently profile abundant choices in reality. Even given measurements holes at some places, we can infer these choices based on records in its close vicinity.

(II) Frequency of choices. This records the consequence of cell selection in reality. $P(S_u)$ is the frequency of choice S_u in the default cell selection. We record the count of instances over time. It is updated as follows. The sum of $P(S_u)$ remains invariant (here, $\sum = 1$) but its individual value changes accordingly as the count of the observed choice (say, S_v) increases. When one specific action π is taken, we take a similar way to update its occurrence frequency.

(III) Performance of choices. This accumulates global knowledge on performance of choices at this location. Rather than storing a huge amount of raw samples over time for X_{S_u}, we record its performance statistics over time, which is represented by multiple percentiles, $\Omega = [\psi^{(\rho_1)}, \psi^{(\rho_2)}, \cdots, \psi^{(\rho_W)}]$. W and ρ_ω $(\omega = 1, 2, \cdots, W)$ are constants configured at the start. They are ascending percentiles such that (min) $0 = \rho_1 < \cdots < \rho_\omega < \rho_{\omega+1} < \cdots < \rho_W = 100$(max). Offline profiling is simple. Given $N[0]$ samples for one serving cell set S_u at location L_i, we calculate its performance statistical vector.

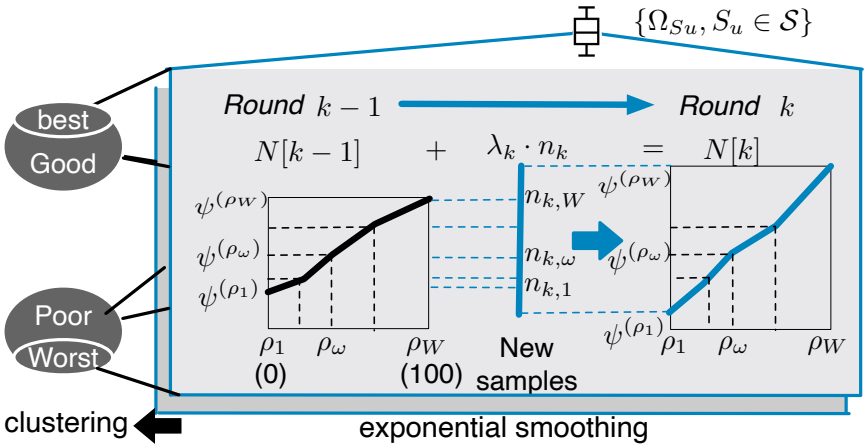


Figure 4.7.: Profiles and online performance profiling.

Online profiling is illustrated in Fig. 4.7. We develop a fast algorithm to update $\Omega[k]$, given the previous profile $\Omega[k-1]$ and n_k performance samples measured at round k. We update the total amount of effective samples as $N[k] = \lceil N[k-1] + \lambda_k \cdot n_k \rceil$.

Here, λ_k is a tuning parameter to pace the rate of forgetting historical records. It is used to balance sample staleness and approximation accuracy. Without loss of generality, we assume that new samples are sorted in ascending order. Namely, $\phi_{k,1} \leq \phi_{k,2} \leq \ldots \leq \phi_{k,n_k}$. We iteratively update $\psi^{(\rho_\omega)}[k]$ when ω increases from 1 to W. Clearly, when $\omega = 1, W$, we update the minimum and maximum as follows:

$$\psi^{(\rho_1)}[k] = \min(\psi^{(\rho_1)}[k-1], \phi_{k,1}), \tag{4.3}$$

$$\psi^{(\rho_W)}[k] = \max(\psi^{(\rho_W)}[k-1], \phi_{k,n_k}). \tag{4.4}$$

We then iteratively update $\psi^{(\rho_\omega)}[k]$, $2 \leq \omega \leq W-1$. The mathematical derivation looks sophisticated, and we present its simple heuristics first. There is no need to change Ω if new samples *perfectly* match with the existing performance profile. That is,

$$\phi_{k,n_{k,\omega}} \leq \psi^{(\rho_\omega)}[k-1] \leq \phi_{k,n_{k,\omega}+1}, 2 \leq \omega \leq W-1. \tag{4.5}$$

$n_{k,\omega} = \lceil n_k \cdot \rho_\omega / 100 \rceil$ is the position of ρ_ω-percentile sample in the new sample set. Otherwise, the profile should be updated. The displacement between $n_{k,\omega}$ and the sample position corresponding to $\psi^{(\rho_\omega)}[k-1]$ marks the potential update scope. The core idea is to approximate the old sample sets by assuming a uniform distribution between adjacent percentiles. We omit its mathematical derivation. The above update is iteratively performed with modest computation overhead. In turn, we will update $\psi^{(\rho_1)}[k] \rightarrow \psi^{(\rho_2)}[k]$ (based on $\psi^{(\rho_1)}[k]$ and $\psi^{(\rho_2)}[k-1]$) $\rightarrow, \cdots, \rightarrow \psi^{(\rho_\omega)}[k]$ (based on $\psi^{(\rho_{\omega-1})}[k]$ and $\psi^{(\rho_{x \leq \omega})}[k-1]$), and so on. The above generic-form works for any W and $\{\rho_\omega\}$. In this work, we consider a common setting like $W = 5$, $\{\rho_\omega\} = \{0\%, 25\%, 50\%, 75\%, 100\%\}$. We validate that our performance profiles are accurate enough for iCellSpeed.

To facilitate the iCustomize module (*e.g.*, configure the decision criteria), we aggregate per-set performance into per-location profile (illustrated by the outer box of Fig. 4.7). We run a clustering algorithm to learn the top and bottom groups, referred to as *good* and *poor*. Note that the *best* and *worst* cell set belong to the *good* and *poor*

groups, respectively. In case the performance profiles of the *best* and *worst* sets are similar, iCellSpeed is not needed because there is no significant performance missed. Otherwise, we are able to learn *good* and *poor* groups and use their performance profiles to determine thresholds used in iCustomize.

4) Implementation. We implement iCellSpeed on rooted Android smartphones. It is conceptually implementable on non-rooted phones, but we currently prototype it on rooted phones for two reasons. First, blocking/locking a band is unavailable through the existing APIs of commodity Android OS releases (*e.g.*, class `telephony` [24]). We use an encapsulated library over confidential secret codes. Root is not required if ROM is customized [27]. Second, we collect raw traces for debugging and evaluation. These traces include packet traces captured by tcpdump and mobile network signaling messages collected by MobileInsight [7]. Both work with rooted phones only. The second feature is not necessary for normal operations of iCellSpeed. We implement several intuitive rules in the proof-of-concept prototype.

4.1.4 Evaluation Methodology and Datasets

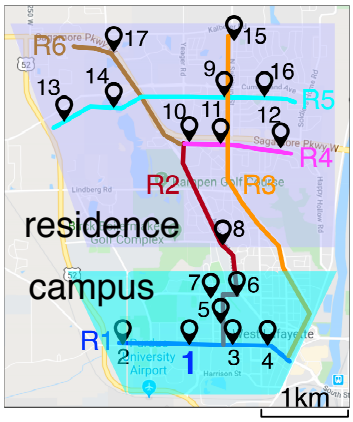


Figure 4.8.: Map at West Lafayette, IN.

Our evaluation is primarily conducted in a small city C1, West Lafayette, IN (4 Km × 4.5 Km), as showed in Fig. 4.8. We also consider several locations and routes in

three other cities – C2 (Los Angeles, CA), C3 (Austin, TX) and C4 (Lafayette, IN) – to validate its effectiveness and wide applicability in diverse real-world circumstances. We run two types of experiments without (A) and with (B) enabling iCellSpeed. We perform both static and driving tests. In static tests, we randomly choose 24 locations, including 17 locations in C1 in two representative zones: campus (aka, urban) and residence (suburban) and 2-3 locations each in other three cities. In driving tests, we use 10 fixed routes (Tab. 4.2) in four cities. These routes are popular in the test cities (*e.g.*, routine routes between work and home) and cover representative types. U/S represents urban/suburban and L/P represents local (< 35mph) and parkway (45-55 mph).

Table 4.2. Driving Routes for iCellSpeed.

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	Total
@	C1	C1	C1	C1	C1	C1	C2	C3	C4	C4	
Distance (Km)	2.9	3.6	5.8	1.8	4.2	2.4	2.0	2.6	3.2	1.8	30.3
Type	U,L	U+S,L	U+S,L	S,P	S,L	S,P	U,L	U,L	U,L	U,P	

We assess data performance in terms of downlink speed. We primarily use heavy traffic load which keeps downloading a large file (500 MB) from our lab server. We monitor the server's outbound link rate and ensure that it is not a speed bottleneck in our mobile network experiments. We later evaluate iCellSpeed while running some applications like video steaming and conferencing. We measure all four US operators while AT&T is considered in all the cases, because the unlimited plan of AT&T only temporarily throttles rate if the network is busy after using more than 100 GB. Sprint has the smallest dataset (10.8 hours, mainly on 2 routes in C1) due to its most strict rate throttling. We purchase multiple lines to ensure that the rate is not throttled when we run downloading experiments. For each operator, one test phone is used at one time to avoid contention for radio access unless specified. We use eight phones out of three models: Google Pixel 3/2/2XL. They use Qualcomm Snapdragon 835/845/855 chipsets which all support 4.5G. The results are not phone-specific. Similar findings are observed in all four operators unless specified. To

evaluate iCellSpeed, we first conduct extensive real-world measurements (A only) from Sep 2019 to Feb 2020, and then run A+B experiments together, primarily in Feb, March and June 2020. Our dataset D1 (Tab. 4.3) collects data speed samples for about 372 hours (static + driving) over 5,953 Km (driving).

Table 4.3. Datasets of iCellSpeed.

	D1 (A+B, heavy load)				**D2 (A, light load)**				**Total**
	A	V	T	S	A	V	T	S	
Time (hr)	247.9	68.6	45.1	10.8	111.2	231.6	137.7	68.5	921.4
Distance (km)	2,561	1,977	1,182	233	1,221	1,493	1,259	1,140	11,066

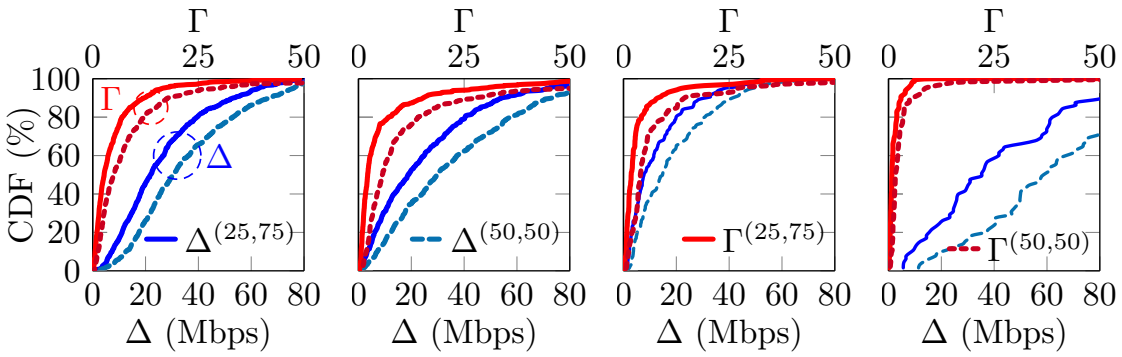
To learn real-world cell deployment and characterize abundant choices available, we perform a wider-area driving experiment in C1. In addition to six routes, we do a city-scale scan to cover every road multiple times (main roads: ≥ 30, almost all local roads: ≥ 5) in C1. We use mice traffic (ping Google every second) to keep radio connectivity active at all time. We run experiments primarily from July to Dec 2019 and get dataset D2 over 549 hours and 5,113 Km.

Both datasets are public available at [28].

4.1.5 Reality Check Without iCellSpeed

1) Missed potentials in reality. We first present our real-world measurements without enabling iCellSpeed in D1. This helps to better understand the test locations for iCellSpeed's evaluation. Our reality check shows that significant performance potential miss is frequently observed at many places for all four operators (Findings F1, F4 and F5). At all 24 selected locations (AT&T), we observe significant performance gaps as we see at L1 (Fig. 4.3). The median speed gap is at least 10 Mbps and up to 74 Mbps. Here, we present the results of the driving tests in C1, which covers a wider area than the static tests. We divide the roads into small grids (each approximately 55 m x 42 m) and retrieve missed performance per grid. We use a pair of metrics $\Delta^{(\rho 1,\rho 2)}$ and $\Gamma^{(\rho 1,\rho 2)}$ to characterize the absolute and relative gaps between

the $\rho 1$-percentile performance of the *best* set and the $\rho 2$-percentile performance of the *worst* set at each grid. We use two pairs: (1) $\rho 1 = \rho 2 = 50$, (2) $\rho 1 = 25, \rho 2 = 75$. Clearly, the latter is a more conservative approximation. Fig. 4.9 plots their CDFs across all the grids with enough runs and samples. Note that insufficient runs can not capture real-world diversity on serving cells and insufficient samples can not capture data speed dynamics. Hence, we only consider grids with $>$ 10 runs and samples $>$ 100 seconds. There are 690, 438, 139 and 66 grids considered for A, V, T, S, respectively. We want to highlight that the actual gap at each grid can be still underestimated given our limited measurement scale. Even in a more conservative way, we see $\Delta^{(25,75)} > 20$ Mbps at more than 55%, 48%, 20% and 74% for A, V, T, S, respectively. $\Gamma^{(25,75)}$ is at least 1 (speed doubled) at more than half of locations for all four operators. A and V have similar results on significant performance gaps. It is mainly due to their larger cell diversity than T and S. We notice that in Sprint, the absolute gap (Δ) is larger but the relative gap (Γ) is smaller. This is because the observed speed is much higher than other carriers, up to 160 Mbps. The lowest data speed is larger than 20 Mbps at almost all places, much faster than several Mbps or even hundreds of Kbps observed for other operators.

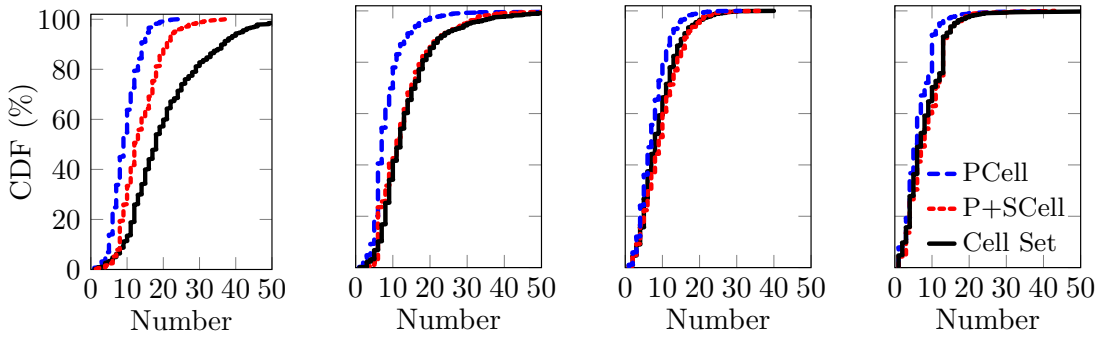


(a) AT&T (b) Verizon (c) T-Mobile (d) Sprint

Figure 4.9.: CDF of the observed speed gaps across C1.

2) Cell deployment in reality. We use a city-scale measurement in C1 to show that dense deployment is the norm for all four operators (Finding F7). We see similar

or even denser deployment in other cities, particularly in C2 which is one of top-3 US cities. We combine our driving tests in D1 and D2 and plot the CDF of the number of serving cells and sets observed across the measured grids in Fig. 4.10. We exclude grids with insufficient runs and samples in the same way. We see more than 9 PCells (12 P+SCells) and 18 cell sets at more than 50% grids in AT&T. T and S have more than 5 PCells (7 P+SCells) at more than 50% places. We also performed limited test at several locations in rural areas and find that it is much less dense for all four operators. We see that the performance gap is not significant without abundant choices. As a result, we believe that abundant cell choices are available in the cities.



(a) AT&T (b) Verizon (c) T-Mobile (d) Sprint

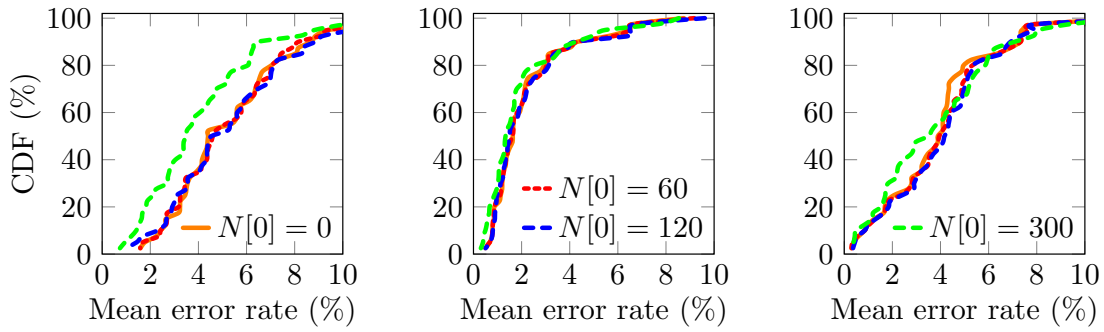
Figure 4.10.: CDF of the number of serving cells and sets in C1.

We also check all the cells used in C1 by four operators. We see that AT&T deploys 501 unique cells over 19 channels at 6 bands (2, 4, 5, 12, 30, 66). But the use of these channels is quite uneven. The most popular channels ($> 10\%$,) are 850, 2000, 5145, 2175 and 2425 in descending order. Only three channel bandwidths of 5, 10, 20 MHz are used. We admit that such device-side measurement may be still incomplete despite our extensive study. We observe similar results for other three operators.

4.1.6 Micro-Benchmark Evaluation

We first use the test results at the static locations to evaluate how iCellSpeed's main components work in reality.

1) iProfile. We compare performance profiling accuracy with different parameters. In particular, we test with different numbers of initial samples, $N[0] = 0, 60, 120, 300$, and three smoothing weights $\lambda = 1, 1.1, 1.5$. iProfile takes every experiment run by a chronological order. Meanwhile, the ground truth is calculated out of all the samples at this moment. Fig. 4.11 shows the average error rate for all popular cell sets (sample number > 600) at all static locations when $N[0]$ varies from 0 to 300. We show 25, 50, 75-th performance percentiles of cell set, and omit 0 (min) and 100 (max)-th percentiles because they are updated accurately. Fig. 4.12 uses one instance to illustrate the impact of smoothing weight over time (samples). We show performance profile of cell set 1 {5145(16) - 850(16)} at L1 with $\lambda = 1, 1.1, 1.5$ and $N[0] = 60$.

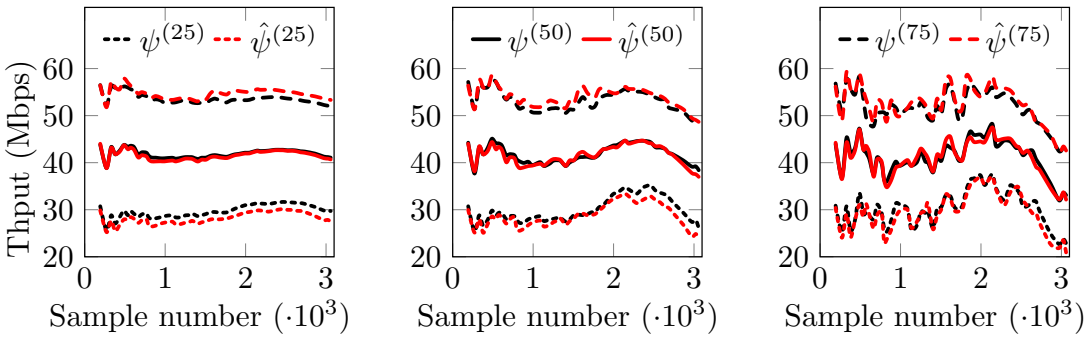


(a) 25th-percentile (b) 50th-percentile (c) 75th-percentile

Figure 4.11.: CDF of average performance profiling error rate over all popular cell sets at all the static locations.

We have three observations. First, iProfile achieves accurate estimation, by storing several performance percentiles, not a huge number of raw samples. Second, the accuracy results are very similar given various sizes of initial samples (even $N[0] = 0$). The low-percentile (25) is more sensitive to the initial sample size because fewer performance samples are considered for a low percentile. Third, the estimated profiles are quite accurate regardless of λ. Note that the smoothing weight of 1.5 induces more fluctuation in Fig. 4.12, which is consistent with our observations at other locations. This is because higher weight accelerates the pace of forgetting historical

data. Ideally, the weight should be tuned according to the elapsed time since last update. Considering most of our evaluation experiments are heavily conducted within one month (every few days), we set $\lambda = 1.5^{\lfloor week \rfloor}$ in our implementation. In the following evaluation, the default parameters are $\lambda = 1$ (1.5^0), $N[0] = 60$.



(a) $\lambda = 1$ (b) $\lambda = 1.1$ (c) $\lambda = 1.5$

Figure 4.12.: Profiling accuracy in one example (Set 1 at L1)

2) iCustomize. We evaluate the impact of the rules of "speed-not-OK", which determines whether iCellSpeed needs further actions. We define 4 rules accordingly:

Rule I: $\psi^{(75)}_{current} < 0.8 \cdot \psi^{(25)}_{best}$ or $\psi^{(25)}_{best} - 10$,

Rule II: $\psi^{(75)}_{current} < \psi^{(25)}_{best}$,

Rule III: $\psi^{(50)}_{current} < 0.7 \cdot \psi^{(50)}_{best}$ or $\psi^{(50)}_{best} - 15$,

Rule IV: $\psi^{(50)}_{current} < 0.9 \cdot \psi^{(50)}_{best}$ or $\psi^{(50)}_{best} - 5$.

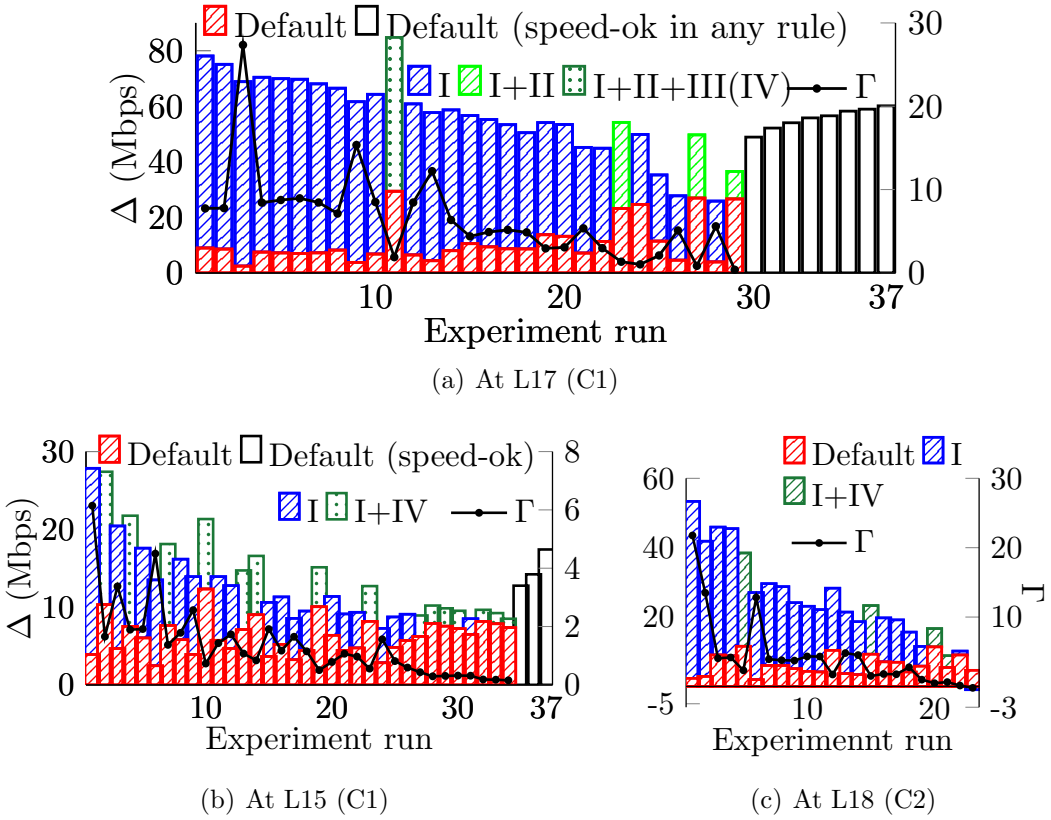
$\psi^{(p)}_{current}$ and $\psi^{(p)}_{best}$ are ρ-th percentile of performance in the current run and in the existing profile. We run the experiment as follows. We start file downloading on the device and manually disable iCellSpeed for the first $2-3$ minutes. That is, iCustomize is running (monitor performance for decision making), but *none* of its decisions is made. Afterwards, as long as *any* rule above is satisfied, the device takes action correspondingly. We use the collected traces to evaluate the rule impact.

Fig. 4.13 presents three showcases at L17 (C1), L15 (C1) and L18 (C2). We use average speed gain to evaluate the impact of the above four rules. In those plots, red bars represents default performance before the device action is taken, the

stacked bars above red ones represent performance gains achieved by iCellSpeed with different rules. There are also default runs with pretty high data speed (none of rules are satisfied), referring to black bars. We have two main findings at L17. First, all the rules work well. They detect all runs with big improvement room and do not bother with good runs (the last 8 runs). iCellSpeed exploits great potentials at L17 with speed gains of $20 - 70$ Mbps, up to 27-fold. Second, these rules work slightly different because Rules I to IV become more aggressive. There is small difference in not-OK runs detected by four rules at L17. Since the default performance is too poor in most runs, so even the most conservative rule (Rule I) is met. Rules I - IV miss potential gains in 4 runs (11, 23, 27, 29), 1 run (11) and 0 runs. Note the missed gains are relatively small and the only exception is at run 11, where the default speed is not too bad ($\sim$ 30Mbps) but the gain empowered by iCellSpeed is extraordinarily large. Generally, the speed-not-OK rule is a tuning knob to balance speed gains and the missed rate. The more aggressive rule brings smaller miss rate, but more likely with smaller gain.

We further apply Rules I and IV to other two locations. At L15, iCustomize improves poor situations by 8 Mbps and 7 Mbps on average for I and IV, given a limited achievable gain (median of the best cell set is 17 Mbps). At L18, iCustomize decides to take action in all 23 runs. It makes good catch in 19 runs with more than $1\times$ enhancement. Dynamics in performance provided by the target cell set accounts for small or negative gain in other runs, especially like run 23. This implies that more action rounds should be considered.

3) Disruption. Due to implementation constraints, iCellSpeed has to disrupt ongoing traffic while taking actions to influence the default cell selection. We run experiments to measure the disruption time at both application (APP) and radio resource control (RRC) layers. We evaluate several actions of iCellSpeed: blocking one or two frequency bands, locking one specific band, and resetting mobile networks. In our test phones, blocking any band automatically blocks band 66. Fig. 4.14 shows blocking or locking band have similar disruption time at APP layer, which is slightly



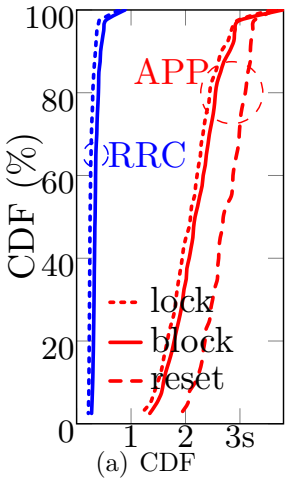
(a) At L17 (C1)

(b) At L15 (C1)

(c) At L18 (C2)

Figure 4.13.: Showcases of speed gains under different speed-not-ok rules.

less than resetting mobile network (2,694 ms). Locking disrupts RRC and APP for 248 ms and 2092 ms (median). Blocking has a little larger disruption time: 340 ms (RRC) and 2176 ms (APP). This matches with expectation because locking limits the spectrum bands to scan and has lower overhead. Resetting mobile network results in the longest disruption because it has to restart Radio Interface Layer (RIL) daemon [29]. We cannot measure its RRC disruption because it powers off the radio directly without releasing RRC connection. We would like to emphasize that the disruption at RRC is unavoidable as a solution compatible to the existing mechanism and infrastructure. The disruption from RRC to APP can be reduced with advances on mobile phone OSes and chipsets.



(a) CDF

	RRC (ms)
Lock	248
Block	340
Reset	-
	App (ms)
Lock	2092
Block	2176
Reset	2694

(b) median

Figure 4.14.: Disruption time.

4.1.7 Data Speed Gains Evaluation

Static tests (AT&T). We use two metrics to evaluate iCellSpeed.

Fig. 4.15 presents its speed gains at all test locations in AT&T. When a device action is taken in one run k, iCellSpeed's gain is calculated as Δ_k or Γ_k, for the absolute or relative gap between the average throughput in the default and customized runs. Note that the gain can be negative if the customized run is worse. We show the speed after the first round, without showing the final result after multiple rounds because iCellSpeed is eventually no worse than the default one. We clearly see that iCellSpeed significantly boosts data speed at many locations. It boosts data speed at *all* locations in terms of the 25-th percentile absolute gain. The 50-th percentile (median) gain is larger than 10 Mbps at 19 out of 24 locations (79.2%). Note that our test locations are *randomly* selected and the achieved gain is bounded by the missed performance potentials which vary across locations (Fig. 4.9(a)). This is why the absolute gain is not significant at all the locations. We see that the 75-th percentile gain is smaller than 10 Mbps at 4 out of 24 locations (here, 15, 16 in C1, 19 in C2 and 22 in C3). Hence, we use the relative gain Γ to deal with variance in the bound of the best achievable performance. Actually, at 15 out of 24 locations (62.5%), the

relative gain is larger than 100% in more than half of runs, up to 28.4× (at location L17).

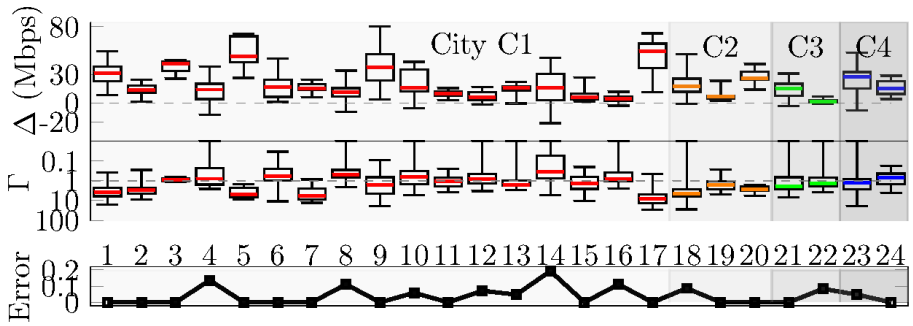


Figure 4.15.: Performance gains at all static locations for AT&T.

The second metric is the error rate at the first round, *i.e.*, the number of runs with negative gains over all device-customized runs. We observe that 14 out of 24 locations have zero error rate. The error rate is below 0.1 at 6 out of the rest 10 locations. This indicates that iCellSpeed reliably tames uncertainty and balances gains and risks well. iCellSpeed corrects the mistake finally while its intermediate performance may get hurt.

Other operators. We evaluate iCellSpeed for Verizon and T-Mobile. Rate throttling makes it hard for us to evaluate Sprint. We see similar results with significant gains in Verizon (plots omitted). iCellSpeed increases data speed by more than 30 Mbps at 57% of test locations (up to 91Mbps). It at least doubles speed at 87% of test locations (up to 27.3x). We do not often observe expected gains for T-Mobile. We observe that resetting mobile networks sometimes boosts data speeds but its impact is random, depending on which cell set being selected. We find that most missing performance in T-Mobile is from the use or combined use of band 66. However, our current implementation automatically disables band 66 when blocking any bad band. It is the implementation constraint that prevents us from increasing data speed through iCellSpeed. Gains are possible once this constraint is released.

Driving tests. Fig. 4.16 demonstrate iCellSpeed's speed gains using three examples on driving routes R1, R3 and R6. We use the upper plot (R1) to illustrate how iCellSpeed works. We drive the same route without (default) and with iCellSpeed. In the default run, the device suffers poor performance no more than 15 Mbps from the point of 270 m. It is connected to cell set {2425(363)-None-None} (270-464 m) and {66911(421)-None-None} (464-1047 m). For a run with iCellSpeed, it detects poor performance with {2425(363)-None-None} in the area of 450-480 m. Then iCellSpeed takes the action of blocking band 5 and 66 and thus moves to a good cell set. The average performance grows from 13 Mbps to 64 Mbps.

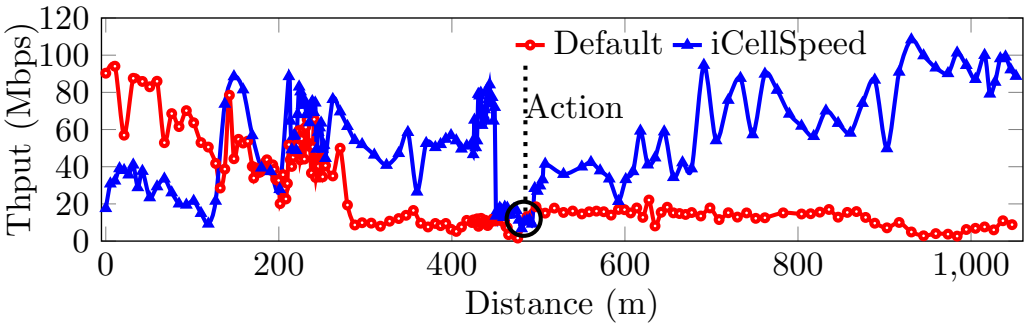
We define $\gamma = (\bar{\psi}_{\text{iCellSpeed}} - \bar{\psi}_{\text{default}})/\bar{\psi}_{\text{default}}$ to quantify the speed gain in a driving test, where $\bar{\psi}_{\star}$ is the average speed by $\star$ (default or iCellSpeed) over the same route segment after iCellSpeed is triggered (including 0Mbps during 2-second disruption). Note that we use the average speed over the same segment (distance), not over the same duration because the driving time changes at each run. We see 352% and 120% gains in other two examples (R3 and R6). Tab. 4.4 shows the average gain observed on six routes in C1 where iCellSpeed is in use.

Table 4.4. Speed gains by iCellSpeed on 6 driving routes in C1.

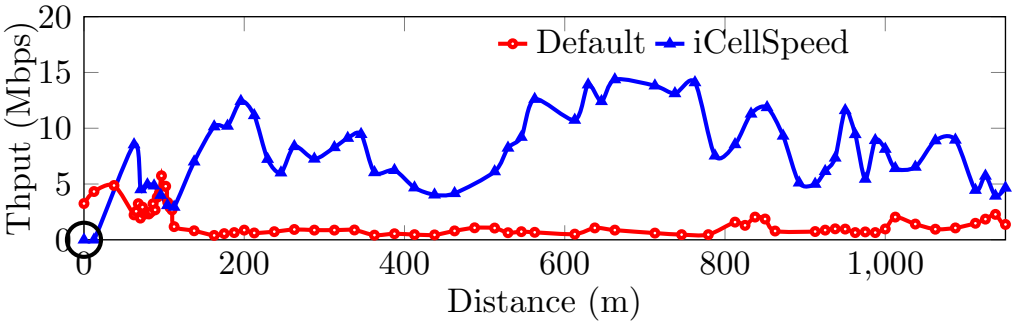
Route	R1	R2	R3	R4	R5	R6
Gain (γ)	275%	115%	181%	97%	135%	119%

Other applications. We test iCellSpeed with three popular applications with elephant flows: DASH video streaming, video conferencing and file downloading. Note that iCellSpeed is not applicable to mice flows only because it is not triggered in a short flow lifespan. Fig. 4.17 plots the results at 8 representative locations (with various data speeds and iCellSpeed gains) including 7 locations in C1 and 1 location (L18) in C2 using AT&T.

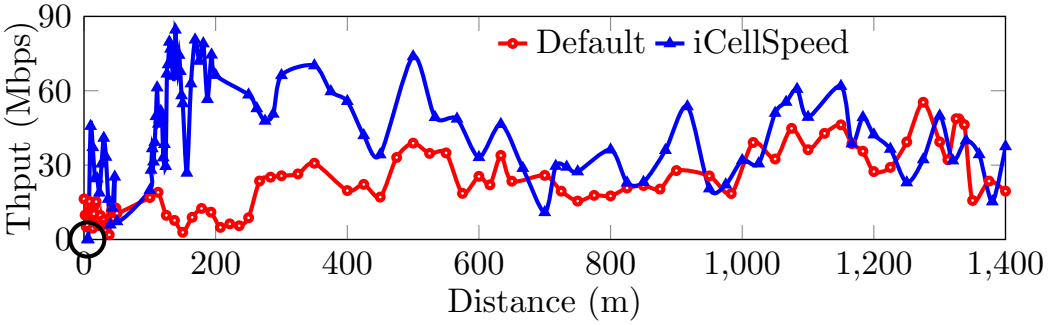
For video streaming, we test with a demo video at bitmovin [30] with constant bitrate encoding (CBR) and variable bitrate encoding (VBR). For video conferencing, we use Zoom to set up a call between a mobile phone and another device via WiFi



(a) R1



(b) R3



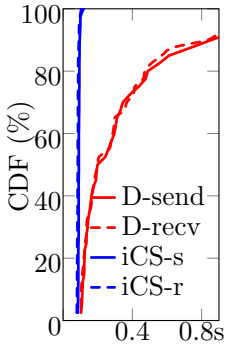
(c) R6

Figure 4.16.: Three examples of iCellSpeed's speed gains over distinct routes (R1, R3 and R6).

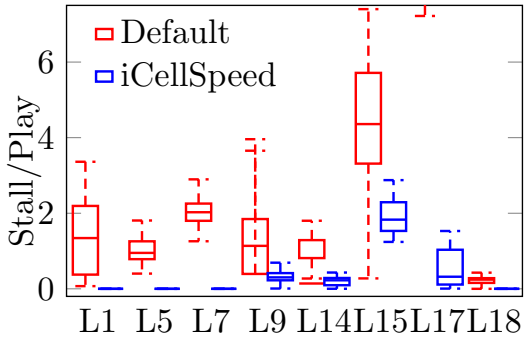
with abundant bandwidth. For file downloading, we consider two sizes: 25 MB and 100 MB. In this test, we see the worst data performance at L17 and thus choose L17 as an example to demonstrate that iCellSpeed has greatly boosted performance for all test applications in Fig. 4.17(a). Specifically, iCellSpeed helps VBR streaming to increase it dominant bitrate from 360p to 1080p (1K). It drops the stall/play ratio for

	VBR/ Main Res.	CBR Stall/Play 1K	CBR Stall/Play 4K	Zoom Delay (ms) Send	Zoom Delay (ms) Recv	Download Time (s) 25MB	Download Time (s) 100MB
D	360p	4.5	41.7	210	201	55.1	329.6
iCS	1080p	0	0.84	91	85	17.7	82.4
D/iCS	-	∞	**49.6**	**2.3**	**2.4**	**3.1**	**4.0**

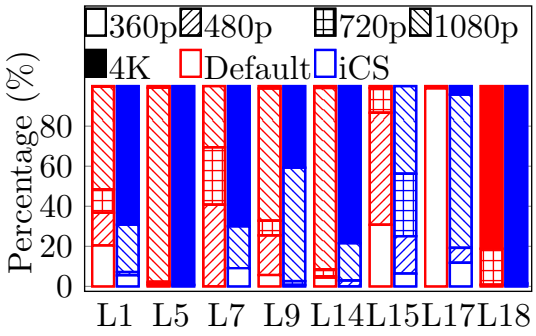
(a) Multiple applications at L17 (D: default, iCS: iCellSpeed)



(b) Zoom at L17



(c) 4K CBR video streaming



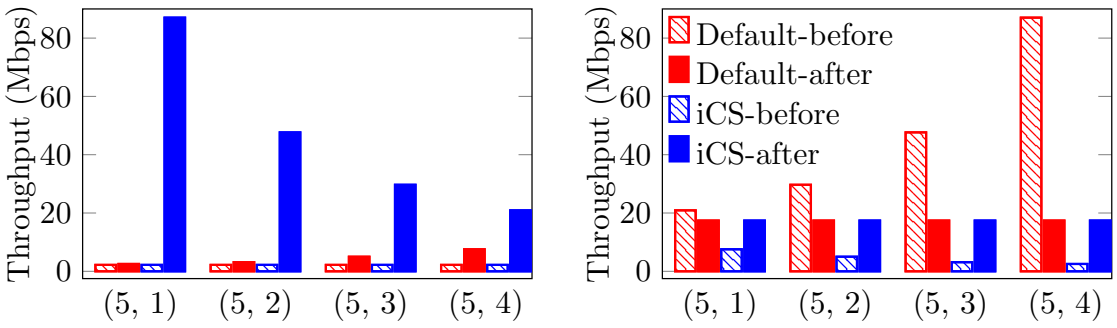
(d) VBR video streaming

Figure 4.17.: iCellSpeed's performance gains for test applications at multiple locations in AT&T.

CBR video streaming, from 4.5 to 0 for a 1K video (from 41.7 to 0.84 for a 4K video), making at least 1K video steaming affordable at L17. In Zoom video conferencing, it reduces send/uplink latency by 2.3x (downlink: 2.4x), from 210 ms (201 ms) to 91 ms (85 ms). It accelerates file downloading by 3.1x (25 MB) and 4x (100 MB). We also present CBR and VBR streaming results at test locations (Fig. 4.17(c) and 4.17(d)). We see that iCellSpeed enables 4K video streaming at 5 locations (L1, L5, L7, L14, L18). All these gains are attributed to increased speeds by iCellSpeed. We notice that 4K CBR is not an acceptable viewing option in these default runs, which all suffer from extremely high stall/play rate except at L18. We choose it as a stress test to see how badly it can be without iCellSpeed. Similar enhancements are observed in

the VBR tests, with more acceptable viewing experience by default, thanks to lower bitrates in use.

Multiple devices. We next evaluate how iCellSpeed performs in a multi-device scenario. We use (n, m) test phones where n is the total number of co-located phones and m is the number of phones running iCellSpeed ($m = 1, \cdots, n-1$). We consider in two scenarios: (a) all the phones are initially served by poor cells, (b) only those running iCellSpeed are initially served by poor cells and others are served by good cells (which actually run iCellSpeed to move to good cells first and then disable iCellSpeed). The first setting is to evaluate how iCellSpeed's effectiveness scales up, and the second is to evaluate its impact on those devices without iCellSpeed. Fig. 4.18 plots the results before and after iCellSpeed takes effects at L5 (AT&T) in both settings. The results at other locations are similar.



(a) n phones are poor

(b) m phones are poor

Figure 4.18.: iCellSpeed in (n, m) multi-device tests at L5 in AT&T.

We test with n up to 5 and present the results when $n = 5$, as similar results are observed at $n = 2, 3, 4$. We have three observations. First, we clearly see that iCellSpeed boosts performance of those iCellSpeed-enabled phones previously served by the poor cells in both scenarios. Second, the gain declines as m grows when all the phones are previously served by the poor cells. The data speed grows from 2.2Mbps to 87Mbps (39.5x), 47.7Mbps (21.7x), 29.7 Mbps (13.5x) and 20.9Mbps (9.5x) when m grows from 1 to 4. Third, this results in the dropped speed for those phones which

are previously served by the good cells. In Fig. 4.18(b), the phone without iCellSpeed gets its speed from 87 Mbps to 17.4Mbps when four phones switch to the good cells. This is expected because iCellSpeed is to exploit under-utilized capabilities, but not to increase raw capabilities. Once more devices move to the good cell(s), they compete resources and thus the obtained share drops. We observe significant enhancements in our real-world tests. This implies that the network capabilities are often under-utilized in presence of a number of real user phones beyond our control.

Energy overhead. We further assess iCellSpeed's energy consumption in two scenarios where the device is idle (with iCellSpeed being in the background but not triggered) or active with heavy traffic (here, YouTube, with iCellSpeed being triggered). We test it with a Pixel 2 device (2700 mAh) and Tab. 4.5 shows the results.

Table 4.5. Battery usage.

	Case 1: Idle							**Case 2: Active**	
	Phone idle	Google Play	Android System	Mobile Network	Screen	UI	**iCS**	YouTube	**iCS**
Usage	12%	6%	6%	5%	2%	1%	$<$ **1%**	45%	**5%**

In the idle setting, we see that extra energy consumed by iCellSpeed is negligible, compared to built-in services and components running in the background. In the active setting, we keep playing an YouTube video and find that iCellSpeed consumes 5% extra energy when YouTube uses 45% energy. The energy overhead is mainly used to monitor GPS, cellular signaling messages and system level throughput, which can be cut with energy-efficient monitoring.

4.1.8 Other possibilities and remaining issues

We discuss other possibilities and remaining issues.

Are larger gains possible? Absolutely yes. The gain would be larger when more device power is available, for example, when the device can directly lock the desired cells instead of blocking some bands to indirectly influence cell selection. The gain

would be larger if the device supports swift band switching, without the 2-second disruption which is constrained by current practice of device chipsets and OSes.

Should we do it on-device only? Absolutely no. Instead, we argue that network is a better place to solve this problem. The gain is much larger or even reaches its full potential when the changes are allowed at the network side. For instance, it performs a global performance-driven optimization, which is not myopia constrained by partial observations at runtime. In this work, our aim is to offer a working solution even with modest gains and demonstrate feasibility to increase speed missed by current cell selection practice.

Does it hurt other devices and/or network? Maybe. The operators may intend to sacrifice user experience with rational, e.g, load balancing for network-side optimization, or throttling data for those without premium plans. We do not argue that they have to select cells that offer the best performance to mobile users. Our goal is to pursue better data performance which is sometimes unnecessarily missed in reality. We believe that boosting performance with no need of changing physical infrastructure is aligned with the interests of both operators and users in some real-wold circumstances. If they do not match, the network always holds the right and final power to decide what to serve the device. We notice that iCellSpeed may benefit the device but at the cost of performance degradation of other devices (Fig. 4.18(b)). This is because the default selection is unfair and should take the blame at the first place. In theory, iCellSpeed may oscillate when a large number of iCellSpeed-enabled devices are synced to intervene cell selection and impact each other. It is not observed in practice as iCellSpeed removes this cell choice once it underperforms. In the worst case, it conservatively goes back to the default selection.

Will the problem disappear in 5G? It will not go away as 5G proceeds. The identified issue of missed performance potentials conceptually exists in 5G that still takes radio signal quality, not the resulted performance into account [31]. This issue is likely even worse in 5G with much denser deployment, more spectrum choices, and bigger performance gaps contributed by advanced technologies (*e.g.*, 10Gbps vs tens

of Mbps). It was reported that the device failed to get 5G where 5G was available in an early 5G measurement [32].

4.2 Another Case For Multiple-carrier Network Access

In addition to infrastructure upgrades from carriers, a promising alternative to ensure complete coverage or highest access quality at any place and anytime is to leverage multiple carrier networks at the end device. In reality, most regions are covered by several carriers (say, Verizon, T-Mobile, Sprint, and AT&T in the US). With multi-carrier access, the device may select the best carrier over time and improve its overall access quality.

A cellular carrier deploys and operates its mobile network (called public land mobile network or PLMN) to offer services to its subscribers. Each PLMN has many cells across geographical areas. Each location is covered by multiple cells within one PLMN and across several PLMNs (*e.g.*, Verizon, AT&T, T-Mobile, Sprint).

When the home PLMN cannot serve its subscribers (*e.g.*, in a foreign country), the device may roam to other carriers (visiting networks). This is realized through the PLMN selection procedure between carriers [33], which is a mandatory function for all commodity phones. It supports both automatic (based on a pre-defined PLMN priority list) and manual modes. As shown in the right plot of Figure 4.19, once triggered by certain events (*e.g.*, no home PLMN service), PLMN selection should first scan the available carriers, and then choose one based on the pre-defined criteria (*e.g.*, preference) or the user manual operation. If the device decides to switch, it will deregister from the current carrier network and then register to a new one. In this process, network access may be temporarily unavailable. This is acceptable since inter-carrier switch is assumed to be infrequent, thus having limited impacts.

Recent industrial efforts aim at providing mobile device access to multiple carriers with a single SIM/e-SIM card. The exciting Google Fi [34] has taken the lead to provide 3G/4G multi-carrier access in practice. Other similar efforts through universal

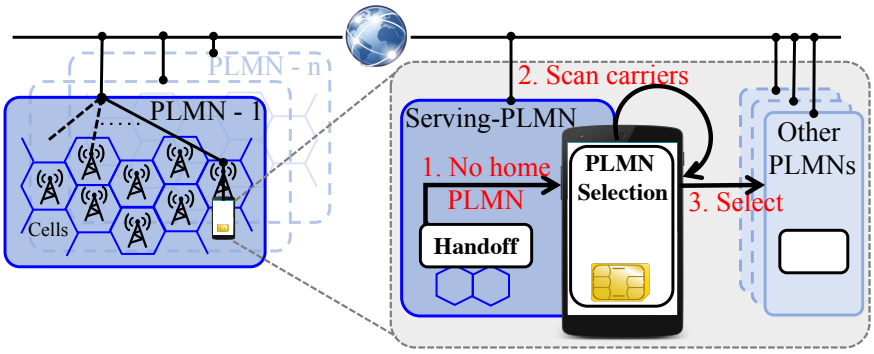


Figure 4.19.: Multi-carrier network access and inter-carrier switch via PLMN selection.

SIM card include Apple SIM [35] and Samsung e-SIM [36]. With the SIM/e-SIM card, the device can access multiple cellular carriers (*e.g.*, T-Mobile and Sprint in Google Project Fi). Given only one cellular interface, the device uses one carrier at a time.

However, our empirical study shows that, the full benefits of multi-carrier access can be constrained by design. We examine Google Fi over two carriers (T-Mobile and Sprint), and discover three issues, all of which are independent of its excellent implementations: (P1) The anticipated switch is never triggered even when the serving carrier's coverage is pretty weak, (P2) The switch takes rather long time (tens of seconds or minutes) and prolongs service unavailability, and (P3) the device fails to choose the high-quality network (*e.g.*, selecting 3G with weaker coverage rather than 4G with stronger coverage).

Here, we seek to devise a solution that works with the current 3G/4G network, in line with the ongoing industrial efforts. Specifically, we address the following problem: *Can we leverage low-level cellular information and mechanisms at the device to further improve multi-carrier access?* Our study yields a positive answer. We propose iCellular, a client-side service to let mobile devices customize their own cellular network access. Complementing the design of Google Fi, iCellular further leverages low-level, runtime cellular information at the device during its carrier selection. iCellular is built on top of current 3G/4G mechanisms at the device, but applies cross-layer adaptations to ensure responsive multi-carrier access with minimal disruption. To facilitate

the device to make proper decisions, iCellular exploits online learning to predict the performance of heterogeneous carriers, and provides built-in strategies for better usability. It further safeguards access decisions with fault prevention techniques. We implement iCellular on commodity phone models (Nexus 6 and Nexus 6P) and assess its performance with Google Fi. Our evaluation shows that, iCellular can achieve 3.74x throughput improvement and 1.9x latency reduction on average by selecting the best mobile carrier. Meanwhile, iCellular has negligible impacts on the device's data service and OS resource utilization (less than 2% CPU usage), approximates the lower bounds of responsiveness and switch disruption, and shields its selection strategies from decision faults.

4.2.1 Multi-carrier Access: Promises And Issues

We run experiments to quantify the benefits of multi-carrier access, and identify the downsides of the today's efforts. The identified limitations are independent of implementations, but rooted in the 3G/4G design.

1) Methodology. We conduct both controlled experiments and a one-month user study using two Nexus 6 phones with Google Fi, which was released in May 2015. Google Fi provides access to two U.S. carriers (T-Mobile and Sprint) at this time. It develops an automatic carrier selection on commodity phones using a proprietary mechanism. Unfortunately, details of its switching algorithm have not been published. We contacted Google Fi team and learned that this algorithm aims at optimizing consumer experience, and considers network performance, battery usage and data activity during selection. We further inferred its decision and execution strategies from our experiments.

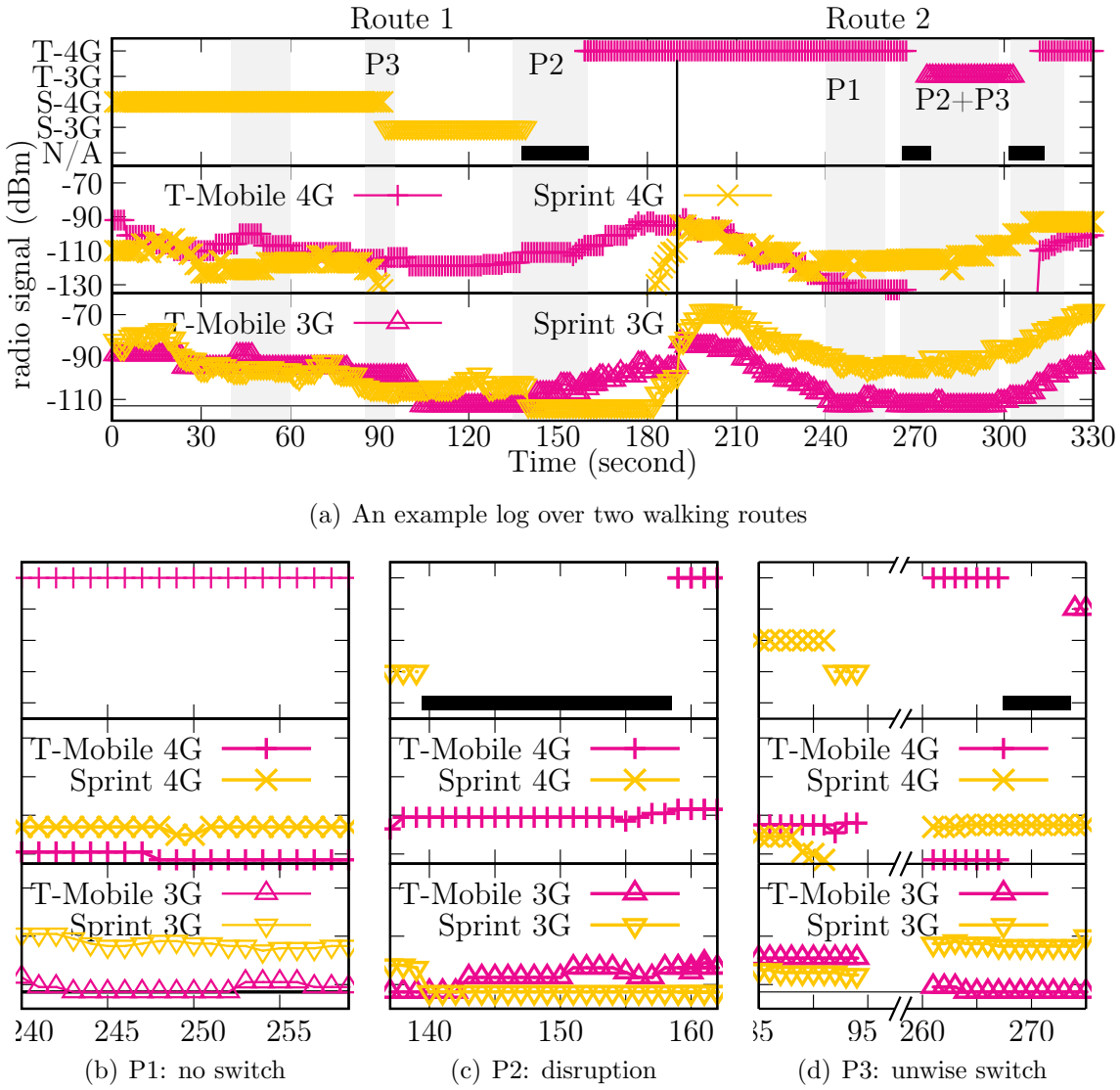
In each controlled test, we use a Nexus 6 phone with a Google Fi SIM card, and test with Google Fi's automatic carrier selection mode. We walk along two routes within the campus buildings at UCLA and OSU at the idle mode (no data/voice, screen off). We walk slowly (< 1 m/s) and record the serving carrier ("T" for T-Mobile, "S"

for Sprint) and its network type (4G or 3G) per second. Meanwhile, we carry other accompanying phones to record the radio signal strength of each access option (T-4G, T-3G, S-4G, S-3G). We run each test 10 times and similar results are consistently observed in all the tests. In the user study (07/31/15 to 09/02/15), we use the Google Fi-enabled phone as usual and collect background device and cellular events with MobileInsight, an in-phone cellular monitoring tool [7]. We have collected 4.9 GB logs with MobileInsight in total, with 274,351 messages from radio resource control (RRC), 16,470 messages from mobility management (MM), and 5,365 messages from session management (SM). We next present the results from the controlled experiments as motivating examples. The user study to be described later further confirms that these issues are common in practice.

2) Motivating Examples Merits of multi-carrier access. We first verify that exploiting multiple carriers is indeed beneficial to service availability and access quality. Figure 4.20(a) shows the results from the controlled experiments over two routes. On the first route [0s,190s), Sprint gradually becomes weaker and then fades away, but its dead zone is covered by T-Mobile. On the second route [190s, 330s], in contrast, Sprint offers stronger coverage, even at locations with extremely weak coverages from T-Mobile. Multi-carrier access indeed helps to enhance network service availability by boosting radio coverage. For example, in [160s, 180s], the phone switches to T-Mobile and retains its radio access while Sprint is not available. Moreover, we confirm that it further improves data access throughput and user experiences. The Google Fi indeed offers a major step forward on mobile Internet access.

Our examples further reveal three issues, which demonstrate that the benefits of multi-carrier access have not been fully achieved.

P1. No anticipated inter-carrier switch. It is desirable for the device to migrate to another available carrier network for better access quality, when the device perceives degraded quality from its current, serving carrier. However, our experiments show that, the device often gets stuck in one carrier network, and misses the better network access (*e.g.*, during [40s, 60s] and [240s, 260s] of Figure 4.20). As shown in



(a) An example log over two walking routes

(b) P1: no switch

(c) P2: disruption

(d) P3: unwise switch

Figure 4.20.: An example log for serving carriers and networks and three problematic instances through Google Fi.

Figure 4.20(b), T-Mobile experiences extremely weak radio coverage (< -130 dBm in 4G and < -110 dBm in 3G), but the phone never makes any attempt to move to Sprint, regardless of how strong Sprint's radio signal is. As a result, the device fails to improve its access quality. Moreover, we find that the expected switch often occurs until its access to the original carrier (here, T-Mobile) is lost. This is rooted in the fact that the inter-carrier switch is triggered when the serving carrier fails.

Therefore, the device becomes out of service in this scenario, although better carrier access remains available.

P2. Long switch time and service disruption. Even when inter-carrier switch is eventually triggered, it may disrupt access for tens of seconds or even several minutes. In the example of Figure 4.20(c), the phone starts Sprint→T-Mobile roaming at the 140th second, but it takes 17.3s to gain access to T-Mobile 4G. This duration is much longer than the typical handoff latency (possibly several seconds [37]). It is likely to halt or even abort any ongoing data service. We look into the event logs (Figure 4.21) to examine why the switch is slow. It turns out that, most of the switch time is wasted on an *exhaustive* scanning of all possible cells, including nearby cells from AT&T and Verizon. In this example, it spends 14.7 s on radio-band scanning and 2.6 s on completing the registration (attachment) to the new carrier (here, T-Mobile). Note that, such heavy scanning overhead is not incurred by any implementation glitch. Instead, it is rooted in the Google Fi's design, which selects a new carrier network only after an exhaustive scanning process. In this work, we want to show that such large latency is unnecessary. It can be reduced without compromising inter-carrier selection.

Time	Event	
11:19:57.414	**Out-of-service. Start network search**	RF band scanning: 14.7s
11:19:57.628	Scanning AT&T 4G cell 1, unavailable	
11:19:57.748	Scanning AT&T 4G cell 2, unavailable	
…	…	
11:20:11.788	Scanning Verizon 4G cell 1, unavailable	
…	…	
11:20:12.188	**Scanning T-Mobile 4G cell 1, available**	Network registration: 2.6s
11:20:12.771	Attach request (to T-Mobile 4G)	
11:20:14.788	Attach accept	

Figure 4.21.: Event logs during P2 of Fig. 4.20(c).

P3. Unwise decision and unnecessary performance degradation. Our next finding is that, the device fails to migrate to the better choice, thus unable to enjoy the full benefits of multi-carrier access. The phone often moves to 3G offered

by the same carrier, rather than the 4G network from the other carrier that yields higher speed. Figure 4.20(d) illustrates two such instances. After entering an area without Sprint 4G at the 91st second, the device switches to Sprint 3G, despite stronger radio signals from T-Mobile 4G. This indicates that the intra-carrier handoff is preferred over the inter-carrier switch in practice. Unfortunately, such a preference choice prevents the inter-carrier switch from taking effect. Even worse, obstacles still remain even when the network access to the original carrier has been shortly disrupted. For instance, during [267s, 273s], the original carrier (T-Mobile 3G) is still chosen. In this case, T-Mobile 4G and 3G networks almost have no coverage. In short, the device acts as a single-carrier phone in most cases, even with the multi-carrier access capability. Inter-carrier switch is not triggered as expected.

3) Insights. The above examples also shed lights on how to solve the three problems. The key is to leverage low-level cellular information and mechanisms at the device when selecting access from multiple carriers.

Specifically, performing the anticipated switch (P1) states that, the device performs inter-carrier switch upon detecting a better carrier, even when the serving carrier is still available. This further requires the device to learn all available carriers and their quality at runtime. Note that such information can be obtained from the low-level cellular events. However, the default operation on commodity phones will not do so. Moreover, the naive approach of forcing the phone to proactively scan other carriers at any time may lead to temporary disconnection from the current carrier network. We elaborate on how we address these issues later.

To reduce the switch time (P2), the device should refrain from exhaustive search of all carriers at all times. This requires the device to perform fine-grained control on which carriers should be scanned. It can be done by configuring the low-level mechanism for monitoring.

To make a wise selection decision (P3), the device should treat all intra-carrier handoffs and inter-carrier switches equally, and select the best carrier network. This

requires the device to directly initiate the inter-carrier switch when needed. This also calls for leveraging the low-level cellular mechanism.

In summary, low-level domain knowledge can be exploited to effectively address all three issues. However, the default operation mode on commodity phones does not expose such fine-grained cellular information and mechanisms to higher layers. The reason is that, the 3G/4G network follows the design paradigm of "smart core, dumb end" with the single-carrier usage scenario in mind. The end device does not need to exploit such information when selecting its carrier access. Since such low-level, cellular-specific domain knowledge is not available for the default operation mode, it might be the reason why Google Fi has not explored this direction in its current design.

4.2.2 iCellular Design And Implementation

We now present iCellular, which explores an alternative dimension to improve multi-carrier access. iCellular complements the design of Google Fi by leveraging low-level cellular information and mechanisms. It seeks to further empower the end device to have more control on its carrier selection, while addressing the issues in §4.2.1.

For incremental deployability, iCellular is built on top of the PLMN selection [5, 33], a standardized mechanism mandatory on all phones. Note that, however, the basic PLMN selection suffers from similar issues in §4.2.1: migrating to other carriers is not preferred unless the home carrier fails (P1), the exhaustive scanning (P2) and the preferable intra-carrier handoffs (P3) are still in use. The reason is that, the default PLMN selection scheme is designed under the premise of single-carrier access. While roaming to other carriers is allowed, it is not preferred by the home carrier unless it fails to offer network access to its subscribers. So the basic PLMN selection has the following features: (1) *Passive triggering/monitoring:* When being served by one carrier, the device should not monitor other carriers or trigger the selection until

the current one fails (*i.e.*, out of coverage), (2) *Network-controlled selection:* The device should select the new carrier based on the preferences pre-defined by the home carrier and stored in the SIM card, (3) *Hard switch:* The device should deregister from the old carrier first, and then register to the new one. We thus need to adapt the PLMN selection scheme to the multi-carrier context by using low-level cellular events.

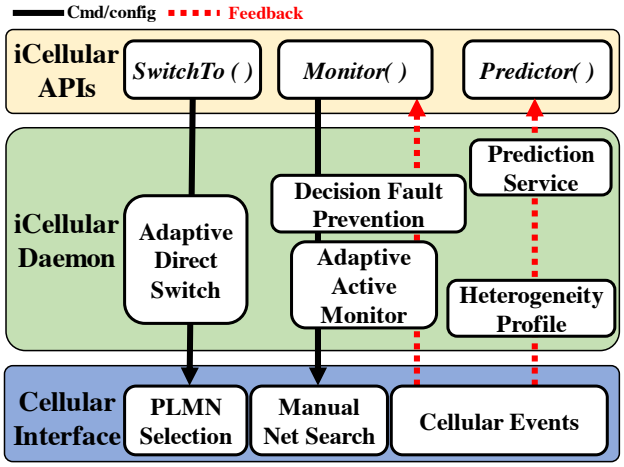


Figure 4.22.: iCellular system architecture.

Figure 4.22 illustrates an overview of iCellular. In brief, iCellular systematically enhances the devices' role in every step of inter-carrier switch with runtime cellular information, spanning triggering/monitoring, decision making and switch execution. To be incrementally deployable on commodity phones, we build iCellular on top of the existing mechanisms from the phone's cellular interface [38]. We exploit the freedom given by the standards, which allow devices to tune configurations and operations to some extent. To ensure responsiveness and minimal disruption, iCellular applies cross-layer adaptations over existing mechanisms. To facilitate the devices to make wise decisions, iCellular offers cross-layer online learning service to predict network performance, and protects devices from decision faults. To enable adaptation, prediction and decision fault prevention, iCellular incorporates realtime feedbacks extracted from low-level cellular events. Different from approaches using additional diagnosis

engine (*e.g.*, QXDM [39]) or software-defined radio (*e.g.*, LTEye [40]), we devise an in-phone mechanism to collect realtime cellular events, cellular events are summarized in Table 4.6). These components are designed to be scalable, without incurring heavy signaling overhead to both the device and the network.

Table 4.6. Cellular events used in iCellular.

Function	Method	Cellular Events	Type
Active monitor	Disruption avoidance	Paging	Meas
		Paging cycle	Config
	Minimal search	Radio meas	Meas
		RRC SIB 1	Config
Prediction service	QoS profile	EPS/PDP setup	Config
	Radio profile	RRC reconfig	Config
Decision fault prevention	Access control	RRC SIB1	Config
	Interplay with net mobility	Cell reselection in RRC SIB 3-8	Config
	Function completeness	GMM/EMM location update	Config

paragraphb1) Adaptive Monitoring. To enable device-initiated selection, the first task is to gather runtime information on available carrier networks. This is done through *active monitoring*. It allows a device to scan other carriers even while being served by one. This would prevent the device from missing a better carrier network (P1 and P3 in §4.2.1). For this purpose, the only viable mechanism on commodity phones is the `manual network search` [33]. It was designed to let a device manually scan all available carriers. Once initiated, the device scans neighbor carriers' frequency bands, extracts the network status from the broadcasted system information block, and measures their radio quality. No extra signaling overhead is incurred, since the active monitoring approach does not activate signaling exchanges between the device and the network. To be incrementally deployable, we decide to realize active monitoring on top of the manual network search.

Note that naive manual search does not satisfy properties of minimal-disruption and responsiveness. First, scanning neighbor carriers may disrupt the network service. The device has to re-synchronize to other carriers' frequency bands, during which it

cannot exchange traffic with the current carrier. Second, it is *exhaustive* to all carriers by design. Even if the device is not interested in certain carriers (*e.g.*, no roaming contract), this function would still scan them, thus delaying the device's decision and wasting more power. The challenge is that, both issues cannot be directly addressed with application-level information only. iCellular thus devises cross-layer adaptions for both issues.

Disruption avoidance. To minimize disruptions on ongoing services, iCellular schedules scanning events only when the device has no application traffic delivery. This requires iCellular to monitor the uplink and downlink traffic actvities. While the uplink one can be directly known from the device itself, the status for downlink traffic is hard to predict. Traffic may arrive while the device has re-synchronized to other carriers' cells. If so, its reception could be delayed or even lost.

iCellular prevents this by using the low-level cellular event feedback. We observe that in the 3G/4G network, the downlink data reception is regulated by the periodical paging cycle (*e.g.*, discontinuous reception in 4G [16, 41]). To save power, the 3G/4G base station assigns inactivity timers for the device. The device periodically wakes up from the sleep mode, monitors the paging channel to check downlink data availability, and moves to the sleep mode again if no traffic is coming. iCellular obtains this cycle configuration from the radio resource control (RRC) messages, and schedules its scanning operations only during the sleep mode. Figure 4.23 shows our one-month logs of 4G per-cell search time at a mobile device with Google Fi. It shows that, 79.2% of cells can be scanned in less than one paging cycle. Others need more cycles to complete the scanning. With this design, no paging event is interrupted by monitoring.

One valid concern is that, the monitoring results may become obsolete due to continuous data transmissions, thus leading to wrong decisions. This is unlikely to happen in practice for two reasons. First, most traffic tends to be bursty, which leaves sufficient idle period for background monitoring. Second, network performance tends to vary smoothly, and stale monitoring results do not affect the final selection decision.

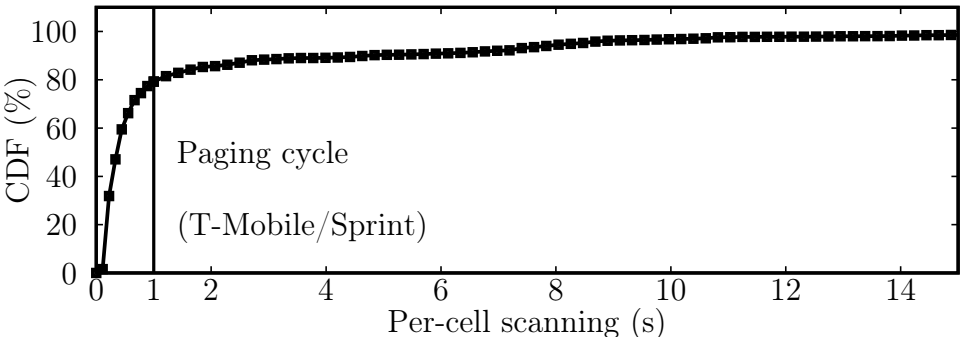


Figure 4.23.: Cell scan time with Google Fi during one month.

Furthermore, iCellular compares the elapsed time between the decision making and the measurement. Obsolete measurements outside the time window (say, 1 minute) will not be used.

Minimal search. Instead of exhausting all carrier networks, iCellular scales the monitoring by restricting the manual search only to those specified by the device. To realize this idea, the practical issue is that no such option is available in the manual network search mechanism. We thus leverage adaptation of the PLMN preference. Given the list of carrier networks of interests, iCellular configures the cellular interface to let the manual network search scan these carriers first. This is achieved by assigning them with highest PLMN preferences. During the manual search, iCellular listens to the cellular events to see which carrier is being scanned. These events include the per-cell radio quality measurements, and its system information block with PLMN identifiers. Once iCellular detects that the device has finished scanning of the device-specified carriers, it terminates the manual network search function.

Monitoring-decision parallelism. Sometimes there is no need to complete all the monitoring to determine the target carrier network. For example, if the user prefers 4G, it can decide to switch whenever a good 4G is reported, without waiting for 3G results. To support this, iCellular allows devices to make decisions with partial results, thus further accelerating the process. Instead of waiting for all scanning results, iCellular triggers the decision callback whenever new results are available.

2) Direct Inter-carrier Switch. iCellular aims at reducing the disruption time incurred by inter-carrier switching as much as it can. We find that, there is enough room for this because *most service disruption time is caused by frequency band scanning*. With the active monitoring function, iCellular does not need to scan the carrier networks during switch. Specifically, given a target carrier network, iCellular makes a direct switch by configuring the target carrier with highest PLMN preference. It then triggers a manual PLMN selection to the target carrier network. This way, the device would directly switch to the target without unnecessary scanning.

We next show how iCellular approximates to the lower bound of the switch time. In cellular networks, switching to another network requires at least de-registration from the old network (detach), and registration to the new network (attach). According to [42], detach time is negligible, since the device can detach directly without interactions to the old carrier network. So the minimal disruption time in switch is roughly equal to attach time, *i.e.*, $T_{switch,min} \approx T_{attach}$. For iCellular, no extra attempts to other carrier networks are made. Since it is on top of the PLMN selection, the scanning of the target carrier still remains. Therefore, the switch time is

$$T_{switch,iCellular} = n_t T_t + T_{attach} = n_t T_t + T_{switch,min} \tag{4.6}$$

where n_t and T_t are the cell count and per-cell scanning time for the target carrier network, respectively. Compared with attach time, this extra overhead is usually negligible in practice. Figure 4.24 verifies this with our one-month background monitoring results in Google Fi. It shows that, iCellular indeed approximates the lower bound, despite this minor overhead.

3) Prediction for Heterogeneous Carriers. To decide which carrier network to switch to, the device may gather performance information on each carrier network. Ideally, the device needs to measure every available carrier network's current performance (*e.g.*, latency or throughput) and make decisions. Unfortunately, this is

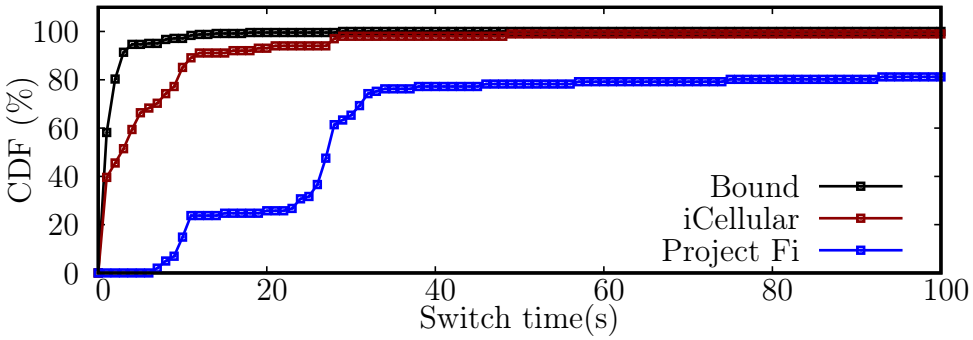


Figure 4.24.: Switch time comparison between Google Fi, iCellular and lower bound.

deemed impossible. The device can only measure the serving network's performance, and other candidates' performances cannot be measured without registration.

Given this fact, iCellular decides to assist the device to predict each carrier's performance. Our predication is based on the regression tree algorithm [43]. It models the network/application performance (y) as a function of a feature vector (x_1, x_2), where x_1 is runtime radio measurement and x_2 is carrier network profiles (elaborated below). The model is established using a pre-stored tree at bootstrap and then recursively updated with new online samples. Note that radio measurement alone is insufficient to predict performance, because different carriers may apply heterogeneous radio technologies and resource configurations. Our prediction works as follows.

Prediction metric (y). This metric is used to rank the performances of all available networks. We explore both network-level (link throughput, radio latency) and application-level ones (*e.g.*, web loading latency, video suspension time). They are obtained from both network and application events. We want to point out that the app-specific metric often leads to the same selection decision (see the evaluation). This is because the performance characteristics of a carrier network tend to have consistent impacts on all applications.

Training sample collection. The training sample (x, y) for a network is collected in the background, without interrupting the device's normal usage. A new training sample is collected when a new observation of the performance metric y is

generated (*e.g.*, throughput from physical layer, loading time for Web-page download, latency per second for VoIP). In the meantime, radio measurement and network profiles for the serving network are recorded as $x = (x_1, x_2)$. For the radio quality x_1, iCellular extracts the serving network's RSRP (if 4G) or RSCP (if 3G) from the runtime active monitor. For the network profile, iCellular currently collects two types (Table 4.7): (1) QoS profile from the data bearer context in session management, which includes the delay class and peak/maximum throughput, (2) radio parameters from the RRC configuration message, which includes the physical and MAC layer configurations. Note that the device cannot gain these profiles at runtime without registration to the carrier network of interest. To address this issue, we observe that network profiles are quite predictable. This is validated by our 1-month user study. Table 4.7 lists the predictability of some parameters from this log. For each parameter, we choose the one with the highest probability, and shows its occurrence probability. Note that, most QoS and radio configurations are invariant of time and location. The reason is that, the carriers tend to apply well-tested operation rules (*e.g.*, link adaptation and scheduling), with minor tunings to each base station/controller. As a result, we only store a set of unique values, and reuse it for all the applicable samples until changes are found.

Table 4.7. Heterogeneous cellular network profiles.

Profile		**Sprint**		**T-Mobile**	
		Value	**Prob**	**Value**	**Prob**
QoS	Traffic class	Background	100%	Interactive	97.5%
	Delay class	4 (best effort)	100%	1	100%
	Max dlink rate	200Mbps	100%	256Mbps	100%
	Max ulink rate	200Mbps	100%	44Mbps	100%
Radio	Duplex type	TDD	88.3%	FDD	100%
	Paging cycle	100–200ms	81.5%	100ms	99.4%
	Handoff priority	2/3/6	100%	2/3/6	100%

Online predication and training. iCellular uses an online regression tree algorithm [43] as its predictor. The predictor is represented as a tree, with each interior node as a test condition over x (radio measurements and profile fields). Each decision

is made upon the arrival of the feature vector x. It estimates the per-network metric y and selects the one with the highest rank.

iCellular updates the predictor's decision tree in the online fashion when a new sample arrives. At the bootstrap phase, it pre-stores a regression tree based on an offline training as the basis. Given a new sample (x, y), iCellular first determines whether a predictor update is needed. It runs the existing predictor over the heterogeneity information and runtime radio measurements, and obtains an estimated metric $y\prime$. If $|y - y\prime| = \min_{z \in leaf} |y - z|$, which implies the current sample fits well with the existing model, no update is needed. Otherwise, the predictor is updated as follows. Given the new sample and the existing tree, iCellular searches a new field (measurement or profile) that best splits the samples by minimizing the impurities in the two children nodes (based on the least-square criterion). Given this new split, we create a new pair of leaves for this new field, and completes the update of the prediction tree. Note that iCellular responds to new changes, and does not need to permanently store all training samples. This way, iCellular is scalable in storage and computation.

4) Decision Fault Prevention. Letting a device customize its access strategy can be a double-edged sword. With improper strategies, the device may make faulty switch decisions and cause unexpected service disruption. Figure 4.25 shows three categories of failures caused by decision faults, all of which can only be detected with low-level cellular information:

Failure 1: No network access. Certain networks may be temporarily inaccessible. For example, our user study reports that, a Sprint 4G base station experiences a 10-min maintenance, during which access is denied.

Failure 2: No voice service. In some scenarios, the target carrier network cannot provide complete voice services. Figure 4.26 shows an instance from our user study. T-Mobile provides its voice service using circuit-switched-fall-back (CSFB), which moves the device to 3G for the voice call. However, there exist areas not covered by T-Mobile 3G (*e.g.*, signal strength lower than -95 dBm according to [3]).

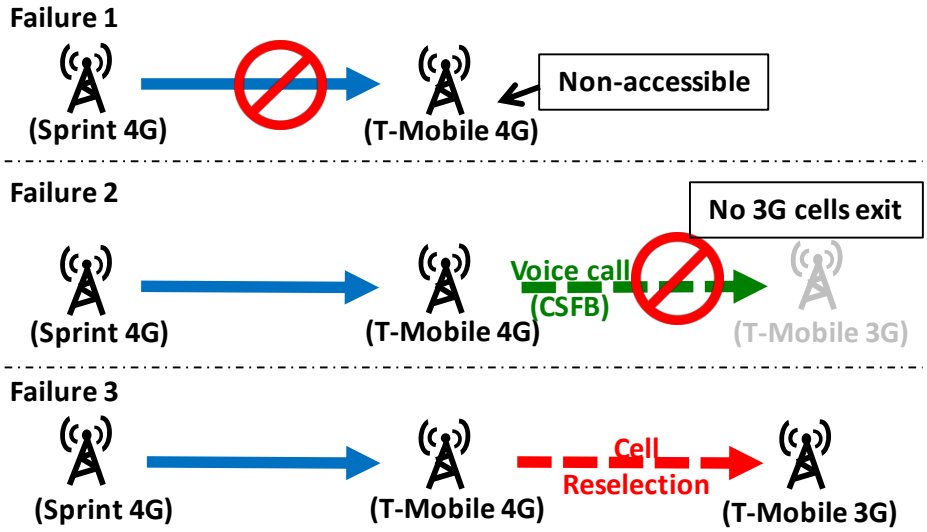


Figure 4.25.: Three types of improper switch decisions.

In this scenario, the user in Sprint 4G should not switch to T-Mobile 4G, which cannot support voice calls without the 3G infrastructure.

Failure 3: Unexpected low-speed data service. The user selection may not be honored by the individual carrier's handoff rules. Figure 4.27 reports an instance from our user study. The user under Sprint 4G may decide to switch to one T-Mobile 4G. However, under the same condition, T-Mobile's mobility rules (*e.g.*, cell re-selection [5]) would switch its 4G users to its 3G. In this case, the user's decision to T-Mobile 4G is improper, because the target network (T-Mobile 3G) is not preferred, and this switch incurs unnecessary disruptions.

To prevent decision faults, iCellular chooses to safeguard the device's decisions from those faulty ones. It checks whether each carrier network has any of the above problems, and excludes such carriers from the monitoring results. This prevents the device from switching to these carrier networks. To this end, iCellular first profiles each carrier's low-level access-control list from the RRC system-info-block message [16], data/voice preference configuration from registration/location update messages [42], and the network-side mobility rules from the RRC configuration message [5, 16]. At runtime, for each candidate carrier, it checks if it is in the forbidden list (Failure

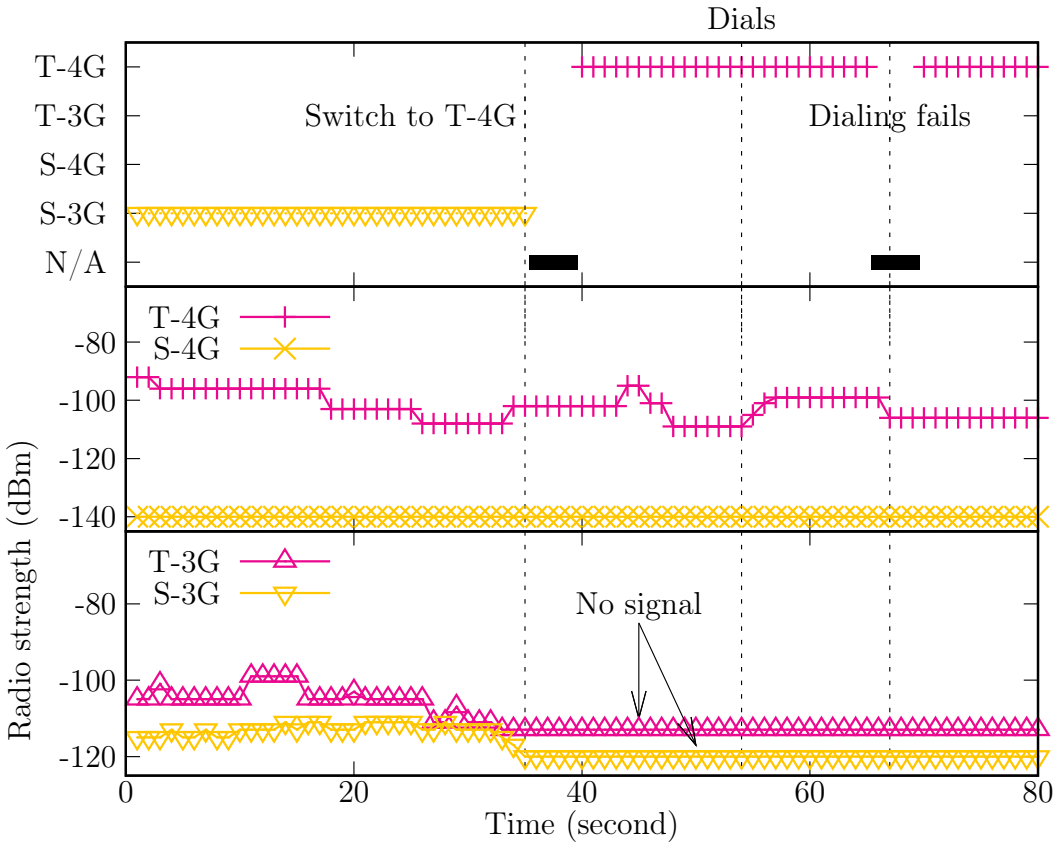


Figure 4.26.: Switch to a network with no voice support.

Time	Event
17:49:07.520	**Deregister from Sprint 4G**
17:49:13.433	Scanning T-Mobile 4G cell 1, available
17:49:13.508	**Cell reselection config in SIB6: switch to 3G when $RSRP_{4G}$<-120dBm**
17:49:15.142	Attach accept
17:49:20.106	**$RSRP_{4G}$= -122dBm**
17:49:21.326	**Cell reselection to T-Mobile 3G**

Figure 4.27.: Interplay between user and network's mobility.

1), has no voice service with satisfactory 3G radio quality (Failure 2), or has satisfied

mobility rules for further switch (Failure 3). If any condition is satisfied, it would be removed from the monitoring list.

5) Cellular Events Collection. As shown before, iCellular relies on low-level cellular events to perform cross-layer adaptations over the existing mechanisms, predict the network performance, and avoid possible switch faults. The cellular events include the signaling messages exchanged between the device and the network, and radio quality/load measurements. Table 4.6 summarizes the events required by iCellular. Note that some events (*e.g.*, paging) should be extracted at realtime for feedbacks. Unfortunately, obtaining realtime cellular events on commodity phones is not readily available as these events are not exposed to mobile OS or applications. There exist commercial tools (*e.g.*, QXDM [39]) and research projects (*e.g.*, LTEye [40]) to extract them. However, they require an external platform (*e.g.*, laptop or a special hardware (USRP)) to connect to the mobile device, which limits the device's flexible movement and its applicability. They cannot meet iCellular's realtime requirements. To this end, we use the in-phone solution MobileInsight [7] and finally expose them to iCellular. This solution can be deployed on commodity phones without hardware changes.

6) Implementation We have implemented iCellular on Motorola Nexus 6 and Huawei Nexus 6P. They run Android OS 5.1 and 6.0 using Qualcomm Snapdragon 805 and 810 chipsets, respectively. Both support 4G LTE, 3G HSPA/UMTS/CDMA and 2G GSM. To activate access to multiple cellular networks, we have installed Google Fi SIM card on Nexus 6/6P, which supports T-Mobile and Sprint 3G/4G. Figure 4.28 illustrates the system implementation. iCellular runs as a daemon service on a rooted phone.

Basic APIs. iCellular allows the device to control its cellular access strategies through three APIs: `Monitor()` for active monitoring, `Predictor()` for performance prediction and `SwitchTo()` for direct switching. The decision fault tolerance is enabled by default.

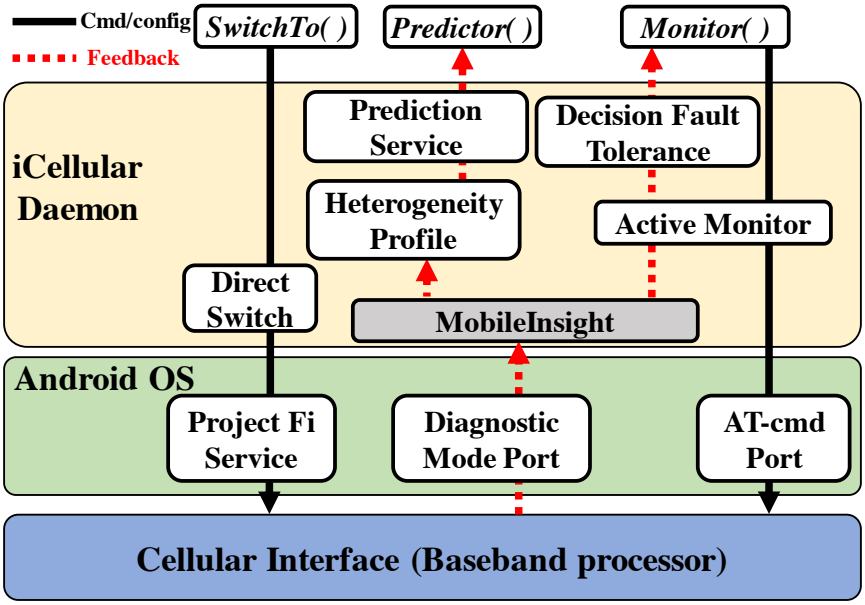


Figure 4.28.: Overview of iCellular implementation.

Usability-Flexibility trade-off. The above basic APIs provide most flexible means to customize access strategies. In practice, however, there is no need for most normal users to customize the strategies from the scratch. To support better usability, iCellular provides some built-in strategies on top of the basic APIs. Devices can choose these pre-defined ones, rather than build customized versions by themselves. We have developed three strategies: prediction-based, radio quality only and profile only.

Adaptive active monitoring. We implement `Monitor()` with manual search and adaptations. Our prototype initiates the search with an AT query command `AT+COPS=?`. The non-disruption and minimal search adaptations are implemented for events of Table 4.6.

Adaptive direct switch. We implement the `SwitchTo()` on top of PLMN selection, with dynamic adaptations for direct switch. Ideally, this can be executed with the AT command `AT+COPS=manual,carrier,network`. However, this command is forbidden by the cellular interface of Nexus 6/6P. We thus take an alternative approach. We modify the preferred network type through Android's API

`setPreferredNetworkType`, and change the carrier with Google Fi's secret code. Admittedly, this approach may incur extra switch overhead, but it is still acceptable.

Prediction for heterogenous carriers We implement `Predictor()` in two steps. First, we implement the online sample collection, which collects radio measurements, RRC configurations and QoS profiles as features. We also define a callback to collect the network/application-level performance metrics. We then implement the online regression tree algorithm for training and prediction.

Decision fault prevention. The fault prevention function is implemented as a shim layer between active monitoring and basic APIs. It detects the potential switch faults based on monitoring results and heterogeneity profiling, and excludes the unreachable carrier networks from the monitoring results. We further add a runtime checker in `SwitchTo()`, and prevent devices from selecting carriers not in the scanning results.

Cellular events collection. We use the built-in realtime cellular loggers from MobileInsight and redirect the events to the phone memory.

4.2.3 Overall Performance Evaluation

We evaluate iCellular along two dimensions. We first present the overall performance improvement by iCellular with smart multi-carrier access (§4.2.3), and then show iCellular satisfies various design properties in §4.2.2 (§4.2.4). All experiments are conducted on commodity Nexus 6 phones with iCellular in two cities of Los Angeles (west coast) and Columbus (Midwest), mainly around two campuses. The results on Nexus 6P are similar.

We use four representative applications to assess iCellular: SpeedTest (bulk file transfer), Web (interactive latency for small volume traffic), YouTube (video streaming) and Skype (real-time VoIP). We evaluate each application with quality-of-experience metrics whenever possible, *i.e.*, downlink speed for SpeedTest, page-loading time for Web [44] (measured with Firefox), video suspension time for YouTube [45] (measured

by its APIs), and latency for Skype [46] (measured with its tech info panel). We run both pedestrian mobility and static tests. Along the walking routes, we uniformly sample locations. Note that Google Fi's automatic selection protects the device's data connectivity by deferring its switch to the idle mode. For fair comparisons, we move to each sampled location in the idle mode (no voice/data, screen off), wait for sufficiently long time ($\geq$1min) for potential switch during idle, and then start to test each application. We have at least five test runs and use the median value for evaluation.

We compare iCellular and its variants, with two baselines: *(i)* **Google Fi's automatic selection** and *(ii)* **Optimal strategy:** We obtain the optimal access option by exhausting the application or network performance at each location. It may not be achieved in reality, but it serves as an ideal performance benchmark. We test three built-in iCellular decision strategies: (1) **Prediction-based:** the default strategy in iCellular, which chooses the carrier with the best ranking metric from the predictor. The predictor is trained based on our one-month user-study logs, and tested over different routes. (2) **Radio-only**: the *de-facto* handoff strategy in 3G/4G. We implement the standardized cell re-selection scheme [5]. Whenever a network 4G with its signal strength higher than -110 dBm (defined in [5]) exists, the strongest 4G carrier is chosen. Otherwise, we choose the strongest 3G network. (3) **Profile-only:** the device is migrated to the carrier network with the highest QoS (see Table 4.7). For our iCellular strategies, we use the carrier list with all network types supported by Google Fi (*i.e.*, 3G and 4G in T-Mobile and Sprint).

Figure 4.29 plots their performances in eight instances (locations), which belong to three categories: both carriers with acceptable coverage (Case 1-2), one carrier with acceptable coverage but the other not (Case 3-5), both carriers with weak coverage and one is even weaker (Case 6-8). We further compare them with the optimal one in two dimensions: accuracy toward the optimality, and the performance gap/improvement.

Accuracy toward optimality. We compare the probability that each scheme reaches the optimal network. Let I and I_{opt} be the access options chosen by the test

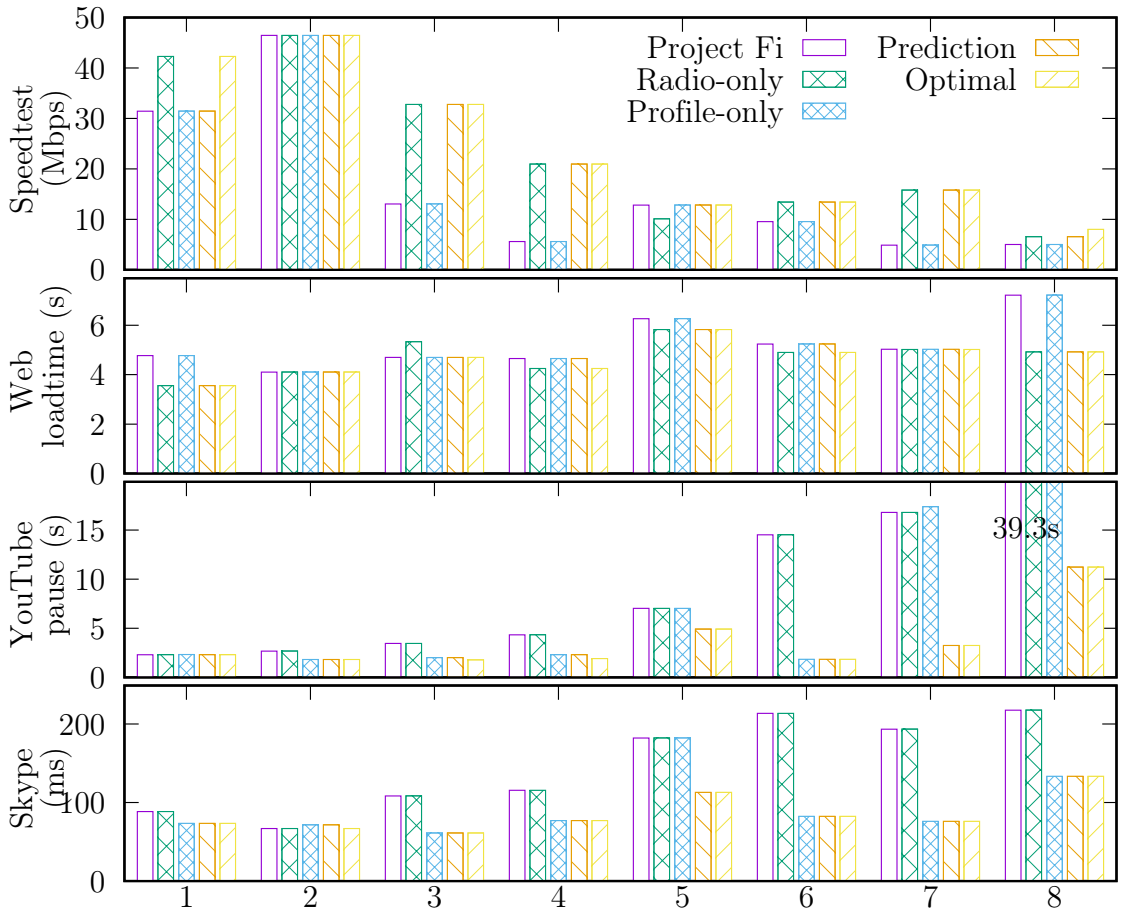


Figure 4.29.: Performance of Speed Test, Web, YouTube, Skype using various multi-carrier access schemes.

scheme and the optimal strategy. We define the hit ratio as the matching samples $|(I \doteq I_{opt})|$ over all test samples. Table 4.8 shows the hit ratios of all schemes by different applications. iCellular's prediction-based strategy makes a wiser multi-carrier access decision. The hit ratios are 73.6%, 57.8%, 50.9% and 92.5% in SpeedTest, Web, YouTube and Skype, respectively. They are relatively small in Web and YouTube, but do not incur much performance degradation (explained later). They are usually higher than Google Fi's automatic selection except for Web. The mobility speed has minor impact on the prediction accuracy, since it does not affect sample collection. Both radio measurements and cellular network profiles contribute to the high

accuracy, but their impacts on all apps vary. We calculate their normalized variable importance in the regression tree (defined in [47]) and Table 4.9 shows their weights for four apps. We also find that, the metric specific for one app often locates the better network for other apps at the same location. The reason is that, the characteristics of one carrier network tend to have consistent impact on all apps. When the performance gap between two carriers is significant, it would exhibit on all application-level metrics.

Table 4.8. Statistics of accuracy toward the optimality.

	Project Fi	**Radio-only**	**Profile-only**	**Prediction**
SpeedTest	47.3%	63.1%	36.8%	73.6%
Web	57.9%	73.6%	31.6%	57.8%
YouTube	16.9%	22.6%	49.1%	50.9%
Skype	24.5%	7.6%	84.9%	92.5%

Table 4.9. Weights of radio measurement and network profiles in iCellular's prediction strategy.

	Speed Test	**Web**	**YouTube**	**Skype**
Radio meas	36.5%	72.7%	26.4%	8.7%
Heterogeneity profile	63.5%	27.3%	73.6%	91.3%

Data service performance. We next examine the data performance by different schemes. We define the gap ratio $\gamma = |x - x^*|/x^*$, where x is the performance using various access strategies, x_{opt} is the optimal performance. We plot CDF of γ in Figure 4.30 and present the hit ratios and statistics of γ^+ in Table 4.10. Compared with Google Fi, iCellular narrows its performance gap (*e.g.*, reducing the maximal speed loss from 73.7% (19.7 Mbps) to 25.7%, and the maximal video suspension time gap from 28.1 s to 3.2 s). The performance gain varies with locations (see Figure 4.29). With acceptable coverage (Case 1-2), Google Fi's performance also approximates the optimal one. However, at locations with weak coverage, iCellular improves the device performance more visibly. The performance gain varies with applications (traffic patterns). Compared with other traffic, iCellular provides relatively small improvement

for Web browsing. The reason is that, the Web traffic volume is relatively small, and no large performance distinction appears among various access options. However, for heavy traffic (*e.g.*, file transfer), video streaming and voice calls, iCellular substantially improves the performance. The average improvement of iCellular over Google Fi approximates $\gamma_{fi} - \gamma_{icellular}$. On average, iCellular increases 23.8% downlink speed and reduces 7.3% loading time in Web, 37% suspension time in YouTube, 60.4% latency in Skype. Since iCellular often selects the optimal access, the maximal gain over Google Fi can be up to 46.5% in Web, 6.9x in YouTube, 1.9x in Skype, and 3.74x in SpeedTest.

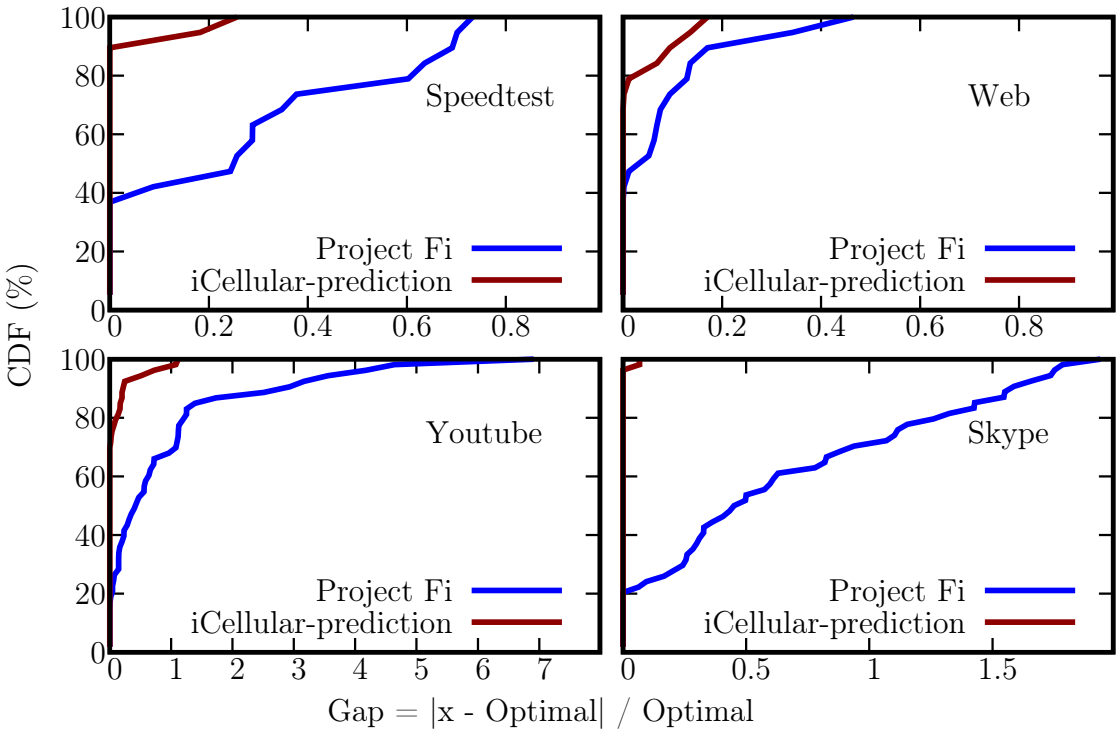


Figure 4.30.: The performance gaps from Google Fi and iCellular's prediction strategy to the optimality.

Comparison between iCellular's built-in strategies. iCellular's prediction strategy best approximates the optimal strategy. It outperforms radio-only and profile-only variants. We also see that, the importance of profile and radio measurements varies across applications. For example, our log-event analysis shows that, T-Mobile

Table 4.10. Performance gaps from the optimal one.

	Project Fi		**iCellular-prediction**	
	med(γ) ($\|x - x^*\|$)	max(γ) ($\|x - x^*\|$)	med(γ) ($\|x - x^*\|$)	max(γ) ($\|x - x^*\|$)
Speed Test (speed)	36.2% 3.8Mbps	73.3% 19.8Mbps	**12.4%** 1.4Mbps	25.7% 9.8Mbps
Web (loadtime)	8.5% 0.5s	46.5% 2.3s	**1.2%** 0.2s	17% 0.7s
YouTube (Pause)	55% 1.4s	690% 28.1s	**18%** 0.3s	111% 3.2s
Skype (Latency)	62.9% 64ms	193.8% 117ms	**2.5%** 4.4ms	6.7% 4.5ms

assigns Google Fi devices to the interactive traffic class (Table 4.7), which is optimized for delay-sensitive service [48]. Note, This QoS is specific to Google Fi. We verify that a T-Mobile device with Samsung S5 is assigned lower background class. Instead, Sprint only allocates the best-effort traffic class to these devices. This explains why the profile-only strategy's performance approximates the optimal strategy for Skype. It also implies that, for a given application (*e.g.*, Skype), simpler strategy (rather than prediction), which incurs smaller system overhead, can be available for close-to-optimal performance.

4.2.4 Micro-Benchmark Evaluation

We next present the micro-benchmark evaluations on iCellular's key components, and validate that they are efficient. We examine the active monitoring, direct switch and fault prevention, as well as the overhead of signaling, CPU, memory and battery usage.

Efficiency. We examine iCellular's efficiency through two adaptive module tests. First, we show that, iCellular's adaptive monitoring is able to accelerate carrier scanning. We compare it with the default manual search, and record the total search time and the number of cells scanned at 100 different locations. Figure 4.31 shows that, with adaptive search, 70% of the complete search can be completed within 10 s, 64%

shorter than the exhaustive manual search. Note that devices are allowed to switch before the complete search, so it waits shorter in practice. Figure 4.31(b) counts the scanned cells, and validates that such savings come from avoiding those unnecessary cell scans. The search time and the number of cells vary with locations and the cell density.

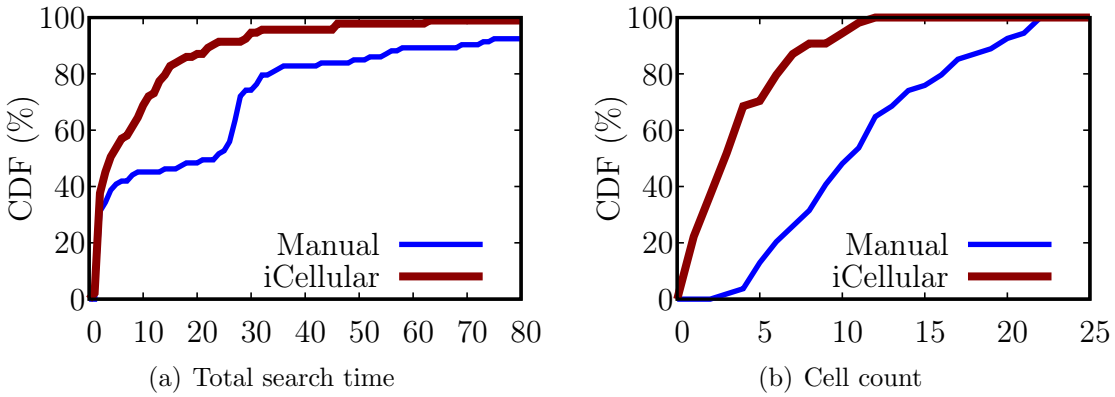


Figure 4.31.: iCellular's adaptive monitoring avoids exhaustive search.

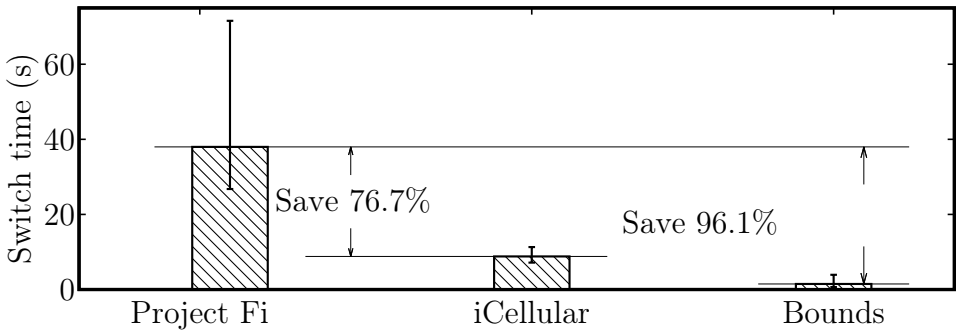


Figure 4.32.: Inter-carrier switch time.

Second, we examine how well iCellular's adaptive switch reduces service disruption. In this experiment, we place the phone at the border of two carriers' coverages, and test the switch time needed for iCellular and Google Fi for 50 runs. The inter-carrier switch time is defined as the duration from the de-registration from the old carrier to the registration to the new carrier. For comparison purposes, we also calculate the lower bound based on the MobileInsight event logs. Figure 4.32 shows that,

iCellular saves 76.7% switch time on average, compared with Google Fi. However, the current iCellular prototype has not achieved the minimal switch time: it still requires 8.8s on average. Under high-speed mobility, this may delay the switch to the optimal carrier network. We dig into the event logs, and discover that, the current bottleneck lies in the SIM card reconfiguration. The current iCellular implementation relies on Google Fi's system service. It has to wait until the SIM card is reconfigured to switch to another carrier. In the experiments, we find that most of the switch times (7.3 s on average) are spent on the SIM card reconfiguration, which is beyond the control of iCellular. The phone has no network service in this period. The lower bound implies that, with better SIM card implementation, iCellular could save up to 96.1% of switch time compared with the Google Fi.

Fault prevention. We next verify that iCellular handles fault scenarios and prevents devices from switching to unwise carrier networks. All three failure types have been observed in our one-month user study. Note that the failure scenarios are not very common in reality. We observe one instance of the forbidden access, where a Sprint 4G base station sets the access-barring option for 10 min (possibly under maintenance). We observe another instance of Figure 4.26, where T-mobile 4G is available but T-Mobile 3G is not available. Since T-Mobile 4G does not provide Voice over LTE (VoLTE) to Google Fi and has to rely on its 3G network (using circuit-switching Fallback) for voice calls [49]. Consequently, the correct decision should be to not switch to T-Mobile 4G, since voice calls are not reachable there. iCellular detects it from the profiled call preference and location update messages, and excludes this access option from the candidate list. We also observe uncoordinated mobility rules between the network and the device (Figure 4.27). We validate that iCellular can detect and avoid them.

Impact on applications in monitoring. We show that iCellular's active monitor does not disrupt the ongoing data service at the device. We run the active monitor 100 times with/without applications and its active data transfer. We test with four applications and the results with/without iCellular's monitoring are similar. Figure 4.33

shows the performance with/without iCellular's monitoring for YouTube and Skype. Enabling/disabling active monitoring has comparable application performance. As explained in §4.2.2, this is because the carrier scanning procedure is performed only in the absence of traffic.

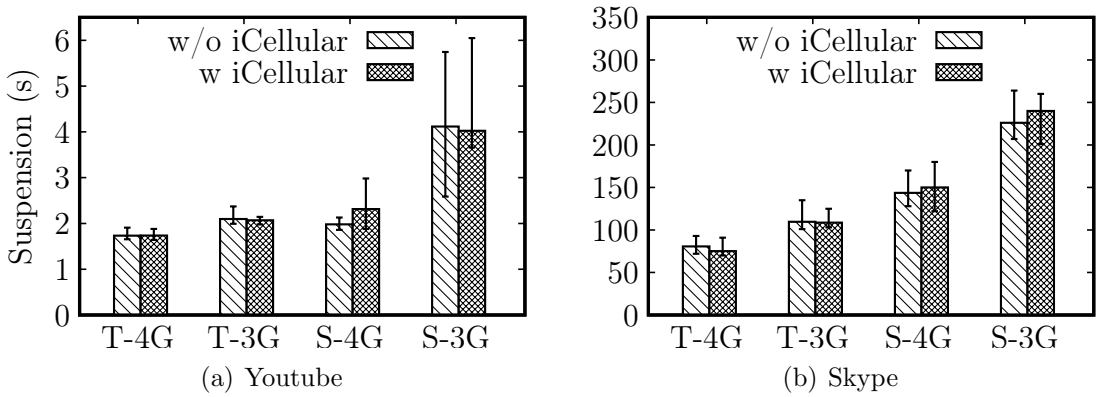


Figure 4.33.: iCellular's active monitoring has minor impacts on data performance.

Signaling overhead. We show that iCellular incurs moderate signaling messages to the device and the network. We record the device-side signaling message rate under three conditions (when running our performance tests): (1) *Idle:* No monitoring/switch functions are active. No extra cellular signaling messages are generated. (2) *Monitor:* iCellular initiates its active monitoring. The device should receive more broadcasted signals. However, no extra signaling messages are generated to the network. (3) *Switch:* iCellular initiates the switch to the new carrier network. Because of the registration, extra signaling messages are generated to both the device and the network. For all scenarios, we count the radio-level (from RRC layer), core-network level (from mobility and session management layers) and the total signaling rate. Figure 4.34 shows that, the maximum observed signaling message rate is 32 message/sec.

CPU and memory. In all our tests, the maximum CPU utilization is below 2%, while the maximum memory usage is below 20 MB (including virtual memory). Figure 4.35 shows a 20 min log during a driving test, where its maximum memory usage is 16.45 MB.

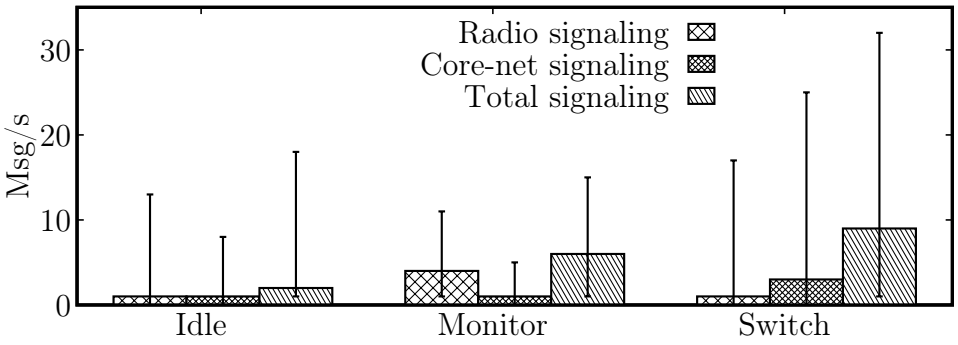


Figure 4.34.: Cellular signaling overhead.

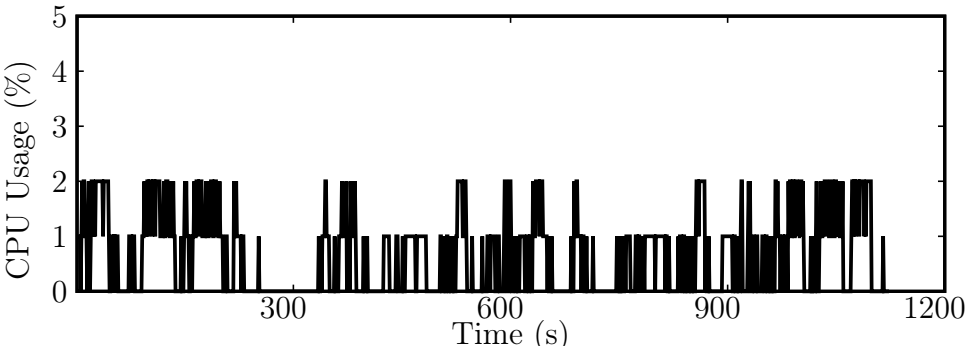


Figure 4.35.: CPU usage of iCellular.

Energy consumption. Since we cannot directly measure the consumed power at Nexus 6/6P with an external power meter (its battery is sealed, and hard to remove), we take an application-level approach. We use a fully-charged Nexus 6 phone and run it for 24 hours. We use an app called GO-Power-Master [50] to record energy consumption for each component/app. Figure 4.11 shows one record, where iCellular explicitly consumes about 4.75% of battery. Its energy can be further optimized (*e.g.*, with sleep mode and periodical monitoring).

Table 4.11. Battery usage of iCellular.

	WeChat	**iCellular**	Skype	Gmail	Go Power Master	Google Play
Usage	21.03%	**4.75%**	4.54%	3.33%	1.21%	0.73%

5 RELATED WORK

In this chapter, we summarize the literature related to research efforts on radio access network in the context of cellular networks in recent years.

5.1 Radio Access Network of Cellular Networks

Radio Access Platform and Analysis. Kumar *et al.* [40] proposed LTEye, an open platform implemented over USRP software radios to monitor and analyze LTE radio performance. It provides fine granularity information and deep insights in LTE physical layer of deployed networks. Gudipati *et al.* [51] proposed SoftRAN, a software defined centralized control plane for radio access networks which can effectively perform load balancing, interference management and other tasks. Nikaein *et al.* [52] presented OpenAirInterface as a suitably flexible platform that offer a wide range of experimentation modes from real-world experimentation to controlled and scalable evaluations for LTE and 5G mobile networks.

Radio Signal and Energy Saving. Balasubramanian *et al.* [53] conducted measurement study to characterize energy consumption of 3G, GSM and Wi-Fi and identified high tail energy overhead for 3G and GSM. Huang *et al.* [54] empirically model the power consumption of commercial LTE network and found LTE is much less power efficient compared with Wi-Fi and even less efficient than 3G due to the long high power tail. Athivarapu *et al.* [55] proposed RadioJockey to cut down extra energy consumption in HSPA and LTE caused by mobile device keep remaining in high energy state after each radio communication spurt. They used program execution traces to predict the end of communication spurts and accurately invoke fast dormancy without increasing signaling overhead due to frequent state transitions.

Management-plane Diagnostics and Design. Tu *et al.* [37] showed how mobility affects mobile data accounting with an accounting gap due to accounting discrepancy caused by reasons like handoff and insufficient coverage. Tu *et al.* [56] proposed signaling diagnosis tools and uncover six instances of problematic interactions in control-plane protocols of cellular networks. Yazici *et al.* [57] proposed an all-SDN network architecture for 5G systems with a show case of unified handoff, mobility and routing management.

Carrier Aggregation Pedersen *et al.* [58] presented a tutorial of carrier aggregation functionality in LTE-A regarding basic concepts, control mechanisms and performance aspects. Reinikainen [59] measured carrier aggregation downlink performance using 2×20 MHz bandwidth with 2×2 MIMO configuration in lab environment.

TCP over Cellular. Elaarag [60] made a survey of research to improve TCP performance over mobile wireless networks. Huang *et al.* [61] discovered various inefficiencies in TCP over LTE such as undesired slow start, significantly under-utilization of available bandwidth. They called for more LTE-friendly transport protocol and application behaviour. Zaki *et al.* [62] proposed an end-to-end congestion control protocol that uses delay measurements to quickly react to cellular network capacity changes. Lu *et al.* [63] proposed CQIC (Channel Quality Indicator Control), a cross-layer congestion control which leverages the rich physical layer information exchanged between base stations and mobile phones to predict the capacity of cellular link. Goyal *et al.* [64] proposed Accel-Brake Control protocol which rapidly acquires any capacity opens up and responds promptly to congestion for cellular networks.

Application Performance over Cellular Xie *et al.* [65] proposed piStream to efficiently monitor the LTE base station's physical layer resource allocation so that it can estimate available bandwidth and achieve high video quality with minimal stalling rate for adaptive HTTP video streaming over LTE. Xie *et al.* [66] proposed Cellular Link-Aware Web Loading to boost mobile web loading using physical layer

information to estimate and fully utilize available network bandwidth. Mandalari *et al.* [67] provided an in-depth measurement study of mobile network configurations, performance characterization and content discrimination for international data roaming within Europe. Tan *et al.* [68] identified signaling operations as latency bottleneck for moible virtual reality application and proposed LTE-VR which leverages cross-layer design and channel information to reduce various latency sources in signaling operations. Balasingam *et al.* [69] proposed to detect if the LTE downlink is bottleneck by monitoring output patterns of proprietary downlink scheduling algorithms at base station. With the detection, they improve video streaming significantly in case of middlebox-forced slow-start restart.

5.2 Mobility Support in Cellular Networks

Performance On The Go. Jang *et al.* [70] conducted measurement study of the UDP and TCP performance over 3G/3.5G networks in fast moving scenarios on a highway and high-speed train and showed far worse performance than stationary cases. Tso *et al.* [71] conducted a large scale empirical study on the performance of commercial mobile HSPA networks in Hong Kong. In one hand, they showed mobility has largely negative impacts like serious service deterioration or interruption caused by fast-changing wireless environment. In the other hand, they found mobility can improve fairness of bandwidth sharing among users and traffic flows. Javed *et al.* [72] used cellular network conditions measured at mobile device to predict handoffs with 80% accuracy in 3G networks using machine learning approach. Chen *et al.* [73] conducted measurement study using MPTCP over cellular and Wi-Fi links regarding latency and factors with influence. They concluded MPTCP provides a robust data transport and reduces variability in download latency. Siris *et al.* [74] proposed an approach to improve mobile video streaming by utilizing mobility and throughput prediction to prefetch video streaming data in integrated cellular and Wi-Fi networks. Merz *et al.* [75] presented a measurement study of LTE performance at high velocities

up to 200 km/h and showed limited influence of speed if SNR coverage is well and benefit from MIMO spatial multiplexing. Li *et al.* [76] conducted a comprehensive study to investigate TCP behaviour and performance in a high speed environment and called for more adaptive transport protocols. Li *et al.* [77] measured MPTCP behaviour and performance with two cellular carriers on high speed rails and identified better robustness and low efficiency due to high frequent handoff. Xu *et al.* [78] conducted a comprehensive measurement study in LTE networks on how handover decisions implemented by carriers impact throughput performance.

Handoff Mechanism for Heterogeneous Systems. McNair *et al.* [79] studied mobility management techniques for vertical handoffs between different types of network from 3G and beyond to 4G networks. Brunner *et al.* [80] studied different strategies for inter-system handover between WCDMA and GSM and intra-frequency handover parameters for WCDMA. Liu *et al.* [81] analyzed and extended traditional hysteresis based and timer-based algorithms to support inter-system/intra-system handoff decision in complex heterogeneous wireless environment. Chowdhury *et al.* [82] proposed a handover optimization scheme between UMTS based macrocell and femtocell networks and evaluation with simulation. Xenakis *et al.* [83] made a comprehensive discussion on mobility management support for femtocells in LTE-A systems including handover decision procedure, algorithm and and criterion used. Cheelu *et al.* [84] proposed a vertical handoff decision model based on user preferences and expert opinions with analytic hierarchy process aiming to maximize user satisfaction.

Handoff Parameters and Decision Algorithm. Flore *et al.* [85] investigated the impact of operator-configurable cell reselection parameters in UMTS regarding performance metrics under different RF environment. Fathi *et al.* [86] evaluated different low-latency handover schema for VoIP services in 3G systems. Lobinger *et al.* [87] presented simulation results of a coordination system showing interactions of load balancing and handover parameter optimization algorithms. Kitagawa *et al.* [88] pro-

posed a self-optimization algorithm for handover parameters which adaptively adjusts parameters considering handover failure cause to provide mobility robustness. Lee *et al.* [89] proposed a spectrum-aware mobility management scheme for cognitive radio cellular networks with consideration of spatially heterogeneous spectrum availability. Lin *et al.* [90] proposed LTE Hard Handover algorithm with RSRP constraint to minimize number of handovers and system delay while maximize the system throughput. Zhang *et al.* [91] proposed a handover optimization algorithm based on UE's mobility state with significant reduction in signalling overhead for LTE femtocells. Qiang *et al.* [92] proposed a user centered handoff scheme for hybrid 5G environment with multiple optimize objects including maximizing the achievable data receiving rate and minimizing the block probability. Ahmad *et al.* [93] presented a survey of state-of-the-art handover procedures and decision algorithms proposed for LTE/LTE-A. Zhao *et al.* [94] used data from network operators to reproduce and present facts about handoff process and handoff loop.

Traffic Offloading and MPTCP. Balasubramanian *et al.* [95] proposed Wiffler which uses predictions on Wi-Fi connectivity to offload data on Wi-Fi from 3G. Paasch *et al.* [96] proved the feasibility of using MPTCP for transparent mobile/Wi-Fi handover. Dong *et al.* [97] proposed iDEAL, an auction-based framework which allows cellular service provider to offload traffic by buying capacity when needed from third-party resource owners. Nikravesh *et al.* [98] proposed MPFlex which strategically employs multiplexing to improve multipath performance for mobile devices.

Multiple-Operator Network Access Panchal *et al.* [99] explored and simulated performance of capacity sharing and spectrum sharing on traditional infrastructure and virtualized spectrum sharing and virtualized PRB sharing on virtualized infrastructure. Di *et al.* [100] studies the effectiveness of networking sharing among two Irish operators regarding space coverage and performance by analyzing their deployment and traffic traces. Jokinen *et al.* [101] presented a flexible way to share spectrum resources between multiple operators to increase spectral efficiency. Kour *et al.* [102]

surveyed techniques and methods of spectrum sharing for future generation networks, proposed architecture to depict the complete scenario, analyzed security issues discussed advantages to meet high bandwidth requirements.

6 CONCLUSIONS AND FUTURE WORK

This dissertation gives answers to three questions for mobility support in cellular networks:

Q1) How does mobility support perform in reality?

Q2) Does mobility support go wrong? If yes, how and why?

Q3) How can we improve mobility support?

The research contributions of this dissertation are summarized as follows.

We first study how mobility support perform in reality and managed by carriers. We conduct a sizable measurement study on policy-based handoff configurations from 30 mobile carriers in the US and globally. We design a new device-centric tool which collects runtime handoff configurations without the assistance from operators. Our analysis exhibits that extremely complex and diverse configurations are deployed by operators in reality. Follow that, we present the first work to expose mobile network carriers' mobility management policy, which is confidential and even proprietary. We propose a generic device-centric approach to characterize, model and track configuration dynamics and infer the policy behind. Through this closer look, we understand how four US carriers dynamically manage their mobility support in the wild.

Next, we examine the potential undesired handoff behaviors from three perspective and study the root cause behind. First, we identify and study handoff instability and unreachability issues caused by uncoordinated handoff configurations and conflict handoff decision logics. The discovered persistent handoff loops, handoff convergence split and premature convergence, as well as their triggering conditions, have been partially validated in operational networks. Though the incurred damage, in terms of signaling overhead and performance degradation, is not appalling to some users, such problematic issues should be addressed as we seek to build a more dependable, high-performance, mobile network infrastructure. Second, we dig into the extremely

diverse handoff configurations and study how the setting of handoff configuration values affect data performance and user experience when moving. Our study shows that such diverse configurations lead to unexpected negative compound effect to performance and efficiency. Third, We present the arguably first study to unveil missed performance in operational cellular networks. Our measurement over 4 US carriers shows that missed performance indeed happens and happens a lot. We pinpoint the root causes into today's network operations on selecting cells.

Last but not least, we propose two solutions to boost the performance of mobility support by utilizing the intelligence of mobile devices. In the first work, we present the design, implementation and evaluation of iCellSpeed. Our effort is motivated by the premise that today's network-centric cell selection may result in a non-negligible, sub-optimal choice of cells for a given device. Consequently, the user device suffers from large access speed dip from the highest available one in practice, while the network suffers from significantly underutilizing the available resources. Our extensive measurements have confirmed both. The root cause lies in today's network-centric cell selection scheme where the device has minimal influence on decision making. While this might work on the dumb terminals in the past telecom age, it does not work well with the increasingly capable and smart devices in the Internet and AI age. iCellSpeed thus explores a new paradigm of "device-assisted, infrastructure-decided" design for 4.5G and beyond. As a result, it is a win-win game for both the device and the infrastructure. The infrastructure better utilizes its current resources, while the device gains its desirable, higher access speed. iCellSpeed thus improves device performance without upgrading the infrastructure, but via smart decision inputs from users and better utilization for networks. In the second work, we show that the current design of cellular networks limits the device's ability to fully explore multi-carrier access. The fundamental problem is that, existing 3G/4G mobile networks place most decisions and operational complexity on the infrastructure side. This network-centric design is partly inherited from the legacy telecom-based architecture paradigm. As a result, the increasing capability of user devices is not properly exploited. In the

multi-carrier access context, devices may suffer from low-quality access while incurring unnecessary service disruption. In this work, we describe iCellular, which seeks to leverage the fine-grained cellular information and the available mechanism at the device. It thus dynamically selects better mobile carrier through adaptive monitoring and online learning. Our evaluation validates the feasibility of this approach for both single-carrier and multi-carrier networks.

Future work. First, 5G is already here but the identified issues of mobility support do not go away. Considering much denser cell deployment for 5G networks with small cells and heterogeneous radio access technologies to be applied, those issues could be even worse. Examining the behaviors of mobility support in 5G networks and performance impact is very necessary. Second, we evaluate the performance of mobility support using network access speed only while there are other metrics of interest like latency and reliability. The impact of mobility support to these metrics remains unknown and it is promising to explore the potential room for improvement with better mobility support. Last but not least, the original design of mobility support has its limitations as unveiled in our study, *i.e.*, the distributed nature of existing implementation and management and less optimization for data performance. Both of them hurt the user experience and are conflicted with the goal of modern cellular networks. It is time for people to revisit the design of this core feature of cellular networks.

REFERENCES

REFERENCES

[1] Wikipedia. Cellular frequencies. https://en.wikipedia.org/wiki/Cellular_frequencies.

[2] 3GPP. TS36.133: E-UTRA; Requirements for support of radio resource management.

[3] 3GPP. TS25.304: User Equipment (UE) Procedures in Idle Mode and Procedures for Cell Reselection in Connected Mode.

[4] 3GPP. TS25.133: Requirements for support of radio resource management (FDD).

[5] 3GPP. TS36.304: E-UTRA; User Equipment Procedures in Idle Mode.

[6] 3GPP. TS23.009: Handover Procedures.

[7] Yuanjie Li, Chunyi Peng, Zengwen Yuan, Jiayao Li, Haotian Deng, and Tao Wang. Mobileinsight: Extracting and analyzing cellular network information on smartphones. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 202–215, 2016.

[8] MobileInsight Lab. http://milab.cs.purdue.edu.

[9] Mark O Hill. Diversity and evenness: a unifying notation and its consequences. *Ecology*, 54(2):427–432, 1973.

[10] Brian Everitt and Anders Skrondal. *The Cambridge dictionary of statistics*, volume 106. Cambridge University Press Cambridge, 2002.

[11] What are at&t's 4g lte bands in 2018? https://forums.att.com/t5/Phone-Device-Upgrades/What-are-AT-amp-T-s-4G-LTE-bands-in-2018/td-p/5359265, 2018.

[12] 3GPP. TS36.101: E-UTRA; User Equipment (UE) radio transmission and reception.

[13] Frequency calculator. http://niviuk.free.fr/lte_band.php, 2018.

[14] The move to band 30 causing issues with other bands? https://forums.att.com/t5/Apple/The-move-to-Band-30-causing-issues-with-other-bands/td-p/5267142, 2017.

[15] Chunyi Peng and Yuanjie Li. Demystify undesired handoff in cellular networks. In *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9. IEEE, 2016.

[16] 3GPP. TS36.331: E-UTRA; Radio Resource Control; Protocol specification.

[17] 3GPP. TS24.008: Mobile radio interface Layer 3 specification; Core network protocols.

[18] 3GPP. TS25.331: Universal Mobile Telecommunications System (UMTS); Radio Resource Control (RRC); Protocol specification.

[19] HotMobile'20 dataset. http://milab.cs.purdue.edu/hotmobile2020_release/.

[20] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[21] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[22] How to lock on a specific cell tower on samsung galaxy s6. https://www.violetgem.com/blog/lte-wcdma-gsm-how-to-lock-on-a-specific-cell-tower-on-samsung-galaxy-s6/, 2016.

[23] Network signal guru. https://play.google.com/store/apps/details?id=com.qtrun.QuickTest, 2020. Access: 01/21/2020.

[24] Android.telephony. http://developer.android.com/reference/android/telephony/package-summary.html, 2020.

[25] At command user guide. http://gamma.spb.ru/images/pdf/L506_AT_Command_User_Guide_V2.1.pdf, 2016.

[26] Codes google pixel. https://www.hardreset.info/devices/google/google-pixel/codes/, 2020. Access: 01/19/2020.

[27] Android open source project. https://source.android.com/, 2020.

[28] icellspeed dataset release. https://github.com/mssn/iCellSpeed-Dataset, 2020.

[29] Radio layer interface. https://wladimir-tm4pda.github.io/porting/telephony.html, 2020. Access: 06/21/2020.

[30] Bitmovin demos and code examples. https://bitmovin.com/demos/, 2020. Access: 06/20/2020.

[31] 3GPP. TS23.502: Procedures for the 5G System (5GS).

[32] 5g speed report: Early 5g experience provides mixed results. https://www.telecompetitor.com/5g-speed-report-early-5g-experience-provides-mixed-results/, 2020. Access: 03/01/2020.

[33] 3GPP. TS23.122: Non-Access-Stratum (NAS) functions related to Mobile Station (MS) in idle mode.

[34] Google fi. https://fi.google.com/about/, 2020. Access: 08/11/2020.

[35] Apple sim for ipad. https://www.apple.com/ipad/cellular/, 2020. Access: 08/11/2020.

[36] Using your galaxy smartphones with esim. https://www.samsung.com/my/support/mobile-devices/using-your-galaxy-smartphones-with-esim/, 2020. Access: 08/11/2020.

[37] Guan-Hua Tu, Chunyi Peng, Chi-Yu Li, Xingyu Ma, Hongyi Wang, Tao Wang, and Songwu Lu. Accounting for roaming users on mobile data access: Issues and root causes. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 305–318, 2013.

[38] 3GPP. TS27.007: AT command set for User Equipment (UE).

[39] Qualcomm extensible diagnostic monitor. https://www.qualcomm.com/documents/qxdm-professional-qualcomm-extensible-diagnostic-monitor, 2020. Access: 08/15/2020.

[40] Swarun Kumar, Ezzeldin Hamed, Dina Katabi, and Li Erran Li. Lte radio analytics made easy and accessible. *ACM SIGCOMM Computer Communication Review*, 44(4):211–222, 2014.

[41] Sanae Rosen, Haokun Luo, Qi Alfred Chen, Z Morley Mao, Jie Hui, Aaron Drake, and Kevin Lau. Discovering fine-grained rrc state dynamics and performance impacts in cellular networks. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 177–188, 2014.

[42] 3GPP. TS24.301: Non-Access-Stratum (NAS) for EPS.

[43] Stuart L Crawford. Extensions to the cart algorithm. *International Journal of Man-Machine Studies*, 31(2):197–217, 1989.

[44] Victor Agababov, Michael Buettner, Victor Chudnovsky, Mark Cogan, Ben Greenstein, Shane McDaniel, Michael Piatek, Colin Scott, Matt Welsh, and Bolian Yin. Flywheel: Google's data compression proxy for the mobile web. In *12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*, pages 367–380, 2015.

[45] Ricky KP Mok, Edmond WW Chan, and Rocky KC Chang. Measuring the quality of experience of http video streaming. In *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, pages 485–492. IEEE, 2011.

[46] Sofiene Jelassi, Gerardo Rubino, Hugh Melvin, Habib Youssef, and Guy Pujolle. Quality of experience of voip service: A survey of assessment approaches and open issues. *IEEE Communications surveys & tutorials*, 14(2):491–513, 2012.

[47] Estimates of predictor importance for regression tree. https://www.mathworks.com/help/stats/compactregressiontree.predictorimportance.html, 2020. Access: 08/16/2020.

[48] 3GPP. TS23.107: Quality of Service (QoS) concept and architecture.

[49] Guan-Hua Tu, Chunyi Peng, Hongyi Wang, Chi-Yu Li, and Songwu Lu. How voice calls affect data in operational lte networks. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 87–98, 2013.

[50] Go power master. `https://go-power-master.en.uptodown.com/android`, 2020. Access: 08/16/2020.

[51] Aditya Gudipati, Daniel Perry, Li Erran Li, and Sachin Katti. Softran: Software defined radio access network. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pages 25–30, 2013.

[52] Navid Nikaein, Mahesh K Marina, Saravana Manickam, Alex Dawson, Raymond Knopp, and Christian Bonnet. Openairinterface: A flexible platform for 5g research. *ACM SIGCOMM Computer Communication Review*, 44(5):33–38, 2014.

[53] Niranjan Balasubramanian, Aruna Balasubramanian, and Arun Venkataramani. Energy consumption in mobile phones: a measurement study and implications for network applications. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, pages 280–293, 2009.

[54] Junxian Huang, Feng Qian, Alexandre Gerber, Z Morley Mao, Subhabrata Sen, and Oliver Spatscheck. A close examination of performance and power characteristics of 4g lte networks. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 225–238, 2012.

[55] Pavan K Athivarapu, Ranjita Bhagwan, Saikat Guha, Vishnu Navda, Ramachandran Ramjee, Dushyant Arora, Venkat N Padmanabhan, and George Varghese. Radiojockey: mining program execution to optimize cellular radio usage. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 101–112, 2012.

[56] Guan-Hua Tu, Yuanjie Li, Chunyi Peng, Chi-Yu Li, Hongyi Wang, and Songwu Lu. Control-plane protocol interactions in cellular networks. *ACM SIGCOMM Computer Communication Review*, 44(4):223–234, 2014.

[57] Volkan Yazıcı, Ulas C Kozat, and M Oguz Sunay. A new control plane for 5g network architecture with a case study on unified handoff, mobility, and routing management. *IEEE Communications Magazine*, 52(11):76–85, 2014.

[58] Klaus Ingemann Pedersen, Frank Frederiksen, Claudio Rosa, Hung Nguyen, Luis Guilherme Uzeda Garcia, and Yuanye Wang. Carrier aggregation for lte-advanced: functionality and performance aspects. *IEEE Communications Magazine*, 49(6):89–95, 2011.

[59] Antti Reinikainen et al. Performance evaluation of lte-advanced carrier aggregation. 2015.

[60] Hala Elaarag. Improving tcp performance over mobile networks. *ACM Computing Surveys (CSUR)*, 34(3):357–374, 2002.

[61] Junxian Huang, Feng Qian, Yihua Guo, Yuanyuan Zhou, Qiang Xu, Z Morley Mao, Subhabrata Sen, and Oliver Spatscheck. An in-depth study of lte: effect of network protocol and application behavior on performance. *ACM SIGCOMM Computer Communication Review*, 43(4):363–374, 2013.

[62] Yasir Zaki, Thomas Pötsch, Jay Chen, Lakshminarayanan Subramanian, and Carmelita Görg. Adaptive congestion control for unpredictable cellular networks. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pages 509–522, 2015.

[63] Feng Lu, Hao Du, Ankur Jain, Geoffrey M Voelker, Alex C Snoeren, and Andreas Terzis. Cqic: Revisiting cross-layer congestion control for cellular networks. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pages 45–50, 2015.

[64] Prateesh Goyal, Mohammad Alizadeh, and Hari Balakrishnan. Rethinking congestion control for cellular networks. In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, pages 29–35, 2017.

[65] Xiufeng Xie, Xinyu Zhang, Swarun Kumar, and Li Erran Li. pistream: Physical layer informed adaptive video streaming over lte. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 413–425, 2015.

[66] Xiufeng Xie, Xinyu Zhang, and Shilin Zhu. Accelerating mobile web loading using cellular link information. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 427–439, 2017.

[67] Anna Maria Mandalari, Andra Lutu, Ana Custura, Ali Safari Khatouni, Özgü Alay, Marcelo Bagnulo, Vaibhav Bajpai, Anna Brunstrom, Jörg Ott, Marco Mellia, et al. Experience: Implications of roaming in europe. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 179–189, 2018.

[68] Zhaowei Tan, Yuanjie Li, Qianru Li, Zhehui Zhang, Zhehan Li, and Songwu Lu. Supporting mobile vr in lte networks: How close are we? *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(1):1–31, 2018.

[69] Arjun Balasingam, Manu Bansal, Rakesh Misra, Kanthi Nagaraj, Rahul Tandra, Sachin Katti, and Aaron Schulman. Detecting if lte is the bottleneck with bursttracker. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–15, 2019.

[70] Keon Jang, Mongnam Han, Soohyun Cho, Hyung-Keun Ryu, Jaehwa Lee, Yeongseok Lee, and Sue B Moon. 3g and 3.5 g wireless network performance measured from moving cars and high-speed trains. In *Proceedings of the 1st ACM workshop on Mobile internet through cellular networks*, pages 19–24, 2009.

[71] Fung Po Tso, Jin Teng, Weijia Jia, and Dong Xuan. Mobility: A double-edged sword for hspa networks: A large-scale test on hong kong mobile hspa networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(10):1895–1907, 2011.

[72] Umar Javed, Dongsu Han, Ramon Caceres, Jeffrey Pang, Srinivasan Seshan, and Alexander Varshavsky. Predicting handoffs in 3g networks. *ACM SIGOPS Operating Systems Review*, 45(3):65–70, 2012.

[73] Yung-Chih Chen, Yeon-sup Lim, Richard J Gibbens, Erich M Nahum, Ramin Khalili, and Don Towsley. A measurement-based study of multipath tcp performance over wireless networks. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 455–468, 2013.

[74] Vasilios A Siris, Maria Anagnostopoulou, and Dimitris Dimopoulos. Improving mobile video streaming with mobility prediction and prefetching in integrated cellular-wifi networks. In *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pages 699–704. Springer, 2013.

[75] Ruben Merz, Daniel Wenger, Damiano Scanferla, and Stefan Mauron. Performance of lte in a high-velocity environment: A measurement study. In *Proceedings of the 4th workshop on All things cellular: operations, applications, & challenges*, pages 47–52, 2014.

[76] Li Li, Ke Xu, Dan Wang, Chunyi Peng, Kai Zheng, Rashid Mijumbi, and Qingyang Xiao. A longitudinal measurement study of tcp performance and behavior in 3g/4g networks over high speed rails. *IEEE/ACM transactions on networking*, 25(4):2195–2208, 2017.

[77] Li Li, Ke Xu, Tong Li, Kai Zheng, Chunyi Peng, Dan Wang, Xiangxiang Wang, Meng Shen, and Rashid Mijumbi. A measurement study on multi-path tcp with multiple cellular carriers on high speed rails. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 161–175, 2018.

[78] Shichang Xu, Ashkan Nikravesh, and Z Morley Mao. Leveraging context-triggered measurements to characterize lte handover performance. In *International Conference on Passive and Active Network Measurement*, pages 3–17. Springer, 2019.

[79] Janise McNair and Fang Zhu. Vertical handoffs in fourth-generation multinetwork environments. *IEEE Wireless communications*, 11(3):8–15, 2004.

[80] Christopher Brunner, Andrea Garavaglia, Mukesh Mittal, Mohit Narang, and Jose Vargas Bautista. Inter-system handover parameter optimization. In *IEEE Vehicular Technology Conference*, pages 1–6. IEEE, 2006.

[81] Min Liu, Zhongcheng Li, Xiaobing Guo, and Eryk Dutkiewicz. Performance analysis and optimization of handoff algorithms in heterogeneous wireless networks. *IEEE Transactions on Mobile Computing*, 7(7):846–857, 2008.

[82] Mostafa Zaman Chowdhury, Won Ryu, Eunjun Rhee, and Yeong Min Jang. Handover between macrocell and femtocell for umts based networks. In *2009 11th International Conference on Advanced Communication Technology*, volume 1, pages 237–241. IEEE, 2009.

[83] Dionysis Xenakis, Nikos Passas, Lazaros Merakos, and Christos Verikoukis. Mobility management for femtocells in lte-advanced: Key aspects and survey of handover decision algorithms. *IEEE Communications surveys & tutorials*, 16(1):64–91, 2013.

[84] Dhanaraj Cheelu, M Rajasekhara Babu, P Venkatakrishna, and XZ Gao. User preferences and expert opinions based vertical handoff decision strategy with the inclusion of cost parameter for 4g networks. *International Journal of Autonomous and Adaptive Communications Systems*, 10(3):261–278, 2017.

[85] Dino Flore, Christopher Brunner, Francesco Grilli, and Vieri Vanghi. Cell reselection parameter optimization in umts. In *2005 2nd International Symposium on Wireless Communication Systems*, pages 50–53. IEEE, 2005.

[86] Hanane Fathi, Ramjee Prasad, and Shyam Chakraborty. Mobility management for voip in 3g systems: evaluation of low-latency handoff schemes. *IEEE Wireless Communications*, 12(2):96–104, 2005.

[87] Andreas Lobinger, Szymon Stefanski, Thomas Jansen, and Irina Balan. Coordinating handover parameter optimization and load balancing in lte self-optimizing networks. In *2011 IEEE 73rd vehicular technology conference (VTC Spring)*, pages 1–5. IEEE, 2011.

[88] Koichiro Kitagawa, Toshihiko Komine, Toshiaki Yamamoto, and Satoshi Konishi. A handover optimization algorithm with mobility robustness for lte systems. In *2011 IEEE 22nd international symposium on personal, indoor and mobile radio communications*, pages 1647–1651. IEEE, 2011.

[89] Won-Yeol Lee and Ian F Akyildiz. Spectrum-aware mobility management in cognitive radio cellular networks. *IEEE Transactions on Mobile Computing*, 11(4):529–542, 2011.

[90] Cheng-Chung Lin, Kumbesan Sandrasegaran, Huda Adibah Mohd Ramli, and Riyaj Basukala. Optimized performance evaluation of lte hard handover algorithm with average rsrp constraint. *arXiv preprint arXiv:1105.0234*, 2011.

[91] Haijun Zhang, Wenmin Ma, Wei Li, Wei Zheng, Xiangming Wen, and Chunxiao Jiang. Signalling cost evaluation of handover management schemes in lte-advanced femtocell. In *2011 IEEE 73rd vehicular technology conference (VTC Spring)*, pages 1–5. IEEE, 2011.

[92] Li Qiang, Jie Li, and Corinne Touati. A user centered multi-objective handoff scheme for hybrid 5g environments. *IEEE Transactions on Emerging Topics in Computing*, 5(3):380–390, 2016.

[93] Rami Ahmad, Elankovan A Sundararajan, Nor E Othman, and Mahamod Ismail. Handover in lte-advanced wireless networks: state of art and survey of decision algorithm. *Telecommunication Systems*, 66(3):533–558, 2017.

[94] Xiaohui Zhao, Hanyang Ma, Yuan Jin, and Jianguo Yao. Measuring instability of mobility management in cellular networks. *IEEE Network*, 32(5):138–144, 2018.

[95] Aruna Balasubramanian, Ratul Mahajan, and Arun Venkataramani. Augmenting mobile 3g using wifi. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 209–222, 2010.

[96] Christoph Paasch, Gregory Detal, Fabien Duchene, Costin Raiciu, and Olivier Bonaventure. Exploring mobile/wifi handover with multipath tcp. In *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design*, pages 31–36, 2012.

[97] Wei Dong, Swati Rallapalli, Rittwik Jana, Lili Qiu, KK Ramakrishnan, Leo Razoumov, Yin Zhang, and Tae Won Cho. ideal: Incentivized dynamic cellular offloading via auctions. *IEEE/ACM Transactions on Networking*, 22(4):1271–1284, 2013.

[98] Ashkan Nikravesh, Yihua Guo, Feng Qian, Z Morley Mao, and Subhabrata Sen. An in-depth understanding of multipath tcp on mobile devices: Measurement and system design. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 189–201, 2016.

[99] Jignesh S Panchal, Roy D Yates, and Milind M Buddhikot. Mobile network resource sharing options: Performance comparisons. *IEEE Transactions on Wireless Communications*, 12(9):4470–4482, 2013.

[100] Paolo Di Francesco, Francesco Malandrino, and Luiz A DaSilva. Mobile network sharing between operators: A demand trace-driven study. In *Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop*, pages 39–44, 2014.

[101] Markku Jokinen, Marko Mäkeläinen, and Tuomo Hänninen. Co-primary spectrum sharing with inter-operator d2d trial. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 291–294, 2014.

[102] Haneet Kour, Rakesh Kumar Jha, and Sanjeev Jain. A comprehensive survey on spectrum sharing: Architecture, energy efficiency and security issues. *Journal of Network and Computer Applications*, 103:29–57, 2018.

VITA

VITA

Haotian Deng was born in Shanghai, China. He received his B.E.(2013) in software engineering from Tongji University and M.S.(2015) in computer science and engineering from University at Buffalo, The State University of New York. He started his PhD program at The Ohio State University in 2015 and joined Department of Computer Science at Purdue University in 2017. He was a research intern at Alibaba Group in 2019. His research focuses on mobile networking and systems, particually on device-assisted network performance enhancmement and diagnosis in operational 4G/5G networks.