Resolving Policy Conflicts in Multi-Carrier Cellular Access

Zengwen Yuan[†*], Qianru Li[†*], Yuanjie Li[†], Songwu Lu[†], Chunyi Peng[‡], George Varghese[†]

[†]UCLA, [‡]Purdue University

ABSTRACT

Multi-carrier cellular access dynamically selects a preferred wireless carrier by leveraging the availability and diversity of multiple carrier networks at a location. It offers an alternative to the dominant single-carrier paradigm, and shows early signs of success through the operational Project Fi by Google. In this paper, we study the important, yet largely unexplored, problem of inter-carrier switching for multi-carrier access. We show that policy conflicts can arise between inter- and intra-carrier switching, resulting in oscillations among carriers in the worst case akin to BGP looping. We derive the conditions under which such oscillations occur for three categories of popular policy, and validate them with Project Fi whenever possible. We provide practical guidelines to ensure loop-freedom and assess them via trace-driven emulations.

1 INTRODUCTION

Multi-carrier cellular access[1] is a promising technology for improving the cellular service quality of a mobile device. It selects a preferred carrier from multiple carrier networks (e.g. T-Mobile, Sprint, AT&T, and others) deployed at a location. Given the diversity of deployed carriers and radio access technologies (RATs) at any location, multi-carrier access thus has the potential to offer improved coverage and access speed. It further provides a device-based solution without infrastructure upgrade at each carrier network. To date, Google has deployed the first multi-carrier access system in its *Project Fi* [14]. There are other reported efforts at Apple and Samsung [12, 13] and in the upcoming 5G [28].

Multi-carrier access uses a two-tier selection scheme. At the top tier, it allows the device to select and switch to a preferred carrier network (aka inter-carrier switching)[2]. At the bottom tier, it uses the conventional cell selection scheme (i.e., handoff [10, 16]) within a carrier to connect to the target cell (aka intra-carrier cell selection). The top tier inter-carrier selection is implemented at the device, while the low tier is enforced by each carrier at the network side. The two-tier decision calls for different mechanisms at each tier. While the intra-carrier handoff mechanism is already well specified and operated [10, 16, 31, 34, 35], inter-carrier switching is still largely unexplored.

In this paper, we study the problem of inter-carrier switching for multi-carrier access systems. We make a case for policy-based selection as the fundamental instrument for inter-carrier switching at the top tier. A multi-carrier service provider (MCSP) assigns each carrier with certain policy attributes, in the form of preference values or thresholds. These attributes reflect the policy decision that is left entirely up to the MCSP. At a given location, the MCSP uses these attributes to select the most preferred carrier. Policy-based switching possesses several appealing features, including preserving the autonomy and privacy of each carrier's operation, no need for runtime access to fine-grained cell-level information, and reuse of legacy standards for cell-level handoffs, etc. Moreover, such policy-based designs have been used in the operational Google Project Fi, as well as other Internet systems such as BGP [23, 24, 30] and data centers [27, 36, 38].

Despite possessing nice properties, policy-based switching also poses new design issues. We show that policy conflicts can arise between inter-carrier selection and internal cell-level handoffs within each carrier. Such conflicts force the device to oscillate between carriers, disrupt the device's network service, slow down performance, and drain battery at the device. The fundamental challenge is to coordinate between inter-carrier and intra-carrier policies. Each carrier wants to preserve its operation autonomy and privacy from other carriers and the MCSP. Moreover, carrier and cell level policies work at different granularities, and hence are hard to coordinate.

Similar issues were observed in BGP routing, whose flexible policies are considered indispensable by Internet ISPs. It is well known that, without sufficient conditions (e.g., Gao-Rexford [23]), BGP policies can result in routing oscillations. In this paper, we show that similar policy conflicts exist in a new setting. The policy-induced conflicts in BGP routing and multi-carrier cellular access exhibit a striking similarity. However, cellular conflicts may be worse than BGP, because both conventional carriers and the virtual operator of MCSP are active in defining policies from their own standpoint and enforcing them on the phone. Such conflicts incur excessive

*Co-first authors

[1]The "carrier" refers to a mobile network operator here. It is different from the "carrier signal" in a physical layer context. We also use this term with "multi-carrier access" interchangeably for brevity in this paper.

[2]Multi-carrier access can conceptually select more carriers for concurrent access at the device. However, the current practice is still to choose one carrier only, using a single SIM card and for better energy savings (§2).

MobiCom '18, October 29–November 2, 2018, New Delhi, India

ACM ISBN 978-1-4503-5903-0/18/10...$15.00
https://doi.org/10.1145/3241539.3241558

signaling, service disruption and energy spikes (§4.2). Our goal is to analyze when such conflicts arise, and to derive analogous (but very different from [23]) conditions to allow policy flexibility without conflicts.

We have studied three categories of common inter-carrier policies: preference-based, threshold-based, and hybrid. We model policy conflict as a stability problem. We derive the theoretical loop-free conditions for all categories, validate their existence using Project Fi, devise practical guidelines that address the above issues, and assess their effectiveness using trace-driven emulations. Our results are summarized in Table 1. Our key insight is that *to resolve policy conflicts, intra-carrier policy, which has been largely standardized and commonly practiced, should be prioritized over inter-carrier policy. Moreover, inconsistency among policies can be readily prevented by observing a few simple rules.*

The rest of the paper is organized as follows. §2 introduces multi-carrier access. §3 makes a case for policy-based inter-carrier switching, and §4 illustrates the policy conflicts and resulting switching loops. §5 overviews our methodology and results. §6, §7 and §8 provide detailed theoretical results and real-world validation for each category. §9 presents practical guidelines to ensure stable policies and §10 shows emulation-based assessment using Project Fi traces. §11 discusses remaining issues, §12 compares with related work, and §13 concludes the paper.

2 MULTI-CARRIER ACCESS PRIMER

Multi-carrier access leverages the diversity of multiple carrier networks at a given location. It chooses a favorable cellular carrier and a radio access technology(RATs) such as 3G or 4G. Fundamentally, multi-carrier access is based on the premise that a given location is typically covered by multiple carriers and their access qualities differ. It is appealing in several aspects. First, it provides better coverage. In reality, no single carrier can ensure complete coverage at any location [37]. Given that the device has the flexibility to switch among multiple carriers, the obtained coverage is the union of all carriers. Second, it offers better access speed. At a given location, one carrier may only support 3G while another has 4G LTE. The device thus benefits from access to the faster 4G LTE network. Third, it offers a device-based solution which is easier to deploy without changing carrier infrastructure.

Multi-carrier access in reality. The industry has deployed multi-carrier access systems such as Google Project Fi [14], Apple SIM [12], Huawei Skytone [8], and Samsung eSIM [25]. Most notably, since 2015, Project Fi has offered the first such service for the Nexus/Pixel phone models. It supports runtime switching between three U.S. 3G/4G carriers (T-Mobile, Sprint, and U.S. Cellular). Some ongoing standards [28, 39] seek to support multi-carrier access in 5G.

Current multi-carrier access is realized with a single reconfigurable SIM card and intra-carrier mechanisms that are readily available in commodity phones. Given only one

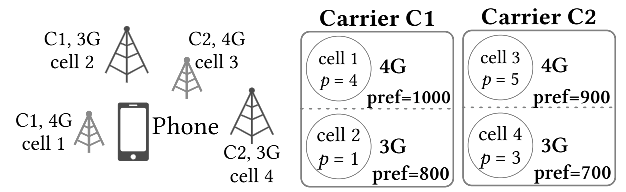


Figure 1: Policy-based inter-carrier switching example

cellular hardware interface, only one carrier is selected and used at a time. When a new carrier is selected, a system app reconfigures the SIM to use the new carrier's profile, so that the device can register with that carrier. Such inter-carrier switching decisions are policy-based (to be elaborated in §3). Afterward, it relies on the carrier's internal mechanism to select the cell, as elaborated below.

Intra-carrier handoff. The above scheme works together with the legacy single-carrier, cell-level selection called *intra-carrier handoff* [2–5]. Within each carrier, a 3G/4G *cell* offers radio network access through the base station to a geographical location.

In practice, each location can be covered by multiple cells. The handoff thus determines whether the device should move from the serving cell to another one, and which cell it should move to.

In 3G/4G, the serving cell controls handoff; each cell has its local handoff decision engine. The decision is based on the per-cell policy (including decision logic and parameters, e.g. per-cell priorities and thresholds for measures) and runtime measures. Once the handoff is executed, it switches to the new serving cell and starts another handoff decision iteration. Note that the local policies are *configurable* to meet diverse requirements, such as selecting the best radio quality, letting operators specify priorities for cells, etc. Some standardized intra-carrier handoff policies are described in §5.

3 THE CASE FOR POLICY-BASED INTER-CARRIER SWITCHING

We next make a case for policy-based selection as the fundamental component for the inter-carrier switching. In policy-based switching, each carrier is assigned certain policy attributes, in the form of a preference value or certain threshold-based forms for specific measures. These attributes reflect the multi-carrier service provider (MCSP, such as Google)'s policy demands (e.g., faster network, better coverage, and roaming agreements with carriers). At a given location, the MCSP uses these carrier attributes to select the most preferred carrier.

Figure 1 illustrates an example. There are two carriers (C1 and C2), and two cells (3G/4G) for each carrier at the given location. The MCSP uses a preference-based policy by specifying preference values for each RAT in each carrier, thus resulting in the preferred order (4G, C1) > (4G, C2) > (3G, C1) > (3G, C2). Given this policy, the MCSP first switches the device to carrier C1, since 4G in C1 is the most preferred choice. Within C1, cell 1 with higher priority ($p = 4$) is

Table 1: Classification of main results (NC: necessary condition; SC: sufficient condition)

Inter-Carrier Policy		**Theorem Results**				**Guide-line**	**Vali-dation**
Form	Subcategory	Reference	Insight	SC?	NC?		
Preference	RAT-aware	Thm. 6.1	Inconsistent preference on RAT	✓	✓	§9.1	§6.1
	RAT-oblivious	Thm. 6.2	Preference conflict w/ unavailability	✓	✓	§9.1	§6.2
Threshold	Inconsistent measures	Thm. 7.1 & 7.2	Some threshold criteria are loop-prone; Use min-measure rule for stability	✓	✓	§9.2	§7.1
	Inconsistent configs	Thm. 7.3 & 7.4	Some thresholds violate stability	No	✓	§9.2	N/A
Hybrid	Preference first	Thm. 8.1	Some threshold criteria are ruled out	No	✓	§9.3	§8
	Threshold first	Thm. 7.1–7.4	Same as threshold theorem	No	✓	§9.3	§8

selected based on the intra-carrier handoff policy. Note that cell 1 is a 4G cell in C1; this is consistent with the inter-carrier policy. In the example, the MCSP checks carrier-level preference only and leaves cell selection decision to carriers. This preserves the operation autonomy of each carrier.

Policy-based inter-carrier switching is needed by the MCSP for three reasons. First, the policy naturally arise at the MCSP level. The MCSP builds its service on top of individual cellular carriers, and has to balance among carriers for both technical concerns (e.g., select the best-performing carrier) and nontechnical interests (e.g., which one is a more favorable partner). Second, the policy issues further exhibit in operational practices, such as dealing with geographical diversities of carriers, or even traffic engineering when distributing cellular data traffic across carriers. Third, the policy allows the MCSP to make *configurable* decisions to accommodate diverse demands (e.g. faster network, better coverage, and partner preference).

The policy-based switching further offers several nice properties. First, it decouples choosing a carrier network from cell selection within the carrier. Consequently, MCSP only needs coarse-grained information on carriers rather than the fine-grained cell information within each carrier network[3]. Second, it leverages the largely standardized intra-carrier mechanism, and keeps the policy design simple at the carrier level. It thus does not require the standardization process again. Last but not the least, it preserves the autonomy and privacy of each individual carrier network. An MCSP works with the carriers without mandating the disclosure of the operational practices of these cellular carriers.

Examples of policy-based switching. We identified three common forms of policy:

- *Preference-based switching.* At each location, the MCSP assigns a local preference value to each carrier. The carrier with the highest preference is selected assuming the same other conditions.
- *Threshold-based switching.* At a given location, the MCSP takes a threshold-based form on certain performance metrics (e.g., latency, throughput or a mix) as each carrier's policy attribute. When the threshold conditions are met, a new carrier would be selected. The goal is to find a carrier, which is better than the serving carrier and meets the threshold requirements.
- *Hybrid switching.* The carrier attributes are specified in the form of local preference *and* threshold forms.

The above forms of policy attributes are simple enough to realize, but still generic enough to cover many practical use cases. Similar forms have also been used in other operational networking systems. The most notable example is that the Internet BGP routing has used the preference attribute in its inter-domain route selection [23, 24]. The preference-based policies are also used in intra-carrier handoff management [2, 5, 31, 34] and WiFi AP selection [19, 20] (in latest Android/Linux). The threshold-based forms are also the common practice for intra-carrier handoff management [2, 5, 31, 34] and WiFi AP selection [9, 17]. As we will show later, the major difference between our form and these efforts is the conflicts with the intra-carrier policy.

Operational system in reality. We have observed that, Google, as a virtual operator of MCSP, has largely adopted policy-based switching when making the inter-carrier selection in its Project Fi. From the Android logs on Pixel/Nexus phones running Project Fi, we confirm that both preferences and thresholds forms are used when selecting a preferred carrier network by Google. Moreover, its recent machine learning-based switching module uses a variant of the threshold-based policy.

Specifically, Project Fi uses a monitor-controller architecture. Each monitor tracks some metrics in parallel and proposes a target carrier to switch to. The controller receives these proposals and decides the target carrier. Notable monitors include a *PoorNetwork* monitor (labeled as `PNP` in `logcat`), which assigns *preferences* on carriers and RATs to facilitate the carrier selection. A *GeoLocation* monitor (labeled as `Flock`) uses the crowdsourced carrier quality data to perform *pairwise comparisons* on target carriers. The newest version also includes a machine learning-based monitor (labeled as `K2so`) that predicts carriers' quality and uses *thresholds* for decisions. Therefore, both preference and

[3] It may be infeasible for an MCSP to access the fine-grained, cell-level information at runtime across all carriers. This has been the practice by Google Project Fi. The current hardware will not allow for the device to obtain all cell-level measurements and metrics unless registered to the carrier. The device has to constantly scan and switch to all available carriers to collect such detailed information; this incurs service outage.

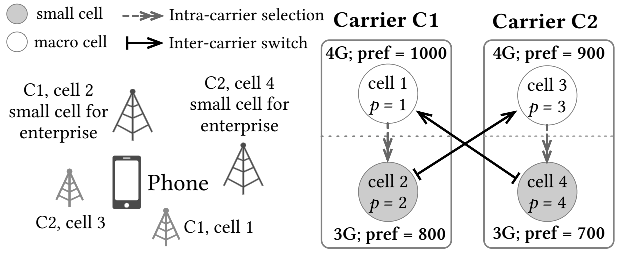


Figure 2: Example of policy conflicts and bad impact

threshold-based policies are adopted in Project Fi's design. Given certain conditions, it may use only one, or both. For example, when location service is disabled, only *PoorNetwork* monitor remains active so the policy is preference only. If *GeoLocation* monitor is active, *PoorNetwork* monitor's decision is usually overshadowed, so effectively only the threshold policy is used.

4 IMPROPER INTER-CARRIER POLICY

The policy-based inter-carrier switching is necessary for an MCSP and possesses appealing features. However, improper policy practice may also yield unexpected behaviors such as loops. In this section, we show an example to illustrate the incurred issues as well as their impacts.

4.1 An Illustrative Example

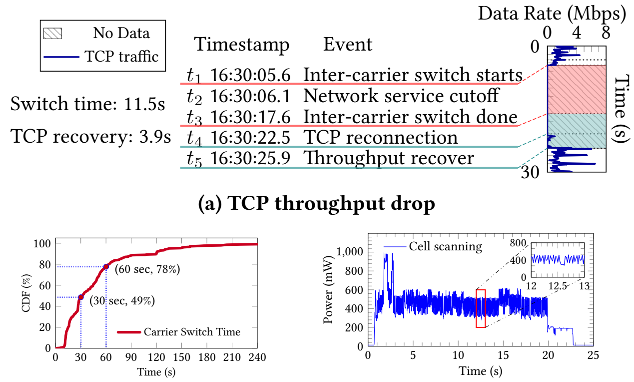
We now show an example to illustrate the policy conflicts and potential negative effects. Consider the scenario in Figure 2. It is an office building environment with two carriers C1 and C2, with two deployed cells belonging to each. The phone remains static with constant wireless channel conditions. It uses multi-carrier cellular access to the two carriers. The inter-carrier policy takes the preference-based form. Given the preference values for each RAT in C1 and C2, the preferred order is given by: (4G, C1) > (4G, C2) > (3G, C1) > (3G, C2). This is a sensible policy by MCSP. It is well grounded at the inter-carrier policy level: 4G is favored over 3G, while carrier C1 is favored over C2 since C1 has generally better performance (e.g., higher access speed). On the other hand, the cell-level policy at the intra-carrier level uses the priority-based policy. In carrier C1, the 4G cell 1 is set with priority $p = 1$, whereas its 3G cell 2 has priority $p = 2$. This is because cell 2 is a deployed urban/enterprise small-cell in the office building that seeks to offload the traffic for local users from the macro-cells. Note that small cells are indeed quite common. Recent data [21] predicts the deployed small cells will reach 11.4M by 2025 with 14% annual growth rate. The rate increases to 36% in nonresidential areas. Similarly, in carrier C2, its 3G cell 4 (also an urban small cell) is assigned a higher priority value 4 than its 4G cell 3. Within each carrier, the intra-carrier cell policy is also well justified.

Policy conflicts then arise between inter-carrier and intra-carrier levels. The inter-carrier level prefers 4G RAT over 3G RAT for better technology and higher access speed, whereas the intra-carrier level favors 3G over 4G for better traffic offloading. *Both are well justified by their own interests based on their knowledge.* The conflict is neither the fault of the MCSP nor the issue of an individual carrier. Instead, it is rooted in the distributed nature of making policy decisions at the MCSP and each carrier. The MCSP uses the carrier-level information only and sets its preference based on what RAT is superior and which carrier offers better performance. Within each carrier, the carrier sets its policy to consider the unique small-cell deployment in the example setting.

The above policy conflicts also result in unexpected behaviors. If the MCSP *strictly* enforces its inter-carrier policy (4G, C1) > (4G, C2) > (3G, C1) > (3G, C2), it will get stuck into *persistent* loops because of its policy conflicts with the intra-carrier policies. The device first switches to carrier C1 based on the inter-carrier preference (4G, C1). However, cell 2 is selected since this cell has higher intra-carrier priority, once the device is in carrier C1. Unfortunately, cell 2 is a 3G cell, but not a 4G cell. Since this is not what the inter-carrier policy dictates, the device goes back to the carrier level. It then selects carrier C2 after the selection failure in C1. Once in C2, the 3G small cell 4 is also chosen for higher priority among the two cells. This is also not what the inter-carrier policy wants. It then repeats the above steps and gets into the persistent loop (4G, C1) $\mapsto$ (3G, cell 2) $\mapsto$ (4G, C2) $\mapsto$ (3G, cell 4) $\mapsto$ (4G, C1) $\mapsto$ . . . . Note that, despite the existence of 4G RATs in both carriers, neither is selected. The inter-carrier policy mandates the continuous search to hopefully settle down at one 4G cell.

Note that it is possible to mitigate the impact of such loops without fixing the inter-carrier policy conflicts. However, this practice has undesirable side effects. Consider two intuitive options at the phone: (a) disable the inter-carrier switching after trying several rounds, and settle with C1's 3G (e.g., via logging switch history or limiting maximum attempts), thus stopping its inter-carrier policy enforcement. This choice directly contradicts with the goal of obtaining 4G access. It also does not allow the device to switch to a better carrier if it is available in the future. (b) disable the 3G access on the phone, so that the device may settle with the 4G Cell 1 in C1. This seems to honor the inter-carrier policy. However, it is against the intra-carrier policy for small cells within carrier C1. It is also not a good option for the device and the user, since it unnecessarily eliminates the 3G option and constrains the selection flexibility.

For the above example scenario, the best option is to switch to carrier C1 (preferred over C2 based on inter-carrier policy) but select the 3G cell 2 (that is favored based on the intra-carrier policy for small cells). This sheds lights on the simple rule that helps to resolve the policy conflicts: *Upon policy conflicts, intra-carrier policy should be prioritized over the inter-carrier policy in the resolution process.* This intuitive rule is also consistent with the two-tier switching scheme. At the carrier level, the MCSP uses policies to specify the general preference, but may not have the accurate information (e.g., small-cell deployment), which is only accessible within the carrier. Therefore, whenever conflict arises, intra-carrier



(a) TCP throughput drop

(b) Service interruption time (c) Modem power consumption

Figure 3: Impact of inter-carrier switch

Trace 1 A loop in Google Project Fi

```
1  14:32:07 Current network: Sprint EHRPD.
2           Already waited for 02:58, need to wait for 27:02 more.
3  14:59:17 Switch request Sprint->T-Mobile is approved. Requester: PNP
4  15:00:09 Switch done. Current network: T-Mobile HSPA.
5  15:00:09 Reset monitor. Elapsed time: 43:29, locked until 6:43:30.
6  21:17:20 Unlock switch. Current network: T-Mobile HSPA.
7           Already waited for 00:00, need to wait for 2:00:00 more.
8  23:18:08 Switch request T-Mobile->Sprint is approved. Requester: PNP
9  23:18:25 Switch done.  Current network: Sprint EHRPD.
10 23:18:25 Reset monitor. Elapsed Time: 9:05:30, locked until 15:01:47.
11 ...
```

policy, which is well defined and practiced by individual carriers, should be prioritized first.

Real-world instance. The above example is conceptual; however, we did observe real instances in Google Project Fi that can be mapped to this example. Trace 1 shows such an example loop when the user remains static[4]. Google sets the preference values for the four available RATs as follows: $P_{T,LTE} = P_{S,LTE} = 1000$ for both T-Mobile and Sprint's LTE, $P_{T,3G} = 700$ for T-Mobile HSPA (3G), and $P_{S,3G} = 800$ for Sprint EHRPD (3G). In the setting, LTE signals in both carriers are weak, and the phone is camped on HSPA in T-Mobile or EHRPD in Sprint. As shown in the trace, the loop Sprint $\mapsto$ T-Mobile $\mapsto$ Sprint is observed, because the inter-carrier policy keeps on switching to the carrier with the highest-preference RAT but could not stay.

Note that Project Fi has implemented engineering techniques to limit such switching frequency. It records switch history and uses a timer to upper-bound the loop frequency (Lines 2, 5, 7, 10), once all carriers have been tried out, similar to Option (a). Such a fix prolongs the period of a loop but without completely eliminating the loop. The loop recurs upon timeout. It further incurs the side effect of letting the device being stuck in a network (detailed in §6.1). We will show more real-world loop instances in §6 and §7. In summary, we believe the scientific approach of eliminating the loop in the first place is the better way to go. With proper policy coordination, we confirm we can do it.

[4]All traces are collected in the latest Project Fi V3-universal.

4.2 Real Impact of Inter-Carrier Loop

Inter-carrier loop disrupts user's cellular service, incurs battery drains at the device, and triggers more signaling on carriers. Its impact aggravates as the loop persists. Moreover, the switching can happen during both idle and active states. We confirm that, in the latest Project Fi's version, a binary parameter[5] is used to enable inter-carrier switching as long as the device uses cellular data.

The phone loses its cellular data and voice service during the switching[6]. Figure 3b shows the time taken by a single switching in Project Fi, from our small-scale user study[7]. About 51% of the records took 30 seconds or more, while 22% of the switchings took more than one minute. TCP throughput tanked during the switching as shown in Figure 3a.

The battery consumption hikes (could be 3× higher than the idle mode [22, 26]). This is rooted in intra-carrier design; phone exhaustively scans cells and keeps the radio on during the switching. See Figure 3c for the power consumption[8]. The average power draw is around 400 mW during cell scanning (a major phase of carrier switching), significantly higher than the idle state. Furthermore, the phone exchanges signaling messages with *every* carrier's *every* RAT it can reach, incurring excessive signaling overhead [31].

4.3 Frequency of Inter-Carrier Loops

The frequency of the inter-carrier loops can be gauged both *temporally* and *spatially*. The temporal frequency is the elapsed time (aka the period) of a loop. It ranges from roughly 8 minutes (in Traces 2, 3 and 4) to over 2 hours (in Trace 1) in Project Fi. Our code analysis of Project Fi shows that, this loop duration is limited by a lockdown timer, which ranges from 5 minutes to 2 hours by default. Once an inter-carrier switching has been executed, the timer forbids any inter-carrier switching in between. This timer prolongs the loop interval and mitigates its impact, but does not eliminate the loop. It may let the device be stuck in a network and reduce the flexibility to select a better carrier when it is available. For example, if the signal becomes worse, the device cannot switch to a better carrier. More details are in Traces 2 (§6.1).

The spatial frequency defines how often the loop is observed geographically. This metric is decided by both the inter-carrier policy form and the carrier's signal coverage. Not all locations incur inter-carrier switching loops. Using Project Fi's coverage and each carrier's actual policy, we show in our emulations (§10) that loops may occur at between 0.003% and 6.16% locations. It gives a lower estimate since indoor signal conditions are more complex.

[5]The variable is called "`allow_switching_if_using_cellular_data`".
[6]The device may still be able to access the Internet via WiFi, but this could be an issue when WiFi is unavailable (e.g. outdoor environment).
[7]It shows 350 records, spanning from 2017/02 to 2018/03 on four phone models that support Project Fi: Nexus 6/6P and Pixel/Pixel 2. All data are collected anonymously and comply with the IRB regulations.
[8]The measurement is conducted on Samsung S5 with minimal background service in comparison with the energy consumption for the airplane mode; the comparison is similar on all phone models.

5 METHODOLOGY AND OVERVIEW

We take a three-step approach to studying the policy conflicts and loops. We model the inter-carrier policy (§5.1) and derive the theoretical stability conditions (§5.3). We further show empirical validations from Project Fi (5.2). We propose practical guidelines for provable stability and assess them via emulations (§9–10).

5.1 System Model

In our system setting, there are a virtual operator of MCSP, several participating carriers for the MCSP, and many users that obtain cellular access via the MCSP. Similar to Google Project Fi, the MCSP installs and runs a software module on the user device with its two-tiered selection procedure. It first selects a preferred carrier for the device via its *inter-carrier switching* based on the MCSP's policy. Once in a carrier, it connects to a target RAT/cell using the *intra-carrier policy* mandated by the carrier.

The MCSP does not have access to the detailed cell-level policy information of each carrier, but makes its decisions based on its coarse-grained knowledge (e.g., what RATs each carrier uses at a given location). This enables each carrier to preserve its operation autonomy and policy privacy from the virtual operator (e.g., Google). To gain key insights, we consider static users and deterministic policies in this paper, while leaving more complex cases to the future work.

Inter-carrier switching. Consider N carriers $C_1, C_2, \ldots, C_N$ at the user's current location. Each carrier has K radio technologies, denoted as $RAT_1, RAT_2, \ldots, RAT_K$. There are $n(i)$ cells in carrier C_i: $c_i^1, c_i^2, ..., c_i^{n(i)}, i \in [1, N]$. Within each carrier, intra-carrier policy selects the serving cell for the device. An *inter-carrier switching* is the transition from one carrier C_i to another carrier C_j specified by the inter-carrier policy at the mobile device. Therefore, We model such a switching as a discrete transition $C_i \mapsto C_j$.

Intra-carrier policy. There are two types of intra-carrier policies in LTE [4, 5]: *Idle-state policy* that is used when there is no active radio connectivity and *active-state policy* that is used otherwise. In multi-carrier access, only the idle-state policy should be considered, because inter-carrier switching occurs in idle-state only (by deregistering from the old carrier and registering to the new carrier)[9]. The idle-state policy is based on the per-cell priority and threshold of measures. We abstract the policy from 3GPP standards [2, 4, 5]: The intra-carrier policy moves the device from cell c^u to c^v *iff*: (1) using *absolute* value: $q(c^v) > Thresh1^{u,v}$ if $p(c^v) > p(c^u)$; (2) using *direct* comparison: $q(c^v) > q(c^u) + \Delta^u$ if $p(c^v) = p(c^u)$; (3) using *indirect* comparison: $q(c^u) < Thresh2^u, q(c^v) > Thresh3^{u,v}$ if $p(c^v) < p(c^u)$. This policy enumeration facilitates our guideline derivation in §9.

Assumptions. We assume a static setting where the user does not move. All cells' performance metrics (e.g. radio signals, latency, throughput, . . .) remain unchanged. Our results can still be generalized if such assumptions do not hold (discussed in §11). We further assume proper, unchanged intra-carrier policy, without incurring loops within each carrier (e.g. via prior results [31]). The device initially is connected to a carrier C_0's RAT_0[10]. It performs inter-carrier switching only when the intra-carrier reselection stabilizes, and uses specific inter-carrier policies to be elaborated next. We further assume the inter-carrier policy is deterministic, with random policy being beyond the scope of this work.

Table 2: Notations

C_i	Carrier i, $i \in [1, N]$
RAT_j	Radio access technology j (e.g. 3G, 4G)
c^k/c_i^k	Cell k (in carrier C_i)
$P_{i,j}/P_i$	Inter-carrier preference on carrier C_i's RAT_j / C_i
$p(c^i)$	Intra-carrier priority of cell c^i
$M, M(C_i)$	Measure M (on C_i) for inter-carrier policy
$Q, q(c^j)$	Measure Q (on c^j) for intra-carrier policy
δ, θ, ϕ	Different inter-carrier thresholds (on carrier)
$\Delta^i, Thresh^{i,j}$	Different intra-carrier thresholds (on c^i/c^j)

Loops and stability. The inter-carrier policy can incur consecutive switchings even under the assumed static condition (§4.1). Formally, an N-carrier loop is an inter-carrier switching sequence, starting from one initial carrier, traverse each carrier *exactly once*, and then switch back to the same initial carrier. For example, switching sequence $C_1 \mapsto C_2 \mapsto \cdots \mapsto C_N \mapsto C_1$ is one instance of N-carrier loop. The *order* of the sequence matters, for example, sequence $C_1 \mapsto C_3 \mapsto C_2 \mapsto C_1$ is a different loop to $C_1 \mapsto C_2 \mapsto C_3 \mapsto C_1$. An N-carrier loop is *persistent* when single instances of N-carrier loop happen repetitively under the *same static* condition. We have the following result (proof in §A.1 in [6]):

Proposition 1. *An N-carrier loop is persistent loop under the static condition and deterministic policy.*

An inter-carrier policy is *stable* iff it will not incur persistent loops. In the following sections, we will derive the theories and guidelines for the inter-carrier policy stability.

5.2 Real-World Validation

We further use Google Project Fi to validate our results in reality. To reconstruct Project Fi's main logic and policies, we collect Android logs that record its decisions and activities. We further validate our findings via limited reverse engineering and online user forum reports. We set specific conditions to make Project Fi's policy consistent with each subcategory. We then observe loop scenarios and compare the empirical findings with our analysis results.

5.3 Roadmap and Overview

This work explores the theoretical conditions and practical guidelines for the policy conflicts (loops) in multi-carrier

[9]As a real example, Project Fi will suspend the inter-carrier switching until the device completes calls or data session and moves back to idle state.

[10]C_0 could be any of the $C_1, C_2, \ldots, C_N$ and RAT_0 could be any of the $RAT_1, RAT_2, \ldots, RAT_K$.

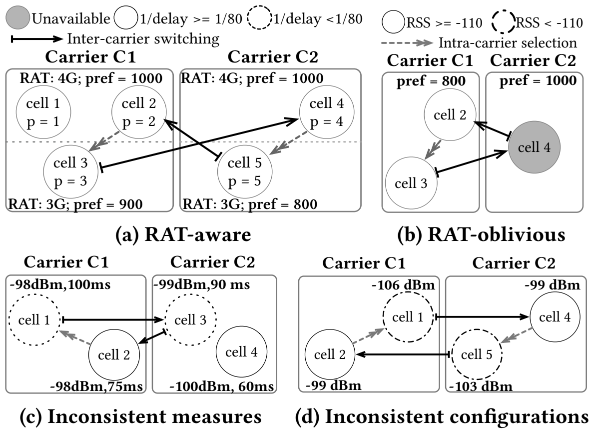


Figure 4: Classification of policy conflicts and loops

access. Figure 4 and Table 1 classify the conflicts based on their causes. Such conflicts can arise from the preference-based, threshold-based, and hybrid inter-carrier policies. We overview each category, examine how it conflicts with the intra-carrier policy, and summarize our results.

Preference-based policy (§6). In this category, the MCSP's inter-carrier preference settings contradict with carriers' priorities for the same carrier or RAT (exemplified in §4.1). Based on the granularity of the preferences that MCSP uses, there are two sub-categories:

- *RAT-aware preference* (§6.1): The inter-carrier policy assigns a preference to each (carrier, RAT) pair (exemplified in Figure 4a and Figure 2). We show that, the stability can be violated when the MCSP's inter-carrier preferences contradict with the carriers' internal priorities.
- *RAT-oblivious preference* (§6.2): The inter-carrier policy assigns a preference to each carrier only. The stability is violated if the inter-carrier preferences conflict with intra-carrier policies on cells that could not provide service (exemplified in Figure 4b, more details in §6.2).

Threshold-based policy (§7). When the MCSP uses the threshold-based policy, it may conflict with the intra-carrier policies and incurs loops in two scenarios:

- *Inconsistencies of measures* (§7.1): The inter-carrier and intra-carrier policies evaluate the same carrier using different types of measures. This could happen since the MCSP and carriers may target different goals (e.g., latency v.s. radio quality, as exemplified in Figure 4c). We show that, some threshold-based evaluation criteria are loop-prone. Moreover, if measures are independent, the necessary *and* sufficient condition for the stability is that the MCSP applies the *minimum measure rule.* If they are correlated, our theorems are still sufficient, but not necessary.
- *Inconsistencies of configurations* (§7.2): Even if the inter-carrier and intra-carrier policies evaluate the same measures, they can conflict with each other due to uncoordinated threshold values. Figure 4d illustrates an example: Under constant and static measures, the inter-carrier switching and intra-carrier handoffs are triggered simultaneously, thus incurring loops. To ensure stability, we derive a set of necessary conditions for different criteria for threshold coordination. The key result is that, such coordination can be performed using *aggregated threshold values* rather than fine-grained thresholds. This simplifies the coordination and prevents carriers from exposing its internal policies to MCSP.

Hybrid policy (§8). When the MCSP uses both preferences and thresholds, we show how above results can be generalized. There are two general approaches to combine the preferences and thresholds: (1) *Preference-first policy:* It evaluates each carrier's preference first, then applies different threshold-based criteria based on the preference relations. We prove that the use of preferences poses more constraints on choosing the threshold-based criteria; (2) *Threshold-first policy:* It applies the same threshold-based criteria to all carriers, and select one with the highest preferences. We show that the results in threshold-based policy still hold here.

6 STABILITY FOR PREFERENCE POLICY

We first study the preference-based inter-carrier policies.

6.1 RAT-Aware Preferences

6.1.1 Policy Form. The MCSP assigns a preference $P_{i,j}$ to carrier C_i's RAT_j (exemplified in Figure 2). It seeks to *select a most preferred carrier* according to such preference list[11]. Let P^i_{max} be the maximum RAT preference in C_i. A simple RAT-aware policy is as follows.

Policy 1 (RAT-aware inter-carrier switching). Let C_i be the serving carrier. Perform inter-carrier switching $C_i \mapsto C_j$ and mark C_j as selected, if (a) $P_{j,k} = P^j_{max} > P_{i,m}, j \neq i$; and (b) C_j has not been selected. When all highest preferred carriers have been selected, clear the marks to allow flexibility.

Policy 1 covers a wide spectrum of RAT-aware inter-carrier policies. For instance, one may want to select a carrier with its preference *higher* than the current one, rather than the carrier with the highest preference. A minimum acceptable preference is also needed then. This is equivalent to setting all carriers above such "minimum preference" with *equal* highest preference values. Policy 1 still applies.

6.1.2 Stability Condition. The stability is violated by the conflicts between inter-carrier preferences and the intra-carrier priorities. To unveil the concrete conflict form, we first prove the following result (proof in §A.2 in [6]):

Lemma 1. Assume preference satisfies $P^1_{max} \geq P^2_{max} \geq \cdots \geq P^N_{max}$, where $P^i_{max} = \max_j P_{i,j}$. An N-carrier loop occurs *iff* the inter-carrier switching sequence $(*)$ $C_1 \mapsto C_2 \mapsto \cdots \mapsto C_N \mapsto C_1$ occurs.

Lemma 1 shows that the ordering of switch sequence in an N-carrier loop follows the preference order. This inspires the N-carrier loop condition below (proof in §A.3 in [6]):

[11] We allow the same preference value for different (carrier, RAT) pairs. Certain tie-breaking rules (e.g., smaller index on carrier first and RAT next) may apply.

Trace 2 Persistent loop by RAT-aware preference

```
1      14:19:47 Lock timer expired. Current network pref: 900.
2      14:19:47 Switch request to Sprint is approved. Requester: PoorNetwork.
3      14:20:48 Switch T-Mobile LTE -> Sprint 1xRTT done.  result:Success.
4  ❶            srcSignalStrength:-101. destSignalStrength:-106.
5      14:20:49 Switched to a worse network. Switch request back to T-Mobile.
6      14:21:01 Switch Sprint 1xRTT -> T-Mobile LTE done.  result:Success.
7               srcSignalStrength:-106. destSignalStrength:-101.
8      14:21:01 Locking plugin until 07:57. Current elapsed time: 04:57
9      14:29:02 Lock timer expired. Current network pref: 900.
10     14:29:02 Switch request to Sprint is approved. Requester: PoorNetwork.
11     14:29:58 Switch T-Mobile LTE -> Sprint 1xRTT done.  result:Success.
12 ❷            srcSignalStrength:-100. destSignalStrength:-105.
13     14:29:58 Switched to a worse network. Switch request back to T-Mobile.
14     14:30:10 Switch Sprint 1xRTT -> T-Mobile LTE done.  result:Success.
15              srcSignalStrength:-106. destSignalStrength:-200.
16     14:30:10 Locking plugin until 17:07. Current elapsed time: 14:07
17     14:38:32 Lock timer expired. Current network pref: 900.
18     14:39:43 Switch T-Mobile LTE -> Sprint 1xRTT done.  result:Success.
19 ❸            srcSignalStrength:-100. destSignalStrength:-104.
20     14:39:44 Switched to a worse network. Switch request back to T-Mobile.
21     14:40:00 Switch Sprint 1xRTT -> T-Mobile LTE done. result:Success.
22              srcSignalStrength:-104. destSignalStrength:-200.
```

Theorem 6.1 (Inter/intra-carrier preference conflict). *Assume the inter-carrier switching takes Policy 1. A persistent N-carrier loop happens* iff. *(a) Every carrier has one or more RATs (denoted RAT_H) assigned with equal, highest preference by the MCSP; and (b) Every carrier's intra-carrier priority and threshold results in reselection from RAT_H to a different RAT_L.*

Theorem 6.1 explains how inter-carrier preferences on RAT contradict with the intra-carrier priorities: The inter-carrier policy will seek C_i's RAT_H, but since C_i's intra-carrier policy move to RAT_L, inter-carrier will switch to another carrier C_j.

6.1.3 Validation. We validate that, persistent loops occur between two carriers, T-Mobile and Sprint, in Google Project Fi. Trace 1 in §4 shows an instance. The loop Sprint ↦ T-Mobile ↦ Sprint is triggered because inter-carrier policy prefers LTE equally but neither carrier can stay in LTE.

We further confirm that, the lockdown timer in Project Fi does not eliminate the loop (Trace 2). Instead, the timer strikes a balance between temporal loop frequency and flexibility to switch to a better carrier. The loop frequency depends on the timer value. When we reduce the lockdown timer value to five minutes, loops occur more frequently, about once every ten minutes (Loops ❶, ❷, and ❸). Moreover, Google's implementation does not eliminate loops and may let the device be stuck in the no-service state. When the T-Mobile LTE cannot offer services (signal strength = −200 dBm at Lines 17 and 25), the phone is forbidden from inter-carrier switching by the timer. The phone thus suffers from a five-minute service outage. With a larger timer value, the outage could be even longer.

6.2 RAT-Oblivious Preference List

We next analyze the RAT-oblivious preference, and discuss its relation with the RAT-aware preferences.

Trace 3 Persistent loops by RAT-oblivious preference

```
   08:14:30 User activity type is STILL, confidence: 100
   08:14:34 Switch request to T-Mobile is approved. Requester: Flock
   08:16:56 Switch Sprint CDMA -> T-Mobile UNKNOWN done. result: TimedOut.
            srcSignalStrength:-117. destSignalStrength:-200.
❹  08:16:57 Wait for 05:00 before attempting another switch.
   08:22:18 Switch request to Sprint is approved. Reason: signal loss.
   08:22:46 Switch T-Mobile UNKNOWN -> Sprint CDMA done. result: Success.
            srcSignalStrength:-200.  destSignalStrength:-117.
   08:22:56 Switch request to T-Mobile is approved. Requester: Flock.
   08:24:56 Switch Sprint CDMA -> T-Mobile UNKNOWN done. result: TimedOut.
            srcSignalStrength:-117.  destSignalStrength:-200.
❺  08:24:59 Wait for 05:00 before attempting another switch.
   08:30:27 Switch request to Sprint is approved. Reason: signal loss.
   08:31:06 Switch T-Mobile UNKNOWN -> Sprint CDMA done.  result: Success.
            srcSignalStrength:-200.  destSignalStrength:-117.
```

6.2.1 Policy Form. The MCSP assigns a preference value P_i to carrier C_i, and still *selects a most preferred carrier*. The RAT-oblivious preference policy is specified as follows.

Policy 2 (RAT-oblivious inter-carrier switching). Perform inter-carrier switching to the highest preference carrier that has not been selected if (a) the serving carrier's preference is not the highest; or (b) the serving carrier is unable to provide cellular services (defined as "unavailable")[12]. When the serving carrier is not usable but all other carriers have been selected, clear the marks.

Note that, Policy 2 is similar to RAT-aware Policy 1. But it is not a subset of that. When the device switches to a carrier by the MCSP's policy, it may not get available service.

6.2.2 Stability Condition. Intuitively, we can draw a similar conclusion to Theorem 6.1. If every carrier may move the device to an unavailable cell, then the inter-carrier policy will keep trying and may form a loop. Theorem 6.2 confirms this intuition and is proved in §A.4 in [6].

Theorem 6.2 (Cell unavailable loop). *An N-carrier loop occurs* iff. *the intra-carrier logic in all carriers moves the device to an unavailable cell.*

Remark. Theorem 6.2 gives the sufficient and necessary condition for N-carrier loop assuming Policy 2.

6.2.3 Validation. In Project Fi, Google distributes a RAT-oblivious preference list to its phone. The switching logic is similar to Policy 2. We have validated the existence of this subcategory. As shown in Trace 3, two loops ❹ and ❺ are observed while the phone is placed statically. For each loop, the phone first switches from Sprint to T-Mobile, due to Project Fi's RAT-oblivious policy that prefers T-Mobile over Sprint. However, T-Mobile has weak signal coverage at the spot (Project Fi records signal strength as −200 dBm). The phone consequently switches back to Sprint upon the 5-min timeout by following Policy 2. This forms Loop ❹.

[12]This could happen due to the bad signal, access denial for base station congestion [1, 4, 7], and RRC connection rejection [1], etc. The MCSP may also decide its unavailability in its policy. The usable cell signal strength is above −140 dBm by 3GPP standards [5]). In Project Fi, an LTE network is deemed not usable if cell signal strength is below the threshold −125 dBm.

After both carriers have been selected, Project Fi resets the counter, and proceeds with choosing its preferred T-Mobile but falling back to Sprint in Loop ❺. We show two loops in the trace due to space limit, but there is no sign of stop. Note that, Project Fi may prolong the loop frequency by regularly resetting the hour-long lock timer, but the lock policy remains unchanged for each day.

7 STABILITY FOR THRESHOLD POLICY

We next study the threshold-based inter-carrier policies.

7.1 Inconsistency of Measures

7.1.1 Policy Form. We consider the following policies.

Inter-carrier policy. An easy option for carrier selection is to *find a carrier whose measure is better than the serving carrier.* Denote the serving carrier's measure as $M(C_s)$, target carrier's measure as $M(C_t)$, and thresholds as δ, θ, and ϕ (all > 0). One may enumerate four basic yet orthogonal comparisons:

F1. $M(C_t) > \theta$ (candidate's measure is higher than threshold)
F2. $M(C_s) < \theta \wedge M(C_t) \geq \phi$ (serving carrier's measure is lower than a threshold, and candidate's measure is higher than another threshold)
F3. $M(C_t) > M(C_s) + \delta$ ($\delta \geq 0$; candidate's measure is offset higher than the serving carrier's)
F4. $M(C_s) < \theta \wedge M(C_t) > M(C_s) + \delta$ ($\delta \geq 0$; serving carrier's measure is lower than a threshold, and candidate's measure is offset higher than the serving carrier's)

Given these criteria, the inter-carrier policy performs the switching $C_i \mapsto C_j$ when C_i and C_j's measures satisfy criterion (F^*) from *F1 – F4*. More complex comparisons can be viewed as the combinations of the criteria above.

Measures of carriers. Assume the inter-carrier policy uses the measure type M, while the intra-carrier policy uses the measure type Q ($Q \neq M$). Denote $M(C_j)$ as the measure M of carrier C_j, $M(c_j^u)$ as the measure of cell c_j^u in C_j, and $M^{min}(C_j) = \min M(c_j^u)$. The MCSP will compute the per-carrier measure C_j based on the per-cell measures $\{M(c_j^u)\}$.

7.1.2 Stability Condition. We first show that, some criteria are inherently loop-prone and thus should not be used in *any* inter-carrier policies (proof in §B.1 in [6]):

Theorem 7.1 (Unstable comparison). *If inter-carrier policy takes Criterion* F1*, then the inter-carrier policy cannot be loop-free no matter how the thresholds are configured.*

F1 violates stability since it does not evaluate the serving carrier's measure. If both the serving and candidate carriers meet *F1*, the device will oscillate between them. For stability, the threshold evaluation must assess both carriers' measures.

We further restrict $\phi \geq \theta$ for *F2* to avoid trivial loops. All theorems regarding *F2* assume $\phi \geq \theta$. For *F2*, *F3*, and *F4*, stability is ensured iff. the following *minimum-measure* rule is applied (proofs in §B.2 in [6]):

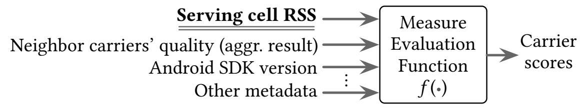


Figure 5: Project Fi's inter-carrier measure does not always satisfy Theorem 7.2.

Theorem 7.2 (Minimum-measure rule). *Assume inter-carrier policy's measure M and intra-carrier measure policy's Q are independent. The stability is violated* if and only if *$M(C_j) - M^{min}(C_j) \leq g(F*)$[13] cannot always hold no matter how per-cell measures change, where $g(F*)$ is defined as:*

$$g(F*) = \begin{cases} \phi - \theta & \text{for F2,} \\ \delta & \text{for F3 or F4.} \end{cases}$$

As a special case, the following sufficient condition offers a simpler rule regardless of the criteria form (F2 – F4):

Lemma 7.1 (Simple minimum-measure rule). *Following the assumption in Theorem 7.2, the threshold policy is stable if the carrier's measure $M(C_j) = M^{min}(C_j)$.*

Fundamentally, both rules are caused by the *different granularities* between the inter- and intra-carrier policies. The inter-carrier policy works at the *RAT/carrier level.* It cannot control the *cell-level* selection, which is done by the intra-carrier policy. With independent measures, the minimum rule is vital for the consistent decision between RAT/carrier switching (inter-carrier policy) and cell selection (intra-carrier policy).

Both results can also be generalized to the *different, yet correlated* measures (e.g., latency and signal strength): Lemma 7.1 still holds. Theorem 7.2 is sufficient, but not necessary.

In reality, there usually exist cells that are never selected by the intra-carrier policy. We can then relax the definition of carrier's measure to consider reachable cells only, to rule out bad or unavailable cells. We thus have the following corollary:

Corollary 1. *Consider the criteria* F2, F3, and F4. *If the carrier's measure is the minimum measure among all* reachable *cells in that carrier, we can ensure loop freedom.*

7.1.3 Validation. We find that, Project Fi does not always ensure the stability condition of Theorem 7.2, thus incurring loops. Google uses a machine learning module ("`K2so`") to compute the metric to rank and select the carriers (i.e., F3 criterion). Figure 5 shows how it computes the metric. It is based on the *serving cell*'s signal strength, and neighboring carrier's aggregate radio quality. Since the measure calculated by `K2so` cannot guarantee to be the minimum, it results in loops.

Trace 4 shows a log of persistent loops incurred by different measures between Project Fi and individual carriers. It illustrates three loops ❻, ❼ and ❽ within 30 minutes. At the spot, Sprint has 2G (1xRTT) coverage while T-Mobile does not have coverage. Based on the location and signal

[13] $M^{min}(C_j) = \min M(c_j^u)$ is the minimum measure of all cells in C_j.

Trace 4 Persistent loops caused by inconsistent measures

```
1      19:57:00 K2so sorted carriers: T-Mobile, USCC, Sprint.
2      19:57:00 Switch request to T-Mobile is approved. Requester: K2so.
3     ⎧19:59:02 Switch Sprint 1xRTT -> T-Mobile UNKNOWN done. result: TimedOut.
4     ⎪         srcSignalStrength:-103. destSignalStrength:-200.
5  (6)⎨20:04:20 K2so sorted carriers: T-Mobile, Sprint, USCC.
6     ⎪20:04:21 Switch request to Sprint is approved. Reason: signal loss.
7     ⎪20:05:11 Switch T-Mobile UNKNOWN -> Sprint 1xRTT done. result: Success.
8     ⎩         srcSignalStrength:-200.  destSignalStrength:-119.
9      20:04:20 K2so sorted carriers: T-Mobile, Sprint, USCC.
10     20:05:12 Switch request to T-Mobile is approved. Requester: K2so
11    ⎧20:07:13 Switch Sprint 1xRTT -> T-Mobile UNKNOWN done. result: TimedOut.
12    ⎪         srcSignalStrength:-119.  destSignalStrength:-200.
13 (7)⎨20:07:14 Wait for 05:00 before attempting another switch.
14    ⎪20:12:17 Switch request to Sprint is approved. Reason: signal loss.
15    ⎪20:12:45 Switch T-Mobile UNKNOWN -> Sprint 1xRTT done.
16    ⎩         srcSignalStrength:-200. destSignalStrength:-108.
17     20:12:46 K2so sorted carriers: T-Mobile, Sprint, USCC.
18     20:12:47 Switch request to Sprint is approved. Reason: signal loss.
19    ⎧20:14:47 Switch Sprint 1xRTT -> T-Mobile UNKNOWN done. result: TimedOut.
20    ⎪         srcSignalStrength:-108. destSignalStrength:-200.
21 (8)⎨20:14:49 Wait for 05:00 before attempting another switch.
22    ⎪20:19:52 Switch request to Sprint is approved. Reason: signal loss.
23    ⎪20:20:25 Switch T-Mobile UNKNOWN -> Sprint LTE done.
24    ⎩         srcSignalStrength:-200. destSignalStrength:-83.
```

conditions, the K2so monitor computes internal measures for each carrier according to Figure 5. At Line 1, the calculated measures for carriers are: T-Mobile 9.95, USCC 8.69, Sprint 7.92. It then uses F3 to sort them and selects the target carrier T-Mobile at Line 2. Since the user is static and signal strength does not change, this comparison remains unchanged. However, without T-Mobile coverage at the spot, Project Fi has to fall back to a carrier with basic service. Sprint was thus always chosen after a short period of time because it could offer 2G service. We disable "switching only once" for the K2so parameter in Project Fi, and show that this is a persistent loop. In fact, in less than half an hour, the loop already occurs 3 times and exhibits no sign of stopping.

7.2 Inconsistency of Configurations

We next consider the scenario that inter-carrier policy and intra-carrier policy use the same measure. In this category, the stability can be violated if the threshold configurations of inter/intra-carrier policies are uncoordinated.

7.2.1 Policy Form. It is the same as §7.1, except that inter-carrier and intra-carrier policies use the same measure M.

7.2.2 Stability Conditions. Given the same measures, Theorem 7.1 still holds, i.e. comparison criteria F1 is always loop-prone regardless of the threshold configurations. For F2 − F4, we have the following necessary conditions:

THEOREM 7.3 (UNSTABLE THRESHOLDS IN F2/F4). *Assume the inter-carrier policy uses* F2 *or* F4. *If the stability is violated, there must exist a carrier C_i with two cells c_i^u and c_i^v which satisfy the condition in* Table 3.

THEOREM 7.4 (UNSTABLE THRESHOLDS IN F3). *Assume the carrier's measure $M(C_j) = M^{max}(C_j)$, and inter-carrier policy uses* F3 *with offset δ. If the stability is violated, there would exist a carrier C_i satisfying:* (1) *There are two cells c_i^u, c_i^v in carrier C_i such that the criterion used for handoff $c_i^u \to c_i^v$ is in the form of* absolute-value*; or* (2) *There exists $l (l > 1)$ different cells $c_i^{u_1}, c_i^{u_2}, \cdots, c_i^{u_l}$. It satisfies that $\delta + \sum_{j=0}^{l-1} h(c_i^{u_j} \to c_i^{u_{j+1}}) < 0$ where function* h() *is defined as:*

$$h(c_i^u \to c_i^v) = \begin{cases} Thresh3^{u,v} - Thresh2^u, & \text{indirect comparison} \\ \Delta^u, & \text{direct comparison} \end{cases}$$

Table 3: Threshold incoordination in Theorem 7.3

Criteria for $c_i^u \to c_i^v$	*F2*, with ϕ, θ	*F4*, with δ, θ
Absolute-value comparison	$\theta > Thresh1_i^{u,v} + \Delta^v \lor \theta > Thresh1_i^{u,v}$	$\theta > Thresh1_i^{u,v} + \Delta^v \lor \theta > Thresh1_i^{u,v}$
Direct comparison	$\theta - \phi > \Delta^u$	$\delta + \Delta^u < 0$
Indirect comparison	$\theta > Thresh3_i^{u,v} + \Delta^v \lor \theta > Thresh3_i^{u,v}$	$\theta > Thresh3_i^{u,v} + \Delta^v \lor \theta > Thresh3_i^{u,v}$

The proofs are in §B.3 and §B.4 in [6]. Notably, both theorems imply that *aggregated intra-carrier thresholds* suffice for coordination with inter-carrier policies (elaborated in §9.2 and Table 6b). The carriers do not necessarily expose all of their per-cell thresholds to the MCSP for coordination.

7.2.3 Validation. We have not found real instances in this category. Current Project Fi always uses measures different from the intra-carrier policies, thus incurring no such conflicts. The theorems are thus serving as future guidelines for this category of policy.

8 STABILITY FOR HYBRID POLICY

The hybrid inter-carrier policies decide the target carrier based on both pre-defined preferences, and runtime measures (and their thresholds). This section generalizes our results in §6–7 to this scenario. In combining the preferences and thresholds, there are two approaches in general:

Preference-first policy. In this approach, the MCSP will first check each candidate carrier's preference, and evaluate its measure (via *F1 – F4*) based on the relations between their preferences and the serving carrier's (higher, lower, or equal). The idle-state intra-carrier policy (§5) belongs to this form. For each preference relation, the inter-carrier policy has the flexibility of choosing the threshold-based criterion (*F1 – F4*). But the following result shows some unstable criteria regardless of the threshold settings (proof in §C.1 in [6]):

THEOREM 8.1 (UNSTABLE COMPARISON WITH PREFERENCE). *In hybrid mechanisms with preference-first, loops will happen under the following combinations of threshold-based criteria: (1) Criterion* F1 *is applied to neighbor carriers with the equal preference; (2) Criterion* F1 *is applied to both neighbor carriers with higher preferences and neighbor carriers with lower preferences; (3) Criterion* F1 *is applied to neighbor carriers with higher preferences and criterion* F3 *is applied to neighbor carriers with lower preferences, or vice versa.*

Compared with Theorem 7.1, the use of preferences poses more constraints on selecting the threshold-based criteria.

Threshold-first policy. In this approach, the MCSP uses one threshold-based criterion for all candidate carriers. For candidates that meet this criterion, the MCSP will select the one with the highest preference. In this category, coordinating the threshold suffices for stability; the preference values do not pose extra constraints. If such hybrid policy is unstable, the corresponding threshold-only mechanism applying the same criterion and thresholds will also be unstable. The results in §7 still hold and can be readily applied here.

Project Fi validation. We have observed that Project Fi may apply preference-first and threshold-first policies in different scenarios. Although its preference-based policy (§6.1.1) and threshold-based policy (§7.1.1) are separate, they can be coupled by its internal per-module priority. When the device has network access, the threshold-based policy is preferred whenever it makes a decision. If the threshold-based policy does not make a decision, the preference-based policy will be used. This corresponds to the threshold-first policy. When the device has no network access, the preference-based policy is elevated with higher priority, thus resulting in preference-first policy. In practice, we have not observed real instances of loops so far.

9 PRACTICAL STABILITY GUIDELINES

Based on the above results, we devise practical guidelines for multi-carrier access stability. We seek to achieve three goals (ordered by their importance):

- **G1: Guaranteed Stability.** We seek guidelines for any-loop-freedom under any static settings.
- **G2: Retaining policy flexibility.** In guaranteeing the stability, our guidelines should still retain high flexibilities for the MCSP and carriers to customize their policies.
- **G3: Protecting internal policies.** Intuitively, enforcing stability implies that the MCSP and carriers should share their internal policies for coordination. This is nontrivial for both technical and non-technical reasons. In regulating the policies, it is desirable to reduce the policy exposures.

In achieving them, there are two practical constraints:

- **R1: Regulating inter-carrier policy only.** Carriers may be reluctant to change their internal policies for the MCSP: These policies not only serve the multi-carrier customers, but also single-carrier customers.
- **R2: Limited visibility to intra-carrier policy.** In regulating its inter-carrier policy, the MCSP may not have full access to the carriers' internal policies.

To derive the guidelines, we begin with the theoretical results in §6–8 that ensure stability (G1). We use them to regulate the inter-carrier policy only (R1), using the aggregate intra-carrier policies from carriers (G3 and R2). For practical applicability, we adopt these guidelines while leaving enough flexibility for carriers and the MCSP (G2). The guidelines are safe for carriers, because none requires that carriers expose their internal policies to MCSP. We next elaborate each.

9.1 Guidelines for Preferences

We devise guidelines for different forms of preferences in §6.

• *RAT-aware preference* (§6.1): Theorem 6.1 mandates the MCSP to regulate its RAT-aware preferences based on carriers' internal thresholds and priorities. However, this requires carriers' fine-grained internal policies, thus violating G3 and R2. We next make a realistic assumption and derive Corollary 2 of Theorem 6.1 (proof in A.5 in [6]).

Assumption 1. The intra-carrier policy will not move the device to a low-priority RAT from a high-priority one, following the idle-state policy of §5.

Corollary 2 (RAT preference conflict). *Under Assumption 1, also assume that the MCSP uses Policy 1. A persistent N-carrier loop happens* iff *both the following conditions hold: (a) every carrier has one or more RATs (denoted RAT_H) assigned with equal, highest preference; and (b) in all carriers, RAT_H does not have the highest intra-carrier priority.*

Assumption 1 holds in general unless the device is at the cell coverage boundaries or has extremely weak signal from the high-priority RAT. Following the above corollary, we can lift G3 and R2 and avoid loops in common settings with *aggregated* carrier priorities:

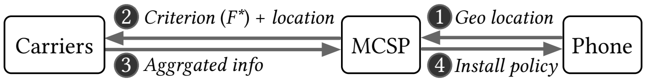
Guideline 1 (Coordination via priority aggregation). If a carrier C_i has and only has the most preferred RAT_H deployed, assign the highest inter-carrier preference to it: $P_{i,H} = P_{max}$. Otherwise, inter-carrier preference assignment should be monotonic on carriers: $\forall i \neq j, P^i_{min} > P^j_{max}$ or $P^j_{min} > P^i_{max}$. The order of monotonicity is flexible but should reflect the MCSP's preference for carriers.

Guideline 1 only requires carriers to share its maximum intra-carrier priority (G3), thus remaining safe for carriers. It still retains high flexibility of the preference settings (G2) since multiple monotonic ordering could exist. For instance, if both C_1 and C_2 have 3G and 4G, the preference order can be either of the following: $P_{1,4G} > P_{1,3G} > P_{2,4G} > P_{2,3G}$, or $P_{2,4G} > P_{2,3G} > P_{1,4G} > P_{1,3G}$. The MCSP may prefer the first ordering if C_1's service quality is generally better than C_2's. In case if C_1 only has 3G, while C_2 has 3G and 4G, the preference order is still flexible to ensure loop-free: $P_{2,4G} > P_{2,3G} > P_{1,3G}$, or $P_{1,3G} > P_{2,4G} > P_{1,3G}$.

• *RAT-oblivious preference* (§6.2): If the MCSP uses the RAT-oblivious preferences, the following guideline (based on Theorem 6.2) ensures stability and meets G1–G3 and R1–R2:

Guideline 2 (Avoid preference-unavailability conflict). Disable carriers whose intra-carrier policy can move the device to an unusable cell.

Guideline 2 ensures stability (G1) since it satisfies Theorem 6.2. It also retains high flexibility (G2): Except the disabled carriers, it allows arbitrary preference settings by the MCSP. It does not require the exposure of the carriers' internal policies (G3): Carriers only report a *binary confirmation* about whether it can move the device to an unusable cell (e.g., poor signal coverage, access denial for base station congestion [1, 4, 7] etc.).



(a) Information exchange flow

Necessary aggregation info. from carriers	*F2*	*F3*	*F4*
$\min\{\min_{c^u \neq c^v} Thresh1^{u,v}, \min_{c^{u'} \neq c^{v'}} Thresh3^{u',v'}\}$	✓		✓
$\min\{\min_{c^u \neq c^v} Thresh1^{u,v} + \Delta^v, \min_{c^{u'} \neq c^{v'}} Thresh3^{u',v'} + \Delta^{v'}\}$	✓		✓
$\min_{c^u} \Delta^u$	✓		✓
$\min\{\min_{c^u \neq c^v} Thresh3^{u,v} - Thresh2^u, \min_{c^{u'}} \Delta^{c^{u'}}\}$		✓	
$\lvert\{c^u \mid \Delta^u < 0 \vee \exists c_v, Thresh3^{u,v} < Thresh2^u\}\rvert$[a]		✓	

[a] The number of cells with $\Delta < 0$ or $Thresh3 - Thresh2 < 0$.

(b) Aggregation items needed for coordination

Figure 6: Threshold coordination for Guideline 5

9.2 Guidelines for Thresholds

We offer guidelines for various threshold-based policies (§7).

• *Inconsistency of measures* (§7.1): If the MCSP uses different measures from carriers, the following guideline helps the MCSP rule out the loop-prone criteria (Theorem 7.1):

Guideline 3 (Avoid loop-prone criteria). If the inter-carrier policy uses different measures from the intra-carrier policies, it should not use Criterion *F1* to evaluate carriers.

Next, the MCSP should regulate how it determines the measure for each carrier (based on per-cell measure metric). In principle, Theorem 7.2 provides necessary and sufficient conditions for loop-freedom. In addition, Lemma 7.1 gives more practical conditions to ensure loop-freedom. However, they may not be desired due to their limited flexibility in reality (G2). Consider a carrier that deploys 2G, 3G, and 4G. Using its own measure Q, the carrier may never move the device to 2G. However, based on the minimum-measure rule, the MCSP has to use 2G's measurement on M (such as latency) in determining the measure, which may be unfavorable. The guideline below relaxes this constraint while still satisfying Lemma 7.1:

Guideline 4 (Relaxed minimum measure). Consider the inter-carrier switching policy that uses different measures (M) from the intra-carrier policies (Q) If a carrier's internal policy would only move the device to a subset of its cells (under Q), the MCSP should apply Lemma 7.1 to this subset.

Compared with Lemma 7.1, Guideline 4 mitigates the impact of the minimum measure. In the above example, if 2G is not selected in intra-carrier policy, its measures (e.g. latency) would not need to be considered in inter-carrier policy either. This guideline does not require exposure of intra-carrier policy either (G3): Each carrier only reports a list of cells that its internal policy will not select.

• *Inconsistency of configurations* (§7.2): If the MCSP uses the same measure as the carriers', it should coordinate its thresholds for stability. In principle, the MCSP requires access to all carriers' per-cell thresholds, which however violates G3 and R2. To prevent it, we use the *aggregated thresholds* based on Theorem 7.3 and 7.4, and devise the following guideline:

Guideline 5 (Coordination via aggregated thresholds). If the inter-carrier policy takes criterion *F2* or *F4*, set the inter-carrier thresholds to satisfy conditions in Theorem 7.3. If the inter-carrier policy takes criterion *F3*, set the inter-carrier thresholds to satisfy conditions in Theorem 7.4.

Note that, to coordinate thresholds, the MCSP will query each carrier with a criterion (*F2* – *F4*). The carrier returns aggregated information about intra-policy threshold (Figure 6a). The forms of those aggregation are listed in Figure 6b. More details to follow in §9.4.

9.3 Guidelines for Hybrid Policy

If the MCSP deploys the preference-first policy, Theorem 8.1 offers the following guideline. Note that it does not require access to intra-carrier policy (G3), nor regulating the preferences or thresholds (G2).

Guideline 6 (Loop-prone criteria given preferences). If the hybrid inter-carrier policy uses preference-first, it should not use Criteria *F1* and *F3* under the conditions in Theorem 8.1.

If the MCSP deploys the threshold-first policy, §8 has shown that Theorem 7.1 and *necessary conditions* of loops in Theorem 7.2, 7.3 and 7.4 still hold. This implies that the MCSP does not need to regulate the preferences, as shown in the following guideline:

Guideline 7 (Threshold-first). If the hybrid inter-carrier policy uses threshold-first, it only needs to regulate its thresholds by following Guideline 4–5.

9.4 Applying Guidelines to Project Fi

We show how to apply the guidelines in a Project Fi-like setting. The MCSP coordinates with carriers and updates its policy (Figure 6a) with four steps:

(1) The device reports its location to the MCSP (via the Project Fi app) since the policy is location dependent.
(2) The MCSP queries all participating carriers. The query is based on the device location and the policy used;
(3) Each carrier computes an aggregated answer to the query;
(4) The MCSP collects the responses, selects the target carrier(s) suitable for the device, specifies and installs it in a policy update on the device.

The procedures can also be completed *offline*. The MCSP could query all locations (thus making Step (1) not mandatory) and pre-compute its policies across locations. At runtime, the device retrieves the pre-computed policy.

We use Guideline 1 as one example. Suppose the MCSP uses the RAT-aware preference as its inter-carrier policy. It updates its policy at location X (from the phone in Step (1)) with two carriers, C_1 and C_2. According to Guideline 1, the query from the MCSP is "What is your most preferred RAT at location X? Do you have only one RAT deployed at X?" in Step (2). Assume the response from C_1 is "[4G; No]", and

Table 4: Project Fi coverage

City	Total Grids[a]	Has 4G LTE	Only 3G	Only 2G/No service
Los Angeles	122 335 (1261 km^2)	120 480 (98.48%)	1850 (1.51%)	5 (<0.01%)
St. Louis	136 350 (1295 km^2)	101 773 (74.64%)	34 574 (25.36%)	3 (<0.01%)

[a] Each grid's resolution is 0.001°, resulting in equivalently 110 m × 110 m grid.

Table 5: Intra-carrier policy statistics

Cell priority	1	2	3	4	5	6
Count #	35719	3116	17300	4	11851	2698
Percentage (%)	50.5	4.4	24.5	<0.1	16.7	3.8

Table 6: Emulation settings

Scenario	C_1.LTE ($c^{1,2,3}$), C_1.3G (c^4), C_2.LTE ($c^{5,6,7}$), C_2.3G (c^8)
Cell RSS range	LTE: [−124, −80] dBm; 3G: [−120, −75] dBm
Intra-priority	Enumeration of 1, 3, 5 ordering
Intra-threshold	$Thresh1, 2, 3, \Delta$ varies
Inter-preference	75 combinations
Inter-threshold	*F2*: $\theta, \phi \in [-115, -109]$ dBm; *F3*: $\delta \in [0, 4]$ dB

the reply from C_2 is "[3G; Yes]". The MCSP can adopt either of the two policies while ensuring loop-freedom: (a) $P_{2,3G} > P_{1,4G} > P_{1,3G} > P_{1,2G}$, or (b) $P_{1,4G} > P_{1,3G} > P_{1,2G} > P_{2,3G}$. The MCSP then pushes the policy to the device in Step (4).

We also see early signs of using Guideline 2 in Project Fi. Specifically, it includes a "`ServerPolicy`" monitor. Google's backend server knows when and where a carrier's service is unavailable. It then updates the enabled carrier list in its policy using the same mechanism as Step (4). The device will not switch to that carrier after this update. This follows Guideline 2 to avoid unavailable carriers.

Applying other guidelines follows similar procedures. We have shown such procedures are safe, practical and available to the MCSP and carriers. It neither exposes carriers' internal operation secrets, nor incurs significant communication overhead. The MCSP cannot infer the specifics of carrier's internal policies using the *aggregated* responses of Step (3). The MCSP can readily form the queries and analyze the responses. Note that all communication channels required in Figure 6a are readily available in Project Fi, as its `ServerPolicy` monitor proves.

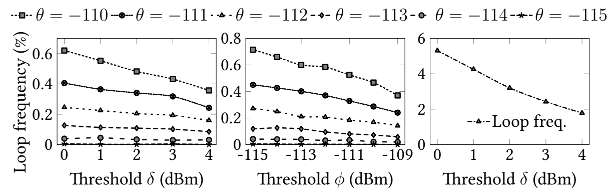
10 VALIDATIONS OF GUIDELINES

We next assess the occurrence of conflicts in reality, and the effectiveness of our guidelines. To detect loops in a large-scale setting, we use trace-driven emulations to complement our real-world validations. We use Project Fi's real coverage data and logic for the inter-carrier policy, as well as carriers' configurations from intra-carrier policy traces.

Emulation with operational traces. To approximate the real-world multi-carrier access at a large geographical scale, we extract emulation parameters from the operational traces. To obtain real cell coverage, we crawled Project Fi's coverage data [15] as of 03/07/2018 for Los Angeles, CA (a large city) and St. Louis, MO (a mid-sized city). The coverage statistics are summarized in Table 4. For each cell, we assign its intra-carrier priorities based on the operational T-Mobile/Sprint/USCC configurations (analyzed

Preference list	Loop freq. (%)
Only 3G assigned the highest pref.	6.160
One of C_1.LTE and C_2.LTE assigned the highest pref.	0.088
Both C_1.LTE and C_2.LTE assigned the highest pref.	0.003

(a) RAT-aware preference policy



(b) Using *F2* (c) **Using *F4*** (d) **Using *F3***

(b) Threshold-based policy

Figure 7: Loop occurrence and evidence of guidelines

from 50GB traces of the MobileInsight [33] public dataset). Table 5 summarizes these priorities. We further select the most common and representative intra-carrier thresholds[14]: $Thresh1 \in [-115, -117]$ for LTE and $Thresh1 = -108$ for 3G; $Thresh2 \in [-120, -116]$ for LTE and $Thresh2 = -108$ for 3G; $Thresh3 \in [-120, -116]$ for LTE and $Thresh3 = -114$ for 3G, $\Delta \in \{-2, 2, 3\}$. The cell signal strengths observed in the dataset range in [−124, −80] for LTE and [−120, −75] for 3G.

With these data, our emulation uses the settings in Table 6. We set two carriers, C_1 and C_2, both offer LTE and 3G according to Project Fi's coverage. In the emulation, we vary each cell's signal strength according to the observed range. We enumerate reasonable inter-carrier policies as follows: First, for the preference policy, we enumerate all RAT-aware preference lists, which result in 75 different preference orderings. Out of the 75 different orderings, 40 are loop-prone. They fall into three categories in Table 7a. Second, for the threshold policy, we use criteria F2, F3, and F4 (F1 is always loop-prone according to Theorem 7.1), and set $\theta, \phi \in [-115, -109]$ for *F2*, and $\delta \in [0, 4]$ for *F3*. The setting is the same for *F4*. These ranges will not cause trivial loops. We repeat the emulation for 75 different settings of preference-policy and have 1.5M rounds in total; For the threshold-policy, we do emulation for 77 different settings involving criteria *F2, F3, F4*, and also have totally 1.5M rounds. Figure 7 shows the results.

Spatial frequency of loops. Figure 7a summarizes the spatial frequency of loops for the preference-based policy. The frequency ranges between 0.003% and 6.16%. It offers a conservative estimate, given that indoor signals are more complex. The higher degree of conflicts between inter- and intra-carrier policies, the more frequent loops across locations. Note that, the preference setting with 3G being assigned the highest preference is the most unstable. It is likely to occur if the 3G access is from small enterprise cells.

For threshold-based policy, Figure 7b shows the frequency of loops versus configurations on θ, δ or ϕ . For *F2, F3*, and *F4*, the frequency of loop drops as θ decreases, ϕ increases or δ

[14]The unit for θ and ϕ is dBm and that for δ is dB.

increases. This is consistent with Guideline 5, thus indirectly validating the effectiveness of our guideline.

Effectiveness of guidelines. To evaluate the effectiveness of our guidelines, we rule out loop-prone inter-carrier policies following our guidelines in §9, and repeat the simulation under the same settings. We have validated that no loops will occur after this regulation. Moreover, as shown above, Figure 7b also validates the effectiveness of our guideline.

11 DISCUSSION

Dynamic policy updates. We so far assume invariant policies for MCSP and carriers (§5). It is possible that the MCSP dynamically updates its policies. Our results can be generalized. Assume that stability is ensured before the update. A policy update is *safe* iff stability is still guaranteed after this update. The following proposition (proof in §D.1 in [6]) offers conditions for safe preference and threshold updates, thus extending our results to the dynamic scenarios:

Proposition 2 (Safe policy update). *The following inter-carrier policy updates are safe: (1) Increasing inter-carrier preferences for the most preferred carrier; (2) Decreasing θ, ϕ or increasing δ in criteria* F2, F3, and F4.

Dynamic measures. Our results are obtained by assuming the measures (e.g., signal strengths, latency) are constant. When the measures are dynamic, transient loops may also occur (i.e., "ping-pong" loops). Such a loop is not necessarily bad; there are standard techniques (e.g., maximum attempts, similar to Project Fi in §4) to mitigate it. Our guidelines focus on persistent loops caused by policy conflicts.

Mobility case. As the device moves, the inter-carrier policy also changes with locations. This can be viewed as a sequence of addition/deletion of carriers/RATs/cells (each associated with intra-carrier policies). The policy guidelines in §9 can thus be recursively applied to these sequences.

Our results also apply to the PLMN selection for roaming [5]. PLMN selection is a mandatory function for all commodity phones. As the device leaves the coverage of its home carrier, PLMN selection searches the visiting carrier network based on pre-defined RAT-aware preferences. The results in §6 are thus applicable to regulate its stability.

12 RELATED WORK

Multi-carrier access offers a promising alternative to the dominant single-carrier paradigm. Early systems support multi-carrier access inside commodity phones using dual SIM cards [11, 18] or a single SIM card [12–14]. Recent research has focused on improving various aspects of multi-carrier access such as performance [29, 32] and concurrent access to multiple carriers in 5G [28]. We complement prior work by investigating policy management for multi-carrier access, a topic not studied so far. While our study draws insights from Project Fi, our paper is also forward-looking by extending the policy-based switching to a more generic setting (more carriers, more policy types, etc.).

Instability and policy inconsistencies have been examined in other networking systems, such as BGP routing [23, 24, 30], and SDN and data center networks [27, 36, 38]. Our paper differs from them because of the setting (mobile networks) and mechanism (inter-carrier switching). The results from other networking systems are thus not applicable to our context. While stability results have been recently reported for configurable handoffs within a carrier [31, 34], our problem is different because we examine policy conflicts among inter-carrier policies, and between inter- and intra-carrier policies. Specifically, our work differs from [31, 34] in the causes of policy conflicts, the need for two-tiered system modeling, and the policy guidelines we provide at the inter-carrier level despite assuming policy autonomy within each carrier.

13 CONCLUSION

Multi-carrier cellular access promises to provide mobile users with better service than single-carrier access. Google Project Fi [14] already shows early signs of success and great benefits without requiring cellular infrastructure upgrades. Multi-carrier access requires dynamically selecting a preferred carrier before proceeding to cell selection inside the carrier. While the detailed carrier selection algorithm may evolve over time, our paper argues that some basic framework must be specified at the inter-carrier level to fulfill the potential of multi-carrier access.

Our paper makes a case for policy-based switching as the basic mechanism for inter-carrier selection. On one hand, inter-carrier policy has nice features and is needed by MCSPs. This is evident from operational practice in Project Fi and historical lessons from BGP and data center networks. On the other hand, policy-based design is a double-edged sword that introduces new research issues. We have shown that policy conflicts can arise between the customized inter-carrier switching policy and the standardized intra-carrier cell selection (handoff). We identify several such cases and provide practical guidelines to resolve such policy conflicts. We seek to be forward-looking by abstracting and generalizing to a more generic setting beyond Project Fi. Our ultimate goal is to embrace, rather than suppress, the new challenges introduced by policy-based switching, and to make multi-carrier cellular access as successful and commonplace as multiprotocol networks.

ACKNOWLEDGMENTS

We greatly appreciate our anonymous shepherd and reviewers for their constructive comments. This work was partially supported by NSF Grants: CNS-1526985, CNS-1526456, CNS-1423576, CNS-1421440, CNS-1749049, CNS-1750953, and ICN-WEN 1719403.

A PROOF OF MAIN RESULTS

We provide the complete proofs to our main results in §6–§8 in [6] due to space limit.

REFERENCES

[1] 3GPP. 2012. TS22.011: Service accessibility. (2012). http://www.3gpp.org/ftp/Specs/html-info/22011.htm

[2] 3GPP. 2012. TS25.304: User Equipment (UE) Procedures in Idle Mode and Procedures for Cell Reselection in Connected Mode. (2012). http://www.3gpp.org/ftp/Specs/html-info/25304.htm

[3] 3GPP. 2012. TS25.331: Radio Resource Control (RRC). (2012). http://www.3gpp.org/ftp/Specs/html-info/25331.htm

[4] 3GPP. 2012. TS36.331: Radio Resource Control (RRC). (2012). http://www.3gpp.org/ftp/Specs/html-info/36331.htm

[5] 3GPP. 2013. TS36.304: User Equipment Procedures in Idle Mode. (2013).

[6] Zengwen Yuan, Qianru Li, Yuanjie Li, Songwu Lu, Chunyi Peng, and George Varghese. 2018. Supplementary Material to Resolving Policy Conflicts in Multi-Carrier Cellular Access. (2018). http://metro.cs.ucla.edu/papers/mobicom18-multicarrier-proof.pdf

[7] Kenichiro Aoyagi, Wuri A. Hapsari, Shinya Takeda, and Itsuma Tanaka. 2015. Access Class Control Technology in LTE/LTE-Advanced Systems. *NTT DOCOMO Technical Journal* 17, 2 (2015), 65–76.

[8] Archclearing. 2015. ARCH Insight: Alternative Roaming Providers spring up in China. (2015). http://www.archclearing.com/html/News/Views/ALTERNATIVE/ALTERNATIVE.htm

[9] Aruna Balasubramanian, Ratul Mahajan, and Arun Venkataramani. 2010. Augmenting Mobile 3G Using WiFi. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys '10)*. ACM, New York, NY, USA, 209–222. DOI: http://dx.doi.org/10.1145/1814433.1814456

[10] Mostafa Zaman Chowdhury, Won Ryu, Eunjun Rhee, and Yeong Min Jang. 2009. Handover between macrocell and femtocell for UMTS based networks. In *2009 11th International Conference on Advanced Communication Technology*, Vol. 01. 237–241.

[11] Wikipedia. 2018. Dual SIM. (2018). https://en.wikipedia.org/wiki/Dual_SIM

[12] Apple Inc. 2018. Apple SIM. (2018). https://www.apple.com/ipad/apple-sim/

[13] Daniel Cooper. 2016. Samsung's next smartwatch comes with an e-SIM. (Feburary 2016). https://www.engadget.com/2016/02/18/samsung-gear-s2-esim/

[14] Google. 2018. Project Fi. (2018). https://fi.google.com/.

[15] Google. 2018. Project Fi Coverage Map. (2018). https://fi.google.com/coverage.

[16] ZTE Corporation. 2013. ZTE UMTS Handover Description. (September 2013). http://www.slideshare.net/quyetnguyenhong/zte-umtshandoverdescription

[17] Andrei Croitoru, Dragos Niculescu, and Costin Raiciu. 2015. Towards Wifi Mobility without Fast Handover. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI '15)*. USENIX Association, Oakland, CA, 219–234.

[18] Supratim Deb, Kanthi Nagaraj, and Vikram Srinivasan. 2011. MOTA: Engineering an Operator Agnostic Mobile Service. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking (MobiCom '11)*. ACM, New York, NY, USA, 133–144. DOI: http://dx.doi.org/10.1145/2030613.2030629

[19] Android Developers. 2018. WifiConfiguration – Android API Reference. (June 2018). https://developer.android.com/reference/android/net/wifi/WifiConfiguration.html

[20] ArchLinux Wiki. 2018. WPA supplicant. (June 2018). https://wiki.archlinux.org/index.php/WPA_supplicant

[21] Small Cell Forum. 2018. Small cells market status report (release 10.0). (Feburary 2018). http://scf.io/en/white_papers/Market_status_report_December_2017_Special_edition.php

[22] Ning Ding, Daniel Wagner, Xiaomeng Chen, Abhinav Pathak, Y. Charlie Hu, and Andrew Rice. 2013. Characterizing and Modeling the Impact of Wireless Signal Strength on Smartphone Battery Drain. In *Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '13)*. ACM, New York, NY, USA, 29–40. DOI: http://dx.doi.org/10.1145/2465529.2466586

[23] Lixin Gao and Jennifer Rexford. 2001. Stable Internet Routing Without Global Coordination. *IEEE/ACM Transactions on Networking* 9, 6 (Dec. 2001), 681–692. DOI: http://dx.doi.org/10.1109/90.974523

[24] Timothy G. Griffin and Gordon Wilfong. 1999. An Analysis of BGP Convergence Properties. In *Proceedings of the ACM SIGCOMM 1999 Conference (SIGCOMM '99)*. ACM, New York, NY, USA, 277–288. DOI: http://dx.doi.org/10.1145/316188.316231

[25] GSMA. 2018. The SIM for the next Generation of Connected Consumer Devices. (2018). https://www.gsma.com/esim/

[26] Junxian Huang, Feng Qian, Alexandre Gerber, Z. Morley Mao, Subhabrata Sen, and Oliver Spatscheck. 2012. 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 (MobiSys '12)*. ACM, New York, NY, USA, 225–238. DOI: http://dx.doi.org/10.1145/2307636.2307658

[27] Xin Jin, Hongqiang Harry Liu, Rohan Gandhi, Srikanth Kandula, Ratul Mahajan, Ming Zhang, Jennifer Rexford, and Roger Wattenhofer. 2014. Dynamic Scheduling of Network Updates. In *Proceedings of the 2014 ACM Conference on SIGCOMM (SIGCOMM '14)*. ACM, New York, NY, USA, 539–550. DOI: http://dx.doi.org/10.1145/2619239.2626307

[28] S. Kanugovi, S. Vasudevan, F. Baboescu, J. Zhu, S. Peng, J. Mueller, and S. Seo. 2018. Multiple Access Management Services. (April 2018). https://datatracker.ietf.org/doc/draft-kanugovi-intarea-mams-framework/ IETF Internet Draft.

[29] Parishad Karimi, Ivan Seskar, and Dipankar Raychaudhuri. 2016. Achieving high-performance cellular data Services with multi-network access. In *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–6. DOI: http://dx.doi.org/10.1109/GLOCOM.2016.7841563

[30] Craig Labovitz, G. Robert Malan, and Farnam Jahanian. 1998. Internet Routing Instability. *IEEE/ACM Transactions on Networking* 6, 5 (Oct 1998), 515–528. DOI: http://dx.doi.org/10.1109/90.731185

[31] Yuanjie Li, Haotian Deng, Jiayao Li, Chunyi Peng, and Songwu Lu. 2016. Instability in Distributed Mobility Management: Revisiting Configuration Management in 3G/4G Mobile Networks. In *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science (SIGMETRICS '16)*. ACM, New York, NY, USA, 261–272. DOI: http://dx.doi.org/10.1145/2896377.2901457

[32] Yuanjie Li, Haotian Deng, Chunyi Peng, Zengwen Yuan, Guan-Hua Tu, Jiayao Li, and Songwu Lu. 2016. iCellular: device-customized cellular network access on commodity smartphones. In *Proceedings of the 13th USENIX Conference on Networked Systems Design and Implementation (NSDI '16)*. USENIX Association, Santa Clara, CA, 643–656.

[33] Yuanjie Li, Chunyi Peng, Zengwen Yuan, Jiayao Li, Haotian Deng, and Tao Wang. 2016. MobileInsight: Extracting and Analyzing Cellular Network Information on Smartphones. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom '16)*. ACM, New York, NY, USA, 202–215. DOI: http://dx.doi.org/10.1145/2973750.2973751

[34] Yuanjie Li, Jiaqi Xu, Chunyi Peng, and Songwu Lu. 2016. A First Look at Unstable Mobility Management in Cellular Networks. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications (HotMobile '16)*. ACM, New York, NY, USA, 15–20. DOI: http://dx.doi.org/10.1145/2873587.2873599

[35] Fei Liu, Petri Mahonen, and Marina Petrova. 2014. A handover scheme towards downlink traffic load balance in heterogeneous cellular networks. In *2014 IEEE International Conference on Communications (ICC)*. 4875–4880. DOI: http://dx.doi.org/10.1109/ICC.2014.6884092

[36] Hongqiang Harry Liu, Xin Wu, Ming Zhang, Lihua Yuan, Roger Wattenhofer, and David Maltz. 2013. zUpdate: Updating Data Center Networks with Zero Loss. In *Proceedings of the ACM SIGCOMM 2013 Conference (SIGCOMM '13)*. ACM, New York, NY, USA, 411–422. DOI: http://dx.doi.org/10.1145/2486001.2486005

[37] OpenSignal. 2017. State of Mobile Networks: USA (Regional Performance). (Aug 2017). https://opensignal.com/reports/2017/08/usa/state-of-the-mobile-network.

[38] Peng Sun, Ratul Mahajan, Jennifer Rexford, Lihua Yuan, Ming Zhang, and Ahsan Arefin. 2014. A Network-State Management Service. In

Proceedings of the ACM SIGCOMM 2014 Conference (SIGCOMM '14). ACM, New York, NY, USA, 563–574. DOI: http://dx.doi.org/10.1145/2619239.2626298

[39] Next Generation Mobile Networks 5G Initiative Team. 2015. NGMN 5G white paper. (Feburary 2015). https://www.ngmn.org/5g-white-paper/5g-white-paper.html.