Towards Automated Intelligence in 5G Systems

Haotian Deng[‡], Qianru Li[†], Yuanjie Li[†], Songwu Lu[†], Chunyi Peng[‡],
Taqi Raza[†],Zhaowei Tan[†], Zengwen Yuan[†], Zhehui Zhang[†]
[†] UCLA Computer Science, Los Angeles, CA 90095
[‡]Dept. Computer Science Engineering, The Ohio State University, Columbus, OH 43210

***Abstract*—In this paper, we call for a paradigm shift away from the wireless-access focused research efforts on 5G networked systems. We believe that the architectural limitations should share equal blame on issues of performance, reliability, and security. We thus identify architectural weakness on both sides of the mobile clients and the 4G network infrastructure. Our recent findings show that, contrary to commonly held perceptions, many design and operational issues arise not due to poor wireless link qualities. Instead, they are rooted in such architectural downsides. To address these issues, we further propose a new approach of enabling automated intelligence inside the 4G/5G network systems. We next describe our ongoing efforts along two dimensions: empowering date-driven smart clients and constructing verifiable network infrastructure. We report some early results and discuss possible next steps.**

I. Introduction

Mobile Internet access has become an essential component in our daily life for its *anywhere, anytime* convenience. The mobile data traffic volume has grown 18-fold over the past five years, and is forecast to reach 49 exabytes (49 million TB) per month by 2021 [1]. As a matter of fact, mobile devices have already replaced the desktops/laptops to become the primary means to access the Internet since 2012. The underlying technology enabler is the mobile networked systems, namely, the current 4th-generation (4G) and upcoming 5G networks. To date, they are the only large-scale networked systems, in par with the wired Internet, which can provide ubiquitous network services.

Despite its great success and popularity, the current 4G system is not without problems. Users regularly complain about bad performance, lost network access, reduced availability, and common failures and even attacks. The conventional wisdom seems to mainly point its fingers to the wireless link, which may perceive poor radio channel qualities as the main root cause for most user-perceived issues. For example, slow data speed is caused by weak radio coverage at some places; service disruption is resulted from no coverage without an in-time seamless handoff.

However, we believe this is not the case, at least not the sole case. Our recent efforts [2]–[7] have shown that, the current networked architecture, as well as its software in the form of protocols, should equally share the blame. Tracing back to its roots, the mobile networked system has mainly taken the legacy telecom design tenets, despite wrapping it up with an all-IP-based design (from the wired Internet) in the current 4G system. Consequently, it still observes the "*dumb terminal, smart core*" philosophy: Most critical functions are placed inside the network infrastructure, while exposing limited capabilities to client devices in terms of accessing the runtime network information or taking adaptive actions. This results in two fundamental downsides: (1) Opaqueness at the client: The mobile OS and apps lack information regarding the underlying "black-box" network operations, thus constraining their full potentials to react to failures, slow speed and security vulnerabilities. (2) Complexity and lack of verification at the Infrastructure: The infrastructure suffers from complex designs and operations. It is distributed in nature, with rich interactions between diverse components (wireless, mobility, security) running at different nodes. This may lead to many problematic behaviors under various usage settings. The fundamental problem is that, the current design lacks intelligence on both sides of the client and the infrastructure. While intelligence should not be interpreted in yes/no, binary forms, we believe the appropriate networked system need to know *what*, *why* and *how* regarding operations and behaviors.

We thus propose a paradigm shift in the research agenda on 5G technology, away from the wireless-access focused R&D efforts. Based on these observations and findings, we explore to empower automated intelligence in the mobile networked system. In the broader context, we follow the *"smart client, simple (and verifiable) infrastructure"* design guidelines. By pushing more network visibilities and functions to the end devices, the mobile clients could play more active roles in improving its user-experienced performance, reliability and security. Furthermore, we propose to verify and simplify the infrastructure design and operations. Our overall solution injects a new knowledge plane on top of the existing information plane, which consists of all three aspects of control-plane, management-plane, and data-plane operations. Therefore, we can directly access low-level behaviors, synthesize them, and reason about their outcomes, and instruct the applications and OS for more intelligent adaptations.

Within the above vision, we share our recent efforts along two dimensions: (1) **Data-driven smart clients**: We enhance intelligence at client devices with a data-driven approach. It constructs and exploits data analytics on the "black-box" cellular operations, and enhances critical services (*e.g.*, roaming) with cellular insights, thus improving performance and security for mobile applications. This yields immediate benefits for clients. (2) **Verifiable infrastructure**: We enforce provably correct and reliable infrastructure design and operations, by applying distributed computing and model-checking techniques.

Authors in alphabetical order.

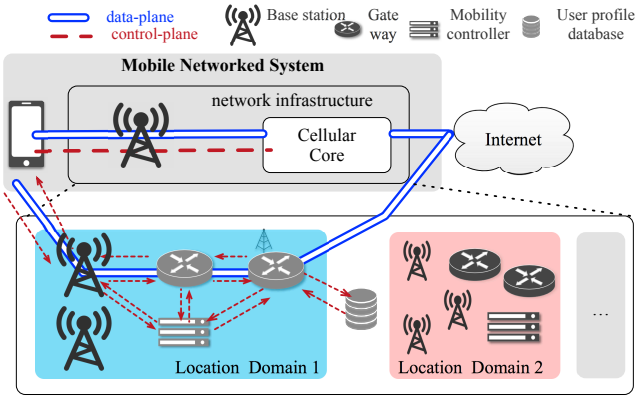


Figure 1: 4G LTE mobile networked system and its network infrastructure.

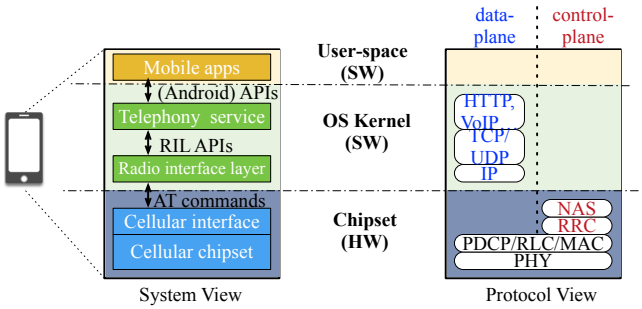


Figure 2: Mobile device system and its protocol view.

This not only helps the operators in their current management, but also offers infrastructure primitives for next-generation mobile systems. We share our current findings, and identify open issues for the next-step research.

The rest of the paper is structured as follows. §II introduces the background of the 4G mobile networked system, and §III pinpoints its limitations. §IV overviews our high-level proposal of enabling automated intelligence in the system; §V and §VI describe our ongoing efforts on both sides of the client and the infrastructure through illustrative examples. §VII discusses the next steps, and §VIII concludes the paper.

II. Current 4G LTE System

To date, the only large-scale wireless platform that provides "anywhere, anytime" indoor/outdoor network services is the cellular system (4G and upcoming 5G). In this section, we provide a brief and necessary background of the 4G system.

The current 4G LTE system generally follows the all-IP based paradigm by adapting its core subsystems (such as radio resource management, mobility support, security, etc.) from the telecom legacy design. In general, it consists of three main parts: the network infrastructure, the clients, and the wireless links in between. Figure 1 illustrates the overall system.

Network infrastructure The infrastructure (shown in Figure 1) provides key functions to enable "anywhere, anytime" network services to users, including wireless access, seamless mobility support, connectivity control, security, billing functions. It consists of a radio access network (RAN) and a core network. The RAN uses a large number of base stations to offer wireless access to the mobile client. The core network connects the RAN to the wired Internet. As illustrated in Figure 1, it consists of gateways (GWs), the mobility controller (MC), and the user profile server.

To scale to large coverage, the infrastructure further uses a two-tier structure. Each geographic area is covered by multiple, possibly overlapping base stations (or so-called cells) to ensure seamless services. The base stations enable the fine-grained *handoff*, through which the device reconnects to a new cell as it roams. At the coarse granularity, base stations in a geographic area are grouped and managed by a central mobility controller, which regulates device roaming between areas (called *location domain*) through *location update*.

Mobile clients At the client side, the current OS and mobile apps in the software space have limited access to cellular-specific information at runtime. As shown in Figure 2 (Android used as an example), cellular-specific protocols are implemented within the chipset (e.g., Qualcomm Snapdragon). Consequently, runtime network information is not accessible to the software stack. The OS may access certain cellular functions and state through the *de facto* radio interface layer (RIL) across the software and hardware boundary. The RIL is vendor specific, and relies on the standardized `AT` commands. The OS further exposes a subset of the RIL library to the API, e.g., `TelephonyManager` class for Android [8], [9], which is further used by mobile apps in the user-space.

Protocol stack Similar to the Internet, mobile network protocols have adopted the layered structure. The protocols operate on both data and control planes, as well as on the management plane. The data plane is responsible for actual data transfer. The control plane provides a variety of signaling functions to facilitate the data-plane operations. The management plane specifies policies and configurations for each protocol. Each protocol runs on both sides of the client and the infrastructure.

Figure 2 (right) illustrates the cellular protocol stack, which has three parts. The first is to enable radio access between the device and the base station. Physical (L1) and link (L2) functionalities, including PHY, MAC, RLC (Radio Link Control) and PDCP (Packet Data Convergence Protocol), are implemented. The second part is the control-plane protocols, which are split into access stratum (AS) and non-access stratum (NAS). AS regulates radio access through the Radio Resource Control (RRC) protocol. RRC is mainly for radio resource allocation and radio connection management; it also helps to transfer signaling messages over the air. NAS is responsible for conveying non-radio signaling messages between the device and the core network. Two protocols of mobility management (MM) and session management (SM) also operate on the control plane. MM offers location updates and mobility support for call/data sessions, while SM creates and mandates voice calls and data sessions. The last piece is the data-plane protocols above IP, which are not cellular specific but use the standard TCP/IP suite.

III. Limitations of Current 4G LTE

We now identify key problems with the current 4G mobile systems. It turns out that, most problems above can be attributed to the architectural limitations. Following the traditions of the telecom community, the mobile networked system design today still follows the "*dumb terminal, smart core*" philosophy: Most critical functions are placed inside the network infrastructure, leaving end clients limited capabilities of accessing the network information or taking actions. In summary, the system lacks intelligence regarding what, why and how when problems (e.g., service disruption, no access to the network, handoff loops) arise.

A. Device: Lack of Network Knowledge

On the client side, the mobile OS and apps lack open access to the below-IP cellular network operations. This creates bearers for developers and researchers to understand the exploit the distributed, large-scale network behaviors.

We now illustrate a showcase example using Google *Project Fi* [10]. The exciting *Project Fi* leverages multiple carrier networks at the end device. With multiple carriers in place, the device may select the best one over time, thus improving its access quality. Our empirical study shows that, the full benefits of multi-carrier access can be constrained by the current design of *Project Fi*. We examine *Project Fi* over two US carriers (T-Mobile and Sprint), and have found at least two main issues independent of its excellent implementations [5].

P1. No anticipated inter-carrier switch. We first discover that the anticipated switch is never triggered even when the serving carrier's coverage is pretty weak. Note that it is desirable for the device to migrate to another available carrier network for better access quality, when the device perceives degraded quality from its current, serving carrier. However, our experiments show that, the device often gets stuck in one carrier network, and misses the better network access. In one setting, T-Mobile experiences extremely weak radio coverage ($<$ -130 dBm in 4G and $<$ -110 dBm in 3G), but the phone never makes any attempt to move to Sprint, regardless of how strong Sprint's radio signal is. As a result, the device fails to improve its access quality. Moreover, we find that the expected switch often occurs until its access to the original carrier (here, T-Mobile) is lost. This is rooted in the fact that the inter-carrier switch is triggered when the serving carrier fails. Therefore, the device becomes out of service in this scenario, although better carrier access remains available.

P2. Long switch time and service disruption. We next find that the switch takes rather long time and prolongs service unavailability. Even when inter-carrier switch is eventually triggered, it may disrupt access for tens of seconds or even several minutes. In one instance, the phone starts Sprint→T-Mobile roaming, but it takes 17.3s to gain access to T-Mobile 4G. This duration is much longer than the typical handoff latency (possibly several seconds [11]). It is likely to halt or even abort any ongoing data service. We examine why the switch is slow. It turns out that, most of the switch time is wasted on an *exhaustive* scanning of all possible cells, including nearby cells from AT&T and Verizon. In the above setting, it spends 14.7s on radio-band scanning and 2.6s on completing the registration to the new carrier (here, T-Mobile). Note that, such heavy scanning overhead is not incurred by any implementation glitch. Instead, it is rooted in the *Project Fi*'s design, which selects a new carrier network only after an exhaustive scanning process. We can show that such large latency is unnecessary. It can be reduced without compromising inter-carrier selection.

The fundamental problem is that, the device does not know what has happened inside the network, which acts as a black-box to each user client. Consequently, whenever things go wrong, the device is clueless on what and why; nor does it know how to react.

B. Infrastructure: Complexity and Lack of Verification

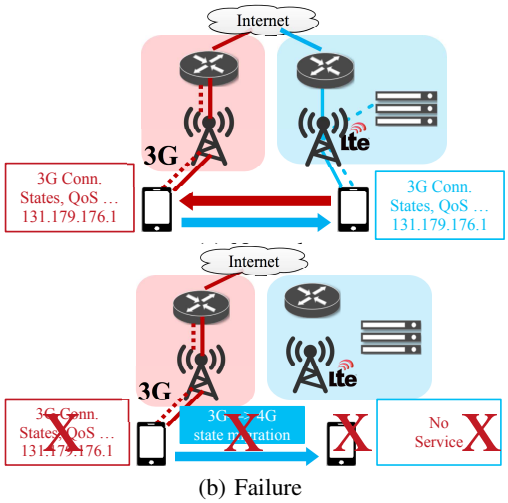
We next elaborate on two main downsides of the current infrastructure.

• **Complexity.** The complexity for design and operations is observed on both control and management planes. On the control plane, signaling protocols regulate mobility support, radio resource control, session management for data and voice, etc. Patterns of inter-protocol communication on the control plane are much richer than their Internet counterparts. In addition to the inter-layer case, they exhibit in both cross-domain and cross-system scenarios in cellular networks. Although each signaling protocol may be well designed individually, proper interactions among them in the networked environment are not guaranteed. Moreover, such problematic control plane procedures have negative impact on data-plane performance.

On the management plane, policies and configurations are defined by operators. To accommodate diverse user or carrier requests (e.g., good radio coverage, high-speed access, and load balancing), the management plane is *configurable* by design: Each base station and client can customize its own handoff parameters (such as preference values, radio signal thresholds to trigger measurement reports and handoffs) and decision logic (i.e., under what conditions handoff would be triggered). While each individual node's policy or configuration may be well justified, the interplay among policies at different cells can be problematic.

• **Lack of verification.** We find that, functions on both control and management planes suffer from lack of verifications. Consequently, they may lead to many problematic behaviors under various usage settings. These issues are not incurred by poor wireless link quality, but rooted in the control-plane and management-plane designs.

Control plane We illustrate the problem via a simple example (Figure 3). In the example, the device is initially in 4G and has its connectivity state (called EPS bearer context) activated. It then switches to 3G in one of the three usage scenarios. The EPS bearer context is subsequently deleted from 4G to release resource reservation. While in 3G, the corresponding connectivity state (called PDP context) can



(b) Failure

Figure 3: A 4G-3G switch example.

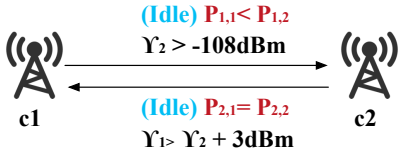


Figure 4: Instability incurred by misconfigurations.

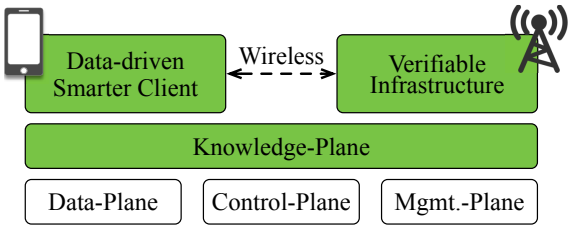


Figure 5: Enabling automated intelligence in 4G/5G.

also be deactivated for various reasons (e.g., insufficient radio resource, unacceptable QoS). However, when later switching back to 4G, the device cannot register to the 4G network, since 4G only supports PS services and EPS bearer context is required. It detaches itself and becomes out of service in 4G.

We next understand the root cause and the impact in three aspects. We first see why the PDP context is deleted in 3G. The EPS bearer context or the PDP context is essential to enabling PS services. Since 4G only supports the packet-switched (PS) mode, its state is mandatory for data service and signaling exchange. Whenever it cannot be constructed, no service access is available based on the 4G standards. On the other hand, the PDP context in 3G is allowed to be deactivated. It is not mandatory in 3G. Since 3G supports both PS and circuit switched (CS) modes, a user can still use the CS voice service without the PDP context. Deactivation of the PDP context is common in 3G. Both the network and the user device can initiate it.

Note that most smartphones do not support dual radios for both 3G and 4G. Each phone thus access one network at any time. Once being deregistered by 4G, the device has access to neither 4G nor 3G. This can last a few seconds. Of course, the device may immediately seek to re-register to 4G. It leaves the "out-of-service" state once registration succeeds. Otherwise, it keeps trying until the maximum retry count is reached. When all retries fail, the device may start to try 3G.

The fundamental problem is that, the correctness and properties of control-plane protocols are never carefully verified, nor guaranteed, under different settings. This issue is particularly visible when multiple signaling protocols interact among each other.

Management plane We further show that, management-plane misconfigurations, if left unchecked, can incur persistent handoff loops that last forever. Figure 4 illustrates a simple two-cell example. In the case, each cell locally configures its preference value, but these preferences are not globally coordinated. Specifically, c_1 configures that c_2 is more preferred to c_1 itself, but c_2 assigns equal preference to both cells. The persistent loop happens if the signal strength satisfies $\gamma_2 > \Theta_{2,1}^{high}(-108\text{dBm})$, and $\gamma_1 > \gamma_2 + \Theta_{2,1}^{high}(3\text{dBm})$. Note that, this loop can occur for *any* threshold settings (in the achievable range). Such handoff loops, despite the device being stationary at the fixed location, can have negative impacts at both the device and the infrastructure. The device continues to drain its battery, where the infrastructure generates more unwarranted signaling messages.

The fundamental problem is that, the correctness and properties of management-plane protocols are not verified for different policy configurations. Such "locally-justified, globally-uncoordinated" policy settings may lead to unexpected behaviors such as oscillations and unreachability among cells, even under excellent and constant radio qualities.

IV. Our Proposal: Enabling Automated Intelligence in Mobile Networked Systems

We propose to enable automated intelligence in the mobile networked system (shown in Figure 5). In the AI field, intelligence is defined as the ability of perceiving the environments and determining the rational actions given complex and diverse scenarios. While we do not believe a full-blown AI system is feasible for complex real systems, a mobile networked system with an enhanced degree of automated intelligence is the way to go. By enabling both client and infrastructure intelligence, the mobile networked system could reason about the deficiencies it suffers from, and improve its reliability, performance and security.

To this end, our proposal follows the "smart client, verifiable yet simple infrastructure" paradigm. By pushing more network knowledge and functions to the end host, the mobile clients could play more active roles in improving the user-experienced reliability, performance and security. Moreover, this helps to simplify the infrastructure design and operations. With the reduced complexity, the infrastructure could enhance both correctness and efficiency for control and management planes.

Illustrated in Figure 5, the resulting architecture will add a new "knowledge plane" on top of the underlying information

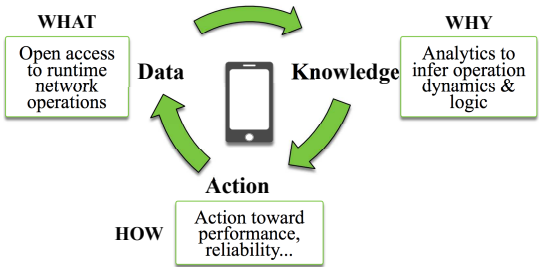


Figure 6: Data-driven smart client.

plane, which helps to learn and reason about the systems behaviors.

A. Data-Driven Smart Clients

This work enhances end intelligence with a *data-driven* design approach. It exploits the data analytics for the "black-box" cellular operations, and enhances critical services (*e.g.*, roaming) with cellular insights, thus improving application performance and security.

The client-side research is driven by the premise that, *the end devices should play a more active role in 4G/5G systems.* This substantiates a long-standing aspiration for enhanced intelligence at the end host. The challenge is that, 4G systems today still assume the "dumb terminal" design, thus masking rich network information from clients. To this end, we pursue the *data-driven* approach, without changing the infrastructure. We build systems to open up network information access, as well as its analytics, to "black-box" cellular operations. Such network data, which were not accessible before, can be leveraged to enhance critical network services.

B. Verifiable Infrastructure.

We further propose to offer provably reliable and efficient infrastructure controls and managements, by applying distributed computing and verification techniques to the mobile infrastructure design. It not only simplifies the designs/operations today, but also serves as building blocks for the future 5G systems. Along this direction, we need to pursue two dimensions: (1) verifying the correctness of existing infrastructure solutions on both control and management planes; (2) devising new verifiable primitives for the future infrastructure.

V. Ongoing Efforts on Data-Driven Client

Our goal is to ensure that the end device harnesses network information for enhanced intelligence. The approach is to let the client have access to runtime network data and analytics. Consequently, the client knows what has happened, reasons why certain actions are observed, and decides how to react. This is illustrated in Figure 6.

By exploiting data analytics, our proposal lets mobile OS and apps access the "black-box" network operations, and adapt the existing client-side mechanisms. We build systems that, for the first time, open up the network information access and its analytics of "black-box" cellular operations. Moreover,

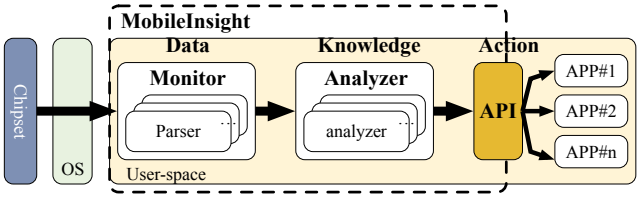


Figure 7: MobileInsight: In-device network data analytics.

with in-device network data and analytics, we enhance critical network services and improve application performance and security. This approach is readily deployable on the current clients. We next describe our ongoing efforts along the above directions.

A. MobileInsight: Enabling Runtime Cellular Analytics

The first step toward the data-driven approach is to let every client monitor and analyze "black-box" cellular operations. This calls for runtime, fine-grained information (protocol states, parameters, operation logic, etc.) from full-stack cellular protocols (physical/link layer, radio resource control, mobility management, data session management) inside the commodity phones. Unfortunately, no existing approach can meet this requirement.

We have thus designed MobileInsight, the first such open-source community tool, which enables in-device access to runtime, full-stack cellular network data and analytics [12]. In a nutshell, MobileInsight runs as a user-space service on COTS smartphones (root access required for some phone models). It does not require any extra support from operators, or additional hardware (USRP, PC or testing equipments).

MobileInsight leverages a side channel inside the commodity smartphone chipset, and extracts cellular operations from signaling messages between the device and the network. These control-plane messages regulate essential utility functions of radio access, mobility management, security, data/voice service quality, to name a few. Given these messages, it further enables in-device analytics for cellular protocols. We not only infer runtime protocol state machines and dynamics on the device side, but also infer protocol operation logic (*e.g.*, handoff policy from the carrier) from the network. MobileInsight further offers a simple and extensible API, which further facilitates researchers and developers to build new applications and services.

Figure 7 illustrates the software structure of MobileInsight, which has two components.

(i) Monitor: It first exposes raw cellular logs from the cellular interface to the device user-space at runtime, and then parses them into protocol messages and extracts their carried information elements. It builds an extensible modular framework, where each parser works on a per-protocol basis. The parsed messages are then fed to the analyzer.

(ii) Analyzer: Given the extracted messages, the analyzer aims to unveil protocol dynamics and operation logics. Based on the observed messages and the anticipated behavior model (from cellular domain knowledge), the analyzer infers protocol

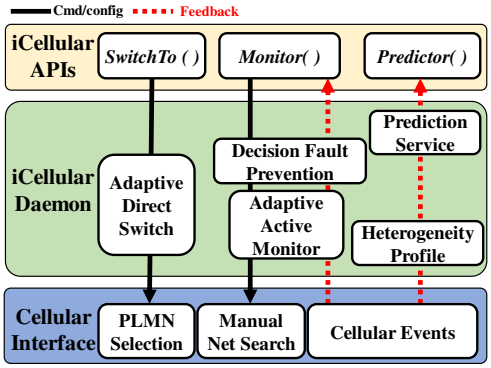


Figure 8: iCellular: Exploiting MobileInsight in Google Fi.

states, triggering conditions for state transitions, and protocol's taken actions. Moreover, it infers certain protocol operation logic (e.g., handoff) that uses operator-defined policies and configurations. It offers built-in abstraction per protocol and allows for customize these analyzers.

Therefore, MobileInsight runs as a user-space service, which offers a pure software-based solution for in-device collection and analytics of cellular protocol information. It infers protocol operations and key configurations by exploiting messages exchanged between the device and the network at the hardware chipset. It supports *fine-grained*, *per-message* information retrieval and analysis from a set of cellular-specific protocols on the control plane and at lower layers. It not only unveils *what* is going on with cellular-specific operations, but also sheds light on *why* and *how*.

B. Showcase application: iCellular

With in-device data analytics on the underlying cellular operations, the researchers and developers can better understand and exploit the closed, large-scale cellular network system. Many new research and mobile applications are made possible with this capability, including the below-IP mobile data analytics, failure diagnosis, application performance enhancements, and security loophole detections.

We next present a showcase example, iCellular [5], which is empowered by data-driven smart client. iCellular is a device-centric solution that enhances multi-carrier roaming on commodity phones. It utilizes the fine-grained, cellular-specific domain knowledge to address issues such as P1 and P2.

iCellular complements the design of *Project Fi* by leveraging low-level cellular information and mechanisms. It further empowers the device to have more control on its carrier selection. It facilitates proactive and intelligent carrier selection through machine learning techniques: it predicts the performance of heterogeneous carriers, and safeguards selections from decision faults in predictions.

Figure 8 illustrates an overview of iCellular [5]. It enhances the devices' role in every step of inter-carrier switch with runtime cellular information, spanning triggering/monitoring, decision making and switch execution. To be incrementally deployable on commodity phones, we build iCellular on top of the existing mechanisms from the phone's cellular interface. We exploit the freedom given by the standards, which allow devices to tune configurations and operations to some extent. To ensure responsiveness and minimal disruption, iCellular applies cross-layer adaptations over existing mechanisms. To facilitate the devices to make wise decisions, iCellular offers cross-layer online learning service to predict network performance, and protects devices from decision faults. To enable adaptation, prediction and decision fault prevention, iCellular incorporates realtime feedbacks extracted from low-level cellular events. Different from approaches using additional diagnosis engine (e.g., QXDM [13]) or software-defined radio, we devise an in-phone mechanism to collect realtime cellular events. These components are designed to be scalable, without incurring heavy signaling overhead to both the device and the network.

iCellular addresses both issues of P1 and P2. P1 states that, the device performs inter-carrier switch upon detecting a better carrier, even when the serving carrier is still available. This further requires the device to learn all available carriers and their quality at runtime. Note that such information can be obtained from the low-level cellular events. However, the default operation on commodity phones will not do so. Moreover, the naive approach of forcing the phone to proactively scan other carriers at any time may lead to temporary disconnection from the current carrier network. To reduce the switch time (P2), the device should refrain from exhaustive search of all carriers at all times. This requires the device to perform fine-grained control on which carriers should be scanned. It can be done by configuring the low-level mechanism for monitoring. Our empirical results show that, iCellular can improve the current *Project Fi* throughput by 23.8% on average, and 3.74x at maximum. It further reduces latency by 60.4% and 1.9x at maximum.

VI. Ongoing Efforts on Verifiable Infrastructure

Our overall goal is to study verifiable infrastructure solutions. Our current efforts focus on verifying the correctness of existing infrastructure solutions on both control and management planes.

A. Management-Plane Verification

We first verify configurations on the management plane, which affect the handoff operations. Our recent effort shows that, conflicting configurations and logics exist in reality. They violate two structural properties: (1) *stability*, whose violation will force the device to permanently oscillate among base stations, even under fixed radio/location conditions; (2) *reachability*, whose violation will let the device get stuck in a suboptimal network (e.g., 2G despite available 4G). Both property violations can result in data/voice service disruptions for users, and/or signaling storms for network carriers.

We thus design MMDiag, an in-device solution to detecting and validating instability. Figure 9 plots the design of MMDiag, which is divided into two phases: detection and validation. In the detection phase (left), the core is an MM automata, which explores possible instability cases through

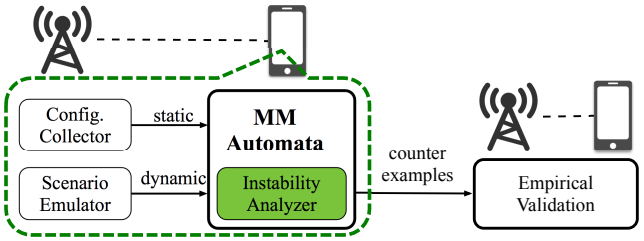


Figure 9: MMDiag for management-plane verification.

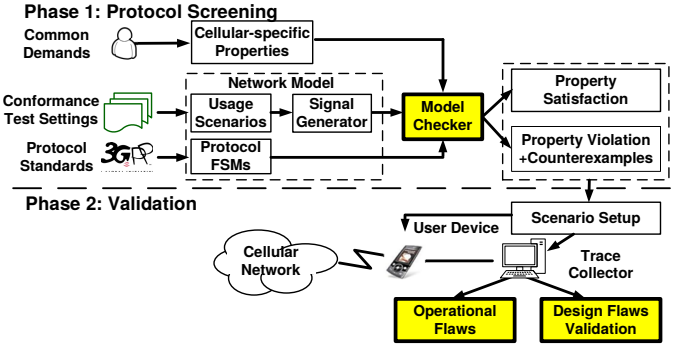


Figure 10: *CNetVerifier* for control-plane verification.

an instability analyzer and reports counterexamples if found. It models the MM decision logic based on the 3GPP standards and feeds this model with real configurations collected directly from the device and indirectly from the serving cell, as well as dynamic environment settings created for various scenarios. The instability is inferred through examining two derived instability conditions. Once they are found, we move to the device-based validation phase (right). For each counterexample, we set up the corresponding experimental scenario and conduct measurements in operational networks for validation.

Our current experiments with MMDiag over two US mobile carriers have found 21 instances that lead to handoff instability [4]. Such cases are observed over imprudent 4G-4G upgrade, 4G and femtocell settings, and hybrid 2G/3G/4G and femtocell settings.

B. Control-Plane Verification

We next verify control-plane solutions, and have thus developed *CNetVerifier* as shown in Figure 10. It helps to uncover two types of issues: *(i) design problems* originated from the 3GPP standards, and *(ii) operational slips* originated from the carrier practice.

CNetVerifier takes a two-phase approach. During the *screening* phase, *CNetVerifier* first explores *possible* logical design defects in control-plane protocols via model-checking techniques, and produces counterexamples due to design defects. Once they are found, we proceed to the *validation* phase. For each counterexample, we set up the corresponding experimental scenario and conduct measurements over operational networks for validation.

We first devise domain-specific screening techniques. It works as follows. First, we model signaling protocol interactions, and define cellular-oriented properties. Second, given these inputs, *CNetVerifier* checks whether a set of desired properties are satisfied. It thus generates a counterexample for each concrete instance of property violation, which indicates a possible design defect.

Specifically, modeling cellular protocols is derived from the 3GPP standards [14]–[17], which specify the operations for each protocol. Moreover, to model usage settings, we take the random sampling approach. We assign each usage scenario with certain probability, and randomly sample all possible usage scenarios. Specifically, for scenarios with limited options (*e.g.*, device switch on/off, all types of accept/reject requests, all inter-system switch techniques), we enumerate all possible combinations. For scenarios with unbounded options (*e.g.*, user mobility at various speed, traffic arrival patterns), we implement a run-time signal generator that randomly activates these options at any time. Last, each customizable parameter is initialized with a random value. By increasing the sampling rate, we expect that more defects can be revealed.

We further define three cellular-oriented properties: (1) Packet data services should be always available once device attached to 3G/4G, unless being explicitly deactivated. (2) Call services should also be always available. In particular, each call request should not be rejected or delayed without any explicit user operation (*e.g.*, hanging up at the originating device). (3) Inter-system mobility support should be offered upon request. For example, a 3G↔4G switch request should be served if both 3G/4G are available. We consider inter-system mobility only because intra-system mobility is seamlessly supported in practice. Note that (1) and (2) represent the expected behaviors for network services, while (3) is for mobility support. In *CNetVerifier*, these properties act as logical constraints on the data/call/mobility states.

Finally, we perform the formal model checking procedure. First, the model checker creates the entire state space by interleaving all FSMs for each individual protocol. With the constraints of three properties, some states will be marked with "error." Then we run the depth-first algorithm to explore the state transitions from the initial state (*i.e.*, the device attempting to attach to 3G/4G networks) under various usage scenarios. Once an error state is hit, a counterexample is generated for the property violation. The model checker finally generates all counterexamples and their violated properties for further experimental validation.

Given counterexamples for design defects, the validation phase needs to conduct experiments, collect protocol traces from real networks and compare them with the anticipated operations. Therefore, we leverage MobileInsight to retrieve protocol traces from user devices.

Our assessments with *CNetVerifier* over two US carriers have discovered six classes of control-plane problems. For a user study with about 20 users over two weeks, we have found 41 problematic instances that result from control-plane protocols. More details can be found in [3], [7].

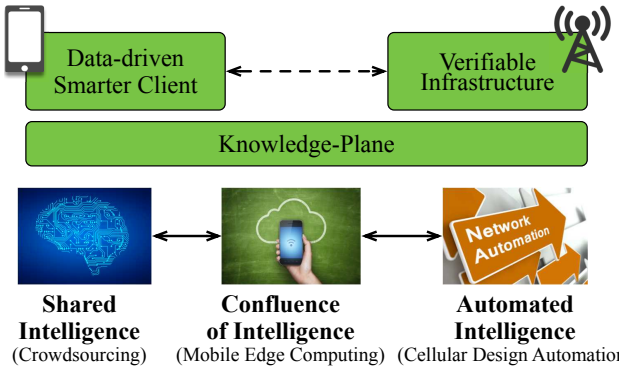


Figure 11: Intelligence-as-a-Service for 5G.

VII. Next Steps

We are at a critical stage of the mobile networked system revolution. The future 5G system has been under early standardization[1] and deployment (like Facebook's Telecom Infra Project[2]), while the beyond-5G research is being actively pursued. This offers a unique and exciting opportunity to revisit the fundamental questions in networking and mobile computing.

Along this direction, we thus propose **Intelligence-as-a-Service (IaaS)**, illustrated in Figure 11, as a new architectural primitive for future mobile networked systems to fulfill its potentials. IaaS has three elements:

Shared Client Intelligence Mobile big data can be an excellent facilitator for upcoming 5G system evolution. With massive IoT and virtual/augmented reality devices everywhere, more end intelligences will become indispensable. We thus propose to build an *information plane*, a pervasive interface that shares the networked data and extracted information between clients.

A first step toward this primitive can be a "full-stack network information map" that offers all-layer operations (from physical to data session layer) over time and in space. This is achieved by crowdsourcing massive network data from mobile devices temporally and spatially. It can be instrumental, for example, to phones when selecting the best carrier in multi-carrier access (*e.g.*, Google Project Fi). It can also be used as a client-based research platform, complementary to the $50M efforts from the White House[5] to build infrastructure-side 5G research platform.

Automated Intelligence for Infrastructure In the current cellular infrastructure, errors and failures have occurred quite frequently. They may further aggravate in 5G and beyond-5G, with more complexities from heterogeneous radio technologies, different device and infrastructure vendors, and dynamic network function virtualization. Our current effort has shown that, formal methods and distributed computing techniques are effective to prevent them and reduce manual efforts. It is possible to generalize it and construct cellular design automation (analogous to electronic design automation (EDA) in hardware design) tools for formal verification, full-coverage testing, and automatic configuration/policy synthesis.

Confluence of Client and Infrastructure Intelligence Another possible direction is intelligence fusion from both clients and the infrastructure. By sharing both information and knowledge, clients and the infrastructure could cooperatively take more intelligent actions to maximize reliability, performance and security/privacy. A promising approach is mobile edge computing: by creating *wireless edges*, the devices can offload computations *locally*. This architectural shift will raise interesting research questions, including (1) how to discover, migrate and manage computations on the edge? (2) how to ensure correct computations during inter-edge device mobility? and (3) ow to scale computations to massive IoT devices?, to name a few. These questions may be solved by bridging network communication, distributed computing, and networked storage systems.

VIII. Conclusion

The Internet is going mobile. While many clean-slate designs are explored in recent years, the current and operational 4G systems still play a critical role in the immediate future. Along this direction, the expected outcome would be a renovated 5G system design. Unfortunately, the dominant research efforts in both industry and academia have been made to accelerate the wireless link speed. While this offers an interesting direction, we believe that an equally important area, which remains largely unexplored, is the system architectural strength and weakness. This is exactly the focus of this work.

In this paper, we propose to enhance automated intelligence in the entire mobile networked system, spanning both the client and the infrastructure. We further clarify that our view on intelligence is not defined as a binary, zero or one entity. Instead, it is progressive. As more data are available and more verifiable solutions are in place, the system intelligence level will increase. The detailed approach is through data-driven design at the client and verifiable software solution to the infrastructure. If succeeded, our proposal may enable the applications, users, and operators to learn what, why and how regarding the system behaviors at runtime.

Acknowledgment

We acknowledge the generous support of National Science Foundation (NSF) on the early stage of this work through awards (CNS-1526456, CNS-1526985, CNS-1423576 and CNS-1421440). Any opinions, findings, and conclusion or recommendations expressed in this work are those of the authors and do not necessarily reflect the view of NSF or the US government.

References

[1] Cisco Visual Networking Index, "Global Mobile Data Traffic Forecast Update, 2016?2021 White Paper," 2017, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html.

[1] http://www.3gpp.org/news-events/3gpp-news/1787-ontrack_5g

[2] https://www.technologyreview.com/s/600875/facebook-enters-the-race-to-build-5g-networks/

[2] G. Tu, C. Peng, H. Wang, C. Li, and S. Lu, "How Voice Calls Affect Data in Operational LTE Networks," in *MobiCom*, Oct. 2013.

[3] G.-H. Tu, Y. Li, C. Peng, C.-Y. Li, H. Wang, and S. Lu, "Control-Plane Protocol Interactions in Cellular Networks," in *SIGCOMM*, 2014.

[4] Y. Li, H. Deng, J. Li, C. Peng, and S. Lu, "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*. ACM, 2016, pp. 261–272.

[5] Y. Li, H. Deng, C. Peng, G.-H. Tu, J. Li, Z. Yuan, and S. Lu, "iCellular: Define Your Own Cellular Network Access on Commodity Smartphones," in *Accepted by USENIX NSDI*, March 2016, draft available: http://arxiv.org/abs/1510.08027.

[6] C. Peng, Y. Li, Z. Li, J. Zhao, and J. Xu, "Understanding and Diagnosing Real-World Femtocell Performance Problems," in *INFOCOM*, April 2016.

[7] G.-H. Tu, Y. Li, C. Peng, C.-Y. Li, and S. Lu, "Detecting problematic control-plane protocol interactions in mobile networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 1209–1222, 2016.

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

[9] "Android telephonymanager class," http://developer.android.com/reference/android/telephony/TelephonyManager.html.

[10] Google, "Project fi," 2015, https://fi.google.com/about/.

[11] G. Tu, C. Peng, C. Li, X. Ma, H. Wang, T. Wang, and S. Lu, "Accounting for roaming users on mobile data access: Issues and root causes," in *MobiSys*, Jun. 2013.

[12] Y. Li, C. Peng, Z. Yuan, J. Li, H. Deng, and T. Wang, "Mobileinsight: Extracting and analyzing cellular network information on smartphones," in *ACM MobiCom*, 2016.

[13] Qualcomm, "QxDM Professional - QUALCOMM eXtensible Diagnostic Monitor," http://www.qualcomm.com/media/documents/tags/qxdm.

[14] 3GPP, "TS24.008: Core network protocols; Stage 3," Jun. 2014. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/24.008.htm

[15] ——, "TS25.331: Radio Resource Control (RRC)," 2006. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/25331.htm

[16] ——, "TS24.301: Non-Access-Stratum (NAS) for EPS; ," Jun. 2013. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/24301.htm

[17] ——, "TS36.331: Radio Resource Control (RRC)," Mar. 2015. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/36331.htm