Characterizing Radio Resource Allocation for 3G Networks

Feng Qian
University of Michigan

Zhaoguang Wang
University of Michigan

Alexandre Gerber
AT&T Labs Research

Z. Morley Mao
University of Michigan

Subhabrata Sen
AT&T Labs Research

Oliver Spatscheck
AT&T Labs Research

ABSTRACT

3G cellular data networks have recently witnessed explosive growth. In this work, we focus on UMTS, one of the most popular 3G mobile communication technologies. Our work is the first to accurately infer, for any UMTS network, the state machine (both transitions and timer values) that guides the radio resource allocation policy through a light-weight probing scheme. We systematically characterize the impact of operational state machine settings by analyzing traces collected from a commercial UMTS network, and pinpoint the inefficiencies caused by the interplay between smartphone applications and the state machine behavior. Besides basic characterizations, we explore the optimal state machine settings in terms of several critical timer values evaluated using real network traces. Our findings suggest that the fundamental limitation of the current state machine design is its *static* nature of treating all traffic according to the same inactivity timers, making it difficult to balance tradeoffs among radio resource usage efficiency, network management overhead, device radio energy consumption, and performance. To the best of our knowledge, our work is the first empirical study that employs real cellular traces to investigate the optimality of UMTS state machine configurations. Our analysis also demonstrates that traffic patterns impose significant impact on radio resource and energy consumption. In particular, We propose a simple improvement that reduces YouTube streaming energy by 80% by leveraging an existing feature called *fast dormancy* supported by the 3GPP specifications.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design – Wireless Communication; C.4 [**Performance of Systems**]: Measurement Techniques

General Terms

Measurement, Algorithms

IMC'10, November 1–3, 2010, Melbourne, Australia.

Keywords

UMTS, 3G Networks, RRC state machine, Inactivity timers, Tail effects, Smartphones, Multimedia streaming

1. INTRODUCTION

3G cellular data networks have recently witnessed rapid growth, especially due to the emergence of smartphones. In this paper, we focus on the UMTS (the Universal Mobile Telecommunications System) 3G network, which is among the most popular 3G mobile communication technologies. As evidence of this popularity, in Q3 2008, among the 400 million of worldwide 3G users, 300 million were UMTS subscribers [1].

Compared to WiFi, 3G systems operate under more radio resource constraints. To efficiently utilize the limited radio resources, UMTS introduces for each *user equipment* (UE, *i.e.,* a smartphone) a *radio resource control* (RRC) state machine that determines radio resource usage affecting device energy consumption and user experience. Usually a UE (user equipment) can be in one of three states, each with different amount of allocated radio resources. The transitions between states also have significant impact on the UMTS system. Frequent state promotions (resource allocation) may lead to unacceptably long delays for the UE, as well as additional processing overheads for the radio access network [10, 21]. State demotions (resource release) are controlled by critical inactivity timers affecting radio resource utilization and UE energy consumption.

The current design of the RRC state machine appears to be ad-hoc with statically configured parameters. Our work systematically studies its design using real cellular traces from a large cellular ISP and analyzes the effect on important factors from both the network operator's perspectives, namely radio resource usage efficiency and management overhead, and from end-user's perspective, namely device energy consumption and application performance. We examine tradeoffs among these factors. In particular, we focus on settings of critical inactivity timer values that determine when to release radio resources after a period of inactivity.

As an example of the challenge in balancing the tradeoffs, using real UMTS cellular traces we observe that decreasing one inactivity timer by three seconds can reduce the overall radio resource usage by 40%, but increasing the number of state promotions by 31%. On the other hand, increasing the inactivity timer effectively enhances end user experience and reduces the management overhead, however at the expense of low efficiency in radio resource utilization and energy consumption. Intuitively, the optimal timer settings heavily depend on application traffic patterns and thus can benefit from *traffic awareness* via trace-driven tuning.

In this paper, we undertake a detailed exploration of the RRC state machine and its optimizations using real traces from a large cellular ISP. In particular, we make the following contributions.

1. Accurate inference of the RRC state machine. Different carriers may adopt different state machine models with varied parameters. Accurate inference is therefore the very first necessary step towards characterizing and improving the RRC state machine. We propose a novel inference technique purely based on probing from the user device. It systematically discovers the state transitions by strategically adjusting the packet dynamics. We applied our algorithm to two UMTS carriers and validated its accuracy by measuring the device power consumption.

2. Characterization of state machine behaviors. The current RRC state machine parameters are either empirically configured in an ad-hoc manner [7], or determined using analytical traffic models [20, 32]. The latter approach, however, suffers from several limitations. *(i)* The expressiveness of an analytical model is quite limited and is unlikely to capture, using a statistical distribution with a few parameters, the characteristics of real-world traffic patterns generated by millions of cellular users. *(ii)* The existence of concurrent applications accessing the network further increases the difficulty of modeling the packet dynamics.

We systematically characterize the impact of existing operational state machines by analyzing traces collected from a commercial UMTS network. We found that short data transfers severely suffer from the state promotion delay, and significant portions (up to 45.3%) of the occupation time of the high-speed dedicated transmission channel is wasted on the idle time period matching the inactivity timer value before a state demotion, which is called *tail* time [14]. We explore the optimal timer values by replaying the trace against different state machine settings with varying parameters. Our findings suggest that the fundamental limitation of the current state machine design is its *static* nature of treating all packets according to the same inactivity timer, making it difficult to balance the aforementioned tradeoffs. We also observe that applications exhibit different sensitivities to the change of inactivity timers due to their different traffic patterns. To the best of our knowledge, our work is the first empirical study that employs real cellular traces to investigate the optimality of RRC state machine configurations.

3. Analysis of multimedia streaming strategies. We study the streaming approaches employed by Pandora audio [5] and YouTube video streaming, the two most popular smartphone multimedia streaming applications contributing large traffic volume. Our analysis demonstrates that traffic patterns impose significant impact on the radio resource and energy consumption, again due to the interplay between the UE application and the RRC state machine. The current Pandora approach incurs long tail periods, which waste 50% of the dedicated channel time and 59% of the radio energy, while the YouTube strategy suffers from long dedicated channel occupation time due to bandwidth underutilization. We propose a simple improvement that saves the YouTube streaming energy by 80% by leveraging an existing feature called *fast dormancy* supported by 3GPP specifications [8, 9].

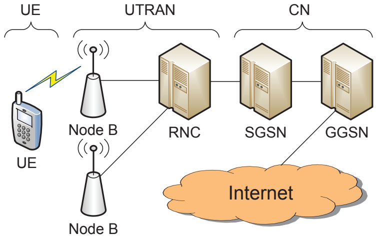


Figure 1: The UMTS architecture

4. Data preprocessing scheme for cellular data traces. We are the first to highlight the need and address the challenge of preprocessing cellular network traces, which differ from Internet traffic data in that cellular data can reflect significant delays imposed by state promotions transitioning from low power to higher power state due to radio resource allocation overhead. To extract the actual traffic patterns, *e.g.,* for the purpose of evaluating alternate designs of the RRC state machine, such delays must be eliminated.

2. BACKGROUND

This section provides sufficient background for further discussions of this paper.

2.1 The UMTS Network

As illustrated in Figure 1, the UMTS network consists of three subsystems: User Equipments (UE), UMTS Terrestrial Radio Access Network (UTRAN), and the Core Network (CN). UEs are essentially mobile handsets carried by end users. The UTRAN allows connectivity between a UE and the CN. It consists of two components: base stations, called Node-Bs, and Radio Network Controllers (RNC), which control multiple Node-Bs. Most UTRAN features such as packet scheduling, radio resource control, and handover control are implemented at the RNC. The centralized CN is the backbone of the cellular network. In particular the GGSN (Gateway GPRS Support Node) within the CN serves as a gateway hiding UMTS internal infrastructures from the external network.

2.2 The RRC State Machine

In the context of UMTS, the *radio resource* refers to WCDMA codes that are potential bottleneck resources of the network. To efficiently utilize the limited radio resources, the UMTS radio resource control (RRC) protocol introduces a state machine associated with each UE. There are typically three RRC states as described below [25, 19].

IDLE. This is the default state when a UE is turned on. The UE has not yet established an RRC connection with the RNC, thus no radio resource is allocated, and the UE cannot transfer any user data (as opposed to control data).

CELL_DCH. The RRC connection is established and a UE is usually allocated dedicated transport channels in both downlink (DL, RNC to UE) and uplink (UL, UE to RNC) direction. This state allows a UE to fully utilize radio resources for user data transmission. We refer to CELL_DCH as DCH henceforth. A UE can access HSDPA/HSUPA (High Speed Downlink/Uplink Packet Access) mode, if supported by the infrastructure, at DCH state. For HSDPA, the high speed transport channel is not dedicated, but shared by a limited number (*e.g.,* 32) of users [19]. Further, when a large

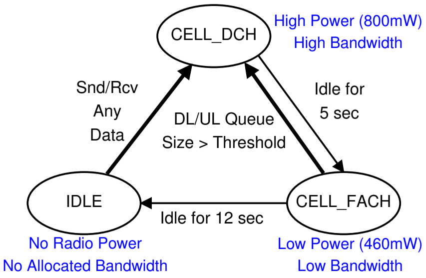


Figure 2: The RRC state machine for the 3G UMTS network of Carrier 1

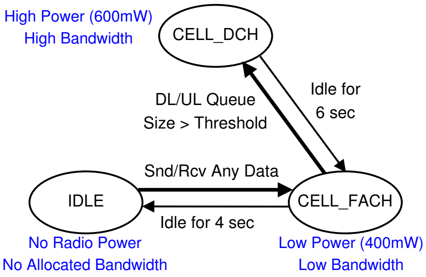


Figure 3: The RRC state machine for the 3G UMTS network of Carrier 2

Table 1: Inferred parameters for two carriers

Inactivity timer	Carrier 1	Carrier 2
α: DCH→FACH	5 sec	6 sec
β: FACH→IDLE	12 sec	4 sec
Promotion time	Carrier 1	Carrier 2
IDLE→FACH	N/A	0.6 sec
IDLE→DCH	2 sec	N/A
FACH→DCH	1.5 sec	1.3 sec
RLC Buffer threshold	Carrier 1	Carrier 2
FACH→DCH(UL)	543 ± 25 B	151 ± 14 B
FACH→DCH(DL)	475 ± 23 B	119 ± 17 B
State radio power	Carrier 1	Carrier 2
DCH/FACH/IDLE	800/460/0 mW	600/400/0 mW
Promotion radio power	Carrier 1	Carrier 2
IDLE→FACH	N/A	410 mW
IDLE→DCH	550 mW	N/A
FACH→DCH	700 mW	480 mW

Table 2: Optimize radio resources: the key tradeoff

Increase α or β timers	**Decrease α or β timers**
ΔD increases *Increase* tail time (§5.2) *Waste* radio resources	ΔD decreases *Decrease* tail time *Save* radio resources
ΔS decreases *Reduce* state promotions *Reduce* RNC overhead *Improve* user experiences	ΔS increases *Increase* state promotions *Increase* RNC overhead *Degrade* user experiences
ΔE increases *Waste* UE radio energy	ΔE decreases *Save* UE radio energy

number of UEs are in DCH state, the radio resources may be exhausted due to the lack of channelization codes in the cell. Then some UEs have to use low-speed shared channels although their RRC states are still DCH.

CELL_FACH. The RRC connection is established but there is no dedicated channel allocated to a UE. Instead, the UE can only transmit user data through shared low-speed channels that are typically less than 15kbps. We refer to CELL_FACH as FACH from this point on. FACH is designed for applications requiring very low data throughput rate. It consumes much less radio resources than DCH does.

RRC states impact a UE's energy consumption. A UE at IDLE consumes almost no energy from its radio interface. The radio power consumption for DCH is 50% to 100% higher than that for FACH (Table 1). While within the same state, the radio power is fairly stable regardless of the data throughput when the signal strength is stable. Further, the RRC state machine is maintained at both the UE and the RNC. The two peer entities are always synchronized via control channels except during transient and error situations. Also note that both the downlink (DL) and the uplink (UL) use the same state machine.

In the RRC state machine, there are two types of state transitions. *State promotions*, including IDLE→FACH, IDLE→DCH, and FACH→DCH, switch from a state with lower radio resource and UE energy utilization to another state consuming more resources and UE energy. *State demotions*, consisting of DCH→FACH, FACH→IDLE, and DCH→IDLE, go in the reverse direction. Depending on the starting state, a state promotion is triggered by either any user data transmission activity, if the UE is at IDLE, or the per-UE queue size, called Radio Link Controller (RLC) buffer size, exceeding a threshold in either direction, if the UE is at FACH.

The state demotions are triggered by two inactivity timers maintained by the RNC. We denote the DCH→FACH timer as α, and the FACH→IDLE timer as β. At DCH, the RNC resets the α timer to T seconds, a fixed threshold, whenever it observes any UL/DL data frame. If there is no user data transmission activity for T seconds, the α timer times out and the state is demoted to FACH. A similar scheme is used for the β timer for the FACH→IDLE demotion.

Promotions involve more work than demotions do. In particular, state promotions incur a long "ramp-up" latency of up to 2 seconds during which tens of control messages are exchanged between a UE and the RNC for resource allocation (*e.g.,* radio bearer reconfiguration and RRC connection setup). Excessive state promotions increase the management overhead at the RNC and degrade user experience [10, 21, 28], especially for short data transfers, which we investigate in §5.1.

Figures 2 and 3 depict our inferred state machine models for two large UMTS carriers, based on our inference methodology described in §3. Their difference naturally introduces the problem of seeking the optimal state machine configuration to better balance radio resource utilization and performance. We quantitatively compare both carriers in §6.4.

2.3 Tradeoff Considerations to Optimize Resource Allocation

As discussed in §1, the RRC state machine introduces tradeoffs among radio resource utilization, UE energy consumption, end user experience, and management overheads at the RNC. We need to quantify these factors to analyze the tradeoff. Given a cellular trace and a state machine configuration C, we compute three metrics to characterize the above factors. Previous work either consider only one factor [20, 32] or focus on other metrics (*e.g.,* dropping rate due to congestion [22] and web page response time [31]), using analytical models. We detail our methodology for computing the three metrics in §6.

- The DCH state occupation time, denoted by $D(C)$, quantifies the overall radio resources consumed by UEs on dedicated channels in DCH state. We ignore the relatively low radio resources allocated for shared low-speed channels on FACH.
- The number of state promotions, denoted by $S(C)$, is the total number of IDLE→DCH, IDLE→FACH, and FACH→DCH promotions. $S(C)$ quantifies the overhead brought by state promotions that worsen user experience and increase the management overhead at the RNC. We ignore the state demotion overhead as it is significantly smaller compared with the state promotion overhead.
- The energy consumption, denoted by $E(C)$, is the total energy consumed by radio interfaces of all UEs in the trace.

We are interested in relative changes of D, S, E when we switch to a new state machine using the same trace. Let C be the default state machine used as the comparison baseline, and let C' be a new state machine configuration. The relative change of D, denoted as ΔD, is computed by $\Delta D(C') = (D(C') - D(C))/D(C)$. We have similar definitions for ΔS and ΔE.

As we shall see throughout this paper, the key tradeoff expressed in our notations is that, for any state machine setting, increasing ΔS causes both ΔD and ΔE to decrease (there may exist exceptions when ΔS is too large). In other words, if more state promotions are allowed, then we can save more radio resources and UE energy. Ideally, we want to find a state machine configuration C' such that $\Delta D(C')$ and $\Delta E(C')$ are significantly negative, while $\Delta S(C')$ is reasonably small. This important tradeoff is summarized in Table 2.

3. INFERRING THE STATE MACHINE

We describe an end-host based probing technique for inferring the state machine, which we validate using power measurements. Accurate inference of the state machine and its parameters is the first necessary step towards characterizing and improving the RRC state machine. We study two large 3G carriers with results presented in Figures 2, 3, and Table 1, which will be used in our simulation program described in §4.

3.1 Methodology

We make the following assumptions for the inference algorithm. *(i)* There are at most three states: IDLE, FACH, and DCH. DCH is the state allowing high data rate transfer. *(ii)* The time granularity for inactivity timers is assumed to be seconds. Our algorithms can easily adapt to finer granularities. *(iii)* The state promotion delay is significantly longer than (at least two times as) a normal RTT for both DCH and FACH (less than 300 ms based on our measurements). This is reasonable due to the promotion overhead explained earlier. *(iv)* We roughly know the range of RLC buffer thresholds (64B–1KB) that trigger the FACH→DCH promotion. We detail our methodology below.

State promotion inference determines one of the two promotion procedures adopted by UMTS: $P1$: IDLE→FACH→DCH, or $P2$: IDLE→DCH. Algorithm 1 illustrates how we distinguish between $P1$ and $P2$, where *min* and *max* denote RLC buffer sizes that does not trigger, and does trigger, the FACH→DCH promotion, respectively. Note that IDLE→DCH or IDLE→FACH always happens regardless of the RLC buffer size. The idea is to distinguish $P1$ and $P2$ by detecting the presence of the FACH→DCH promotion. We set *min* and *max* to 28 bytes (an empty UDP packet plus an IP header) and 1K bytes, respectively. If $P1$ holds, then the state is promoted to FACH after Step 2, and then further promoted to DCH at Step 3. Thus Δt includes an additional FACH→DCH promotion delay. Otherwise, for $P2$, Δt does not include the promotion delay since the state is already DCH after Step 2.

Algorithm 1 State promotion inference

```
1: Keep UE on IDLE.
2: UE sends $min$ bytes. Server echoes $min$ bytes.
3: UE sends $max$ bytes. Server echoes $min$ bytes.
4: UE records the RTT $\Delta t$ for Step 3.
5: Report $P1$ iff $\Delta t \gg$ normal RTT. Otherwise report $P2$.
```

Algorithm 2 State demotion inference

```
1: for $n = 0$ to 30 do
2:     UE sends $max$ bytes. Server echoes $min$ bytes.
3:     UE sleeps for $n$ sec.
4:     UE sends $min$ bytes. Server echoes $min$ bytes.
5:     UE records the RTT $\Delta t_1(i)$ for Step 4.
6: end for
7: for $n = 0$ to 30 do
8:     UE sends $max$ bytes. Server echoes $min$ bytes
9:     UE sleeps for $n$ sec.
10:    UE sends $max$ bytes. Server echoes $min$ bytes.
11:    UE records the RTT $\Delta t_2(i)$ for Step 10.
12: end for
13: Report $D1$ iff $\Delta t_1(\cdot)$ and $\Delta t_2(\cdot)$ are similar, else report $D2$.
```

State demotion inference determines whether UMTS uses $D1$: DCH→IDLE or $D2$: DCH→FACH→IDLE. The inference method is shown in Algorithm 2, which consists of two experiments. The first experiment (Steps 1 to 6) comprises of 30 runs. In each run, the UE goes to DCH by sending *max* bytes (Step 2), sleeps for n seconds (Step 3), then sends *min* bytes (Step 4). Recall that *min* and *max* denote RLC buffer sizes that does not trigger, and does trigger, the FACH→DCH promotion, respectively. By increasing n from 0 to 30, we fully exercise all the states experienced by the UE at the beginning of Step 4 due to the inactivity timer effects. The second experiment (Step 7 to 12) is similar to the first one except that after Step 10, the UE always promotes to DCH. In contrast, after Step 4, if there exists a DCH→FACH demotion, the UE will be in FACH. Therefore, for $D1$ the observed RTTs for two experiments, $\Delta t_1(0..30)$ and $\Delta t_2(0..30)$, will be similar. On the other hand, for $D2$, *i.e.,* the state is demoted to FACH (the α timer), then back to IDLE (the β timer), then for $\lfloor\alpha\rfloor < i \leq \lfloor\alpha+\beta\rfloor$, the difference between $\Delta t_1(i)$ and $\Delta t_2(i)$ is roughly the FACH→DCH promotion delay.

Inferring other parameters. Given the process of inferring the state transitions, it is easy to infer related parameters. First, the inactivity timers can be directly obtained from $\Delta t_1(\cdot)$ and $\Delta t_2(\cdot)$ computed by Algorithm 2. For the case where the demotion is DCH→FACH→IDLE, we can deduce α and β from the fact that $\Delta t_1(0...\lfloor\alpha+\beta\rfloor)$ are smaller than $\Delta t_1(\lceil\alpha+\beta\rceil...30)$, and $\Delta t_2(0...\lfloor\alpha\rfloor)$ are smaller than

$\Delta t_2(\lceil\alpha\rceil...30)$. This is because in the first experiment in Algorithm 2, a state promotion (IDLE→FACH or IDLE→DCH) will not happen until $n \geq \lceil\alpha+\beta\rceil$, while in the second experiment, a state promotion from IDLE or FACH happens when $n \geq \lceil\alpha\rceil$. Similarly, for the case where the demotion is DCH→IDLE, let the only inactivity timer be γ. Then we will observe that $\Delta t_1(0...\lfloor\gamma\rfloor)$ are much smaller than $\Delta t_1(\lceil\gamma\rceil...30)$. Second, to infer the promotion delay X→Y, we measure the entire RTT including the promotion, then subtract from it the normal RTT (*i.e.,* the RTT not including the promotion) on state Y. Finally, using the promotion delay as an indicator, we infer the RLC buffer threshold by performing binary search for the packet size that exactly triggers the FACH→DCH promotion, in each direction.

3.2 Results on State Machine Inference

We present the inference results for state machines used by two large UMTS carriers: Carrier 1 and 2. For each carrier, we repeat Algorithm 1 and Algorithm 2 for three times, ensuring that in each experiment *(i)* the server does not experience a timeout; *(ii)* in tcpdump trace, we never observe other user data transmission that may trigger a state transition; *(iii)* the 3G connection is never dropped. The entire experiment is discarded if any of these conditions is violated.

For the state promotion inference, the normal RTTs for Carrier 1 and 2 are less than 0.3 sec, and the measured Δt values in Algorithm 1 are 0.2 sec for Carrier 1, and 1.5 sec for Carrier 2, for all three trials. Based on Algorithm 1, we conclude that the promotion procedures for Carrier 1 and Carrier 2 are IDLE→DCH and IDLE→FACH→DCH, respectively. For the state demotion inference, we notice the qualitative difference between $\Delta t_1(5...16)$ in Figure 4(a) and $\Delta t_2(5...16)$ in Figure 4(b), indicating that the state demotion procedure for Carrier 1 is DCH→FACH→IDLE. Similarly, Figure 5(a) and Figure 5(b) imply that Carrier 2 also uses DCH→FACH→IDLE, due to the obvious difference between $\Delta t_1(6..9)$ and $\Delta t_2(6..9)$.

We note that for Carrier 1, $\Delta t_1(17...30)$ and $\Delta t_2(17...30)$ are roughly the same, because in Algorithm 2, for $17 \leq n \leq 30$, either sending min bytes (Step 4) or sending max bytes (Step 10) triggers an IDLE→DCH promotion, which is the only promotion transition for Carrier 1. In contrast, for Carrier 2, $\Delta t_1(10...30)$ is smaller than $\Delta t_2(10...30)$. Carrier 2 may perform two types of promotions depending on the RLC buffer size, therefore for $10 \leq n \leq 30$ in Algorithm 2, sending min bytes in Step 4 and sending max bytes in Step 10 will trigger IDLE→FACH and IDLE→FACH→DCH, respectively, resulting in different promotion delays. This observation does not affect the inference results of Algorithm 2 for either carrier.

Given the $\Delta t_1(\cdot)$ and $\Delta t_2(\cdot)$ values computed by Algorithm 2, it is easy to infer α and β by following the logic described in §3.1. The inference results are $(\alpha,\beta) = (5sec, 12sec)$ for Carrier 1 and $(\alpha,\beta) = (6sec, 4sec)$ for Carrier 2. To infer the RLC buffer thresholds, we repeat the experiments 30 times and summarize the results in Table 1[1]. Although their variances are higher than those of other parameters, we found that they have very small impact on our measurement results described later.

We investigated the stability of the state machine and found that the state transitions and timer values do not change over the four months of our study (July to November 2009). They are also independent of the time of the day, the Access Point Name (APN), or the location[2]. Also the state promotion delays are largely static.

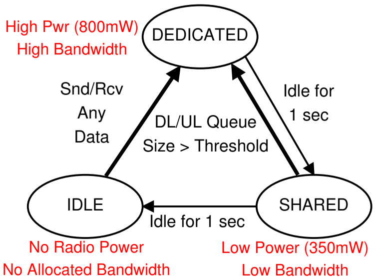


Figure 6: The RRC state machine for the 2G GPRS/EDGE network of Carrier 2

For 2G (GPRS/EDGE) networks, there exists a similar RRC state machine model. The three RRC states are "IDLE", "CELL_SHARED", and "CELL_DEDICATED" [6], corresponding to IDLE, FACH, and DCH in the 3G case, respectively. We applied our inference methodology (using a smaller increment of n in Algorithm 2) on Carrier 2's 2G network, and show the inference results in Figure 6. We observe that the inactivity timers (1 sec) are much shorter, therefore resulting in better efficiency of radio resource utilization and UE energy consumption. The negative impact of short timers is more frequent state transitions. The state promotion delays of IDLE→DEDICATED and SHARED→DEDICATED are both 0.5 sec.

3.3 Validation using Energy Consumption

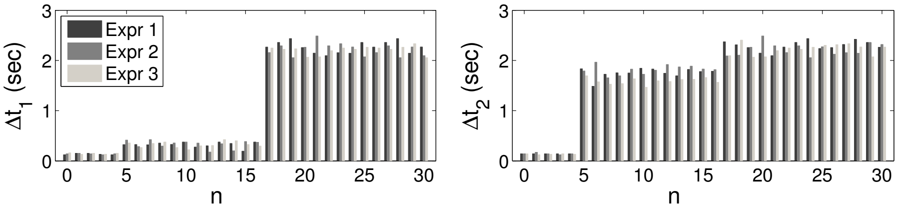
As described in §2, a UE's radio energy consumption differs for each state, a property we may use to infer the state machine. However, accurately measuring energy consumption requires special monitoring equipments. So we use it as validation for our inference algorithms, which only require UE-based probing.

We set up experiments to confirm the inactivity timers and state promotion delays for Carrier 1 by monitoring UE's energy consumption as follows. The battery of an HTC TyTN II smartphone is attached to a hardware power meter [4], which is also connected via USB to a PC that records fine-grained power measurements by sampling the current drawn from the battery at a frequency of 500 Hz. Figure 7 shows one representative experimental run of the validation. During probing, we keep the UE's LCD at the same brightness level, turn off GPS and WiFi, and disable all network activities. After keeping the smartphone in this inactive state for 20 sec, we send a UDP packet at $t = 23.8s$ thus triggering an IDLE→DCH promotion that takes approximately 2 sec as inferred in §3.1. From $t = 26.1s$, the phone remains at the high-power DCH state for about 5 sec, then switches to the low-power FACH state at $t = 31.5s$. Finally at $t = 44.1s$, the phone returns to the IDLE state. The measured inactivity timer values are longer than the inferred ones by about 10%, likely due to the synchronization overhead between the RNC and the UE. We similarly verified

---

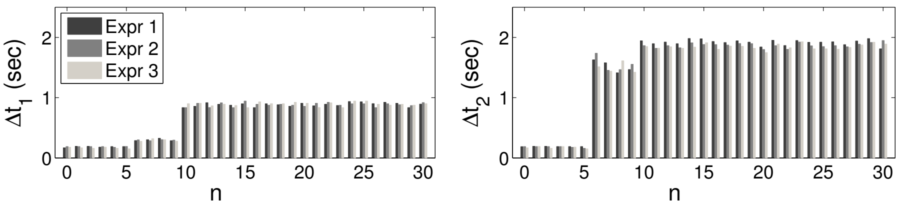
[1] Results in Table 1 were measured in November 2009.

[2] We did probing at two locations that are 600 miles apart in the U.S., using an HTC TyTN II smartphone and a Sierra 3G Air card to perform our experiment.



(a) Algorithm 2 ($\Delta t_1(\cdot)$) (b) Algorithm 2 ($\Delta t_2(\cdot)$)

Figure 4: State machine inference results for Carrier 1



(a) Algorithm 2 ($\Delta t_1(\cdot)$) (b) Algorithm 2 ($\Delta t_2(\cdot)$)

Figure 5: State machine inference results for Carrier 2

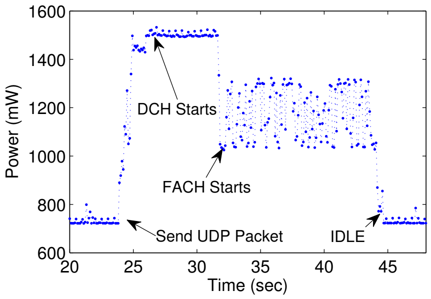


Figure 7: Validation using power measurements

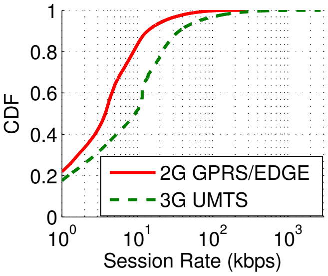


Figure 10: Session rate distribution over sessions longer than 100ms for 2G and 3G data

the FACH→DCH promotion delay, and validated Carrier 2's state machines for both 2G and 3G.

Using the IDLE power as baseline, we compute the power consumption of the 3G radio interface as shown in Table 1[3]. We also infer the RLC buffer thresholds for the FACH→DCH promotion by performing binary search. Instead of using the promotion delay as described in §3.1, we use energy as an indicator to search for the RLC buffer threshold that exactly triggers the promotion, for each direction. The validation results are consistent with our inference results.

4. 3G MEASUREMENT DATA

This section discusses the traffic data used in our study (§5 and §6). We first describe the raw dataset in §4.1, then in §4.2, we explain our preprocessing methodology to eliminate the side effects brought by the RRC state machine on the dataset. This enables us to use the preprocessed trace to experiment with different state machine configurations. In §4.3, we describe the extraction procedure for application-specific data for further analysis.

4.1 The Raw Dataset

Our dataset is a large TCP header packet trace collected from Carrier 1 on January 29, 2010 in the normal course of operations. The collection point is one GGSN that primarily serves 3G UMTS users but also 2G GPRS users. Non-3G traffic are filtered by excluding SGSNs that exclusively serve 2G users. Sampling was performed on a per GTP (GPRS Tunneling Protocol) session basis with a sampling rate of 1:16, so that all packets in both directions from a sampled GTP session were captured. Our trace contains 278 million TCP packets (162 GB data) of 3G traffic continuously captured in 3 hours. Due to concerns of large traffic volume and user privacy issues, we only recorded TCP/IP headers and a 64-bit timestamp for each packet, but no subscriber IDs or phone numbers.

Subsequently, we extract *sessions* (they are different from GTP sessions) from the trace, with each session consisting of all packets transferred by the same UE identified through the private client IP address present in the trace. It is known that cellular public IPs change very quickly [12]. But private IPs of Carrier 1 are much less dynamic, changing only at the interval of tens of minutes. Multiple TCP flows from concurrent applications may be mixed in the same session. We use a threshold of 60 sec of idle time to decide that a session has terminated. Changing this value to other values such as 45 or 75 sec does not qualitatively affect the anal-

---

[3] The power values in Table 1 were measured in good signal strength conditions. [29] shows that signal strength may have significant impact on a UE's radio power consumption.

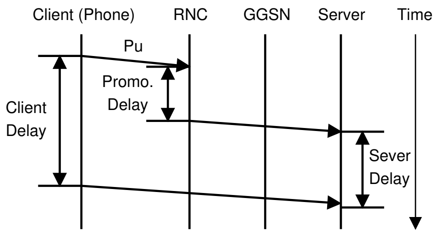


Figure 8: State promotion triggered by an UL packet

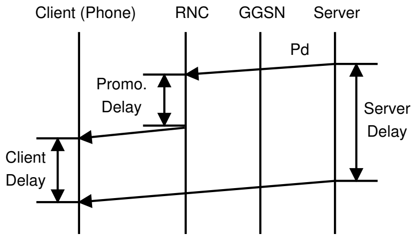


Figure 9: State promotion triggered by a DL packet

ysis results. Besides, as shown in Figure 10, we do observe qualitative difference of session rate (total bytes divided by the duration of a session) between 3G and 2G. Finally, we extracted about 1.0 million 3G sessions from the raw trace.

We discuss three limitations of our dataset. First, similar to previous measurements of wired network using passive trace [33, 26], the finite duration (3 hours) may influence distributions of state machine characteristics. Second, the dataset does not contain UDP traffic. A recent study [26] indicates that UDP accounts for less than 8% of the traffic volume for a wired VPN link. We expect the percentage of UDP traffic for cellular networks to be even smaller because currently UDP-based streaming and gaming applications on smartphones are not as popular as those on PCs. The third limitation relates to the data collection location (GGSN). We address this issue next in §4.2. Overall, we believe that the above limitations do not qualitatively affect our state machine analysis results.

4.2 Data Preprocessing

One issue regarding our cellular dataset is that the packet timestamps recorded by the GGSN are affected by delays introduced by RRC state promotions that may take up to 2 seconds to finish. Ideally we would like to capture traffic traces directly at the RNC to understand the network interaction between UEs and the radio access network so that we can use the trace to experiment with different state machine configurations. However, the data collection point, GGSN, is at the upstream from the RNC where state promotions take place. There are two scenarios for this impact. First, if an uplink packet P_u from client (UE) to the server triggers a state promotion, *e.g.,* from IDLE to DCH, which happens *before* P_u is captured by the GGSN. Thus, the GGSN records a later timestamp than the actual arrival time of P_u as shown in Figure 8. In this case, we need to shift backward in time for packet P_u and all its subsequent packets within this session by the promotion delay to preserve the correct time spacing among packets. This case is detected by simulating the state machine for each session.

The second case depicted in Figure 9 is similar except that the state promotion is triggered by a downlink packet P_d from the server to the client, so that the promotion takes place *after* the GGSN observes P_d. The packet P_d received by the client will be delayed by the state promotion, therefore the echoing uplink packet from the client and any other subsequent packets should have their timestamps shifted back by the promotion delay to exclude the effect of the promotion delay on the inter-packet time spacing.

To verify the accuracy of this preprocessing procedure, we performed controlled experiments shown in Figures 8 and 9. From the UE (client), starting from IDLE, we send two UDP packets P_1 and P_2 (two uplink packets sent in a row in Figure 8, their interval observed at the client is less than α) where P_1 triggers a state promotion, then the difference between the latency between P_1 and P_2 of the client and that of the server is approximately the state promotion delay. Similarly we verified the downlink case.

Table 3: Cellular traces of five applications

Application	Sessions	Bytes	Description
Email-1	15K	1.4 GB	Email provider 1
Email-2	11K	536 MB	Email provider 2
Sync	4.9K	62 MB	Synchronization service
Stream	2.7K	395 MB	Pandora audio streaming [5]
Map	1.8K	60 MB	Interactive mapping

The main purpose of this data preprocessing is to eliminate the effects brought by the non-trivial promotion time while keeping other packet dynamics unchanged. It enables us to experiment with different state machine parameters or to apply a different state machine on the preprocessed trace in our further analysis. Given a session extracted in §4.1, we replay it against the original state machine of Carrier 1. If a state promotion is detected, then we shift relevant packets accordingly to remove the promotion delay. The preprocessing also detects sessions that violate the state machine. For example, we assume the state is FACH after P_k is transmitted. An uplink packet P_{k+1} triggers a FACH→DCH promotion, but the inter-arrival time between P_k and P_{k+1} is less than the FACH→DCH promotion delay. We expect that such violations are mostly caused by the variation of the RLC buffer thresholds (our simulation program assumes that they have fixed values). Sessions violating the state machine account for only 3% of the total sessions, and are not used in our subsequent data analysis.

4.3 Application-specific Data Extraction

From the preprocessed trace, we select five traffic types each corresponding to a particular application as shown in Table 3. For each application, we extract sessions in which at least 95% of the packets have either the source or the destination as one fixed server IP, thus eliminating coexistence of other applications as one session may contain multiple TCP flows of concurrent applications. For example, "Sync" consists of 4.9K sessions of a popular synchronization service. All its sessions access the same server that synchronizes emails, calendar events, contacts *etc.* between PCs and a UE using push-based notification mechanism. We use the five datasets for per-application analysis in §5.2 and §6.3 to study how application traffic patterns affect the tradeoff described in §2.3.