Profiling Resource Usage for Mobile Applications: a Cross-layer Approach

Feng Qian¹, Zhaoguang Wang¹, Alexandre Gerber², Z. Morley Mao¹, Subhabrata Sen², Oliver Spatscheck²

¹University of Michigan  ²AT&T Labs - Research
Introduction

• Typical testing and optimization in cellular data network

• Little focus has been put on their cross-layer interactions
  Many mobile applications are not cellular-friendly.

• The key coupling factor: the RRC State Machine
  – Application traffic patterns trigger state transitions
  – State transitions control radio resource utilization, end-user experience and device energy consumption (battery life)
The RRC State Machine for UMTS Network

- State promotions have **promotion delay**
- State demotions incur **tail times**

![State Machine Diagram]

<table>
<thead>
<tr>
<th>Channel</th>
<th>Radio Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLE</td>
<td>Not allocated</td>
</tr>
<tr>
<td>CELL_FACH</td>
<td>Shared, Low Speed</td>
</tr>
<tr>
<td>CELL_DCH</td>
<td>Dedicated, High Speed</td>
</tr>
</tbody>
</table>
Example: RRC State Machine for a Large Commercial 3G Network

- **DCH**: High Power State (high throughput and power consumption)
- **FACH**: Low Power State (low throughput and power consumption)
- **IDLE**: No radio resource allocated

Tail Time
Waiting inactivity timers to expire
Example of the State Machine Impact: Inefficient Resource Utilization

A significant amount of channel occupation time and battery life is wasted by **scattered bursts**.

State transitions impact end user experience and generate signaling load.

Analysis powered by the ARO tool

<table>
<thead>
<tr>
<th>FACH and DCH</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Wasted</strong> Radio Energy</td>
<td>34%</td>
</tr>
<tr>
<td><strong>Wasted</strong> Channel Occupation Time</td>
<td>33%</td>
</tr>
</tbody>
</table>
ARO: Mobile Application Resource Optimizer

• **Motivations:**
  – Are developers aware of the RRC state machine and its implications on radio resource / energy? **NO.**
  – Do they need a tool for automatically profiling their prototype applications? **YES.**
  – If we provide that visibility, would developers optimize their applications and reduce the network impact? **Hopefully YES.**

• **ARO: Mobile Application Resource Optimizer**
  – Provide visibility of radio resource and energy utilization.
  – Benchmark efficiencies of cellular radio resource and battery life for a specific application
ARO System Architecture

**Online Data Collection**
- Data Collector
- Handset & Carrier Type

**Offline Analysis**
- RRC Analyzer: Infer RRC states from packet traces
- TCP Analyzer: Associate each packet with its transport-layer functionality
- HTTP Analyzer: Associate packets with their application-layer semantics
- Burst Analyzer: Analyze triggering factors of traffic bursts with high resource overhead
- Profiling the App: Quantify resource impact of traffic bursts to reveal resource bottleneck
- Results Visualization: Visualize cross-layer analysis results
ARO System Architecture

Online Data Collection

Data Collector
Handset & Carrier Type

RRC Analyzer
Infer RRC states from packet traces

TCP Analyzer
Associate each packet with its transport-layer functionality

HTTP Analyzer
Associate packets with their application-layer semantics

Burst Analyzer
Analyze triggering factors of traffic bursts with high resource overhead

Profiling the App
Quantify resource impact of traffic bursts to reveal resource bottleneck

Results Visualization
Visualize cross-layer analysis results

Offline Analysis
The Data Collector

• Collects three pieces of information
  – The packet trace
  – User input (e.g., touching the screen)
  – Packet-process correspondence
    • The RRC state transition is triggered by the aggregated traffic of all concurrent applications
    • But we are only interested in our target application.

• Less than 15% runtime overhead when the throughput is as high as 600kbps
ARO System Architecture

Online Data Collection
- Data Collector
- Handset & Carrier Type

Infer RRC states from packet traces

Offline Analysis
- RRC Analyzer
  - Associate each packet with its transport-layer functionality
- TCP Analyzer
  - Associate packets with their application-layer semantics
- HTTP Analyzer
- Burst Analyzer
  - Analyze triggering factors of traffic bursts with high resource overhead
  - Quantify resource impact of traffic bursts to reveal resource bottleneck
- Profiling the App
- Results Visualization
  - Visualize cross-layer analysis results
RRC Analyzer: State Inference

- RRC state inference
  - Taking the packet trace as input, simulate the RRC state machine to infer the RRC states
  - Evaluated by measuring the device power

Example: Web Browsing Traffic on HTC TyTn II Smartphone
RRC Analyzer: Applying the Energy Model

• Apply the energy model
  – Associate each state with a constant power value
  – Based on our measurement using a power-meter

<table>
<thead>
<tr>
<th></th>
<th>TyTn Carrier 1</th>
<th>NexusOne Carrier 1</th>
<th>ADPI * T-Mobile</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(\text{IDLE})$</td>
<td>0</td>
<td>0</td>
<td>10mW</td>
</tr>
<tr>
<td>$P(\text{FACH})$</td>
<td>460mW</td>
<td>450mW</td>
<td>401mW</td>
</tr>
<tr>
<td>$P(\text{DCH})$</td>
<td>800mW</td>
<td>600mW</td>
<td>570mW</td>
</tr>
<tr>
<td>$P(\text{FACH} \rightarrow \text{DCH})$</td>
<td>700mW</td>
<td>550mW</td>
<td>N/A</td>
</tr>
<tr>
<td>$P(\text{IDLE} \rightarrow \text{DCH})$</td>
<td>550mW</td>
<td>530mW</td>
<td>N/A</td>
</tr>
</tbody>
</table>

* Reported by [27] for Android HTC Dream phone
ARO System Architecture

**Online Data Collection**
- Data Collector
- Handset & Carrier Type

**Offline Analysis**
- RRC Analyzer: Infer RRC states from packet traces
- TCP Analyzer: Associate each packet with its transport-layer functionality
- HTTP Analyzer: Associate packets with their application-layer semantics
- Burst Analyzer: Analyze triggering factors of traffic bursts with high resource overhead
- Profiling the App: Quantify resource impact of traffic bursts to reveal resource bottleneck
- Results Visualization: Visualize cross-layer analysis results
TCP / HTTP Analysis

• TCP Analysis
  – Infer transport-layer properties for each TCP packet
    • SYN, FIN, or RESET?
    • Related to loss? (e.g., duplicated ACK / recovery ACK)
    • ...

• HTTP Analysis:
  – HTTP is the dominant app-layer protocol for mobile apps.
  – Model HTTP behaviors
ARO System Architecture

Online Data Collection

- Data Collector
- Handset & Carrier Type

Offline Analysis

1. RRC Analyzer
   - Infer RRC states from packet traces
2. TCP Analyzer
   - Associate each packet with its transport-layer functionality
3. HTTP Analyzer
   - Associate packets with their application-layer semantics
4. Burst Analyzer
   - Analyze triggering factors of traffic bursts with high resource overhead
5. Profiling the App
   - Quantify resource impact of traffic bursts to reveal resource bottleneck
6. Results Visualization
   - Visualize cross-layer analysis results
Burst Analysis

• A burst consists of consecutive packets transferred in a batch (i.e., their IAT is less than a threshold)
• We are interested in short bursts that incur energy / radio resource inefficiencies
• ARO finds the triggering factor of each short burst
  • Triggered by user interaction?
  • By server / network delay?
  • By application delay?
  • By TCP protocol?
ARO System Architecture

**Online Data Collection**
- Data Collector
- Handset & Carrier Type

**RRC Analyzer**
- Infer RRC states from packet traces

**TCP Analyzer**
- Associate each packet with its transport-layer functionality

**HTTP Analyzer**
- Associate packets with their application-layer semantics

**Burst Analyzer**
- Analyze triggering factors of traffic bursts with high resource overhead

**Profiling the App**
- Quantify resource impact of traffic bursts to reveal resource bottleneck

**Results Visualization**
- Visualize cross-layer analysis results
Profiling Applications

• From **RRC Analysis**
  – We know the *radio resource state* and the *radio power* at any given time

• From **Burst analysis**
  – We know the *triggering factor* of each burst
  – We know the *transport-layer* and *application-layer* behavior of each burst

• By “**profiling applications**”, we mean
  – Compute resource consumption of each *burst*
  – Therefore identify the root cause of resource inefficiency.
Metrics for Quantifying Resource Utilization Efficiency

- Handset radio energy consumption
- DCH occupation time
  - Quantifies radio resource utilization
- Total state promotion time (IDLE $\rightarrow$ DCH, FACH $\rightarrow$ DCH)
  - Quantifies signaling overhead
- Details of computing the three metrics (upperbound and lowerbound) in the paper
Case Studies

• Fully implemented for Android platform (7K LoC)
• Study 17 popular Android applications
  – All in the “TOP Free” Section of Android Market
  – Each has 250,000+ downloads as of Dec 2010
• ARO pinpoints resource inefficiency for many popular applications. For example,
  – **Pandora Streaming**
    High radio energy overhead (50%) of periodic measurements
  – **Fox News**
    High radio energy overhead (15%) due to users’ scrolling
  – **Google Search**
    High radio energy overhead (78%) due to real-time query suggestions
Case Study: Pandora Music

Problem: High resource overhead of periodic audience measurements (every 1 min)
Recommendation: Delay transfers and batch them with delay-sensitive transfers

Pandora profiling results (Trace len: 1.45 hours)

<table>
<thead>
<tr>
<th>Burst type</th>
<th>Payloads</th>
<th>Energy</th>
<th>DCH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>LB</td>
<td>UB</td>
</tr>
<tr>
<td>LARGE_BURST</td>
<td>96.4%</td>
<td>35.6%</td>
<td>35.9%</td>
</tr>
<tr>
<td>APP_PERIOD</td>
<td>0.2%</td>
<td>45.9%</td>
<td>46.7%</td>
</tr>
<tr>
<td>APP</td>
<td>3.2%</td>
<td>12.8%</td>
<td>13.4%</td>
</tr>
<tr>
<td>TCP_CONTROL</td>
<td>0.0%</td>
<td>1.2%</td>
<td>1.6%</td>
</tr>
<tr>
<td>TCP_LOSS_RECOVER</td>
<td>0.2%</td>
<td>0.2%</td>
<td>0.6%</td>
</tr>
<tr>
<td>NON_TARGET</td>
<td>0.0%</td>
<td>1.8%</td>
<td>1.8%</td>
</tr>
<tr>
<td>Total</td>
<td>23.6 MB</td>
<td>846 J</td>
<td>895 sec</td>
</tr>
</tbody>
</table>

Problem: High resource overhead of periodic audience measurements (every 1 min)
Recommendation: Delay transfers and batch them with delay-sensitive transfers
Case Study: Fox News

Fox News profiling results (Trace len: 10 mins)

<table>
<thead>
<tr>
<th>Burst type</th>
<th>Payloads</th>
<th>Energy LB</th>
<th>Energy UB</th>
<th>DCH LB</th>
<th>DCH UB</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER_INPUT(Click)</td>
<td>91.0%</td>
<td>56.7%</td>
<td>67.6%</td>
<td>60.2%</td>
<td>70.4%</td>
</tr>
<tr>
<td>USER_INPUT(Scroll)</td>
<td>5.9%</td>
<td>15.2%</td>
<td>17.9%</td>
<td>14.7%</td>
<td>16.7%</td>
</tr>
<tr>
<td>APP_PERIOD</td>
<td>1.5%</td>
<td>5.2%</td>
<td>7.5%</td>
<td>6.1%</td>
<td>7.4%</td>
</tr>
<tr>
<td>TCP_CONTROL</td>
<td>0.0%</td>
<td>0.7%</td>
<td>3.7%</td>
<td>0.0%</td>
<td>2.3%</td>
</tr>
<tr>
<td>TCPLOSS_RECOVER</td>
<td>1.5%</td>
<td>0.7%</td>
<td>2.5%</td>
<td>1.9%</td>
<td>3.2%</td>
</tr>
<tr>
<td>SVR_NET_DELAY</td>
<td>0.1%</td>
<td>0.4%</td>
<td>0.8%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>Total</td>
<td>1.0 MB</td>
<td>276 J</td>
<td>284 sec</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Problem:** Scattered bursts due to scrolling

**Recommendation:** Group transfers of small thumbnail images in one burst
Case Study: Fox News

Problem: Scattered bursts due to scrolling

Recommendation: Group transfers of small thumbnail images in one burst
# Case Study: BBC News

## BBC News profiling results

<table>
<thead>
<tr>
<th>Burst type</th>
<th>Payloads</th>
<th>Energy</th>
<th>DCH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>LB</td>
<td>UB</td>
</tr>
<tr>
<td>TCP_CONTROL</td>
<td>0%</td>
<td>11.3%</td>
<td>24.2%</td>
</tr>
<tr>
<td>USER_INPUT</td>
<td>98.7%</td>
<td>42.5%</td>
<td>73.1%</td>
</tr>
<tr>
<td>SVR_NET_DELAY</td>
<td>1%</td>
<td>0.0%</td>
<td>2.7%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>162 KB</td>
<td>145 J</td>
<td>120 sec</td>
</tr>
</tbody>
</table>

**Problem**: Scattered bursts of delayed FIN/RST packets

**Recommendation**: Close a connection immediately if possible, or within tail time.
Summary

- ARO helps developers design *cellular-friendly* smartphone applications by providing *visibility* of radio resource and energy utilization.
Research Impact

• We contacted developers of top apps (e.g., Pandora). The feedback has been encouragingly positive.

AT&T's analysis of the Pandora application gave us a much better view of how Pandora interacts with low-level cellular network resources. Now that we better understand these interactions, we can optimize our application to make more efficient use of these resources. In fact, we'd like to incorporate AT&T's profiling tool as part of our normal ongoing testing.

Tom Conrad, CTO of PANDORA®


• Production version of ARO is being created.
Case Study: Google Search

Search three key words.
ARO computes energy consumption for three phases
- **I**: Input phase
- **S**: Search phase
- **T**: Tail Phase

**Problem**: High resource overhead of query suggestions and instant search

**Recommendation**: Balance between functionality and resource when battery is low
Case Study: Audio Streaming

**Problem:** Low DCH utilization due to constant-bitrate streaming

**Recommendation:** Buffer data and periodically stream data in one burst

### Constant bitrate vs. bursty streaming

<table>
<thead>
<tr>
<th>Name</th>
<th>Server</th>
<th>bitrate</th>
<th>Radio Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPR News</td>
<td>SHOUTcast</td>
<td>32 kbps</td>
<td>36 J/min</td>
</tr>
<tr>
<td>Tune-in</td>
<td>Icecast</td>
<td>119 kbps</td>
<td>36 J/min</td>
</tr>
<tr>
<td>Iheartradio</td>
<td>QTSS</td>
<td>32 kbps</td>
<td>36 J/min</td>
</tr>
<tr>
<td>Pandora w/ Ad</td>
<td>Apache</td>
<td>bursty</td>
<td>11.2 J/min</td>
</tr>
<tr>
<td>Pandora w/o Ad*</td>
<td>Apache</td>
<td>bursty</td>
<td>4.8 J/min</td>
</tr>
<tr>
<td>Slacker</td>
<td>Apache</td>
<td>bursty</td>
<td>10.9 J/min</td>
</tr>
</tbody>
</table>

* A hypothetical case where all periodic ads are removed.

---

**Problem:** Low DCH utilization due to constant-bitrate streaming

**Recommendation:** Buffer data and periodically stream data in one burst