PARCEL: Proxy Assisted Browsing in Cellular networks for Energy and Latency reduction

Ashiwan Sivakumar\textsuperscript{1}, Shankaranarayanan PN\textsuperscript{1}, Vijay Gopalakrishnan\textsuperscript{2}, Seungjoon Lee\textsuperscript{3*}, Sanjay Rao\textsuperscript{1} and Subhabrata Sen\textsuperscript{2}

\textsuperscript{1}Purdue University
\textsuperscript{2}AT&T Labs – Research
\textsuperscript{3}Two Sigma

* Work done when author was affiliated with AT&T Labs
Mobile Web Browsing in Cellular Networks

- E.g. Our measurements on a subset of Alexa top 500 pages – 6X longer

What are the drivers of poor performance?

Mobile web browsing technology: slow and power-hungry
Characteristics of Modern Webpages

- 100’s of objects from many domains
- Object sizes are small
- Static objects (e.g. images)
- Dynamic objects (e.g. JavaScript-generated) supporting rich interactivity

40% of Alexa top 500 pages : >= 100 objects
95% of obj. in Alexa top 500 pages : < 386 KB
Why Web Downloads Are Slow in Cell Networks?

1. Resolve DNS – foo.com

3. Parse HTML

4. Resolve DNS – bar.com

5. HTTP fetch bar.com/X.js, bar.com/Y.css

6. Process JS && CSS

7. HTTP fetch bar.jpg

Frequent short data transfers in high RTT cell link
- Many small objects from many domains
- HTTP request-response paradigm
- Object dependencies in webpages
Why Web Downloads Have High Radio Energy Usage?

- Cellular radio interface -> a growing component of the total device power
- Complex state machine for energy efficiency
- Different states with different power consuming modes
- Transition to IDLE -> typically >10sec inactivity (hard in web downloads)

**LTE Radio Resource Control State Machine**

- **RRC_CONNECTED**
  - Continuous Reception
  - Short DRX
  - Long DRX

- **RRC_IDLE**
  - DRX

**High radio energy usage caused by**

- ✓ Long page load times
- ✓ Frequent transitions inside high power RRC_CONNECTED
Contributions

- **PARCEL** – a proxy-assisted mobile web browsing architecture
- Key distinction from existing approaches – Judicious refactoring of web browsing functionality
- Benefits over traditional browsers
  - Significantly lower ‘Onload’ latencies
    - Onload: Browser triggers this event after receiving all objects to render an initial version of the page
  - Significantly lower radio energy usage
Outline

• Existing Solutions
• PARCEL Design
• Evaluation Methodology
• Results and Conclusion
Existing Cloud-heavy Thin-client Approaches

- All browsing functionality in the proxy [SkyFire, Opera Mini, Zhao et al ICDCS’11]
- All user interactions (e.g. mouse clicks) communicated to the proxy
  - JavaScript to handle the click executed only in the proxy
- User-interactions incur higher latency and radio energy consumption [Sivakumar et al HotMobile’14]
Other Related Approaches

• New application protocols (e.g. SPDY)
  • All browsing functionality in the client
  • Multiplexes multiple requests and responses unlike HTTP
  • Dependencies in web pages => SPDY poor performance in real world [Erman et al CoNEXT’13, Wang et al NSDI’14]

• Page transformation and compression (e.g. Pagespeed)
  • Compression by itself does not always result in latency and radio energy savings [Sivakumar et al HotMobile’14]

• Split-browsing architecture (e.g. Amazon Silk)
  • Black-box, proprietary
Outline

• Existing Solutions
• **PARCEL Design**
• Evaluation Methodology
• Results and Conclusion
Key Design Considerations of PARCEL

• PARCEL Design Considerations:

  ✓ Minimize per-object HTTP request-response
  ✓ Responsive and energy-efficient client interaction
  ✓ Cellular-friendly and latency-sensitive data transfer
PARCEL Design and Benefits

PARCEL Design and Benefits

1. GET http://www.foo.com
2. GET main html from www.foo.com
3. GET all associated objects (parses & executes)
4. Send all objects to browser
5. Executes like traditional browser (but doesn’t issue any requests)

Client execution – to handle user-interaction

Proxy execution - Only to identify objects to fetch
Latency-efficient Data Transfer Strategy (IND)

Transfer individual objects as they arrive from the server
Energy-efficient Bundled Data Transfer Strategy (X)

Multiple bundles of ‘X’ bytes each

Onload Event
What is the Right Bundle Size?

- Smaller bundle sizes -> latency-efficient
- Larger bundle sizes -> radio energy savings
- More generally depends on
  - Web page size (s)
  - Network bandwidth (B)
  - LTE radio power model parameter (α)
- Our analysis shows, optimal bundle size, \( b^* = \alpha \sqrt{sB} \) (Measured \( \alpha = 0.74 \))
  - E.g. For a 2MB webpage, with LTE speed of 6Mbps – optimal bundle size is 0.9 MB
Practical Issues and Solutions

• How to make the proxy aware of client **cache content and cookies**?
  • Proxy maintains per-client state
  • Tracks object versions to avoid redundant object transfer

• How to make the proxy respect **client-specific customization to pages**?
  • Browser sends attributes like UA, screen resolution and the proxy mimics the client

• How to handle **HTTPS** traffic at the proxy?
  • Personalized trusted proxy setting up independent secure connections
Prototype Implementation Details

- Developed the proxy as a Firefox add-on (uses the parser and JavaScript engine of Firefox)
  - 1.5K lines of JavaScript code

- Developed a custom parcel client application using android webview to render
  - 2K lines of Java code
Outline

- Existing Solutions
- PARCEL Design
- Evaluation Methodology
- Results and Conclusion
Evaluation Setup and Methodology

- In live LTE settings
- Workload: Subset of Alexa top 500 pages
- To minimize page variability
  - Replay recorded pages with WPR
- To minimize radio network variability
  - Multiple back-to-back runs (DIR and PARCEL) in the night
  - Discard runs with poor signal strength
  - Only consider runs with all LTE (discard 3G/LTE hand-off)
- Also evaluated with real web servers – for realism
Metrics Compared

• **Onload Time (OLT)**
  • Time to download all objects until Onload event measured from packet trace collected at the mobile client

• **Total Download Time (TLT)**
  • Time to download all objects beyond onload measured for the experiment duration

• **Total Radio Energy Usage**
  • Compute LTE radio power consumption using open source ARO tool [Qian et al MobiSys’11]
Outline

- Existing Solutions
- PARCEL Design
- Evaluation Methodology
- Results and Conclusion
Latency Benefits With PARCEL

- 70th percentile – 3X
- 50% average reduction
Understanding Latency Benefits

- Minimizing round trips -> reduced client latency
- Faster object identification and fetch

Sample Page Download

![Graph showing cumulative data size in MB over time from start in sec.]

- PARCEL Proxy Timeline
- PARCEL Client Timeline
- DIR Client Timeline

6 sec
Radio Energy Usage With PARCEL

- 60\textsuperscript{th} percentile = 2X
- Average 65% reduction

Reducing the gap between object transfers $\rightarrow$ radio energy savings
Comparing With Cloud-heavy Approaches
PARCEL Performs Well Under User Interaction

Local JS execution avoids unnecessary network communication
Summary of Other Evaluation Results

• **Bundling benefits**
  • All bundling strategies and baseline (IND) benefit over DIR
  • For large pages (>1MB) – bundling provides additional benefits and smaller bundle size (512 KB) better
    • E.g. With < 3% increase in OLT and > 20% radio energy savings
  • For small pages – all bundling strategies perform similar to baseline PARCEL (IND)

• **Real web servers**
  • Median onload time reduction of 3X
Conclusions

- **PARCEL** – optimizes mobile web download process using a proxy
- **Judicious browsing functionality refactoring**
  - Object identification and fetch at the proxy
  - Client executes JS locally to support interaction
- **Latency-efficient cellular-friendly data transfer schemes** for radio energy savings
- **Significant latency and energy reduction** in live LTE settings
Backup
## Table Comparing PARCEL to Existing Proxy-Based Approaches

<table>
<thead>
<tr>
<th></th>
<th>HTTP proxies</th>
<th>SPDY proxies</th>
<th>Cloud browsers</th>
<th>PARCEL</th>
</tr>
</thead>
<tbody>
<tr>
<td># of HTTP requests</td>
<td>Per object</td>
<td>Per object</td>
<td>Single</td>
<td>Single</td>
</tr>
<tr>
<td>Object identification and Fetch</td>
<td>Client</td>
<td>Client</td>
<td>Proxy</td>
<td>Proxy</td>
</tr>
<tr>
<td>Interactive JS</td>
<td>Client</td>
<td>Client</td>
<td>Proxy</td>
<td>Client</td>
</tr>
<tr>
<td>Cellular-friendly transfer</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Bundling Benefits Beyond Baseline

PARCEL

Smaller bundle sizes better (lower OLT increase and provide radio energy savings)