

# スマートフォンのWeb閲覧におけるパケット削減について

富田 大揮<sup>†</sup> 安井 浩之<sup>†</sup> 吉野 邦生<sup>†</sup>

東京都市大学<sup>‡</sup>

## 1. はじめに

近年、携帯電話などのモバイル端末が増え、日常的に持ち歩く人が増加している。その中でスマートフォンは日本でも普及期を迎え、販売台数を伸ばしている。スマートフォンはこれまでの携帯電話より Web へのアクセスが容易で、多くの情報を得ることができる。しかしスマートフォンが普及するに連れて、各携帯キャリアの回線が逼迫してきており問題になっている。

通常スマートフォン上で Web の閲覧をする際、リクエストしたサーバから直接端末側に Web ページ等が送信されるが、本研究では、端末側の画面に表示される部分だけを抽出した Web ページを生成する中継サーバを介すことで、パケットを削減する手法を提案する。これは Web を閲覧する際、必ずしもページ内の全ての情報が必要なわけではなく、画面に映らない部分が余分なデータになると考えたためである。

ユーザが画面表示外の情報を要求する際は、追加で表示外のデータを送信する遅延読み込みを行うことで、通常の閲覧と変わらぬ操作性を実現する。

## 2. 関連研究

### 2.1. Web Caching on Smartphones: Ideal vs. Reality [1]

モバイルネットワークにおける無駄な読み込みを防ぐためには Web のキャッシングが重要である。そこでキャッシングの実装を改善することで、ネットワークトラフィックの削減や携帯端末リソースの削減をもたらすことを実現している。

### 2.2. Opera Mini [2]

Opera社が開発したスマートフォン用ブラウザアプリケーションである。通常のブラウジングと異なり、Webのデータを一度中継サーバにおいて圧縮し、端末側で解凍を行うことで、トラフィックの削減を図っている。

### 2.3 Amazon Silk [3]

Amazon社が開発した Kindle Fire というタブレット端末専用のブラウザアプリケーションである。Opera Mini[2]と同様に、Webのブラウジ

ングをする際一度中継サーバを経由するが、圧縮するのではなく、ひとつのページを読み込む際の複数のセッションを一つにすることで読み込みの数を減らしている。また、サーバ上に Web のキャッシュを保存しておき、先読みすることで無駄な読み込みを防ぎトラフィックの削減を図っている。

## 3. 提案システム

### 3.1. 概要

提案システムは、Web ページ全体ではなく一部の情報が必要とされる場面を想定する。そのため端末側の画面に表示される部分だけを抽出した Web ページを生成する中継サーバを介すことで、パケットを削減するシステムである。ユーザが画面表示外の情報を要求する際は、追加で表示外のデータを送信するといった遅延読み込みを行う。

### 3.2. 構成

システムの構成図を図1に示す。



図1 システム構成

本システムは、Webアクセスを中継するプロキシサーバと、スマートフォン用ブラウザによって構成される。

プロキシサーバでは、ブラウザのレンダリングエンジンであるWebkit[4]を利用し、画面表示部分を特定し、Webページのソースを抽出する。また、画面表示部分以外のソースを別に保存する処理を行う。

スマートフォン側のブラウザでは、Ajaxを用いて、ユーザが操作を行った際、追加で画面表示部以外のデータを受け取るための処理を行う。

### 3.3. 画面非表示部分の特定手法

画面非表示部分の特定のために、Webkit の Render ツリーを利用する。Render ツリーから Web ページ上の各タグのブラウザ上に表示される座標及びサイズ情報を取得し、端末の画面サイズ内に座標が含まれてないタグを画面表示外

Packet reduction in web browsing for a smartphone using a proxy server

<sup>†</sup>Daiki Tomita, Hiroyuki Yasui, Kunio Yoshino

<sup>‡</sup>Tokyo City University

と判定する。

### 3.4. 画面表示部分を抽出したWebページの生成手法

画面表示外と判定されたタグを全て<div>に置き換え、置き換える前の各タグを保存する。

### 3.5. 差分データの送信手法

画面非表示部分の差分データの取得にはJavaScriptを用いる。スマートフォン側で画面外のデータを読みこませる際、画面端から遊びを持ってスクロールできるOverScroll機能をトリガーにしてサーバに対して非同期通信を行う。非同期通信によってサーバに保存されているタグを読み込み、タグの書き換えを行う関数を用いて<div>に置き換えられていたタグを元に戻す。

### 3.6. システム動作

システム動作の流れを図2に示す。

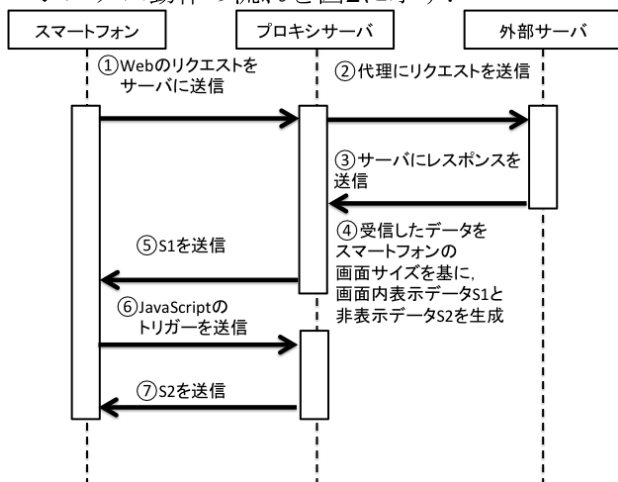


図2システム動作

- ① スマートフォンは、作られたアプリケーションから、プロキシサーバにWebのリクエストを送信する。
- ② プロキシサーバは、代理のリクエストを外部サーバに送信する。
- ③ 外部サーバは、レスポンスをプロキシサーバに送信する。
- ④ プロキシサーバは、WebkitのRenderツリーとスマートフォンの画面サイズを基に受信したデータから、画面表示内データS1と画面非表示データS2を生成する
- ⑤ プロキシサーバは、生成されたS1をスマートフォンに送信する。
- ⑥ スマートフォンは、JavaScriptを実行するためのトリガーを送信する。トリガーはスマートフォンの機能のOverScrollを使用。
- ⑦ プロキシサーバは、トリガーを受信しスマートフォンののにS2を送信する。

## 4. 検証実験

### 4.1. 実験手法

提案システムの検証実験は、評価サイトをブラウジングする際の packets 量を計測する。

評価の流れとして、提案システムは初回の読み込み時に画面内表示データを読み込み、追加で画面外のデータを読み込む。その都度 packets の計測を行い、すべてのロードが終わるまで計測を行う。比較システムはAndroid標準ブラウザで、評価サイトはJavaScriptやFlashを用いない一般的なサイトで行った。

### 4.2. 実験結果

実験は初回ロードから最後のロードまでの packets 量計測し、以下ようになった。

表1. 検証結果 (単位はByte)

ロード数	提案システム		標準ブラウザ
	総量	追加分	
初回	86310		150736
2	95433	9123	
3	123653	28220	
4	145369	21716	
最後	155565	10196	

最終的な packets 量は提案システムが上回る結果となった。これはタグの置き換え及びJavaScriptのトリガーの送信におけるオーバーヘッドが発生したためである。しかしながら、本研究ではページ内の全ての情報が必要になる状況を想定していないシステムであるため、初回時は半分近くに抑えられるなど成果があったといえる。また追加読み込みにかかる時間は、多少のラグをもたらしたが、操作感に影響が出る程ではないと感じた。

## 5. まとめ

実験結果より提案手法は packets 削減に有効であることが示された。今後はJavaScriptやAjaxへの対応を行うことでより汎用性のあるシステム実装を目指す。

## 参考文献

- [1] Feng Qian, Kee Shen Quah, Junxian Huang, Jeffrey Erman, Alexandre Gerber, Z. Morley Mao, Subhabrata Sen, Oliver Spatscheck “Web Caching on Smartphones: Ideal vs. Reality” MobiSys '12 Proc. of the 10th int. conference on Mobile systems, applications, and services pp127- 140(2012)
- [2] Opera Mini- <http://www.opera.com/mobile/>
- [3] Amazon Silk- <http://amazonsilk.wordpress.com/>
- [4] Webkit- <http://www.webkit.org/>