

放送型データベースシステムにおけるクライアントの消費電力を考慮した問合せ処理方式

北島 信哉^{†1} 原 隆浩^{†1} 寺田 努^{†2}
義久 智樹^{†1} 西尾 章治郎^{†1}

近年、サーバがPDAやスマートフォンなどのモバイル端末にデータベースの内容を定期的に放送する放送型データベースシステムが注目されている。放送型データベースシステムにおける問合せ処理手法としては、サーバが問合せ処理を行い、結果をクライアントに放送する方式、クライアントが問合せに関係するテーブル全体を蓄積して問合せ処理を行う方式、サーバとクライアントが協調して問合せ処理を行う方式の3方式が考えられる。これらの方式は電力消費に差があり、また、モバイル端末では利用できる電力に限りがある。そこで本論文では、モバイル端末を用いた実機評価により消費電力の特性を明らかにした後、モバイル端末の電力残量を考慮し、3方式の中から動的に処理方式を選択する手法を提案する。さらに、シミュレーション評価により、提案手法が従来手法と比べて、特に電力残量の少ない端末の生存時間を向上できることを示す。

A Query Processing Method Considering the Energy Consumption for Broadcast Database Systems

SHINYA KITAJIMA,^{†1} TAKAHIRO HARA,^{†1}
TSUTOMU TERADA,^{†2} TOMOKI YOSHIHISA^{†1}
and SHOJIRO NISHIO^{†1}

In recent years, there has been an increasing interest in a broadcast database system where the server periodically broadcasts contents of a database to mobile clients such as PDAs and smartphones. There are three query processing methods in the broadcast database system; (i) the server processes a query and then broadcasts the query result to the client, (ii) the client stores all data that are necessary in processing the query and then processes it locally, and (iii) the server and the client collaborate in processing the query. Generally, mobile clients have limits in power consumption, i.e., battery, and each of the three methods consumes different amount of power for query processing. In

this paper, we reveal the characteristics of power consumption by conducting a preliminary experiment on an implemented prototype, then propose a new query processing method which dynamically chooses a query processing method among the three query processing methods considering energy consumption. We also conduct a simulation experiment to verify that our method improves the lifetime of clients which have a low remaining battery.

1. はじめに

近年、無線通信技術の発展にともない、放送型通信を用いて情報を配信する放送型情報システムが注目されている。放送型情報システムでは、サーバはクライアントへの広い帯域幅を利用して各種のデータを周期的に放送し、クライアントは必要なデータのみを選択して取得する。放送型情報システムでは、クライアント数が増加してもデータ配信のコストがほとんど変わらないため、クライアント数が多い場合に通信品質を落とさず情報配信ができ、さらに、データアクセスのスループット向上が期待できる。

これまでに、放送型情報システムにおける応答時間の短縮を目的とし、放送データのスケジューリング戦略^{1),5),7)}、クライアント側のキャッシュ戦略¹⁾、プッシュ型とプル型の融合戦略^{2),6)}、放送を用いたプル型通信におけるアイテムのプリフェッチ戦略³⁾など多くの研究が行われている。これらの研究では、放送データを単なるデータアイテムとして扱っており、具体的な放送内容やデータ形式に基づいてシステムの効率化を行っているものは少ない。しかし、放送型情報システムでは、アプリケーションに依存してハイパーリンク形式やリレーショナルデータモデル形式など、様々なデータ形式が存在するため、放送するデータの内容や形式に適したデータ処理機構が性能向上の重要な要因となる。

そこで本論文では、サーバがリレーショナルデータベースの内容を繰り返し放送し、ユーザが放送されるデータベースに対して問合せを発行する環境を想定する。このようなシステムを放送型データベースシステムと呼ぶ。放送型データベースシステムにおける問合せ処理方式としては、サーバが問合せ処理を行い、結果をクライアントに放送するオンデマ

^{†1} 大阪大学大学院情報科学研究科マルチメディア工学専攻

Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University

^{†2} 神戸大学大学院工学研究科電気電子工学専攻

Department of Electrical and Electronic Engineering, Graduate School of Engineering, Kobe University

ンド型方式, クライアントが問合せに関係するテーブル全体を蓄積して問合せ処理を行うクライアント型方式, サーバとクライアントが協調して問合せ処理を行う協調型方式¹⁰⁾の3方式が考えられる. これらの方式は, 問合せ発生間隔や問合せ結果サイズなどの環境の変化に応じてその性能に優劣が生じるが, システム環境はつねに変化し続けるため静的に最適な方式を選択することは困難である.

これまでに文献 11) において, クライアントからの問合せがサーバに到着した時点で, 3方式のうち応答時間が最も短い方式を選択する問合せ処理手法が提案されている. さらに文献 12) において, 文献 11) における手法を拡張し, オンデマンド型方式を選択する際に, 問合せの発生頻度に基づいて余裕時間を設定する問合せ処理手法が提案されている. しかし, これらの手法では問合せ処理にかかる消費電力を考慮していないため, 利用できる電力が限られるモバイル環境では, 電力の少ない端末の生存時間が短くなってしまいう問題があった.

そこで本論文では, モバイル端末を用いた実機評価により消費電力の特性を明らかにし, モバイル端末の電力残量を考慮して問合せ処理方式を選択する手法を提案する. さらに, シミュレーション評価により, 提案手法が従来手法と比べて, 特に電力残量の少ない端末の生存時間を向上させることができることを示す. 本研究の新規性は, 電力残量を考慮して問合せ処理方式を選択する点にあり, 端末の生存時間を長くできるという有用性がある.

以下, 2章では放送型データベースシステムについて述べる. 3章で消費電力に関する実機評価について述べ, 4章で提案手法について説明する. 5章では提案方式の性能評価を行い, 6章で本研究について考察する. 最後に7章で本論文のまとめと今後の課題について述べる.

2. 放送型データベースシステム

本研究では, 図 1 に示すように, 放送型情報システムにおいてサーバがリレーショナルデータベースの内容を送信し, ユーザ(クライアント)が問合せを行う放送型データベースシステムを想定する. 放送型データベースシステムは, 以下に示す要素から構成される.

サーバ: サーバは, リレーショナルデータベースの内容を周期的に放送する. また, クライアントからの要求に応じて, 問合せ処理を実行する.

クライアント: 放送を受信するクライアントとしては, 記憶領域, 電力資源, 処理能力の乏しい携帯端末を想定する.

ダウンリンク: サーバからクライアントへの放送帯域は, 2つのチャンネルに分割されてい

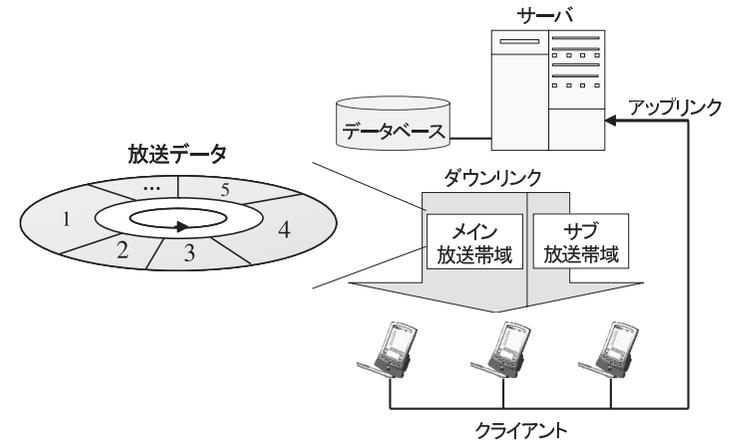


図 1 放送型データベースシステム
Fig. 1 A database broadcasting system.

るものとする. サーバは, 広帯域のメイン放送帯域を用いてデータベースの内容を繰り返し放送し, 狭帯域のサブ放送帯域を用いてそれ以外のデータを放送する.

アップリンク: クライアントからサーバへの狭帯域の通信チャンネルが存在する. クライアントは, このアップリンクを用いて問合せをサーバに送信する.

2.1 想定環境

本研究では, 街中で不特定多数のユーザに周辺情報を配信するようなアプリケーションを想定している. その一例として, 文献 11), 12) でも想定されているショッピングセンタにおける情報サービスがあげられる. このサービスでは, サーバがショッピングセンタ内の広告情報や店舗情報, また店舗で扱っている商品情報を含む最新のデータベースを放送し, ユーザはモバイル端末を持ち歩きながら放送される情報を受信して利用する. 通常, ユーザは端末の記憶容量の制限から放送されているデータすべてを蓄積するのではなく, 放送を受信しながら簡単なフィルタリングを行うことにより, 必要なデータのみを蓄積して閲覧する. たとえば, 漠然と「タイムサービスの情報を知りたい」という要求を持つユーザが, タイムサービスに関する情報のみを蓄積して閲覧するような状況があげられる. このようにつねに放送を受信するユーザの場合, 所持する端末の電力残量は十分にあるものと考えられる.

一方, 「商品 A の画像とその商品を扱っている店舗の地図が欲しい」といった複雑な問合せを行いたいユーザも存在する. このような複雑な問合せをフィルタリングにより実現しよ

うとすると多くのデータを1度受信する必要があるため、端末の記憶容量の制限から問合せを処理できない場合や、消費電力が大きくなるという問題が生じる。そこで、複雑な問合せを行いたいユーザは、サーバに対して問合せを発行する。問合せの応答時間は短いほど好ましいが、ユーザは発行した問合せの応答時間を知ることができないため、ユーザが自らで問合せを処理すべきか、サーバが問合せ処理を行った結果を待つべきかの判断はサーバが行う。本論文では、このような問合せを発行するユーザを対象とし、問合せ処理のための消費電力を低減させることを目的とする。また、電力残量の少ない端末を持つユーザの場合、放送されるデータすべてを受信することで電力を消費するより、必要な情報に対してのみ問合せを発行することで端末の生存時間を延ばす傾向があると考えられる。以上から、問合せを発行するユーザは、放送をフィルタリングしながら受信することはなく、問合せのみを行うものと想定する。

本研究では、問合せを発行するユーザ数百人に対し、放送をフィルタリングしながら受信しているユーザが2倍から5倍程度存在するものと想定している。そのため、放送を用いず、全ユーザの要求をクライアントサーバ型のシステムを用いて処理すると、問合せがサーバに集中し、通信帯域の制限から応答が極端に遅くなるという問題がある。そこで本研究では、サーバからクライアントへの帯域を2つに分割し、放送用の帯域を用いて、全ユーザに対して全データを周期的に放送するものと想定する。これにより、放送内容をフィルタリングすることで要求を満たせるユーザは問合せを発行せずに放送を受信し、複雑な要求を行いたいユーザのみが問合せを発行することが可能となる。その結果として、問合せを発行するユーザ数を減らすことで、より多くのユーザの要求を満たすことができる。

サーバが放送しているデータベースは、店舗の地図画像や商品画像を含み、画像の数はデータベース全体で数千枚、サイズは数百メガバイトとする。問合せを発行するユーザは数百の規模で存在し、各ユーザはつねに放送を受信できる環境にあり、パケットロスによる影響はないものと想定する。ユーザはショッピングをしながら欲しい商品を検索するため、待ち時間が1分程度となったとしてもユーザにとっての体感的な待ち時間はさほど長くはないと考え、問合せ処理にはリアルタイム性を要求しないものとする。ただし、ユーザは各問合せにデッドラインを設定でき、デッドラインの時間内にユーザが問合せの結果を得られない場合は、その問合せは失敗となる。ユーザは問合せをサーバに送信する際、設定したデッドラインや端末の記憶容量、電力残量、処理速度の情報を問合せに付加する。また、ユーザは問合せ結果が表示されるまでの間、液晶のバックライトを切っておき、結果を閲覧する際のみバックライトをつけることで消費電力を抑えることができるが、バックライトのオンオ

フは端末の省電力設定に依存するため、本論文では、液晶のバックライトは切っていないものと想定する。放送帯域は一般的な無線基地局を想定し、10Mbps程度とする。サーバは、つねに同じ放送スケジュールに基づいてデータベースの内容を繰り返し放送し、定期的に放送データのインデックスを放送する。サーバは、すべてのデータが1度ずつ放送される静的な放送スケジュールを作成する。

2.2 消費電力を考慮する必要性

ユーザは自身の端末をサービスに利用できる。また、ユーザは端末を他のサービス、たとえば、音声通話や地図ソフトと連携したナビゲーションなどにも利用することが考えられる。端末はつねに十分に充電されているわけではいたため、電力を使い果たしてしまう可能性がある。ユーザがふと訪れた店舗で安い商品を検索したいというような状況の場合、端末の電力残量が少ないと、必要な検索結果を得る前に端末の電力を使い果たし、検索結果を得られずにユーザが不便を強いられてしまう。そこで、特に電力残量が少ない端末の消費電力を削減することで、このような状況でもユーザ端末の利用可能時間を延長でき、ユーザの利便性を向上できる。

2.3 問合せ処理方式

放送型データベースシステムにおいて、クライアントによる問合せを処理する方式として、以下の3方式および文献11)、12)の提案方式がある。

2.3.1 オンデマンド型方式

クライアントがアップリンクを利用して問合せをサーバに送信し、サーバが問合せ処理を行った後でサブ放送帯域を用いて問合せ結果をクライアントに配信する。クライアントは、サブ放送帯域のみを受信すれば問合せ結果を得られる。

オンデマンド型方式では、問合せ処理のすべてをサーバが実行し、クライアントは放送される結果を受け取るだけでよい。クライアントは問合せを処理するための記憶領域を必要とせず、問合せ結果を得るために必要な電力も小さい。また、発生する問合せ数が少ない場合、問合せ結果が放送されるまでの待ち時間が短く、クライアントはすぐに結果を取得できる。しかし、問合せが頻繁に起こる場合や問合せの結果サイズが大きい場合にサブ放送帯域が枯渇するため、応答時間が長くなる可能性がある。

サブ放送帯域は放送型の通信を想定しているが、現在の想定では放送型通信である必然性はない。しかし、将来的に複数の問合せに対する結果をまとめて放送することで応答時間を短縮する、といった拡張を行うことを考えているため、放送型の通信を用いるものと想定している。

2.3.2 クライアント型方式

クライアントは問合せに関係するすべてのテーブルを自身の記憶領域にいったん蓄え、必要なすべてのデータが揃ってから、自ら問合せ処理を行う。

クライアント型方式では、クライアント上で問合せ処理が完結するため、クライアント数が増加しても、1 放送周期以内に問合せに関係する必要なすべてのデータを蓄積し、問合せ結果を得ることができる。しかし、端末の記憶容量による制約から問合せが処理できない場合がある、クライアントに大きな計算負荷がかかる、問合せ処理にかかる消費電力が大きいといった問題点がある。

2.3.3 協調型方式

クライアントは、アップリンクを利用して問合せをサーバに送信する。問合せを受け取ったサーバは、問合せを処理し、問合せ結果に含まれるタプルに処理用の識別子を付加するとともに、クライアントがデータを処理するためのルールを作成し、サブ放送帯域を用いてクライアントに送信する。クライアントは、自分宛に送信された処理ルールをもとに、問合せ結果の作成に必要なタプルの放送開始時刻と放送終了時刻を把握し、その時間にメイン放送帯域を用いて放送されるデータベースのうち、識別子を参照して必要なデータのみを蓄積し、問合せ結果を再現する¹⁰⁾。

協調型方式では、クライアントは識別子を参照することで、問合せ結果の作成に必要なデータのみを蓄積するため、クライアント型方式に比べてクライアントの記憶領域の使用量を小さくでき、消費電力も小さくなる。また、処理ルールは一般に非常に小さなサイズであることから、オンデマンド型方式に比べてサブ放送帯域の占有時間を短くできる。しかし、タプルに付加された識別子は、クライアントが問合せ結果を作成し終えるまで解放されない。識別子の最大数はあらかじめ決まっているため、問合せが頻繁に起こる場合には識別子が不足し、問合せの成功率が下がってしまう。

識別子数を増やすことは、放送するデータベースにおける識別子用の領域を増やすことに等しく、識別子数を増やすほど放送周期が長くなり、結果として応答時間が長くなるため、問合せの発行頻度に応じて最適な値に設定する必要がある。識別子数を変更すると放送周期が変化し、放送プログラムを変更しなければならないため、識別子数は動的に変更できず、固定するものと想定する。識別子は q_id , c_id の 2 つからなる。 q_id は問合せに対してサーバが一意に与えたクエリ ID を示し、クライアントは q_id を参照し、蓄積するタプルを決定する。 c_id は結合するタプルを示す組合せ ID を示し、クライアントは c_id が等しいタプルどうしを結合し、問合せ結果を得る。1 つの問合せ結果を作成するために複

数のタプルを組み合わせる必要がある場合が多いため、必要な c_id の領域は q_id の領域より多くなる。 q_id , c_id に書き込まれた値は、クライアントが該当する問合せの結果を得られるまで保持しておく必要があるため、応答時間が長い場合には、各領域が再利用できるまでに時間がかかる。そのため、各タプルに多くの q_id , c_id が書き込まれた状態が長く続くことがある。

2.4 LRT 方式

2.3 節で述べたオンデマンド型方式、クライアント型方式、協調型方式をそれぞれ単独で用いた場合、問合せ発生間隔や問合せ結果のサイズなどのシステム環境に応じて、その性能に優劣が生じる。そこで、システムの変化に応じて既存の 3 つの問合せ処理方式の中から最適な方式を選択できれば、システム全体の問合せ成功率、応答時間、消費電力の性能を向上させることができると考えられる。

LRT (Least Response Time) 方式¹¹⁾ では、クライアントからの問合せがサーバに到着すると、サーバはオンデマンド型方式、クライアント型方式、協調型方式の各方式を選択した場合の応答時間をそれぞれ計算し、応答時間が最も短い問合せ処理方式を選択する。応答時間が最も短い問合せ処理方式を選択してもデッドラインを越えてしまう場合には、問合せは失敗する。

また、サブ放送帯域の放送キューにおいて、協調型方式の処理ルールは、応答時間の短縮のため、キュー内のオンデマンド型方式の問合せ結果よりも前に挿入する。ただし、割り込んだ処理ルールにより、すでにキュー内にあるオンデマンド型方式の問合せがデッドラインを越えてしまう場合は、協調型方式を選択できない。

2.5 拡張 LRT 方式

LRT 方式では、協調型方式の処理ルールはキュー内のオンデマンド型方式の送信データの前に割り込むことになる。したがって、挿入位置以降の問合せの応答時間がわずかに長くなり、協調型方式の選択数が増加すると、この応答時間の増加が無視できなくなる。すでにキュー内にあるオンデマンド型方式の問合せがデッドラインを越えることは許さないため、このような状況では、協調型方式を選択できなくなってしまう。

拡張 LRT 方式¹²⁾ では、LRT 方式においてオンデマンド型方式の応答時間を計算する際、協調型方式の処理ルールを挿入するための余裕時間を設定しておき、デッドラインが余裕時間の分だけ短いものとして応答時間を計算する。これにより、オンデマンド型方式を選択された問合せは、キューに挿入された際、デッドラインまで必ず余裕時間以上の余裕が生じる。したがって、協調型方式の選択数が増加した場合でも、割り込みによる応答時間の増分を

表 1 日本 HP 社 iPAQ rx5965 スペック
Table 1 Spec of HP iPAQ rx5965.

項目	性能
OS	Windows Mobile 5.0
プロセッサ	Samsung SC32442 400 MHz
ROM メモリ	2 GB Flash ROM
RAM メモリ	64 MB SDRAM
無線 LAN 機能	IEEE 802.11 b/g
バッテリー	充電式 Li-Ion バッテリー (1700 mAh)

表 2 SHOP テーブル
Table 2 Shop table.

属性名	内容	型	備考
shop_id	店 ID	int	主キー
q_id	クエリ ID	char	
c_id	組合せ ID	char	
genre	商品ジャンル	int	
shop_name	店名	char	
map	店の地図	binary	

余裕時間で相殺できるため、協調型方式が選択できない状況が少なくなる。

3. 消費電力に関する実験

本章では、端末の消費電力について調査するために、実機実験を行う。本章の結果をもとに 4.3 節で消費電力の定式化を行い、5 章の評価で用いる。

3.1 実験環境

実験に用いた端末は、日本 HP 社の iPAQ rx5965 である。スペックを表 1 に示す。実装に用いた言語は C#、開発環境はマイクロソフト社の Visual Studio 2005 である。

消費電力の測定には特別な機器は用いず、オペレーティング・システム (OS) によって提供される電力残量表示を用いた。これは、電力残量を百分率 (%) で表示するものである。端末において特定の動作を一定時間継続して行い、動作時間に対する電力残量の減少率を測定することで、端末の消費電力の傾向を知ることができる。電力残量表示の正確性については、3.2 節で議論する。

無線 LAN によるデータ受信や ROM メモリへのデータ書き込みにかかる消費電力を比較するために、端末においてこれらの処理のみを繰り返し、一定時間、または一定周期ごとに電力残量を記録するプログラムを端末に実装した。また、端末を用いて放送を受信する際の消費電力を調べるために、放送サーバと受信用のアプリケーションを実装した。放送内容は 2.1 節で示したショッピングセンタにおける情報サービスをアプリケーション例として想定し、決定した。

表 2、表 3 に、実装した SHOP テーブル、GOODS テーブルの詳細を示す。店舗は販売品目によってジャンルごとに分類されており、各ジャンルの SHOP テーブル、GOODS テーブルをこの順に放送し、全ジャンル分のテーブルを放送し終わるまでを 1 放送周期と呼ぶ。

表 4 に、実装評価で用いるパラメータとその値を示す。クライアント型方式におけるタッ

表 3 GOODS テーブル
Table 3 Goods table.

属性名	内容	型	備考
goods_id	商品 ID	int	主キー
q_id	クエリ ID	char	
c_id	組合せ ID	char	(リレーション)
goods_name	商品名	char	
image	商品画像	binary	
price	商品価格	int	
shop_id	店 ID	int	(外部キー)

表 4 実機評価に用いるパラメータ
Table 4 Parameter configuration for implementation.

パラメータ名	値
ジャンル数 [個]	10
ジャンル内ショップ数 [個]	5
ショップ内商品数 [個]	200
1 タップルに付加可能な最大識別子数 [個]	200

ブル結合にかかる消費電力を調べるために、放送により受信した 1 ジャンル分の SHOP テーブルと GOODS テーブルの自然結合を繰り返すプログラムを実装した。ショップ数が 5、ショップ内商品数が 200 であるので、結果のタプル数は 1,000 となる。

3.2 予備実験

端末における OS による電力残量表示の正確性について調べるために、予備実験を行った。予備実験では、5 秒に 1 回、実験開始からの経過時間、電力残量、CPU 使用率をログとして書き出すプログラムを端末に実装し、端末を再起動した後に他の処理を行わない状態

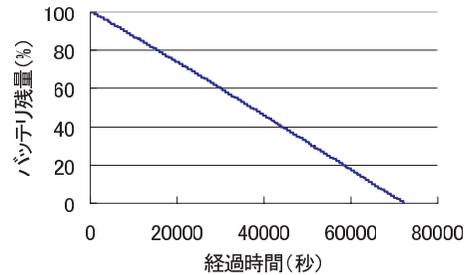


図 2 予備実験の結果

Fig. 2 Result of preliminary experiment.

(アイドル状態)において、このプログラムを端末が満充電の状態からバッテリーが切れるまで実行し続けた。このプログラム自体の負荷は非常に低く、プログラムを動作させずにアイドル状態のまま電力残量が 0 になるまでの時間を確認したところ、プログラムを動作させていた場合との差は 1%未満であったことから、このプログラムを動作させたことによる消費電力への影響は非常に低く、プログラムによる実験への影響は無視できると考えられる。

図 2 に、実験結果を示す。図から、経過時間に比例してほぼ均等に電力残量が減少していることが分かる。経過時間 t と電力残量 P の関係は 1 次関数で近似できるものと仮定し、最小二乗法による重回帰分析を行ったところ、次式ようになった。

$$P = -0.0014t + 100.44 \quad (1)$$

この近似曲線の信頼性を示す決定係数 R^2 は 0.9997 であり、十分に信頼性のある近似といえる。また、3.3 節において述べる他の動作にかかる消費電力についても、同様に 1 次関数で近似した後、 R^2 値を求め、値が 0.95 以上となることを確認した。

以上の結果から、電力残量は時間に比例して減少し、また、OS による電力残量表示の値は十分に信頼できるといえる。以降の実験では OS による電力残量表示を端末の電力残量として扱う。

3.3 実験結果

本節における実験では測定条件を統一するため、端末の省電力設定はすべてオフとした。また、測定に用いた端末では、主電源が入った状態で液晶表示を消すことはできず、液晶のバックライトをレベル 0~20 の 21 段階で調節できる。レベル 0 ではバックライトがオフになるため、表示による消費電力は最も低くなるが、通常の閲覧は難しい状態になる。バックライトをレベル 1 にすると、屋内で閲覧するには十分な明るさになる。実環境では問合せ

表 5 電力消費係数

Table 5 Power consumption coefficient.

処理内容	平均	分散
アイドル	13.2	0.2
無線 LAN オン	60.4	4.8
読み込み	36.2	0.2
書き込み	34.2	0.2
ダウンロード	81.6	0.3
放送データ受信	84.8	0.2
ダブル結合	36.8	0.7

結果が表示されるまでの間、液晶のバックライトを切っておき、結果を閲覧する際のみバックライトをつけることで消費電力を抑えることができるが、バックライトのオンオフは端末の省電力設定に依存するため、端末のバックライトはレベル 1 で固定した。測定時間は 1 回あたり 1 時間とし、測定には同一の端末を使用した。測定後は端末の電力残量が 80%以上になるように充電を行い、端末の再起動を行った後、次の測定に移った。測定後、3.2 節と同様に経過時間 t と電力残量 P の関係を 1 次関数で近似し、得られた 1 次関数の傾きの絶対値をその処理における電力消費係数とした。各処理について 5 回ずつ実験を行った。

結果を表 5 に示す。実験の際、データ書き込み時は一定サイズのファイルを追記モードで作成する、という動作を繰り返した。明示的な削除動作は含んでいないが、ファイルに追記する際に古いデータに上書きすることで、古いデータを削除している。また、比較的単純な動作であるデータ読み込み時と比べ、データ書き込み時の電力消費係数がやや小さくなっている。本実験で用いた端末では ROM に NAND 型のフラッシュメモリが用いられているが、近年のフラッシュメモリ技術の発展による低消費電力化¹⁴⁾ にともない、データ読み込み時とデータ書き込み時の消費電力の差がほとんどなくなってきている。NAND 型フラッシュメモリを用いた SSD (Solid State Drive) の中には、Corsair 製の CMFSSD-128GBG1D (読み込み時の消費電力 0.48 W, 書き込み時の消費電力 0.46 W) のように、書き込み時の消費電力より読み込み時の消費電力が大きいものもあり、この結果は妥当であると考えられる。

ここで、アイドル時の消費電力を端末の最低消費電力と考えると、他の動作にかかる消費電力は実験結果からアイドル時の消費電力を差し引いたものと見なせる。たとえば、ROM へのデータ書き込み時の電力消費係数は表から 34.2 であるが、アイドル時の消費電力係数が 13.2 であるので、ROM へのデータ書き込みのみの電力消費係数は差分の 21.0 と計算できる。以降、このようにアイドル時の消費電力との差分により求めた特定動作にかかる電力

表 6 要素電力消費係数
Table 6 Elemental power consumption coefficient.

処理内容	略号	要素電力消費係数	対アイドル比
アイドル	E_I	13.2	1.0
無線 LAN オン	E_L	47.2	3.6
読み込み	E_R	23.0	1.7
書き込み	E_W	21.0	1.6
ダウンロード	E_D	68.4	5.2
放送データ受信	E_B	71.6	5.4
タプル結合	E_C	23.6	1.8

消費のみを表す係数を、要素電力消費係数と呼ぶ。

表 6 に、表 5 から求めた要素電力消費係数を示す。アイドル時の値のみ表 5 の値をそのまま用いた。対アイドル比とは、各要素電力係数をアイドル時の要素電力消費係数で割った値である。このようにして求めた要素電力消費係数を用いることで、各処理にかかる消費電力を時間比例の一次式で定式化できる。また、各問合せ処理方式を用いて問合せを処理した際の消費電力を定式化し、提案手法で用いる。定式化については、4.3 節で詳しく述べる。

実験の際には、アイドル時に端末のバックライトをオフにした場合の測定も行い、バックライトの消費電力がアイドル時の 2 割未満であることを確認している。

4. ELEC 方式

2.5 節で述べた拡張 LRT 方式では、問合せの成功率の向上と応答時間の低減を目的とし、問合せ処理にかかる消費電力を考慮していないため、電力の少ない端末の生存時間が短くなってしまいう問題があった。そこで本章では、端末の電力残量を考慮して問合せ処理方式を選択し、電力残量の少ない端末の生存時間を向上させる手法を提案する。

4.1 概要

ELEC (Extended LRT considering Energy Consumption) 方式では、サーバは拡張 LRT 方式に従って問合せ処理方式を選択するが、問合せを発行した端末の電力残量が閾値 P_{TH} 以下ならば、電力残量が少ないと判断し、消費電力の少ない問合せ処理方式を優先的に選択する。ここで、すべての問合せに対し、消費電力が最小となる方式を選択しないのは、サブ放送帯域の混雑や識別子数の不足といった問題が生じるためである。新たに問合せを発行した端末の電力残量を P_{NEW} とすると、問合せ処理方式選択の流れは以下になる。

(1) $P_{NEW} \geq P_{TH}$ ならば、拡張 LRT 方式に従って問合せ処理方式を選択

(2) $P_{NEW} < P_{TH}$ ならば、消費電力が最小となる方式を選択

問合せの発生頻度に応じて、最適な閾値 P_{TH} は変化すると考えられる。そこで、サーバに問合せが q 個到着するごとに、閾値 P_{TH} を計算により求める。

4.2 閾値の計算方法

閾値 P_{TH} は、問合せを発行した端末のうち、電力残量が P_{TH} 以下の端末の消費電力が低くなるように決定する。そこで、 P_{TH} を計算する際には、 P_{TH} を変化させて過去 q 個の問合せをサーバ上でシミュレーションにより再現し、電力残量が P_{TH} 以下の端末が発行した問合せにおける 1 問合せあたりの消費電力が最小となる P_{TH} を、次にサーバに到着する q 個の問合せに対する閾値とする。ここで、 q をサンプル問合せ数と呼ぶ。具体的には、以下の手順を問合せが q 個到着するごとに行う。

(1) x 個前にサーバに到着した問合せ Q_x を発行した端末の電力残量を P_x とする。 q 個の要素からなる数列 $A = (P_1, P_2, \dots, P_q)$ に対し、これを昇順に並べ替えたものを閾値候補列 A' とする。

(2) A' の先頭から k 番目の要素を $A'[k]$ と表すこととする。 $A'[j]$ ($j = 1, 2, \dots, q$) に対し、次の手順を行う。

(2a) $P_{TH} = A'[j]$ とし、過去 q 個の問合せをサーバ上でシミュレーションにより再現する。ここで、 i ($i = 1, 2, \dots, q$) 番目に処理した問合せに必要な消費電力を $E_{j,i}$ とする。

(2b) $E_{j,i} \leq P_{TH}$ ($i = 1, 2, \dots, i$) となる最大の i を h_j とし、1 問合せあたりの消費電力 $V_j = \sum_{i=1}^{h_j} E_{j,i}/h_j$ を求める。

(3) V_j が最小となる j を \min_j とし、 $A'[\min_j]$ を次にサーバに到着する q 個の問合せに対する閾値とする。

手順 (2a) において過去 q 個の問合せをサーバ上でシミュレーションにより再現するために、サーバは過去 q 個の問合せの情報と、 q 個前の問合せがサーバに到着した際のサブ放送帯域の待ち行列の状態、使用している識別子の情報を保持しておく。

4.3 消費電力の定式化

消費電力が最小となる問合せ処理方式を決定するために、消費電力の定式化を行う。問合せ処理に必要な電力とは、問合せを発生した後に、必要なテーブルの受信やタプルの結合処理に必要な電力を指す。式中の比例係数は、表 6 における対アイドル比の値を用いた。

4.3.1 要素消費電力

まず、無線 LAN、読み込み・書き込み、タプル結合に必要な電力をそれぞれ定義する。

最低消費電力：端末の電源を入れただけの、何もしていない状態（アイドル状態）の消費電力を最低消費電力 E_I と呼び、以下のように定義する．

$$E_I(t) = t$$

無線 LAN：無線 LAN のスイッチをオンにすることで消費される電力 E_L は、以下のように定義する．

$$E_L(t) = 3.6t$$

読み込み・書き込み：読み込み・書き込みにかかる電力は、読み込みにかかる電力 E_R 、書き込みにかかる電力 E_W で定義する．

$$E_R(t) = 1.7t$$

$$E_W(t) = 1.6t$$

タプル結合：タプル結合にかかる電力 E_C は、以下のように定義する．

$$E_C(t) = 1.8t$$

4.3.2 動作にともなう消費電力

4.3.1 項で定義した要素消費電力をもとに、実際に放送受信、タプル結合処理、書き込み、読み込みの際に必要な電力を定義する．

放送受信：放送を受信しながら、受信したデータを書き込む際の消費電力 E_{rcv} を以下の式で定義する．

$$E_{rcv}(t) = E_I(t) + E_L(t) + E_W(t) = 6.2t \quad (2)$$

タプル結合処理：タプル結合処理にかかる電力 E_{tpr} は以下のように定義する．

$$E_{tpr}(t) = E_I(t) + E_C(t) = 2.8t \quad (3)$$

読み込み：データを処理する前に読み込む際にかかる電力 E_{rd} は以下の式で定義する．

$$E_{rd}(t) = E_I(t) + E_R(t) = 2.7t \quad (4)$$

書き込み：放送を受信しながらの書き込みに必要な電力は式 (2) と等しいとする．その他のデータ書き込みに必要な電力 E_{wrt} は以下のように定義する．

$$E_{wrt}(t) = E_I(t) + E_W(t) = 2.6t \quad (5)$$

4.3.3 問合せ処理に必要な電力

4.3.2 項で定義した消費電力をもとに、オンデマンド型方式、クライアント型方式、協調型方式の問合せ処理に必要な電力を定義する．

オンデマンド型方式：オンデマンド型方式では、問合せ結果をサブ放送帯域から受信するだけである．よって、オンデマンド型方式の問合せ処理にかかる電力 E_{on} は、以下の式で表せる．

$$E_{on}(t) = E_{rcv}(t) = 6.2t \quad (6)$$

クライアント型方式：クライアント型方式では、問合せ処理に必要なテーブルをメイン放送帯域から受信した後、問合せ処理を行う．よって、クライアント型方式の問合せ処理に必要な電力 E_{cl} は、必要テーブルの受信にかかる時間 t_{cl}^{rcv} 、必要テーブルの読み込みにかかる時間 t_{cl}^{rd} 、タプル結合処理にかかる時間 t_{cl}^{tpr} 、問合せ結果の書き込みにかかる時間 t_{cl}^{wrt} を用いて以下の式で表せる．

$$E_{cl}(t) = E_{rcv}(t_{cl}^{rcv}) + E_{rd}(t_{cl}^{rd}) + E_{tpr}(t_{cl}^{tpr}) + E_{wrt}(t_{cl}^{wrt}) \quad (7)$$

協調型方式：協調型方式では、処理ルールを受信した後、処理ルールをもとに必要なタプルのみを受信し、タプル結合処理を行う．よって、クライアント型方式の問合せ処理に必要な電力 E_{co} は、処理ルールの受信にかかる時間 $t_{co}^{rcvrule}$ 、処理ルールの読み込みにかかる時間 t_{co}^{rdrule} 、必要タプルの受信にかかる時間 t_{co}^{rcvtpi} 、必要タプルの読み込みにかかる時間 t_{co}^{rdtpi} 、タプル結合処理にかかる時間 t_{co}^{tpr} 、問合せ結果の書き込みにかかる時間 t_{co}^{wrt} を用いて以下の式で表せる．

$$E_{co} = E_{rcv}(t_{co}^{rcvrule}) + E_{rd}(t_{co}^{rdrule}) + E_{rcv}(t_{co}^{rcvtpi}) + E_{rd}(t_{co}^{rdtpi}) + E_{tpr}(t_{co}^{tpr}) + E_{wrt}(t_{co}^{wrt}) \quad (8)$$

5. 評価

本章では、次に示す 3 つの評価基準を用いて、ELEC 方式の有効性をシミュレーション実験の結果から検証する．

- 問合せ成功率
発生した全問合せのうち、クライアントが問合せ結果を受け取れたものの割合．
- 平均応答時間
問合せが成功した場合における、クライアントが問合せを発行してから、問合せ結果を得るまでの平均時間．ただし、クライアントが問合せをサーバへ送信するのにかかる時間、サーバにおけるデータ処理にかかる時間は、十分に小さいため無視する．
- 平均生存時間
クライアントが到着してから、予定滞在時間が経過した、もしくは端末の電力残量が 0 になったことにより退出するまでの平均時間．ただし、端末は問合せに関わる処理でのみ電力を消費するものとする．

5.1 評価モデル

本評価では、3.1 節と同様に、2.1 節で示したショッピングセンタにおける情報サービスをアプリケーション例として想定し、データベーススキーマと問合せモデルを決定した。

データベーススキーマは、表 2、表 3 で示した SHOP テーブル、GOODS テーブルを持つものとした。問合せは SQL によって記述されるものとする。簡単化のため、店舗テーブルと商品テーブルのタプルサイズは等しいものとする。また、ユーザは店舗テーブルと商品テーブルを自然結合する問合せのみを行うものとし、自然結合した結果のタプルには射影演算は行わないものとする。

シミュレーション評価では、各問合せに対しデッドラインをパラメータとして与え、設定したデッドラインの時間内にクライアントが問合せ結果を受け取れない場合は、問合せは失敗とする。また、端末の記憶容量をパラメータとして与え、問合せ処理の過程でクライアント側に記憶容量以上のデータを蓄積しようとした場合は問合せは失敗とする。

サーバはクライアントからの問合せに対し、3 方式のうちどの方式で処理するか、また、問合せが失敗であるかをクライアントに通知する選択方式通知メッセージをサブ放送帯域を用いて送信する。また、選択方式通知メッセージには必要なデータが放送される時刻が含まれており、メッセージ受信後、クライアントはその時刻まで端末の電源を落としておくことで消費電力を削減させる。選択方式通知メッセージのサイズは無視できるほど小さいものとし、つねにサブ放送帯域の放送キューの先頭に挿入されるものとする。

5.2 シミュレーション環境

表 7 に、評価で用いるパラメータとその値を示す。各パラメータは、2.1 節で示したショッピングセンタにおける情報サービスを想定して決定した。タプル利用率 r は、1 ジャンル分のテーブルの全タプルに対する、問合せ結果に含まれるタプルの割合を表す。また、テーブルサイズの違いにより問合せにかかる消費電力に差が出ると考えられるため、ジャンル内ショップ数は 5 から 8 までを 2 ジャンルずつ、9、10 を 1 ジャンルの合計 10 ジャンルとした。

クライアントは、指数分布に従った間隔でショッピングセンタに到着するものとした。各クライアントはサービスを受けられる端末を 1 台ずつ所持しており、各端末の初期電力量は 200 単位エネルギーから 2,000 単位エネルギーの一様分布、各端末の記憶容量は 1 MB から 100 MB の一様分布とした。また、各端末の CPU 速度は、200、300、400、500 MHz の一様分布とした。ショッピングセンタに到着したクライアントは、クエリ発生間隔に従って問合せを発行する。このとき、デッドラインや端末の記憶容量、電力残量、処理速度のデータ

表 7 評価に用いるパラメータ
Table 7 Parameter configuration.

パラメータ名	値
シミュレーション時間 [秒]	36,000
クライアントの予定滞在時間 [秒]	7,200
クライアントの到着間隔 [秒/人]	20
1 クライアントあたりのクエリ発生間隔 [秒]	300
応答時間のデッドライン [秒]	80
ジャンル数	10
ジャンル内ショップ数	5-10
ショップ内商品数	200
1 タプルのサイズ [KB]	10
1 タプルに付加可能な最大識別子数 [個]	200
メイン放送帯域 [Mbps]	10
サブ放送帯域 [Mbps]	2
処理ルールのサイズ [KB]	1
r の平均	0.03
r の標準偏差	0.01
余裕時間の計算に用いるサンプル数	5

を、問合せとともにサーバに送信する。各クライアントはショッピングセンタに到着したのち、予定滞在時間が経過するとショッピングセンタを出る。簡単化のため、端末の電力残量が 0 になったクライアントはサービスが受けられなくなるため、ショッピングセンタから退出するものとする。クライアントの到着間隔は 20 秒に 1 人、クライアントの予定滞在時間は 7,200 秒としているため、クライアント数は最大で 360 人ほどとなる。また、1 クライアントあたりのクエリ発行間隔を 300 秒としているため、問合せの発行間隔は平均で 0.83 秒に 1 回となる。これらの値は問合せを発行するクライアントのみの値であり、放送を受信しているのみの潜在的なクライアント数は 1,000 人から 1,500 人程度を想定している。協調型方式で用いる識別子の最大数は、問合せ成功率が最も高くなるように 200 とした。これは、協調型方式を用いて同時に処理できる問合せの最大数を表しており、200 個の問合せに対して協調型方式を割り当てた場合に十分な識別子用の領域を確保していることを意味する。たとえば、20 個の問合せに対して協調型方式を割り当て、そのうちクライアントがすでに結果を得た問合せが 5 個あるとすると、使用中の識別子は 15 個、利用できる識別子は残りの 185 個となる。また、3.3 節における実験と同様に、端末のバックライトは常時最小レベルで点灯しているものとする。

評価の際には、各問合せに対し、クライアント型方式における必要テーブルを、クライア

ントの問合せ発生時刻から1放送周期の範囲内でランダムに選択し、選択したテーブルの放送開始時刻と放送終了時刻をパラメータとして与えた。また、協調型方式における必要タプルを、クライアントの問合せ発生時刻から1放送周期の範囲内で問合せ結果サイズに応じてランダムに選択し、選択したタプルの放送開始時刻と放送終了時刻をパラメータとして与えた。式(7)における t_{cl}^{prc} をCPU速度が400MHzである日本HP社のiPAQ rx5965を用いて測定実験を行ったところ、1.8秒となったことから、 t_{cl}^{prc} は処理速度に反比例するものとし、以下の式により決定した。

$$t_{cl}^{prc} = 1.8 \times \frac{400}{CPU[MHz]} \quad (9)$$

また、式(8)における t_{co}^{prc} は、以下の式により決定した。

$$t_{co}^{prc} = t_{cl}^{prc} \times r \quad (10)$$

データの書き込み、読み込み速度は、日本HP社のiPAQ rx5965を用いた測定実験の結果から、それぞれ0.6MB/s、3.5MB/sとした。

5.3 比較手法

本研究では、比較手法として以下の手法と拡張LRT方式を用いる。

NB (No Broadcast) 方式: NB方式では放送を用いず、全帯域(12Mbps)を用いてオンデマンド型方式により問合せを処理する。ただし、他の放送を用いる方式では放送を受信しながら簡単なフィルタリングを行うことで必要なデータを得られるクライアントも、NB方式では問合せを発行するものとし、シミュレーション評価ではNB方式におけるクライアントの到着間隔は5秒/人(ただし、簡単なフィルタ処理を適用できない複雑な問合せを発行するクライアントの到着間隔は20秒/人)とした。

ランダム法: ランダム法では、3方式の中からランダムに方式を選択する。

最小電力法: 最小電力法では、3方式のうち、消費電力が最小となる方式を選択する。ELEC方式における閾値 P_{TH} を無限大とした場合に等しい。

5.4 シミュレーション結果

5.4.1 従来方式との比較

NB方式、ランダム法、最小電力法、拡張LRT方式、ELEC方式における問合せ成功率、平均応答時間を表8に示す。また、端末の初期電力量をもとに200単位エネルギーごとに分類した平均生存時間を図3に示す。ELEC方式におけるサンプル問合せ数 q は50とした。

図表から、NB方式では平均生存時間が最も長い。問合せ成功率と平均応答時間の性能は最も悪いことが分かる。NB方式では、つねにオンデマンド型方式で問合せを処理するた

表8 問合せ成功率と平均応答時間
Table 8 Success rate and response time.

手法	問合せ成功率	平均応答時間
NB方式	81.69%	72.15秒
ランダム法	85.27%	62.86秒
最小電力法	85.84%	63.85秒
拡張LRT方式	99.80%	45.76秒
ELEC	97.95%	52.00秒

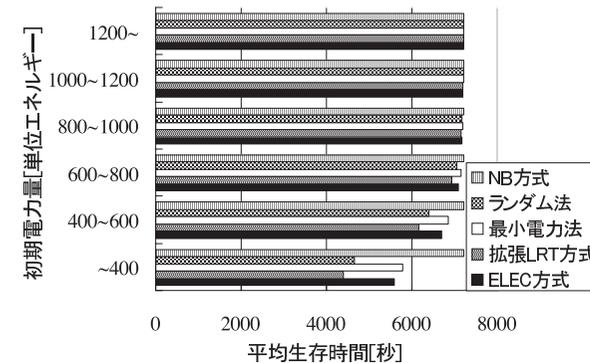


図3 平均生存時間
Fig. 3 Average lifetime.

め、平均生存時間は長くなる。しかし、放送を用いないため、他の方式と比べて処理する問合せ数が多くなり、問合せ成功率や平均応答時間の性能は悪くなる。NB方式では、サブ放送帯域における送信待ちデータはつねに120MBほどとなっており、すべて放送し終えるまでに約80秒かかる。これは問合せのデッドラインの時間である80秒と等しく、サブ放送帯域のみを用いるオンデマンド型方式では問合せが処理できないほどサブ放送帯域は混雑している。また、最小電力法では平均生存時間が長い。問合せ成功率と平均応答時間の性能は拡張LRT方式やELEC方式と比べ、非常に悪いことが分かる。最小電力法では、サーバはクライアントの消費電力のみを考慮しているため、平均生存時間は長くなる。しかし、サブ放送帯域や端末の記憶容量、協調型方式で用いる識別子数などのリソースに制限があるため、消費電力が最小となる問合せ処理方式(主にオンデマンド型方式)を選択し続けると、問合せ成功率や平均応答時間の性能が悪くなる。

ELEC方式における平均生存時間は、拡張LRT方式と比べると長く、問合せ成功率や平

表 9 問合せ成功率と平均応答時間への初期電力量の影響

Table 9 Impact of initial battery power on success rate and response time.

手法	問合せ成功率	平均応答時間
ランダム法	83.11%	63.38 秒
最小電力法	85.30%	63.80 秒
拡張 LRT 方式	99.70%	46.82 秒
ELEC	97.05%	52.91 秒

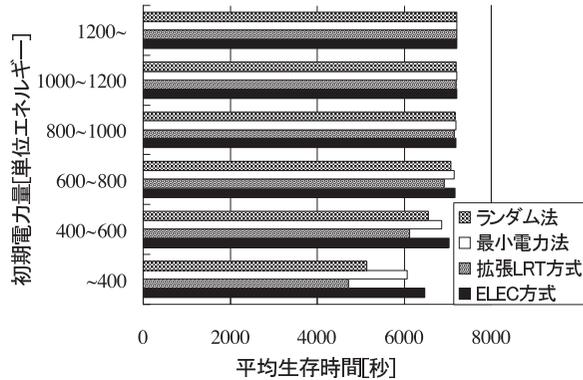


図 4 平均生存時間への初期電力量の影響

Fig. 4 Impact of initial battery power on average lifetime.

均応答時間の悪化がほとんどない。ELEC 方式では、電力残量の少ない端末が発行した問合せに優先的に消費電力が最小となる問合せ処理方式を割り当てることにより、初期電力量の少ない端末の平均生存時間が長くなる。

5.4.2 初期電力量の影響

初期電力量が平均生存時間に影響を及ぼすことが考えられるが、実際の初期電力量の分布がどのようになるか判断するのは難しいため、初期電力量の分布を変化させてシミュレーションを行う。端末の初期電力量を正規分布としたときの問合せ成功率、平均応答時間、閾値の平均を表 9 に示す。また、端末の初期電力量をもとに 200 単位エネルギーごとに分類した平均生存時間を図 4 に示す。

5.4.1 項と比べ、問合せ成功率、平均応答時間の性能はほとんど変化していないが、初期電力量が 400 以下の端末における生存時間が大きく向上している。これは、端末の初期電力量の分布が正規分布の場合には、電力残量の少ない端末数が 5.4.1 項の場合と比べて少な

表 10 問合せ成功率と平均応答時間への q の影響

Table 10 Impact of q on success rate and response time.

q	問合せ成功率	平均応答時間	閾値の平均
10	94.13%	57.05 秒	719 単位エネルギー
50	97.95%	52.00 秒	318 単位エネルギー
100	98.73%	50.33 秒	188 単位エネルギー

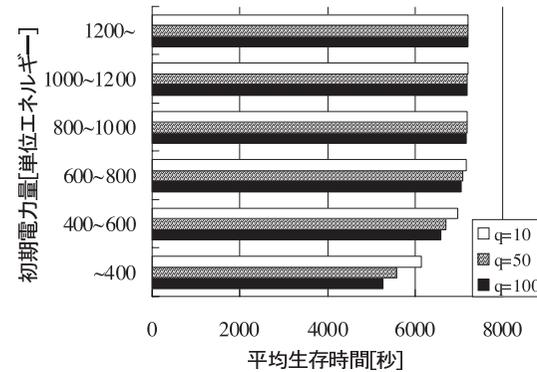


図 5 平均生存時間への q の影響

Fig. 5 Impact of q on average lifetime.

くなり、消費電力の少ない問合せ方式を選ぶ頻度が減るため、サブ放送帯域の混雑による影響が少ない。

また、最小電力法よりも ELEC 方式の平均生存時間が長くなっている。最小電力法ではすべての問合せに対して消費電力が最小となる方式を割り当てようとするため、サブ帯域などの制限から、電力残量が少ない端末に対して消費電力が比較的多い問合せ処理方式が割り当てられる可能性がある。初期電力量の分布が正規分布であるため、電力残量の少ない端末数が少なく、結果として電力残量が閾値以下の端末にのみ消費電力が少ない問合せ処理方式を割り当てる ELEC 方式の生存時間が長くなったと考えられる。

5.4.3 q の影響

ELEC 方式において、 q を 10, 50, 100 と変化させたときの問合せ成功率、平均応答時間を表 10 に示す。また、端末の初期電力量をもとに 200 単位エネルギーごとに分類した平均生存時間を図 5 に示す。

$q = 10$ の場合、 $q = 50$ の場合よりも初期電力量の少ない端末の平均生存時間は長い、問合せ成功率や平均応答時間の性能が悪い。 q の値が小さい場合、閾値候補 A' が少なく、閾値 P_{TH} が高くなりやすい。そのため、消費電力が最小の方式を選択することが多くなり、平均生存時間が長くなったと考えられる。一方、 $q = 100$ の場合、 $q = 50$ の場合と比べ、問合せ成功率や平均応答時間はあまり差がないが、初期電力量の少ない端末の平均生存時間が短くなっている。 q の値が大きい場合、閾値候補が多くなり、閾値が小さくなりやすい。そのため、消費電力が最小の方式を十分に選択できず、初期電力量の少ない端末の平均生存時間が短くなったと考えられる。

6. 考 察

6.1 関連研究

これまでに、放送型情報システムにおいて、モバイル端末の消費電力を削減することを目的とした研究が多く行われている。主な研究として、放送データにインデックスを付加することにより、クライアントが放送帯域を監視する時間を短縮し、消費電力を削減するというものがあげられる。インデックスには、放送されるデータ内容と放送時間が含まれており、クライアントはインデックスを受信することで、必要とするデータが放送される時間を知る。インデックスを受信したクライアントは、必要なデータが放送される時間まで電源を落としておくことで、消費電力を抑えられる。文献 8)、9) では、ツリー構造のインデックスや、インデックスの分散配置について論じている。また、文献 4) では、放送するデータ内容を考慮したインデックス配置手法を提案している。

本研究では、サーバが定期的に放送する放送データのインデックスや、問合せを発生した端末に送信する選択方式通知メッセージにより消費電力を削減しているが、効率的なインデックス構築手法については論じていない。しかし、従来研究の提案機構を本研究に適用することで、インデックスによる放送データへのオーバーヘッドを無視できると想定している。

また、文献 13) では、ラップトップ型のコンピュータを対象として端末の各構成要素における消費電力を調査している。本研究では携帯型端末である PDA を対象とし、データの送受信やタプル結合処理にかかる消費電力を OS による電力残量表示を用いて見積もること、実際のアプリケーションに近い環境を想定して消費電力を調査した点で異なる。

6.2 消費電力の定式化に関する妥当性

本論文では 4.3.3 項において、各問合せ処理方式を用いて問合せ処理を行った場合の消費電力を算出するための定式化を行っている。式 (6)、式 (7)、式 (8) の妥当性を調査するた

めに、3 章で行った実機実験と同様に、各問合せ処理方式を用いた問合せ処理のみを繰り返すプログラムを端末に実装し、実験を行った。

実験の結果から、各問合せ処理方式を用いて問合せ処理を行った場合の消費電力の値は、式 (6)、式 (7)、式 (8) を用いて計算した消費電力の値と比例しており、妥当性があることを確認している。

6.3 端末による差異の影響

本論文では消費電力モデルを構築するにあたり、日本 HP 社の iPAQ rx5965 を用いて実験した結果を用いている。端末の種類が異なる場合でも、消費電力の計算式における係数部分を変化させるだけで提案手法を適用できる。しかし、ユーザは自身の端末をサービスに利用できると想定しているため、システム内に消費電力モデルが異なる端末が混在する場合は考えられる。消費電力モデルの差異が小さい場合にはその影響は小さいため無視できるが、差異が大きい場合には電力残量が少ない端末に誤って消費電力の大きい問合せ処理方式を割り当ててしまい、端末の生存時間を短くしてしまう可能性がある。

実際のシステムにおいて消費電力モデルの差異を反映するために、端末の型番ごとの消費電力モデルをあらかじめサーバに登録しておき、ユーザがサーバに問合せを送信する際に端末の型番を送信することで、その端末の消費電力モデルを消費電力の計算に反映する方法が考えられる。

また、あらかじめ様々な端末の消費電力モデルを個別にサーバに登録しておくのは非現実的であるため、同一メーカーの端末や同年代に発売された構成が類似している端末については消費電力モデルが類似していると考え、同一の消費電力モデルを用いることで、登録する消費電力モデルを削減できる。

6.4 液晶のバックライトの影響

本論文では、ユーザはつねに端末のバックライトを使用しているものと想定し、シミュレーション評価を行った。しかし、実環境では、ユーザは問合せ結果が表示されるまでの間、液晶のバックライトを切っておき、結果を閲覧する際のみバックライトをつけることで消費電力を抑えることができる。

そこで、その影響を詳しく調べるために、問合せ結果を閲覧する時間は問合せを処理する時間と比べて十分に短いと無視できるものとし、問合せ結果を得るまでの間、バックライトをオフにしているものと想定して本論文と同様のシミュレーション評価を行ったところ、結果の傾向は本論文で示したものとほとんど差異がないことを確認した。

6.5 評価環境について

本論文におけるシミュレーション評価では、画像を含む比較的サイズの大きいデータが配信される環境を想定して評価を行った。実環境において配信されるデータのサイズが、将来的にどの程度になるのかを判断することは難しいが、本論文における提案手法はデータサイズの大小にかかわらず適用でき、有効である。

また、本論文ではサーバや放送設備が1つしか存在しないというシンプルな環境を想定して評価を行ったが、システムの構成が異なる場合であっても、本論文における消費電力モデルを用いた問合せ処理手法のアイデアを適用することで、初期電力量の少ない端末における生存時間を向上させることができると考える。

7. ま と め

本論文では、放送型データベースシステムにおいて、モバイル端末の電力消費を考慮して、動的に問合せ処理方式を選択する手法を提案した。提案手法では、閾値を設定することにより電力残量が少ない端末には優先的に消費電力の少ない問合せ処理方式を割り当てる。また、提案手法の有効性を検証するために、問合せ成功率と平均応答時間、生存時間についてシミュレーション評価を行った。シミュレーション評価の結果から、提案手法が従来手法と比べて、問合せ成功率や平均応答時間の性能を保ったまま、初期電力量の少ない端末の生存時間を向上させることができることを確認した。

今後は、問合せの類似性を考慮した問合せ処理方式について検討する予定である。また、パケットロスを考慮した問合せ処理方式について検討する必要がある。

謝辞 本研究の一部は、文部科学省グローバルCOEプログラム(研究拠点形成費)、および文部科学省科学技術振興調整費「先端融合領域イノベーション創出拠点の形成：ゆらぎプロジェクト」、および文部科学省特定領域研究(18049050)の研究助成によるものである。ここに記して謝意を表す。

参 考 文 献

- 1) Acharya, S., Alonso, R., Franklin, M. and Zdonik, S.: Broadcast Disks: Data Management for Asymmetric Communication Environments, *Proc. ACM SIGMOD 1995*, pp.199–210 (1995).
- 2) Acharya, S., Franklin, M. and Zdonik, S.: Balancing Push and Pull for Data Broadcast, *Proc. ACM SIGMOD 1997*, pp.183–194 (1997).
- 3) Aksoy, D., Franklin, M. and Zdonik, S.: Data Staging for On-Demand Broadcast,

Proc. VLDB Conference 2001, pp.571–580 (2001).

- 4) Chehaddeh, Y.C., Hurson, A.R. and Miller, L.L.: Energy-Efficient Indexing on a Broadcast Channel in a Mobile Database Access System, *Proc. International Conference on Information Technology: Coding and Computing (ITCC 2000)*, pp.368–374 (2000).
- 5) Foltz, K., Xu, L. and Bruck, J.: Coding and Scheduling for Efficient Loss-resilient Data Broadcasting, *Proc. International Symposium on Information Theory (ISIT 2003)*, pp.413–419 (2003).
- 6) 箱根 聡, 田辺雅則, 石川裕治, 井上 潮: 放送型通信とオンデマンド型通信を統合した情報提供システム, *情報処理学会論文誌*, Vol.40, No.10, pp.3772–3781 (1999).
- 7) Hu, Q., Lee, D. and Lee, W.: Performance Evaluation of a Wireless Hierarchical Data Dissemination System, *Proc. Mobicom 1999*, pp.163–173 (1999).
- 8) Imielinski, T., Viswanathan, S. and Badrinath, B.R.: Energy Efficient Indexing on Air, *Proc. ACM SIGMOD 1994*, pp.25–36 (1994).
- 9) Imielinski, T., Viswanathan, S. and Badrinath, B.R.: Data on Air: Organization and Access, *IEEE Trans. Knowledge and Data Engineering*, Vol.9, No.3, pp.353–372 (1997).
- 10) 加下雅一, 寺田 努, 原 隆浩, 塚本昌彦, 西尾章治郎: データベース放送システムのためのサーバと移動型クライアントによる協調型問合せ処理方式, *情報処理学会論文誌: データベース*, Vol.44, No.SIG8 (TOD18), pp.92–104 (2003).
- 11) 北島信哉, 寺田 努, 原 隆浩, 西尾章治郎: 放送型データベースシステムにおけるデッドラインを考慮した問合せ処理方式, *電子通信情報学会論文誌*, Vol.J89-D, No.2, pp.151–162 (2006).
- 12) 北島信哉, 寺田 努, 原 隆浩, 西尾章治郎: 放送型データベースシステムにおける問合せ発生頻度に基づいた問合せ処理方式, *情報処理学会研究会報告*, Vol.2006, No.9, pp.175–182 (2006).
- 13) Mahesri, A. and Vardhan, V.: Power Consumption Breakdown on Modern Laptop, *Proc. Workshop on Power-Aware Computing Systems (PACS 2004)* (2004).
- 14) 作井康司: Silicon Movie 時代に向けた大容量 NAND フラッシュメモリ技術, *FED ジャーナル*, Vol.11, No.3, pp.76–88 (2000).

(平成 20 年 11 月 13 日受付)

(平成 21 年 6 月 4 日採録)



北島 信哉 (学生会員)

2005年大阪大学工学部電子情報エネルギー工学科卒業。2006年同大学院情報科学研究科博士前期課程修了。現在、同大学院情報科学研究科博士後期課程在学中。放送型通信およびデータベースシステムに興味を持つ。日本データベース学会の学生会員。



原 隆浩 (正会員)

1995年大阪大学工学部情報システム工学科卒業。1997年同大学院工学研究科博士前期課程修了。同年同大学院工学研究科博士後期課程中退後、同大学院工学研究科情報システム工学専攻助手、2002年同大学院情報科学研究科マルチメディア工学専攻助手、2004年より同大学院情報科学研究科マルチメディア工学専攻准教授となり、現在に至る。工学博士。1996年本学会山下記念研究賞受賞。2000年電気通信普及財団テレコムシステム技術賞受賞。2003年本学会研究開発奨励賞受賞。ネットワーク環境上のデータ管理技術に関する研究に従事。IEEE, ACM, 電子情報通信学会, 日本データベース学会の各会員。



寺田 努 (正会員)

1997年大阪大学工学部情報システム工学科卒業。1999年同大学院工学研究科博士前期課程修了。2000年同大学院工学研究科博士後期課程退学。同年より大阪大学サイバーメディアセンター助手。2005年より同講師。2007年神戸大学大学院工学研究科准教授。現在に至る。2004年より特定非営利活動法人ウェアラブルコンピュータ研究開発機構理事、2004年には英国ランカスター大学客員研究員を兼務。博士(工学)。ウェアラブルコンピューティング、ユビキタスコンピューティングの研究に従事。IEEE, 電子情報通信学会, 日本データベース学会, ヒューマンインタフェース学会の各会員。



義久 智樹 (正会員)

2002年大阪大学工学部電子情報エネルギー工学科卒業。2003年同大学院情報科学研究科マルチメディア工学専攻博士前期課程修了し、2005年同専攻博士後期課程修了。博士(情報科学)。2005年京都大学学術情報メディアセンター助教。2008年大阪大学サイバーメディアセンター講師を経て2009年より同准教授となり、現在に至る。この間、カリフォルニア大学アーバイン校客員研究員。放送型データベースシステム, ストリーミング配信に興味を持つ。電子情報通信学会, IEEE 各会員。



西尾章治郎 (フェロー)

1975年京都大学工学部数理工学科卒業。1980年同大学院工学研究科博士後期課程修了。工学博士。京都大学工学部助手、大阪大学基礎工学部および情報処理教育センター助教、大阪大学大学院工学研究科情報システム工学専攻教授を経て、2002年より大阪大学大学院情報科学研究科マルチメディア工学専攻教授となり、現在に至る。2000年より大阪大学サイバーメディアセンター長、2003年より大阪大学大学院情報科学研究科長、その後2007年より大阪大学理事・副学長に就任。この間、カナダ・ウォータールー大学、ビクトリア大学客員。データベース, マルチメディアシステムの研究に従事。現在, Data & Knowledge Engineering 等の論文誌編集委員。本会理事を歴任。本会論文賞を受賞。電子情報通信学会フェローを含め, ACM, IEEE 等 8 学会の各会員。