

モバイル環境における暗号化コンテンツの配布及び管理システム

小口正人*1 齊藤忠夫*1*2 野原光夫*2

*1 中央大学

〒162-8473 東京都新宿区市谷本村町 42-8

*2 (株)トヨタ IT 開発センター

〒153-0061 東京都目黒区中目黒 2-1-23

oguchi@computer.org

概要

モバイル環境におけるウェブをベースにした情報提供サービスは今後需要が増していくと考えられるが、提供するデータの大容量化やサービスの高機能化が進むに連れ、常時接続が期待できない無線環境において、これをコンテンツプロバイダからユーザまでいかに配布するかが重要な課題となる。現状でウェブをベースにしたシステムは殆どの場合 Thin クライアント型のモデルであるが、クライアント側にデータの保持や処理機能を持たせた Thick クライアント型のモデルは、この問題の解決策として有効と考えられる。

本研究では、ウェブをベースにした Thick クライアント・サーバモデルをモバイル環境に導入し、情報提供サービスを実現するために必要な事柄の検討を行った。まず課金処理などを行うために必要となる暗号化したコンテンツやその暗号鍵の配布と管理のメカニズムを検討し、オフラインでコンテンツの仮決済を行う手法を提案した。また実験システムを構築し、暗号化コンテンツの提供サービスのパフォーマンスを評価した結果、コンテンツをメタコンテンツと分離して異なるルートでクライアントへ提供する方式の実効性が示された。

Encoded Data Distribution and Management System in Mobile Environment

Masato Oguchi*1

Tadao Saito*1*2

Mitsuo Nohara*2

*1Chuo University

*2TOYOTA Infotechnology Center Co., Ltd.

Abstract

Web-based information distribution services seem to become important in the mobile environment. However, as the volume of data increases and higher quality of service is demanded, it becomes a significant problem how to provide such services from content providers to users through wireless lines which might not always be connected. While a thin client model is a mainstream in current web-based systems, a thick client model seems to be effective to solve the problem, in which processes are distributed and executed at clients.

In this paper, a web-based thick client – server model is introduced to a mobile environment. First, a mechanism of data distribution and management system suitable for user's payment of contents is discussed, and a method of temporary payment in an offline state is proposed. An experimental platform is built, and the proposed mechanism is evaluated on it. According to results of evaluations, the proposed method, in which contents are divided into content data and metadata and transferred to clients through different routes, achieves some good performance.

1. はじめに

テレマティクスなどのモバイル環境において、ウェブをベースとした情報やアプリケーションの提供サービスを実現する場合、コンテンツの提供者からクライアント端末までいかにしてデータを配布するかということが重要な課題である。モバイル環境の接続形態として、有線インターネット環境に近年見られるような広帯域常時接続は期待できず、基本的に狭帯域のダイヤルアップ接続やホットスポットでの一時的な接続のみとなり、普段はオフラインである可能性が高い。

分散システムのクライアント・サーバモデルは、機能をサーバとクライアントでどのように分担するかという観点から、サーバで集中的に処理を行いクライアントの機能を最小とする Thin クライアントモデルと、処理をクライアントに分散して実行する Thick クライアントモデルに分類できる[1]。ウェブをベースとした情報提供サービスの場合、現状では Thin クライアントモデルをベースとしたものが主流である。しかしテレマティクスなどのモバイル環境においては、ネットワークへ接続している間に必要なデータ等をクライアントへダウンロードしておきオフラインでもクライアント上でサービスを提供できる Thick クライアントモデルが有効であると考えられる。そのようなモデルを採用し、ホットスポットなど様々なチャネルを用いてデータ配布を行えば、モバイル環境におけるサービスの利用機会は向上する[2]。

ただしこのようなサービスの提供をビジネスとして実現させるためには、データの配布をサーバ側で管理・制御するためにユーザのアカウント処理や課金処理などが必要となる。Thin クライアントモデルの場合このような処理はすべてサーバ側で行われるが、Thick クライアントモデルの場合、データの配布や利用の度にサーバが介入しなければならないとすると、Thick クライアントであることの効果が薄くなる。またウェブをベースとしたサービスのサーバ側アーキテクチャは、バックエンドのアプリケーションサーバやデータベースサーバを用いて動的なサービスを行う複雑な構成のものが近年増加しており、単にデータをクライアントにキャッシュするだけでは、オフラインの状態においてクライアント上でサービスを提供することが難しい[3][4][5][6]。

このような問題に対処する方法としては、データをコンテンツデータと、そのコンテ

ンツに関する情報やユーザの利用権などの情報からなるメタデータに分離して扱う方が有効であると考えられる。例えばコンテンツデータは暗号化して自由に配布し、これを復号するコンテンツ鍵をメタデータとして管理の元にユーザに提供する。メタデータはコンテンツデータに比べるとボリュームが小さく、サーバやクライアントのデータベース内でセキュアに取り扱うのに向いているため、Thick クライアントモデルでメタデータを処理する方法を確立すれば、モバイル環境において効率の良いデータ配信サービスを実現できると考えられる。本研究では、テレマティクスなどのモバイル環境を前提とし、コンテンツデータとメタデータを Thick クライアントモデルで扱うことにより、ユーザにコンテンツ配布・管理サービスを提供するためのシステムを検討する。

2. テレマティクス環境に適したコンテンツ配布・管理システム

2.1 提案システムの概要

本研究における提案システムの全体構造を図1に示す。システムの目的はコンテンツに関するメタデータの管理を行いながら、コンテンツをコンテンツプロバイダよりクライアントへ提供することである。コンテンツデータは暗号化されていて自由にコピーすることが可能であり、コンテンツプロバイダから直接、或はプロキシサイトや他のクライアントを通じて間接的に配布される。コンテンツデータを復号するコンテンツ鍵やユーザの利用権などを示したメタデータは、コンテンツデータとは別にセキュアな通信路を通じて専用のモバイル端末であるクライアントのデータベースに格納され、ユーザがこのデータを直接読み出すことはできない。

クライアントにはコンテンツデータとこれを利用するアプリケーションが収容され、更にコンテンツ鍵やその利用権などを示したメタデータを格納するデータベースが装備されている。ユーザはコンテンツコントロールモジュールのインタフェースを通してのみ、アプリケーションを実行したり復号化されたコンテンツデータへのアクセスを行ったりすることができる。

コンテンツを獲得して利用するクライアントは、決済サイトに接続して決済処理を行う。この具体的な手続きに関しては次項で述べる。基本的にはユーザがコンテンツ鍵の利用権を購入し、決済サイトにおいて

電子決済などにより支払いを行い、この金額がコンテンツプロバイダに送金された後クライアントへ決済完了情報が返され、コンテンツが利用可能となる。一連の決済手続きは、コンテンツプロバイダ側だけでなく、クライアントのデータベースにも記録される。

プロバイダへ送金する。決済サイトはクライアントのデータベースに決済完了情報を送信する。決済を行った後は、そのコンテンツはクライアント内で永続的に利用可能となる。

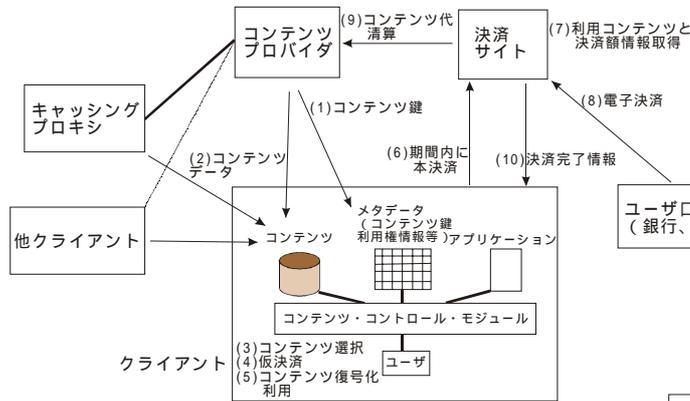


図 1. 提案システムの概要

2.2 コンテンツの獲得と決済手続き

クライアントと決済サイトの動作フローチャートを図2と図3に示す。また以下で述べる仮決済を用いたオフライン決済の動作は図1にも記した。

コンテンツプロバイダはクライアントに対し、まずコンテンツ情報を提供する。ユーザが興味を持つコンテンツであれば、クライアントはさらにコンテンツのメタデータを獲得する。この両者は併せたものでも構わない。次に暗号化されたコンテンツデータを、コンテンツプロバイダから直接的或は間接的に入手する。ここまでの処理により、メタデータはクライアントのデータベース内に保持され、またコンテンツデータは暗号化された状態で保持されている。

次に決済手続きを行うことにより、そのクライアントにおいて、メタデータを利用してコンテンツデータが利用可能になる。決済手続きには以下のようにオンライン決済とオフライン決済が可能である。

・オンライン決済

クライアントは決済サイトに接続した状態で、コンテンツ鍵の利用権を購入して決済を行う。決済サイトでは、ユーザの利用コンテンツ情報や決済額情報を読み出す。電子決済を行う場合には、ユーザの口座から決済額を引き落とし、現金決済の場合には現金を受け取って、決済額をコンテンツ

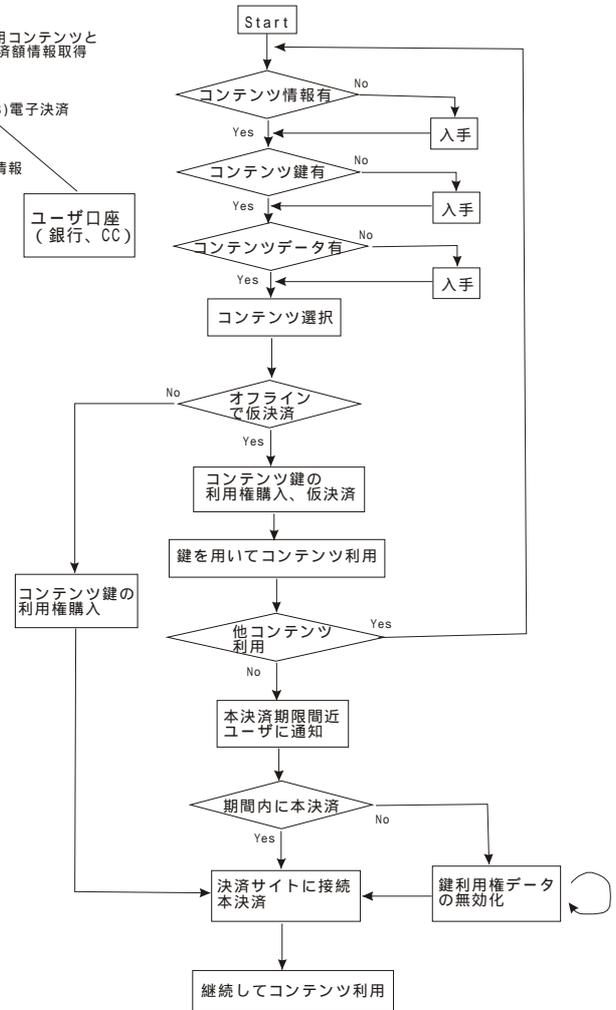


図 2. クライアント動作フローチャート

・オフライン決済

決済サイトに接続した状態でのみしか決済が実行できない場合には、オフライン状態での利用が多いモバイル環境においては利用機会が制限される。そこでオフライン状態においてもコンテンツの利用権の獲得とその仮決済を実行できるようにし、その後ネットワーク接続時に決済サイトにおいて本決済が可能なメカニズムを提案する。これは以下のように動作する。

まずユーザはオフラインの状態で購入するコンテンツを選択し、コンテンツ鍵の利用

用権を購入して仮決済を行う。この情報はクライアントのデータベース上に記録される。クライアントは購入したコンテンツ鍵を利用することができるようになり、コンテンツデータの復号化とコンテンツを利用したアプリケーションの実行が可能になる。クライアントは一定期間内に決済サイトに接続して本決済を行う。本決済が完了したクライアントでは、そのコンテンツを継続して利用できるようになる。もし期間内に本決済が行われない場合には、クライアントのデータベース内のデータが無効化され、本決済が行われるまでシステムが利用不能になる。

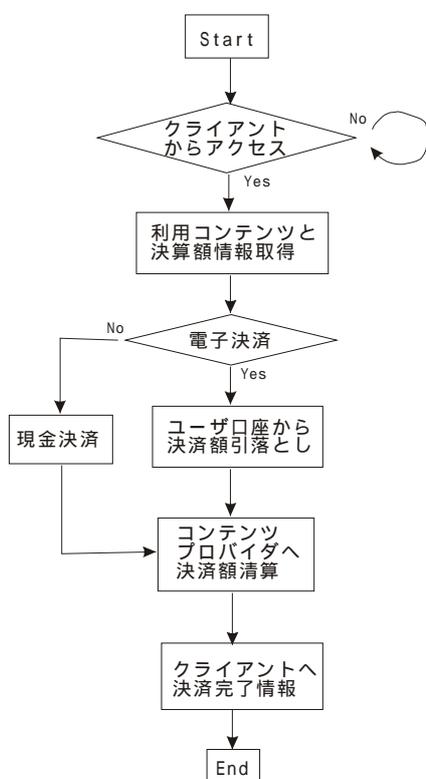


図 3. 決済サイト動作フローチャート

3. クライアントアーキテクチャ

3.1 クライアントモデル概要

分散システムの Thin クライアントモデルと Thick クライアントモデルを比較すると、現状のウェブをベースとしたサービスにおいては Thin クライアントモデルが注目されることが多く、従ってサーバサイド・プログラミングが盛んに研究されている。Thin クライアントのメリットの一つは、

クライアント端末をシンプルにかつコストを低く抑えられることで、このことが現状では有線環境だけでなくモバイル環境においても Thin クライアントモデルが主流である要因となっている。

しかしテレマティクスなどのモバイル環境においては、クライアントにも大量のデータを蓄積・処理する能力のある端末を用いることが可能になってきており[7]、またネットワークに関しては今後無線 LAN や DSRC など広いバンド幅の通信が行える場所、いわゆるホットスポットが部分的に存在する環境が増えてくるものと考えられる。そのような状況においては、データを使用するたびにサーバにアクセスしなければならない Thin クライアントモデルより、ホットスポットで必要なデータを持ってきておき、その後はオフラインや細い回線のみで接続された状態になっても要求を満たすサービスを続けられる Thick クライアントモデルの方が好ましいケースも多い。またクライアント・サーバ間で負荷分散や機能分散が行える Thick クライアントモデルの方が、スケーラビリティが高くなるというメリットも存在する。

しかし現状のモバイル環境では Thin クライアントモデルをベースにしたシステムが主流であり、今後も発展が予想されることから、Thick クライアントモデルのシステム開発を Thin クライアントモデルと共通の枠組みで考えるべきである。現在及び将来的なサービスの基盤を考える場合に、独自のプロトコルや閉じたネットワークでサービスを実現しようとするのは現実的ではなく、現状のウェブをベースにしたサービスを拡張し、Thin クライアントベースのシステムと互換性を取りながら新しいシステムを検討していくべきであると考えられる。

そこで本研究では HTTP (以下において HTTPS も含む) プロトコルで通信を行う現状の Thin クライアントモデルの枠組みに Thick クライアントを導入するシステム構成を検討した。これにより、アプリケーションプログラマが両者の違いを出来るだけ意識せずにシステムを構築できるシームレスな環境が実現する。具体的な提案方式を次節で述べる。

3.2 Thick クライアントモデルとその枠組みの提案

Thin クライアント・サーバモデルの枠組みへ Thick クライアントを導入する具体的

な方法として、以下のようなモデルを提案した。これを図4に示す。

まずサーバ側の構成であるが、現状のWWWにおいて用いられている最も先進的なサーバの形態は、ウェブサーバ、アプリケーションサーバ、データベースサーバの3ティアの構造を持つものである。その基本的な動作としては、ウェブサーバにおいてクライアントからのリクエストを受け付け、リクエストに対応する処理をアプリケーションサーバ上のプログラムで実行し、必要なデータはデータベースサーバへアクセスして処理を行うという形になっている。アプリケーションサーバ上のプログラムとして従来はCGIなどの形式がよく用いられていたが、近年はJavaをベースにしたサーブレットやJSPなどの形式が広まりつつある[8][9]。このような3ティアのサーバ構造はWebをベースにした高度な情報提供サービスに適した形態であり、今後もこのモデルをベースにシステムが発展していくものと予想される。

Thinクライアント・サーバ型のモデルにおいては、このような構造のサーバに対し、クライアントは基本的にブラウザのみでアクセスを行う。この場合アプリケーションの具体的な実行はアプリケーションサーバ上のみで行われ、クライアントは結果を受け取って表示するだけである。データの保持もクライアントでは行われず、基本的にすべてのデータはサーバ内のデータベースに格納される。クライアントとウェブサーバとの通信プロトコルにはHTTPが用いられる。

このようなThinクライアント・サーバ型の環境に対し、Thickクライアントを導入する実現手法を以下のように検討した。

Thickクライアントには、オフラインであってもネットワーク接続時と同じようなサービスを提供できる能力のあることが望ましい。従ってクライアント内にローカルウェブサーバ、ローカルアプリケーションサーバ、ローカルデータベースサーバを導入した。ユーザはThinクライアントの場合と同じように、ブラウザを用いてローカルウェブサーバにアクセスしサービスを受けることができる。ローカルデータベース内に必要なデータが存在すれば、ローカルアプリケーションサーバ上で実行されたプログラムの結果をブラウザがローカルウェブサーバを介して受け取り、オフラインであってもあたかもサーバ側にアクセスしているかのようなサービスを提供できる。

このモデルを実現するためには、ローカ

ルにデータがあった場合にはローカルのみで処理を実行し、データが無かった場合にはサーバ側へデータのリクエストを出し、受け取った後に引き続きローカルで処理を行ってユーザに結果を返すような仕組みが必要である。この仕組みをクライアント内においてローカルアプリケーションサーバ上のミドルウェアという形で実現した。ミドルウェアはローカルウェブサーバからユーザのリクエストを受け取り、ローカルデータベースをチェックしてデータが存在するか確認し、無い場合にはサーバ側へリクエストを出しデータを受け取ってこれをローカルデータベースへ格納し、その後ローカルアプリケーションサーバ上でアプリケーションを実行させて結果をユーザへ返すという働きを行う。サーバ側との通信は、Thinクライアントのブラウザと同じようにサーバ側のウェブサーバとの間でHTTPのコネクションを張り、アプリケーションサーバ上のプログラムとやり取りを行って処理を進める。ローカルウェブサーバへのアクセスもHTTP上で行われており、Thinクライアントモデルの枠組みに馴染むものとなっている。

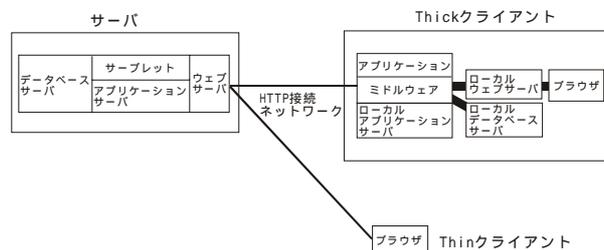


図4. 提案したThickクライアントモデルとその枠組み

4. 実験システムと評価アプリケーション

4.1 実験システムの概要

提案したウェブ環境下でのThickクライアントをベースにしたモデルを評価するために、実験システムを構築した。これを図5に示す。

実験システムは4種のノードがネットワークで接続された環境である。2種の固定ノードとしてデスクトップPCを、2種のモバイルノードとしてノートPCを用いた。固定ノードはコンテンツプロバイダとキャッシングプロキシであり、これらの間は

100MbpsのFast Ethernetで接続されている。固定ノードはモバイルノードとの接続に無線LANインタフェースも持つ。一方モバイルノードは自クライアント及び他クライアントであり、これらはネットワークインタフェースとして無線LANのみを持つ。

無線LANの速度は、コンテンツプロバイダとモバイルノードとの間が基本的に1Mbps、キャッシングプロキシとモバイルノードとの間が11Mbps、モバイルノード間も11Mbpsとした。コンテンツプロバイダとモバイルノードの間は、一般に携帯電話のダイヤルアップ回線などで接続されるケースが多いと考えられ、ここでは例えば第3世代携帯電話の回線などを模して、低ビットレートモードで動作する無線LANを実験に用いた。無線LANカードにはAgere社のチップを搭載したIEEE802.11bのカードを使用した。

また各ノードのハードウェア及びソフトウェアとしては、以下のようなものを用いた。まずコンテンツプロバイダは1.7GHz Pentium4のデスクトップPCで、OSはLinux2.4.7である。これにサーブレットの実行環境としてJDK1.3を実装し、WebサーバとしてApache1.3.20、サーブレットを実行するAPサーバとしてTomcat3.2.1をインストールして両者を連携させた。またバックエンドのDBサーバとしてPostgreSQL7.1.3を用意した。ネイティブプロトコルタイプのJDBCを用いて、サーブレットのモジュールの中からSQLを実行できる環境を構築した。

一方クライアントは800MHz Mobile Pentium IIIのノートPCでOSはLinux2.4.7、Javaクライアントアプリケーションやサーブレットを実行できるようにJDK1.3の環境を整えた。ローカルウェブサーバとして同じくApache1.3.20、サーブレットを実行エンジンとしてTomcat3.2.1を用いた。クライアント側でもローカルDBを動かすためPostgreSQL7.1.3をインストールし、JDBCを通してJavaアプリケーションやサーブレットから直接DBにSQL文を投げられるようにした。実験に用いたクライアントのブラウザはNetscape4.78である。

キャッシングプロキシにはLinux2.4.7をインストールした1GHz Pentium IIIデスクトップPCを用いた。プロキシソフトウェアとしてはSquid2.4を用いた。またP2Pで接続する他クライアントとしては、867MHz Transmeta Crusoe TM5800を搭載したノートPCに、同じくLinux2.4.7を

インストールして用いた。

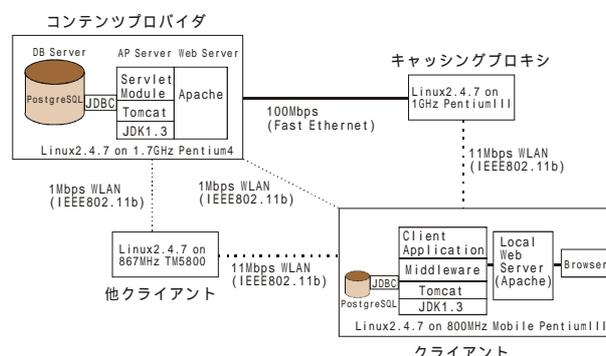


図 5. 実験システムの概要

4.2 評価に用いたアプリケーションの概要

システムを評価するために、以下のようなアプリケーションを実行した。コンテンツプロバイダには1つあたり5MBのコンテンツが複数用意され、コンテンツデータとそのメタデータを別々に管理しながらこれらをクライアントへ提供する。クライアントへの提供ルートにはコンテンツプロバイダからクライアントへの直接接続、キャッシングプロキシを通じた接続、他ノードを通してのP2P接続を考える。メタデータはコンテンツプロバイダからクライアントへの直接接続のみで提供され、コンテンツデータはキャッシングプロキシや他ノードを通して提供することができる。コンテンツデータはOpenSSLのTripleDESで暗号化されており、クライアントへダウンロードされた後、メタデータの中にあるコンテンツ鍵を利用して復号化され利用される。コンテンツデータに関しては、複数のファイルをまとめてクライアントへダウンロードされる場合もある。この評価アプリケーションは、例えば音楽ファイルなどをクライアントへ提供するサービスを想定している。

5. 実験システムの評価

5.1 コンテンツプロバイダークライアント間直接接続

はじめにコンテンツプロバイダとクライアントの直接接続のみの環境において評価を行った。この環境においてダウンロードするコンテンツサイズを変化させた時のサ

ービスにかかる実行時間を測定した。以下の実験において、複数のコンテンツをダウンロードした場合も復号化は最初の 5MB のデータ 1 つについてのみ行っている。実験結果を図 6 に示す。転送するデータ量が増える程コンテンツのデータ転送に長い時間がかかり、トータルの実行時間が極めて長くなっていることがわかる。

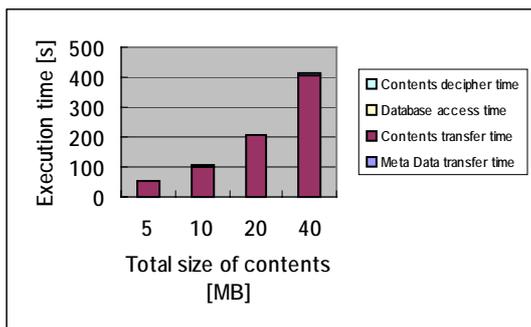


図 6. コンテンツプロバイダ クライアント接続 (1Mbps 無線 LAN 回線)

5.2 キャッシングプロキシの利用

次にキャッシングプロキシを利用し、コンテンツプロバイダ キャッシングプロキシ クライアントという接続構成で実験を行った。コンテンツプロバイダ キャッシングプロキシ間は有線の 100Mbps Fast Ethernet で接続されており、コンテンツプロバイダ クライアント間は 1Mbps の無線 LAN、キャッシングプロキシ クライアント間は 11Mbps の無線 LAN でそれぞれ接続されている。この実験環境は、ホットスポットにおいてクライアントがプロキシと接続しコンテンツデータを受け取るという状況を想定している。

実験結果を図 7 に示す。この結果はプロキシにデータがキャッシュされているケース (キャッシュヒット) である。

実行結果を見ると、キャッシングプロキシを利用したことにより、コンテンツデータの転送時間が大幅に短くなり、いわゆるホットスポットの効果が現れていることがわかる。またキャッシュミスのケースについても実験を行ったが、キャッシュヒットのケースの実行時間とほぼ同じであった。これはクライアントからプロキシに対する無線 LAN でのアクセスに比べ、コンテンツプロバイダ プロキシ間の有線接続の転送速度は十分に速いからであろう。

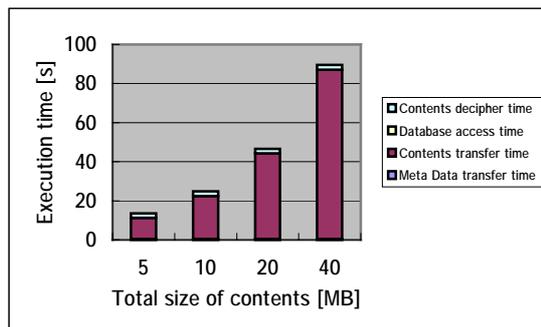


図 7. コンテンツプロバイダ プロキシクライアント接続

5.3 他クライアントとの P2P 接続の利用

今度は固定ノードであるキャッシングプロキシではなく、クライアントと同じモバイルノードである他クライアントが一種のプロキシの役割を果たす P2P 接続環境での評価を行った。クライアント間は 11Mbps の無線 LAN で接続されている。またコンテンツプロバイダと両クライアント間の回線速度は 1Mbps である。

この場合も、他ノードにデータが存在した場合 (キャッシュヒット) と存在しなかった場合 (キャッシュミス) について評価を行った。存在する場合にはそのデータが転送され、コンテンツプロバイダへ他クライアントはアクセスしない。一方他クライアントにデータが存在しなかった場合には、他クライアントが一種のプロキシとして働き、コンテンツプロバイダよりデータを獲得し、かつリクエスト元のクライアントへそのデータを転送する。どちらの場合にも、メタデータはクライアントがコンテンツプロバイダから直接獲得する。実行結果を図 8 と図 9 に示す。

この結果を見ると、まず他クライアントにデータがあった場合 (キャッシュヒット) と無かった場合 (キャッシュミス) とで実行時間が大きく違うことがわかる。この実験は固定ノードであるキャッシングプロキシを用いた場合と異なり、プロキシの役割をモバイルノードが担っている。そのためキャッシュミスが起こった場合には低ビットレートの回線を経由してコンテンツを獲得する必要が生じ、コンテンツデータの転送時間が極めて長くなる。それに対し、他クライアントに必要とするコンテンツが存在した場合には、固定ノードであるキャッ

シングプロキシを用いた場合とほぼ同じ程度の性能が実現されている。

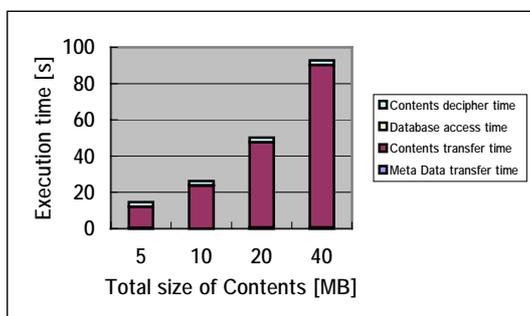


図 8.コンテンツプロバイダ 他クライアント クライアント接続(キャッシュヒット)

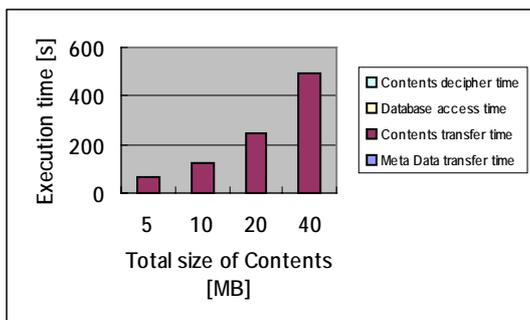


図 9.コンテンツプロバイダ 他クライアント クライアント接続(キャッシュミス)

6. まとめ

本研究ではテレマティクス環境においてクライアントにコンテンツを配布するための枠組みについての検討を行った。テレマティクスでは通常のモバイル環境と比べ、クライアントに比較的高機能、高性能の端末を利用することが可能であり、これにThick クライアントモデルを取り入れることにより、柔軟性に富み質の高いサービスを実現できる可能性がある。本研究ではThick クライアントモデルをベースにし、課金処理などサービスを実現するために必要な処理も含めたコンテンツの管理方式を提案した。この方式ではコンテンツデータとそのメタデータを分けて管理しているため、コンテンツデータは様々な経路で入手することができる。またクライアント側で行うメタデータの処理を工夫することによりオフラインで多くのサービスを実現する

ことを可能としており、これによりネットワークへの接続機会が少ないテレマティクス環境に適した仕組みを実現している。

また実験システムを用いて、提案したモデルの評価を行った。実験結果より、クライアントがコンテンツプロバイダと低ビットレートの回線でしか接続されない場合も、ホットスポットのキャッシングプロキシや他クライアントとのP2P接続を利用することにより、コンテンツデータの転送時間は短くなる。そのような仕組みを用いた結果、十分に現実的と言えるサービスの実行時間を実現することができた。

今後の課題としては、より現実的なネットワーク環境における評価、メタデータを利用したパーソライゼーション等、更に高度なサービス実現のための検討などが挙げられる。

参考文献

- [1] I. Ahmad, "Network computers: The changing face of computing", IEEE Concurrency, vol.8, no.4, pp.9 - 11, 2000.
- [2] J. Ala-Laurila, J. Mikkonen, and J. Rinnemaa, "Wireless LAN Access Network Architecture for Mobile Operators", IEEE Communications Magazine, vol.39, no.11, pp.82 - 89, November 2001.
- [3] K. Moss, "Java Servlets Second Edition", McGraw-Hill, 2000.
- [4] "JDBC Data Access API", <http://java.sun.com/products/jdbc/>
- [5] J. Challenger, A. Iyengar, K. Witting, C. Ferstat, and P. Reed, "A Publishing System for Efficiently Creating Dynamic Web Content", In Proceeding of IEEE INFOCOM 2000, Tel Aviv, Israel, 2000.
- [6] R. Floyd, R. Housel, and C. Tait, "Mobile Web Access Using eNetwork Web Express", IEEE Personal Communications, vol.5, no.5, pp.47 - 52, October 1998.
- [7] D. Clark, "Mobile Processors Begin to Grow Up", IEEE Computer, Vol.35, No.3, pp.22 - 25, March 2002.
- [8] "Java Servlet Technology", <http://java.sun.com/products/servlet/>
- [9] "Java Server Pages", <http://java.sun.com/products/jsp/>