

モバイルインターネットアクセスにおけるリクエストパイプラインの有効性

関口 克己† 鶴巻 宏治† 北口 雅哉† 高橋 修†
NTT DoCoMo マルチメディア研究所

2001年にサービスを開始するIMT-2000では、従来のシステムに比べ高品質、広帯域のパケット通信サービスの提供が予定されている。これにより携帯電話においても固定回線並のインターネットアクセス環境が実現し、高速なWWWブラウジング、マルチメディアメール、音楽配信サービスなど多様なインターネットサービスが利用可能となる。また、インターネットコミュニティにおけるこれまでの様々な研究成果の適用によって、より快適なモバイルインターネット環境の実現も期待できる。

本稿では、インターネットで最もよく利用されているサービスの一つであるWWW(World-Wide Web)に着目し、HTTP/1.1で規定されているリクエストパイプラインの有効性について検証を行った。一般に普及しているWWWブラウザ製品にリクエストパイプライン機能を追加するアドオンソフトを試作し、第三世代携帯電話システムを擬似するシミュレータ環境で効果の確認を行った。

An Implementation of Request Pipelining Method on Mobile Internet Access Environment

Katsumi Sekiguchi, Koji Tsurumaki, Masaya Kitaguchi, Osamu Takahashi
NTT DoCoMo Multimedia Laboratories

IMT-2000 is going to provide high quality and wideband mobile packet switch services. That enable us to use various services like WWW access, Multimedia Mail Service and Music Distribution Service. A lot of research results contributions in Internet community will improve mobile Internet access environment.

In this paper, we focused one of the most popular Internet Service WWW. We implemented the request pipelining defined HTTP/1.1 using plug-in interface of the Microsoft Internet Explorer. We performed simulation of the 3rd generation mobile network and evaluated its transmission efficiency.

1. まえがき

2001年にサービスを開始するIMT-2000では、従来のシステムに比べ高品質、広帯域のパケット通信サービスの提供が予定されている。これにより携帯電話においても固定回線並のインターネットアクセス環境が実現し、高速なWWWブラウジング、マルチメディアメール、音楽配信サービスなど多様なインターネットサービスが利用可能となる。

IMT-2000パケットチャネルの伝送特性の特徴としては、高品質、広帯域、上り下り非対称(サービス開始時)性があげられる。加えて、無線によるパケット転送特有の網内遅延特性を持つことが想定され、従来システムであり着目されなかった遅延特性が、ベアラの広帯域化によって顕在化してくる可能性がある。

従来から、帯域幅と遅延の積(delay*bandwidth product)が比較的大きなネットワーク上での、TCPアプリケーションのパフォーマンスの問題は、IETFなどの公の場で活発に議論されており、研究成果

も数多く提案されている[3,4,5,7]。また、シミュレーション環境における検証結果も、遅延特性がTCPにおける送受信バッファの枯渇やHTTPにおけるレスポンス性の低下の原因となる場合があることを示唆している[1]。

IMT-2000パケットチャネルのような伝送特性を持つネットワーク向けの最適化の試みとして、トランスポート層プロトコルにおけるパラメータセットであるA TCP Profile for W-CDMA(W-TCP)がInternet Draftとして提案されている[2]。またアプリケーション層のプロトコルでもRFC2616[6]で標準化されたHTTP/1.1に規定されているリクエストパイプラインの技法が、WWWアプリケーションのスループット性能、レスポンス性能の特性改善に有効であると考えられる。

本研究においては、特にWWWアプリケーションに着目し、リクエストパイプラインの有効性や実装方法について検討を行っている。本稿では、リクエストパイプラインが動作するHTTPクライアントを、Microsoft社が公開している非同期プラグインインターフェースを利用してInternet Explorerのアドオンソフトウェアとして実装を行い、回線シミュレータを用いた擬似ネットワークで評価し有効性の確認を行った結果について報告する。

†株式会社NTTドコモ マルチメディア研究所
NTT DoCoMo, Inc. Multimedia Laboratories
3-5, Hikarinooka, Yokosuka, Kanagawa, 239-8536, Japan

2. リクエストパイプラインング概説

2.1. パーシステントコネクション

パーシステントコネクションとは、一度の TCP コネクションの確立で、オブジェクトの取得ごとにコネクションを切断することなく複数の HTTP リクエストを送信できるコネクション確立方法である。HTTP/1.0 の実装などで、パーシステントコネクションを用いないで、WWW アクセスを行った場合、各々のオブジェクトの取得ごとに TCP コネクションの確立、切断が行なわれる。TCP のコネクションの確立は 3 ウェイハンドシェイクと呼ばれる動作により行なわれ、HTTP クライアントが複数のオブジェクトを取得する場合、各オブジェクトの取得ごとにこの 3 ウェイハンドシェイクが行われることになり、HTTP サーバおよびネットワークに過大な負荷を生じさせる原因となる。特に RTT が非常に大きいネットワークにおいては、TCP のコネクションの確立、切断に要する時間が大きくなることによりスループット性能に与える影響は多大で

ある。

RFC2616 で標準化されている HTTP/1.1 においては、一度の TCP コネクションの確立で複数のオブジェクトの取得を行うことが可能な、パーシステントコネクションが定義されている。パーシステントコネクションの利用により、一度コネクションを確立してしまえば、それ以降のオブジェクトの取得ごとに発生する TCP コネクションの確立、切断処理を省略することができる。パーシステントコネクションの利点として次のようなことが上げられる。

- TCP コネクション確立、切断処理の削減によるネットワーク、HTTP サーバ、HTTP クライアントの負荷の減少。
- リクエストパイプラインングの利用による効率的なオブジェクトの取得。

HTTP/1.1 では、パーシステントコネクションはデフォルトの動作として規定されている。

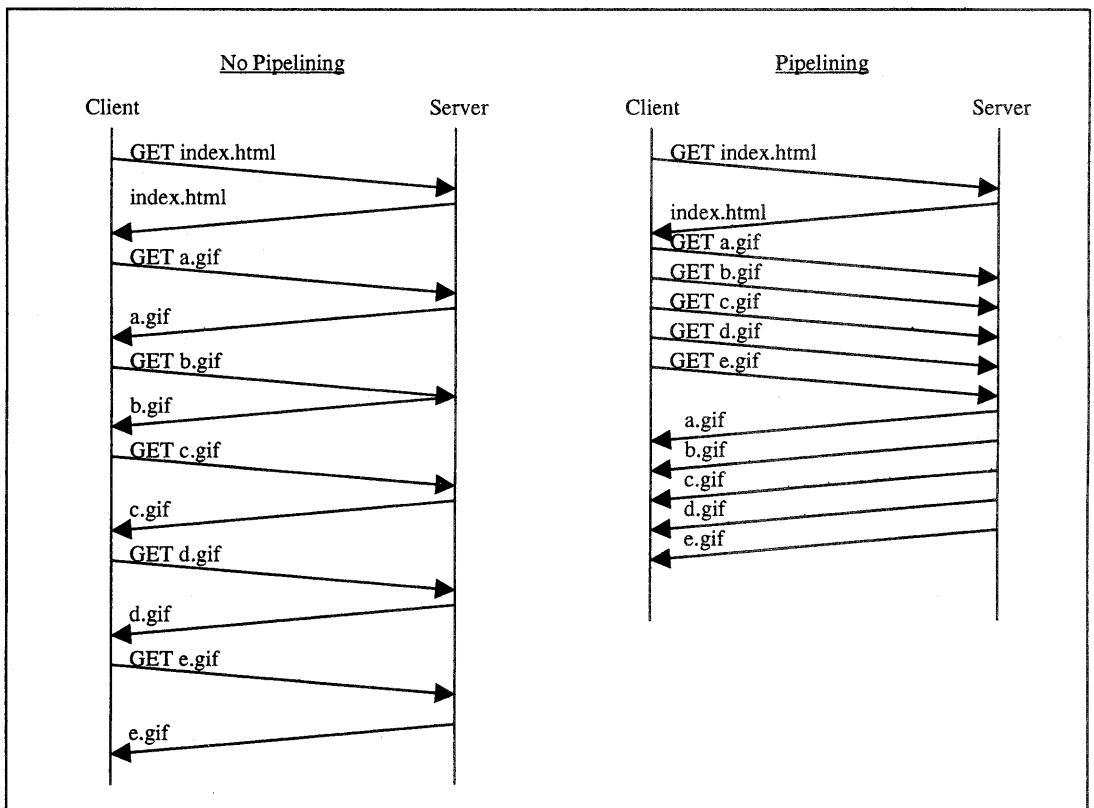


図2.1 HTTPによるオブジェクトの取得

2.2 リクエストパイプライン

前節で述べたように、パーシステントコネクションでは、一度の TCP コネクションの中で複数のリクエストレスポンスを繰り返すという動作が可能となる。リクエストパイプラインでは、パーシステントコネクションの利用を前提として、一度の TCP コネクションの確立で、複数のリクエストの送信を HTTP サーバからのレスポンスを待たずに次々に送信を行うことができる。HTTP サーバ側はリクエストを受信した順に連続的にオブジェクトの送信を行う。

HTTP の GET の動作シーケンスを図 2.1 に示す。このシーケンス図では、HTML ファイル(index.html)に 5 つのインライン画像(a.gif - e.gif)を持ったページの取得の例を示している。HTTP クライアントは index.html の GET を行い、内容の解析を行う。内容の解析後、5 つのインライン画像の GET を行うが、リクエストパイプラインを用いない場合、1 つの画像の GET を送信した後、HTTP サーバから画像本体を受信されるまで同一コネクション上では次の画像の GET の送信を行わない。一方、リクエストパイプラインでは、index.html の内容の解析を行った後、5 つの画像の GET を HTTP サーバからのレスポンスを待たずに連続的に送信を行う。HTTP サーバ側は GET を受信した順に画像本体を連続的に送信する。

リクエストパイプラインでは通常のリクエスト方式の GET 送信後の待ち時間がなく、特に RTT の大きいネットワークで用いた場合、およびオブジェクトサイズが小さくその数が多数あるようなホームページでの利用環境での効果が高いと考えられる。

RFC2616 において、パーシステントコネクションをサポートする HTTP クライアントはリクエストパイプラインの実装が推奨されている。また HTTP サーバにおいては、HTTP クライアントからリクエストパイプラインによるリクエストを受け付けた場合は、それに応答することが必須とされている。

リクエストパイプラインは HTTP サーバでの実装は必須となっているものの、HTTP クライアントでの実装は任意となっており、現在の市場に出ている主要な WWW ブラウザでの実装は行なわれていない状況である。

3. プラグインソフトウェア概説

3.1. システム概要

本システムは、Microsoft 社の Internet Explorer のプラグインとして機能し、WWW ブラウザから HTTP サーバへのリクエストをフックし、WWW ブラウザの標準的なリクエスト方式をリクエストパイプライン方式に変更しアクセスの高速化をはかるものである。本システムは Internet Explorer のプラグインであるため、レジストリにあらかじめ登録をしておけば、起動、終了に関してはユーザ側でまったく意識することなく行うことが出来る。

3.2 システム構成

図 3.1 に本システムの構成を示す。Internet Explorer.exe は Internet Explorer 本体部であり、Urlmon.dll は Internet Explorer の非同期プラグインインターフェースを提供する。Http.dll はリクエストパイプラインのプラグインソフトウェア本体部分であり、urlmon.dll 以下に配置される。Http.dll モジュールは、リクエスト管理、リクエスト受付管理、接続管理、リソース管理、サーバ接続管理、接続チャンネルより構成されている。

http.dll モジュールが起動されると、リクエスト受付管理と接続管理が起動される。Internet Explorer よりリクエストが発行され、urlmon.dll に引き渡されると、urlmon.dll はリクエスト受信を生成し、リクエストを引き渡す。リクエスト受信は引き渡されたリクエストの解析を行い、処理対象外のリクエストだった場合、Internet Explorer に処理を差し戻し、処理対象だった場合、リクエスト管理の起動を行う。リクエスト管理は、Internet Explorer の管理するキャッシュに、リクエストの発行を行おうとしているオブジェクトが存在するか確認を行い、存在する場合、キャッシュの更新日時を取得し、リクエスト情報に更新日条件を付加し、リクエスト受付管理にリクエスト登録要求を発行する。リクエスト受付管理は、リクエストマップにリクエストの登録を行い、接続管理にリクエストの登録通知を発行する。接続管理はリクエスト通知を受け取ると、自身が保持しているリソースマップの検索を行い、該当オブジェクトのリソース管理が存在するか確認を行う。リソース管理が存在しなかった場合、新たにリソース管理の起動を行い、接続管理は該当オブジェクトのリソース管理にリクエスト通知を発行する。リクエスト通知を受け付けたリソース管理は自身のバッファの確認を行い、オブジェクトがすでに取得済みだった場合、接続管理へリソース到着通知を行う。オブジェクトが無かった場合、該当オブジェクトのサーバ接続管理が存在するか、接続管理へ

問い合わせを行う。接続管理は、サーバ管理マップを検索し、該当オブジェクトのサーバ接続管理が確認を行い、存在しなかった場合、サーバ接続管理を新たに立ち上げ、リクエスト要求の発行を行う。リクエスト要求を受け付けたサーバ接続管理は、接続チャンネルの起動を行う。接続チャンネルは HTTP サーバと接続を行い、順次リクエストの発行を行う。

キャッシュ機能は、Internet Explorer が管理しているキャッシュおよびキャッシュ機能をそのまま用いる。プラグインでは、Internet Explorer 本体にキャッシュの有無の問い合わせを行うインターフェースのみ提供し、独自のキャッシュ管理の機能は持たない。

3.2. 機能概要

3.2.1. プラグイン

本システムは、Internet Explorer の非同期プラグインインターフェースを利用したプラグインソフトウェアであり、http.dll を Internet Explorer のレジストリに登録することによって、プラグインとして動作する。

3.2.3. ブラウザとのデータ送受信

http.dll は非同期プラグインインターフェースを提供する urlmon.dll を通じて、WWW ブラウザとデータ送受信を行う機能を提供する。

http.dll は Internet Explorer から urlmon.dll を通じて送信されてくる GET のリクエストのみ受け取る。GET 以外のリクエストメソッドは http.dll の処理対象外であるため、そのまま Internet Explorer へ返却する。ヘッダ項目には Connection: Keep-Alive とキャッシュのための If-Modified-Since のみを付加する。受信した結果はパ

3.2.2. キャッシュ

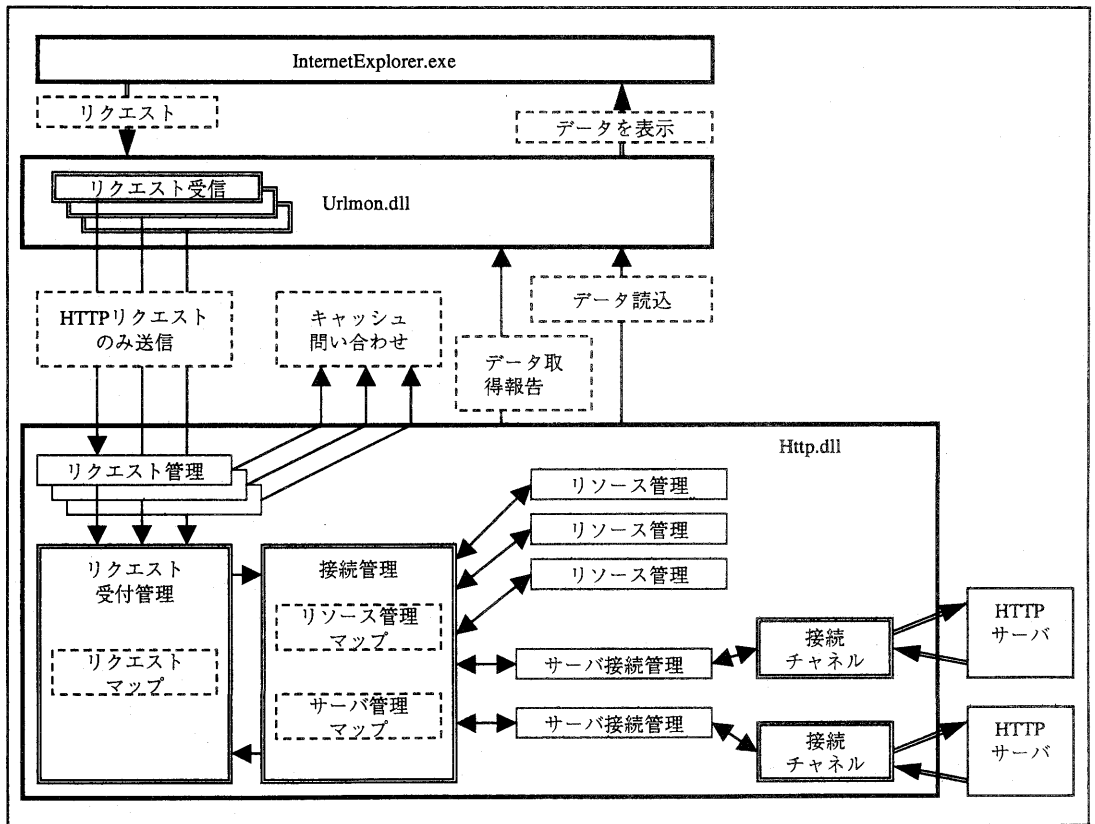


図3.1 プラグイン構成図

ツファに書き込むことで Internet Explorer への送信が行なわれる。

3.2.4. HTTP サーバとのデータ送受信

http.dll は、HTTP サーバがパーシステントコネクションをサポートしているか判断を行い、サポートしている場合、パーシステントコネクションにより TCP コネクションの確立を行う。サポートしていない場合は、通常のコネクションの確立を行う。

HTTP サーバがパーシステントコネクションをサポートしている場合、http.dll はその HTTP サーバがリクエストパイプラインをサポートしているか判断を行い、サポートしている場合はリクエストパイプラインでリクエスト送信を行う。サポートしていない場合は通常のリクエスト送信を行う。http.dll においては、HTTP サーバがパーシステントコネクションおよびリクエストパイプラインをサポートしているかどうかは、リクエスト送信の前には不明である。そのため複数の取得オブジェクトがある場合、最初の1つのオブジェクトに対するリクエストを発行し、その返り値によってサポートされている方式を認識する。

4. 評価環境

4.1. 実験系構成

本研究で用いた実験系の構成を図 4.1 に示す。本実験系においては、HTTP クライアントと HTTP サーバを W-CDMA シミュレータ経由で接続している。各装置間は 10Mbps の Ethernet にて接続している。

HTTP クライアントには、Windows98 に W-TCP のプロファイルを適用し、WWW ブラウザに Internet Explorer 5.0 にリクエストパイプラインのプラグインを登録したものをを用いた。HTTP サーバには、Solaris 7 に Apache 1.3.6 を搭載したものをを用いた。Solaris 7 はあ

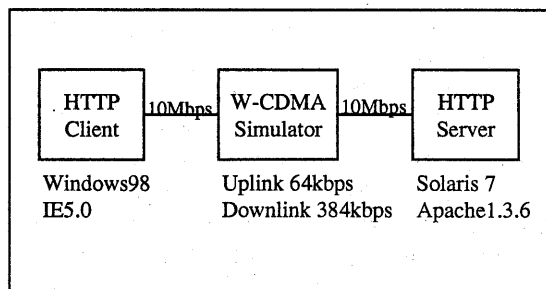


図 4.1 実験系構成図

らかじめ W-TCP に相当するプロファイルを含んでおり、また Apache 1.3.6 はリクエストパイプラインを標準でサポートしているため、サーバ側は通常の HTTP サーバと同一の設定を行っている。W-CDMA シミュレータは IMT-2000 の無線区間の特性を擬似する装置であり、本研究におけるシミュレータの設定は上り 64kbps、下り 384kbps としてある。

実験方法は、HTTP サーバにコンテンツを配置し、HTTP クライアントからオブジェクトの取得を行い、全てのオブジェクトの取得に要した総転送時間の測定を行うこととした。この測定をリクエストパイプラインの有無それぞれについて行った。

4.2. 評価用オブジェクト

HTTP サーバに配置する、評価試験用の取得オブジェクトは、定量的な評価のため、1つの HTML ファイルに一定サイズのオブジェクトを複数配置したテスト用コンテンツと、一般的な環境における特性の評価のため、Internet 上に実際に設置されている一般のホームページのサンプルの 2 通りを用いた。

テスト用コンテンツは、HTML ファイルに一定サイズのインライン画像を貼り付け、オブジェクトの数、サイズごとに複数の HTML ファイルを準備した。その構成を図 4.2 で示す。一般のホームページのサンプルは、ドコモホームページと代表的なポータルサイトから無作為に選んだ。これらのコンテンツのオブジェクト数とサイズを図 4.4 で示す。

Contents	Object
①256byte/obj.	1, 20, 40, 80, 100 files
②2kbyte/obj.	1, 20, 40, 80, 100 files
③4kbyte/obj.	1, 20, 40, 80, 100 files
④8kbyte/obj.	1, 20, 40, 80, 100 files
⑤16kbyte/obj.	1, 20, 40, 80, 100 files

図 4.2 テスト用ホームページの構成

Contents	Object	Average Size
DoCoMo Net	77	1. 2kbyte
Goo	24	2. 0kbyte
Yahoo	5	6. 2kbyte

図 4.3 サンプルホームページの構成

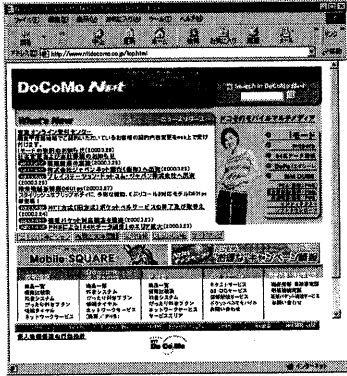


図 4.4 一般ホームページのサンプル

5. 試験結果

5.1. テスト用コンテンツ

本測定のために準備したテスト用コンテンツを取得した場合の測定結果を図 5.1 で示す。グラフは縦軸に全てのオブジェクトの取得が完了するまでの所要時間を示し、横軸に1つの HTML ファイルに含まれるインライン画像数を示している。インライン画像ファイルのサイズは 256byte から 16kbyte までの 5 種類とし、それぞれの結果をグラフ上に示している。

5.2. 一般のホームページ

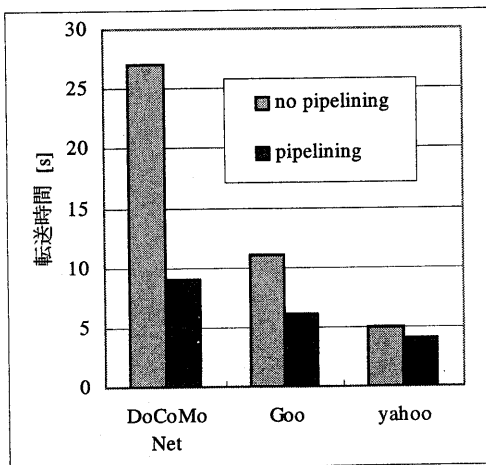


図 5.2 一般ホームページの試験結果

一般のホームページのサンプルを取得した場合の結果を図 5.2 で示す。各コンテンツの左側のグラフがパイプライン無し、右側のグラフがパイプライン有りの測定結果を示す。

6. 考察

テスト用コンテンツを用いた測定において、リクエストパイプラインを用いない場合、オブジェクトサイズ 256byte、オブジェクト数 100 個のコンテンツの総転送時間は 30 秒を要しており、この値は 8kbyte のものと比較して、10 秒以内、16kbyte のものと比較しても 20 秒以内でしかなく、オブジェクトの総転送量を考慮すると、非常に小さい差と言える。リクエストパイプラインを用いない場合、オブジェクトのリクエスト送信後、レスポンスを待つ間必ず、ネットワークの RTT による待ち時間が発生することになり、この値が支配的となり上記のような測定結果になったと考えられる。

リクエストパイプラインを用いた場合、複数のオブジェクトのリクエストを行った時は、上記の RTT の影響が少なくなるため、オブジェクトサイズによらず効果が現れていることが分かる。図 5.1 の 6 番目のグラフに、リクエストパイプラインと、通常のリクエストとの総転送時間の比をとった結果を示す。このグラフの値がすべての領域で 1 以上になっていることから、リクエストパイプラインの効果が確認できる。またこのグラフより、オブジェクトサイズが小さいほどリクエストパイプラインの効果が高いことが分かる。リクエストパイプラインの利用により、オブジェクトサイズ 256byte、オブジェクト数 10 以上およびオブジェクト 4kbyte、オブジェクトサイズ 40 以上で、オブジェクトの取得のための総転送時間を 1/2 にできることが確認できた。

しかし、オブジェクト数 80 以上で、特性が下がっていること、およびオブジェクトサイズ 16kbyte、オブジェクト数 1 の場合、オブジェクトサイズ 256byte、オブジェクト数 40 以上の領域では、効果の出方が乱れており、これらの原因の解析は今後の課題とする。

インターネット上の一般ホームページのサンプルを用いた測定において、用意した全てのコンテンツにおいてリクエストパイプラインの効果が認められた。図 4.3 にそれらの構成を示したが、リクエストパイプラインを用いない場合、オブジェクト数が多く、平均オブジ

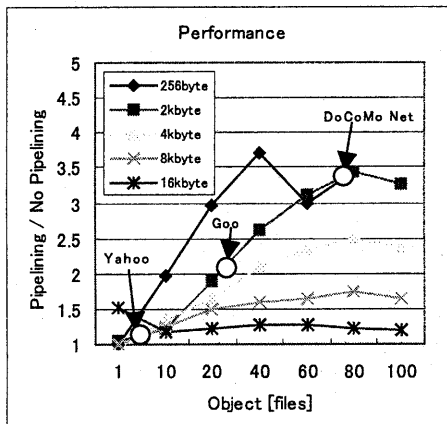
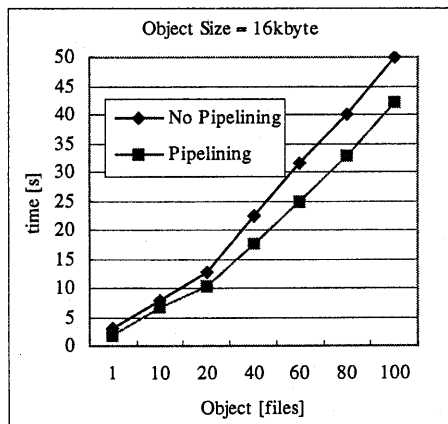
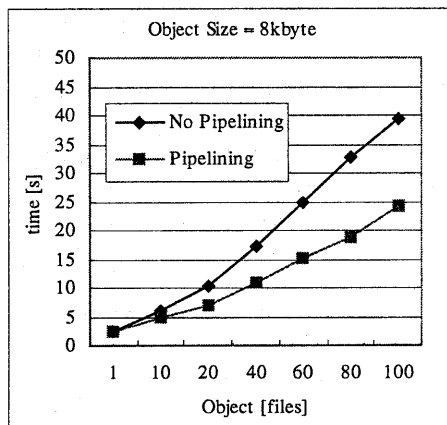
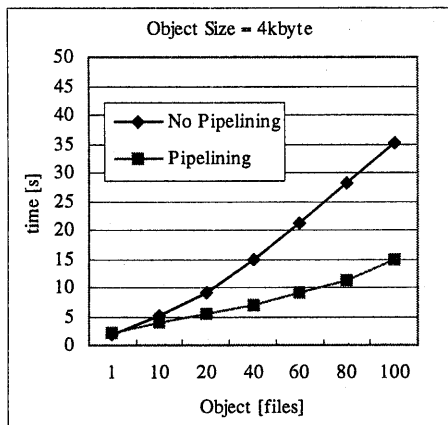
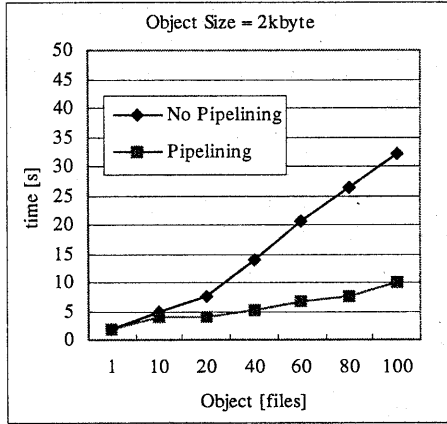
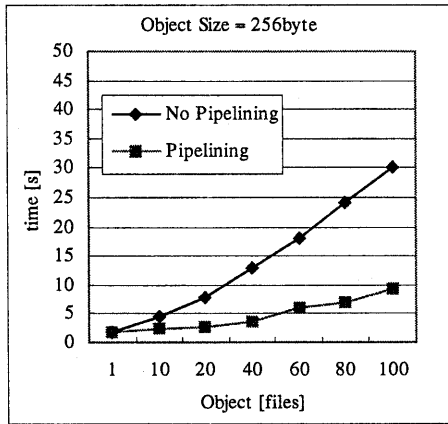


図5.1 テスト用コンテンツの測定結果

エクトサイズが小さなコンテンツでは、ネットワークの遅延の影響を強く受け、スループットが著しく低下していることがわかる。しかし、このような状況においてリクエストパイプライニングを用いることで、オブジェクト数が約80個と最も多い DoCoMo Net のコンテンツにおいて、全てのオブジェクトの受信に27秒を要していたものが、8秒に短縮されることがわかった。

これらのホームページのコンテンツが図5.1のグラフのどの領域に該当するかをグラフ中に示している。テスト用コンテンツと一般のホームページの測定で得られた結果がよく一致していることが分かる。

7. むすび

本研究では、RFC2616 に定義されたリクエストパイプライニングの実装を行い、広帯域、高遅延の伝送特性を持つ擬似回線での特性を評価した。測定の結果、複数のオブジェクトを取得する場合には、常にリクエストパイプライニングの効果が現れていることが分かり有効性の確認をすることができた。また一般のホームページのアクセスではオブジェクト数が小さく、かつ多量に配置されたコンテンツでの効果が特に高く、この結果は定量的なコンテンツを用いた測定より得られた、オブジェクトの数が小さくかつ多量に配置されたコンテンツにおいて特に効果が高いという結果にも一致している。従ってモバイルインターネットアクセスにおける WWW ブラウジングにおいては、リクエストパイプライニングは非常に有効であり、回線の利用効率を高められるということがわかった。

謝辞 本研究を進めるにあたり、実験系の構築およびデータ取得にご協力頂いた、NTT ソフトウェア株式会社の高橋忠氏並びに田中一郎氏に感謝いたします。

参考文献

- [1] 石川, 稲村, 高橋, "W-CDMA 向け TCP プロファイル", MBL/ITS 合同研究会, Nov. 2000
- [2] Internet Draft, "A TCP profile for W-CDMA: 3G wireless packet service."
- [3] RFC1323, "TCP Extensions for High Performance"
- [4] RFC2018, "TCP Selective Acknowledgment Options"
- [5] RFC2414, "Increasing TCP's Initial Window"
- [6] RFC2616, "Hypertext Transfer Protocol --

HTTP/1.1"

[7] RFC2757, "Long Thin Networks"