

ハイパーテキスト間のデータ共有を考慮した放送型情報提供方式

石川 裕治 田辺 雅則 箱守 聡 井上 潮

NTTデータ通信(株) 情報科学研究所
〒210 神奈川県川崎市

E-mail: {ishikawa, tanbe, hako, uinoue}@lit.rd.nttdata.co.jp

概要

モバイルコンピューティング環境において、データを共有しているハイパーテキストの放送方式について述べる。ハイパーテキストはリンク先として、同じデータを共有していることが多い。このような場合、ハイパーテキストを個別に伝送すると共有されているデータを複数回放送することになり効率が悪い。本論文では、共有データを1度だけ放送することで、クライアントのデータ取得時間を短縮する方式を提案し、利用者の待ち時間を評価することにより提案方式の有効性について考察する。

キーワード: モバイルコンピューティング、放送型通信、ハイパーテキスト

Data Broadcast Method for Hypertexts with Shared Files

Yuji Ishikawa, Masanori Tanabe, Satoshi Hakomori, and Ushio Inoue

Laboratory for Information Technology
NTT Data Corporation

Kawasaki city, Kanagawa 210, Japan

E-mail: {ishikawa, tanbe, hako, uinoue}@lit.rd.nttdata.co.jp

Abstract

This paper studies a data broadcast method for hypertexts with shared files under a mobile computing environment. More than one hypertext often shares the same files as their hyperlink destination. If hypertexts with shared files are transferred individually, the same files are broadcast several times and the efficiency of transmission becomes low. In this paper, we propose a method, which broadcasts shared files only once, for reducing wait time of clients to receive files.

Keywords: mobile computing, data broadcasts, hypertext

1 はじめに

近年の携帯型コンピュータおよび無線通信技術のめざましい発展により、携帯端末から無線ネットワークを通じて様々な情報にアクセスできるようになりつつある。現在、携帯端末から携帯電話やPHSを利用してサーバに接続し、数十Kbpsの伝送速度で電子メール等の情報を取得するのが主流である。しかし、今後は、WWW等のマルチメディア情報の提供サービスが普及していくと思われる。

我々は無線ネットワークを用いて、展示会や博物館のような特定の場所に訪れた人々を対象とした情報提供サービスの実現をめざしている。サービスの特徴を以下に挙げる。

特徴1： 提供者側は、テキストや静止画、動画、音声等で構成されたマルチメディアデータを、利用者からの要求に応じて提供する。

特徴2： 多くの利用者が入れ替わり訪れ、ネットワークに頻繁にアクセスする。

特徴1から、サービスを実現するシステムはリンク構造を持つマルチメディアデータを扱う。そのデータを扱う形式の1つにハイパーテキストがあり、WWWで用いられているHTML形式が広く利用されている。そこで、我々のシステムでも利用者にハイパーテキストの形式で情報を提供することを考える。

一方、特徴2から、サービスの実現には高い負荷に耐える無線ネットワークが必要である。例えば、無線通信のインフラの1つである無線LANは、伝送速度が数Mbpsであり、同時接続数は数十である。そのため、多数の利用者がサーバに接続して情報を取得するには十分ではない。

我々はこの問題を解決するため、従来のオンデマンド型の通信方式に放送型の通信方式を組み合わせることで、多数の利用者に効率的に情報を提供する方式を提案している[1]。放送型通信では1回の送信で多数の端末が同時にデータを取得できる。このため、音声や画像を含んだデータ量の大きいハイパーテキストを放送型通信で提供することにより、ネットワークの負荷を減らすことができる。ここで、複数のハイパーテキストは同じデータを共有していることがあり、ハイパーテキストを放送する時にはデータ共有を考慮することが重要である。

本論文では、放送によってハイパーテキストを効率的に伝送することを目的として、ハイパーテキスト間のデータ共有を考慮した放送データ構成方式を提案する。2章では、放送によりハイパーテキストを伝送する方式について述べ、関連研究を紹介する。3章でデータの共有を考慮した放送方式について述べた後、4章で机上評価および結果の考察を行う。最後に5章で全体のまとめを述べる。

2 放送によるハイパーテキストの伝送

2.1 情報提供システムの構成

放送型情報提供システムのシステム構成を図1に示す。システムはデータを提供するサーバと、

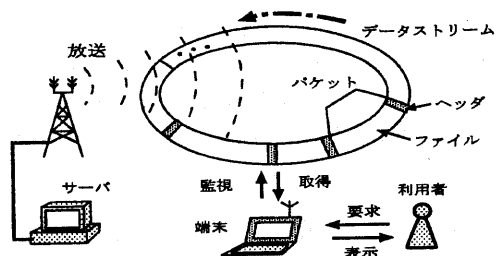


図1: 放送型情報提供システムの構成

データを取得して利用者に提供する端末から構成されている。サーバのデータはファイルにヘッダを付加したパケットの形で提供される。サーバはデータストリームと呼ぶパケット列を構成し、周期的に繰り返し放送する。

一方、端末はヘッダを見て放送データから必要なファイルを取得する。ここで、端末には以下の前提条件をおく。

前提1： 端末にキャッシュはない。

前提2： 端末は、ハイパーテキストとファイル集合の対応表を持っており、各ハイパーテキストが、どのファイルから構成されているかを知ることができる。

2.2 ハイパーテキストの送信と取得

一般にハイパーテキストはハイパーリンクによって関連付けられた複数のファイルから構成される。例えばWWWのブラウザにハイパーテキストを表示する場合、1ページに表示されるハイ

パーテキストは1つのHTMLファイルと複数の画像ファイルで構成されていることが多い。そこで本論文では、1つのハイパーテキストの伝送を、1ページを構成するファイルの集合の伝送として扱う。例えば、ハイパーリンクが張られた4つのファイル $\{e_1, f_1, f_2, f_3\}$ のデータが1ページに表示されるとすると、このハイパーテキストの伝送とは $\{e_1, f_1, f_2, f_3\}$ の4つのファイルを伝送することである。

一方、データストリーム中には各ハイパーテキストを重複なく配置し、かつ、1つのハイパーテキストを構成するファイルも重複なく配置する。すると、端末は1周期待たばどのファイルでも1度だけ取得することができる。つまり、どのハイパーテキストに対しても放送を最悪でも1周期受信していれば、それを構成するファイルをすべて取得することができる。

ハイパーテキストの取得手続きは以下の4ステップから成る。

Step 1: あるハイパーテキスト h の取得要求を利用者が発する。

Step 2: 端末は対応表から h を構成するファイル (例えば $\{e_1, f_1, f_2, f_3\}$) を知る。

Step 3: 放送データを監視し、ヘッダを見て必要なパケットをすべて取得する。

Step 4: そのパケットからハイパーテキストを構成し、利用者に提供する。

Step3の待ち時間はデータストリームの構成によって変わる。ハイパーテキストを取得する際の待ち時間を最小にするためには、それを構成するファイルのパケットを連続して配置する方式(連続配置方式)が良い[2]。さらに、[2]では各ハイパーテキストの連続したパケット列を1つの大きなパケット(パッケージ)にまとめる方式(パッケージング方式)を提案した。連続配置方式に比べ、パッケージング方式には次のような利点がある。

- ヘッダの検出回数が減るためパケットの取得処理が簡便になる。
- 端末に対する前提2(2.1節)の対応表が不要である。

以上の理由からデータストリームの構成にはパッケージング方式をとる。

2.3 関連研究

放送を用いてデータを提供する方法に関する研究としてはAcharyaらによるBroadcast Disks[3]がある。この論文ではデータのアクセス確率に基づいて各データの放送頻度を決定し、待ち時間の期待値を最小化する方式が述べられている。また、Imielinski[4]らは放送方式とオンデマンド方式の2つの情報提供方式を組み合わせ、多数の利用者に多種の情報を提供する方式を提案している。これらの研究ではいずれもデータ間のリンク構造を考慮した伝送方式の議論はなされていない。

一方、モバイルコンピューティング環境におけるハイパーテキストの伝送方式に関する研究としてはWebExpress[5]がある。この研究では、端末がキャッシュの内容に関する情報をサーバに送信し、サーバはキャッシュとの差分だけを伝送することでデータ伝送量の削減を図る方式を提案している。この方式は、サーバが各端末に応じた処理を行うため、放送型通信には不向きである。

3 共有データを考慮したハイパーテキストの伝送方式

3.1 取得時間の短縮のための課題

2つのハイパーテキストが同じデータファイルを用いてページを構成している場合を考える。この例を図2に示す。

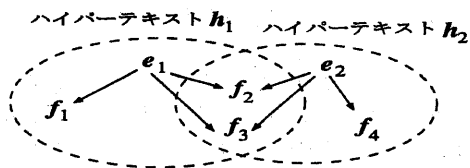


図2: ハイパーテキスト間のデータ共有

図2の場合、 h_1 と h_2 を別々にパッケージングして伝送すると $\{f_2, f_3\}$ を2回送ってしまうことになる。つまり、データ共有を考慮せずにハイパーテキストを個別に伝送すると、共有データを複数回送信してしまい伝送効率が悪くなる。

この問題を解決するには、ハイパーテキストを共有データとその他のデータに分け、共有データは1つにまとめて1度だけ伝送すればよい。例えば h_1 なら $\{e_1, f_1\}$ と $\{f_2, f_3\}$ 、 h_2 なら

ら $\{e_2, f_4\}$ と $\{f_2, f_3\}$ に分けられる。従って $\{e_1, f_1\}$ 、 $\{e_2, f_4\}$ および $\{f_2, f_3\}$ から3つのパッケージ H_1, H_2, F を作り、データストリーム中に配置する。このとき h_1, h_2 のどちらを取得する場合も2つのパッケージ (H_1 と F あるいは H_2 と F) を取得する必要があるが、 H_1, H_2, F をデータストリーム中にどのように配置するかにより、 h_1, h_2 の取得時間は変わってくる。

すなわち、データを共有するハイパーテキストの取得時間を短縮するためには、共有データを考慮したデータストリーム内のパッケージ配置方式を求めることが課題となる。

3.2 パッケージの配置方式

一般性を失う事なく議論を簡単にするため、単位時間当たりサイズ1のデータが送信されるものとする。つまりデータストリームのサイズが T であるとすると、周期も T である。

共有ファイルが複数存在する場合、各ファイルを共有するハイパーテキストの集合が一致しない場合があるが、ここでは、一致する場合に限定して議論する。つまりデータストリームに含まれる m_{all} 個のハイパーテキストは、あるファイル集合 $\{f_i\}$ を共有する m 個のハイパーテキスト $\{h_i\}$ と、共有関係を持たない $m_{all} - m$ 個のハイパーテキスト $\{h'_i\}$ に分けることができる。尚、 h_i と h'_i のアクセス確率をそれぞれ p_i, p'_i とする。また $\{f_i\}$ のパッケージを F 、各 h_i の $\{f_i\}$ 以外のデータファイルのパッケージを H_i 、各 h'_i のパッケージを H'_i とする。 F, H_i, H'_i のサイズをそれぞれ γ, σ_i, s_i とする。

各 h_i のパッケージを F と H_i に分割し、データストリーム中に F を1つだけ配置することを考える。この場合の周期 T' は $T - \gamma(m-1)$ である。

h_i を取得する場合、 F と H_i の2つのパッケージを取得しなければならない。そのため、データストリームに F と H_i が離れて配置されている場合には、取得時間の期待値は増加する。 h_i の取得時間の期待値 E_i は、データストリーム中の F と H_i の距離を d_i とすると以下ようになる。

$$E_i = \frac{T'}{2} + s_i - \frac{s_i^2 - \gamma s_i + \gamma^2}{2T'} - \frac{d_i^2 - T'd_i + s_i d_i}{T'} \quad (1)$$

一方、 h'_i を取得する場合、 H'_i の取得時間の期

待値 E'_i は $T'/2 + s'_i$ である。

したがって、取得時間の期待値は次のように表される。

$$\mathcal{E} = \sum_{i=1}^m p_i E_i + \sum_{i=1}^{m_{all}-m} p'_i E'_i \quad (2)$$

つまり 3.1 節で挙げた課題は、(2) 式の値を最小化する $\{H_1, \dots, H_m\}$ の配置を見つける、という問題を解くことで解決される。配置によって変化するのは d_i のみであるから、(1) 式の第4項の値だけを問題の対象とする。よって上記の問題は次の値 D を最大化する配置を求める問題に置き換えることができる。

$$D = \sum_{i=1}^m p_i (d_i - t_i)^2 \quad \text{ただし } t_i = \frac{(T' - s_i)}{2} \quad (3)$$

各 H_i はできるだけ F の近くに配置したいので、すべての H_i を F の前後に連続して配置すべきである。しかし、これだけの条件では最適解の探索に $m!$ のオーダーの計算時間がかかる。探索空間を狭めるためには、より強力な制約条件が必要になる。

最適な配置の条件を探るため、図3のように隣り合って配置された2つのパッケージ H_i と H_j について考察する。 $d_i = d$ とし、 H_i の方が F に近

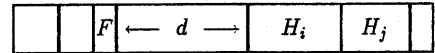


図3: データストリームの例

いとすると $d_j = d + \sigma_j$ となる。この時の D は、 D' を i 番目と j 番目以外の項の和として、以下のように表される。

$$D_{i,j} = D' + p_i (d - t_i)^2 + p_j (d + \sigma_j - t_j)^2 \quad (4)$$

次に H_i と H_j を入れ換える。この時 D は、他の F からのパッケージの距離が変化しないことに注意して、以下の値をとる。

$$D_{j,i} = D' + p_i (d + \sigma_j - t_i)^2 + p_j (d - t_j)^2 \quad (5)$$

(4) 式と (5) 式より次が成り立つ。

$$D_{i,j} > D_{j,i} \Leftrightarrow \frac{p_i}{\sigma_i} > \frac{p_j}{\sigma_j} \quad (6)$$

これは p_i/σ_i の値の大小だけで H_i と H_j の順番を決めて良いことを示している。

つまり、各 H_i を F の前に置か後ろに置かが決まれば、 H_i の順番の最適解を p_i/σ_i の値の

大小により決定できる。しかし、この場合でも最適解を見つけるための計算時間のオーダーは 2^m であり実用的ではない。

そこで、以下の2つの方針に基づき、かつ、 $m \log(m)$ のオーダーの計算時間で準最適解を見つける配置決定方式‘順序配置方式’を提案する。

方針1: すべての H_i を F の前後に連続して配置し、 H_i と F をできるだけ接近させる。各 d_i を小さくすることで D を大きくする。

方針2: p_i/σ_i の値に基づいて H_i を整列する。

順序配置方式の手続きは以下のようなになる。

手順1: H_i を p_i/σ_i の値で昇順に並べる。

手順2: F の前に置くハイパーテキストのリストを before, F の後ろに置くハイパーテキストのリストを after とする。

手順3: before と after のハイパーテキスト列の長さを比較し、その値が小さい方のリストに H_i を手順1の順序で、順次加えて行く。

例えば H_1, \dots, H_5 が p_i/σ_i の値で昇順に並べられており、 σ_i の値をそれぞれ 4, 2, 3, 5, 1 とすると、 H_1, \dots, H_5 は順序配置方式によってデータストリーム中に図4のように配列される。

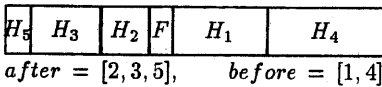


図4: 順序配置方式の例

4 評価

机上評価の目的は以下の2つである。ここでは、ファイルの共有関係を考慮しない時のデータ伝送方式を従来方式と呼ぶ。

- 順序配置方式によって求められる準最適解が、どの程度良質な解であるかを明らかにする。
- 従来方式と順序配置方式の \mathcal{E} の値を比較する。

よって評価は予備実験、方式比較の2段階に分けて行った。

実験用パラメータには以下の仮定をおいた。

伝送速度: 1 Mbps とする。

p_i, p'_i : Zipfの法則[6]に従う。つまり p_i が k 番目に大きい値である時、 p_i は $1/k$ に比例する。 p'_i も同様である。

s_i, s'_i : サイズ s_i のパッケージが存在する確率 $p(s_i)$ は指数分布 $p(s_i) = \frac{\exp(-s_i/\lambda)}{\lambda}$ をとる。 λ は s_i の平均値であり 200Kbyte とする。 s'_i も同様である。尚、 p_i と s_i 、および p'_i と s'_i に相関は無いとする。

γ : λ の 1/2 (100Kbyte) とする。

m_{all} : 放送によって提供されるハイパーテキストの総数は 100個とする。この場合の周期は 20 秒、データストリームのサイズは 20 Mbyte である。

4.1 予備実験

先にも述べたように最適解の計算時間のオーダーは 2^m であるため、予備実験では $2 \leq m \leq 24$ の範囲で検証する。

比較のため、次の2つの方式についても取得時間を計算し、最適配置方式との差を求めた。

ランダム配置方式: 1周期のデータ中に $H_i (i = 1, \dots, m)$ を、間隔を均等にして配置する。

H_i の配置順序は無作為に決定する。この方式は共有データを1つにまとめた点が従来方式と異なるが、方針1・2は適用されていない。

逆順序配置方式: 順序配置方式の手順1において降順に並べる。この方式は方式2が適用されていない。

実験の結果を図5に示す。図5の横の軸は共有データを持つハイパーテキストの数 m であり、縦の軸は最適配置方式の取得時間の期待値 \mathcal{E} の値に対する各方式の \mathcal{E} の差である。ランダム配置方

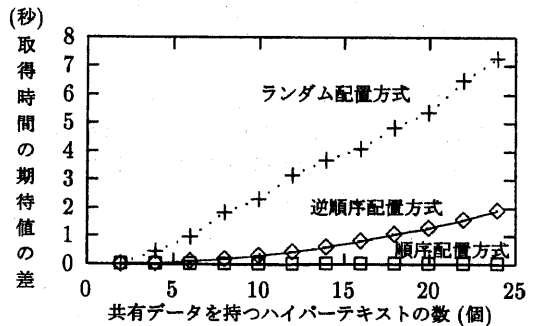


図5: 最適配置方式との取得時間の差

式と逆順序配置方式では、差が m の増加とともに大きくなっている事がわかる。しかし、順序配置方

式では差が 0.03 % 以下におさまっており、良質な準最適解を得ている。また、逆順序配置方式と順序配置方式を比較すると順序配置方式の差の方が小さいため、方針 2 をとることにより ε の値を小さくしていることがわかる。ランダム配置方式は逆順序配置方式以上に順序配置方式からのずれが大きく、方針 2 と合わせて方針 1 をとることにより ε の値を大幅に改善していることがわかる。

4.2 従来方式と提案方式の比較

図 6 は、共有データを持つハイパーテキストの数 m と従来方式、ランダム配置方式、順序配置方式の各 ε の値の関係を示している。従来方式は

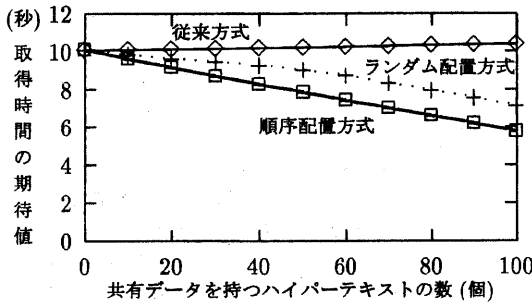


図 6: 各方式の取得時間の比較

ファイルの共有を考慮しない場合のデータ伝送方式なので m の値が変化しても ε の値は変化しない。一方、ランダム配置方式と順序配置方式では m の増加とともに ε の値が減少している。また、ランダム配置方式より順序配置方式の方が取得時間短縮の効果は約 2 倍である。図 6 の結果は、単に共有データを 1 つにまとめるだけでなく、方針 1・2 を適用することで効果を 2 倍程度にできることを示している。

次に、 γ の値を $\frac{1}{5}\lambda$, $\frac{1}{2}\lambda$, $\frac{4}{5}\lambda$ と変えたときの、順序配置方式の ε の値を図 7 に示す。図 7 から γ の値が大きいほど順序配置方式の時間短縮の効果は大きくなることがわかる。

5 まとめ

本論文では放送によるハイパーテキストの伝送方式について述べた。ファイルの共有がある場合に、利用者の待ち時間を改善するため、ファイルの共有関係を考慮した放送データ構成方式‘順序配列方式’を提案した。机上計算の結果、ファイルの

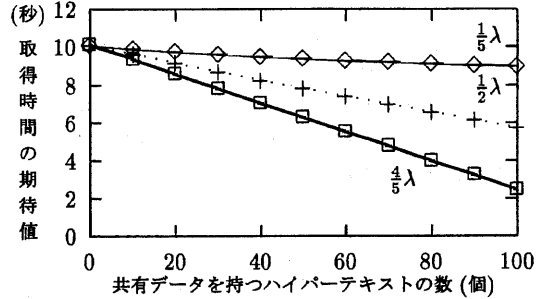


図 7: 共有データのサイズと取得時間の関係

共有関係がある場合、従来方式よりも、順序配列方式のほうがパッケージ取得のための待ち時間の期待値が小さくなることを示した。また、その効果は、ファイルを共有するハイパーテキストの数および共有されるファイルのサイズにほぼ比例して大きくなることが明らかになった。

今後は、キャッシュの利用や放送頻度最適化方式など、他の有力な方式との組合せを考慮し、より一層の時間短縮を目指す予定である。

参考文献

- [1] 箱守 他: 放送型通信 / 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 マルチメディア、分散・協調とモバイルワークショップ, Jul. 1997
- [2] 石川 他: 放送型情報提供システムにおけるハイパーテキストの伝送方式, 第 5 4 回情報処理学会全国大会, Vol.3, pp.593-594, 1997.
- [3] S.Acharya, et al.: "Dissemination-based Data Delivery Using Broadcast Disks," IEEE Personal Communication, Vol. 2, No. 6, Dec. 1995.
- [4] T.Imielinski, et al.: "Adaptive Wireless Information Systems," Proc. of SIGDBS Conf., pp. 19-41, Tokyo, Japan, Oct. 1994.
- [5] B.C.Housel, et al.: "WebExpress: A System for Optimizing Web Browsing in a Wireless Environment," Proc. of the Second Annual Int. Conf. On Mobile Computing an Networking, pp.108-116, 1996.
- [6] Knuth, D.: "The Art of Computer Programming", Vol.3, Addison Wesley, 1981.