

二重循環法によるデータ放送スケジュール

青野正宏[†] 渡辺尚[‡] 水野忠則[‡]

データ放送のスケジュール決定において、考慮すべき事項に次の3つがある。第1は受信者が受信を開始してから、目的のデータを受信するまでの待ち時間をなるべく短くすること、第2は目的とする情報が更新されたとき、放送するまでの遅延時間をできる限り短くすること、第3に目的とする情報がいつ放送されるかを受信者が予測できること。つまり周期が固定であることが望ましい。待ち時間と遅延時間を減少させるためには、受信者の目的データ受信率とデータ更新率に応じて放送することが考えられる。そのとき、どのような周期を与えれば良いかを考察する。さらに二重循環法と呼ぶ周期の一定性を保ち、スケジュール決定を容易にするアルゴリズムについて、提案する。

Data Broadcast Scheduling by Double Cyclic Sequencing Method

Masahiro Aono[†] Takashi Watanabe[‡] and Tadanori Mizuno[‡]

There are three items which it should consider about a schedule plan of data broadcast. The 1st is to make the waiting time as short as possible until receivers accept purposed data after they begin to listen. The 2nd is to make delay time as short as possible to broadcasting from the time when data was updated. The 3rd is that receivers can estimate when purpose data is broadcast. In other words, it is desirable that periods are fixed. To decrease waiting time and delay time, data are broadcast according to the rate of data reception and the rate of data update. We consider what periods we should give. Moreover, we propose an algorithm which keeps fixed period and make easy to decide the schedule. We called it Double Cyclic Sequencing method.

1. 概説

モバイル環境において端末がデータを得る有力な手段のひとつにデータ放送がある。無線の覆域範囲内であれば、放送の受信者（以下、リスナーと表現する。）の数と帯域とは直接の関係がなく、受信者が多ければ多いほど帯域を有効活用していることになる。

放送サーバは定期的に繰り返して、または必要に応じて不定期にデータを放送する。リスナーは自分に必要なデータのみを聴取して、自分自身のローカルなファイルに収納して使用する。いわば空中にデータベースを持つ考え方である^{1,2,3)}。

このようなデータ放送において放送スケジュールを検討するとき考慮すべき事項として、ひとつはリスナーが聴取を開始してから、目的とするデータを受信するまでの平均待ち時間である。もう一つは、聴取を続けて更新情報を得る場合に、その遅延時間である。定期的に放送を行う場合、その時間は周期に比例する

が、聴取率にばらつきがある場合、サーバは聴取率が高い情報を大きい頻度で放送し、聴取率が低い情報を小さい頻度で放送すれば全体としての待ち時間や遅延時間が少なくなる。理想的な放送の配分は待ち時間の場合は、後述するように従来の研究から明らかにされている。ただし、理想的配分を実現するためには、出力候補を決定するための演算負荷が大きい。

また、遅延時間減少の手段についてはあまり明確にされていない。この点について考察する。リスナーは自分が興味あるデータを常時受信しているものとする。放送されるデータは時々更新される。この場合、サーバが更新されたデータをできるだけ早く放送することが、リスナーにとって早く情報を得られることになる。更新データが放送されないうちに、それに関する新しい更新データがサーバに入力されれば、サーバは古い更新データは捨てる。しかし、評価としては、元のは古い更新データをリスナーに知らせることができなかったため、データ放送後それに関する更新データが最初にサーバに入力されてから、最新の更新データが出力されるまでの時間を遅延時間とする。この条件で遅延時間を小さくする放送スケジュールを考察する。このとき、データ更新の都度、更新されたデ

[†]三菱電機
Mitsubishi Electric Corp.
[‡]静岡大学
Shizuoka University

ータを放送する方式の場合、リスナーにとっていつ何が放送されるか予想できないことになる。リスナーにとって、バッテリーを節約するためには必要な情報が放送される時のみ、端末を動かし、その他の時間帯は休眠状態とするのが望ましい。それにはリスナーが求めるデータの出力時期を予測できるよう安定した周期で放送されることが望ましい。前述のデータの理想的配分を行った場合も、周期が一定にならないことが一般的である。安定した周期でかつ待ち時間や遅延時間を最小に近づける手法について検討を行う。

その手法として、データの聴取率分布とデータの更新率に対して、どのような放送の配分を行えば良いかについて考察する。図1に待ちと遅延の概念を示す。

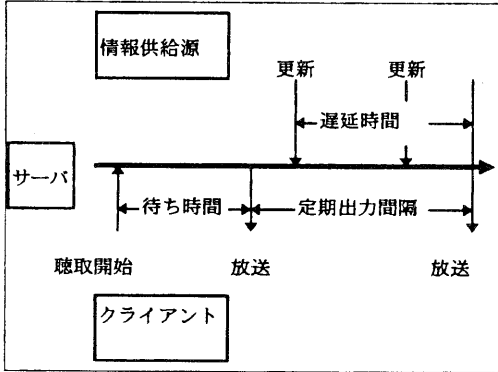


図1 待ち時間と遅延時間
Fig.1 waiting time and latency time

2. 関連研究

繰り返し出力されるデータをデータベースとみなす考え方は⁴⁾などのDataCycleに嚆矢として見られるが、単一周期の基礎的なスケジュールについて述べているのみである。Imielinskiらは^{1,2,5,6,7)}においてモバイル端末においてバッテリー節約のため、聴取回数を少なくするスケジュール方式について論じている。それは最初にデータを得るまでの聴取回数抑制が目的であり、インデックスを付す方法について述べている。本稿におけるバッテリー節約は、1度聴取した後に、データ更新に備えて聴取し続ける場合に焦点をおいた。Zdonikらは^{8,9,10)}において、Broadcast Disksという概念を打ち出し、目的とするデータまでの待ち時間を少なくする手法について論じている。聴取率の高い情報は小さい周期で、低い情報は大きい周期で放送すること、各情報の周期は一定とする概念を打ち出している。しかし、放送周期をどのように決定し、配分する

かという点については触れられていない。聴取率の分布に対する放送の配分率は、^{11,12,13)}などで研究されている。特にVaidyaらは、¹²⁾においてデータ長が可変長の場合にまで拡張し、可変長における最適スケジュールと、演算負荷を軽減するbucket法と呼ぶ手法を提案している。しかし、この手法は周期が安定していない。我々は、周期を安定させ、演算負荷を軽減しながら、待ち時間を低く抑える一般的手法について提案している。また、^{13,14)}においてデータの更新について研究がなされているが、データの更新率と遅延時間の関係についてなされたものではない。

3. 聴取率と更新率に対する放送配分の比

放送の単位を一定の大きさに固定化し、この1単位を1ページと称することとし、1ページを放送する時間をスロットと呼ぶこととする。また、ひとつのデータの単位をファイルと呼ぶこととする。ファイルがm個存在するものとし、ファイル番号を $i=1, 2, 3, \dots, m$ とする。ここで各ファイルの聴取率の分布を p_i ($0 < p_i \leq 1$) とする。また、各ファイルが1スロットの間に更新される確率を q_i ($0 \leq q_i \leq 1$) とする。まず、ファイルを固定長と仮定(1ファイル=1ページ)し、 p_i と q_i から、望ましいファイルiの待ち時間を最小とする放送出力配分比 w_i と遅延時間を最小とする放送配分比 r_i を求める。待ち時間を求める場合は、データの更新率は無関係である。以下の形で定式化する。

$$w_i = \frac{p_i^\alpha}{\sum_{j=1}^m p_j^\alpha} \cdot \nabla_i \dots \dots \dots (1)$$

全体としての平均待ち時間は、次の式で得られる。

$$WAITTIME = \frac{\sum_{j=1}^m \frac{p_j^\alpha}{w_j}}{2 \sum_{j=1}^m p_j} \dots \dots \dots (2)$$

$\alpha=0$ の場合各 w_i は全て等しくなり全てのファイルを順次繰り返し放送することとなる。 $\alpha=1$ の場合は聴取頻度の割合に応じて放送することになる。 $\alpha=0.5$ の場合は、聴取頻度の平方根の割合に応じて放送することになる。 $\alpha=0$ とすると実行方式は単純であり、平均待ち時間は周期の半分となる。平等ではあるが、効率的ではない。 $\alpha=1$ とすると、直感的には公平そうであるが聴取頻度が小さいファイルの待ち時間が極めて大きくなり、全体としての待ち時間が増える。 $\alpha=0.5$ とするのが、最も全体としての待ち時間を減少させることは、既存の研究から明らかにされて

いる。

次に、ファイルが更新される場合について、以下の式で r_i を求めるものとする。

$$r_i = \frac{p_i^\alpha \cdot q_i^\beta}{\sum_{j=1}^m p_j^\alpha \cdot q_j^\beta} \nabla_i \quad \dots\dots(3)$$

ここで、 $d_i = 1/r_i$ とおくと、 d_i は i の放送周期となる。 i のデータ更新から出力までの時間は、次の式で与えられる。

$$\begin{aligned} LATENCY_i &= \sum_{k=0}^{d_i-1} ((d_i - k) \cdot q_i (1 - q_i)^k) \quad \dots\dots(4) \\ &= d_i + 1 - \frac{1 - (1 - q_i)^{d_i+1}}{q_i} \end{aligned}$$

ここで

$$LATENCY = \frac{\sum_{j=1}^m p_j \cdot LATENCY_j}{\sum_{j=1}^m p_j} \quad \dots\dots(5)$$

を最小とする α, β を求める。これは解析的に求めるのは困難であり、聴取率と更新率の分布によっても値が変わってくる。ここでは、いくつかのケースを試算する。想定する分布は次の Zip 分布と呼ばれる分布に従うものとする。

$$p_i = \frac{(1/i^\theta)}{\sum_{j=1}^m (1/j^\theta)} \quad \dots\dots(6)$$

q_i も同様とする。ここで、 $\theta = 0 \sim 1.5$ に変化させるものとする。 $\theta = 0$ は分布が平坦であることを示す。 θ の値が大きくなるほど、聴取率の歪みは大きくなる。その他、参考として等差分布、正規分布 ($\sigma = 0 \sim 3$) のパターンを検討する。図 2 に、 $m=20$ の場合について分布例を示す。

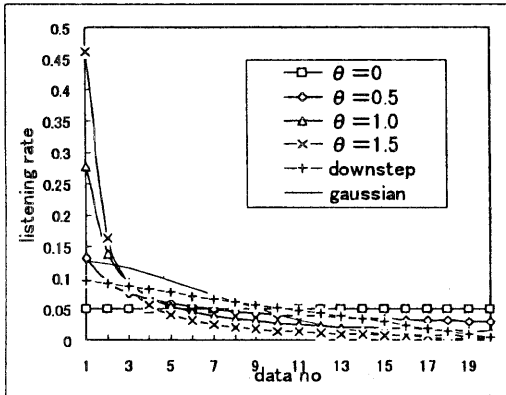


図 2. 聴取率分布
Fig.2 Distribution of Listening Rate

この聴取率の分布ごとにファイル更新確率に対し

てどのように遅延を最小とする配分率係数 α, β が変化するかを探索法により求める。図 3 は、各ファイルの更新確率が等しくし、聴取の分布にバラツキを持たせた場合である。分布の形状により、変換指数最適値の差異があるが、ファイルの更新率が高ければ、遅延時間は更新間隔に近づくので $\alpha = 0.5$ となる。ファイルの更新率が低ければ、より均等に近づく。しかし、更新率が低くてもある限度で収束している。

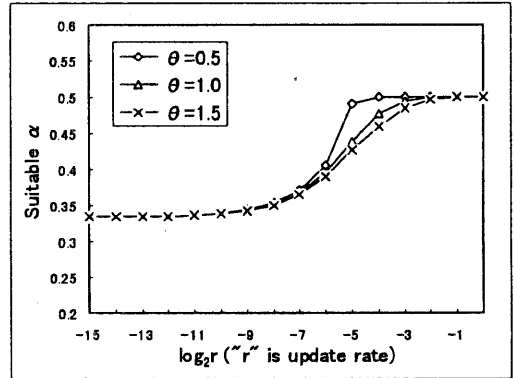


図 3 最適 α 値 (更新率一定: 100 ファイル)
Fig.3. Suitable α (Update Rate is constant, 100files)

図 4 は、聴取の分布を一定として、ファイルの更新率にバラツキを持たせた場合の図の例である。更新率が高い場合は、 $\beta = 0$ となり、全て平等に出力すれば良い。しかし、更新率が下がれば、ある程度、更新率を勘案した配分とすれば遅延時間は最適となる。

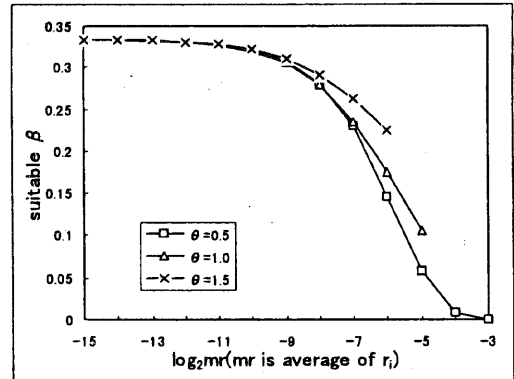


図 4. 最適 β 値 (聴取率一定: ファイル数=100)
Fig.4. Suitable β (Listening rate is constant, 100files)

各ファイルが可変長の場合の聴取率は各ページ単位で考える。各ページの聴取率の合計がファイルの聴取率として放送出力配分の基準となるので、ページの

聴取率としては、ファイルの聴取率からページ数を除した値とすれば、最適の値が得られる。

各ファイルが可変長の場合のデータの更新率に関しては、変更のあったページのみが放送されるというように捉えれば、ファイルが固定長の場合に還元できる。

4. スケジュールの設定

各ファイルの出力配分比率を定めた後、なるべく周期出力配分に近づける具体的なファイルの出力の決定は、次の方法を用いる。

各ファイルはポイントを持つ。初期値は全てゼロとする。出力ファイルはポイントが最も低いファイルを選ぶ。(同点の場合は任意) 選ばれたファイルを出力するが、このときポイントに当該周期 c_i を加算する。ファイルはポイント順のチェーンを作っておく。適当に順位の間中間にインデックスを設けておき、加算後のファイルの順位の再設定を行う。この方法により、ほぼ与えられた配分比に近似した割合で出力することができる。この方法を周期個別選択法 (IDS: Individual Selecting Method) と呼ぶこととする。

ファイル数が少ない場合は IDS 法の適用は容易であるが、ファイル数が多くなると、この方法ではインデックスのメンテナンスなどで実行時の計算負荷が大きくなる。また、サーバでは計算可能であるとしても、リスナーにとって自己が必要とするファイルを得る時刻を正確に予測することが困難である。サーバから放送するファイルの情報に次回のそのファイルの放送予定時刻を含めて放送するという手法も考えられるが、リスナーにとって受信ミスが発生した場合は判らなくなってしまう。

近似した配分率のファイルの配分率を同じとみなすことにより、全体としての待ち時間や遅延時間の増加を抑えつつ、選択を簡単にすることができる。¹²⁾では、Bucket 法と名づけた手法を提案している。放送ファイルをいくつかのグループ (その数を numgrp とする。) に分け、各グループ内では順次放送することにより、出力ファイル選択の判断対象を絞るものである。グループ分けの手法は次のとおりである。 $w_1 \geq w_2 \geq \dots \geq w_m$ とし、 $def = (w_1 - w_m) / numgrp$ とする。 w_i が $w_1 - (k-1) * def \geq w_i > w_1 - k * def$ ($k = 0 \sim numgrp$ の整数) の場合、グループ k に属するとする。また、 w_m はグループ numgrp に属するとする。各グループの平均聴取頻度を p_b とする。各グループから候補のファイルをひとつづつ選択し、 $p_b \times t_i^2$ が最大の

ファイルを選択する。選択されたファイルはそのグループ内で次の出力候補としては最後尾に回される。

この方法で出力候補を選択する計算は簡略化できる。しかし、まだ毎回の出力判断が必要という問題がある。このため、我々はより簡易な手法として、二重循環法 (DCS: Double Cyclic Sequencing Method) と名づけた手法を提案する。この手法は以下のとおりである。放送ファイルをいくつかのグループに分ける点は Bucket 法と同じであるが、各グループに属するファイルの w_i の和がなるべく等しくなるようグループ分けを行う。図 5 に 2 グループに分割した場合の、Bucket 法と DCS 法の概念の差を示す。Bucket 法は w_i の最高と最低の差で等分しているのに対し、DCS 法は w_i の面積で等分している。

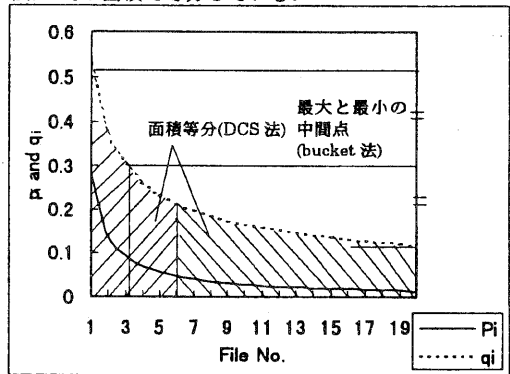


図 5 Bucket 法と DCS 法の相違
Fig.5 Difference between bucket and DCS method

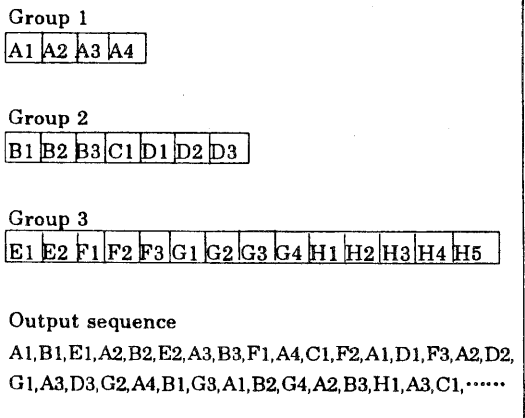


図 6 DCS 法出力例
Fig. 6 Example of output sequence by DCS method

以下にロジックを示す。gid_{1-numgrp} は各グループの先頭ファイルを指すものである。i, j, sum は計算のた

めの一時的使用値である。

step1 : $gid_i=1, j=2, sum=0$ とする。

step2 : $i=1$ とし, j が numgrp を越えなければ STEP3 ~6 を実行し step2 に戻る。

step3 : sum に w_j を加算する。

step4 : sum が $(j-1)/numgrp$ より小であれば step5 を省略する。

step5 : gid_j を $i+1$ とする。 j をカウントアップする。

step6 : i をカウントアップする。

各グループ内の出力順位は循環的に固定とする。また、グループ選択も各グループ間で循環させる。つまりグループ1の1番めのファイル、2の1番め...1の2番めという順に選択していく。(図6参照)

この手法は、放送するファイルを放送頻度別にグループ分けする点と各ファイルの放送間隔を固定とする点で⁹⁾における Multidisk と類似しているが、次の点で multidisk の手法を発展させている。ひとつは、⁹⁾ではグループ分けの基準について明示していないが、本手法では、その基準を定めたことである。もう一つは、Multidisk では、どのグループをどの順序で出力するかを明らかにしていないが、本手法は必ず1スロットごとに放送するファイルのグループを順次移していくというルールを明確化している。

この手法はファイルが可変長の場合、一つのファイルの各ページを連続して出力するのではなく、間を置いて出力することを意味している。周期のグループ数が numc、可変長ファイルが nump ページあるとすると、全ページ聴取には、 $numc \times (nump-1) + 1$ 単位時間が必要とし、連続聴取の nump 単位時間の場合よりも、 $(numc-1) \times (nump-1)$ 単位時間だけ、余計に時間がかかる。これは DCS 法の方法の短所である。しかし、その代わりに各ファイルの各ページ単位で周期が安定する。さらに、更新データを受信する場合には、変更のあったページだけを取り込むという観点からは、必ずしもファイルの全部を読む必要もない。ただ、本稿ではリスナーがファイル受信を欲してから、またはデータの更新が発生してから、聴取するファイルの最後を受信するまでを待ち時間または遅延時間として評価の対象とする。

bucket 法では、可変長ファイルをひとつの出力単位として扱っており、途中で分割することはしない。その代わり周期が不安定となる。ひとつのファイルのなかに他のファイルのページが割り込むことは、必ずしも全ての場合に不利に働くわけではない。簡単な例を挙げる。待ち時間を評価対象とし、A、Bの2種類のファイルがあるとすると。Aは長さ4ページでBは1

ページである。Bの聴取頻度はAの4倍とする。AとBの出力比は $(1/4)^{1/2} : (4/1)^{1/2} = 1 : 4$ となる。このとき、2つの出力スケジュールを考える。図7に示すように直列方式と交互方式である。直列方式の場合、Aの待ちは、平均7で、Bの待ちは平均1.75である。交互方式の場合Aの待ちは平均11でBの待ちは平均1である。加重平均をとると、待ちは直列方式が2.8で交互方式が3.0となり、直列方式が有利である。しかし、同様にAが長さ8ページでBは1ページとし、Bの聴取頻度はAの8倍とした場合を計算すると、直列方式の待ちは3.7で交互方式の待ちは3.4となり、交互方式が有利となる。この点について次の章で評価するものとする。

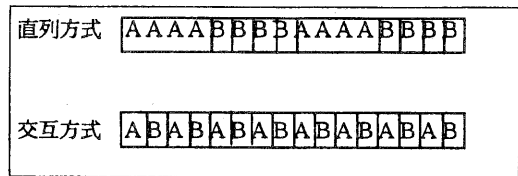


図7 不定長ファイル直列方式と交互方式

Fig.7 Series and Interlace Arrangement for variable length files.

5. 方式評価

以下に、いくつかの場合を想定して望ましい放送スケジュールを検討し、確率計算もしくはシミュレーションにより確認を行う。ファイルの長さについては、すべて1ページの固定長とした平坦型、聴取頻度が高いファイルのページ数を少なく(最小1ページ)、聴取頻度の高いファイルのページ数を多く(最大10ページ)して等差的に配分した増加型とその逆である減少型、各ファイルの長さを1~10ページの一様乱数で与えたランダム型の4パターンを準備した。

表1 パラメータの定義

Table 1 parameter description

パラメータ	定義
スケジュール手法	FLAT法, IDS法, bucket法, DCS法
聴取率の分布	Zipf分布 ($\theta=0, 0.2, 0.5, 1.0, 1.5$), 等差分布, 正規分布
ファイル長分布	平坦型, 増加型, 減少型, ランダム型
ファイル長	平坦型=1固定 その他平均5.5 特定の試算ランダム型で2~100
放送ファイル数	numdata=100, 1000
グループの数	numgrp=2~20 (デフォルト値は5とする。)
配分パラメータ	べき乗変換係数: $\alpha=0.5\sim 1$ (デフォルト値は0.5とする。)
更新ファイル密度	numdent=1~100%

その他のパラメータとして、放送ファイルの数 = numdata, Bucket 法, IDS 法で採用したファイルのグループの数 = numgrp, ファイルの分布密度から遅延時間係数を求めるべき乗パラメータ = α , 更新ファイルの発生密度を使用する。表 1 にパラメータの条件を示す。

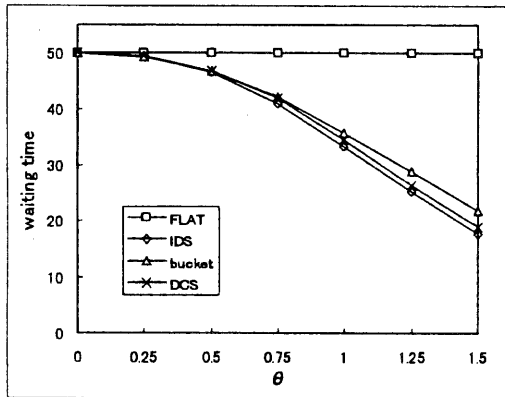


図8 各手法を用いた場合の待ち時間 (100 ファイル)
Fig.8 Waiting time with each method (100 files)

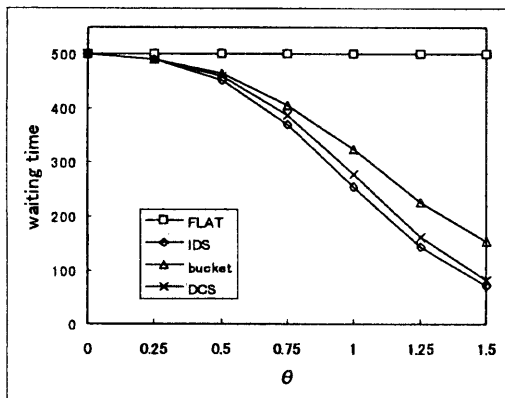


図9 各手法を用いた場合の待ち時間 (1000 ファイル)
Fig.9 Waiting time with each method (1000 files)

図 8 と 9 は、ファイル長 = 固定長, numgrp = 5, $\alpha = 0.5$ とし, numdata = 100, 1000 の両ケースと, FLAT 法, IDS 法, bucket 法, DCS 法の 4 手法と, $\theta = 0 \sim 1.5$ とした場合の Zipf 分布の組み合わせについて計算した結果である。計算の結果, ① Zipf 分布のひずみが大きいほど平均待ち時間が減少する。② IDS 法が最も効果的であるが, 一般に DCS 法が bucket 法より効果的であり, IDS 法との大きな差異がないので実用可能な手法である。③ ファイル数が多いとき, DCS 法が IDS 法との差異が小さいのに対し, bucket 法は差

異が大きくなる。とくに分布のひずみが大きいときに顕著である。

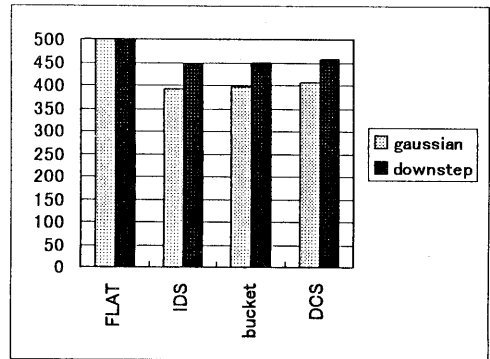


図 10 正規分布と等差分布の場合の待ち時間
Fig.10 Waiting time at Distribution of gaussian and down step

特殊な分布として, 等差分布と正規分布の場合について, FLAT 法, IDS 法, bucket 法, DCS 法の 4 手法について, ファイル長 = 固定長, numgrp = 5, $\alpha = 0.5$ とし, numdata = 1000 について平均待ち時間を計算した結果について図 10 に示す。等差分布, 正規分布ともにこの場合は bucket 法が DCS 法より多少良い結果を示している。しかし, その差はほとんどない。

グループの数を変化させた場合の待ち時間の変化を示す。 $\alpha = 0.5$, Zipf 分布係数 $\theta = 1.5$, ファイル数 numdata = 1000 として, bucket 法と DCS 法について, numgrp = 2 ~ 20 に変化させた場合の例を図 11 に示す。

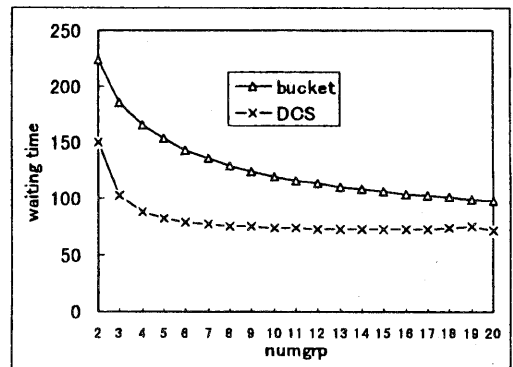


図 11 グループ数による待ち時間
fig.11 Waiting time according to number of groups

グループの数を増やせば待ち時間は漸減する。しかし, DCS 法ではグループの数を増やすと飽和状態に達するが, bucket 法は漸減を続けている。これは,

DCS 法は機械的に割り当てを循環させており、グループの数を増やし過ぎてもグループ間の配分のアンバランスが増大するのに対し、bucket 法は各グループの比較を毎回行うので、グループの数を増やせば IDS 法に近づいて行くためである。上記の例では、DCS 法でせいぜい 6~7 グループで十分であるといえる。

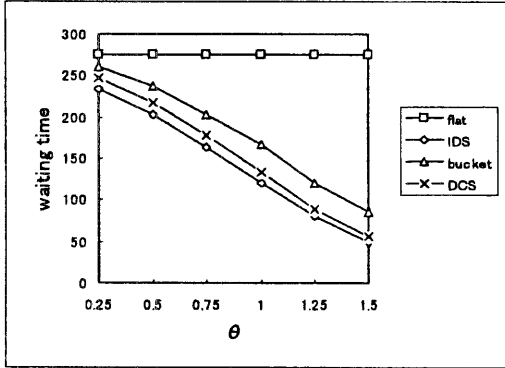


図 12 ファイル長増加型待ち時間
Fig.12 Waiting time with distribution of increased type file length

的に DCS 法より待ち時間が大きくなっている。減少型分布モデルは聴取率が高いファイルの長さが大きくなっており、ファイル受信途中で待ちのロスが生ずる DCS 法に不利に働くからである。増加型では逆に DCS 法に有利に作用する。平均的なランダム型分布モデルでも bucket 法より DCS 法が良い結果を示している。

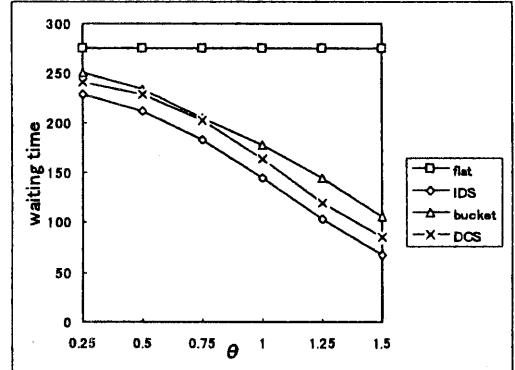


図 14 ファイル長ランダム型待ち時間
Fig.14 Waiting time with distribution of random type file length

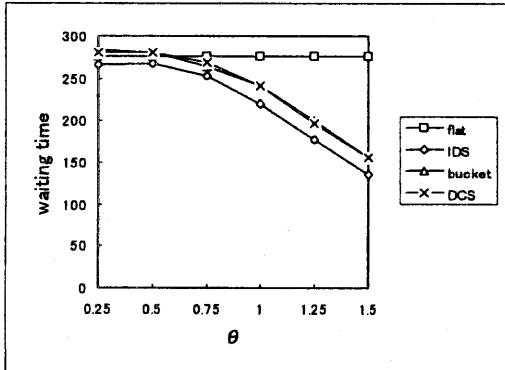


図 13 ファイル長減少型待ち時間
Fig.13 Waiting time with distribution of decreased type file length

ファイル長が可変の場合について計算する。図 12 は、ファイル長を増加型、numgrp=5、 $\alpha=0.5$ とした場合、聴取率の各分布、numdata=100、FLAT 法、IDS 法、bucket 法、DCS 法の 4 手法について示したものである。図 13 は、ファイル長を減少型について同様に、図 14 はファイル長をランダム型にして同様に示したものである。

これより、IDS 法の待ち時間をもっとも小さい。DCS 法は IDS 法より待ち時間は大きいとその増加分はそれほど大きくない。bucket 法は長さが減少型分布モデルで DCS 法より優れた場合も存在するが、全体

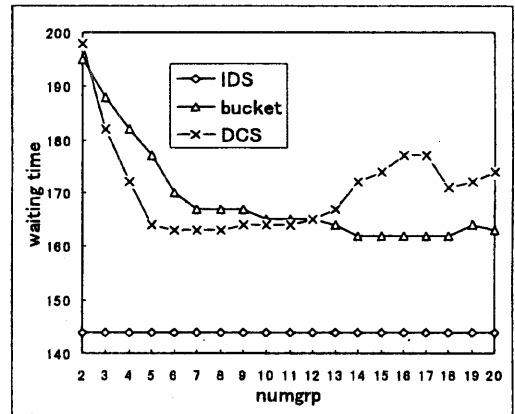


図 15 グループ数による待ち時間分布
Fig.15 Distribution of Waiting time according to number of groups.

しかし、グループの数が多くなると、DCS 法ではファイルの最初から最後まで出力する間隔が長くなる。図 15 は、ファイル長をランダム型、聴取率の分布を $\theta=1.0$ 、 $\alpha=0.5$ 、numdata=100 とし、numgrp = 2~20 まで変化させた場合の IDS 法、bucket 法、DCS 法の 3 手法について示したものである。(IDS 法はグループとは無関係であるが、理想値を示すため参考として示してある。)

図 11 の固定長の場合でファイル数が 1000 の場合

と比較して、グループ数による特徴がより明確に表れてくる。グループ数を増やすと、DCS 法はファイル中のページを間欠的にしか出力しないため、待ち時間はある数から逆に増加する。bucket 法はそのような影響はない。しかし、グループの数を少々増やしてもグループの数が少ない、DCS 法と大差ない。

表2はファイル長をランダム型、聴取率分布を $\theta=1.0$ 、numdata=100、numgrp=5、ページ数を1、10、100の各々について、IDS 法、bucket 法、DCS 法の3手法について示したものである。ページ数が多くてもDCS 法は一般にbucket 法より効率が良いが、理想値との乖離は大きくなる。ファイル中のページを間欠的にしか出力しないため、待ち時間は相対的に不利になる。ページの単位を細かく定めると帯域上のロスは少なくなるが、ページ数が増え、待ち時間上では不利益となるので、帯域の効率と待ち時間との関係はトレードオフとなる。

表2 ファイル長と待ち時間
Table 2 File length vs. Waiting time

平均ファイル長	IDS	bucket	DCS
1 (固定長)	33.3	35.8	34.5
10 (ランダム)	274	363	326
100 (ランダム)	2386	3341	3045

6. まとめ

本論では、データ放送において、リスナーが目的とするファイルを得るまでの時間を最短にすることに加えて、ファイルが更新されたときから、放送されるまでの時間を短くするためのデータ放送のファイルの出力配分比について述べた。次いで、安定した周期で望ましい配分比に近く、かつスケジュール決定が容易となる二重循環法について提案を行った。特に可変長のファイルでもページ単位に区切ってスケジュールすることにより、定期性を保っている。また、先行研究で提案されている手法と比べても、待ち時間/遅延時間の減少に一般に効果的であることを示した。

参考文献

- 1) T. Imielinski, B. R. Badrinath, "Efficient Filtering of Data on Air", Proc. of 4th International Conference on EDBT (Extending DataBase Technology), Cambridge- U. K. pp248-258, March 1994
- 2) T. Imielinski, S. Viswanathan, B. R. Badrinath, "ENERGY EFFICIENT INDEXING ON AIR", Proc.

- of ACM-SIGMOD, International Conference on Data Management, Minnesota, pp25-36, May 1994
- 3) Stan Zdonik, Michael Franklin, Rafael Alonso, Swarup Acharya, "Are Disks in the Air Just Pie in the Sky?" IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA. December 1994
- 4) T.F.Bowen, G.Gopal, G.Herman, T.Hickey, K.C. Lee, W. H. Mansfield, J.Railz, A. Weinrib, "The Datacycle Architecture", Communications of ACM, Vol.35, No12, pp71-81, December 1992
- 5) T. Imielinski, S. Viswanathan, B. R. Badrinath, "DATA ON AIR: ORGANIZATION AND ACCESS", IEEE Transactions in Data and Knowledge Engineering, 1995
- 6) T. Imielinski, S. Viswanathan, "WIRELESS PUBLISHING: ISSUES AND SOLUTIONS", MOBILE COMPUTING KLUWER ACADEMIC PUBLISHERS pp299-329 1996
- 7) Narayanan Shivakumar, Suresh Venkatsubramanian, "Energy-Efficient Indexing For Information Dissemination In Wireless System", ACM-Baltzer. Journal of Mobile Network and Nomadic Applications December 1996 pp.433-446
- 8) Swarup Acharya, Michael Franklin, Stan Zdonik, "Dissemination-based Data Delivery Using broadcast Disks", IEEE Personal communications Vol12. No6. December 1995
- 9) Swarup Acharya, Michael Franklin, Stan Zdonik, "Prefetching from a Broadcast Disks", 12th International Conference on Data Engineering (ICDE 96), New Orleans, LA, February, 1996
- 10) Michael Franklin, Stan Zdonik, "Dissemination-Based Information Systems", IEEE Data Engineering Bulletin, Vol.19, No.3, September 1996
- 11) T.Imielinski, S.Viswanathan, "Adaptive Wireless Information System", Proc. of SIGDBS(Special Interest Group in DataBase Systems) Conference, pp.19-41, Tokyo Japan, October 1994
- 12) Nitin H. Vaidya, Sohail Hameed, "Scheduling Data Broadcast in Asymmetric Communication", Proc. of Workshop on Satellite-based Information Services(WOSBIS), New York November 1996
- 13) Chi-Jiun, Leandros Tassioulas, "Broadcast Scheduling for Information Distribution", IEEE INFOCOM Kobe Japan 1997
- 14) Daniel Barbara, Tomasz Imielinski, "Sleepers and Workholic: Caching Strategies in Mobile Environments", Proc. ACM SIGMOD Conf. Minneapolis, MN, May, 1994
- 15) Swarup Acharya, Michael Franklin, Stan Zdonik, "Disseminating Updates on Broadcast Disks", Source: 22nd International Conference on Very Large Data Bases (VLDB96) Bombay India, 1996