

アドホックネットワークにおけるマルチキャストを用いた データ転送のための転送開始時刻の決定について

篠原 昌子[†] 原 隆浩[†] 西尾章治郎[†]

[†] 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
〒565-0871 大阪府吹田市山田丘1-5
E-mail: †{sinohara.masako,hara,nishio}@ist.osaka-u.ac.jp

あらまし アドホックネットワークでは、移動体が要求されたデータをユニキャストで転送した場合、頻繁に要求されるデータが何度も転送され、データをもつ移動体や転送経路上の移動体の消費電力が大きくなる。これまでに筆者らは、データの可用性を損なわず、トラフィックの削減により移動体の生存時間を長くすることを目的として、マルチキャストを用いたデータ転送方式を提案した。しかし従来方式では、デッドラインが長い場合、トポロジ変化によって転送経路が無効になる機会が多くなり、データの可用性が低下してしまう。そこで本稿では、従来方式を拡張し、デッドラインが長い環境においてもデータの可用性を維持するデータ転送方式を提案する。拡張方式では、データ要求が転送された経路の有効時間を考慮して、その経路上の移動体による要求のデータ転送開始時刻を決定する。

キーワード アドホックネットワーク, マルチキャスト, データ転送開始時刻

On Determination of Starting Time for Data Transmission using Multicast in Ad Hoc Networks

Masako SHINOHARA[†], Takahiro HARA[†], and Shojiro NISHIO[†]

[†] Dept. of Multimedia Eng., Grad. Sch. of Information Science and Technology, Osaka Univ.
1-5 Yamadaoka, Suita-shi, Osaka, 565-0871, Japan
E-mail: †{sinohara.masako,hara,nishio}@ist.osaka-u.ac.jp

Abstract In ad hoc networks, if mobile hosts transmit requested data items by unicasting, mobile hosts that hold or relay data items frequently requested, have to transmit them many times and consume a large amount of power. In our previous work, we have proposed a data transmission method for not only maintaining data availability but also prolonging the lifetime of mobile hosts by reducing the traffic. However, when the deadline for a data request is very long, our previous method does not work well because the transmission path becomes invalid due to the topology change. In this paper, we extend our previous method to maintain data availability of requests with long deadline. In the extended method, when a mobile host receives a data request, the host estimates the effective duration of the transmission path along which the request is transmitted, and then, determines the starting time of other requests from mobile hosts that exist along the transmission path.

Key words ad hoc networks, multicast, starting time of data transmission

1. はじめに

近年、無線通信技術の発展と計算機の小型化に伴い、ルータ機能をもつ移動体のみで一時的なネットワークを形成するアドホックネットワークへの関心が高まっている。アドホックネットワークでは、既存の通信インフラを必要とせずに、移動体同士で自律分散的にネットワークを構築できるため、災害時の救助活動における救助隊員間のデータ共有やイベント会場での情

報共有への利用が期待されている。このようなアプリケーションでは、移動体同士でデータを共有し、互いのもつデータにアクセスするものも多いため [3], [7], データ要求の効率的な処理が重要となる。

ここで、アドホックネットワークでは、移動体の電力容量に制限があり、データ転送による消費電力が全体の大きな割合を占める場合が多い [2]。したがって、データ転送の機会が多い移動体は、他の移動体より消費電力が大きくなり、電力を早く使

い果たしてしまう。電力を使い果たした移動体がネットワークから退出すると、その移動体のもつデータにアクセスできなくなる。さらに、ネットワークから退出した移動体が多くなると、ネットワークが疎になり、移動体が他の移動体と接続できる機会が減るため、データの可用性が低下してしまう。このような問題を改善するため、データの可用性を維持させつつ、移動体の生存時間をできるだけ長くすることが非常に重要となる。

ここでアドホックネットワークにおいて、移動体が要求されたデータをユニキャストを用いて転送した場合、他の移動体から頻繁に要求されるデータが何度も転送され、データをもつ移動体や転送経路上の移動体は、消費電力が大きくなるという問題があった。そこで筆者らは、文献[8]において、データの可用性を損なわず、トラヒックの削減により移動体の生存時間を長くすることを目的として、マルチキャストを用いたデータ転送方式を提案した。この方式では、各移動体は、データの取得時間のデッドラインを設定し、データを要求する。要求されたデータをもつ移動体は、複数の移動体からのデータ要求を一旦保留しておき、後からマルチキャストを用いてまとめて転送する。

しかし、文献[8]の従来方式では、データ転送を開始する時刻を、設定されたデッドラインとデータ転送に要する時間のみに基づいて決定したため、デッドラインが長い場合、トポロジ変化によって転送経路が無効になる機会が多くなり、データの可用性が低下してしまう。そこで本稿では、従来方式を拡張し、デッドラインが長い環境においてもデータの可用性を維持するデータ転送方式を提案する。拡張方式では、デッドラインが長い場合に、データ要求が転送された経路の有効時間を考慮して、その経路上の移動体による要求のデータ転送開始時刻を決定する。

以下では、2.で関連研究について紹介する。3.で想定環境について述べる。4.で文献[8]で提案したデータ転送方式について述べ、5.で本稿で提案する拡張方式について述べる。6.でシミュレーション実験の結果を示す。最後に7.で本稿のまとめと今後の課題について述べる。

2. 関連研究

近年、アドホックネットワークのルーティングに関する研究では、マルチキャストを用いたルーティング方式が数多く提案されている。

文献[6]では、AODV[5]を拡張し、マルチキャストに対応したツリー型のルーティングプロトコル MAODV(Multicast Ad hoc On-Demand Distance Vector Protocol)を提案している。このプロトコルでは、マルチキャストグループごとに共有木を形成し、移動体が新たに参加する場合や、移動体の移動や離脱によってリンクが切断した場合には、木を再構成することで、グループ内の経路を常時維持する。また文献[4]では、マルチキャストグループ内の移動体およびそれらの移動体間の経路上の移動体を一つの転送グループとして扱う、メッシュ型のルーティングプロトコル ODMRP(On-Demand Multicast Routing Protocol)を提案している。このプロトコルでは、移動体が明

示的なメッセージなしにグループから離脱できるように、経路生成やグループ内の移動体の管理をオンデマンドで行うことで、制御トラヒックを削減する。

これらのルーティングプロトコルは、マルチキャスト通信を用いることで、トラヒックを削減できる。しかし、データ転送の前にマルチキャストグループを作成、管理しておく必要がある上、アプリケーションレベルでの効率化は行われていない。

3. 想定環境

本稿では、協調作業における作業の効率化を図るため、数十から百台程度の移動体で構成される中規模なアドホックネットワークを利用して、モバイルユーザが情報共有を行う環境を想定する。例えば、イベントや展示会の会場では、企業などの団体がブースを出展し、来場者に製品やサービスを展示する。この場合、各団体のブースにある計算機は、製品やサービスに関する情報、ブースでのイベントスケジュール、および会場内で配布するファイル(壁紙やムービーなど)をデータとして保持する。また、来場者は自身のもつ小型端末を用いてこれらのデータにアクセスしながら、自身の行動を決定する。このような環境では、各データをユニキャストで転送すると、膨大なトラヒックによって帯域や電力を消費してしまう。

アドホックネットワーク内に、 m 個の移動体(識別子: M_1, M_2, \dots, M_m)が存在し、各々が自由に移動する。また、 n 個のデータ(識別子: D_1, D_2, \dots, D_n)が存在し、各々が特定の移動体にオリジナルデータとして保持されている。簡単化のため、各移動体は、全ての移動体の識別子とそれらが保持しているデータを把握しているものとする。

各移動体は、自身および他の移動体のもつデータにアクセスする。自身がアクセス対象のデータをもつ場合、その要求は即座に成功となる。そうでない場合、データの取得時間のデッドラインを設定し、データを要求する。相互接続している移動体からデータを要求された移動体は、自身のもつデータを送信する。ここで、相互接続している移動体とは、1ホップ以上の無線リンクで相互に通信可能な移動体の集合を指す。簡単化のため、移動体は複製配置を行わないものとする。なお、移動体が複製を配置する環境では、オリジナルデータをもつ移動体と同様に、要求されたデータ(複製)を転送することで対処できる。

ユニキャストを用いたデータ転送には既存の下位層のデータ転送プロトコルを利用する。なお、本稿で提案するデータ転送方式は、利用するデータ転送プロトコルと独立に動作するため、データ転送プロトコルを特に限定しない。

4. データ転送方式

本章では、筆者らが文献[8]で提案したデータ転送方式について説明する。この方式では、移動体は、データ要求した移動体がデッドラインの時間内にデータを取得できるよう、データ転送を開始する時刻を決定する。また、同じデータを要求している複数の移動体にデータをまとめて転送できるよう、データ転送木を作成する。

以下では、まず、データアクセスの処理について説明する。

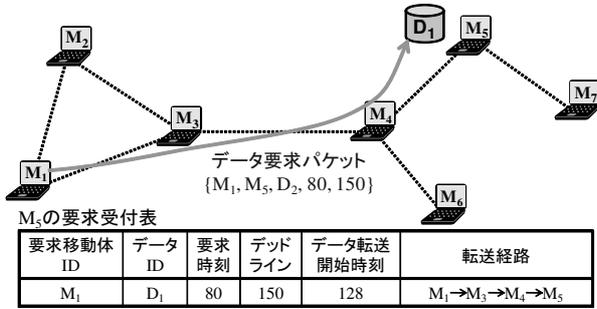


図1 データアクセス
Fig.1 Data access.

次に、移動体が他の移動体から要求されたデータの転送開始時刻を決定する方法、およびデータを複数の移動体にまとめて転送するためのデータ転送木を作成する方法について説明する。最後に、作成したデータ転送木を用いたデータ転送の手順について説明する。

4.1 データアクセス

本節では、移動体が、他の移動体のもつデータにアクセスする際の動作について説明する。

データ要求した移動体：

データの取得時間のデッドラインを設定し、相互接続している移動体にデータ要求パケットをフラッディングする。このパケットには、自身の識別子、データをもつ移動体の識別子、要求したデータの識別子、データを要求した時刻、および設定したデッドラインが含まれる。

デッドラインの時間内にデータを取得できた場合、そのアクセスは成功となる。一方、時間内に取得できなかった場合（データをもつ移動体と相互接続していない場合を含む）、そのアクセスは失敗となる。

データをもつ移動体：

データ要求パケットを受信すると、4.2の方法に従ってデータ転送開始時刻を求める。その後、パケットに含まれる要求した移動体の識別子、データの識別子、要求時刻、デッドライン、および転送経路に加え、求めたデータ転送開始時刻を要求受付表に記録する。なお、データ要求パケットには、このパケットをフラッディングした移動体の識別子が含まれているため、データをもつ移動体はパケットの転送経路を把握できる。

その他の移動体：

データ要求パケットを初めて受信した場合、自身の識別子を追加してフラッディングを続ける。

図1は、時刻80において、移動体M₁がデッドラインを時刻150に設定し、移動体M₅のもつデータD₁を要求する動作を示す。表は、データ要求パケットを送受信した後の、M₅の要求受付表に追加される情報を示す。

4.2 データ転送開始時刻の決定

転送開始までの時間が長いほど、その時間内に同じデータを要求する移動体数は増加すると考えられるため、移動体は、次式を用いて、データ転送開始時刻 T_{st} を求める。

$$T_{st} = T_{dl} - \frac{1}{BW} \times size \times hop - \tau_1. \quad (1)$$

ここで、 T_{dl} はデータ要求パケットに設定されたデッドライン、 $size$ は要求されたデータのサイズ、 hop はデータを要求した移動体までのホップ数を示す。また、 BW は隣接する移動体間で1秒間に転送可能なデータサイズ（平均転送効率）を示し、 τ_1 は事前に設定される正の定数で、移動体がハードディスクアクセスやCPU計算、4.4で用いる制御パケットの送受信、およびトポロジ変化への対処などに用いるための時間を示す。つまり、式(1)の右辺第2項は、移動体がデータを要求した移動体まで転送するのに要する時間の予測値を示し、 T_{st} はデータを要求した移動体のアクセスが成功するために、移動体がデータを転送する最遅開始時刻を示す。

また、式(1)では、データ転送開始時刻が現在時刻より早い場合、移動体はデータを転送しないため、データを要求した移動体のアクセスは失敗となる。一方、デッドラインがデータの転送に要する時間より長い場合、移動体が要求されたデータを即座に転送すれば、データを要求した移動体はデッドラインの時間内にデータを取得できるため、次式を用いて、式(1)を拡張する。

$$T_{st} = T_{now} \quad \text{if } T_{st} < T_{now} < T_{st} + \tau_1. \quad (2)$$

ここで、 T_{now} は現在時刻を示す。

図1を用いて、データ転送開始時刻を決定する例を説明する。データD₁のサイズを1[MB]、 $BW=2[\text{Mbps}]=0.25[\text{MB}/\text{秒}]$ 、 $\tau_1=10[\text{秒}]$ とすると、データ転送開始時刻は次のようになる。

$$T_{st} = 150 - \frac{1}{0.25} \times 1 \times 3 - 10 = 128. \quad (3)$$

4.3 データ転送木の作成

移動体は、あるデータへの要求がデータ転送開始時刻になった場合、自身を根とし、同じデータを要求している複数の移動体を他の節点とするデータ転送木を作成する。データ転送木は、データ転送開始時刻となった要求の転送経路を初期状態とし、以下の手順によって、他の要求の転送経路を統合することで作成する。

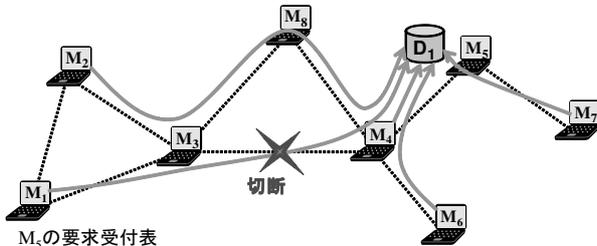
(1) 移動体は、要求受付表から、未選択で要求時刻が最も古い要求を選択する。

(2) 選択した要求の転送経路とデータ転送木の両方に含まれる移動体を探索し、その中でデータ要求した移動体に最も近い移動体をCCH(Closest Common Host)とする。

(3) CCHが自身（データ転送木の根）である場合、選択した要求の転送経路は、データ転送木に含まれる経路との共通部分がない。このような要求は転送経路を統合しても意味がないため、この要求の処理を終了して手順(1)に戻る。

そうでない場合、移動体は、データ要求した移動体からCCHまでの経路をデータ転送木に統合する。

(4) CCHから自身までの経路が、データ転送木と選択した要求の転送経路において異なる場合、それぞれの経路を転送経路に含む要求の要求時刻が遅い方の経路に更新する。これは、移動体の移動によってネットワークトポロジが変化する場合、要求時刻が遅い要求の転送経路の方が、最新のネットワークトポロジを反映している可能性が高いためである。



M5の要求受付表

要求移動体 ID	データ ID	要求時刻	デッドライン	データ転送開始時刻	転送経路
M1	D1	80	150	128	M1→M3→M4→M5
M2	D1	120	170	144	M2→M3→M6→M4→M5
M4	D1	100	190	176	M4→M5
M6	D1	90	180	162	M6→M4→M5
M7	D1	110	160	146	M7→M5

図2 データ転送木の作成

Fig. 2 Creation of a data transmission tree.

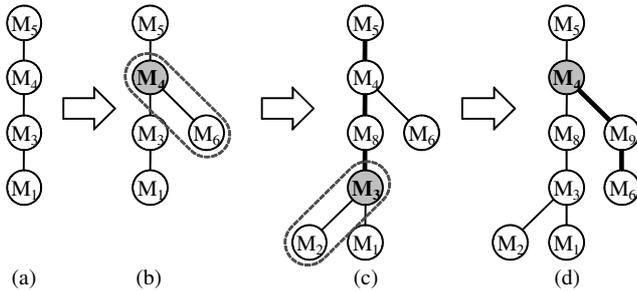


図3 データ転送木の更新

Fig. 3 Update of the data transmission tree.

図2と図3を用いて、 M_5 がデータ転送木を作成する動作を説明する。図2の表は M_5 の要求受付表を示す。以下では、 M_i からのデータ要求をそれぞれ要求 M_i と呼ぶ。 M_5 は、時刻128(要求 M_1 のデータ転送開始時刻)になると、図3(a)のように、要求 M_1 の転送経路をデータ転送木の初期状態とする。まず M_5 は、要求 M_1 以外で要求時刻が最も古い要求 M_6 を選択し、その転送経路と転送木から M_4 をCCHとする。 M_5 は、 M_6 からCCHである M_4 までの経路 $M_6 - M_4$ をデータ転送木に統合する。図3(b)は、要求 M_6 の処理が終わった後のデータ転送木を示し、灰色の丸はCCH、点線で囲まれた部分は統合された経路を示す。次に M_5 は、要求 M_4 を選択し、 M_4 をCCHとするが、統合する経路はないため、データ転送木は変化しない。また、要求 M_7 を選択するが、CCHが自身となる(統合してもトラフィックが変化しない)ため、この要求の転送経路をデータ転送木に統合しない。最後に M_5 は、要求 M_2 の転送経路から M_3 をCCHとし、経路 $M_2 - M_3$ を統合する。ここで、CCHである M_3 から M_5 までの経路が、データ転送木と要求 M_2 の転送経路において異なるため、より要求時刻の遅い要求 M_2 の転送経路 $M_3 - M_8 - M_4$ に更新する。図3(c)は、要求 M_2 の処理が終わった後のデータ転送木を示し、太線は更新された経路を示す。

4.4 データ転送

本節では、移動体が、4.3で作成したデータ転送木を用いて、データを転送する手順について説明する。

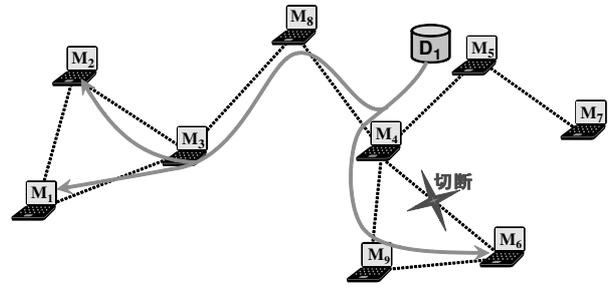


図4 データ転送

Fig. 4 Data transmission.

アドホックネットワークでは、移動体の移動によってネットワークトポロジが変化するため、データ要求パケットの転送経路がデータ転送時においても有効とは限らない。そこで移動体は、データ転送木に含まれる経路が有効か否かを調べるため、データ転送木において自身の子となっている移動体に確認パケットを送信する。確認パケットには、自身の識別子と作成したデータ転送木が含まれる。

確認パケットを受信した移動体は、パケットに含まれるデータ転送木における自身の子に確認パケットを転送する。このとき、トポロジ変化により送信先の移動体と接続していなければ、送信先の移動体を宛先とする経路探索を行う。経路を発見できた場合、データ転送木を発見した経路で更新し、確認パケットを送信する。一方、経路を発見できなかった場合、データ転送木から送信先の移動体を根とする部分木を削除する。

データ転送木における自身の子がない移動体は、確認応答パケットを自身の親となる移動体に送信する。確認応答パケットには、データ転送する移動体の識別子と確認パケットに含まれるデータ転送木に関する情報が含まれる。移動体は、全ての子から確認応答パケットを受信すると、パケットに含まれるデータ転送木をまとめ、自身の親に更新したデータ転送木に関する情報を含む確認応答パケットを送信する。

データ転送する移動体は、全ての子から確認応答パケットを受信すると、同様にパケットに含まれるデータ転送木をまとめ、更新したデータ転送木に沿ってデータパケットを転送する。データパケットには、データ転送する移動体の識別子、転送するデータを要求している(複数の)移動体の識別子、更新したデータ転送木、およびデータ本体が含まれる。

図4を用いて、図3(c)に示すデータ転送木を作成した M_5 が、データを転送する動作を示す。ここで、 M_4 は、データ転送木において自身の子となっている M_6 と直接接続していないため、データ転送木を経路探索によって発見した経路 $M_4 - M_9 - M_6$ に更新する。図3(d)は、 M_4 が経路探索によって経路を更新した後のデータ転送木を示し、太線は更新された経路を示す。

5. データ転送開始時刻の拡張

本稿で提案するデータ転送方式では、文献[8]の従来方式を拡張し、データ要求が転送された経路の有効時間を考慮して、その経路上の移動体による要求のデータ転送開始時刻を決定する。まず、データ要求パケットを受信した移動体は、データ要求

表 1 データ転送開始時刻

Table 1 Starting time of data transmission.

要求	最遅開始時刻	転送開始期限	データ転送開始時刻
M_1	128	113	113
M_2	144	154	144
M_4	176	131 → 154	131 → 154
M_6	162	122	122
M_7	146	141	141

の転送経路が有効な間にデータを転送できるよう、最遅開始時刻 T_{last} と転送開始期限 T_{limit} を求め、これらのうち早い方の時刻をデータ転送開始時刻 T'_{st} として設定する。最遅開始時刻 T_{last} は、従来方式におけるデータ転送開始時刻 T_{st} であり、式 (1) および式 (2) を用いて求める。一方、転送開始期限 T_{limit} は、次式を用いて求める。

$$T_{limit} = T_{now} + \tau_2. \quad (4)$$

τ_2 は事前に設定される正の定数で、データ要求が転送された経路が無効になるまでの時間の予測値を示し、 T_{limit} はデータ転送に用いる経路が有効である期限を示す。これにより、データ転送開始時刻は、デッドラインの時間内にデータを有効な転送経路で転送できる最も遅い時刻になる。

ここで、データ要求が転送された経路は、その経路上の移動体が要求している他のデータの転送にも利用できる。また、後から受信した要求の転送経路の方が、無効になるまでの時間が長い場合、その経路を用いた方が転送開始期限は遅くなる。4.2 で述べたとおり、転送開始までの時間が長いほど、一度のマルチキャストで多くの移動体にデータをまとめて転送できるため、データ転送開始時刻はできる限り遅い方が望ましい。そこで、データ要求パケットを受信した移動体は、その転送経路上の移動体による要求の転送開始期限も式 (4) を用いて更新する。また、更新後の転送開始期限が、最遅開始時刻よりも早い場合、データ転送開始時刻も更新する。

デッドラインが短い場合、最遅開始時刻が早いため、データ転送開始時刻は最遅開始時刻に設定される機会が多く、拡張方式は、従来方式と同じ動作になる。一方、デッドラインが長い場合、最遅開始時刻が遅いため、データ転送開始時刻は転送開始期限に設定される機会が多い。この場合、拡張方式は従来方式に比べて、トポロジ変化により転送経路が無効になる前にデータを転送できるため、データの可用性の低下を抑制できる。しかし、データ転送開始時刻が早くなるため、一度のマルチキャストでデータをまとめて転送できる移動体数が減少し、トラヒックの削減度合いが小さくなる。また、転送経路に含まれやすい移動体による要求は、転送開始期限が更新される。その場合、有効な転送経路を確保しながら、データ転送を開始するまでの時間が長くなるため、データの可用性を維持しつつ、トラヒックを削減できる。

表 1 は、図 2 に示すデータ要求に対して、 M_5 が設定した最遅開始時刻、転送開始期限、およびデータ転送開始時刻を示す。ただし、データ要求パケットを隣接する移動体に転送するのに

表 2 各パケットのサイズ

Table 2 Size of each packet.

パケット名	サイズ [B]
データ要求	$40 + 4 \times (\text{転送経路上の移動体数})$
確認	$24 + 8 \times (\text{転送木に含まれるリンク数})$
確認応答	$24 + 8 \times (\text{転送木に含まれるリンク数})$
データ	$1,024 \times 10^3 + 24 + 8 \times (\text{転送木に含まれる要求数})$ $+ 8 \times (\text{転送木に含まれるリンク数})$

要する時間は 1[秒] とし、 $\tau_2=30$ [秒] とする。例えば、 M_5 が要求 M_1 を受信すると、最遅開始時刻は式 (3) より時刻 128、転送開始期限は時刻 113 (= 83 + 30) となるため、データ転送開始時刻はより早い転送開始期限 (時刻 113) に設定される。一方、要求 M_2 を受信すると、最遅開始時刻の方が転送開始期限より早くなるため、データ転送開始時刻は最遅開始時刻 (時刻 144) に設定される。また、要求 M_2 の転送経路には M_4 が含まれるため、要求 M_4 の転送開始期限は要求 M_2 の転送開始期限 (時刻 154) に更新され、データ転送開始時刻も更新される。

6. 性能評価

本章では、拡張方式の性能評価のために行ったシミュレーション実験の結果を示す。この実験では、ネットワークシミュレータ QualNet4.0 [9] を用いた。

6.1 シミュレーション環境

1,000[m] × 1,000[m] の 2 次元平面上の領域に、50 台の移動体 (M_1, \dots, M_{50}) が存在する。各移動体はランダムウェイポイントモデル [1] に従い、0.5 から 1.5[m/秒] の速度で移動する。停止時間は 30[秒] とした。各移動体は、IEEE802.11b を使用し、伝送速度 11[Mbps]、電波伝搬距離が 100[m] 程度となる送信電力でデータを送信する。

ネットワーク内には、10 種類のデータ (D_1, \dots, D_{10}) が存在し、データ D_i は移動体 M_i ($i = 1, \dots, N$) にオリジナルデータとして保持されるものとした。各データのサイズは 102.4[KB] とした。各移動体は、平均 30 秒の指数分布に基づいた間隔でデータ要求を発行し、アクセス対象のデータを 10 種類の中からランダムに決定する。データの取得時間のデッドラインは DL 秒とした。また、 τ_1 は 30 秒、 τ_2 は 60 秒とした。データ転送で用いる各パケットのサイズを表 2 に示す。

以上のシミュレーション環境において、各移動体の初期位置をランダムに決定し、1800 秒を経過させたときの以下の評価値について調べた。

アクセス成功率 :

アクセス発生回数に対するアクセス成功回数の割合。ただし、アクセス発生回数は、シミュレーション時間内に発生したアクセス要求の総数のうち、デッドラインがシミュレーション時間内に設定されているアクセス要求の数を表し、アクセス成功回数は、データを要求した移動体がデッドラインの時間内にデータを取得できた要求の総数を表す。

制御トラヒック :

シミュレーション時間内にデータ転送によって送信された制

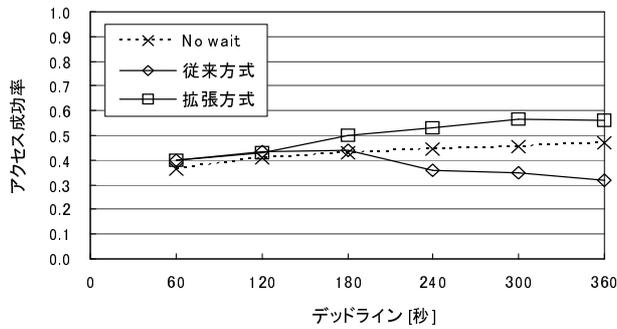


図 5 デッドラインとアクセス成功率
Fig. 5 Deadline and data accessibility.

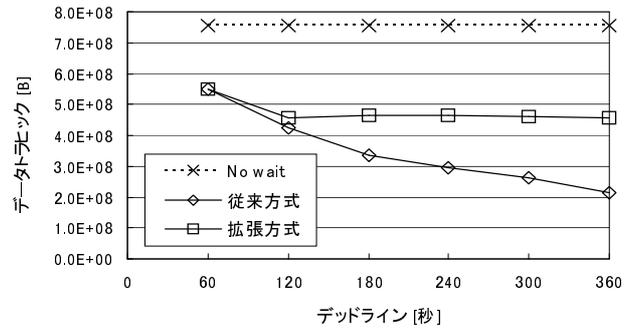


図 7 デッドラインとデータトラフィック
Fig. 7 Deadline and data traffic.

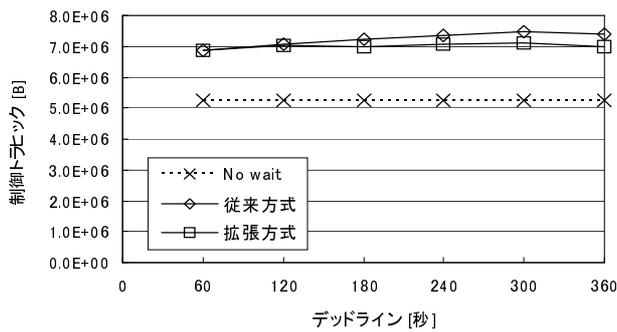


図 6 デッドラインと制御トラフィック
Fig. 6 Deadline and control traffic.

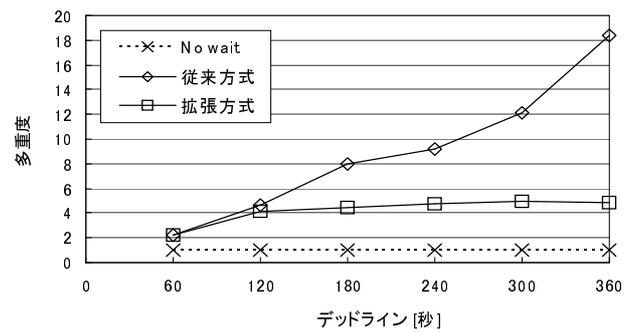


図 8 デッドラインと多重度
Fig. 8 Deadline and multiplicity.

御パケット（データ要求，確認，および確認応答パケット）の総バイト数。

データトラフィック：

シミュレーション時間内にデータ転送によって送信されたデータパケットの総バイト数。

多重度：

データをもつ移動体が一度のマルチキャストを用いてデータを転送できた移動体数の平均。

6.2 評価結果

デッドラインを変化させたときの性能を調べた。その結果を図 5, 図 6, 図 7, および図 8 に示す。これらの図において、横軸はデッドラインを表している。縦軸は、図 5 ではアクセス成功率、図 6 では制御トラフィック、図 7 ではデータトラフィック、図 8 では多重度を表している。なお、比較のため、各移動体が、データ要求パケットを受信すると、即座にユニキャストでデータを転送する場合の結果を 'No wait' として示す。また、文献 [8] で提案したデータ転送方式の結果を従来方式として示す。

図 5 の結果より、デッドラインが長くなると、拡張方式のアクセス成功率は高くなる。これは、デッドラインが長くなると、移動体のデータアクセスが失敗するまでの時間が長くなり、それまでにデータを取得できる機会が増加するからである。拡張方式は 'No wait' に比べて、常にアクセス成功率が高い。これは、拡張方式では、移動体がマルチキャストを用いてデータを転送することで、データトラフィックが小さい (図 7) ため、電

波干渉によってパケットが損失する機会が少ないからである。またデッドラインが長い範囲では、拡張方式は従来方式に比べて、アクセス成功率が高い。これは、拡張方式では、転送開始期限を設定することで、トポロジ変化によって転送経路が無効になる前にデータを転送するため、アクセス成功率が低下しない一方、従来方式では、転送経路が無効になり、移動体が要求したデータを取得できなくなる機会が増加するため、アクセス成功率が低くなる。

図 6 の結果より、デッドラインが長くなると、拡張方式の制御トラフィックはやや大きくなる。これは、データのトラフィックが小さくなる (図 7) ことで、制御パケットが損失する機会が減少するからである。'No wait' では、データ要求パケットを受信した移動体が、即座にデータを転送するため、制御トラフィックはデッドラインにほとんど影響を受けない。拡張方式は 'No wait' に比べて、制御トラフィックが大きい。これは、'No wait' では、拡張方式に比べて損失する制御パケットが多いことに加え、データ要求パケットを受信した移動体が即座にデータを転送するため、確認パケットおよび確認応答パケットによるトラフィックが発生しないからである。また、デッドラインが長い範囲では、拡張方式は従来方式に比べて、制御トラフィックが大きい。これは従来方式のデータトラフィックが小さくなり (図 7), 制御パケットが損失する機会が少なくなるからである。

図 7 の結果より、デッドラインが長くなると、拡張方式のデータトラフィックは小さくなった後、ほぼ一定になる。デッドラ

インが長くなると、データ転送開始時刻までの時間が長くなるため、多くの移動体にデータをまとめて転送でき、データトラヒックは小さくなる。デッドラインがさらに長くなると、データ転送開始時刻が転送開始期限に設定される機会が多くなるため、データトラヒックはデッドラインに影響を受けなく（一定に）なる。‘No wait’のデータトラヒックは、図6の結果における考察と同様の理由により、デッドラインにほとんど影響を受けない。拡張方式は‘No wait’に比べて、データトラヒックが小さい。これは、拡張方式では、移動体がマルチキャストを用いて、同じデータを要求している複数の移動体にデータをまとめて転送することで、トラヒックを効果的に削減できていることを示す。データトラヒックは、制御トラヒックに比べて100倍以上大きいので、拡張方式はトラヒックを大きく削減できていると言える。なお、デッドラインが長い範囲では、拡張方式は従来方式に比べて、データトラヒックが大きい。これは、従来方式では、トポロジ変化によって転送経路が無効になり、移動体が要求したデータを取得できなくなる機会が増加するからである。

図8の結果より、デッドラインが長くなると、拡張方式の多重度は大きくなった後、ほぼ一定になる。デッドラインが短い場合、最遅開始時刻が早いので、データの転送を開始するまでに別のデータ要求を受信する機会が少なく、多重度は小さい。デッドラインが長くなると、データ転送開始までの時間が長くなるため、多重度は大きくなる。デッドラインがさらに長くなると、図7の結果における考察と同様、データ転送開始時刻が転送開始期限に設定される機会が多くなるため、デッドラインに影響を受けなく（一定に）なる。なお、従来方式では、データ転送開始時刻は常に最遅開始時刻に設定されるため、デッドラインが長くなるほど、多重度が大きくなる。また‘No wait’では、データをまとめずに転送するため、多重度は常に1である。

7. まとめ

本稿では、文献[8]で提案したデータ転送方式を拡張し、デッドラインが長い環境においてもデータの可用性を維持するデータ転送方式を提案した。拡張方式では、デッドラインが長い場合に、データ要求が転送された経路の有効時間を考慮して、その経路上の移動体による要求のデータ転送開始時刻を決定する。シミュレーション実験の結果より、拡張方式は、即座にデータをユニキャストで転送する場合と比較して、アクセス成功率を維持しつつ、トラヒックを削減することを確認した。またデッドラインが長い範囲においては、従来方式と比較して、アクセス成功率の低下を抑制できることを確認した。

今後は、従来方式および拡張方式におけるパラメータの値を、環境に応じて適切に決定する方法について検討する予定である。また、移動体が複製を配置する環境を想定して、データ転送によるトラヒックをさらに削減する方式について検討する予定である。

謝辞 本研究の一部は、文部科学省科学研究費補助金・基盤研究(A)(17200006)、特別研究員奨励費(19-2204)、平成20年度

総務省委託研究「ユビキタスサービスプラットフォーム技術の研究開発」、および財団法人国際コミュニケーション基金の研究助成によるものである。ここに記して謝意を表す。

文 献

- [1] Camp, T., Boleng, J. and Davies, V.: A Survey of Mobility Models for Ad Hoc Network Research, *Wireless Communications and Mobile Computing (WCMC)*, Vol. 2, No. 5, pp. 483–502 (2002).
- [2] Feeney, L. M. and Nilsson, M.: Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment, *Proc. IEEE INFOCOM'01*, pp. 1548–1557 (2001).
- [3] 原 隆浩: アドホックネットワークにおけるデータ利用性向上のための複製配置, 電子情報通信学会和文論文誌 B, Vol. J84-B, No. 3, pp. 632–642 (2001).
- [4] Lee, S.-J., Gerla, M. and Chiang, C.-C.: On-Demand Multicast Routing Protocol, *Proc. IEEE WCNC'99*, pp. 1298–1304 (1999).
- [5] Perkins, C. E. and Royer, E. M.: Ad-hoc On-Demand Distance Vector Routing, *Proc. IEEE WMCSA'99*, pp. 90–100 (1999).
- [6] Royer, E. M. and Perkins, C. E.: Multicast Operation of the Ad-Hoc On-Demand Distance Vector Routing Protocol, *Proc. IEEE/ACM MOBICOM'99*, pp. 207–218 (1999).
- [7] 篠原昌子, 林 秀樹, 原 隆浩, 西尾章治郎: アドホックネットワークにおける消費電力を考慮したデータ複製配置方法, 情報処理学会論文誌, Vol. 47, No. 1, pp. 15–27 (2006).
- [8] 篠原昌子, 原 隆浩, 西尾章治郎: アドホックネットワークにおける効率的なデータ転送に関する一考察, 情報処理学会研究報告(2007-MBL-43), Vol. 2007, No. 116, pp. 25–32 (2007).
- [9] Scalable Network Technologies: “QualNet”, <URL:<http://www.scalable-networks.com>>.