

ノード間の通信可能時間に基づく DTN 性能向上
方式の研究

筑波大学
図書館情報メディア研究科
2017 年 3 月
HAN JIA

目次

第 1 章	序論	1
第 2 章	関連研究	2
2.1	DTN(Delay/Disruption Tolerant Network)	2
2.2	DTN 評価指標	3
2.3	DTN ルーティング方式	3
2.3.1	Epidemic	3
2.3.2	PRoPHETv2	4
2.3.3	Spray and Wait	6
2.4	バッファ管理	6
2.4.1	送信方式	6
2.4.2	廃棄方式	7
2.5	DTN の送信方式の問題	7
第 3 章	通信可能時間に基づく送信制御方式の性能解析	9
3.1	通信可能時間の計算方法	9
3.2	通信可能時間に基づく送信制御方式	9
3.3	実験環境	10
3.3.1	シミュレーションの設定	11
3.4	実験結果	13
3.5	性能解析のまとめ	17
第 4 章	通信可能時間に基づく送信制御方式における性能改善策の提案と評価	18
4.1	通信可能時間に基づく送信制御方式の改善策	18
4.2	シミュレーションの設定	19
4.3	改善策の評価	19
4.4	実験結果の解析	23
第 5 章	結論	25
謝辞	26
参考文献	27

図目次

図 1	DTN とインターネットにおけるプロトコル階層モデル	2
図 2	DTN のデータ送信手順	3
図 3	ノード間のやり取り	4
図 4	Chain Topology 問題の例	5
図 5	ノード間のやり取り	5
図 6	通常データと anti-packet の構造	7
図 7	データの送信フローチャート	10
図 8	ヘルシンキの地図上の 4500m*3000m 範囲	11
図 9	地図上の道路	12
図 10	送信停止データ数(the number of aborted data)	14
図 11	データ到達率(delivery_prob)	14
図 12	データ転送回数(the number of relayed data)	15
図 13	データ廃棄数(the number of dropped data)	16
図 14	オーバーヘッド率(overhead_ratio)	16
図 15	改良したデータ送信のフローチャート	19
図 16	送信停止データ数(the number of aborted data)	20
図 17	データ到達率(delivery_prob)	21
図 18	データ転送回数(the number of relayed data)	21
図 19	データ廃棄数(the number of dropped data)	22
図 20	オーバーヘッド率(overhead_ratio)	23

表目次

表 1	シミュレーションの設定	13
表 2	各指標の比較結果	17
表 3	各指標の比較結果	24

第1章 序論

近年、移動体通信において、基地局等の通信インフラを用いず、移動体同士が直接通信するネットワークとして、モバイルアドホックネットワーク(MANETs : Mobile ad hoc networks)や高度交通システム (ITS : Intelligent Transport Systems) などが注目されている。これらのネットワークの特徴として、ノードの移動によってネットワークポロジが常に変化する、2つのノードが通信圏内に入った時だけ送信できる、ネットワークの分断が頻繁に発生する、という特徴がある。しかし、従来のTCP/IP通信技術による通信では、物理的なリンクが安定していることを前提としているので、遅延や分断が発生しやすいネットワークには、従来のTCP/IP通信技術は向いていない。

そこで近年、このような問題を克服する通信技術として、DTN(Delay/Disruption Tolerant Network) [1]-[3]が提案されている。DTNは元々宇宙空間の通信技術の研究から生まれてきたものだが、災害などで基地局に甚大な被害が起こった場合など高遅延が頻繁に発生するような劣悪な環境でも重要な情報を送受信できると期待されている。また、交通網のような分断の問題がある通信を実現するための研究も続けられている。

DTNでは、遅延や分断が発生しやすい環境において通信を行うために蓄積運搬形転送方式 (Store Carry and Forward) [3]の通信を行うバンドル(bundle)層を採用する。バンドル層の基本プロトコル仕様はRFC5050[4]で規定されている。バンドル層においてデータ転送の単位となるバンドルのサイズや内容については特に規定はなく、バンドルはファイルのセグメントでも、単一のファイルでも、複数のファイルでも良い。セグメントを用いる場合は、分解組立(フラグメンテーション)機能が別途必要になる。

DTNでは、送信状態が常に変化している。2つのノードはお互いの通信圏内に入ると送信できるが、ノードの移動によりお互いの通信範囲から離れると送信が中断され、データの送信は失敗に終わる。その際、データ送信に要した通信時間や受信ノードのバッファリソースは、無駄に消費されるだけでなく、他の通信チャンスまで奪ってしまう。そこで本研究では、僅かな通信可能時間やバッファリソースを無駄にしないように、ノード間の通信可能時間を予測し、時間内に送信できるデータだけを送信する制御方式を提案する。

まず、既存ルーティング方式であるEpidemic[5]とPRoPHETv2 [6]を対象に、通信可能時間に基づく送信制御の有無で性能の違いを調べる。実験には、ヘルシンキの街で人と車が移動しながらデータを送受するDTNシミュレータ(The One Simulator)を用い、データ到達率、データ転送回数、データ廃棄数、オーバーヘッド率、送信停止データ数の5つの指標で性能評価を行う。実験の結果、既存方式に通信可能時間に基づく送信制御を加えるだけでは、却って性能が低下することを示し、その原因分析を行う。次に、その結果に基づいた改善策として、送信確認パケット(anti-packet)を用いる方式、さらにanti-packetをデータパケットより優先する方式(anti-first)を、通信可能時間に基づく送信制御方式と適宜組み合わせることで性能の比較評価を行う。その結果、通信可能時間に基づく送信制御と送信確認パケット(anti-first)を優先的に処理する方式を組み合わせた場合が最も性能が良く、DTNにおいて最も重要な指標であるデータ到達率が、Epidemicの2倍以上、PRoPHETv2の1.5倍以上になることを示す。

以降、本論文は以下のように構成する。第2章ではDTNのルーティング方式とバッファ管理方式および問題点について説明し、第3章ではノード間の通信可能時間に基づく送信制御方式の提案と性能解析実験による評価について述べる。第4章では、改善した送信制御方式の提案と評価を行う。最後に、第5章にて結論を述べる。

第2章 関連研究

本節では、本研究の関連技術としてDTNの技術及びDTNの評価指標について整理し、DTNルーティング方式、バッファ管理について紹介する。また、DTNの通信時の問題点についても明らかにする。

2.1 DTN(Delay/Disruption Tolerant Network)

惑星間通信研究から生まれてきた DTN は、高遅延や通信路の断絶が発生しやすい環境においても、信頼性がある通信を実現する通信方式であり、モバイルアドホックネットワーク、車々間通信、無線センサネットワークなど、様々な環境下における通信技術として、また災害発生時にも利用可能な通信手段として注目されている。DTN 技術の標準化を行うグループとして IRTF(Internet Research Task Force)の中の研究グループ DTNRG[7](The Delay Tolerant Networking Research Group)が挙げられる。DTN では、遅延や分断が発生しやすい環境において通信を実現するため、蓄積運搬形転送方式[3]の通信を行うバンドル(bundle)層を採用する。図 1 に、DTN とインターネットにおけるプロトコル階層モデルを示す。バンドル層は、永続ストレージを持ち、転送されてきたバンドルと呼ばれる可変長データを他の中継ノードへの通信が可能となるまで永続ストレージに一時蓄積する。バンドル層のための基本プロトコル仕様は RFC5050[4]で提案されている。DTN のアーキテクチャ全体を説明した RFC4838[8]もあり、それは惑星間ネットワークのためのアーキテクチャの起源である。

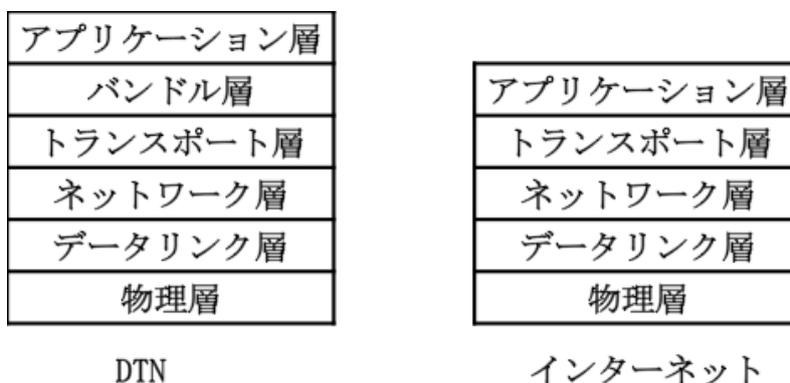


図 1 DTN とインターネットにおけるプロトコル階層モデル

図 2 に、DTN のデータ送信手順を示す。送信元は移動して中継ノードと出会う時、送信元のアプリケーションで生成されたデータすなわちバンドルを中継ノードに送信する。中継ノードは一時バンドルを永続ストレージに蓄積する。中継ノードは移動して、また他のノードと通信できるようになるとバンドルを渡す。これを蓄積運搬形転送方式と呼ぶ。データが宛先に着くと宛先のアプリケーションに渡す。この方式により、送信元は宛先に出会わなくても、宛先にデータを届けることができる。DTN はこの送信手順により、高遅延や通信路の断絶が発生しやすい環境においても通信を実現してきた。

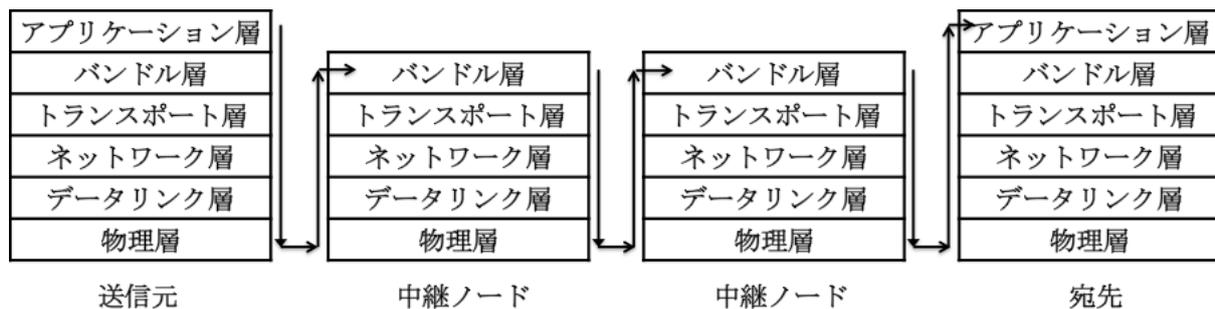


図 2 DTN のデータ送信手順

2.2 DTN 評価指標

今まで先駆者たち、様々な面から DTN の性能を向上するための提案方式を評価したが、ここで一般的な評価指標を紹介する。

- データ到達率(delivery_prob)は生成されたデータのうち宛先に到達したデータの割合である。データの到達率は DTN の性能評価指標の中に、最も重要な指標である。
- データ転送回数(the number of relayed data)はノードからノードへデータを転送する回数である。
- データ廃棄数(the number of dropped data)はバッファが一杯になった時、新たなデータを受け取るため、廃棄されたデータの数である。
- オーバーヘッド率(overhead_ratio)=(データ転送回数-到達したデータ数)/到達したデータ数
- 到達遅延(latency_av)は宛先へ到達したデータのみについて、生成されてから到着するまでに必要な平均時間である。

2.3 DTN ルーティング方式

DTN におけるルーティング方式については、多くの研究が行われ、データの複製を制限しない方式とデータの複製を制限する方式に分かれる。前者として Epidemic[5]が挙げられる。後者として、PRoPHETv2 [6] や Spray and Wait[9,10]が挙げられる。

2.3.1 Epidemic

Epidemic は vahdat らによって提案され、Flooding と呼ばれることもある。Epidemic においては、送信元は全ての通信可能ノードに対してデータの複製を作成していく。これにより、複製データが他のノードに送信され、複製データのいずれかが目的地ノードに到着する。また、自身が既に持っているデータを通信相手から受けることを防ぐため、Epidemic はサマリーベクターメッセージ(SVM: Summary Vector)のやり取りも行っている。サマリーベクターメッセージはノードが保有するデータのリストを含む。ノード間のやり取りを図 3 に示す。ノード A がノード B と通信可能となった時、まず A が自分のサマリーベクターメッセージを B に送る。B はそのメッセージを受信し、B が保有するデータのリストと比較し、B が持っていないデータのリストをリクエストとして A に送信する。そして、A は B が持っていないデータを B に送信する。

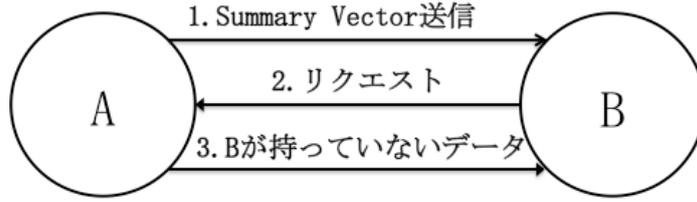


図 3 ノード間のやり取り

2.3.2 PRoPHETv2

PRoPHETv2はPRoPHET[11]の改良手法としてGrasicらによって提案されたルーティング方式である。

まず、PRoPHETを紹介する。PRoPHETはLindgrenらによって提案された方式である。PRoPHETはデータの宛先への到達確率の高いノードを選択してデータを転送する方式である。PRoPHETでは、各ノードが以下のように宛先への到達確率を計算する。ここでノードA、B、Cを使って説明する。 $P_{(A,B)} \in [0,1]$ をノードAから宛先ノードBへの到達確率として計算する。 $P_{init} \in [0,1]$ は初期値として、ノードAがノードBと出会う時、 $P_{(A,B)}$ を以下のアルゴリズムに従い、計算する。

$$P_{(A,B)} = P_{(A,B)old} + (1 - P_{(A,B)old}) * P_{init} \quad (1)$$

また、長い期間ノードBとずっと出会っていない時、定数 $\gamma \in [0,1]$ を用いて宛先ノードBへの到達確率の更新を行う。 k は最後に更新が行われた時刻から経過した単位時間の回数である。アルゴリズムは以下のようなになる。

$$P_{(A,B)} = P_{(A,B)old} * \gamma^k \quad (2)$$

また、ノードAがノードBと出会った時、ノードAから宛先ノードCへの到達確率 $P_{(A,C)} \in [0,1]$ も更新する。なお $\beta \in [0,1]$ は定数である。

$$P_{(A,C)} = P_{(A,C)old} + (1 - P_{(A,C)old}) * P_{(A,B)} * P_{(B,C)} * \beta \quad (3)$$

GrasicらはPRoPHETのParking Lot問題とChain Topology問題を分析し、改めて公式(1)と(2)を定義してPRoPHETv2を提案した。

PRoPHETでは、ノードたちが短時間内に出会いを繰り返しても、公式(1)に従い、宛先への到達確率を更新するのがParking Lot問題である。この問題はよく無線デバイスを搭載した大規模な駐車場で発生した。これらのノードは一旦離れ、短時間後再び出会う時、公式(1)に従い、宛先への到達確率を更新する。しかし、現世界では、再びの出会いは何時間もしくは何日後のことである。その結果、宛先への到達確率の予測は大きい誤差が発生する。この問題を解決するため、Grasicらは公式(1)を修正した。 $P_{max} \in [0,1]$ は初期値として使用される。 I_{typ} はコネクシオンの想定された典型的な時間間隔のパラメータである。 $Intvl_B$ はノードBと出会ってから現在までの時間である。

$$P_{(A,B)} = P_{(A,B)old} + (1 - P_{(A,B)old}) * P_{init} \quad (4)$$

$$P_{init} = P_{max} * (Intvl_B / I_{typ}) ; \text{if } 0 \leq Intvl_B \leq I_{typ}$$

$$P_{init} = P_{max} ; \text{otherwise}$$

図4はChain Topology問題の例を示す。最初、ノードAとノードBは一回だけ出会う。その後、ノードB、C、Dは周期的に出会い、公式(3)に従い、 $P_{(A,C)}$ と $P_{(A,D)}$ を更新する。一定期間の後、ノードAとノードC、Dは一回も出会ったことがないのに、 $P_{(A,C)}$ と $P_{(A,D)}$ は $P_{(A,B)}$ より高くなってしまふ。この問題を解決するため、Grasicらは公式(3)修正した。

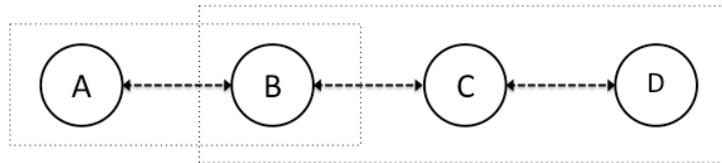


図 4 Chain Topology 問題の例

$P_{(A,C)}$ の新しい方式計算方式は以下のようなになる。 $\beta \in [0, 1]$ は定数である。

$$P_{(A,C)} = \max(P_{(A,C)old} , P_{(A,B)} * P_{(B,C)} * \beta) \quad (5)$$

図5はノード間のやり取りを示す。ノードAとノードBが出会う時、まずお互いにサマリーベクタと各データの宛先への到達確率を交換する。そしてノードは受け取った情報により自分の各宛先への到達確率を更新する。Bが持っていないデータの宛先が他のノードCであり、しかも $P_{(B,C)} > P_{(A,C)}$ であるなら、AはデータをBに送信する。

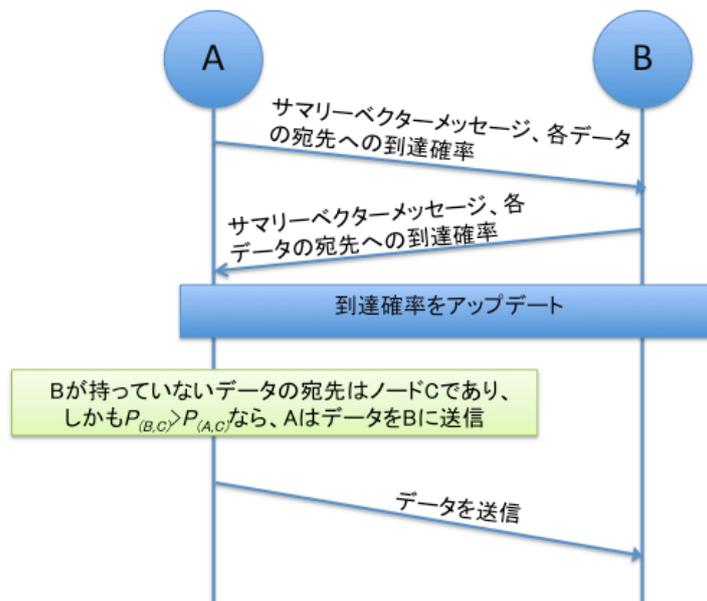


図 5 ノード間のやり取り

2.3.3 Spray and Wait

Spray and wait[9,10]はSpyropoulos らによって提案された手法であり、Epidemicの考え方に基づくルーティング方式である。Spray and waitはデータあたり行う可能な複製数 N を制限する方式である。Spray and waitには、2つの段階がある。一つはSpray段階であり、データの生成側が出会う各ノードにデータを複製していく。Spray段階は決められた数の複製数を達成すると終了する。もう一つはWait段階であり、データの複製を持っているノードはデータの宛先ノードと出会う時しか送信しない。

また、Binary Sprayという方式もある。データの複製を持つノードAはデータDの複製を持っていないノードBと出会う時、データDを送る。その際、Aが持っているデータDの可能な複製数は半分の $N/2$ になり、Bが持っているデータDの可能な複製数も $N/2$ になる。ノードが持っているデータDの複製数が1になると、Spray段階が終わり、Wait段階になる。

2.4 バッファ管理

ノードのバッファサイズやノード間の通信時間が有限である時、ルーティングの性能はバッファ管理手法により変化する。バッファ管理手法は送信方式と廃棄方式に分けられる。

2.4.1 送信方式

ノード間の通信時間が有限であると全てのデータを転送できないので、データの送信順序が非常に重要になる。送信方式は優先度設定ありと優先度設定無しの2種類に分けられる。優先度は一部分のデータの到達率を高くするために設定される。

優先度設定無し

ランダムと FIFO(First In First Out)がある。

- ランダムはバッファの中からランダムにデータを選択し、出会うノードに送信する方式である。
- FIFO は先に受信したデータを先に会うノードに送信する方式である。

優先度設定あり

Oldest First OutやLIFO(Last In First Out)やLeast Forwarded First Outなどが挙げられる。

- Oldest First Out はバッファの中にある一番古いデータから先に優先送信する方式である。DTN 技術において、各々のデータは TTL(Time to Live)を持っている。TTL とは、一つのデータが生成されてからネットワークの中に存在できる時間である。TTL の時間が切れると、ネットワーク内の全てのノードからそのデータを削除する。Oldest First Out 送信方式は TTL の残りが少ないデータの到達率を高めるため、そのデータを優先してネットワークの中に広げようとする方式である。
- LIFO は受信したばかりのデータから先に送信する方式である。この方式は受信したばかりのデータが新しいデータと考え、新しいデータを広げようとして先に送信する。
- Least Forwarded First Out は転送回数が少ないデータから先に送信する方式である。転送回数が少ないデータは、ネットワークの中にそのデータを持っているノードが少なくないと考えられる。このようなデータを優先することでネットワーク中にデータを広げてデータの到達率を高める方式である。

2.4.2 廃棄方式

廃棄方式は2種類に分類できる。一つはバッファが一杯になる前にデータを削除する。もう一つはデータを受信する際、バッファが一杯になって新たなデータを受信するため、データを削除する。

前者の例としては、Time-Based TTL とフィードバックによりデータを削除する方法が挙げられる。

- Time-Based TTL は残る TTL が0になると、データを削除する方式である。
- フィードバックによりデータを削除する方法は、データが宛先に到達すると、宛先ノードはネットワーク内に存在する他のすべてのノードに対して到達した情報をフィードバックし、この情報を受信したノードは送信済のデータを削除する方式である。代表としては Haas らによって提案された Vaccine 回復手法がある[12]。Vaccine 回復手法はデータが宛先に到達した後、そのデータと同じ ID を持つ受信確認パケット anti-packet というデータを生成し、他のデータと同じように送信する。anti-packet を受信したノードは、anti-packet の ID をバッファ内に保存している通常のデータの ID とマッチングし、一致したら、バッファからその通常データを廃棄する。図 6 に、通常データと anti-packet の構造を示す。本研究でも、この方法を利用している。

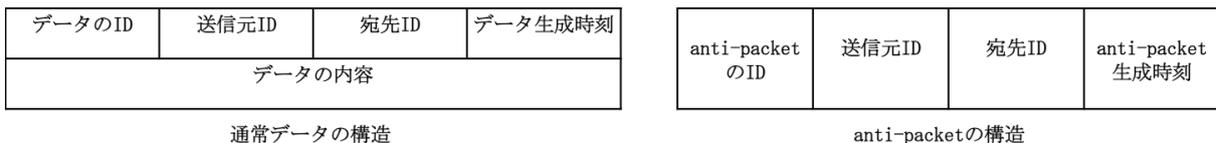


図 6 通常データと anti-packet の構造

後者の例としては、MOFO(Most Forwarded First Out)や Oldest Drop や N-Drop[13]が挙げられる。

- MOFO はバッファが一杯になると、バッファ内で最も転送回数の多いデータから先に廃棄する。転送回数が多いデータは既にネットワーク中に広がっていると考えられるので、そのデータを削除することで転送回数が少ないデータを広げられる廃棄方式である。
- Oldest Drop は最も古いデータから廃棄する方式であり、すなわち残る TTL が最も少ないデータから廃棄する方式である。古いデータは既にネットワーク中に広がっていると考えられるし、TTL の残りが少ないので、広げてもすぐ TTL が満了して削除されることが考えられるため、廃棄しても影響が少ない方式である。
- N-Drop[13]は N 回の転送を達成したデータから廃棄する方式である。もし N 回転送されたデータがないなら、最後のデータから先に廃棄する。

2.5 DTN の送信方式の問題

DTN の性能はノード間の通信可能時間に依存する。ノードの移動により送信途中でノードがお互いの通信範囲から外れると送信が中断され、データ送信は失敗に終わる。その際、受信ノードはそのデータの受け取った部分を削除するため、送信によって消費されたリソースは無駄になり、他のデータの送信チャンスも減少する。こういったデータ送信の中断やデータ廃棄に伴う網リソースの無効消費が DTN の性能を低下させる要因となる。そこで本研究では、ノード間の通信可能時間を予測し、時間内に送信できるデータしか送信しないよう

にすることで送信の失敗を減少させ、DTN 性能を向上する送信制御方式を提案する。

第3章 通信可能時間に基づく送信制御方式の性能解析

本章では、通信可能時間の計算方法及び通信可能時間に基づく送信制御の方式を説明し、The ONE シミュレータ [14,15] を用いて評価を行う。その結果により、DTN における通信可能時間に基づく送信制御方式の性能を解析する。

3.1 通信可能時間の計算方法

通信可能時間に基づく送信制御方式を紹介する前に、まず、通信可能時間の計算方法を説明する。通信可能時間 T を計算するために、ノードの速度や移動方向を知る必要があるので、各ノードは GPS を装備していると仮定する。今はノード A がノード B と通信可能になると考える。A、B の移動速度が v_A 、 v_B であり、A、B の移動方向が θ_A 、 θ_B ($0 \leq \theta_A, \theta_B < 2\pi$) であり、A、B の通信範囲は r_A 、 r_B である。また A、B の現在位置は (x_A, y_A) 、 (x_B, y_B) である。計算方式は以下に示す。

$$T = \frac{\sqrt{r^2(b^2 + d^2) - (ad - bc)^2} - (ab + cd)}{b^2 + d^2} \quad (6)$$

計算方式の中に、 a 、 b 、 c 、 d 、 r の計算は以下に示す。

- $a = x_A - x_B$;
- $b = v_A * \cos\theta_A - v_B * \cos\theta_B$;
- $c = y_A - y_B$;
- $d = v_A * \sin\theta_A - v_B * \sin\theta_B$;
- $\text{if}(r_A < r_B) \{r = r_A\} \quad \text{else}\{r = r_B\}$

3.2 通信可能時間に基づく送信制御方式

通信可能時間に基づく送信制御方式では、2つのノードが通信できる時、サマリーベクターメッセージを交換し、その情報を用いてノード間の通信可能時間を計算する。そして、その時間内に送信可能なデータしか送信しない。今、ノード A がノード B と通信可能になると仮定する。最初、ノード A、B はお互いにサマリーベクターメッセージを送信する。その時、通信可能時間を計算するためにノードの移動速度、移動方向、通信範囲及びその時点での位置の情報も相手に送信する。そしてノードは受け取った情報により通信可能時間を計算する。ノード間のやり取りはルーティング方式によって違う。Epidemic では、ノードは図 3 のようにやり取りを行い、PRoPHETv2 では、ノードは図 5 のようにやり取りを行う。

通信可能時間の計算はノードの GPS 情報に含まれる現在位置、移動速度、移動方向のデータを利用して式(6)で行う。しかし、ノードは相手ノードに向かって真直ぐ進むわけではなく、道路に沿って移動するので、通信可能時間は実際とは異なり、式(6)で計算した通信可能時間には多少の誤差が生じる。そこで、推定誤差がデータの到達率に影響を与えないように、

A は宛先が B であるデータがあれば、最初は通信可能時間 T を考えずに、そのデータを最優先で B に送信することとする。

図 7 に、データ送信のフローチャートを示す。宛先が B であるデータがなければ、A は FIFO によりデータリストを取り扱う。

- そして、Epidemic では、順番にデータを取り出し、通信可能時間 T 内に送信を完了でき、しかも B が持っていないデータを B に送信する。
- PРоPHETv2 では、順番に取り出した通信可能時間 T 内に送信を完了でき、データの宛先への到達確率が B より低くかつ B が持っていないデータを B に送信する。

その後、通信可能時間を更新し、再びバッファ内に条件を満足するデータがあるかどうかを判断して送信する。もしバッファ内に条件を満足するデータがないなら、A、B はまだお互いの通信範囲にいても、送信しない。

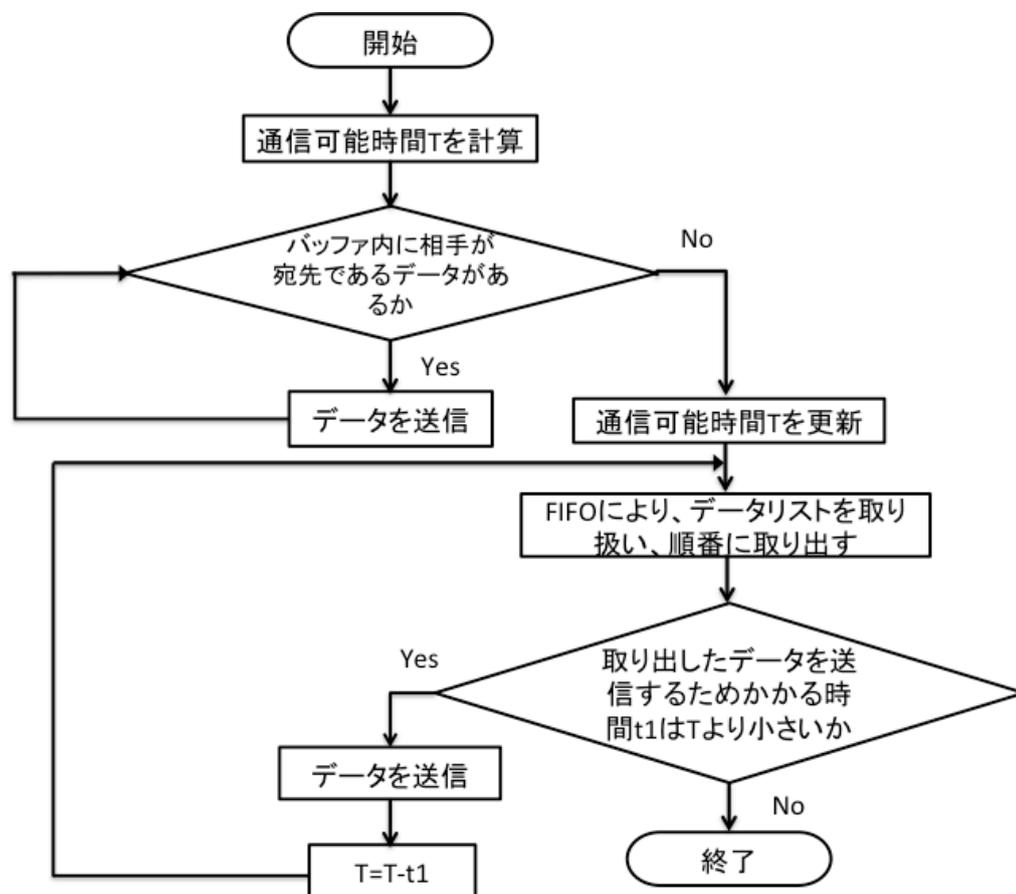


図 7 データの送信フローチャート

3.3 実験環境

本研究では、Java ベースのシミュレータである The ONE(The Opportunistic Network Environment)シミュレータを使い、DTN の性能を評価する。このシミュレータは異なるノードの移動モデルを生成できるだけでなく、DTN 性能を評価するための指標の測量機能も万全である。

本研究の評価指標としては以下の五点を用いる。前の四点の指標の説明は 2.2 で行ったので、ここは省略である。また、本研究では、送信が中断されることで失敗した送信数を減少

させ、DTN の性能を向上させることが目的とするので、停止されたデータ数も測定する。これは送信停止データ数という五点目の指標として評価する。

- データ到達率
- データ転送回数
- データ廃棄数
- オーバーヘッド率
- 送信停止データ数(the number of aborted data)

3.3.1 シミュレーションの設定

シミュレーションの設定を表 1 に示す。本研究はフィンランドのヘルシンキの地図上の 4500m*3000m 範囲をシミュレーション範囲として 36000s のシミュレーションを行った。また、実験はシナリオごとに 10 回で行う。そして、図 8 に、ヘルシンキの地図上の 4500m*3000m 範囲を示す。

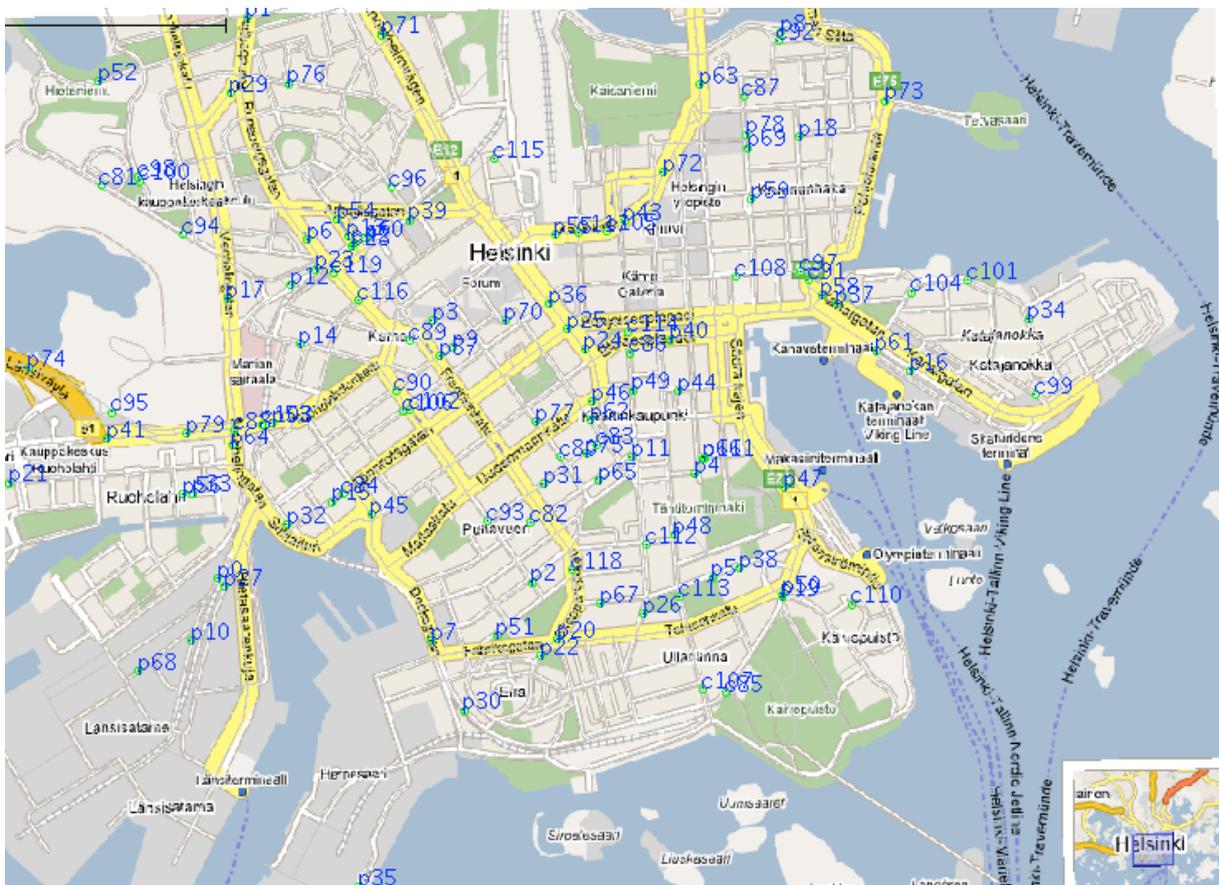


図 8 ヘルシンキの地図上の 4500m*3000m 範囲

- グループ：
 - 二種類のノードグループを設定する。一つは歩行者ノード 80 人で組み、もう一つは 40 台の自動車ノードで組む。
- 移動速度：
 - それぞれの速度の変化範囲は 0.5~1.5m/s と 2.7~13.9m/s であり、目的地に到着

した時、一時停止する。一時停止時間は 0~120 秒の変化範囲からランダムに選択する。ノードは、一時停止時間が終了すると改めて速度を変化範囲からランダムに選択し、目的地もランダムに地図上から選択して再び移動を開始する。

- 移動モデル：

歩行者は **shortest path map based movement model** に基づいて移動し、自動車は **Map route movement** に基づいて移動する。移動モデルの **warm up** 時間は 1000 秒である。図 9 に、地図上の道路を示す。ノードはその道路に沿って移動する。

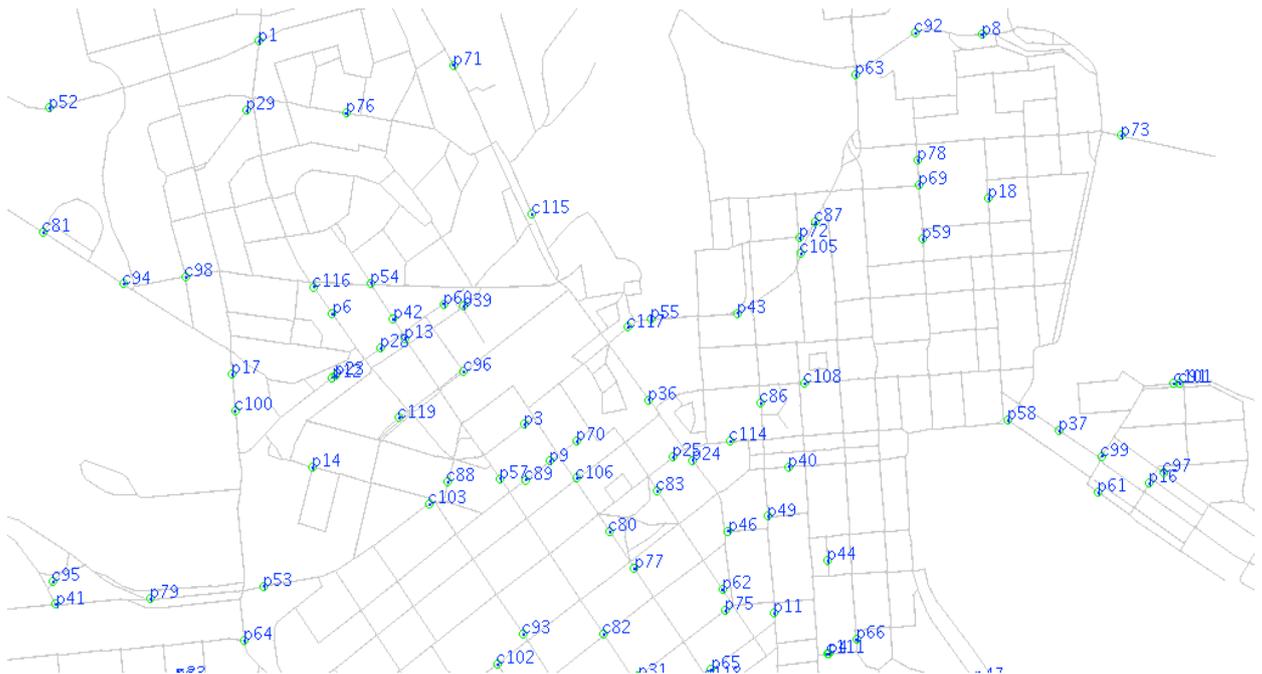


図 9 地図上の道路

- DTN のルーティング方式：

Epidemic と **PRoPHETv2** のルーティング方式を利用する。

- 通信条件：

各ノードの通信可能範囲は 10m であり、データ送信速度は 250KBps である。

- データ生成：

データの生成は、シミュレーション時刻が 1000 秒に達すると開始し、24000 秒になると停止する。データの生成間隔は 30~40 秒の変化範囲からランダムに選択する。また、データの送信元と宛先のペアは全てのノードからランダムに選択する。データのサイズは 300KB~600KB の変化範囲からランダムに選択し、データの TTL は 300 分とする。

- バッファ管理手法：

バッファが一杯になると、本研究では、**Oldest Drop** を用いてデータを廃棄する。

- バッファサイズ：

ノードのバッファサイズが有限である時は、DTN ルーティングの性能がバッファサイズによって変化すると考え、ノードのバッファサイズを 4MB から 7 MB まで変化させて DTN の性能を評価する。

表 1 シミュレーションの設定

Parameteres	Values
シミュレーション範囲	4500m * 3000m
シミュレーション時間	36000s
シミュレーション回数	シナリオごとに10回で行う
グループ1(G1)-Nodes types(speed)	80 Pedestrians(0.5~1.5m/s)
グループ2(G2)-Nodes types(speed)	40 Cars(2.7~ 13.9m/s)
G1の移動モデル	Shortest path map based movement model
G2の移動モデル	Map route movement
移動停止時間	[0,120s]
移動モデルwarm up時間	1000s
DTNルーティング方式	Epidemic PRoPHETv2($P_{max}=0.5$ 、 $\beta=0.9$ 、 $\gamma=0.999885791$)
通信可能範囲	10m
通信帯域	250kBps
データサイズ範囲	[300kB,600kB]
データの生成間隔	one new data every 30 to 40 seconds
バッファサイズ	4,5,6,7 (MB)
データのTTL	300(Min)
データ生成開始時刻	シミュレーション時間が1000sである時開始
データ生成停止時刻	シミュレーション時間が24000sである時停止
バッファ管理手法(データ廃棄方式)	Oldest Drop

3.4 実験結果

図10に、送信停止データ数の結果を示す。図中の Epidemic with time と PRoPHETv2 with time は、元の Epidemic と PRoPHETv2 に本研究で提案する通信可能時間に基づく送信制御を加えたものである。図10より、Epidemic with time と PRoPHETv2 with time は、Epidemic と PRoPHETv2 に比べて送信停止データ数は平均して3%以下になったと分かった。しかし、Epidemic with time と PRoPHETv2 with time でもまだ停止されたデータがあると言うことは、通信可能時間の計算に多少誤差があることを意味する。一方、提案方式ではこの誤差がデータ到達率を低下させないように、通信相手が宛先であるデータがあれば、通信可能時間 T には関係なく、最優先で送信する方式としている。したがって、送信停止データ数もその部分の送信失敗数も含めている。

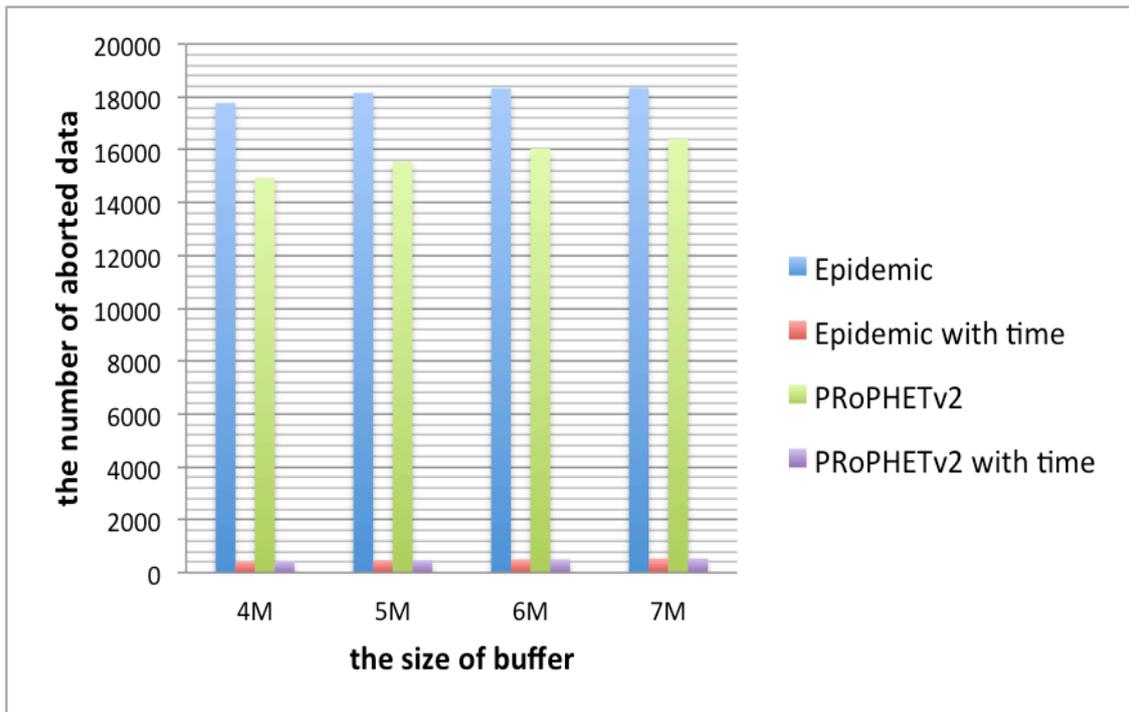


図 10 送信停止データ数 (the number of aborted data)

図 11 は DTN における最も重要な指標であるデータ到達率の結果を示す。図 11 より、Epidemic with time と PRoPHETv2 with time は、それぞれ Epidemic と PRoPHETv2 よりわずかに性能が悪いことが分かった。

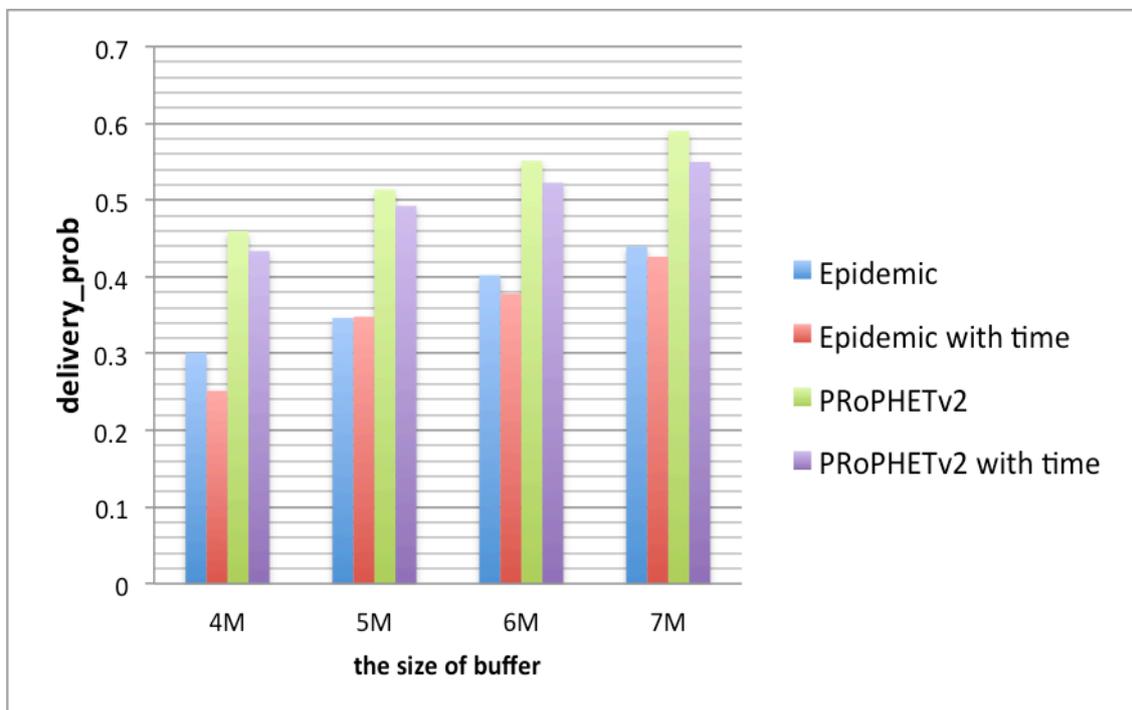


図 11 データ到達率 (delivery_prob)

次に、データ転送回数を図 12 に示す。Epidemic with time と PРоPHETv2 with time は、Epidemic と PРоPHETv2 に対してそれぞれ 12% 以上転送回数が増加した。これは、Epidemic with time と PРоPHETv2 with time では通信可能時間を考慮して送信するため、無駄なデータ送信時間が減り、結果としてデータを転送する機会が増えたためであると考えられる。

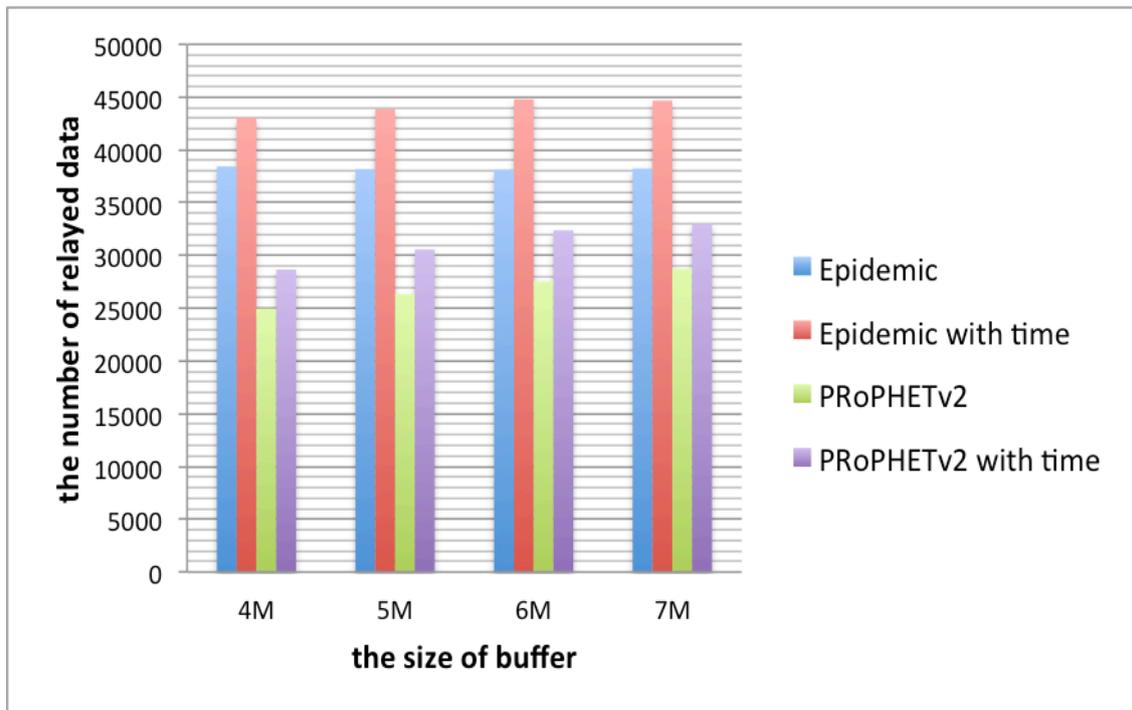


図 12 データ転送回数 (the number of relayed data)

通信できる機会が限られる DTN においては、データの転送回数を増やせばデータの到達率は高まると想定されるが、逆にデータ到達率は低下してしまった。この理由はデータ廃棄数から判明できる。図 13 にデータ廃棄数を示す。Epidemic はデータの複製を制限しないので、データの転送回数は PРоPHETv2 よりもかなり多い。その結果、ノードのバッファが満杯になりやすく、データの廃棄数も多くなる。図 13 を見ると Epidemic with time のデータ廃棄数は Epidemic より約 16% 以上多く、バッファが満杯になりやすいことがわかる。通信可能時間を考慮した送信制御方式はデータの転送回数を増加できるが、回数が多くなり過ぎ、かえってネットワークの負荷になってしまった。PРоPHETv2 はデータの複製を制限する方式であるので、Epidemic よりノードに残るバッファ空間大きく、バッファが満杯になりにくい。PРоPHETv2 with time でも PРоPHETv2 よりバッファ廃棄数は 10% 以上増加している。これは Epidemic with time と同じように、増加したデータ送信回数が多すぎ、ネットワークの負荷になってしまったためと考えられる。

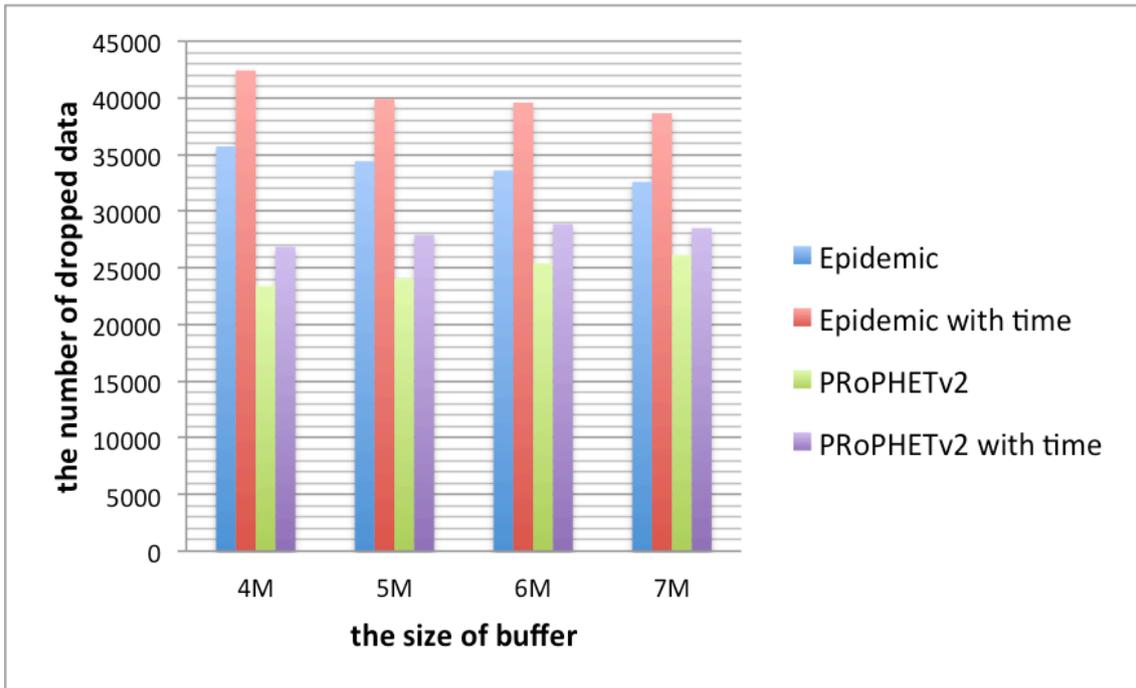


図 13 データ廃棄数 (the number of dropped data)

最後に、オーバーヘッド率を見て行く。図 14 にオーバーヘッド率の結果を示す。図により、Epidemic with time と PRoPHETv2 with time は Epidemic と PRoPHETv2 よりオーバーヘッド率が高いことがわかる。Epidemic with time と PRoPHETv2 with time はデータの廃棄数が多く、一個のデータを宛先まで送信するために多くのデータ転送を行っており、これがオーバーヘッド率の増加につながっている。しかし、リソースが非常に限られている DTN において、オーバーヘッド率が高いのは大きな問題である。

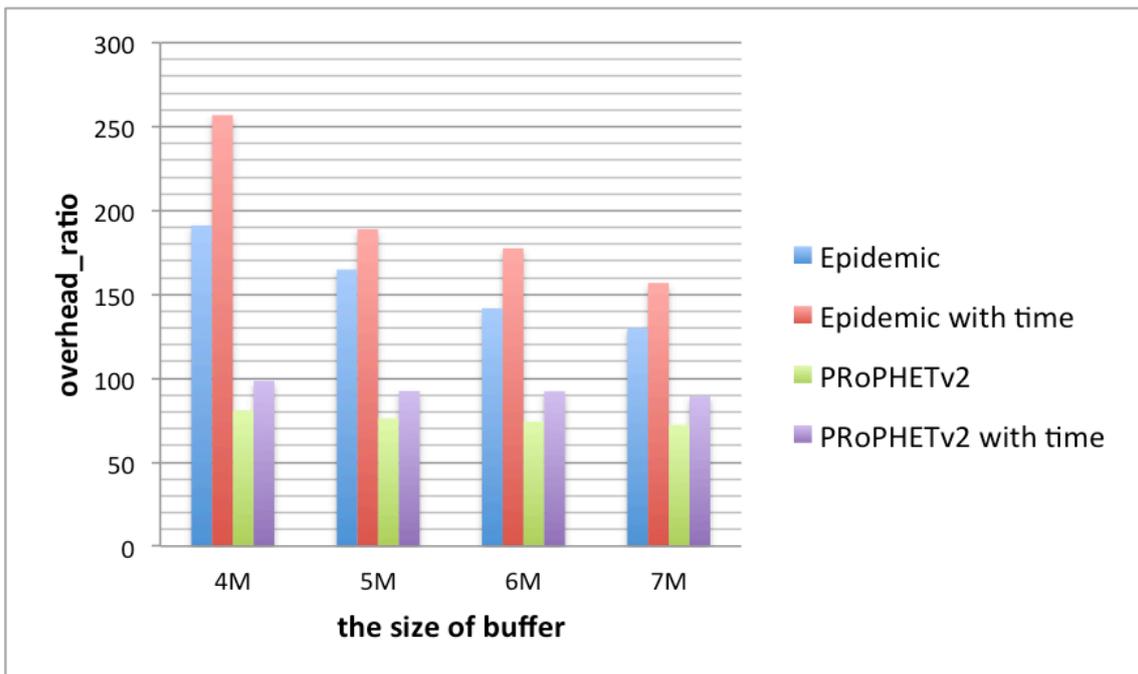


図 14 オーバーヘッド率 (overhead_ratio)

3.5 性能解析のまとめ

表 2 に、各指標の比較結果を示す。実験結果より、通信可能時間に基づく送信制御を行うと、データ到達率はかえって低下することが判明した。通信可能時間を考慮したことにより、データの送信チャンスは増加したが、データ送信量が多くなりすぎてバッファが満杯になり、データ廃棄数が増加して、未到達データを廃棄して DTN 性能が悪化したと考えられる。そこで、この問題を解決し、通信可能時間に基づく送信制御の効果を出せるように改良提案を行った。次章にて改良案の提案と評価を行う。

表 2 各指標の比較結果

方式	送信停止 データ数	データ転送 回数	データ 到達率	データ 廃棄数	オーバーヘッ ド率
Epidemic with timeは Epidemicより	97%減少 	10~18% 増加 	3~19% 減少 	16~19% 増加 	15~34% 増加 
PRoPHETv2 with time はPRoPHETv2より	97%減少 	15~18% 増加 	6~7% 減少 	14~17% 増加 	21~23% 増加 

第4章 通信可能時間に基づく送信制御方式における性能改善策の提案と評価

本章では、まず通信可能時間に基づく送信制御方式の改善策を述べ、実験環境を紹介する。そして、シミュレーションにより、改善策を評価する。最後に、実験結果を解析する。

4.1 通信可能時間に基づく送信制御方式の改善策

3章の通信可能時間に基づく送信制御方式の性能解析の結果により、通信可能時間に基づく送信制御方式は通信可能時間を有効的に利用し、データの転送量を増加したが、バッファ負荷も増大した。その結果、バッファが一杯になりやすく、データ廃棄数が増加し、未到達データを廃棄してDTN性能が悪化したと考えられる。本章では、その問題を解決するため、Vaccine回復手法を利用する。この考え方を選択する理由はいくつかがある。一つ目は、Vaccine回復手法はバッファが一杯になる前にデータを削除する方式である。リソースが非常に限られた環境においては、バッファが一杯になる前に、既に宛先に着いたデータを削除することで、バッファの負荷を減少できる。二つ目は、DTNにおいては各ノードが独立しているため、データが宛先に到達しても他のノードはそれを知る術がなく、他のノードは知らないままネットワーク内に既に到達したデータを広げていく。その結果、リソースを無駄に消費するだけでなく、他のデータを送信するチャンスも奪ってしまう。Vaccine回復手法は他のノードに対して受信確認パケット(anti-packet)を送信し、anti-packetを受信したノードがバッファから該データを破棄する方式であるので、前に説明した問題を解決できる。

更に森山は、バッファが満杯になりやすい環境においてはVaccine回復手法の効果が小さいことを明らかにし、Vaccine回復手法に対する改善手法[16]を提案している。バッファが満杯になりやすい環境では、anti-packetの拡散も遅くなるため十分な効果が得られない。そこで、森山はバッファの満杯を抑制する目的で、anti-packetを優先的に受け入れる流入制御を提案した。本研究においても、anti-packetを優先的に送信する考え方を利用し、通信可能時間に基づく送信制御方式で問題となったデータ転送回数の増加によるバッファの圧迫を緩和する。

これから、例を挙げて改善策を説明する。今、ノードAがノードBと通信可能になると仮定する。ノードAとノードBのやり取りは各ルーティング方式のプロトコルに従って行う。また、自身が既に持っているanti-packetを通信相手から受けることを防ぐため、ノードが保有するanti-packetのリストをサマリーベクターメッセージに入れて送信する。改良したデータ送信のフローチャートを図15に示す。今回も計算した通信可能時間Tは多少誤差があると考え、最初は通信可能時間Tを考慮せず、最優先でanti-packetを送信する。その後の手順は3.2で説明した手順と同じである。

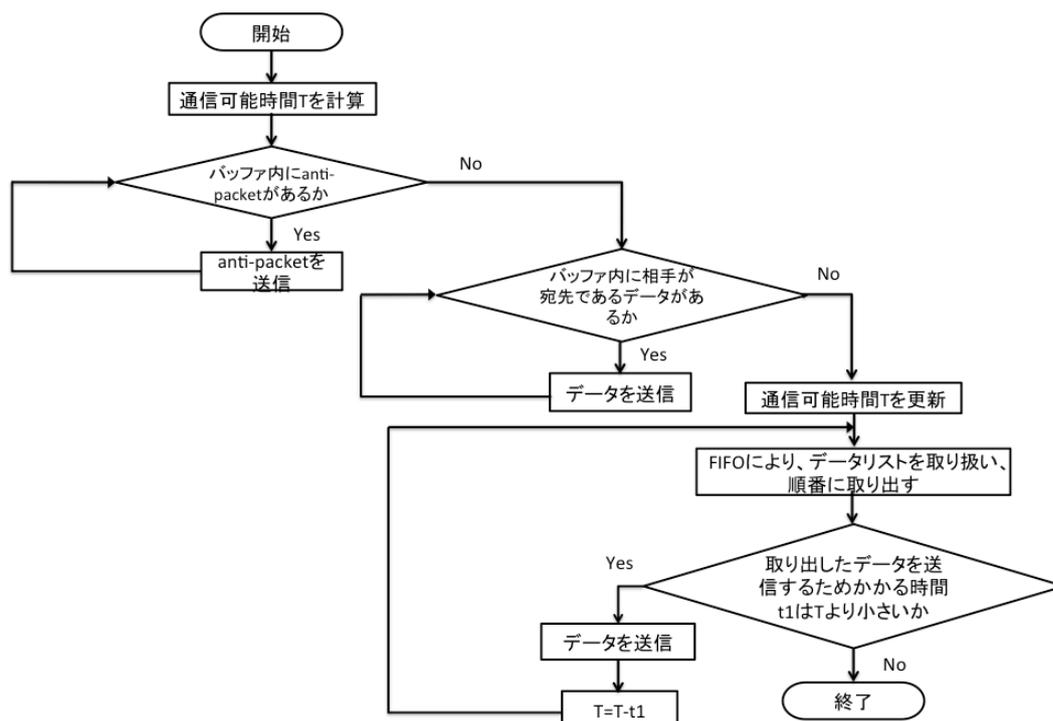


図 15 改良したデータ送信のフローチャート

4.2 シミュレーションの設定

実験の設定は、3章のシミュレーション設定と同じである。また、DTN 性能を評価する指標も 3章のシミュレーションと同じで、データ到達率、データ転送回数、データ廃棄数、オーバーヘッド率、送信停止データ数の 5つの指標で DTN 性能を評価する。比較対象としては、Epidemic は Epidemic、Epidemic with anti、Epidemic with anti-first、Epidemic with anti-first and time の 4つを用いる。一方、PRoPHETv2 についても PRoPHETv2、PRoPHETv2 with anti、PRoPHETv2 with anti-first、PRoPHETv2 with anti-first and time の 4つを比較する。なお、“with anti”は anti-packet を通常のデータと同じように送信すること、“with anti-first”は anti-packet を通常のデータより優先して送信することである。最後の“with anti-first and time”は、本研究で提案する方式であり、anti-packet を通常のデータより優先して送信し、通常のデータを送信する際は通信可能時間を考慮して時間内に送信を完了できるデータだけを送信する。

また、anti-packet のサイズは 1KB とし、データの TTL が 0 になると、ネットワークからそのデータ全ての複製を削除する。その際、削除したデータに対応する anti-packet がネットワーク内に残っても意味がないと考え、本研究では、各々の anti-packet にも TTL を設定する。anti-packet の TTL は、anti-packet を生成する時、データの残る TTL と同じである。そうすると、データの TTL が切れる時には、その anti-packet の TTL も切れる。データとデータの anti-packet が一緒にネットワークから削除される。

4.3 改善策の評価

図 16 に、通信可能時間の不足により送信停止データ数の結果を示す。提案した Epidemic with anti-first and time と PRoPHETv2 with anti-first and time では、バッファサイズに

関わらず、送信停止データ数は Epidemic と PRoPHETv2 の 4%以下に減少した。また、Epidemic with anti-first と PRoPHETv2 with anti-first の 6%以下に減少した。

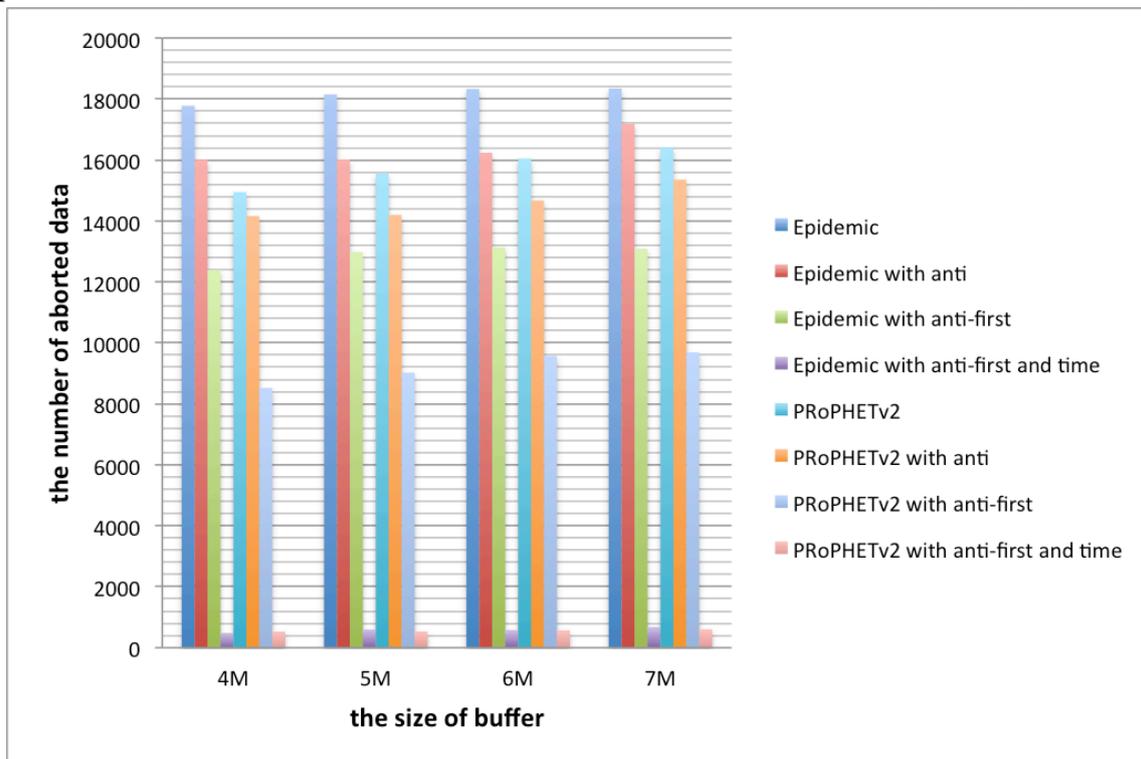


図 16 送信停止データ数(the number of aborted data)

図 17 に、DTN における最も重要な指標であるデータ到達率の結果を示す。全ての方式において、バッファサイズが増えるとデータ到達率は上昇した。特に、anti-packet を優先する with anti-first と with anti-first and time は、既存のルーティング方式や with anti と比較してかなり性能が良くなった。また、with anti-first and time では、Epidemic with anti-first より少なくとも 8%以上到達率が向上し、特にバッファサイズが 5MB である時、18%到達率を向上していた。一方、PRoPHETv2 with anti-first and time では、バッファサイズが小さくても、PRoPHETv2 with anti-first より平均約 7%性能を向上できていた。

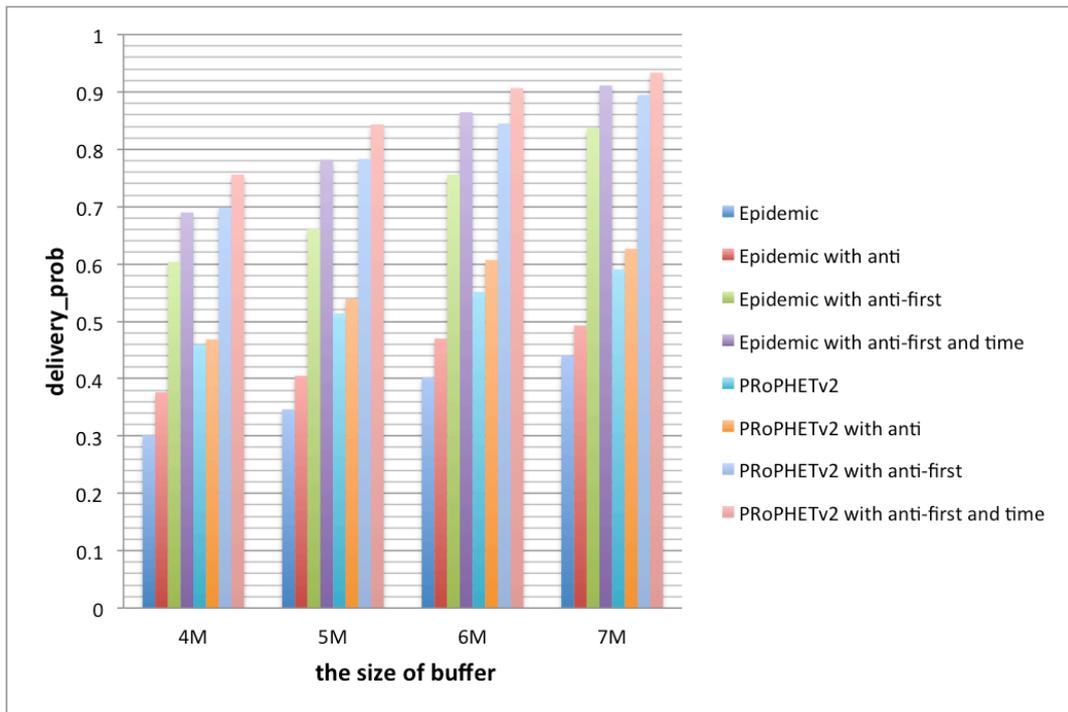


図 17 データ到達率(delivery_prob)

次に、データ転送回数を見ていく。図 18 にデータ転送回数を示す。Epidemic と PRoPHETv2 のデータ転送回数が一番多く、with anti-first のデータ転送回数が一番少ない。一方、with anti-first and time は、二番目に転送回数が少ないが、with anti-first と殆ど同等であった。

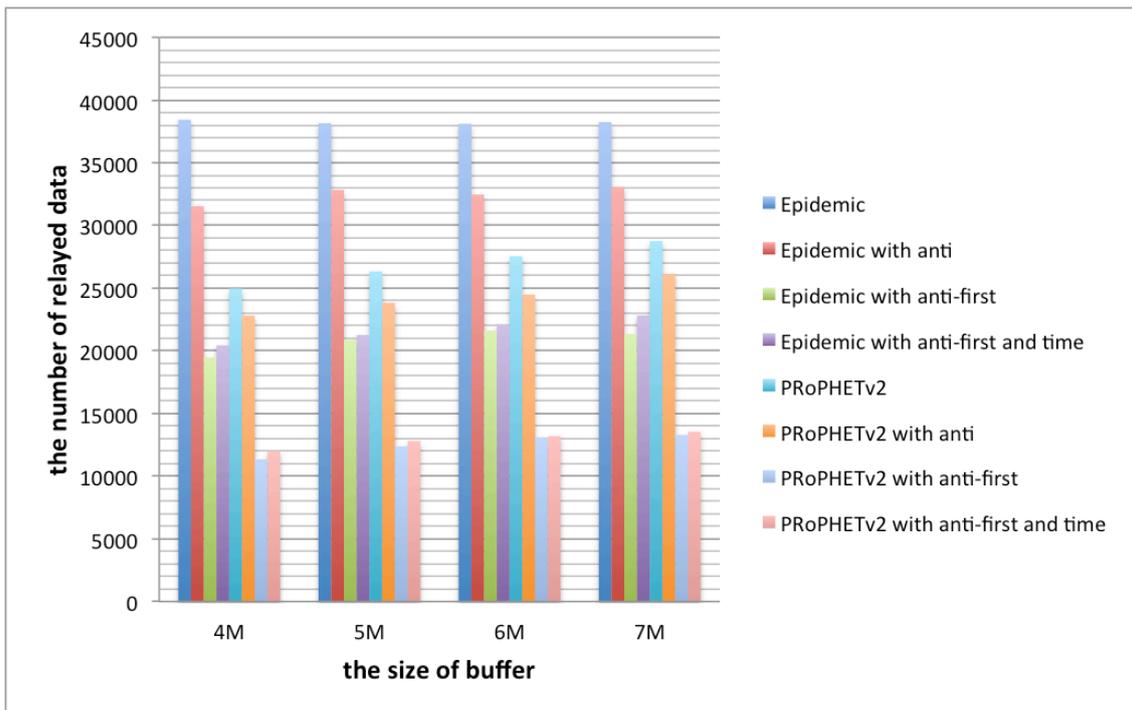


図 18 データ転送回数(the number of relayed data)

図 19 にデータ廃棄数を示す。バッファのサイズが増加すると共に、各方式のデータ廃棄数が減少していた。一番少ないのは with anti-first and time 方式である。Epidemic with anti-first and time のデータ廃棄数は、Epidemic with anti-first よりも 20%以上減少していた。一方、PRoPHETv2 with anti-first and time は PRoPHETv2 with anti-first より 16%以上減少し、特にバッファサイズが大きい時、データ廃棄数が約 45%減少していた。

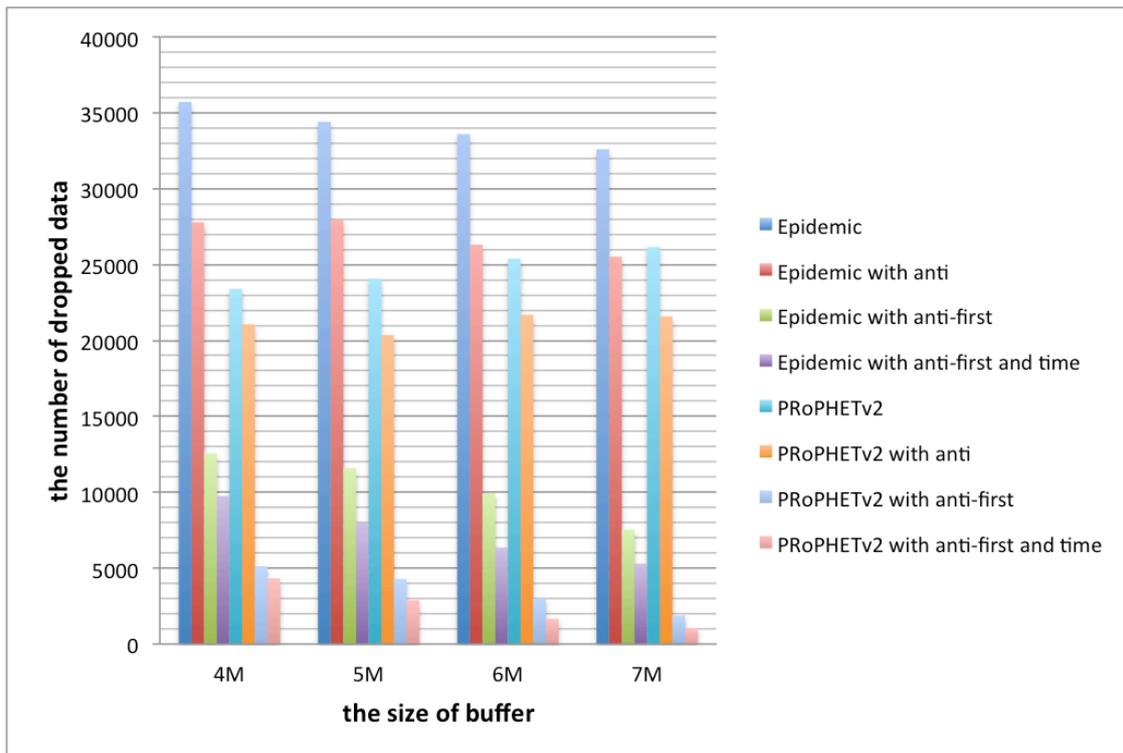


図 19 データ廃棄数(the number of dropped data)

最後に、オーバーヘッド率を見て行く。図 20 にオーバーヘッド率を示す。バッファサイズが増加すると共に、with anti-first のオーバーヘッド率は with anti-first and time のオーバーヘッド率と近づいていたが、一番少ないのは with anti-first and time である。

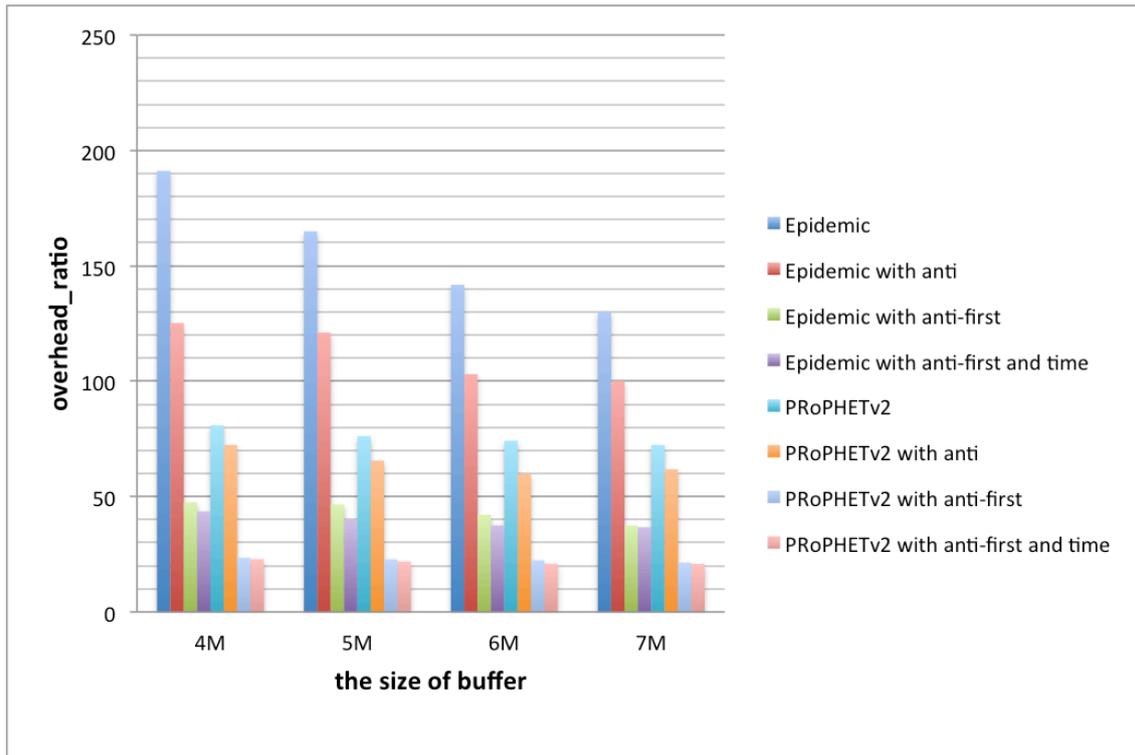


図 20 オーバーヘッド率(overhead_ratio)

4.4 実験結果の解析

表 3 に、各指標の比較評価を示す。結果により、改善策は各面から DTN の性能を向上できることが明らかになった。これから、それぞれの指標ごとに見ていく。

1) データ転送回数

図 12 を見ると、with time は既存のルーティング方式よりデータ転送回数が 12%以上増加している。一方、図 18 より、with anti-first and time のデータ転送回数は既存のルーティング方式より 40%以上減少している。with anti-first and time のデータの転送回数が減った理由は以下の 2 つである。

- 一つは anti-packet を送信するために通信可能時間の一部を利用したため、通常データを送信できる時間が少なくなった。
- もう一つは anti-packet により他のノードにデータ到達情報を届けたため、無効なデータ転送が減少した。

また、with anti-first and time のデータ転送回数は with anti-first のデータ転送回数より多いので、通信可能時間に基づく送信制御方式は有効に機能したと言える。

2) データ廃棄数

図 13 より、Epidemic with time は、Epidemic よりデータ廃棄数が 16%以上増加し、PRoPHETv2 より 10%以上増加することが分かった。一方、図 19 を見ると、いずれのルーティング方式においても anti-first とすることによりデータ廃棄数は激減し、anti-first with time とするとデータ廃棄数は更に減少している。Epidemic with anti-first and time のデータ廃棄数は Epidemic with anti-first よりも 20%以上減少し、PRoPHETv2 with anti-first and time のデータの廃棄数は PRoPHETv2 with anti-first より 16%以上減少している。

その理由は、バッファが一杯になる前であれば、通信可能時間に基づく送信制御がネットワーク内のデータ転送回数を増加させても影響はなく、むしろデータが宛先に到達する可能性が増大する。データが宛先に到着すると anti-packet がネットワーク内に拡散され、各ノードのバッファ内に残っている当該データが一掃されてバッファ負荷は軽減され、さらに通信可能時間に基づく送信制御によって無効なデータ転送が減るためにデータが宛先に着く可能性が増大する。その結果、with anti-first and time のデータ廃棄数は with anti-first のそれより少なくなる。

3) データ到達率

図 11 より、Epidemic with time と PRoPHETv2 with time のデータ到達率は Epidemic と PRoPHETv2 のそれぞれより若干低下していることが分かった。一方、図 17 より、いずれのルーティング方式でも、with anti を with anti-first に変えることでデータ到達率は大きく向上し、with anti-first and time とすると到達率はさらに向上した。結果として、Epidemic with anti-first and time は既存の Epidemic より約 2 倍以上、PRoPHETv2 with anti-first and time は既存の PRoPHETv2 より約 1.5 倍以上の性能になっている。この性能改善効果はバッファサイズが小さいほど大きいことも判明した。また、Epidemic with anti-first and time では、性能がいい Epidemic with anti-first よりも 8%から 18%の性能向上を確認できた。また、PRoPHETv2 with anti-first and time では PRoPHETv2 with anti-first より平均約 7%性能を向上できていた。

表 3 各指標の比較結果

方式	送信停止 データ数	データ転 送回数	データ 到達率	データ 廃棄数	オーバ ヘッド率
Epidemic with antiは Epidemicより	6~12% 減少 ↘	14~18% 減少 ↘	12~25 増加 ↗	19~22% 減少 ↘	23~35% 減少 ↘
Epidemic with anti-firstは Epidemicより	28~30% 減少 ↘	43~49% 減少 ↘	88~94% 増加 ↗	65~77% 減少 ↘	70~75% 減少 ↘
Epidemic with anti-first and timeはEpidemicより	96~97% 減少 ↘	42~47% 減少 ↘	1.1~1.3 増加 ↗	73~84% 減少 ↘	72~77% 減少 ↘
PRoPHETv2 with antiは PRoPHETv2より	5~9% 減少 ↘	9~11% 減少 ↘	2~10% 増加 ↗	10~17% 減少 ↘	11~24% 減少 ↘
PRoPHETv2 with anti-first はPRoPHETv2より	40~43% 減少 ↘	52~55% 減少 ↘	52~53% 増加 ↗	78~93% 減少 ↘	70~71% 減少 ↘
PRoPHETv2 with anti-first and timeはPRoPHETv2より	96~97% 減少 ↘	51~53% 減少 ↘	58~65% 増加 ↗	82~96% 減少 ↘	71~72% 減少 ↘

第5章 結論

本研究は、ノード間の通信可能時間に基づく送信制御により DTN の性能を向上させることを目指して行ったものである。

既存のルーティング方式として Epidemic と PRoPHETv2 を例に取り、これに提案方式である通信可能時間に基づく送信制御や anti-packet を加えた方式について、データ転送回数、データ廃棄数、データ到達率、オーバーヘッド率、送信停止回数の 5 つの性能指標をシミュレーションにより比較評価した。

先ず、既存方式とそれに通信可能時間に基づく送信制御を加えた方式を比較したところ、送信制御を行うと既存方式より却って性能が悪くなった。各 DTN 性能指標の解析により、この性能低下は、通信可能時間に基づく送信制御方式は通信可能時間を有効に利用し、データ転送回数が増加してバッファ負荷も増大し、その結果データ廃棄数が増加し、未到達データを廃棄して DTN 性能が悪化したことが原因であると判明した。

改善策として anti-packet を優先的に送信する方式(with anti-first and time)を加え、既存ルーティング方式、既存方式の anti-packet あり(with anti)、既存方式の anti-packet 優先方式 (with anti-first) の 3 つの方式を比較対象としてシミュレーションを行った。その結果、改善策は他の 3 つの方式より性能を向上させることができた。特に、DTN において最も重要な指標であるデータの到達率は、改善した Epidemic は既存 Epidemic の 2 倍以上になり、改善した PRoPHETv2 も既存 PRoPHETv2 の 1.5 倍以上にすることができた。通信可能時間に基づく送信制御は、上記の 3 つの方式の中で最も性能が良い with anti-first よりも更に平均 7%性能を向上できた。

一方、本研究では、5 つの DTN 性能指標を用いて方式評価を行ったが、まだ評価していない到達遅延時間 (latency)、ホップ数 (hop count)、データ保留時間 (buffering time) についても評価を行い、更なる性能改善策を検討することが今後の課題である。

また、anti-packet の送信が通常データ送信を圧迫することも問題である。anti-packet の送信も限られた通信可能時間やバッファ容量を占有するため、たとえ anti-packet のサイズが非常に小さくても、量が多いとネットワークに負荷を与え、DTN 性能を低下させる。anti-packet を逆方向に送信されるデータパケットに相乗りさせる等、改善策の検討が必要である。

謝辞

本研究を行うに際し、筑波大学大学院図書館情報メディア研究科の川原崎雅敏教授には、研究目標の設定から研究手順、そして本稿執筆に至るまで、手厚い御指導を戴いた。ここに深謝の意を表す。また、技術面、そして精神面で研究をサポートして下さった筑波大学ユビキタスネットワークング研究室の土屋俊貴、同研究室の皆様、そして卒業後であるにも拘らず、本研究に対する助言を下された同研究室 OB の田中大地先輩に深い感謝を表す。

参考文献

- [1] FALL, Kevin. A delay-tolerant network architecture for challenged internets. In: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications. ACM, 2003. p. 27-34.
- [2] BURLEIGH, Scott, et al. Delay-tolerant networking: an approach to interplanetary internet. IEEE Communications Magazine, 2003, 41.6: 128-136.
- [3] 鶴正人, et al. DTN 技術の現状と展望. 電子情報通信学会 通信ソサイエティマガジン, 2011, 2011.16: 16_57-16_68.
- [4] K.Scott,S.Burleigh,BundleProtocolSpecification,RFC5050,TheMITRECorporation, NASA Jet Propulsion Laboratory, Nov. 2007.
- [5] A. Vahdat, D. Becker, Epidemic routing for partially connected ad hoc networks, Technical Report CS-200006, Duke University, Apr. 2000.
- [6] Grasic, S., Davies, E., Lindgren, A., & Doria, A. (2011, September). The evolution of a DTN routing protocol-PRoPHETv2. In Proceedings of the 6th ACM workshop on Challenged networks (pp. 27-30). ACM.
- [7] Delay tolerant networking research group: <http://www.dtnrg.org/>. viewed on dec 20th 2011.
- [8] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-Tolerant Networking Architecture. RFC 4838 (Informational), April 2007.
- [9] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, page 259. ACM, 2005.
- [10] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," IEEE/ACM Transactions on Networking, vol. 16, no. 1, pp. 77–90, 2008.
- [11] A. Lindgren, A. Doria and O. Schelen, "Probablistic routing in intermittently connected networks", LNCS, 3126, pp. 239–254 (2004).
- [12] Haas, Z. J., & Small, T. (2006). A new networking model for biological applications of ad hoc sensor networks. IEEE/ACM Transactions on Networking, 14(1), 27-40.
- [13] Li, Y., Zhao, L., Liu, Z., & Liu, Q. (2009). N-drop: Congestion control strategy under epidemic routing in DTN. In Research center for wireless information networks, Chongqing University of posts & telecommunication, Chongqing 400065, china (pp. 457–460). ACM.
- [14] Keränen, Ari, Jörg Ott, and Teemu Kärkkäinen. "The ONE simulator for DTN protocol evaluation." Proceedings of the 2nd international conference on simulation tools and techniques. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.
- [15] Keranen, Ari. "Opportunistic network environment simulator." Special Assignment report, Helsinki University of Technology, Department of Communications and Networking (2008).

[16] 森山京平(2012).DTNにおける優先制御と流入制御を用いたバッファ管理手法の提案.奈良先端科学技術大学院大学情報科学研究科情報処理学専攻修士論文(未公刊).