The University of Osaka
Institutional Knowledge Archive

Title	モバイルアドホックネットワークにおける蓄積運搬転送型ルーチング方式に関する研究
Author(s)	木村, 共孝
Citation	大阪大学, 2015, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/52213
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

博士学位論文

モバイルアドホックネットワークにおける蓄積運搬転送型ルーチング方式に関する研究

木村 共孝

2015年1月

大阪大学大学院工学研究科
電気電子情報工学専攻

内容梗概

本論文は，著者が大阪大学大学院工学研究科電気電子情報工学専攻在学中に行ったモバイルアドホックネットワークにおける蓄積運搬転送型ルーチング方式に関する研究成果をまとめたものである．

近年，送信元端末から受信先端末までの経路上において大きな遅延が発生したり，通信接続が頻繁に途切れたりするような劣悪な環境下においても通信を行いたいという要望が高まっている．このような劣悪な環境では，送受信端末間に安定したパスが存在しないため，インターネットで用いられる TCP/IP (Transmission Control Protocol/ Internet Protocol) などの従来の通信技術では，通信を行うことができない．そこで，DTN (Delay/Disruption/Disconnect Tolerant Networking) 技術と呼ばれる新たな技術について近年，活発に研究が進められている．

本論文では，DTN 技術の一つである蓄積運搬転送型ルーチング（Store-Carry-Forward Routing）に着目する．蓄積運搬転送型ルーチングは端末間の接続性の弱いモバイルアドホックネットワークにおいて用いられ，端末の移動を利用してメッセージの配送を行う．各端末は新たに発生したメッセージや中継メッセージを受信すると一旦，それらをバッファに蓄積する．移動中それらのメッセージを保持し，別の端末と交信可能になった際にメッセージの転送を行う．この手順を繰り返すことによって最終的に受信先端末へメッセージを配送できる．

メッセージを受信先端末へ配送できる可能性を高めるために，メッセージの複製を転送することが検討されている．メッセージを保持する端末が別の端末に遭遇するとそのメッセージの複製を転送し，その後，両方の端末がメッセージを保持する．複数の端末がメッセージの複製を保持するため，複製を持ついずれか一つの端末が受信先端末に遭遇すればメッセージの配送が完了する．よって，メッセージの複製を転送すればするほど受信先端末へ小さな配送遅延でメッセージ配送できる．しかし，蓄積運搬転送型ルーチングが用いられる環境ではバッファやエネルギーなどのネットワーク資源が限られているため，遭遇した全ての端末にメッセージの複製を転送する単純な手法では，ネットワーク資源がすぐに枯渇し，メッセージの配送を全く行えなくなる．したがって，配送遅延の増加を抑えつつ，如何にネットワーク資源消費を削減するかが蓄積運搬転送型ルーチングにおける重要な技術課題である．

この技術課題を解決するために，これまで多くのルーチング方式が考案されてきた．蓄積運搬転送型ルーチング方式が考え始められた初期の頃には全ての端末のモビリティが同質な状況を想定し，ホップ数や複製生成数など各端末において容易に把握できる情報に基づいてメッセージの複製の転送を制御するルーチング

方式が検討されてきた．しかし，端末のモビリティが同質でない場合，遭遇する端末や場所に偏りが生じる．そこで，これらの偏りを表す特徴量を遭遇履歴情報から推定し，その推定結果に応じてメッセージの複製の転送を制御するルーチング方式が考案されている．遭遇履歴情報から推定可能な情報の具体例としては，端末間の遭遇率，遭遇時間間隔，端末密度などが挙げられる．本論文では，遭遇履歴情報から推定可能なこれらの情報の総称を「ネットワーク情報」と呼ぶことにする．さらに，初期の頃に考えられた各端末において容易に把握可能な情報に基づいてメッセージの転送を制御する方式を「ネットワーク情報を用いないルーチング方式」，遭遇履歴から推定可能な情報に基づいてメッセージの転送を制御する方式を「ネットワーク情報を用いるルーチング方式」と呼ぶことにする．

ネットワーク情報の利用が有効である状況とほとんど役に立たない状況が存在するため，本論文では，ネットワーク情報を用いる方式と用いない方式のそれぞれについて新たなルーチング方式を検討する．遭遇する端末や場所の偏りの存在が明らかな場合には，それらの偏りを表すネットワーク情報を用いることによって効率的なメッセージ配送を実現できる．しかし，想定どおりの偏りが存在しない場合，効率的なメッセージ配送を実現できる可能性が低いだけでなく，そのネットワーク情報を推定するために不要な計算負荷を被る．よって，ネットワーク情報がほとんど役に立たない場合やメッセージ配送の効率性を多少犠牲にしても良い場合，ネットワーク情報を用いないルーチング方式によってメッセージ配送を行う方が良い．

ネットワーク情報を用いないルーチング方式に関する既存の研究では，設定パラメタがないが，達成可能なメッセージ配送の効率性が低い方式，もしくは，達成可能なメッセージ配送の効率性は高いが，設定パラメタに対する感度が高い方式が考案されてきた．ネットワーク情報を用いずに設定パラメタを適切に設定することは困難であるため，ネットワーク情報がほとんど役に立たない場合には，設定パラメタに対する感度が低く，かつ，達成可能なメッセージ配送効率が高いルーチング方式が有用である．そこで，本論文では，複製の生成権利を有する端末を制限するルーチング方式を提案し，このルーチング方式が上記の定性的な特性を有することを数学モデルを用いた解析とリアルトレースデータを用いたシミュレーション実験によって示す．

また，ネットワーク情報を用いるルーチング方式の既存の研究では，遭遇する端末の偏りを表す特徴量に着目してメッセージの複製の転送を制御しており，遭遇する場所の偏りを表す特徴量に着目したルーチング方式は少ない．しかし，場所によってネットワーク環境が大きく異なる状況では，それぞれの場所に適したメッセージの複製の転送を制御した方が良い．例えば，場所によって端末間の遭遇頻度に偏りがある場合，遭遇頻度が小さな領域においてはメッセージの複製を生成する機会が少ないため積極的に複製を転送した方が良い．一方で，端末間の遭遇頻度が大きな領域においてはメッセージの複製を生成する機会が多いため，複製の転送を抑制したとしても配送遅延の増加にほとんど影響を与えない．このように場所毎に偏りがある場合には，遭遇場所の特徴量を用いたルーチング方式が

有用である．

そこで，本論文では，遭遇場所の特徴量の一つである端末密度に着目し，端末密度を用いた新たなルーチング方式を提案する．端末密度を推定するために，各端末が GPS (Global Positioning System) などの位置測定技術によって自身の位置を計測できる状況を想定し，各端末が別の端末と遭遇する度に遭遇地点情報を記録する．さらに，記録した遭遇地点情報を遭遇時に端末間で交換する．このようにして収集した遭遇地点情報に基づいて端末密度の推定を行い，その推定結果に応じてメッセージの複製の制御することで効率的なメッセージ配送を行えることをシミュレーション実験によって示す．

本論文は，以下に示す 5 章によって構成される．

第 1 章では，まず DTN 技術とその課題について述べる．次に，蓄積運搬転送型ルーチングについて詳しく述べる．

第 2 章では，複製の生成権利を有する端末を制限するマルチスプレッダルーチング方式を提案する．まず，マルチスプレッダルーチング方式のシステムのダイナミクスを有限状態吸収マルコフ連鎖として定式化し，このマルコフ連鎖の吸収状態への初到達時間として得られる平均配送遅延やネットワーク資源消費量を数値計算によって求める．次に，平均配送遅延とネットワーク資源消費量の間にあるトレードオフの評価を行う．さらに，リアルトレースデータを用いたシミュレーション実験によって既存のネットワーク情報を用いないルーチング方式との性能比較を行う．

第 3 章では，各端末が位置測定技術を用いることのできる状況を想定し，端末密度推定に基づく確率的転送方式を提案する．この方式では，端末密度に応じてメッセージの転送確率を制御する．また，端末密度を推定する複数の端末密度推定法について述べる．シミュレーション実験によって端末密度推定に基づく確率的転送方式の有効性を示す．さらに，端末密度推定の誤差がルーチング性能に与える影響について議論する．

第 4 章では，端末密度を考慮した効用型ルーチング方式について述べる．この方式では，端末密度と効用値に基づいてメッセージの複製の転送制御を行う．遭遇場所と遭遇場所の両方の情報を考慮することによって，エネルギー消費量に関して単独の情報を用いるよりも効率的なメッセージ配送を行えることをシミュレーション実験によって示す．

最後に第 5 章において本論文の結論を述べる．

謝辞

本論文は筆者が大阪大学大学院工学研究科の博士後期課程において研究した成果をまとめたものです．研究成果をまとめるに当たり，多くの方々のご支援とご指導を賜りました．ここに，研究過程においてお世話になった方々に心からの誠意と感謝を申し上げます．

本論文の主査として，また本論文に関する研究の全過程を通じ，懇切丁寧なるご指導，ご鞭撻を賜った 大阪大学大学院工学研究科電気電子情報工学専攻 滝根 哲哉 教授に深甚なる感謝の意を表し，心より厚くお礼申し上げます．本研究を遂行するに当たり行き詰まりを感じる事が幾度となくありましたが，そのたびに暖かく，かつ，適切な助言をしていただきました．時間を惜しむことなく，本研究に関する議論をしていただけたことに深く感謝いたします．大阪大学工学部の 4 回生のときから 8 年間にも渡る長い間，ありがとうございました．

本論文の執筆にあたり，副査として査読をしていただいた 大阪大学産業科学研究所 駒谷 和範 教授に心よりお礼申し上げます．筆者の文字通りの拙稿に対して，的確なご指摘，丁寧なご助言をいただきました．

本論文の執筆にあたり，同じく副査として査読していただいた 大阪大学工学研究科電気電子情報工学専攻 松田 崇弘 准教授に厚くお礼申し上げます．松田 准教授には本論文に関わる研究の全過程において数多くの有益なご助言をいただきました．研究へのご指導のみならず，生活面においても定期的に声を掛けていただき，精神的に大きな支援をしていただきました．親身になって様々な相談に乗って下さったことに深く感謝いたします．

大阪大学大学院工学研究科において，ご指導，ご教授を賜った 大阪大学大学院工学研究科電気電子情報工学専攻 北山 研一 教授，馬場口 登 教授，三瓶 政一 教授, 井上 恭 教授，鷲尾 隆 教授を始めとする各教官の方々には，様々なご助言，ご提案，ご指摘を頂戴いたしました．心よりお礼申し上げます．

本研究を遂行するにあたり，熱心なご指導，的確なご助言を頂いた 奈良先端科学技術大学院大学 の 笹部 昌弘 准教授に深甚なる感謝の意を表します．また，公私に渡って様々なご助言をしていただいた 関西大学 の 平田 孝志 准教授，ならびに， 摂南大学 の 工藤 隆則 講師に厚くお礼申し上げます．

ロバストネットワーク工学領域研究室において様々なご支援を頂いた 下屋敷 優美 事務補佐員，ならびに 橋本 幸子 前事務補佐員，そして公私に渡って暖かいご助言，ご支援を賜った 後藤 嘉代子 元技官に心よりお礼申し上げます．

卒業生を含め，ロバストネットワーク工学領域の諸氏には，日頃より多くのご助言，ご協力を頂き，種々の面でお世話になりました．特に，井上 文彰 さん，城

之内 達郎 さん，松浦 翼 さんとは研究に関する相談や議論を多くさせていただき，有意義な時間を過ごすことができました．ここに深謝申し上げます．

目次

内容梗概 iii

謝辞 vii

第 1 章　序論 1
1.1 DTN 技術 1
1.1.1 DTN 技術が利用される状況 1
1.1.2 DTN 技術とその課題 2
1.2 蓄積運搬転送型ルーチング 5
1.2.1 従来の MANET におけるルーチングとの相違点 6
1.2.2 技術課題 7
1.2.3 ネットワーク情報を用いないルーチング方式 9
1.2.4 ネットワーク情報を用いるルーチング方式 13
1.2.5 研究目的 18
1.3 論文の構成 19

第 2 章　マルチスプレッダルーチング方式の提案とその評価 21
2.1 緒言 21
2.2 VACCINE 回復手法を用いたマルチスプレッダルーチング方式 . . . 22
2.2.1 マルチスプレッダルーチング方式 22
2.2.2 VACCINE 回復手法 22
2.2.3 端末におけるメッセージ処理手順 23
2.3 マルコフ解析 23
2.3.1 数学モデル 23
2.3.2 メッセージの生存期間における吸収マルコフ連鎖 24
2.3.3 システム状態の半順序集合 29
2.4 性能評価 32
2.4.1 平均配送遅延と平均メッセージ生存時間 32
2.4.2 バッファ使用量 33
2.4.3 エネルギー消費量 34
2.4.4 トレードオフの評価 36
2.4.5 既存のネットワーク情報を用いないルーチング方式との性能比較 40
2.4.6 リアルトレースデータを用いたシミュレーション結果 44

2.5 結言 46

第 3 章 端末密度推定に基づく確率転送方式の提案とその評価 51
3.1 緒言 51
3.2 端末密度推定に基づく確率転送方式 52
3.2.1 提案方式の概要 52
3.2.2 端末密度推定法 53
3.2.3 メッセージ転送基準 56
3.3 性能評価 56
3.3.1 シミュレーションモデル 56
3.3.2 端末密度推定の結果 57
3.3.3 端末密度を考慮した確率転送方式の性能評価 65
3.4 結言 66

第 4 章 端末密度を考慮した効用型ルーチング方式の提案とその評価 71
4.1 緒言 71
4.2 システムモデル 72
4.3 効用型ルーチング方式 72
4.3.1 メッセージ転送の手順 72
4.3.2 効用値の計算手順 72
4.4 端末密度を考慮した効用型ルーチング方式 73
4.4.1 提案方式のメッセージ転送手順 73
4.5 性能評価 74
4.5.1 シミュレーションモデル 75
4.5.2 性能評価指標 76
4.5.3 確率 p_{LD} の影響 76
4.5.4 確率 p_{HD} の影響 78
4.6 結言 81

第 5 章 結論 87

付 録 A マルチスプレッダルーチング方式における状態遷移率 89
A.1 状態 $(\boldsymbol{x}, k, 0)$ からの遷移率 89
A.2 状態 $(\boldsymbol{x}, k, m)$ からの遷移率 90

付 録 B マルチスプレッダルーチング方式の性能評価 93
B.1 平均メッセージ生存時間 93
B.2 システム内の平均メッセージ累積数 95
B.3 平均複製転送回数 98

付 録 **C** 端末密度分布の分散 **101**
C.1 パラメトリック密度推定法 **101**
C.2 ノンパラメトリック密度推定法 **102**

付 録 **D** ガウスカーネル密度推定法 **105**
D.1 遭遇地点情報リスト $\mathcal{L}_v$ の更新手順 **105**
D.2 平滑化パラメタ h の影響 **105**

参考文献 **109**

研究業績 **119**

図 目 次

1.1 インターネットと DTN におけるプロトコル階層 **3**
1.2 惑星間インターネットのメッセージ配送過程 **4**
1.3 異種ネットワーク間のルーチング **5**
1.4 蓄積運搬転送型ルーチングを用いたメッセージ転送 **7**
1.5 従来の MANET におけるルーチングを用いたメッセージ転送 . . . **7**
1.6 ネットワーク情報を用いないルーチング方式（黒丸は送信元端末，二重丸は受信先端末，斜線の丸は中継端末，白丸はメッセージを持たない端末を表す） **12**
1.7 設定パラメタに対する感度と達成可能なメッセージ配送の効率性の関係 **13**
1.8 ネットワーク情報を用いるルーチング方式 **15**

2.1 $\mathcal{C}_P$ の生成手順 **25**
2.2 $\mathcal{C}_7$ の生成手順（$\mathcal{C}_7 \backslash \mathcal{J}_7$ 内から次に選択される要素を下線で示す） . **25**
2.3 $\mathcal{X}_P(0)$ の生成手順（$\boldsymbol{x}'_j$ は式 (2.3) で与えられる） **28**
2.4 $\mathcal{X}_P(m)$ と $\mathcal{S}_P(m)$ の生成手順（$\boldsymbol{x}'_j$ は式 (2.3) で与えられる） **29**
2.5 $\mathcal{Y}_P(m)$ と $\mathcal{A}_P(m)$ の生成手順 **30**
2.6 $\mathcal{S}_P(0)$ に含まれる実行可能な状態 $(\boldsymbol{x}, k, 0)$ の数 **32**
2.7 $\mathrm{E}[T_D]$ を計算するための後進再帰手順 **33**
2.8 平均配送遅延 $\mathrm{E}[T_D]$ **34**
2.9 一つのメッセージの発生によってランダムに選ばれた端末が被る端末負荷 O_L **35**
2.10 平均複製転送回数 $\mathrm{E}[N_D]$ **36**
2.11 ある一つのメッセージに関する情報の平均送信回数 $\mathrm{E}[N_I]$ **37**
2.12 バッファ使用量に関するトレードオフ指標 I_{BR} **38**
2.13 エネルギー消費量に関するトレードオフ指標 I_{EC} $(\alpha = 0.02)$ **39**
2.14 最適パレートフロント: I_{BR} と I_{EC} の関係 **39**
2.15 平均配送遅延 $\mathrm{E}[T_D]$ **41**
2.16 ランダムに選ばれた端末が一つのメッセージの発生によって被る端末負荷 O_L **42**
2.17 バッファ使用量に関するトレードオフ指標 I_{BR} **42**
2.18 平均複製回数 $\mathrm{E}[N_D]$ **43**
2.19 エネルギー消費に関するトレードオフ指標 I_{EC} **43**

2.20 不到達率 **47**
2.21 平均配送遅延 $\mathrm{E}[\overline{T}_D]$ **48**
2.22 バッファ使用量に関するトレードオフ指標 I_{BR} **49**

3.1 端末 v が端末 u に遭遇した際の手順 (p_F は転送確率を表す) **53**
3.2 グループ $\mathcal{V}_i$ $(i = 2, 3, \ldots, 10)$ 内の端末が移動する領域 **57**
3.3 密度モデルにおける端末密度分布の推定結果 **59**
3.4 均質密度モデルにおける端末密度分布の推定結果 **60**
3.5 非均質密度モデルにおける高密度領域 ($d_v = \mathrm{E}[I_v] + \sigma[I_v]$ の場合) . . **61**
3.6 非均質密度モデルにおける高密度領域 ($d_v = \mathrm{E}[I_v] + \sigma[I_v]/10$ の場合) **62**
3.7 均質密度モデルにおける高密度領域 ($d_v = \mathrm{E}[I_v] + \sigma[I_v]$ の場合) . . **63**
3.8 均質密度モデルにおける高密度領域 ($d_v = \mathrm{E}[I_v] + \sigma[I_v]/10$ の場合) . **64**
3.9 非均質密度モデルにおける平均複製転送回数 $\mathrm{E}[N_D]$ に対する平均配送遅延 $\mathrm{E}[T_D]$ の変化（受信先端末が低密度領域から選択される場合） **68**
3.10 非均質密度モデルにおける平均複製転送回数 $\mathrm{E}[N_D]$ に対する平均配送遅延 $\mathrm{E}[T_D]$ の変化（受信先端末が高密度領域から選択される場合） **69**
3.11 均質密度モデルにおける平均複製転送回数 $\mathrm{E}[N_D]$ に対する平均配送遅延 $\mathrm{E}[T_D]$ の変化 **70**

4.1 端末密度を考慮した効用型ルーチング方式の手順 **74**
4.2 領域 $\mathcal{A}_i$ $(i = 1, 2, \cdots, 5)$ **75**
4.3 グループ $\mathcal{V}_j$ $(j = 2, 3, 4, 5)$ における目的地点の状態遷移図 **76**
4.4 $\mathrm{E}[N_D]$ に対する $\mathrm{E}[T_D]$ の変化 ($p_{\mathrm{HD}} = 0$ の場合) **80**
4.5 $\mathrm{E}[N_D]$ に対する $\mathrm{E}[T_D]$ の変化 ($p_{\mathrm{LD}} = 1$ の場合) **83**

B.1 $\mathrm{E}[T_E]$ を計算するための後進再帰手順 **95**
B.2 平均メッセージ生存時間 $\mathrm{E}[T_E]$ **96**
B.3 回復手法を用いてメッセージを削除するのに要する平均時間 $\mathrm{E}[T_R]$ **97**

D.1 遭遇履歴情報 $\mathcal{L}_v$ の更新手順 **106**
D.2 推定された端末密度分布 $I_v(\boldsymbol{x})$ ($h = 100, 500$ の場合) **107**
D.3 閾値 d_v を上回る $I_v(\boldsymbol{x})$ の領域（$h = 100, 500$ の場合） **108**

表 目 次

1.1 ネットワーク情報を用いた方式の分類 **14**
1.2 ネットワーク情報を用いるルーチング方式 **17**

2.1 状態 $(\boldsymbol{x}, k, 0) \in \mathcal{S}_P(0)$ からの状態遷移先とその遷移率 **28**
2.2 $\mathcal{S}_P(m)$ $(m \geq 2)$ に含まれる状態 $(\boldsymbol{x}, k, m)$ からの遷移先とそれらの遷移率 **28**
2.3 I_{BR} に対する最適パラメタ ($N = 100$ の場合) **41**

3.1 ウォームアップ期間の長さ **66**

4.1 シミュレーションシナリオ **76**
4.2 シナリオ $\mathcal{V} \to \mathcal{V}$ における送受信端末ペア $(\mathcal{S}, \mathcal{D})$ の割合 **81**

記号一覧

$\mathcal{A}$	閉領域
$\mathcal{A}_P(m)$	アンチパケット数が m（$m = 2, 3, \ldots, N+1$）となった直後のシステムの実行可能な状態の集合
c_i	$\mathcal{C}_P$ の i（$i = 1, 2, \ldots, n(P)$）番目に大きな要素
$\mathcal{C}_P$	全ての実行可能なカウンタ値の集合
h	ガウスカーネル密度推定法における平滑化パラメタ
I_{BR}	バッファ使用量に関するトレードオフ指標
I_{EC}	エネルギー消費に関するトレードオフ指標
$I_v(\boldsymbol{x})$	端末 v が推定した地点 $\boldsymbol{x}$ における端末密度
$K(t)$	時刻 t（$t \geq 0$）におけるオリジナルとメッセージの複製の総数
L	定常状態においてランダムに選ばれた端末が保持しているメッセージ数
$\mathcal{L}_v$	端末 v が保持する遭遇地点情報リスト
$M(t)$	時刻 t $(t \geq 0)$ におけるアンチパケットの総数
N	送信元端末を除くネットワーク内の端末数
N_D	ネットワーク内からメッセージが消滅するまでにメッセージの複製が転送された回数
N_I	メッセージの消滅までにそのメッセージに関する情報が送信された回数
$N_R^{[\boldsymbol{x},k]}$	状態（$\boldsymbol{x}, k, m$）におけるメッセージを持つ端末の数
$N_S^{[\boldsymbol{x}]}$	状態（$\boldsymbol{x}, k, m$）における散布端末の総数
O_L	一つのメッセージの発生によってランダムに選択された端末が被る負荷
P	散布端末の最大数
$\mathcal{S}_P(m)$	アンチパケット数が m $(m = 0, 2, 3, \ldots, N+1)$ のときのシステムの実行可能な状態の集合
T_D	メッセージの配送遅延
T_E	メッセージの生成から消滅までに要する時間
T_R	回復手法によってメッセージの複製の削除に要する時間
$U_v(d)$	端末 d に対する端末 v の効用値

$\mathcal{V}$	ネットワーク内の端末の集合
$X_j(t)$	時刻 t においてカウンタ値 c_j のメッセージを持つ散布端末の数
$\mathcal{X}_P(m)$	アンチパケット数が m（$m = 0, 2, 3, \ldots, N+1$）のときの散布端末の実行可能な状態の集合
$\mathcal{Y}(m)$	アンチパケット数が m （$m = 2, 3, \ldots, N+1$）となった直後の散布端末の実行可能な状態の集合
λ	二つの端末が遭遇する率
$\boldsymbol{\mu}_i$	パラメトリック密度推定法における平均ベクトル
ρ	ランダムに選択された端末が持つ平均メッセージ数
Σ_i	パラメトリック密度推定法における共分散行列

第1章

序論

近年，送信元端末から受信先端末までの経路上において大きな遅延が発生したり，通信接続が頻繁に途切れるような劣悪な環境においても通信を行いたいという要望が高まっている．このような環境では，送受信端末間に安定したパスが存在しないため，インターネットで用いられる TCP/IP (Transmission Control Protocol/Internet Protocol) などの従来の通信技術では，通信を行うことができない．そこで，DTN (Delay/Disruption/Disconnect Tolerant Networking) 技術 [20] と呼ばれる新たな技術について近年，活発に研究が進められている．本章では，まず，DTN 技術について解説する．次に，DTN 技術の代表例である蓄積運搬転送型ルーチング（Store-Carry-Forward Routing）に着目し，その研究動向について述べる．最後に，本論文の研究目的と構成について述べる．

1.1 DTN 技術

本節では，まず，DTN 技術が利用される状況について述べる．次に，そのような状況において通信を実現する DTN 技術とその課題について解説する．

1.1.1 DTN 技術が利用される状況

DTN 技術の研究は，惑星間インターネット [8, 53] から始まった．惑星間インターネットでは，地球上のインターネットと地球外の惑星上に構成された通信ネットワークを人工衛星などを介して相互に接続することを目的とする．この目的を達成するために必要な DTN 技術は以下の二つに大きく分けられる．

- 長い遅延や断続的な通信接続の切断に耐え得る通信技術
- 異なるプロトコルを用いるネットワーク間の相互接続技術

惑星間インターネットでは，地球や惑星上の端末から人工衛星までの伝搬遅延が長く，また人工衛星の空間的な位置関係によって断続的な通信切断が生じるため，前者の技術が必要となる．また，地球や惑星上の端末と人工衛星の間の通信には

TCP/IP と異なるプロトコルが用いられる．さらに，惑星上のネットワークが，惑星の環境をセンシングを行う無線端末によって構成される場合，この惑星上のネットワーク内においても TCP/IP と異なるプロトコルを用いて通信が行われる．よって，地球上のインターネットの端末と惑星のネットワーク上の端末の間の通信を確立するためには後者の技術が必要となる．

惑星間インターネットのみならず，地球上の様々な通信ネットワークにおいても大きな遅延の発生や断続的な通信接続の切断は生じる．そこで，地球上の通信ネットワークに DTN 技術を用いることが検討されている．DTN 技術の利用によって通信が可能となる状況の代表例として大規模災害の発生直後 [68]，発展途上国の通信インフラ未整備の状況 [59]，車車間通信 [58]，深海探査 [57]，野生動物調査 [37] などが挙げられる．

特に，近年，日本では大規模災害発生後に DTN 技術を利用することが注目されている [68]．大規模災害発生直後には安否情報や被災状況の確認を行いたいという要望が高まるが，通信設備の破壊や過剰な通信接続が発生するため，必要な情報を取得できないという問題が生じる．実際に，2011 年 3 月に起きた東日本大震災では，基地局などの通信設備が津波によって破壊されたため，携帯電話機を持っていたとしてもほとんど通信を行えない状況となった．また，通信設備が破壊されずに残った地域においても，基地局に過度な接続が集中し通信を行えない状況となった．そこで，DTN 技術を用いることによって，通信設備が破壊された地域における通信の確立や通信設備が残った地域においてはデータのオフロードの達成が期待されている．

1.1.2 DTN 技術とその課題

これまでにさまざまな DTN 技術が議論されてきた．DTN 技術の例としては，異種ネットワークの相互接続技術 [11, 18, 66]，輻輳制御 [18, 19]，フロー制御 [63]，ルーチング [10, 36, 75, 53, 81, 89]，アプリケーション層プロトコルの設計 [84]，セキュリティ [9, 39, 67, 77] が挙げられる．

異種ネットワークの相互接続技術 異なるプロトコルを用いるネットワークを相互に接続するために，バンドル層（Bundle Layer）の導入が提案されている [66]．バンドル層は，従来のインターネットにおけるプロトコル階層のアプリケーション層とトランスポート層の間に挿入される（図 1.1 を参照）．バンドル層の PDU (Protocol Data Unit) はバンドルと呼ばれる．バンドルのヘッダ部分には送受信端末の識別子など通信に必要な全ての情報が書き込まれ，ペイロード部分には ADU (Application Data Unit) であるメッセージの情報が格納される．バンドルとメッセージは実質的にほとんど変わらないため，これらを区別する必要がない場合には，本論文ではバンドルをメッセージと呼ぶことにする．

バンドル層が実装される端末は，DTN ホスト，DTN ルータ，DTN ゲートウェイの三種類に分類できる．DTN ホストはメッセージの送信元端末，あるいは，受信先端末である．DTN ルータ や DTN ゲートウェイは，別の端末との接続が常に

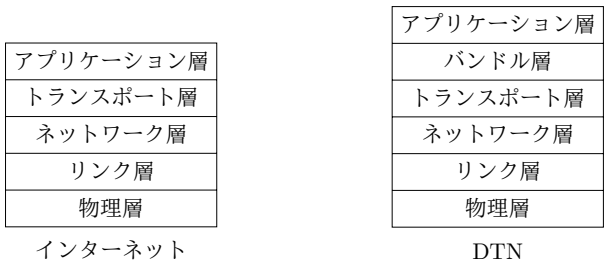


図 1.1: インターネットと DTN におけるプロトコル階層

は存在せず間欠的に生じる状況で用いられ，転送先となる端末と通信が確立するまで永続ストレージ（persistent storage）にバンドルを保持し続ける．DTN ルータが一つのネットワーク内においてバンドルの中継を担うのに対して，DTN ゲートウェイは複数のプロトコルスタックを持ち異種ネットワーク間の中継を担う．

どのように異種ネットワーク間をメッセージが転送されていくか説明するために，地球上の端末から別の惑星上の端末へのメッセージ配送の例を用いる．図 1.2 は惑星間インターネットにおけるメッセージの配送過程を表している．送信元ホストから地球上の送信局までは TCP/IP が用いられるため，図中では省略しているが複数のルータがパケットを中継し，送信局に必要なパケットが全て到達した際にバンドルの再構成が行われる．地球上の送信局から惑星上の受信局までは TCP/IP と異なったプロトコルによってメッセージが転送されるため，地球上の送信局は二つのプロトコルスタックを持つ DTN ゲートウェイである．送信局と人工衛星は空間的な位置関係によって通信を行えないことがあるため，接続が長時間途切れている間は送信局の永続ストレージにバンドルが蓄積される．さらに，惑星上の受信局から受信先ホストへの通信プロトコルは宇宙空間で用いるプロトコルと異なったものを用いるため，惑星上の受信局も DTN ゲートウェイである．図中では省略しているが，惑星上のネットワークが移動端末によって構成されている場合などには全ての端末が DTN ルータとなるため，複数の DTN ルータを介してメッセージが受信先端末に配送される．

異種ネットワークの相互接続を実現する上で重要な課題は，名前解決である．インターネットでは，通信が開始される際に受信先端末の名前を DNS (Domain Name System) を用いて IP アドレスに変換し，名前解決を行う．このような方式は事前名前解決（early binding）方式と呼ばれる．しかしながら，DTN 技術が用いられる環境では，通信の開始時に DNS サーバのような名前解決を行う端末への接続が困難であったり，ネットワーク毎に名前の命名規則が異なっているなどの問題があり，事前名前解決方式を用いることができるとは限らない．

この問題を解決するために，受信先端末の識別子を所属するネットワーク名と端末名の組によって表現する [11, 18]．このように識別子を表現することによって，受信先端末と異なるネットワーク内の端末がバンドルを受け取った際には，名前解決を行わずに受信先端末の所属ネットワーク名のみを用いてバンドルの転送を行える．すなわち，受信先端末のネットワーク名の参照後，受信先端末の所属ネットワークに近づく DTN ゲートウェイにバンドルを転送する．これを繰り返すこ

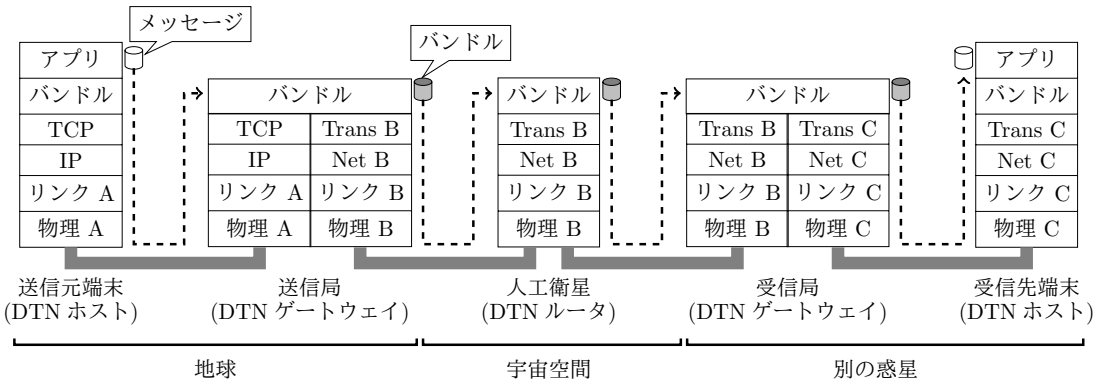


図 1.2: 惑星間インターネットのメッセージ配送過程

とで，最終的に受信先端末が所属するネットワークの DTN ゲートウェイにバンドルを転送できる．そして，受信先端末と同じネットワークに所属する DTN ゲートウェイがバンドルを受信した際に初めて，端末名をアドレスに変換する．この名前解決の方法は，逐次名前解決（late binding）方式と呼ばれる．

逐次名前解決方式を実現するためには，各 DTN ゲートウェイにおいて同一ネットワーク内の端末の名前解決を行える必要があるが，ネットワーク内の端末の存在の有無をどのように把握するかや，端末の名前とアドレスの組をどのように管理するかなどが課題とされている [20].

輻輳制御 DTN ルータや DTN ゲートウェイへのメッセージ転送が増加すると永続ストレージが枯渇する可能性があるため，この資源競合を制御する必要がある．DTN 環境における輻輳制御では，メッセージの配送完了の確認応答（acknowledgment）をどのように行うかに大きく影響を受ける．確認応答を受け取った端末はメッセージを破棄し，永続ストレージ内に蓄積されるメッセージの数を減らすことで，ストレージの資源競合を緩和できる．そこで，DTN 技術が用いられる環境に適したさまざまな確認応答の方法が議論されている [18, 19].

フロー制御 TCP によるフロー制御はエンド端末によって行われるが，DTN におけるフロー制御ではホップ毎に制御する必要がある．すなわち，メッセージを持つ端末が，次の中継先となる端末への転送レートを制御する．ホップ毎の制御であるため，輻輳制御と完全に独立した問題とみなすことができず，輻輳制御を考慮したフロー制御が必要とされている [19, 63].

ルーチング DTN におけるルーチングは，複数の異種ネットワーク間のルーチング [53] であり，単独のネットワーク内におけるルーチング [10, 36, 75, 81, 89] に分類できる．

前者は，送信元端末の所属ネットワーク内の DTN ゲートウェイから受信先端末の所属ネットワーク内の DTN ゲートウェイまでの経路を決定する（図 1.3 を参照）．バンドルを受信した DTN ゲートウェイが，ヘッダ部分から受信先端末の所属ネットワークのアドレスを参照し，そのネットワークに近づく DTN ゲートウェイにバンドルを転送する．

後者は，一つのネットワーク内にある複数の DTN ルータを介して如何にメッ

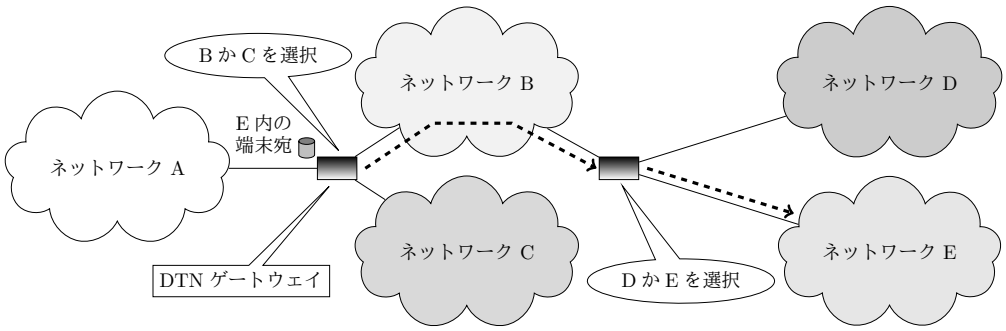


図 1.3: 異種ネットワーク間のルーチング

セージを中継させるかを決定する．DTN 技術が用いられる環境ではネットワークに関する情報の取得が困難であることが多いため，従来のネットワークに対するルーチングよりも技術的に解決しなければならない課題が多くあり，活発に研究が進められている．本論文では，後者のルーチングに注目し，1.2 節で技術課題やこれまでに提案されているルーチング方式について詳しく述べる．

アプリケーション層プロトコルの設計 DTN におけるアプリケーション層プロトコルの設計では，アプリケーションが利用される様々なシチュエーションを包括して設計を行う必要があり，従来のインターネットにおけるアプリケーションプロトコルの設計よりも困難が伴う．

セキュリティ DTN では十分に管理されていない不特定の端末を介してメッセージが配送される可能性が高いため，機密性を保証することが難しい．また，中継途中においてメッセージを容易に改竄できるため，完全性の保証についても困難が伴う．これらの課題を解決するために，インターネットで用いられている公開鍵暗号方式を用いてメッセージを暗号化したり，ディジタル署名を付加することが考えられるが，DTN では信頼できる第三者機関（Trusted Third Party）と常時接続ができないため，公開鍵の配布が困難であるなどの問題があり，実用化に向けて独自のセキュリティ技術の開発が必要とされている [9, 39, 67, 77].

1.2 蓄積運搬転送型ルーチング

本論文では，端末間の接続性の弱い間欠的接続モバイルアドホックネットワーク（Intermittently Connected Mobile Ad-Hoc Network）に焦点を当てる．このネットワークでは送信元端末から受信先端末へのパスがほとんどいかなる時刻においても存在しない．しかしながら，ネットワーク内の端末が移動するため，時折，端末が別の端末と遭遇し，その端末間において局所的な通信接続が確立される．そこで，この間欠的な通信接続を利用した蓄積運搬転送型ルーチング (Store-Carry-Forward Routing) が提唱されている（図 1.4 を参照）．メッセージを受信した端末は，一旦そのメッセージをバッファに蓄積する．端末の移動中，そのメッセージを保持しながら運搬し，別の端末と遭遇した際にメッセージの転送を行う．この手順を繰りかえすことによって最終的に受信先端末へメッセージを配送できる．また，蓄

積運搬転送型ルーチングでは，受信先端末を除く全ての端末がメッセージを中継する DTN ルータとなる可能性がある．

これまで，受信先端末が複数あるマルチキャスト通信 [23] や複数の端末のうちのいずれか一つにメッセージ配送すれば良いエニキャスト通信 [24] についても議論されているが，本論文では，受信先端末が一つであるユニキャスト通信についてのみ考える． また，一部の端末が他の端末よりも高性能であり，それらの端末を有効に利用することも考えられているが [86]，本論文では全ての端末の通信性能がほとんど等しい状況に焦点を当てる．

本節では，まず，蓄積運搬転送型ルーチングと従来のモバイルアドホックネットワーク（MANET: Mobile Ad-hoc NETwork）で用いられるルーチングの違いを明らかにする．次に，蓄積運搬転送型ルーチングの技術課題について述べる．さらに，蓄積運搬転送型ルーチングを，遭遇履歴情報から推定可能なネットワーク内の情報を利用するか否かによって分類し，それぞれの代表的な蓄積運搬転送型ルーチング方式について解説する．最後に，本論文の研究目的を述べる．

1.2.1 従来の MANET におけるルーチングとの相違点

従来の MANET で用いられるルーチング方式の代表例として AODV (Ad-hoc On Demand Vector) [60] や DSR (Dynamic Source Routing) [35] が挙げられる．これらのルーチング方式では，送受信端末間にパスが存在することを前提として設計されており，マルチホップ転送によってメッセージが受信先端末に届けられる（図 1.5 を参照）．従来の MANET ルーチングでは，端末の移動によるトポロジーの変化に対応することはできるのだが，送受信端末間のパスが存在しない場合にはメッセージ配送を全く行えない．

一方で，蓄積運搬転送型ルーチングでは，送受信端末間のパスが存在しない場合においても，メッセージを配送することができる．しかしながら，メッセージの転送先と通信を確立できない場合，メッセージはバッファに長期間蓄積されるため，従来の MANET ルーチングと比較して大きな配送遅延となる傾向がある．また，従来の MANET ルーチングでは送受信端末間のパスの存在を確認した上で通信が行われるためメッセージ配送の信頼性が高いのだが，蓄積運搬転送型ルーチングを用いる場合には，メッセージを配送できる保証はなく，メッセージ配送の信頼性が低い．

なお，接続性の強い端末群と弱い端末が混在しているネットワーク環境では，これらのルーチングを併用することが検討されている [29]．接続性の強い端末群には従来の MANET ルーチングを用いて通信を行い，接続性の弱い端末に対してのみ蓄積運搬転送型ルーチングを適用することが可能である．

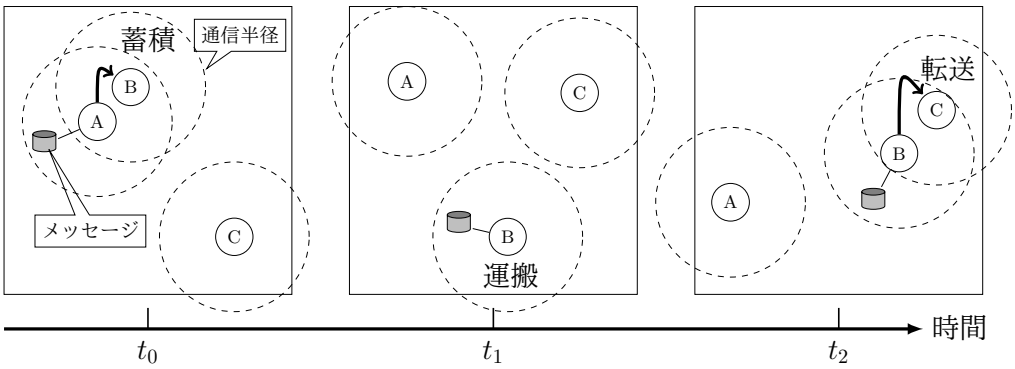


図 1.4: 蓄積運搬転送型ルーチングを用いたメッセージ転送

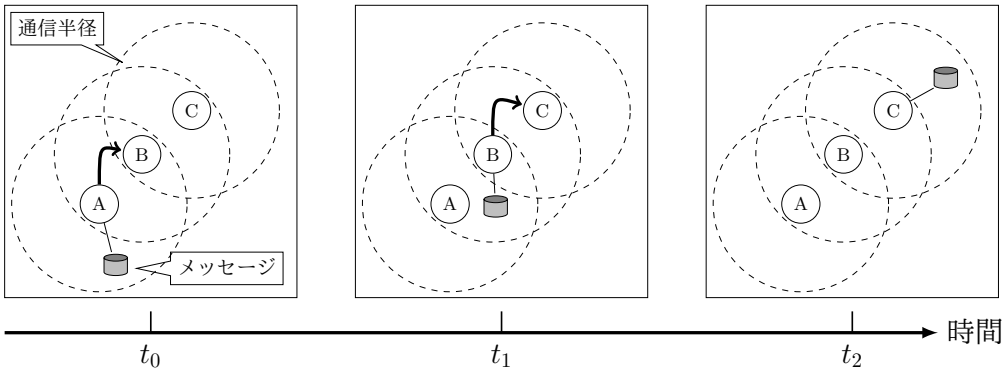


図 1.5: 従来の MANET におけるルーチングを用いたメッセージ転送

1.2.2 技術課題

蓄積運搬転送型ルーチングによるメッセージ配送が困難な要因の一つは，ネットワーク資源が限られていることである．メッセージ配送に用いられる主なネットワーク資源としては，バッファ容量，エネルギー，帯域が挙げられる．

バッファ容量 端末が搭載する永続ストレージの容量に依存し，保持できるメッセージの最大数を決定する．一般に，DTN ではメッセージを保持している時間が長く，かつ，メッセージサイズも大きいため，端末が大きなストレージを搭載していたとしてもバッファ溢れが生じる可能性がある．バッファ溢れを防ぐためには，重要度の低いメッセージを破棄する必要がある．保持するメッセージの中から重要度の低いメッセージを選択する手法は，バッファ管理（Buffer Management）手法 [42] と呼ばれる．

エネルギー 端末が持つバッテリーに蓄えられており，メッセージの転送，受信，あるいは中継先の選択時に消費される．一般に，DTN では端末が保持できるエネルギーの最大量が少ないため，エネルギー消費を抑えたメッセージ配送を行うことが重要である．また，近年，太陽光パネルなどのエネルギーを補給する装置を端末に搭載することが考え始められており，エネルギー補給量を考慮したメッセージ配送が検討されている [48].

帯域 端末の遭遇時に転送可能なメッセージ数を決定する．帯域が小さい場合，

全てのメッセージを転送できないため，1 回の遭遇で転送できるメッセージ数を見積もったり，重要なメッセージから転送を行う必要がある．また，帯域が十分大きかったとしても，遭遇時の接続時間が短い場合には全てのメッセージを転送できないため，転送可能なメッセージ数は端末のモビリティにも大きく依存する．

このようなネットワーク資源に制約のある環境において，蓄積運搬転送型ルーチングの性能の良し悪しを評価する指標の代表例として，メッセージの不到達率，配送遅延，ネットワーク資源消費量が挙げられる．

不到達率 発生した総メッセージ数に対する受信先端末へ到達しなかったメッセージの割合である．蓄積運搬転送型ルーチングは劣悪な通信環境下で用いられるため，最も重要な評価指標である．

配送遅延 送信元端末で発生したメッセージを受信先端末に配送するまでに要する時間である．不到達率が十分小さな状況において重要となる性能評価指標である．

ネットワーク資源消費量 メッセージの配送で消費される主なネットワーク資源には，バッファとエネルギーがある．バッファ消費量を評価するためには，各端末が保持する平均メッセージ数が用いられる．一方で，エネルギー消費量を見積もるために，メッセージの転送回数が用いられることが多い．これは，メッセージの転送が通信によって生じるエネルギー消費の主因であるからである．

一般に，不到達率とネットワーク資源消費量の間にはトレードオフの関係があり，ネットワーク資源を消費すればするほど不到達率が小さくなる．また，不到達率が十分小さな場合には配送遅延が重要な指標になるが，配送遅延とネットワーク資源消費量の間にもトレードオフの関係がある．したがって，如何にこれらのトレードオフのバランスのとれた効率の良いメッセージ配送を実現できるかが蓄積運搬転送型ルーチングの技術課題である．

この技術課題を解決するために，これまでさまざまなルーチング方式が考案されてきた [10, 36, 75, 81, 89]．蓄積運搬転送型ルーチングが考え始められた初期の頃には，ネットワーク内の各端末が互いに遭遇可能で，端末間の遭遇頻度も等しい環境を想定し，単純なルーチング方式が考えられてきた．具体的なルーチング方式については 1.2.3 節で紹介するが，これらのルーチング方式は遭遇履歴情報を用いることによって推定できるネットワークに関する情報をほとんど利用しないという特徴を持つ．

端末が同質な想定環境は，端末のモビリティとしてランダムウェイポイント（Random Waypoint）[6] やランダムディレクション（Random Direction）[54] を用いることによってモデル化できる．これらのモビリティモデルの場合，任意の二つの端末の遭遇時間間隔は指数分布に従うことが示されている [25, 26]．したがって，これらのルーチング方式を評価する際にはシミュレーション実験によって得られる定量的な結果に加えて，数学モデルを用いた解析的な評価もでき，ルーチングの定性的な特性についても明らかにされている．

しかしながら，実際の多くのネットワークでは，端末ペアの一部が全く遭遇できないことや端末間の遭遇頻度に偏りがあることが大半である [22]．このようなネットワークにおいて初期の頃に考えられた単純なルーチング方式を用いてメッ

セージを配送することはできるが，効率的なメッセージ配送を行える可能性は極めて低い．

そこで，近年，端末のモビリティやネットワーク資源量などのネットワークに関する情報を遭遇履歴情報から推定し，その推定結果に応じてメッセージの転送を制御するルーチング方式が考えられている．遭遇履歴情報から推定可能な情報の具体例としては，端末間の遭遇頻度，遭遇時間間隔や端末密度などが挙げられる．以降では，遭遇履歴情報から推定可能な情報の総称を「ネットワーク情報」と呼ぶことにする．ネットワーク情報を用いるルーチング方式の具体例については 1.2.4 節で紹介するが，これらのルーチング方式は用いられる環境を想定し，その環境下において効率的なメッセージ配送を行えるように設計されている．よって，想定環境以外ではネットワーク情報を正確に取得できないなどの問題が発生し，これらのルーチング方式は正常に動作しない可能性が高い．したがって，これらのルーチング方式を用いる際には，ネットワーク情報を正確に取得できるか否かを熟慮し，効率的なメッセージ配送を行える状況であることを事前に確認する必要がある．

また，ネットワーク情報を用いるルーチング方式は，初期の頃に考えられたネットワーク情報を用いないルーチング方式をベースとして，ネットワーク情報に基づいて改良を加えたものが大半である．したがって，ネットワーク情報が不正確な場合や全く役に立たない場合には，これらのルーチング方式はネットワーク情報を用いないルーチング方式と同じ挙動になることがある．また，これらのルーチング方式の性能は，主にシミュレーション実験によって定量的な評価のみが行われており，定性的な議論は少ない．そこで，近年，解析的な評価をする試みが始められている [61].

1.2.3 ネットワーク情報を用いないルーチング方式

ネットワーク情報を用いずに，メッセージ配送を行う方式の代表例に直接配送（Direct Delivery）方式，感染型ルーチング（Epidemic Routing）方式 [80]，2 ホップ転送 (Two-Hop Forwarding) 方式 [26, 27, 36, 87, 89]，確率転送 (Probabilistic Forwarding) 方式 [2, 50]，Spray and Wait 方式 [71, 74] がある．

直接配送方式（図 1.6(a)） メッセージを発生した送信元端末が受信先端末と直接遭遇したときにのみ，メッセージの転送を行う．最もネットワーク資源消費が少ないが，直接遭遇しない限りメッセージ配送できないため，不到達率が大きくなる．また，メッセージが配送されたとしても非常に配送遅延が大きくなる．

感染型ルーチング方式（図 1.6(b)） メッセージを持つ端末が別の端末に遭遇する度に，必ず複製の転送を行う．したがって，メッセージの配送遅延が最も小さくなるが，ネットワーク資源が過剰に消費される．

2 ホップ転送方式（図 1.6(c)） 送信元端末のみがメッセージの複製の権利を保有しており，送信元端末からメッセージの複製を受信した端末はメッセージの複製の権利を持たない．したがって，この方式では高々 2 ホップで受信先端末にメッ

セージが配送される．また，2 ホップ転送方式の拡張方式として m ホップ転送方式（$m = 3, 4, \ldots$）が考えられている．この拡張方式では，高々 m ホップで受信先端末にメッセージ配送されるが，$m \geq 3$ の場合，ネットワーク資源を過剰に消費し，効率的なメッセージ配送を行えないことが知られている [87].

確率転送方式（図 1.6(d)） メッセージを持つ端末が別の端末に遭遇すると，確率 p $(0 \leq p \leq 1)$ でメッセージの複製を転送する．$p = 0$ の場合，直接配送方式と一致し，$p = 1$ の場合，感染型ルーチング方式と一致する．また，確率転送方式の拡張方式として (p, q) 感染型ルーチング方式が考えられている [50]．この方式では，送信元端末がメッセージの複製を転送する確率を q $(0 \leq q \leq 1)$ とし，中継端末が転送する確率を p とする．よって，$p = q$ の場合，確率転送方式と一致し，$q = 1, p = 0$ の場合，2 ホップ転送方式と一致する．

Spray and Wait 方式（図 1.6(e)） 生成するメッセージの複製の最大個数を P 個に制限する方式である．メッセージの生成時にカウンタを P に設定する．カウンタの値が n であるメッセージを持つ端末が別の端末に遭遇すると，カウンタの値を $\lfloor n/2 \rfloor$ としたメッセージの複製を転送し，自身のカウンタを $\lceil n/2 \rceil$ に更新する．カウンタの値が $n = 1$ であるメッセージを持つ場合，複製の生成権利を有しないため，受信先端末と遭遇した際にのみメッセージの転送を行う．Spray and Wait 方式ではパラメタ P を適切に設定することで，効率の良いメッセージ配送を行えることが示されている．

直接配送方式以外のルーチング方式ではメッセージの複製が生成されるため，受信先端末にメッセージが配送された後にもネットワーク内に複製が残存する．これらの不要なメッセージの複製はバッファを無駄に消費する．さらに，通信の完了を把握できないため，これらのメッセージを持つ端末が複製をまき散らし続けることになる．この問題を解決するために，ネットワーク内から不要なメッセージを除去する回復手法（recovery scheme）を併用しなければならない．

文献 [28, 87] では様々な回復手法が考察されている．一般に，不要となったメッセージの除去には，タイマーを利用した方式と明示的通知を行う方式がある．タイマーを利用した方式では全てのメッセージを有限時間内に除去可能であるが，タイマーを適切に設定しなければならない．タイマーを小さく設定し過ぎると，受信先端末へ配送される前にメッセージが削除されるため不到達率が増加する．一方で，タイマーを大きく設定し過ぎると，ネットワーク内に不要なメッセージが残存し続けるため，無駄なネットワーク資源の消費が増加する．

また，明示的通知を行う方式では，受信先端末がメッセージを受信すると，ネットワーク内に残存するメッセージを消去するためのアンチパケット（anti packet）をブロードキャストする．アンチパケットを受信した端末は，対応するメッセージを保持していればそのメッセージの削除を行う．アンチパケットは受信確認信号の役割も果たすため，送信元端末はメッセージが正しく転送されたことを知ることができる．なお，タイマーを利用した方式と明示的通知を行う方式は併用することが可能である．

これまで，これらのルーチング方式の性能は解析的に分析されてきた．文献 [25,

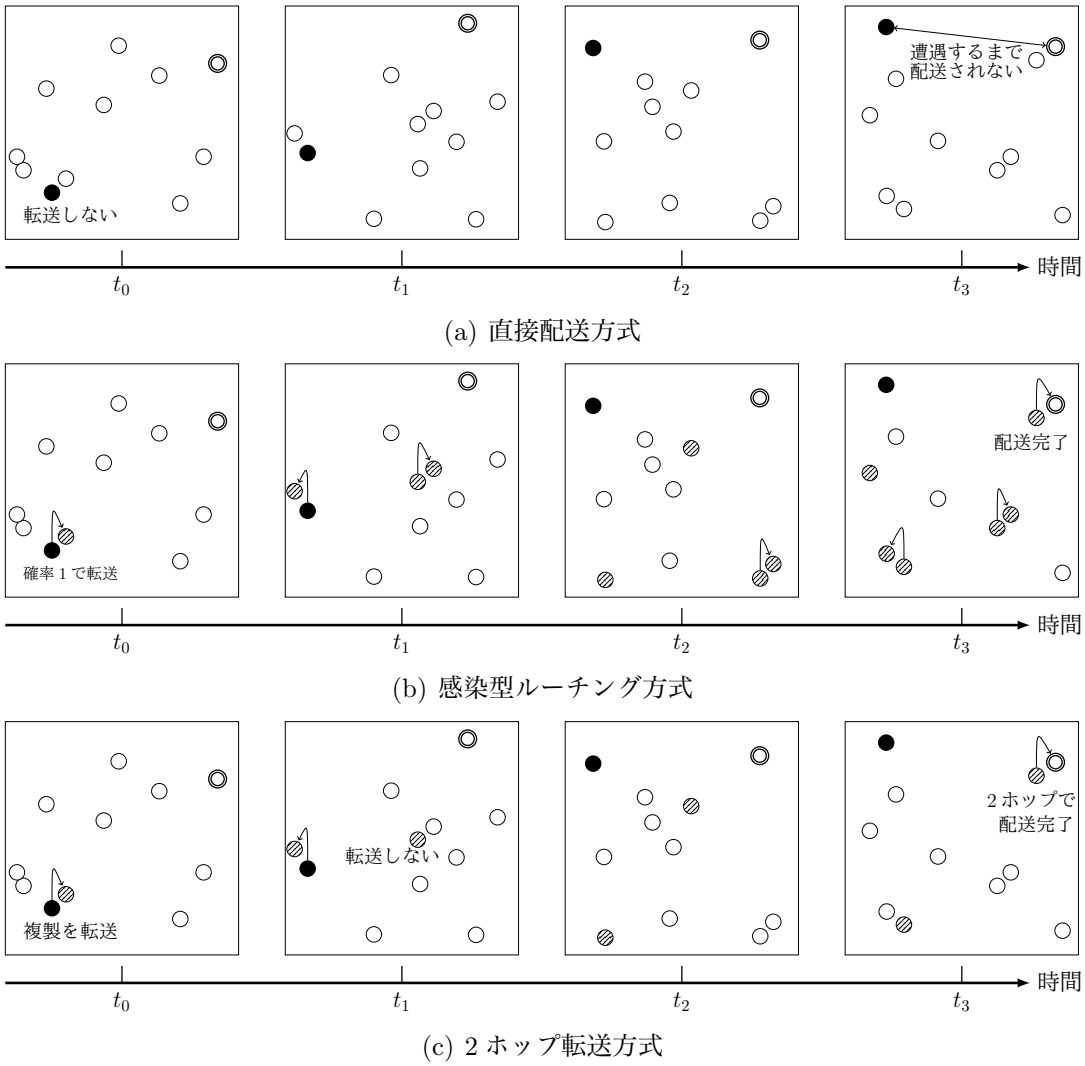
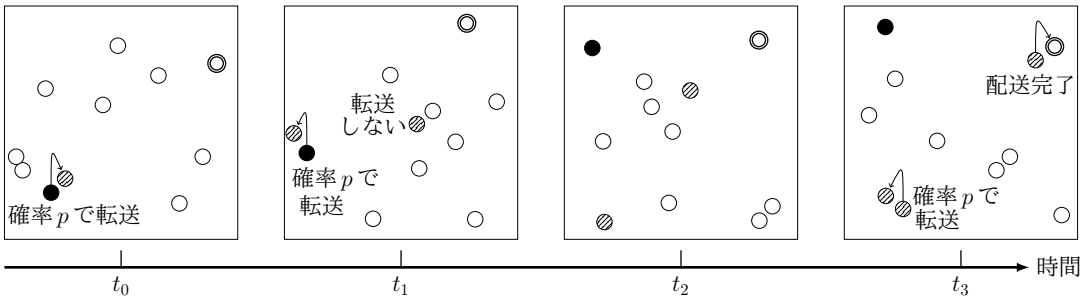


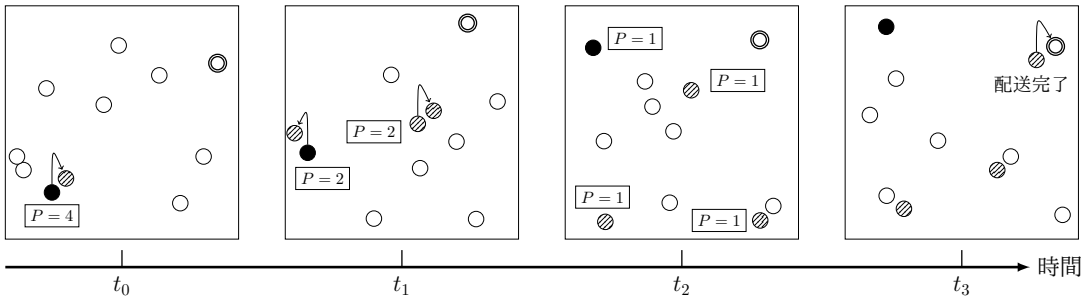
図 1.6: ネットワーク情報を用いないルーチング方式（黒丸は送信元端末，二重丸は受信先端末，斜線の丸は中継端末，白丸はメッセージを持たない端末を表す）

26] では，システムの状態を有限状態吸収マルコフ連鎖として定式化し，感染型ルーチング方式と 2 ホップ転送方式の平均配送遅延を得ている．また，文献 [28, 50] では，回復手法を考慮したマルコフモデルが議論されている．回復手法によるメッセージの削除を考慮することで平均配送遅延に加えてネットワーク資源消費に関連する複数の指標を得ている．さらに，平均配送遅延とネットワーク資源消費量の間のトレードオフの関係について議論が行われている．感染型ルーチング方式よりも 2 ホップ転送方式はネットワーク資源消費量に関して効率の良いメッセージ配送を実現できること，ならびに，2 ホップ転送方式よりも少しだけ積極的にメッセージの複製を転送するとある基準の下で効率の良いメッセージ配送を行えることが示されている．

さらに，文献 [87] では，さまざまな回復手法を考慮したマルコフモデルを流体近似 (fluid approximation) することが考えられている．流体近似によって性能評



(d) 確率転送方式



(e) Spray and Wait 方式（$P = 4$ の場合）

図 1.6: ネットワーク情報を用いないルーチング方式（黒丸は送信元端末，二重丸は受信先端末，斜線の丸は中継端末，白丸はメッセージを持たない端末を表す）

価指標を閉形式として表せるが，文献 [87] で示されるようにマルコフモデルを用いた理論解析の結果と一致しない．文献 [2] では，流体近似を用いて確率転送方式のルーチング性能を評価し，エネルギー消費量に関して最適な転送確率を導出している．さらに，文献 [69] では，流体近似モデルを用いて，送信元端末が遭遇した端末にメッセージの複製を転送するか否かを動的に決定する 2 ホップ転送（Dynamic Two-Hop）方式のシステムの状態をマルコフ決定過程として定式化し，最適な動的転送規則について議論が行われている．

これらの解析的な分析によって得られている結果を，最適なパラメタを用いたときに達成可能なメッセージ配送の効率性と設定パラメタに対する感度の関係によってまとめる（図 1.7 を参照）．Spray and Wait 方式では，最大複製個数のパラメタ P を適切に設定することで良い効率性を達成できるが，パラメタの設定が適切でない場合には効率性が大きく悪化する．一方で，感染型ルーチング方式，2 ホップ転送方式及び直接配送方式には設定パラメタがないが，Spray and Wait 方式よりもメッセージ配送の効率性が低い．また，確率転送方式や (p, q) 感染型ルーチング方式は設定パラメタに対する感度が高く，メッセージの転送確率の設定によってメッセージ配送の効率性が大きく変化する [50]．これらの結果から，ネットワーク情報を用いないルーチング方式における研究では，設定パラメタに対する感度が低く，かつ，メッセージ配送の効用性の高いルーチング方式についてほとんど議論されていないことが分かる．

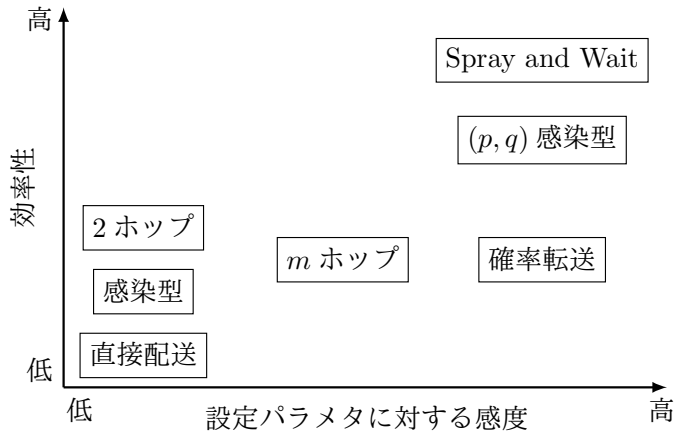


図 1.7: 設定パラメタに対する感度と達成可能なメッセージ配送の効率性の関係

1.2.4 ネットワーク情報を用いるルーチング方式

これまで，遭遇履歴情報から推定できる様々なネットワーク情報を用いたルーチング方式が考えられてきた．表 1.2 に既存のネットワーク情報を用いるルーチング方式の一覧を示す．用いられるネットワーク情報の大半は，遭遇端末の特徴に着目した情報と遭遇場所の特徴に着目した情報のいずれかに分類できる．前者の情報を用いたルーチング方式が「どの端末」と遭遇するかによってメッセージの転送を制御するのに対して，後者の情報を用いたルーチング方式は「どの場所」で遭遇するかによってメッセージの転送を制御する．

また，ネットワーク情報を用いる方式は，ネットワーク中に一つの複製しか存在させない単一複製（single copy）方式，もしくは，複数の複製を生成させる複数複製（multi copy）方式の二通りに分類できる．本論文では，これらの分類を組み合わせることによって，ネットワーク情報を用いる方式を 4 パターンに分ける．表記の簡単化のため，表 1.1 に示すように各パターンのルーチング方式を呼ぶことにする．また，図 1.8 に本節で詳しく紹介するルーチング方式をまとめる．

まず，一つ目として端末型単一複製方式について述べる．この方式では，メッセージを持つ端末が別の端末に遭遇すると，前者はメッセージを転送するか否かを後者の特徴量の良し悪しによって決定する．メッセージの転送を行った際には，前者の端末はメッセージの破棄を行う．この手順によってネットワーク内には，一つの複製のみが存在することになる．用いられる特徴量としては，最後の遭遇からの経過時間 [72, 73]，端末間の遭遇率 [13]，移動方向 [44]，ソーシャル関係情報 [14, 32, 33, 52] などが挙げられる．一般に，ネットワーク内の全ての端末がこれらの遭遇端末の特徴量を推定し，特徴量を正確な値に維持するには定期的な更新を行わなければならない．特徴量の管理が最も容易なルーチング方式の一つに，バッファのキュー長 (queue length) に基づくバックプレッシャー（backpressure）ルーチング方式がある [15]．この方式では，特徴量として二つの端末が遭遇した際のキュー長の差が用いられるため，ほとんど特徴量の管理を行う必要がないが，エネルギー消費量に関して効率的なメッセージ配送を行えることが示されている．

表 1.1: ネットワーク情報を用いた方式の分類

特徴量＼複製数	単一	複数
遭遇端末	端末型単一複製	端末型複数複製
遭遇場所	場所型単一複製	場所型複数複製

二つ目として端末型複数複製方式について述べる．この方式では，ネットワーク情報を用いる既存のルーチング方式と比較して，少ないネットワーク資源消費量で低配送遅延を達成できるのだが，遭遇端末の特徴量の管理と回復手法の両方を併用しなければならないため，複雑な処理を行わなければならない．これまでに非常に多くの方式が考えられてきた [1, 20, 46, 85]．ここでは，効用型ルーチング（Utility-based Routing）と Spray and Wait 方式をベースとしたルーチング方式 [55, 72] を紹介する．

効用型ルーチングでは，遭遇端末の特徴量として受信先端末との" 近さ" を表す効用値 (utility value) が用いられる．ここで言う" 近さ" とは，受信先端末と遭遇する潜在的な可能性を意味する．ある端末 X が受信先端末と頻繁に遭遇する場合，端末 X の効用値は大きい．さらに，ある端末 Y が端末 X と頻繁に遭遇する場合，端末 Y は端末 X を介して受信先端末にメッセージを配送できるため，端末 Y の効用値も大きくなる．効用型ルーチングでは，端末 C 宛のメッセージを持つ端末 A が端末 B に遭遇した際に，端末 B の端末 C に関する効用値に基づいてメッセージの複製を転送するか否かを決定する．PRoPHET ルーチング方式 [46] では，端末 B の効用値が端末 A よりも大きければ，複製の転送を行う．Delegation 転送方式 [17] では，メッセージを持つ各端末が閾値を保持しており，その閾値よりも大きな端末と遭遇した際にのみ複製の転送を行う．端末 B の効用値が端末 A の閾値よりも大きいとすると，複製の転送後，二つの端末は互いの閾値を端末 B の効用値の値に設定する．この手順によって，Delegation 転送方式は効用値の高い端末にのみに複製の転送を行うことを狙っている．

Spray and Wait 方式をベースとしたさまざまなルーチング方式が考えられている [55, 72, 78]．代表例として Spray and Focus 方式 [72] と EBR (Encounter-Based Routing) 方式 [55] を紹介する．Spray and Focus 方式では，メッセージの最大個数 P を Spray and Wait 方式と同様に事前に設定する．複製の権利を有しないメッセージ保持端末は受信先端末と遭遇するのをひたすら待つのではなく，受信先端末と遭遇しやすい端末に遭遇した場合，メッセージの転送を行い，自身の持っているメッセージの複製を消去する．EBR 方式においてもメッセージの最大個数 P を事前に設定する．そして，メッセージの生成時，送信元端末が P 個の複製を生成する．n 個の複製を持つ端末 A が別の端末 B に遭遇した状況を考え，Λ_A と Λ_B はそれぞれ端末 A と端末 B の遭遇率を表しているとする．このとき，端末 A は端末 B に $\lfloor n\Lambda_B/(\Lambda_A+\Lambda_B)\rfloor$ 個の複製を転送し，$\lceil n\Lambda_A/(\Lambda_A+\Lambda_B)\rceil$ 個の複製を保持し続ける．この手順は，多くの端末と遭遇しやすければしやすいほど，受信先端

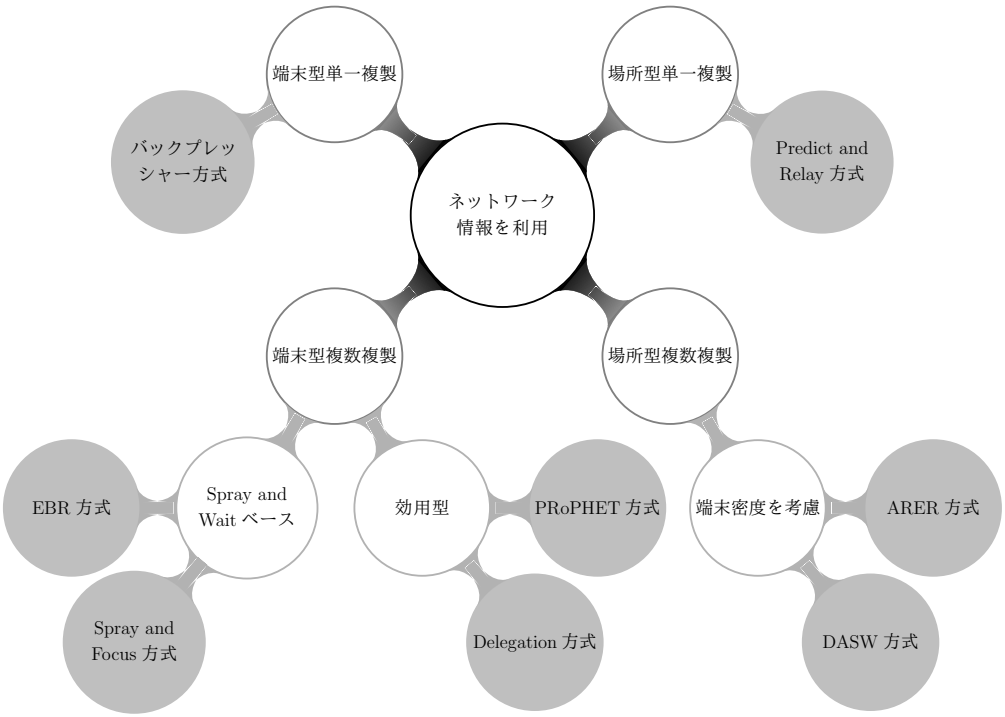


図 1.8: ネットワーク情報を用いるルーチング方式

末とも遭遇しやすいという特徴に着目して設計されている．

三つ目として場所型単一複製方式について述べる．Predict and Relay 方式 [82] では，端末が全ての場所をランダムに移動するのではなく，ある特定の領域のみを移動することを前提として設計が行われている．メッセージの転送は，現在いる領域から別の領域に移動する遷移確率とそれぞれの領域に滞在する平均の時間に基づいて決定される．この方式では，事前にネットワーク全体を複数の領域に区切る必要が有り，領域の区切り方がルーチング性能に大きな影響を与える．

最後に，場所型複数複製方式について述べる．遭遇場所毎の端末密度に着目した複数複製方式が考えられている [30, 79, 83]．文献 [79] では，一定期間内の遭遇数に基づいて端末密度を推定することが考えられている．一定期間内の遭遇回数が小さな場合，端末密度が小さいものと判断され，メッセージの複製が積極的に転送される．また，DASW (Density Aware Spray and Wait) [30] 方式では，端末間の遭遇時間間隔に基づいて端末密度を推定することが考えられている．遭遇時間間隔は，各端末が別の端末に遭遇する度に記録される．ある端末が記録している遭遇時間間隔が短い場合，現在いる地点の端末密度が高いものと推定する．メッセージを持つ端末が密度の高いところから低いところへ向かって移動している場合，密度の高いところに留まる可能性の高い端末にメッセージが転送される．ARER (Adaptive Randomized Epidemic Routing) 方式 [83] では，メッセージの複製を持つ端末の密度の情報を利用する．同じメッセージの複製を持つ端末と頻繁に遭遇する際には，そのメッセージの転送を抑制する．これらの既存の研究では，各端末が自身の位置を把握できないという前提の上で設計されている．さらに，各端末が直接把握

できる情報のみを用いて端末密度を推定している.

表 1.2: ネットワーク情報を用いるルーチング方式

ルーチング方式	用いられるネットワーク情報	分類	複製数
BP [15]	バッファのキュー長	端末型	単一
BubbleRap [32, 33]	ソーシャル関係情報	端末型	単一
CREST [76]	遭遇までの残余時間	端末型	単一
SimBet [14]	ソーシャル関係情報	端末型	単一
SimBetAge [47]	ソーシャル関係，時系列情報	端末型	単一
ORION [51]	遭遇する端末の時系列情報	端末型	単一
PeopleRank [52]	ソーシャル関係情報	端末型	単一
LCD [38]	遭遇時の接続時間	端末型	単一
MH* [13]	端末間の遭遇率	端末型	単一
MOVE [44]	移動方向	端末型	単一
BWAR [1]	バッファのキュー長	端末型	複数
DEEOF [49]	端末間の距離	端末型	複数
DF [17]	効用値	端末型	複数
DF++ [3]	効用値，バッファ容量	端末型	複数
DiPRoPHET [70]	効用値，端末間距離	端末型	複数
EBR [55]	遭遇率	端末型	複数
EFLR [64]	遭遇時の接続時間	端末型	複数
E-PRoPHET [45]	効用値，遭遇時の接続時間	端末型	複数
NECTAR [56]	遭遇時の接続時間，ホップ数	端末型	複数
ORWAR [65]	端末の移動速度，通信半径	端末型	複数
PF [16]	端末間の遭遇率	端末型	複数
PREP [62]	遭遇時の接続時間	端末型	複数
PROCS [34]	遭遇する端末の時系列情報	端末型	複数
PRoPHET [46]	効用値	端末型	複数
PRoPHET+ [31]	効用値，ネットワーク資源量	端末型	複数
RAPID [4]	遭遇時間間隔	端末型	複数
SaF [72]	最後の遭遇からの経過時間	端末型	複数
SMART [78]	遭遇回数，遭遇時間間隔	端末型	複数
文献 [7]	到達率の推定値	端末型	複数
文献 [85]	移動方向	端末型	複数
文献 [88]	配送遅延の見積り値	端末型	複数
PR [82]	遭遇場所	場所型	単一
ARER [83]	端末間の遭遇回数	場所型	複数
DASW [30]	遭遇時間間隔	場所型	複数
文献 [79]	一定期間内の遭遇回数	場所型	複数

1.2.5 研究目的

前節で述べたように，ネットワーク情報を用いるルーチング方式は，遭遇する端末や場所の偏りが存在することを前提として設計されているため想定通りのネットワーク環境であれば，ネットワーク情報を用いないルーチング方式よりも効率の良いメッセージ配送を行えるが，ネットワーク情報を推定するため計算負荷が大きい．また，想定通りのネットワーク環境でなければ，効率的なメッセージ配送を実現できる可能性が低いだけでなく，不要な計算負荷を被ることになる．例えば，端末のモビリティが完全にランダムな環境や頻繁に変化するような環境では，ネットワーク情報を用いるルーチング方式によって効率的なメッセージ配送を実現することは難しい．したがって，メッセージ配送の効率性を多少犠牲にしても良い場合やネットワーク情報がほとんど役に立たない場合には，ネットワーク情報を用いないルーチング方式の利用が有効である．一方で，メッセージ配送の効率性が重要である場合やネットワーク情報が明らかに役立つ場合には，ネットワーク情報を用いるルーチング方式が有用である．そこで，本論文では，ネットワーク情報を用いない方式と用いる方式それぞれについて下記の二つの新たなルーチング方式を検討し，これらのルーチング方式を用いたメッセージ配送が効率的であることを示す．

- 複製の生成権利を有する端末の数を制限するルーチング方式
- 端末密度に着目したルーチング方式

前者のルーチング方式では，複製の生成権利を有する端末の数を制限する際にネットワーク情報を用いない．図 1.7 に示したようにネットワーク情報を用いないルーチングに関する既存の研究では，設定パラメタに対する感度が小さく，かつ，メッセージ配送の効率性が高いルーチング方式についてこれまでほとんど検討されていない．本論文では，複製の生成権利を有する端末の数を制限するルーチング方式が上記の定性的な特徴を有することを数学モデルを用いた解析とリアルトレースデータを用いたシミュレーション実験によって示す．

上記の定性的な特徴を持つルーチング方式は，ネットワーク情報が取得できない場合や取得したとしてもほとんど役に立たない場合に有用である．このような状況の具体例としては，大規模災害発生直後が挙げられる．大規模災害発生直後は，直前の状況と大きく変わるため，以前に得たネットワーク情報を持っていたとしてもほとんど役に立たない．また，平常時と異なり端末は特定の場所のみを移動するのではなく，避難や救助活動を行うために移動が不規則になるため，ネットワーク情報の取得が困難である．

また，後者のルーチング方式は，遭遇場所の特徴量の一つである端末密度に応じてメッセージの複製の転送を制御するため，ネットワーク情報を用いるルーチング方式に分類される．1.2.3 節で示したように遭遇端末に着目したルーチング方式と比較して遭遇場所に着目したルーチング方式は非常に少なく，十分な議論が行われていない．しかしながら，場所によってネットワーク環境が大きく異なる

状況では，それぞれの場所に適したメッセージの複製の転送を制御することが有効である．例えば，場所によって端末間の遭遇頻度に偏りがある場合，遭遇頻度が小さな領域においてはメッセージの複製を生成する機会が少ないため積極的に複製を転送した方が良い．一方で，端末間の遭遇頻度が大きな領域においてはメッセージの複製を生成する機会が多いため，複製の転送を抑制したとしても配送遅延の増加にほとんど影響を与えない．このように場所毎に偏りがある場合には，遭遇場所の特徴量を用いたルーチング方式が有用である．

端末密度を推定するために，本論文では，各端末が GPS (Global Positioning System) などの位置測定技術を用いて自身の位置を把握できる状況を想定する．各端末が別の端末と遭遇した際にその遭遇地点の位置を記録し，さらに，記録した遭遇位置情報を遭遇時に端末間で交換することによってネットワーク全体の遭遇位置情報を収集する．このようにして収集した遭遇位置情報に基づいて端末密度を推定する複数の方法について検討し，それらの推定結果の利用がメッセージ配送に有効であることをシミュレーション実験によって示す．

さらに，遭遇端末の特徴量である効用値と遭遇場所の特徴量である端末密度の両方を用いてメッセージの複製の転送を制御するルーチング方式を考える．表 1.2 に示したようにこれまで遭遇端末と遭遇場所の特徴量を併用することについてほとんど議論が行われていない．そこで，本論文では,「どの場所」で「どの端末」と遭遇するかを同時に考慮することによって，どのような相乗効果が得られるかを明らかにする．

1.3 論文の構成

本論文の次章以降は以下のように構成される．

第２章では，複製の生成権利を有する端末の最大数を制限するマルチスプレッダルーチング方式を提案する．まず，マルチスプレッダルーチング方式のシステムのダイナミクスを有限状態吸収マルコフ連鎖として定式化し，このマルコフ連鎖の吸収状態への初到達時間として得られる平均配送遅延やネットワーク資源消費量を数値計算によって求める．次に，平均配送遅延とネットワーク資源消費量の間にあるトレードオフの評価を行う．さらに，リアルトレースデータを用いたシミュレーション実験によってネットワーク情報を用いない既存のルーチング方式との性能比較を行う．本章の内容は，研究業績の雑誌論文 ⟨1⟩，研究会 ⟨1⟩ に関連する．

第３章では，端末密度推定に基づく確率的蓄積運搬転送方式を提案する．この方式では，各端末が端末密度推定を行い，推定された端末密度応じてメッセージの複製の転送確率を制御する．シミュレーション実験によって，端末密度推定に基づく確率的蓄積運搬転送方式の有効性を示す．さらに，端末密度推定の誤差がルーチング性能に与える影響について議論する．本章の内容は，研究業績の雑誌論文 ⟨2⟩，国際会議 ⟨1⟩，研究会 ⟨5⟩, ⟨6⟩, ⟨7⟩ に関連する．

第 4 章では，端末密度を考慮した効用型ルーチング方式について述べる．この方式では，端末密度と効用値に応じてメッセージの複製の転送を制御する．遭遇場所と遭遇地点の両方の情報を考慮することで，単独の情報を用いるよりもエネルギー消費量に関して効率的なメッセージ配送を行えることをシミュレーション実験によって示す．本章の内容は，研究業績の国際会議 ⟨2⟩，研究会 ⟨8⟩, ⟨9⟩ に関連する．

最後に，第 5 章においては本論文の結論を述べる．

第2章

マルチスプレッダルーチング方式の提案とその評価

2.1 緒言

本章では，メッセージの複製を転送する権利を有する散布端末の最大数を予め定めた P $(P \geq 1)$ 個に制限するマルチスプレッダルーチング方式を提案する．散布端末の最大数を制限するためにネットワーク情報を用いないため，マルチスプレッダルーチング方式はネットワーク情報を用いないルーチング方式に分類される．また，マルチスプレッダルーチング方式では，特別な場合として感染型ルーチング方式（$P = \infty$）と２ホップ転送方式（$P = 1$）を包含する．文献 [50] において感染型ルーチング方式は達成可能な遅延性能に対してネットワーク資源を過剰に消費すること，２ホップ転送方式は過度に抑制していること，ならびに，２ホップ転送方式よりも少し積極的にメッセージの複製を転送すれば最適なメッセージ配送を行えることが示されている．したがって，散布端末の最大数 P を調整することによって配送遅延とネットワーク資源消費量の間にあるトレードオフのバランスのとれた効率的なメッセージ配送の実現が期待できる．

本章では，まず，全ての端末が同質であり，独立でランダムに移動するという前提のもとで，マルチスプレッダルーチング方式の性能について理論解析を行う．平均配送遅延やネットワーク資源消費量などの複数の指標を連続時間吸収マルコフ連鎖の吸収状態への初到達時間と関連付ける．これらの吸収マルコフ連鎖の状態空間を半順序集合として特徴付けることによって，様々な指標の後進再帰公式が得られることを示す．これらの後進再帰公式を用いた数値計算によって得られる結果を用いて配送遅延とネットワーク資源消費量の間のトレードオフについて議論し，小さな P に設定したマルチスプレッダルーチング方式が効率的なメッセージ配送を実現できること，および，設定パラメタ P に対する感度が低いことを示す．さらに，リアルトレースデータを用いたシミュレーション実験を行い，端末のモビリティが同質でない場合においても理論解析の結論と同じ傾向がみられることを示す．

2.2 VACCINE回復手法を用いたマルチスプレッダルーチング方式

2.2.1 マルチスプレッダルーチング方式

メッセージのヘッダにカウンタ値 c の情報が格納されているとする．カウンタ値 $c \geq 1$ のメッセージを持つ端末は，メッセージの複製を転送する権利を有しているものとする．以降では，このような複製を転送する権利を有している端末を散布端末と呼ぶ．一方で，カウンタ値 $c = 0$ のメッセージを持つ端末は複製を転送する権利を持たないものとし，中継端末と呼ぶ．新たなメッセージの生成後，下記の手順 [71, 74] に従って送信元端末はその端末自身を含めて最大で P 個の散布端末の生成を試みる．

新規メッセージの発生直後，メッセージのカウンタ値 c は P $(P \geq 1)$ に設定される．カウンタ値 $c \geq 1$ のメッセージを持つ散布端末が，別の端末 X に遭遇した場合，散布端末は端末 X にカウンタ値を $\lfloor c/2 \rfloor$ としたメッセージの複製を転送する．このとき，散布端末は自身の持っているメッセージのカウンタ値を $\lceil c/2 \rceil$ に更新する．したがって，$c \geq 2$ の場合，$\lfloor c/2 \rfloor \geq 1$ となり，端末 X は散布端末になる．また，一旦，端末が散布端末 $(c \geq 1)$ になると $\lceil c/2 \rceil \geq 1$ であるため，別の端末にメッセージの複製を転送したとしても散布端末のままであり続ける．さらに，$c = \lceil c/2 \rceil + \lfloor c/2 \rfloor$ が成り立つため，この手順を繰り返すことによって最終的に P 個の散布端末を生成できる．一方で，$c = 1$ の場合，$\lfloor c/2 \rfloor = 0$ となるため，メッセージの複製を受信した端末 X は中継端末となる．散布端末，もしくは，中継端末のいずれか一つが受信先端末と遭遇すれば受信先端末にメッセージが転送され，メッセージ配送が完了する．

さらに，カウンタ値 $c \geq 2$ のメッセージを持つ散布端末が中継先端末と遭遇した場合，散布端末と中継端末のカウンタ値をそれぞれ $\lceil c/2 \rceil$ と $\lfloor c/2 \rfloor$ に更新する．この手順によって中継端末は散布端末となるため，P 個の散布端末の生成を速めることができる．

2.2.2 VACCINE 回復手法

回復手法として最も積極的にメッセージの複製を削除する VACCINE 回復手法 [28] を採用する．VACCINE 回復手法では，受信先端末が別の端末からメッセージを受信し，メッセージの配送が完了すると，受信先端末においてアンチパケットが生成される．このとき同時に，メッセージを配送した端末においてもアンチパケットが生成される．したがって，メッセージ配送の完了直後には，このメッセージに対応するアンチパケットがネットワーク内に二つ存在することになる．これらのアンチパケットはネットワーク内の全ての端末に感染型ルーチングを用いてブロードキャストされる．対応するメッセージを保持している端末がアンチパケットを受信した際には，そのメッセージの削除を行う．さらに，アンチ

パケットを保持している端末は対応するメッセージの複製を受信しない．この回復手法では，全てのメッセージの複製がネットワーク内から既に消されていたとしてもそれを知る手段がないため，全ての端末にアンチパケットが拡散されるまでアンチパケットの転送が行われ続ける．

2.2.3 端末におけるメッセージ処理手順

VACCINE 回復手法を用いたマルチスプレッダールーチング方式は以下の手順に従って各端末はメッセージの処理を行う [50]．メッセージを蓄積するために，各端末はバッファを保持しているのとする．また，各端末は保持しているメッセージとアンチパケットに関するリストを保持しており，それらのリストにはメッセージの識別子と生成時刻が記録されている．ある端末が別の端末に遭遇した場合，最初に二つのリストの交換を必ず行う．このリスト交換によって，遭遇した二つの端末がアンチパケットの情報を共有でき，新しく受信したアンチパケットに対応するメッセージを保持していれば削除を行うことができる．さらに，メッセージに関するリストから，マルチスプレッダルーチング方式によってどのメッセージの複製を転送しなくてはならないかを把握できる．このメッセージに関するリストの交換は，いずれの蓄積運搬転送型ルーチング方式においても行わなければならない処理である．したがって，VACCINE 回復手法による主要なオーバーヘッドはアンチパケットのリストのサイズとその管理によるものである．リストに含まれるアンチパケットの最大数を M_{anti} とする．リストに含まれるアンチパケット数が M_{anti} を上回る場合，各端末は生成時刻の古い順にアンチパケットを削除する．この手順によってリストに含まれるアンチパケットの情報は，最新の M_{anti} 個のアンチパケットに関するものとなる．本章の残りの部分では，全ての端末が上記の手順に従ってメッセージの転送を行うものとする．

2.3 マルコフ解析

本節では，メッセージの配送遅延とメッセージの生存期間内におけるシステムのダイナミックスについて考える．ただし，メッセージの生存期間とは，メッセージの生成から全てのメッセージの複製がネットワーク中から削除されるまでの間と定義する．まず，これらを連続時間吸収マルコフ連鎖における吸収状態への初到達時間と関連付ける．さらに，吸収マルコフ連鎖の状態空間を半順序集合として特徴付けする．この特徴付けは，初到達時間を効率的に求めるために必要な後進再帰公式を得る際に役立つ．

2.3.1 数学モデル

ある閉領域内に送受信端末を含めて $N+1$ 台の移動端末が存在するものとする．時刻 0 において，送信元端末がオリジナルのメッセージを生成する．このメッセー

ジは VACCINE 回復手法を用いたマルチスプレッダルーチング方式によって受信先端末へ配送される．ネットワーク内の全ての端末は独立，かつ，ランダムに移動するものと仮定し，特定の二つの端末の遭遇時間間隔は独立で同一のパラメタ λ の指数分布に従うものとする．文献 [25, 26] においてランダムウェイポイントモデルやランダムディレクションモデルなどのランダムモビリティモデルに従う際には指数遭遇時間間隔の仮定が成り立つことが示されている．さらに，率 λ は二つの端末の相対速度に比例し，端末が移動する領域の面積に反比例することが示されている．

2.3.2 メッセージの生存期間における吸収マルコフ連鎖

メッセージの生存期間におけるシステムのダイナミクスを連続時間吸収マルコフ連鎖として定式化する．システムの状態は散布端末の状態，オリジナルのメッセージとその複製の総数，アンチパケットの総数によって表すことができる．本節では，このマルコフ連鎖の実行可能な状態空間と状態間の遷移率について述べる．

散布端末の状態は，散布端末の数とカウンタ値によって区別することができる．散布端末の最大数 P に対する実行可能な全てのカウンタ値の集合を $\mathcal{C}_P$ とする．集合 $\mathcal{C}_P$ は図 2.1 に示すアルゴリズムによって求められる．また，具体例として $\mathcal{C}_7$ を求める手順を図 2.2 に示す．ただし，$\mathcal{J}_P$ はアルゴリズムを停止するために必要な補助的な集合を表し，アルゴリズムによって既に選択されている $\mathcal{C}_P$ の要素が追加される．$\mathcal{C}_P$ の要素の大きな方から順に番号を割り振り，i $(i = 1, 2, \ldots, n(P))$ 番目に大きな要素を c_i と表す．ただし，$n(P) = |\mathcal{C}_P|$ とする．したがって，以下の関係が成り立つ．

$$P = c_1 > c_2 > \cdots > c_{n(P)-1} > c_{n(P)} = 1$$

$P = 7$ の場合，$n(7) = 5$ であり，$(c_1, c_2, c_3, c_4, c_5) = (7, 4, 3, 2, 1)$ である（図 2.2 を参照）．

ここで，$1 \times n(P)$ のベクトル $\boldsymbol{X}(t)$ $(t \geq 0)$ を $\boldsymbol{X}(t) = (X_1(t), X_2(t), \ldots, X_{n(P)})$ と定義する．ただし，j $(j = 1, 2, \ldots, n(P))$ 番目の要素 $X_j(t)$ は時刻 t にカウンタ値 c_j のメッセージを持つ散布端末数を表す．定義から，$\boldsymbol{X}(0) = (1, 0, \ldots, 0)$ である．ここで，時刻 t_1 に送信元端末がメッセージの生成後初めて別の端末に遭遇した状況を考える．このとき，$\boldsymbol{X}(t_1)$ は次式によって与えられる．

$$\boldsymbol{X}(t_1) = \begin{cases} (0, 2, 0, \ldots, 0), & P \text{ が偶数} \\ (0, 1, 1, 0, \ldots, 0), & P \text{ が奇数} \end{cases}$$

また，一旦，$\boldsymbol{X}(t)$ が $\boldsymbol{X}(t) = (0, \ldots, 0, P)$ になると，その後アンチパケットを受信するまで状態遷移が起こらない．

次に，$K(t)$ を時刻 t におけるオリジナルのメッセージとその複製の総数と定義

```
入力: P
出力: C_P
C_P := {P}，J_P := ∅ とする．
WHILE C_P ≠ J_P DO
  C_P\J_P から任意の要素 c を選び，
    J_P := J_P ∪ {c} とする．
  IF c ≥ 2 THEN
    C_P := C_P ∪ {⌈c/2⌉} ∪ {⌊c/2⌋}
  ENDIF
ENDWHILE
```

図 2.1: $\mathcal{C}_P$ の生成手順

初期:	$\mathcal{C}_7 = \{\underline{7}\}$	$\mathcal{J}_7 = \emptyset$
1 回目のループ後:	$\mathcal{C}_7 = \{7, \underline{4}, 3\}$	$\mathcal{J}_7 = \{7\}$
2 回目のループ後:	$\mathcal{C}_7 = \{7, 4, \underline{3}, 2\}$	$\mathcal{J}_7 = \{7, 4\}$
3 回目のループ後:	$\mathcal{C}_7 = \{7, 4, 3, \underline{2}, 1\}$	$\mathcal{J}_7 = \{7, 4, 3\}$
4 回目のループ後:	$\mathcal{C}_7 = \{7, 4, 3, 2, \underline{1}\}$	$\mathcal{J}_7 = \{7, 4, 3, 2\}$
5 回目のループ後:	$\mathcal{C}_7 = \{7, 4, 3, 2, 1\}$	$\mathcal{J}_7 = \{7, 4, 3, 2, 1\}$

図 2.2: $\mathcal{C}_7$ の生成手順（$\mathcal{C}_7 \backslash \mathcal{J}_7$ 内から次に選択される要素を下線で示す）

する．定義から，以下の不等式が成り立つ．

$$\sum_{j=1}^{n(P)} X_j(t) \leq K(t) \leq N$$

また，$K(t) - \sum_{j=1}^{n(P)} X_j(t)$ は時刻 t における中継端末の総数を表す．さらに，$M(t)$ を時刻 t におけるアンチパケット数を表すものと定義する．定義から，以下の不等式が成り立つ．

$$0 \leq M(t) \leq N + 1 - K(t)$$

メッセージの生存期間内のシステムのダイナミックスを連続時間マルコフ連鎖 $\{\boldsymbol{Z}(t);\ t \geq 0\}$ として定式化する．ただし，

$$\boldsymbol{Z}(t) = (\boldsymbol{X}(t), K(t), M(t))$$

である．

メッセージの生存期間は，受信先端末にメッセージが配送された時刻の前後によって二つに分けられる．前半の部分では，アンチパケットが存在せず（$M(t) = 0$），$c_1 X_1(t) + c_2 X_2(t) + \cdots + c_{n(P)} X_{n(P)}(t) = P$ が成り立つ．一方で，後半の部分では，少なくとも二つの端末がアンチパケットを保持しており（$M(t) \geq 2$），$c_1 X_1(t) + c_2 X_2(t) + \cdots + c_{n(P)} X_{n(P)}(t) \leq P$ が成り立つ．以下では，システムのダイナミックスをメッセージ配送時刻の前後に分けて議論する．それぞれの部分においてマルコフ連鎖 $\{\boldsymbol{Z}(t);\ t \geq 0\}$ の実行可能な状態空間とそれらの状態間の遷

移率について述べる．また，表記の簡単化のため，カウンタ値 c のメッセージを持つ端末を" カウンタ値 c の端末" と呼ぶことにする．

散布端末の状態を表す $1 \times n(P)$ のベクトルを $\boldsymbol{x} = (x_1, x_2, \ldots, x_{n(P)})$ と定義する．また，システムの状態を $(\boldsymbol{x}, k, m)$ と表記する．ただし，k はオリジナルのメッセージとその複製の総数を表し，m はアンチパケット数を表すものとする．さらに，$N_S^{[\boldsymbol{x}]}$ を状態 $(\boldsymbol{x}, k, m)$ のときの散布端末数を表すものとする．

$$N_S^{[\boldsymbol{x}]} = \sum_{j=1}^{n(P)} x_j \tag{2.1}$$

$N_R^{[\boldsymbol{x},k]}$ は状態 $(\boldsymbol{x}, k, m)$ のときの中継端末数を表す．

$$N_R^{[\boldsymbol{x},k]} = k - N_S^{[\boldsymbol{x}]} \tag{2.2}$$

時刻 $t = 0$ においてオリジナルのメッセージが発生するため，$\boldsymbol{Z}(0) = (\boldsymbol{X}(0), K(0), M(0)) = (\boldsymbol{e}_1, 1, 0)$ である．ただし，$\boldsymbol{e}_j$ $(j = 1, 2, \ldots, n(P))$ は j 番目の要素が 1 の $1 \times n(P)$ 単位ベクトルである．受信先端末にメッセージが配送されるまでは $M(t) = 0$ であり，配送後は $M(t) \geq 2$ となるため，メッセージの生存期間は状態 $(\boldsymbol{0}, 0, m)$ $(m = 2, 3, \ldots, N+1)$ のいずれかの状態へ到達した時点で終了する．以降では，散布端末の状態とシステムの状態をアンチパケット数によって分類する．

$\mathcal{X}_P(m)$ と $\mathcal{S}_P(m)$ $(m = 0, 2, 3, \ldots, N+1)$ をそれぞれ，アンチパケットがネットワーク中に m 個存在する場合の散布端末とシステムの実行可能な状態の集合と定義する．アンチパケット数 $M(t)$ は非減少であるため，散布端末の状態 $\boldsymbol{X}(t)$ は以下の順序に従う遷移のみ可能である．

$$\mathcal{X}_P(0) \Rightarrow \mathcal{X}_P(2) \Rightarrow \mathcal{X}_P(3) \Rightarrow \cdots \Rightarrow \mathcal{X}_P(N+1)$$

同様に，システムの状態 $\boldsymbol{Z}(t)$ は以下の順序に従う遷移のみが可能である．

$$\mathcal{S}_P(0) \Rightarrow \mathcal{S}_P(2) \Rightarrow \mathcal{S}_P(3) \Rightarrow \cdots \Rightarrow \mathcal{S}_P(N+1)$$

$\mathcal{X}_P(m)$ と $\mathcal{S}_P(m)$ $(m = 0, 2, 3, \ldots, N+1)$ を得るために，$\mathcal{Y}(m)$ と $\mathcal{A}_P(m)$ を定義する．$\mathcal{Y}(m)$ と $\mathcal{A}_P(m)$ はそれぞれ，アンチパケット数が m になった直後の散布端末とシステムの実行可能な状態の集合を表す．$\mathcal{Y}_P(m)$ と $\mathcal{X}_P(m)$ には以下の関係が成り立つ．

$$\mathcal{Y}_P(m) \subset \mathcal{X}_P(m), \qquad m = 2, 3, \ldots, N+1$$

最初に，$\mathcal{X}_P(0), \mathcal{S}_P(0)$ 及び $\mathcal{S}_P(0)$ に含まれる状態からの遷移率について述べる．$M(t) = 0$ の場合，ネットワーク内の端末は四つのタイプに分類できる．散布端末，中継端末，不保持端末（メッセージを保持しない端末），受信先端末の四つである．さらに，散布端末はメッセージのカウンタ値によって二つのタイプに細分化する．一つはカウンタ値 $c \geq 2$ で，散布端末を生成できる端末である．もう一つはカウ

ンタ値 $c=1$ で中継端末のみを生成できる端末である．これらの分類を考慮することで表 2.1 を得ることができる．表 2.1 では，状態 $(\boldsymbol{x},k,0)\in\mathcal{S}_P(0)$ からの遷移先の状態とその遷移率を表している．付録 A.1 において状態 $(\boldsymbol{x},k,0)$ からの遷移率について詳しく述べる．また，表 2.1 において同じセル中の上段と下段は，それぞれ遭遇後の状態とその状態への遷移率を表している．ただし，$f_P(c_j)$ を

$$f_P(c_j)=j,\qquad c_j\in\mathcal{C}_P$$

とし，$j=1,2,\ldots,n(P)-1$ に対して $\boldsymbol{x}'_j$ は次式で与えられる．

$$\boldsymbol{x}'_j=\boldsymbol{x}-\boldsymbol{e}_j+\boldsymbol{e}_{f_P(\lceil c_j/2\rceil)}+\boldsymbol{e}_{f_P(\lfloor c_j/2\rfloor)} \tag{2.3}$$

$\mathcal{X}_P(0)$ を求める手順を図 2.3 に示す．さらに，中継端末を生成できるのは $x_{n(P)}\geq 1$ の場合のみであるため，$\mathcal{S}_P(0)$ は次式で与えられる．

$$\begin{aligned}\mathcal{S}_P(0) &= \{(\boldsymbol{x},N_S^{[\boldsymbol{x}]},0);\ \boldsymbol{x}\in\mathcal{X}_P(0),x_{n(P)}=0\}\\ &\quad\cup\{(\boldsymbol{x},k,0);\ \boldsymbol{x}\in\mathcal{X}_P(0),x_{n(P)}\geq 1,\\ &\qquad\qquad k=N_S^{[\boldsymbol{x}]},N_S^{[\boldsymbol{x}]}+1,\ldots,N\}\end{aligned}$$

次に，$\mathcal{X}_P(2)$ と $\mathcal{S}_P(2)$ を得るために，受信先端末にメッセージが配送された直後のシステムの状態について考える．図 2.1 から $\mathcal{Y}_P(2)$ と $\mathcal{A}_P(2)$ は以下のように得られる．

$$\begin{aligned}\mathcal{Y}_P(2) &= \{\boldsymbol{x}-\boldsymbol{e}_j;\ \boldsymbol{x}\in\mathcal{X}_P(0),x_j\geq 1\ (j=1,2,\ldots,n(P))\}\\ &\quad\cup\{\boldsymbol{x};\ (\boldsymbol{x},k,0)\in\mathcal{S}_P(0),N_R^{[\boldsymbol{x},k]}\geq 1\}\\ \mathcal{A}_P(2) &= \{(\boldsymbol{x}-\boldsymbol{e}_j,k-1,2);\ (\boldsymbol{x},k,0)\in\mathcal{S}_P(0),\\ &\quad x_j\geq 1\ (j=1,2,\ldots,n(P))\}\\ &\quad\cup\{(\boldsymbol{x},k-1,2);\ (\boldsymbol{x},k,0)\in\mathcal{S}_P(0),N_R^{[\boldsymbol{x},k]}\geq 1\}\end{aligned}$$

アンチパケットを持つ端末が存在する場合，ネットワーク内の端末は四つに分類できる．散布端末，中継端末，不保持端末（メッセージもアンチパケットも持たない端末），アンチパケット保持端末である．表 2.2 は状態 $(\boldsymbol{x},k,m)\in\mathcal{S}_P(m)$ $(m\geq 2)$ からの遷移先とその遷移率をまとめている．各セルの上段と下段はそれぞれ，遭遇後の状態とその状態への遷移率を表している．付録 A.2 において状態 $(\boldsymbol{x},k,m)$ からの遷移率については詳しく述べる．

表 2.2 から $m=2,3,\ldots,N+1$ に対して $\mathcal{X}_P(m)$ と $\mathcal{S}_P(m)$ はそれぞれ $\mathcal{Y}_P(m)$ と $\mathcal{A}_P(m)$ に含まれる全ての状態と，アンチパケット数が m の場合に $\mathcal{Y}_P(m)$ と $\mathcal{A}_P(m)$ 内の状態から到達可能な状態を含んだ集合であることが分かる．一方で，$\mathcal{Y}_P(m)$ と $\mathcal{A}_P(m)$ はそれぞれ，$m=3,4,\ldots,N+1$ に対して $\mathcal{X}_P(m-1)$ と $\mathcal{S}_P(m-1)$ から到達可能な m 個のアンチパケットが存在する全ての状態を含んだ集合である．既に述べたように $(\mathcal{Y}_P(2),\mathcal{A}_P(2))$ は得ることができるため，図 2.4 と図 2.5 の手順によって $(\mathcal{X}_P(m),\mathcal{S}_P(m))$ $(m=2,3,\ldots,N+1)$ と $(\mathcal{Y}_P(m),\mathcal{A}_P(m))$ $(m=3,4,\ldots,N+1)$ を交互に生成できる．

表 2.1: 状態 $(\boldsymbol{x}, k, 0) \in \mathcal{S}_P(0)$ からの状態遷移先とその遷移率

	中継端末	不保持端末	受信先端末
カウンタ値 c_j (≥ 2) の散布端末	$(\boldsymbol{x}'_j, k, 0)$ $N_R^{[\boldsymbol{x},k]} x_j \lambda$	$(\boldsymbol{x}'_j, k+1, 0)$ $(N-k) x_j \lambda$	$(\boldsymbol{x} - \boldsymbol{e}_j, k-1, 2)$ $x_j \lambda$
カウンタ値 $c_{n(P)}$ $(=1)$ の散布端末	—	$(\boldsymbol{x}, k+1, 0)$ $(N-k) x_{n(P)} \lambda$	$(\boldsymbol{x} - \boldsymbol{e}_{n(P)}, k-1, 2)$ $x_{n(P)} \lambda$
中継端末	—	—	$(\boldsymbol{x}, k-1, 2)$ $N_R^{[\boldsymbol{x},k]} \lambda$

表 2.2: $\mathcal{S}_P(m)$ $(m \geq 2)$ に含まれる状態 $(\boldsymbol{x}, k, m)$ からの遷移先とそれらの遷移率

	中継端末	不保持端末	アンチパケット保持端末
c_j (>2) の散布端末	$(\boldsymbol{x}'_j, k, m)$ $N_R^{[\boldsymbol{x},k]} x_j \lambda$	$(\boldsymbol{x}'_j, k+1, m)$ $(N+1-k-m) x_j \lambda$	$(\boldsymbol{x} - \boldsymbol{e}_j, k-1, m+1)$ $m x_j \lambda$
$c_{n(P)}$ $(=1)$ の散布端末	— —	$(\boldsymbol{x}, k+1, m)$ $(N+1-k-m) x_{n(P)} \lambda$	$(\boldsymbol{x} - \boldsymbol{e}_{n(P)}, k-1, m+1)$ $m x_{n(P)} \lambda$
中継端末	— —	— —	$(\boldsymbol{x}, k-1, m+1)$ $m N_R^{[\boldsymbol{x},k]} \lambda$
不保持端末	— —	— —	$(\boldsymbol{x}, k, m+1)$ $m(N+1-k-m) \lambda$

```
入力: P, C_P
出力: X_P(0)
X_P(0) = {e_1}, J_P = ∅ とする.
WHILE X_P(0) ≠ J_P DO
  X_P(0)\J_P から任意の要素 x を選び,
    J_P := J_P ∪ {x} とする.
  FOR j := 1 STEP +1 UNTIL n(P) − 1 DO
    IF x_j ≥ 1 THEN
      X_P(0) := X_P(0) ∪ {x'_j}
    ENDIF
  ENDFOR
ENDWHILE
```

図 2.3: $\mathcal{X}_P(0)$ の生成手順（$\boldsymbol{x}'_j$ は式 (2.3) で与えられる）

```
入力: P, 𝒴_P(m), 𝒜_P(m)
出力: 𝒳_P(m), 𝒮_P(m)
𝒳_P(m) := 𝒴_P(m), 𝒮_P(m) := 𝒜_P(m), 𝒥_P := ∅ とする.
WHILE 𝒮_P(m) ≠ 𝒥_P DO
  𝒮_P(m)\𝒥_P から任意の要素 (x, k, m) を選び,
    𝒥_P := 𝒥_P ∪ {(x, k, m)} とする.
  FOR j := 1 STEP +1 UNTIL n(P) − 1 DO
    IF x_j ≥ 1 THEN
      IF k > N_S^[x] THEN
        𝒳_P(m) := 𝒳_P(m) ∪ {x'_j};
        𝒮_P(m) := 𝒮_P(m) ∪ {(x'_j, k, m)}
      ENDIF
      IF k + m ≤ N THEN
        𝒳_P(m) := 𝒳_P(m) ∪ {x'_j};
        𝒮_P(m) := 𝒮_P(m) ∪ {(x'_j, k + 1, m)}
      ENDIF
    ENDIF
  ENDFOR
  IF x_n(P) ≥ 1 AND k + m ≤ N THEN
    𝒮_P(m) := 𝒮_P(m) ∪ {(x, k + 1, m)}
  ENDIF
ENDWHILE
```

図 2.4: $\mathcal{X}_P(m)$ と $\mathcal{S}_P(m)$ の生成手順（$\boldsymbol{x}'_j$ は式 (2.3) で与えられる）

2.3.3 システム状態の半順序集合

メッセージの生存期間内のアンチパケット数 m は非減少である．さらに，ある固定された m $(m = 0, 2, 3, \ldots, N)$ に対して散布端末数 $N_S^{[\boldsymbol{x}]}$ と，オリジナルのメッセージとその複製の総数 k は非減少である．したがって，各 $\mathcal{S}_P(m)$ $(m = 0, 2, 3, \ldots, N)$ に対して以下の二項関係 $\preceq$ を導入できる．

$$
\begin{aligned}
&(\boldsymbol{x}_1, k_1, m) \preceq (\boldsymbol{x}_2, k_2, m) \\
&\qquad \Leftrightarrow \ (N_S^{[\boldsymbol{x}_1]} < N_S^{[\boldsymbol{x}_2]}) \text{ または } (N_S^{[\boldsymbol{x}_1]} = N_S^{[\boldsymbol{x}_2]} \text{ かつ } k_1 < k_2)
\end{aligned}
$$

$(\boldsymbol{x}_1, k_1, m) \preceq (\boldsymbol{x}_2, k_2, m)$ の場合，状態 $(\boldsymbol{x}_2, k_2, m)$ はいかなるサンプルパス上においても状態 $(\boldsymbol{x}_1, k_1, m)$ より前に出現することはない．また，$(\boldsymbol{x}_1, k_1, m) \preceq (\boldsymbol{x}_2, k_2, m)$ も $(\boldsymbol{x}_1, k_1, m) \succeq (\boldsymbol{x}_2, k_2, m)$ も満たさない場合，どちらか一方の状態のみがサンプルパス上に出現することが可能である．したがって，ある固定された m $(m = 0, 2, 3, \ldots, N)$ に対して，$(\mathcal{S}_P(m), \preceq)$ は半順序集合と考えられ，集合 $\mathcal{S}_P(m)$ の要素を二項関係 $\preceq$ を用いて 1 から $|\mathcal{S}_P(m)|$ まで番号を付けられる．ただし，タイブレイクの場合には任意に番号を割り当てるものとする．

後で詳細に述べるが，興味のある全ての性能評価指標は有限状態吸収マルコフ連鎖 $\{\boldsymbol{Z}(t);\ t \geq 0\}$ の初到達時間と極めて関連が深い．初到達時間に関連する量は，システムの線形連立方程式を満たすことが知られている [43]．したがって，一

```
入力: P, S_P(m-1)
出力: Y_P(m), A_P(m)
Y_P(m) := ∅, A_P(m) := ∅, J_P := ∅ とする.
WHILE J_P ≠ S_P(m-1) DO
  S_P(m-1)\J_P から任意の要素 (x, k, m) を選び,
    J_P := J_P ∪ {(x, k, m)} とする.
  FOR j := 1 STEP +1 UNTIL n(P) DO
    IF x_j ≥ 1 THEN
      Y_P(m) := Y_P(m) ∪ {x - e_j}
      A_P(m) := A_P(m) ∪ {(x - e_j, k-1, m)}
    ENDIF
  ENDFOR
  IF N_S^[x] < k THEN
    Y_P(m) := Y_P(m) ∪ {x}
    A_P(m) := A_P(m) ∪ {(x, k-1, m)}
  ENDIF
  IF k + m ≤ N THEN
    Y_P(m) := Y_P(m) ∪ {x}
    A_P(m) := A_P(m) ∪ {(x, k, m)}
  ENDIF
ENDWHILE
```

図 2.5: $\mathcal{Y}_P(m)$ と $\mathcal{A}_P(m)$ の生成手順

般に，興味のある性能評価指標の値を得るためには，システムの線形連立方程式を解かなければならない．しかしながら，ここで考えているシステムでは，実行可能な状態の集合が半順序集合になっており，性能評価指標の値は単純な連続した置き換えを繰り返すことによって効率よく求められる．本節の残りの部分では，メッセージの配送遅延の求め方を説明することで，どのようにすれば数値計算を効率的に行うことができるかを明らかにする．

定義から，メッセージの配送遅延は，$\boldsymbol{Z}(0) = (\boldsymbol{e}_1, 1, 0)$ で条件付けされたマルコフ連鎖 $\{\boldsymbol{Z}(t);\ t \geq 0\}$ の集合 $\mathcal{A}_P(2)$ に含まれるいずれかの状態へ到達するまでの初到達時間 T_D と一致する．形式的には T_D は次式で定義される．

$$T_D = \inf\{t \geq 0;\ \boldsymbol{Z}(t) \in \mathcal{A}_P(2)\} \tag{2.4}$$

平均配送遅延 $\mathrm{E}[T_D] := \mathrm{E}[T_D \mid \boldsymbol{Z}(0) = (\boldsymbol{e}_1, 1, 0)]$ を再帰的な計算によって求めるために，$T_D(\boldsymbol{x}, k, 0)$ $((\boldsymbol{x}, k, 0) \in \mathcal{S}_P(0))$ を次式で定義する．

$$T_D(\boldsymbol{x}, k, 0) = \mathrm{E}[T_D \mid \boldsymbol{Z}(0) = (\boldsymbol{x}, k, 0)]$$

定義から，$\mathrm{E}[T_D] = T_D(\boldsymbol{e}_1, 1, 0)$ である．$t \in [0, T_D)$ の場合 $M(t) = 0$ であるため，

$(\boldsymbol{x}, k, 0) \in \mathcal{S}_P(0)$ に対して表 2.1 から $T_D(\boldsymbol{x}, k, 0)$ が次式を満たすことが分かる．

$$\begin{aligned} T_D(\boldsymbol{x}, k, 0) &= \frac{1}{\lambda_1(\boldsymbol{x}, k, 0)} + \frac{(N-k)x_{n(P)}\lambda}{\lambda_1(\boldsymbol{x}, k, 0)} \cdot T_D(\boldsymbol{x}, k+1, 0) \\ &\quad + \sum_{j=1}^{n(P)-1} \frac{(N-k)x_j\lambda}{\lambda_1(\boldsymbol{x}, k, 0)} \cdot T_D(\boldsymbol{x}'_j, k+1, 0) \\ &\quad + \sum_{j=1}^{n(P)-1} \frac{N_R^{[\boldsymbol{x},k]}x_j\lambda}{\lambda_1(\boldsymbol{x}, k, 0)} \cdot T_D(\boldsymbol{x}'_j, k, 0) \end{aligned} \tag{2.5}$$

ただし，$T_D(\boldsymbol{x}, 0, 0) = T_D(\boldsymbol{x}, N+1, 0) = 0$ であり，$\lambda_1(\boldsymbol{x}, k, 0)$ $((\boldsymbol{x}, k, 0) \in \mathcal{S}_P(0))$ は状態 $(\boldsymbol{x}, k, 0)$ からの全ての遷移率の和である．

$$\begin{aligned} \lambda_1(\boldsymbol{x}, k, 0) &= \sum_{j=1}^{n(P)-1} (N_R^{[\boldsymbol{x},k]} + N - k + 1)x_j\lambda \\ &\quad + (N-k+1)x_{n(P)}\lambda + N_R^{[\boldsymbol{x},k]}\lambda \end{aligned}$$

式 (2.5) の右辺の第 1 項目は状態 $(\boldsymbol{x}, k, 0)$ に滞在する平均時間であり，第 2 項目の最初の因子は状態 $(\boldsymbol{x}, k, 0)$ に滞在後に状態 $(\boldsymbol{x}, k+1, 0)$ へ遷移する確率を表している．他の項に関しても同様に解釈することができる．

$\mathrm{E}[T_D] = T_D(\boldsymbol{e}_1, 1, 0)$ の計算手順について述べる．式 (2.5) は $T_D(\boldsymbol{x}, k, 0)$ $((\boldsymbol{x}, k, 0) \in \mathcal{S}_P(0))$ に対する線形連立方程式を与えている．すなわち，式 (2.5) は以下の形式に書き換えられる．

$$\boldsymbol{t}_D = \boldsymbol{G}\boldsymbol{t}_D + \boldsymbol{b} \tag{2.6}$$

ただし，$\boldsymbol{t}_D$ と $\boldsymbol{b}$ はそれぞれ $T_D(\boldsymbol{x}, k, 0)$ と $1/\lambda_1(\boldsymbol{x}, k, 0)$ を要素とする列ベクトルであり，$\boldsymbol{G}$ は係数行列である．図 2.6 に示すように小さな P の場合を除いて，状態空間のサイズ $|\mathcal{S}_P(0)|$ が大きくなる．さらに，係数行列 $\boldsymbol{G}$ の各行は高々 $2n(P)-1$ 個の要素のみが非ゼロであるため，$\boldsymbol{G}$ はスパースである．したがって，LU 分解のような汎用的なソルバーによってシステムの線形連立方程式の解を求めるのは効率的でない．

この問題を解決するために，$(\mathcal{S}_P(0), \preceq)$ が半順序集合であることを利用する．$(\boldsymbol{x}_1, k_1, 0) \preceq (\boldsymbol{x}_2, k_2, 0)$ の場合，いかなるサンプルパス上においても状態 $(\boldsymbol{x}_1, k_1, 0)$ より前に $(\boldsymbol{x}_2, k_2, 0)$ が出現しないため，式 (2.5) では以下の関係が成り立つことが分かる．

$$\begin{aligned} (\boldsymbol{x}, k, 0) &\preceq (\boldsymbol{x}, k+1, 0), \\ (\boldsymbol{x}, k, 0) &\preceq (\boldsymbol{x}'_j, k+1, 0), \qquad j = 1, 2, \ldots, n(P)-1 \\ (\boldsymbol{x}, k, 0) &\preceq (\boldsymbol{x}'_j, k, 0), \qquad j = 1, 2, \ldots, n(P)-1 \end{aligned}$$

さらに，$(\boldsymbol{x}_1, k_1, m) \not\preceq (\boldsymbol{x}_2, k_2, m)$ かつ $(\boldsymbol{x}_1, k_1, m) \not\succeq (\boldsymbol{x}_2, k_2, m)$ を満たすいかなる二つの状態 $(\boldsymbol{x}_1, k_1, 0)$，$(\boldsymbol{x}_2, k_2, 0)$ は同じサンプルパス上に出現しない．したがって，

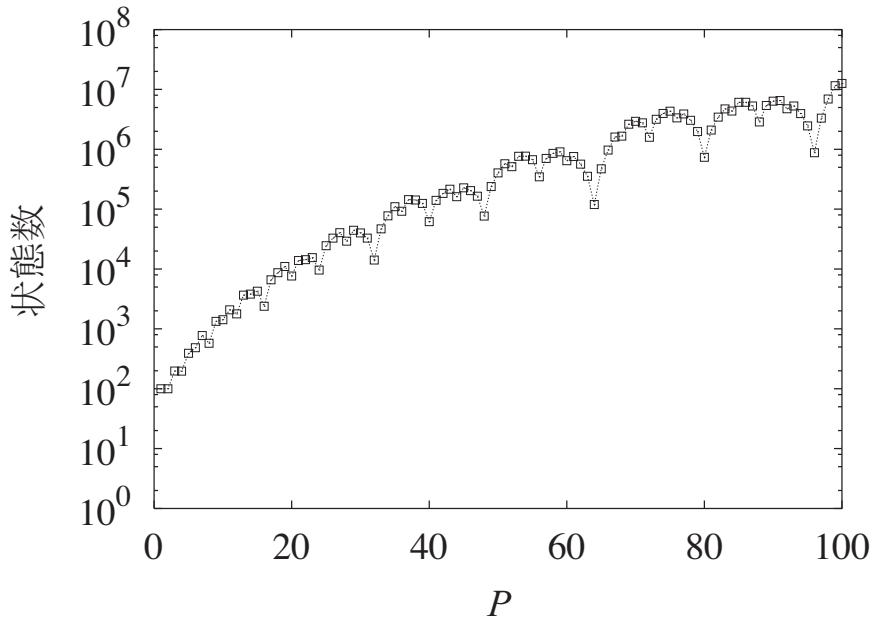


図 2.6: $\mathcal{S}_P(0)$ に含まれる実行可能な状態 $(\boldsymbol{x}, k, 0)$ の数

$(\mathcal{S}_P(0), \preceq)$ によって集合 $\mathcal{S}_P(0)$ に含まれる状態 $(\boldsymbol{x}, k, 0)$ を番号付けすることで，$T_D(\boldsymbol{x}, k, 0)$ を番号の大きな方から順々に得ることができる．

図 2.7 は平均配送遅延 $\mathrm{E}[T_D]$ を計算するための後進再帰手順を示している．ただし，表記の簡単化のため，$\boldsymbol{x}^{[l]} = (x_1^{[l]}, x_2^{[l]}, \ldots, x_{n(P)}^{[l]})$ は $\mathcal{X}_P(0)$ の l 番目の要素であるものとする．また，$k > N_S^{[\boldsymbol{x}]}$ である状態 $(\boldsymbol{x}, k, 0)$ は $x_{n(P)} \geq 1$ の場合，かつ，その場合に限り，実行可能である．

2.4 性能評価

本節では，まずマルチスプレッダルーチング方式の基本的な特性を明らかにし，マルコフ解析とシミュレーション実験によって得られる数値結果を用いてネットワーク資源消費量と平均配送遅延の間のトレードオフについて議論する．次に，マルチスプレッダルーチング方式とネットワーク情報を用いない既存のルーチング方式との性能比較を行う．最後にリアルトレースデータを用いたシミュレーション実験の結果を示す．

また，単位時間としてある端末が別の端末と遭遇する時間の平均である $1/(\lambda N)$ を用いる．この設定では端末密度が常に固定され，N の増加は端末が移動する閉領域の面積の増加を意味する [25, 26]．

2.4.1 平均配送遅延と平均メッセージ生存時間

最初に，平均配送遅延 $\mathrm{E}[T_D]$ について議論する．図 2.8 は散布端末の最大数 P に対する $\mathrm{E}[T_D]$ の変化を示している．$\mathrm{E}[T_D]$ は P に対して単調減少な凸関数であ

```
入力: P, 𝒳_P(0)
出力: E[T_D]
FOR l := |𝒳_P(0)| STEP −1 UNTIL 1 DO
  IF x^[l]_n(P) ≥ 1 THEN
    FOR k := N STEP −1 UNTIL N_S^[x^[l]]
      式 (2.5) を用いて T_D(x^[l], k, 0) を計算する.
    ENDFOR
  ELSE
    式 (2.5) を用いて T_D(x^[l], N_S^[x^[l]], 0) を計算する.
  ENDIF
ENDFOR
E[T_D] := T_D(e_1, 1, 0) とする.
```

図 2.7: $\mathrm{E}[T_D]$ を計算するための後進再帰手順

り，感染型ルーチング ($P = \infty$) の $\mathrm{E}[T_D]$ に急激に近づいていくことが分かる．したがって，P が小さい場合にのみ，P の増加は $\mathrm{E}[T_D]$ の減少に貢献する．

次に，メッセージの生存時間について考える．メッセージの生存時間は，メッセージの発生からそのメッセージのオリジナルと全ての複製がネットワーク中から消滅するまでの時間である．メッセージの生存時間は，$\boldsymbol{Z}(0) = (\boldsymbol{e}_1, 1, 0)$ という条件のもとでのマルコフ連鎖 $\{\boldsymbol{Z}(t);\ t \geq 0\}$ において，状態の集合 $(\mathbf{0}, 0, m)$ $(m = 2, 3, \ldots, N+1)$ のいずれかの状態へ吸収されるまでの初到達時間 T_E とみなすことができる．形式的には T_E は次式によって定義される．

$$T_E = \inf\{t \geq 0;\ \boldsymbol{Z}(t) \in \mathcal{E}_P\} \tag{2.7}$$

ただし，$\mathcal{E}_P$ は吸収状態の集合 $\{(\mathbf{0}, 0, m);\ m = 2, 3, \ldots, N+1\}$ である．平均メッセージ配送遅延 $\mathrm{E}[T_D]$ と同様にして，平均生存期間 $\mathrm{E}[T_E] := \mathrm{E}[T_E \mid \boldsymbol{Z}(0) = (\boldsymbol{e}_1, 1, 0)]$ を計算するための後進再帰手順を考えることができる．詳細は付録 B.1 において述べる．$\mathrm{E}[T_E]$ の性質は $\mathrm{E}[T_D]$ とほとんど等しい結果であるため，$\mathrm{E}[T_E]$ の結果は付録 B.1 に示す．

2.4.2 バッファ使用量

次に，一つのメッセージに対してランダムに選ばれた端末が被る負荷 O_L について議論する．端末負荷 O_L の形式的な定義は次式で与えられる．

$$O_L = \frac{R(\boldsymbol{e}_1, 1, 0)}{N+1}$$

ただし，$R(\boldsymbol{e}_1, 1, 0)$ は $\boldsymbol{Z}(0) = (\boldsymbol{e}_1, 1, 0)$ という条件のもとでのメッセージの生存期間中にシステム内に存在したメッセージの平均累積数を表す．

$$R(\boldsymbol{e}_1, 1, 0) = \mathrm{E}\left[\int_0^{T_E} K(t)dt \mid \boldsymbol{Z}(0) = (\boldsymbol{e}_1, 1, 0)\right] \tag{2.8}$$

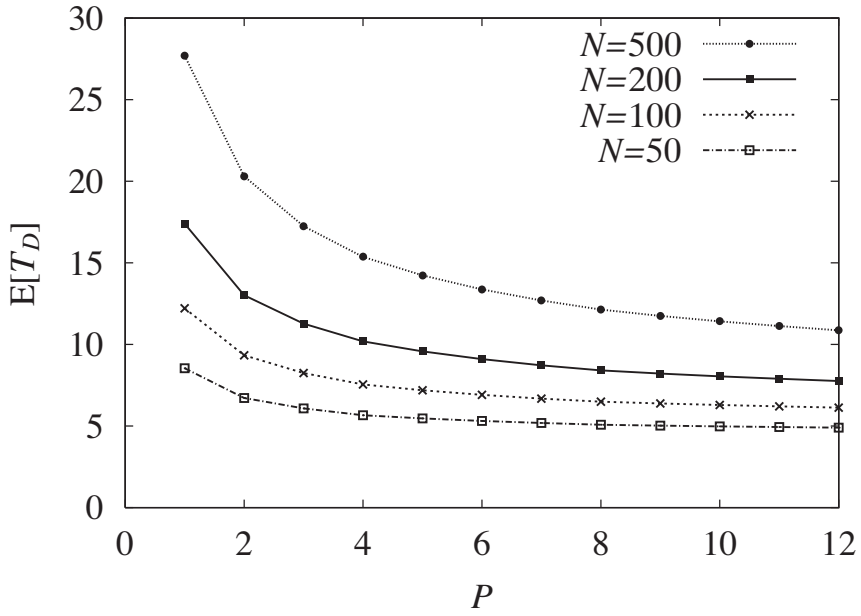


図 2.8: 平均配送遅延 E[T_D]

$R(\boldsymbol{e}_1, 1, 0)$ の計算手順は付録 B.2 に示す．

端末負荷 O_L は以下のように解釈することができる [50]．各端末が無限容量のバッファを持ち，単位時間当たり平均で Λ 個のオリジナルのメッセージを生成する状況を考える．ランダムに選ばれた端末が持つ平均メッセージ数 ρ は $\rho = (N+1)\Lambda O_L$ で与えられる．これは，G/G/∞ 待ち行列にリトルの公式を適用することで得られる．バッファのサイズが十分大きな場合，ランダムに選ばれた端末の平均メッセージ数 ρ は端末負荷 O_L に比例する．したがって，端末負荷 O_L をバッファ使用量の指標としてみなすことができる．

図 2.9 は散布端末の最大数 P に対する端末負荷 O_L の変化を示している．P の増加に対して単調に O_L は増加していることが分かる．直感的に明らかなように P の増加に伴って必要となるバッファ使用量は増加する．さらに，O_L は P に関する凹関数になっており，小さな P に対して O_L の増加は大きい．したがって，バッファ使用量に関して P の増加は悪い影響を与える．特に，P が小さな場合にその影響が大きい．よって，散布端末の最大数 P の増加に伴って平均配送遅延 E[T_D] は減少できるが，一方で端末負荷 O_L の増加を招く．すなわち，E[T_D] と O_L の間にはトレードオフの関係があることが分かる．

2.4.3 エネルギー消費量

二つの端末が遭遇した場合，これらの端末が保持するメッセージとアンチパケットに関すリストを必ず交換する．この情報交換後，メッセージの複製の転送を行う．リストに含まれるアンチパケット数は時間に対して非減少であり，一旦アンチパケット数が M_{anti} に達すると，その後変化しない（節 2.2.3 を参照）．したがっ

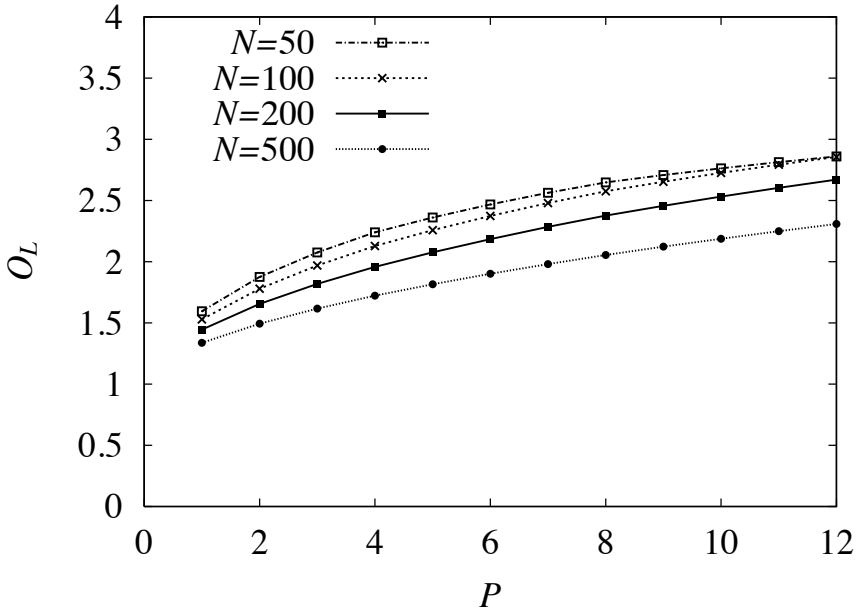


図 2.9: 一つのメッセージの発生によってランダムに選ばれた端末が被る端末負荷 O_L

て，以下では，ネットワークが長期間運用されており，各端末が M_{anti} 個のアンチパケットを保持していることを前提として議論する．この前提によってアンチパケットに関するリストの交換によるエネルギー消費は散布端末の最大数 P と無関係になるため，エネルギー消費は次の二つの要因によって生じる．一つはメッセージの複製の送信によるものであり，もう一つはメッセージに関するリストの送信によるものである．以下では，これら二つの要因について評価を行う．

まず，メッセージの消滅までに複製を転送した回数の平均である平均複製転送回数 $\mathrm{E}[N_D]$ について考える．$\mathrm{E}[N_D]$ の計算手順は付録 B.3 で述べる．図 2.10 は散布端末の最大数 P に対する平均複製転送回数 $\mathrm{E}[N_D]$ を示している．$\mathrm{E}[N_D]$ は P の増加凹関数になっており，P が小さなとき $\mathrm{E}[N_D]$ の増加が大きい．

次に，ある一つのメッセージに関する情報が送信される平均回数 $\mathrm{E}[N_I]$ について考える．$\mathrm{E}[N_I]$ は次式で与えられる [50]．

$$\mathrm{E}[N_I] = N\lambda R(\boldsymbol{e}_1, 1, 0)$$

ただし，$R(\boldsymbol{e}_1, 1, 0)$ は式 (2.8) で定義される．図 2.11 は散布端末の最大数 P に対する $\mathrm{E}[N_I]$ の変化を示している．$\mathrm{E}[N_I]$ もまた，P の増加凹関数であることが分かる．

エネルギー消費に関係する二つの要因 $\mathrm{E}[N_D]$ と $\mathrm{E}[N_I]$ がいずれも散布端末の最大数 P に対して単調に増加するため，エネルギー消費は P に対して増加する．よって，P の増加に伴って平均配送遅延 $\mathrm{E}[T_D]$ は減少するが，一方でエネルギー消費が増加を招く．したがって，$\mathrm{E}[T_D]$ とエネルギー消費量の間にはトレードオフの関係がある．

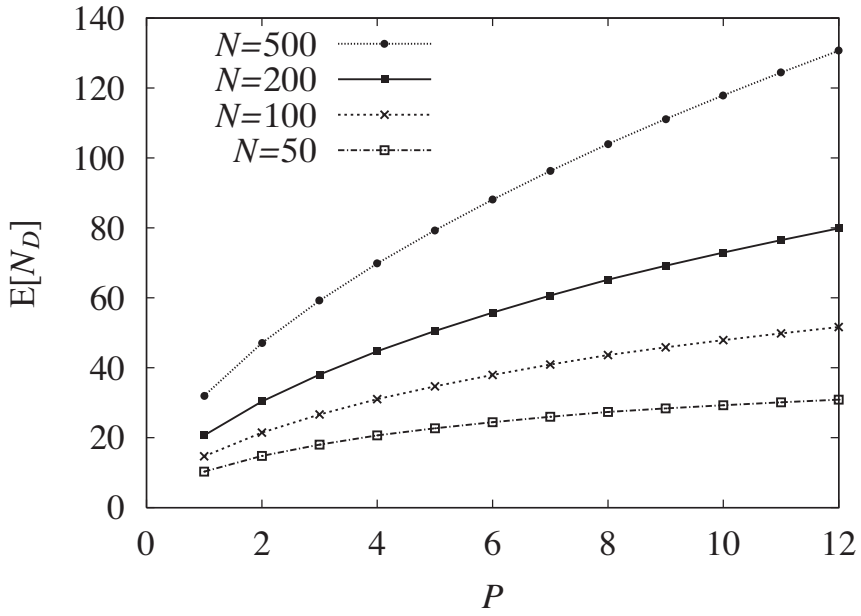


図 2.10: 平均複製転送回数 $\mathrm{E}[N_D]$

2.4.4 トレードオフの評価

平均配送遅延 $\mathrm{E}[T_D]$ と端末負荷 O_L の間のトレードオフの評価を行うために，評価指標 I_{BR} [50] を用いる．

$$I_{\mathrm{BR}} = \mathrm{E}[T_D]O_L/C_{\mathrm{BR}}$$

ただし，正規化定数 C_{BR} は感染型ルーチング（$P = \infty$）の $\mathrm{E}[T_D]O_L$ で与えられる．I_{BR} は以下の考えに基づいて設計されている．平均配送遅延が感染型ルーチング方式の n 倍で，端末負荷が $1/n$ 倍であるルーチング方式があるとする．このルーチング方式と感染型ルーチング方式のバッファ使用量に関する効率性は同等であると見なす．したがって，I_{BR} が 1 より小さいルーチング方式の場合，感染型ルーチング方式よりもバッファ使用量に関して効率的であると評価する．

図 2.12 は散布端末の最大数 P に対するバッファ使用量に関するトレードオフ指標 I_{BR} の変化を示している．I_{BR} は P に関する凸関数になっており，$N = 50, 100, 200, 500$ のときにそれぞれ $P = 2, 4, 5, 8$ で最小値をとる．したがって，小さな P を設定したマルチスプレッダルーチング方式がバッファ使用量に関して最も効率的である．この結果は文献 [50] の結論と一致している．すなわち，最適なルーチング方式は 2 ホップ転送方式 $(P = 1)$ よりも少し積極的にメッセージの複製を転送するものである．また，N の増加に伴って最適な P に対する I_{BR} は減少する．したがって，感染型ルーチング方式に対して適切にパラメタを設定したマルチスプレッダルーチング方式の性能は，端末数 N が増加すればするほど相対的に改善されていく．この特徴は，２ホップ転送方式（$P = 1$）の性能が端末数 N の増加に伴って相対的に悪化することと対照的である．

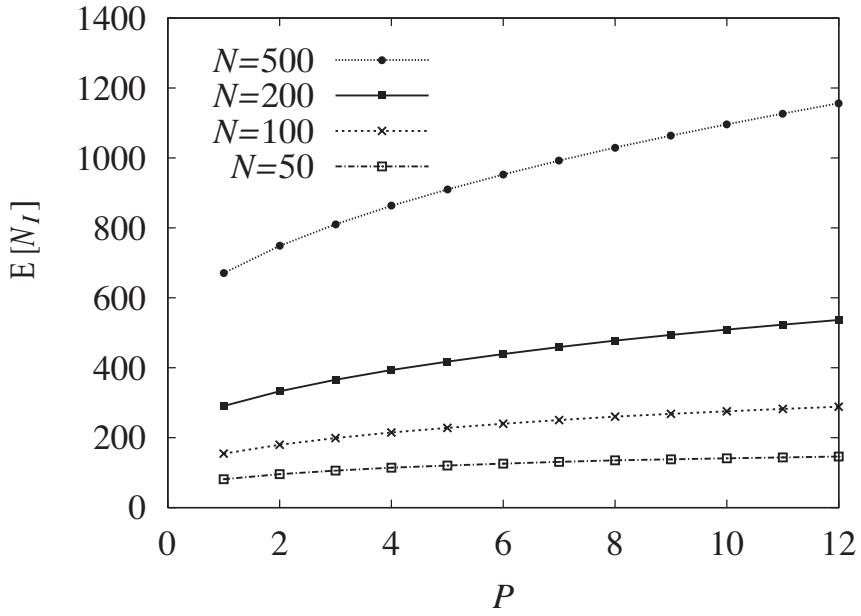


図 2.11: ある一つのメッセージに関する情報の平均送信回数 $\mathrm{E}[N_I]$

次に平均配送遅延 $\mathrm{E}[T_D]$ とエネルギー消費に関するトレードオフについて評価する．この評価を行うために，トレードオフ指標 I_{EC} [50] を用いる．

$$I_{\mathrm{EC}} = \mathrm{E}[T_D](\mathrm{E}[N_D] + \alpha\mathrm{E}[N_I])/C_{\mathrm{EC}}$$

ただし，正規化定数 C_{EC} は，感染型ルーチング方式の $\mathrm{E}[T_D](\mathrm{E}[N_D] + \alpha\mathrm{E}[N_I])$ である．また，α はメッセージの複製の転送に要するエネルギー消費量に対するメッセージに関する情報の伝送に要するエネルギー消費量の比率を表す．例えば，もしメッセージに関する情報量が 20 バイトでメッセージ長が 1,000 バイトの場合，データ伝送によるエネルギー消費は情報量に比例するものと仮定し，$\alpha = 0.02$ と設定する．I_{EC} は I_{BR} の考え方と同じ論理的根拠に基づいて設計されている．

図 2.13 は散布端末の最大数 P に対するエネルギー消費に関するトレードオフ指標 I_{EC} の変化を示している．ただし，$\alpha = 0.02$ と設定した．I_{EC} は P に対して単調に増加し，いずれの P に対しても 1 より小さい．したがって，マルチスプレッダルーチング方式は感染型ルーチング方式 $(P = \infty)$ よりもエネルギー消費に関して効率的である．2 ホップ転送方式 $(P = 1)$ は N に依らずエネルギー消費量に関して最も効率的である．また，N の増加に伴ってエネルギー消費の効率性が増加していることが分かる．

トレードオフ指標 I_{BR} と I_{EC} はそれぞれ異なる特性を持つため，これらのトレードオフ指標に関してパレート最適な P $(P = 1, 2, \ldots, 12)$ について考える．図 2.14 は散布端末の最大数 P に対する $(I_{\mathrm{EC}}, I_{\mathrm{BR}})$ を表している．$N = 50$, 100, 200, 500 に対してパレート最適フロントはそれぞれ $\{1, 2\}$, $\{1, 2, 3, 4\}$, $\{1, 2, \ldots, 5\}$, $\{1, 2, \ldots, 8\}$ である．端末数の増加に伴ってパレート最適フロントは広がるが，いずれの端末数に対してもパレート最適な P が小さいことが分かる．また，2 ホッ

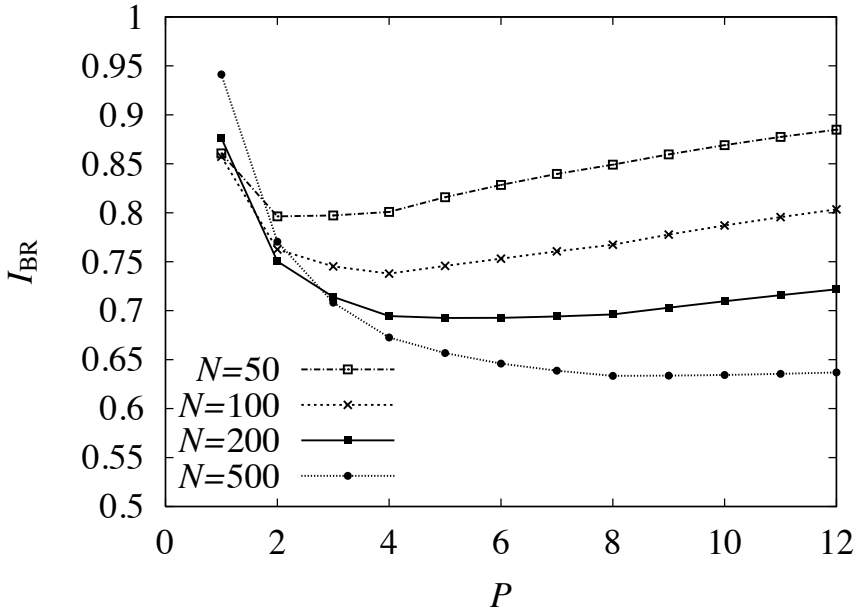


図 2.12: バッファ使用量に関するトレードオフ指標 I_{BR}

プ転送方式（$P = 1$）は常にパレート最適であり，合理的なメッセージ配送戦略であることが分かる．しかしながら，わずかに P を増加させることでエネルギー消費に関する効率性を少し悪化させることになるが，バッファ使用量に関する効率性を大きく改善できる．

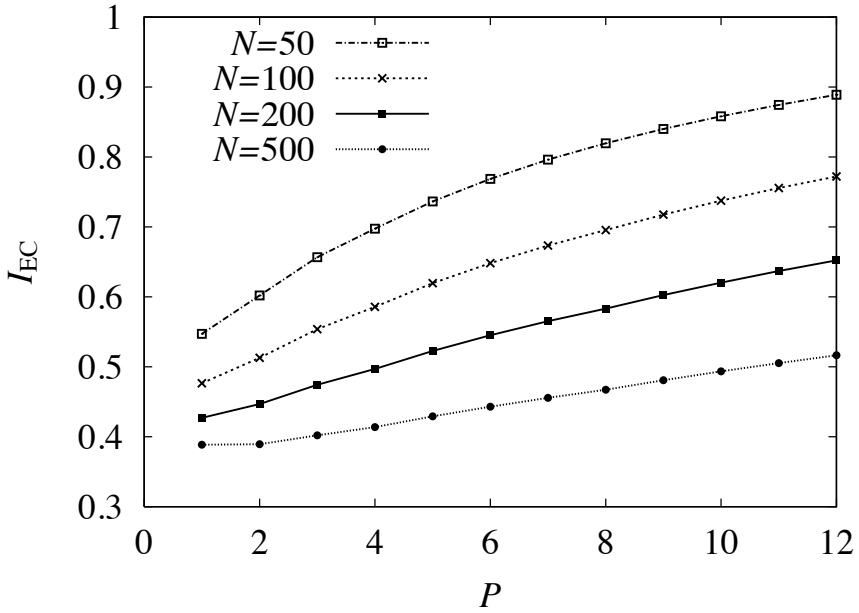


図 2.13: エネルギー消費量に関するトレードオフ指標 I_{EC} ($\alpha = 0.02$)

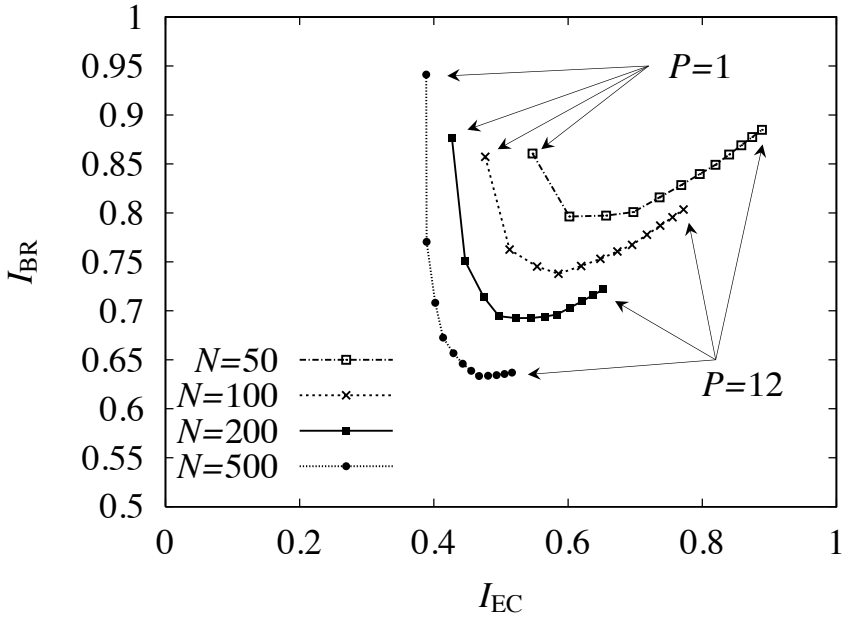


図 2.14: 最適パレートフロント: I_{BR} と I_{EC} の関係

2.4.5 既存のネットワーク情報を用いないルーチング方式との性能比較

本節では，マルチスプレッダルーチング方式と既存のネットワーク情報を用いないルーチング方式との性能比較を行う．具体的には Spray and Wait 方式と (p, q) 感染型ルーチング方式に含まれる代表的な転送方式を比較方式とする．(p, q) 感染型ルーチング方式では，送信元端末が別の端末に遭遇した場合確率 q で，メッセージの複製を持つ端末が別の端末に遭遇した場合確率 p でメッセージの複製を転送する．以降では，感染型ルーチング方式（$p = q = 1$），２ホップ転送方式（$p = q > 0$），確率転送方式（$p = q > 0$），$(p, 1)$ 感染型ルーチング方式を（p, q）感染型ルーチング方式の代表例として比較方式に採用する．また，これらの全ての比較方式に回復手法として VACCINE 回復手法を併用させるものとする．

ネットワーク情報がほとんど役に立たない状況においては，マルチスプレッダルーチング方式や Spray and Wait 方式のパラメタ P や (p, q) 感染型ルーチング方式のパラメタ p を最適な値に設定することは困難である．したがって，ネットワーク内の端末数を推測し，その推測値に基づいてこれらのパラメタの値を設定する必要がある．このような状況を考慮するために，トレードオフ指標 I_{BR} に関して $N = 100$ の場合に最適なパラメタを設定する（表 2.3 を参照）．このとき，N を変化させ，誤った推測がルーチング方式の性能にどのような影響を与えるかを調べる．I_{BR} に関して最適な確率転送方式（$p = q$）は $p = q = 1$ の場合，すなわち，感染型ルーチング方式と一致するため，以降では，感染型ルーチング方式の結果のみを示す．

図 2.15 は N に対する平均配送遅延 $\mathrm{E}[T_D]$ を示している．$\mathrm{E}[T_D]$ は N の増加関数になっていることが分かる．感染型ルーチング方式は端末数に関して感度が低く，安定した性能を示している．マルチスプレッダルーチング方式，２ホップ転送方式および $(p, 1)$ 感染型ルーチング方式の平均配送遅延は N に関して凹関数になっているが，Spray and Wait 方式の平均配送遅延はほとんど線形に増加している．また，N に対して広い範囲においてマルチスプレッダルーチング方式は五つの方式の中で２番目に良い性能であることを示している．

図 2.16 は N に対する端末負荷 O_L の変化を示している．感染型ルーチング方式の O_L は N に対して増加するが，一方で他のルーチング方式の O_L は N に関して減少凸関数である．マルチスプレッダルーチング方式は 2 番目に大きな負荷を被っているが，これによって平均配送遅延は 2 番目に良い性能を達成できていることが分かる（図 2.15 を参照）．これらの観察から，マルチスプレッダルーチング方式はメッセージの複製をかなり積極的に転送していることが分かる．

図 2.17 はバッファ使用量に関するトレードオフ指標 I_{BR} を示している．ただし，感染型ルーチング方式の場合，$I_{\mathrm{BR}} = 1$ である．四つのルーチング方式の I_{BR} は 1 より小さくなっており，感染型ルーチング方式よりもバッファ使用量に関して効率的なメッセージ配送を行えていることが分かる．さらに，マルチスプレッダルーチング方式と Spray and Wait 方式は２ホップ転送方式や $(p, 1)$ 感染型ルーチ

表 2.3: I_{BR} に対する最適パラメタ ($N = 100$ の場合)

ルーチング方式	最適パラメタ
マルチスプレッダ	$P = 4$
Spray and Wait	$P = 23$
確率転送方式	$p = q = 1$
$(p, 1)$ 感染型	$p = 0.015$

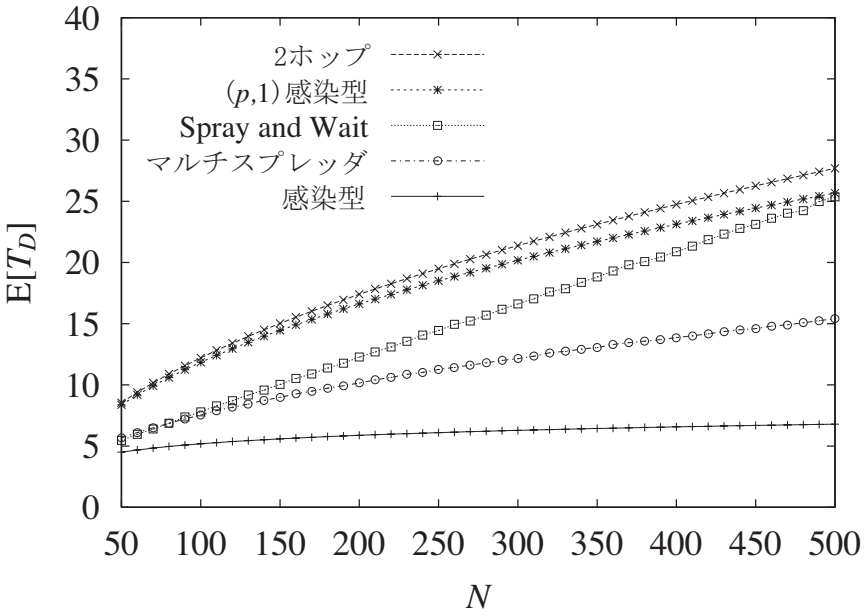


図 2.15: 平均配送遅延 $\mathrm{E}[T_D]$

ング方式よりも良い性能を示している．広範囲の N に対して N の増加に伴いマルチスプレッダルーチング方式の I_{BR} は減少しているが，他のルーチング方式の I_{BR} は N の凸関数であり，$N = 100$ の近くにおいて最小値をとっている．このマルチスプレッダルーチング方式における独自の特徴は N の増加に伴い $\mathrm{E}[T_D]$ は緩やかに増加し，一方で O_L は緩やかに減少することに起因している．したがって，小さな P に設定されたマルチスプレッダルーチング方式は広範囲の N に対してそれなりに良い性能を達成できており，N に対してロバストである．

図 2.18 は N に対する平均複製転送回数 $\mathrm{E}[N_D]$ を示している．$\mathrm{E}[N_D]$ は N に関する増加凹関数である．マルチスプレッダルーチング方式はメッセージの複製を積極的に転送するため，$\mathrm{E}[N_D]$ に関して 2 番目に悪い性能を示している．Spray and Wait 方式では中継端末の最大数を制限しているため，$\mathrm{E}[N_D]$ は $P = 23$ よりも常に小さい．

図 2.19 は N に対するエネルギー消費量に関するトレードオフ指標 I_{EC} の変化を示している．各ルーチング方式の I_{EC} は N に関する減少凸関数であり，いずれの N に対しても 1 より小さい．したがって，四つのルーチング方式は感染型ルー

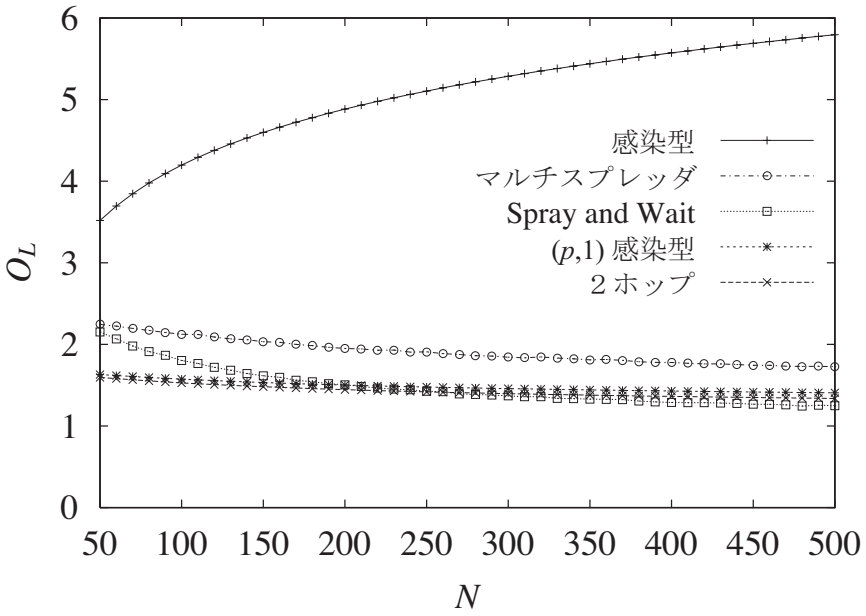


図 2.16: ランダムに選ばれた端末が一つのメッセージの発生によって被る端末負荷 O_L

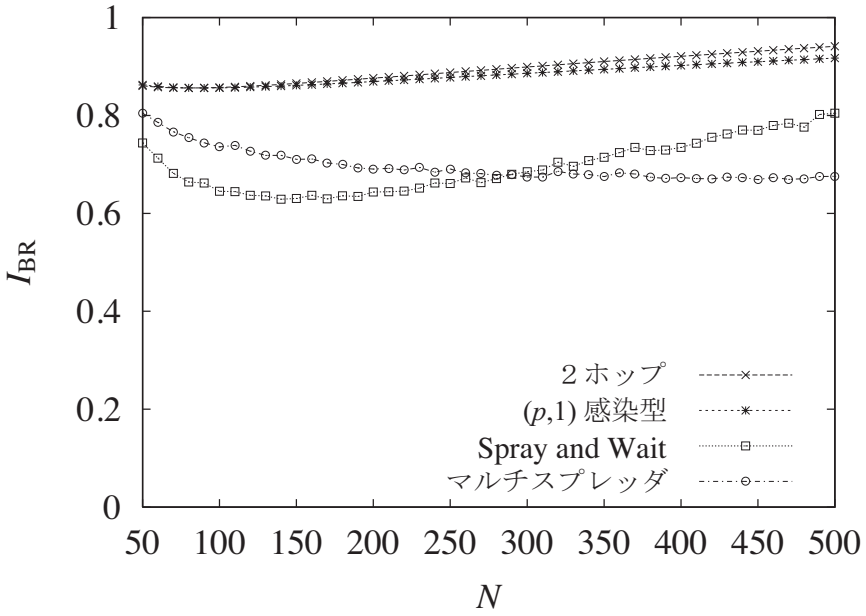


図 2.17: バッファ使用量に関するトレードオフ指標 I_{BR}

チング方式よりもエネルギー消費量に関して効率的なメッセージ配送を行えていることが分かる．これらの四つのルーチング方式のうち，マルチスプレッダルーチング方式はエネルギー消費量に関する効率性が最も低い．一方で，Spray and Wait 方式は中継端末の最大数を制限するため，広範囲の N に対して最も良い性能を示しているが，Spray and Wait 方式の I_{EC} は N が小さな場合 N に対する感度がかなり高いことが分かる．

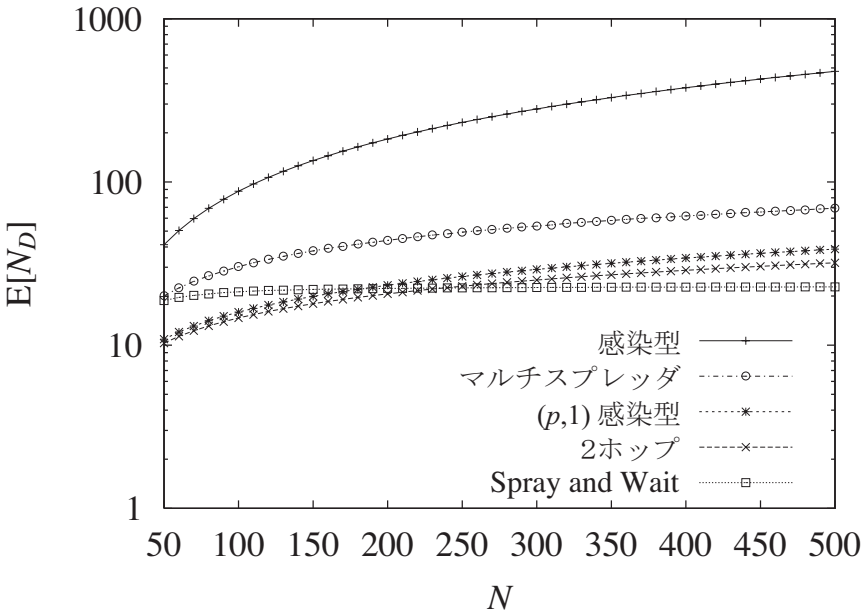


図 2.18: 平均複製回数 E[N_D]

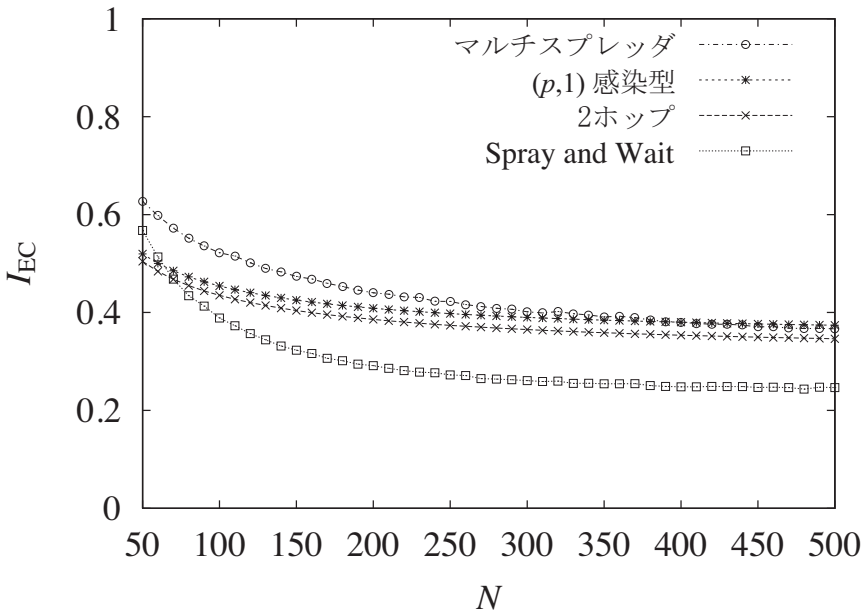


図 2.19: エネルギー消費に関するトレードオフ指標 I_{EC}

2.4.6 リアルトレースデータを用いたシミュレーション結果

これまでの議論では，同質な端末が独立でランダムに移動することを前提として，VACCINE 回復手法を用いたマルチスプレッダルーチング方式の基本的な性質について明らかにしてきた．特に，マルチスプレッダルーチング方式では P を小さく設定することが最適であることを示した．本節では，二つの異なるリアルトレースデータを用いたシミュレーション実験においても，同質なモデルの場合の結果と同様の傾向がみられるかを確かめる．シミュレーション実験において用いるトレースデータとしては Haggle データセット [12] のトレースデータ [90] とミラノ大学 (milano university) におけるトレースデータ [21, 22, 91] である．さらに，マルチスプレッダルーチング方式と Spray and Wait 方式の定性的な違いを明らかにする．Spray and Wait 方式では各メッセージに対して高々P 個の中継端末を生成するのに対してマルチスプレッダルーチング方式では P 個の散布端末が中継端末を制限なく生成できる．この特性の違いがどのように性能に影響を与えるかを明らかにする．

Haggle データセットのトレースデータには，バルセロナで開催された国際会議 INFOCOM 2006 においてモバイル端末 iMotes を持った 78 人の参加者（$N = 77$）の遭遇情報が記録されている．シミュレーション実験では，国際会議の初日から最終日に対応する 50,000 秒から 340,808 秒までのデータを用いた．この期間に含まれる遭遇の総回数は 105,427 回である．任意の二つの端末が遭遇する率とランダムに選ばれた二つの端末が遭遇する率の平均をそれぞれ Λ, λ とする．

$$\Lambda = \frac{105,427}{340,808 - 50,000} \approx 0.36$$

したがって， $\lambda = \Lambda/\{N(N+1)/2\} \approx 1.2 \times 10^{-4}$ であり，ランダムに選ばれた端末が別の端末と遭遇する平均遭遇時間間隔は $(N\lambda)^{-1} \approx 107.58$ である．

一方で，ミラノ大学におけるトレースデータには Pocket Mobility Trace Recorder と呼ばれる無線端末を持つ 44 人（$N = 43$）の参加者が 19 日間わたる遭遇情報が記録されている．シミュレーション実験では，400,000 秒から 1,632,832 秒までのデータを用いた．この期間内の遭遇の総回数は 7,592 回である．

$$\Lambda = \frac{7,592}{1,632,832 - 400,000} \approx 6.2 \times 10^{-3}$$

したがって， $\lambda = \Lambda/\{N(N+1)/2\} \approx 6.5 \times 10^{-6}$ であり，ランダムに選ばれた端末が別の端末に遭遇する平均時間間隔は $(N\lambda)^{-1} \approx 3.6 \times 10^{3}$ である．

これらのシミュレーション実験では端末が持つバッファ容量は十分に大きいものとした．Haggle データセットのトレースデータを用いた実験では時刻 $t = 50,000 + 10,000i$ $(i = 0, 1, \ldots, 5)$ に，ミラノ大学のトレースデータを用いた実験では時刻 $t = 400,000 + 50,000i$ $(i = 0, 1, \ldots, 6)$ に各端末が N 個のメッセージを発生させ，それぞれ異なる N 個の端末を受信先端末とする．しかしながら，これらのメッセージの中には感染型ルーチング方式を用いたとしてもメッセージ配送を行えないメッ

セージが Haggle データセットの場合 183 個，ミラノ大学のトレースデータの場合 1430 個含まれているため，これらのメッセージは発生させないものとした．したがって，Haggle データセットの場合，$35,853\ (= 6N(N+1) - 183)$ 個のメッセージを発生させ，ミラノ大学のトレースデータの場合，$11,814\ (= 7N(N+1) - 1430)$ 個のメッセージを発生させた．

まず，最初に不到達率について調べる．不到達率はシミュレーション実験の終了時点において受信先端末に到達していないメッセージの割合と定義する．図 2.20(a) と図 2.20(b) はそれぞれ Haggle データセットとミラノ大学のトレースデータを用いた場合のパラメタ P に対する不到達率の変化を示している．直接配送方式（$P = 1$ の場合の Spray and Wait 方式）では Haggle データセットよりもミラノ大学のトレースデータを用いた場合の不到達率が大きい．この結果から ミラノ大学のトレースデータより Haggle データセットのトレースデータを用いた場合の方がメッセージ配送がはるかに容易であることが分かる．以降では，Haggle データセットとミラノ大学のトレースデータを用いたシミュレーション実験をそれぞれ Haggle シナリオとミラノシナリオと呼ぶことにする．

図 2.20 から $P = 1$ (2 ホップ転送方式) の場合を除いてマルチスプレッダルーチング方式の不到達率はいずれのシナリオにおいてもゼロになっている．よって，$P \geq 2$ のマルチスプレッダルーチング方式は不到達に関して感染型ルーチング方式と同等の性能を示している．一方で，Spray and Wait 方式の不到達率がゼロとなるのは Haggle シナリオの場合 $P \geq 8$ であり，ミラノシナリオの場合 $P \geq 21$ である．この結果から Spray and Wait 方式は端末間の遭遇の程度に対する感度が高いことが分かる．これに対して，小さな P のマルチスプレッダルーチング方式はどちらのシナリオにおいても良い性能を示しており，Spray and Wait 方式よりもマルチスプレッダルーチング方式がシナリオの違いに対してロバストである．

次に，平均配送遅延と端末負荷について考える．シミュレーション実験の終了時点において不到達であるメッセージは，さらに長期間にわたってトレースデータを収集していれば受信先端末にメッセージ配送ができていた可能性がある．そこで，ここでは二つの配送遅延の評価方法を考える．一つはシミュレーション実験が終了するまでに到達したメッセージのみを対象とした平均配送遅延であり，もう一つは不到達のメッセージはシミュレーション実験の終了時点において受信先端末に配送されたものと考えることで全てのメッセージを対象とした平均配送遅延である．ここで，$\overline{T}_D$ を平均遭遇時間間隔 $1/(N\lambda)$ によって正規化された平均配送遅延 $\overline{T}_D$ と定義する．

$$\overline{T}_D = T_D/(N\lambda)^{-1}$$

図 2.21(a) と図 2.21(b) はそれぞれ Haggle シナリオとミラノシナリオの場合のパラメタ P に対する正規化された平均配送遅延 $\mathrm{E}[\overline{T}_D]$ を示している．ただし，ラベルの “(all)” は平均配送遅延の計算に全てのメッセージを対象としていることを表す．いずれのシナリオにおいても，マルチスプレッダルーチング方式の $\mathrm{E}[\overline{T}_D]$ は感染型ルーチング方式（$P = \infty$）の $\mathrm{E}[\overline{T}_D]$ に急激に近づいている．一方で，Spray and Wait 方式の $\mathrm{E}[\overline{T}_D]$ はシナリオに依存している．ミラノシナリオよりも Haggle

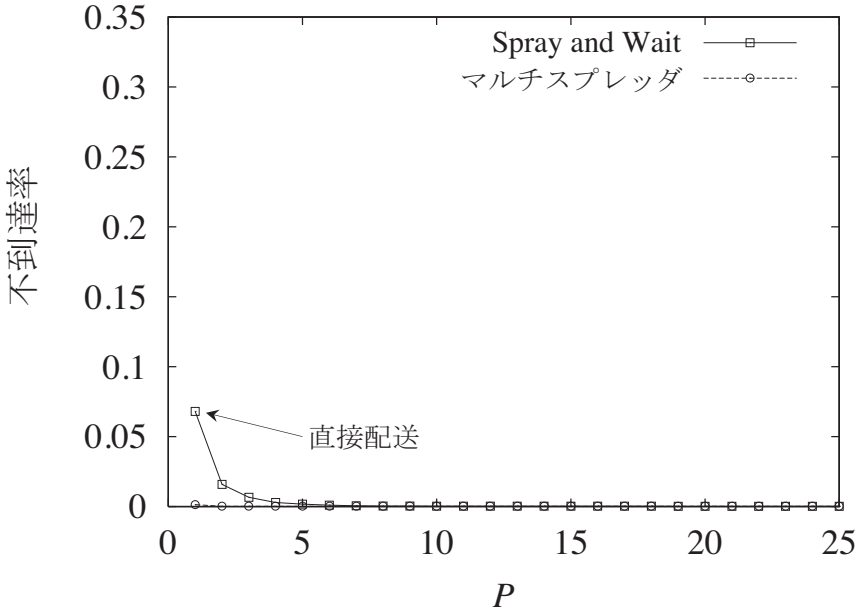
シナリオの方が Spray and Wait 方式の $\mathrm{E}[\overline{T}_D]$ の方が緩やかに感染型ルーチング方式の $\mathrm{E}[\overline{T}_D]$ に近づいている．この結果から Spray and Wait 方式のパラメタ P の設定が容易でないことが分かる．

図 2.22(a) と図 2.22(b) はそれぞれ Haggle シナリオとミラノシナリオの場合のパラメタ P に対するバッファ使用量に関するトレードオフ指標 I_{BR} の変化を示している．図 2.22(a) と図 2.22(b) から，マルチスプレッダルーチング方式の I_{BR} の最小値が $P = 4$ と $P = 2$ のときに達成している．この結果は節 2.4.4 における観察と一致している．すなわち，マルチスプレッダルーチング方式では小さな P が最適である．一方で，Spray and Wait 方式では最適な P がシナリオによって大きく異なる．また，P が適切でない場合，I_{BR} が感染型ルーチング方式よりも大きくなる．さらに，ミラノシナリオよりも Haggle シナリオの最適な P がかなり大きいことが分かる．よって，適切にパラメタを設定した Spray and Wait 方式はマルチスプレッダルーチング方式よりも I_{BR} に関して効率的なメッセージ配送を実現できるのだが，Spray and Wait 方式の P はかなり慎重に設定する必要がある．

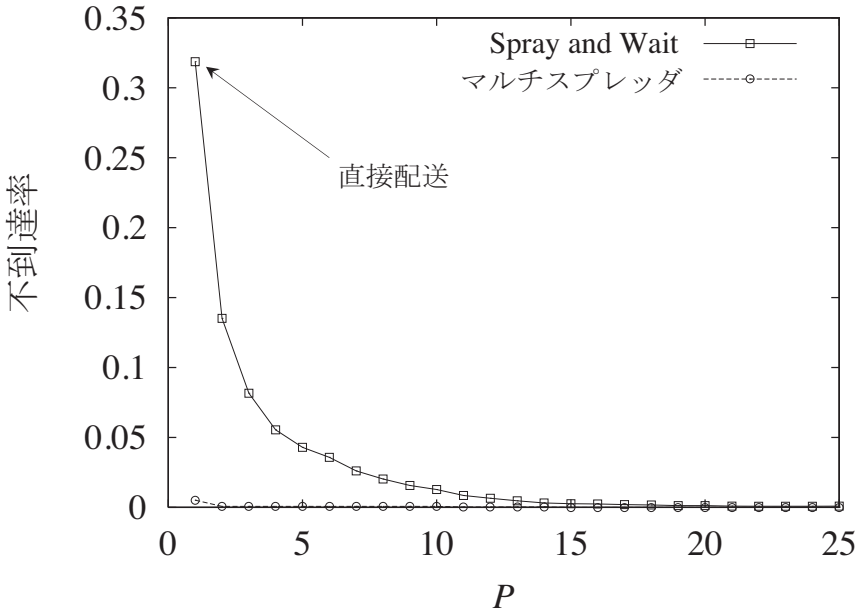
2.5 結言

本章では，散布端末の最大数を予め定めた P 個に制限するマルチスプレッダルーチング方式を提案した．マルチスプレッダルーチング方式では，特別な場合として 2 ホップ転送方式（$P = 1$）と感染型ルーチング方式（$P = \infty$）を含んでいる．まず，端末が同質であり，独立でランダムに移動しているという前提のもとで，マルチスプレッダルーチング方式の性能について理論解析を行った．具体的には，平均配送遅延やネットワーク資源消費量などの複数の指標が連続時間マルコフ連鎖の吸収状態への初到達時間と関連付けられること示した．さらに，性能評価指標を効率的に数値計算できる後進再帰アルゴリズムを示した．数値計算によって得られた結果に基づいて平均配送遅延とバッファ使用量やエネルギー消費量の間にあるトレードオフについて議論を行い，パレート最適な P が存在することを示した．パレート最適な P の範囲はシチュエーションに依存するのだが，小さな P に設定しておけば良いことを示した．さらに，この特徴がリアルトレースデータを用いたシミュレーション実験においても同じ傾向であることを確かめた．

また，リアルトレースデータを用いたシミュレーション実験によって，マルチスプレッダルーチング方式と Spray and Wait 方式の性能を比較し，マルチスプレッダルーチング方式が様々なネットワーク環境において安定した性能を達成できることを示した．この特徴はネットワーク情報がほとんど役に立たない場合に有効である．一方，Spray and Wait 方式の効率性は高いのだが，高々 P 個の中継端末のみが生成可能なため制御パラメタ P に対する感度がかなり高い．したがって，Spray and Wait 方式の最適なパラメタ P の推定が困難な場合，マルチスプレッダルーチング方式の利用が有効である．

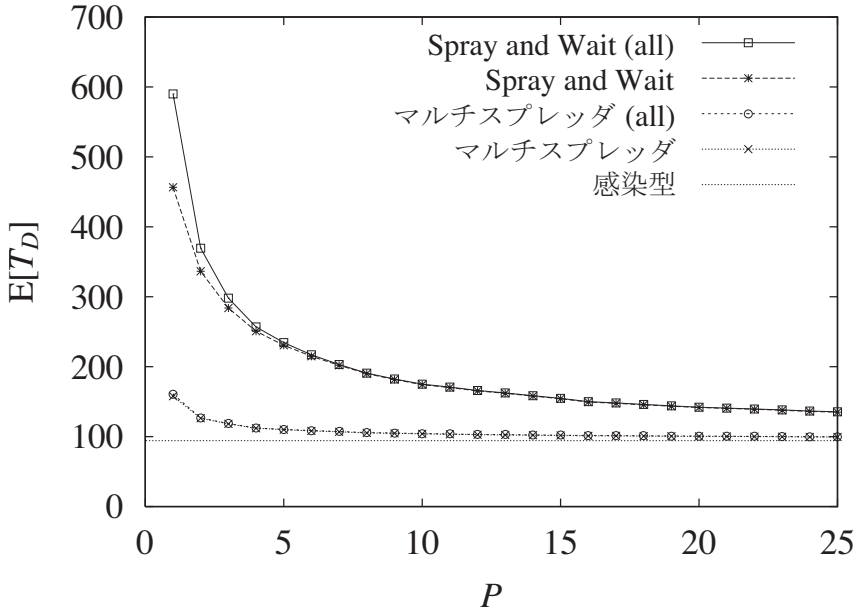


(a) Haggle データセット

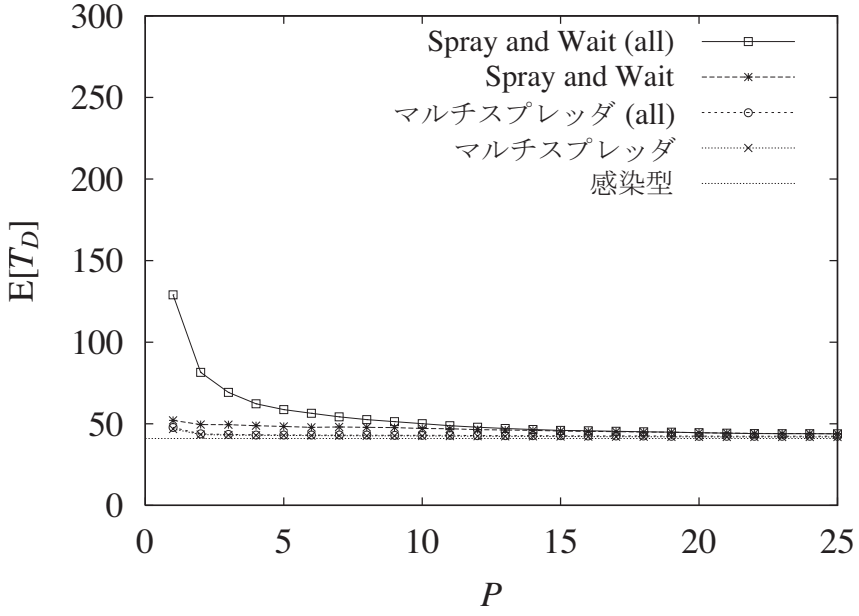


(b) ミラノ大学におけるトレースデータ

図 2.20: 不到達率

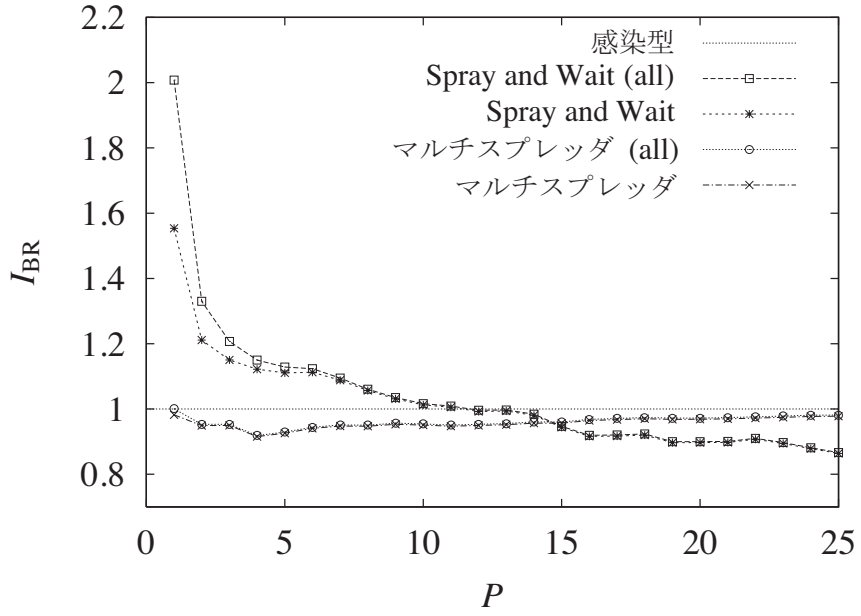


(a) Haggle シナリオ

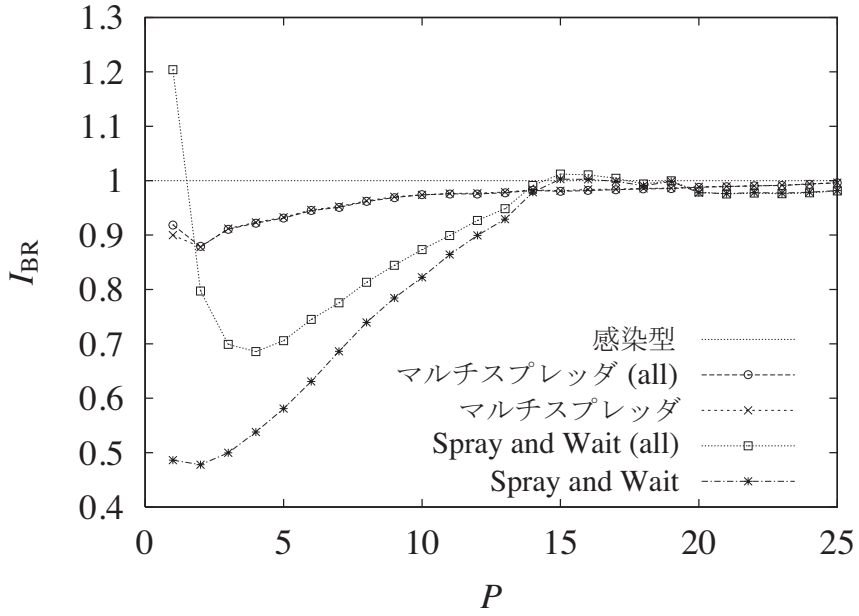


(b) ミラノシナリオ

図 2.21: 平均配送遅延 $\mathrm{E}[\overline{T}_D]$



(a) Haggle シナリオ



(b) ミラノシナリオ

図 2.22: バッファ使用量に関するトレードオフ指標 I_{BR}

第3章

端末密度推定に基づく確率転送方式の提案とその評価

3.1 緒言

本章では，場所型複数複製方式に分類される新たなルーチング方式について検討する．表 1.2 に示したように，既存のネットワーク情報を用いるルーチング方式では，端末型のルーチング方式と比較して場所型のルーチング方式に関する議論は少ない．そこで，本論文では，遭遇場所の特徴量の一つである端末密度を考慮した確率転送方式を提案し，端末密度情報の利用が効率的なメッセージ配送の実現に有効であることを示す．

提案方式では，端末密度の違いによってメッセージの拡散能力が大きく異なる点に着目する．端末密度の高い領域では端末間の遭遇が頻繁に発生するため，メッセージの拡散能力が高く，メッセージの複製の転送を抑制したとしても遅延性能への影響が少ない．一方で，端末密度の低い領域ではメッセージの拡散能力を維持するために，メッセージの複製の転送を積極的に行う方が良い．

そこで，この特性に着目した提案方式では，メッセージを持つ端末が別の端末と遭遇した場合，遭遇地点の端末密度に応じてメッセージの転送確率を制御する．遭遇地点の端末密度が高い場合には小さな確率で，低い場合には大きな確率でメッセージの複製を転送する．このように転送確率を制御することで，高密度領域ではメッセージ拡散を抑制し，低密度領域ではメッセージの拡散能力を維持する．

提案方式を実現するためには，端末密度の推定を行わなければならない．端末密度推定を行うために，各端末が GPS (Global Positioning System) などの位置測定装置を備えているものとし，別の端末と遭遇する度に遭遇地点の位置情報を記録する．さらに，記録した遭遇地点情報を端末間で交換し合うことによって，各端末がネットワーク全体の遭遇位置情報を収集するものとする．各端末は収集した遭遇地点情報に基づいて端末密度分布を推定することができる．

シミュレーション実験によって端末密度推定に基づく確率転送方式の有効性を示す．さらに，端末密度推定の精度に対する提案方式のロバスト性について議論する．端末密度推定による誤差は避けることができないため，推定精度に対する感度が高ければ，提案方式はほとんど役に立たないものとなる．提案方式のロバ

スト性を調べるために，以下の二つの点について議論を行う．一つ目は，ネットワーク内の端末密度が不均質な状況を想定し，推定精度の違いがルーチング性能にどのような影響を与えるかを明らかにする．具体的には，推定精度の異なる三つの推定法を考え，これらの推定法の違いが遅延性能にどのような影響を与えるか調べる．

二つ目としてネットワーク内の端末密度が均質な状況に対して提案方式を用いた際にどのような影響が現れるか議論する．既に述べたように，提案方式は端末密度が不均質な状況を想定して設計されている．よって，全ての端末が端末密度を正確に推定できたとしても，端末密度が均質なネットワーク環境においては効率的なメッセージ配送を期待できない．さらに，不正確な端末密度推定によって各端末は数カ所の地点を高密度と判別するため，誤った認識に基づいてメッセージの複製の転送が制御される．このような状況においてもルーチング性能が大きく悪化するのであれば，提案方式はほとんど役に立たないものとなる．

3.2 端末密度推定に基づく確率転送方式

3.2.1 提案方式の概要

本節では，端末密度推定に基づく確率転送方式の概要を述べる．$\mathcal{V}$ を閉領域内を移動する端末の集合と定義する．各端末は位置測定技術によって自身の位置を把握できているものとする．議論の簡単化のために，各メッセージ毎に一つの受信先端末へメッセージ配送を行うユニキャスト通信のみを考える．各メッセージに対して端末を送信元端末，受信先端末，複製保持端末，不保持端末の 4 種類に分類する．送信元端末はメッセージを生成した端末であり，受信先端末はメッセージの配送先である．複製保持端末はメッセージの複製を保持する端末である．他の全ての端末は不保持端末である．さらに，送信元端末と複製保持端末を区別する必要がない場合には，これらの端末を保持端末と呼ぶことにする．

保持端末が不保持端末と遭遇した場合，保持端末は確率 p_F でメッセージの複製を不保持端末に転送する．転送確率 p_F は，保持端末の遭遇時の位置における端末密度に応じて制御する．具体的には，保持端末が端末密度の高い領域に位置する場合，転送確率 p_F を p_{HD} に設定し，低い領域に位置する場合，p_{LD} に設定する．ただし，$0 \leq p_{\mathrm{HD}} \leq p_{\mathrm{LD}} \leq 1$ である．

端末密度推定に基づく確率転送方式を実現するためには，三つの構成技術が必要となる．すなわち，遭遇地点収集法，端末密度推定法，メッセージ転送基準の三つである．図 3.1 は提案方式のメッセージ転送手順を示している．遭遇地点収集法として以下の手順を採用する．ここで，$\mathcal{L}_v = \{\boldsymbol{x} \mid \boldsymbol{x} \in \mathbb{R}^2\}$ $(v \in \mathcal{V})$ を端末 v が保持する遭遇地点リストを表すものとする．端末 v が端末 $u \in \mathcal{V}$ と地点 $\boldsymbol{x}$ で遭遇した場合，端末 v は端末 u から $\mathcal{L}_u$ を受信する．$\mathcal{L}_u$ の受信後，$\mathcal{L}_v$ を $\mathcal{L}'_v \subseteq \mathcal{L}_v \cup \mathcal{L}_u \cup \{\boldsymbol{x}\}$ に更新する．各端末 v はその端末自身が保持する遭遇地点情報リスト $\mathcal{L}_v$ に基づいて端末密度分布を推定する．端末密度分布を推定する方法に

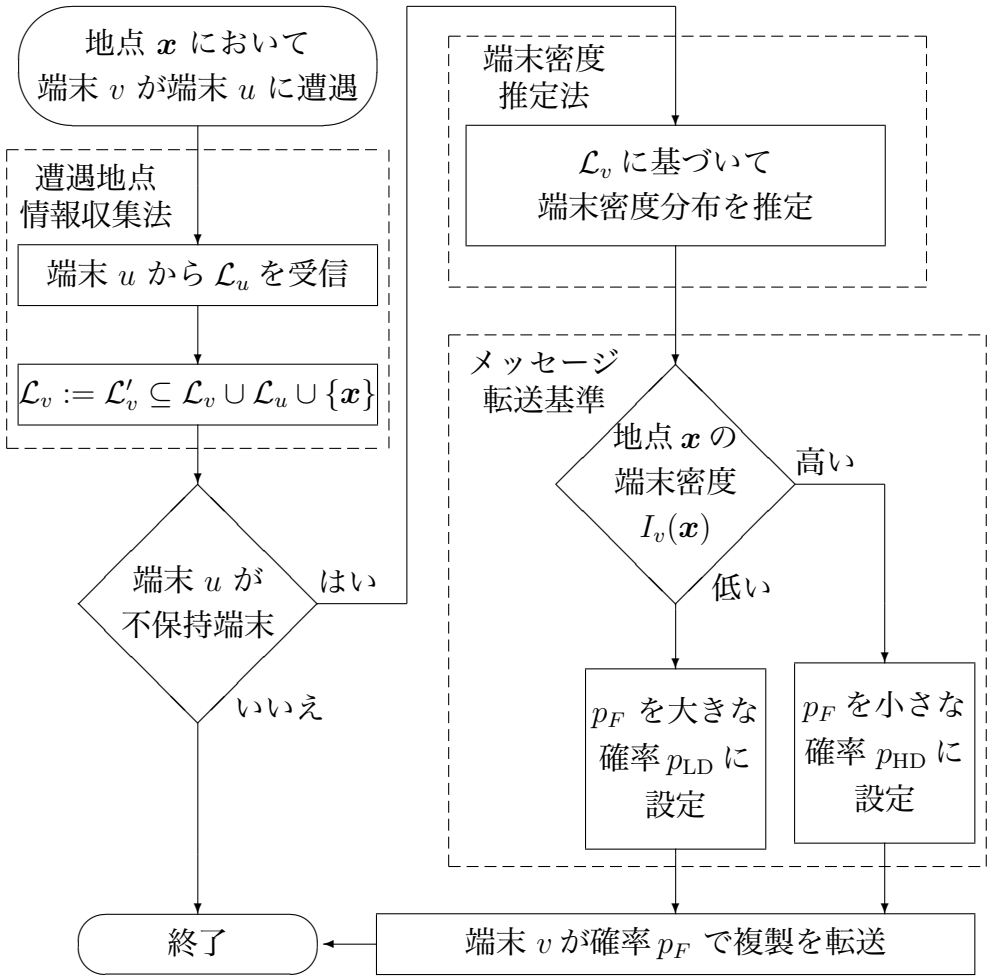


図 3.1: 端末 v が端末 u に遭遇した際の手順 (p_F は転送確率を表す)

ついては 3.2.2 節において詳しく述べる．さらに，メッセージ転送基準では一つの閾値 d_v を用いて端末密度の高い場所と低い場所を区別する．閾値 d_v は推定された端末密度分布の平均と標準偏差を用いて定義される．メッセージ転送基準については 3.2.3 節において詳しく述べる．

3.2.2 端末密度推定法

端末密度の変動がほとんど起こらず，遭遇地点情報を十分多く収集できる場合，ヒストグラム法によって端末密度分布を推定できる．ヒストグラム法では，領域全体を小領域に分割し，各小領域において遭遇数をカウントすることによって領域全体の端末密度分布を推定する．端末密度の時間的な変動が大きい場合や収集できる遭遇地点情報が少ない場合には，ヒストグラム法よりも少ない情報量で端末密度を推定する方法が必要となる．本論文では，このような端末密度推定法として EM アルゴリズムを用いたパラメトリック密度推定法とガウスカーネル関数を用いたノンパラメトリック密度推定法の二つを考える．どちらの密度推定法に

おいても，推定される端末密度分布は混合ガウス分布として与えられる．したがって，これらの密度推定法によって得られる分布は用いる遭遇地点情報量に関わらず滑らかである．

パラメトリック密度推定法

パラメトリック密度推定法では，各端末 v は遭遇地点情報リスト $\mathcal{L}_v$ に基づいて混合ガウス分布のパラメタ推定を行う．$I_v(\boldsymbol{x})$ は端末 v が推定する端末密度分布を表す．ただし，$\boldsymbol{x}$ は閉領域内の一点を表すものとする．$I_v(\boldsymbol{x})$ は次式で与えられる．

$$I_v(\boldsymbol{x}) = \sum_{i=1}^{K} \frac{\pi_i}{2\pi|\Sigma_i|^{1/2}} \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)}{2}\right) \tag{3.1}$$

ただし，K は事前に定義される混合ガウス分布のパラメタ，$\boldsymbol{\mu}_i \in \mathbb{R}^2$ は平均ベクトル，Σ_i は 2×2 の共分散行列，π_i は混合率を表す．EM アルゴリズム [5] を用いることによって $I_v(\boldsymbol{x})$ の最尤推定解を求めることができる．EM アルゴリズムでは，E ステップと M ステップと呼ばれる手順を交互に繰り返し実行することで，最終的にパラメタが収束し，その収束した値が $I_v(\boldsymbol{x})$ の最尤推定解である．

EM アルゴリズムを実行するために，K の設定および平均ベクトルの初期値 $\boldsymbol{\mu}_i^{(0)}$，共分散行列の初期値 $\boldsymbol{\mu}_i \in \mathbb{R}^2$，混合率の初期値 $\pi_i^{(0)}$ の設定を行わなければならない．本論文では，$\pi_i^{(0)}$ を $\pi_i^{(0)} = 1/K$ とし，$\Sigma_i^{(0)}$ を 2×2 単位行列にする．K と $\boldsymbol{\mu}_i^{(0)}$ を設定には，ヒストグラム法を用いた大まかな密度推定が役に立つ．つまり，大まかな密度推定の結果から K を相対的に密度の高い領域の数に設定し，$\boldsymbol{\mu}_i^{(0)}$ をそれらの高密度領域に含まれる 1 点に設定する．本論文では，各端末が適切な K と $\boldsymbol{\mu}_i^{(0)}$ を設定できるものと仮定し，大まかな密度推定の結果に基づいて K と $\boldsymbol{\mu}_i^{(0)}$ を設定する方法については今後の課題とする．

一旦，π_i，$\boldsymbol{\mu}_i$，Σ_i が決定されると，その後 $\mathcal{L}_v$ を保持する必要はない．パラメタの値を更新する方法についてはネットワークの環境に大きく依存するため，本論文では議論しないものとする．端末密度の変動がほとんどない場合には，ジャンピングウィンドウを用いるのが最も簡単な更新方法である．一方で，端末密度の変動が大きな場合には，パラメタの更新にスライディングウィンドウを用いる方が良い可能性が高い．

端末密度分布の平均と分散は以下のように求めることができる．ここで，$\mathcal{A}$ をネットワーク内の端末が移動する閉領域とする．また，$I_v(\boldsymbol{x})$ は閉領域 $\mathcal{A}$ の境界付近における影響を無視できるものと仮定する．このとき，$I_v(\boldsymbol{x})$ の平均 $\mathrm{E}[I_v]$ は近似的に次式で表される．

$$\mathrm{E}[I_v] = \int_{\boldsymbol{x}\in\mathcal{A}} I_v(\boldsymbol{x})\frac{d\boldsymbol{x}}{|\mathcal{A}|} \approx \int_{\boldsymbol{x}\in\mathbb{R}^2} I_v(\boldsymbol{x})\frac{d\boldsymbol{x}}{|\mathcal{A}|} = \frac{1}{|\mathcal{A}|} \tag{3.2}$$

ただし，$|\mathcal{A}|$ は閉領域 $\mathcal{A}$ の面積を表す．同様にして，$I_v(\boldsymbol{x})$ の分散 $\sigma^2[I_v]$ は近似的に次式で与えられる．

$$\begin{aligned}\sigma^2[I_v] \approx & \sum_{\substack{i,j=1\\ i\neq j}}^{K} c_{i,j} \exp\left(\frac{\boldsymbol{b}_{i,j}^T H_{i,j}^{-1}\boldsymbol{b}_{i,j} - \boldsymbol{\mu}_i^T\Sigma_i^{-1}\boldsymbol{\mu}_i - \boldsymbol{\mu}_j^T\Sigma_j^{-1}\boldsymbol{\mu}_j}{2}\right)\\ & + \sum_{i=1}^{K}\frac{\pi_i^2}{4\pi|\Sigma_i|^{1/2}|\mathcal{A}|} - \left(\frac{1}{|\mathcal{A}|}\right)^2 \end{aligned} \tag{3.3}$$

ただし，$H_{i,j}$, $\boldsymbol{b}_{i,j}$，$c_{i,j}$ は次式で与えられる．

$$H_{i,j} = \Sigma_i^{-1} + \Sigma_j^{-1} \tag{3.4}$$

$$\boldsymbol{b}_{i,j} = \Sigma_i^{-1}\boldsymbol{\mu}_i + \Sigma_j^{-1}\boldsymbol{\mu}_j \tag{3.5}$$

$$c_{i,j} = \frac{\pi_i\pi_j|H_{i,j}^{-1}|^{1/2}}{2\pi|\Sigma_i|^{1/2}|\Sigma_j|^{1/2}|\mathcal{A}|} \tag{3.6}$$

付録 C.1 において式 (3.3) の導出手順を述べる．また，領域 $\mathcal{A}$ を全平面 $\mathbb{R}$ に置き換える近似を行っているため，多くの端末が $\mathcal{A}$ の境界付近に位置している場合には推定誤差の増加を招く．

ノンパラメトリック密度推定法

本節では，ガウスカーネル関数を用いたノンパラメトリック密度推定法について述べる．端末密度分布 $I_v(\boldsymbol{x})$ は次式で与えられる．

$$I_v(\boldsymbol{x}) = \frac{1}{|\mathcal{L}_v|}\sum_{\boldsymbol{x}_i\in\mathcal{L}_v}\frac{1}{2\pi h^2}\exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{x}_i)^T(\boldsymbol{x}-\boldsymbol{x}_i)}{2h^2}\right) \tag{3.7}$$

ただし，h は平滑化パラメタである．式 (3.7) から，$\mathcal{L}_v$ の要素 $\boldsymbol{x}$ が端末密度分布 $I_v(\boldsymbol{x})$ のパラメタになっていることが分かる．したがって，端末密度を推定するために各端末 v は $\mathcal{L}_v$ を保持し続ける必要がある．保持する $\mathcal{L}_v$ のサイズを小さく抑えるために，本論文では，各端末が別の端末毎の最新の遭遇地点情報のみを保持する方法を採用する．二つの端末が遭遇し，互いの遭遇地点情報リストを交換した際に，同じ識別子の端末の情報が含まれている際には古い方の情報の削除を行う．付録 D.1 においてこの更新手順について詳述する．この手順によって $\mathcal{L}_v$ に含まれる遭遇地点情報は高々 $|\mathcal{V}|-1$ 個である．一般に，蓄積運搬転送型ルーチングが用いられる環境では，$|\mathcal{V}|$ は小さいため，少ない情報量で $I_v(\boldsymbol{x})$ を推定することになる．

パラメトリック密度推定法の場合と同様に，端末密度分布の境界付近における影響を無視することによって端末密度分布の平均 $\mathrm{E}[I_v]$ と分散 $\sigma^2[I_v]$ を得られる．具体的には，$I_v(\boldsymbol{x})$ の平均 $\mathrm{E}[I_v]$ は式（3.2）で与えられ，$I_v(\boldsymbol{x})$ の分散 $\sigma^2[I_v]$ は次

式によって近似的に与えられる．

$$\sigma^2[I_v] \approx -\left(\frac{1}{|\mathcal{A}|}\right)^2 + \frac{1}{4\pi h^2|\mathcal{L}_v|^2|\mathcal{A}|}\left[|\mathcal{L}_v| + \sum_{\substack{\boldsymbol{x}_i,\boldsymbol{x}_j\in\mathcal{L}_v\\ \boldsymbol{x}_i\neq\boldsymbol{x}_j}} \exp\left(-\frac{(\boldsymbol{x}_i-\boldsymbol{x}_j)^T(\boldsymbol{x}_i-\boldsymbol{x}_j)}{4h^2}\right)\right] \tag{3.8}$$

付録 C.2 において上式の導出手順について述べる．平滑化パラメタ h の増加に伴って二次元ガウス分布の裾が増加するため，境界付近における影響を無視できなくなり，$\mathrm{E}[I_v]$ と $\sigma^2[I_v]$ の推定誤差が増加する（付録 D.2 参照）．

3.2.3 メッセージ転送基準

保持端末 v が不保持端末 u に地点 $\boldsymbol{x}_v$ において遭遇した場合，端末 v は $I_v(\boldsymbol{x}_v)$ を計算し，その計算結果に基づいて転送確率 $p_F := p_F(I_v(\boldsymbol{x}_v))$ を設定する．本論文では，転送確率 p_F を以下のように制御する．

$$p_F(I_v(\boldsymbol{x}_v)) = \begin{cases} p_{\mathrm{HD}}, & I_v(\boldsymbol{x}_v) \geq d_v \text{ の場合} \\ p_{\mathrm{LD}}, & I_v(\boldsymbol{x}_v) < d_v \text{ の場合} \end{cases}$$

ただし，$0 \leq p_{\mathrm{HD}} \leq p_{\mathrm{LD}} \leq 1$ であり，d_v は閾値パラメタである．閾値 d_v は端末密度分布の平均 $\mathrm{E}[I_v]$ と標準偏差 $\sigma[I_v]$ を用いて定義される．提案方式では，$p_{\mathrm{HD}} = p_{\mathrm{LD}}$ とする場合，保持端末が不保持端末にメッセージの複製を遭遇地点に関わらず，同じ確率で転送する確率転送方式と一致する．さらに，$p_{\mathrm{HD}} = p_{\mathrm{LD}} = 1$ の場合，感染型ルーチング方式と一致する．

3.3 性能評価

シミュレーション実験によって端末密度推定に基づく確率転送方式の性能評価を行う．

3.3.1 シミュレーションモデル

二つのシミュレーションモデルを考える．一つは非均質密度モデル（heterogeneous density model）であり，もう一つは均質密度モデル（homogeneous density model）である．非均質密度モデルでは場所毎に端末密度の高低があるのに対して，均質密度モデルではいずれの場所においてもほとんど端末密度が等しい．端末密度推定に基づく確率転送方式は非均質密度モデルにおいて効率的なメッセージ配送を行えるように設計されている．一方で，均質密度モデルは，提案法式の想定範囲外の状況であるため，最悪なケースであると言える．これらの二つ密度モデ

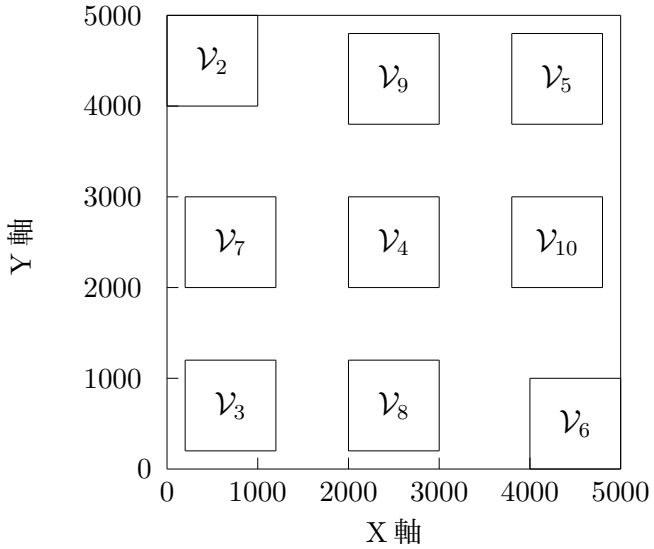


図 3.2: グループ $\mathcal{V}_i$ $(i = 2, 3, \ldots, 10)$ 内の端末が移動する領域

ルを ONE (Opportunistic Networking Environment) シミュレータ [40] を用いて実装した．いずれのシミュレーションモデルにおいても，5000 [m] × 5000 [m] の正方領域 $\mathcal{A}$ 内に 200 台の移動端末が存在しているものとする（図 3.2 を参照）．さらに，各端末の通信半径を $R = 50$ [m] とする．また，バッファ容量は十分に大きいものとし，メッセージの損失は起こらないものとする．

非均質密度モデルでは，ネットワーク内の端末を 10 個のグループ $\mathcal{V}_i$ $(i = 1, 2, \ldots, 10)$ に分割し，$|\mathcal{V}_i| = 30$ $(i = 1, 2, \ldots, 6)$, $|\mathcal{V}_i| = 5$ $(i = 7, 8, 9, 10)$ とする．グループ $\mathcal{V}_1$ 内の端末はランダムウェイポイントモデル [6] によって領域全体 $\mathcal{A}$ を移動する．残りのグループ $\mathcal{V}_i$ $(i = 2, 3, \ldots, 10)$ の端末は各々異なる中心を持つ 1000 [m] × 1000 [m] の正方領域内をランダムウェイポイントモデルによって移動する．各グループ $\mathcal{V}_i$ $(i = 2, 3, \ldots, 10)$ の端末が移動する領域の中心は (500,4500), (700,700), (2500,2500), (4300,4300), (4500,500), (700,2500), (2500,700), (2500,4300), and (4300,2500) である．各端末が移動する速度 [m/s] は目的地点に到達する度に，[0.8, 1.8] の一様分布に従って選ばれる．

一方で，均質密度モデルでは全ての端末が領域全体 $\mathcal{A}$ を以下のモビリティに従って移動する．各端末が目的地点 $\boldsymbol{x}_T$ に到達する度に，次の目的地点を地点 $\boldsymbol{x}_T$ を中心とした 400 [m] × 400 [m] の正方領域からランダムに一点を選択する．また，各端末の移動速度 [m/s] は [0.8, 1.8] の一様分布に従って選ばれる．

3.3.2 端末密度推定の結果

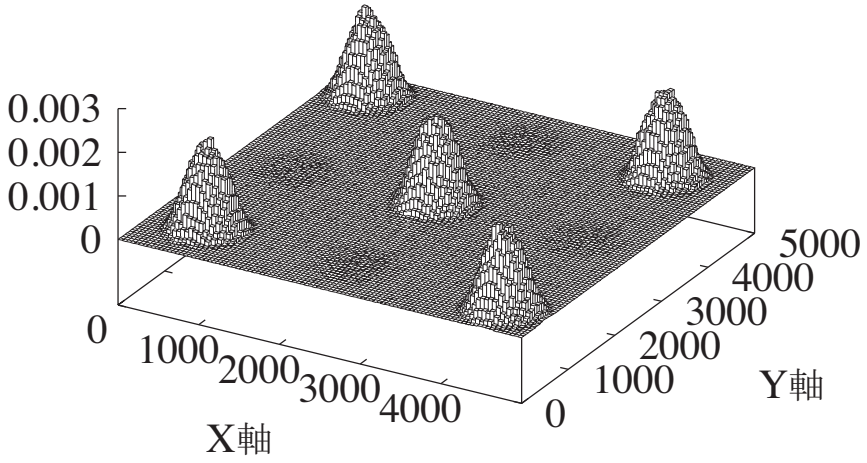
端末密度推定の誤差がルーチング性能に与える影響を明らかにするために，均質密度モデルと非均質密度モデルに対してそれぞれ三つの端末密度推定法によって得られる分布を考える．一つ目は多くの遭遇地点情報（約 527,000 個）を用いたヒストグラム法によって得られる端末密度分布である．この分布は，真の端末密

度分布と近いものとなる．ヒストグラム法では，領域全体 $\mathcal{A}$ を 100×100 の小領域に分割し，各小領域における遭遇回数をカウントする．残りの二つの密度推定法は，3.2 節で述べたパラメトリック密度推定法とガウスカーネル密度推定法である．パラメトリック密度推定法では $I_v(\boldsymbol{x})$ を求めるために端末 $v \in \mathcal{V}_1$ が 100,000 秒の間に収集した 40,000 個の遭遇地点情報，つまり $|\mathcal{L}_v| = 40,000$ の情報を用いて推定を行う．非均質密度モデルと均質密度モデルに対して混合数 K をそれぞれ $K = 5$ と $K = 1$ とした．このような設定によって，ヒストグラム法による真の分布と異なった端末密度分布を得ることができる．また，ガウスカーネル密度推定法では，平滑化パラメタを $h = 200$ とした．さらに，推定に用いる遭遇地点情報は，ある端末 $v \in \mathcal{V}_1$ が付録 D.1 の更新手順に従って収集し，端末 v 以外の 199 台の最新の遭遇地点情報のみを利用するものとする．

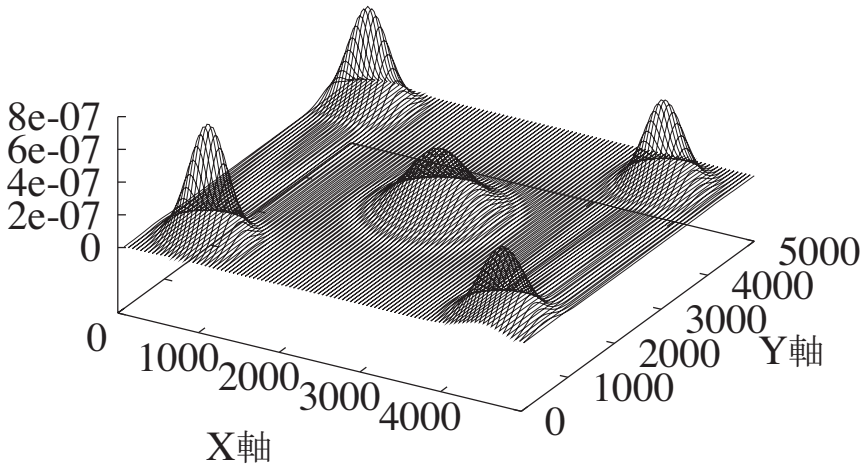
図 3.3 と図 3.4 はそれぞれ非均質密度モデルと均質密度モデルにおける端末密度推定の結果を示している．非均質密度モデルにおいては，図 3.3(b) からパラメトリック密度推定法では中央の山がヒストグラム法よりも滑らかであることが分かる．これは，$K = 5$ であるため，全てのガウス要素が端末密度の高い領域に位置し，中央の山が周りに存在する端末密度の低い領域から最も影響を受けるからである．ガウスカーネル密度推定法では，複数の歪な山が存在している．一方で，均質密度モデルでは，$I_v(\boldsymbol{x})$ は三つの端末密度推定法によって完全に異なった結果となっている．パラメトリック密度推定法では $K = 1$ であるため，単峰で滑らかな形状をしている．一方で，ガウスカーネル密度推定法では $I_v(\boldsymbol{x})$ は複数の峰があり，不規則な形状をしている．

端末密度推定の不正確さが高密度領域の判定に大きく影響を与える．これを明らかにするために，二つの閾値の設定を考える．一つは $d_v = \mathrm{E}[I_v] + \sigma[I_v]$ であり，もう一つは $d_v = \mathrm{E}[I_v] + \sigma[I_v]/10$ である．図 3.5 と図 3.6 はそれぞれ高い閾値 $d_v = \mathrm{E}[I_v] + \sigma[I_v]$ の場合と低い閾値 $d_v = \mathrm{E}[I_v] + \sigma[I_v]/10$ の場合の非均質密度モデルにおける高密度領域を示している．高い閾値に設定する場合，パラメトリック密度推定法の方が正確な推定を行えているが，ガウスカーネル密度推定法を用いたとしてもある程度正確に高密度領域を推定できている．この推定精度の違いは，推定に用いられる情報量の違いに起因している．低い閾値に設定する場合，パラメトリック密度推定法では正確に推定を行えているが，一方で，ガウスカーネル密度推定法では低密度領域の一部を高密度領域と見積もっており，端末密度推定の精度が悪い．

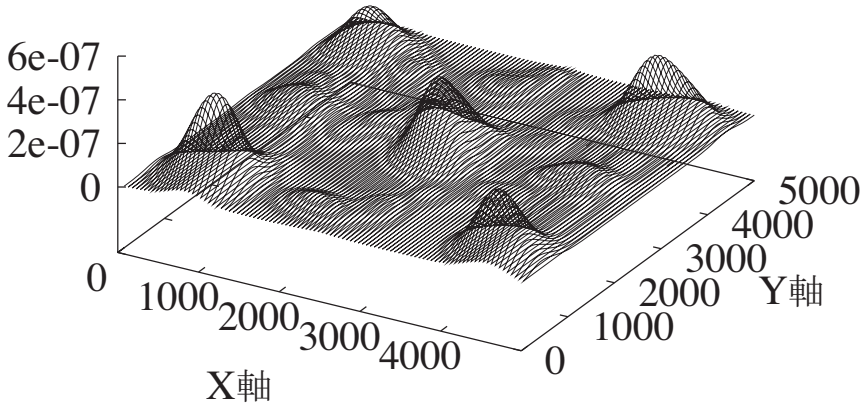
図 3.7 と図 3.8 はそれぞれ閾値を高く設定した場合と低く設定した場合の均質密度モデルにおける高密度領域を示している．明らかに，均質密度モデルでは，パラメトリック密度推定法とガウスカーネル密度推定法のどちらの密度推定法を用いたとしても正しく密度推定を行えていないことが分かる．



(a) 真の分布 (ヒストグラム法)

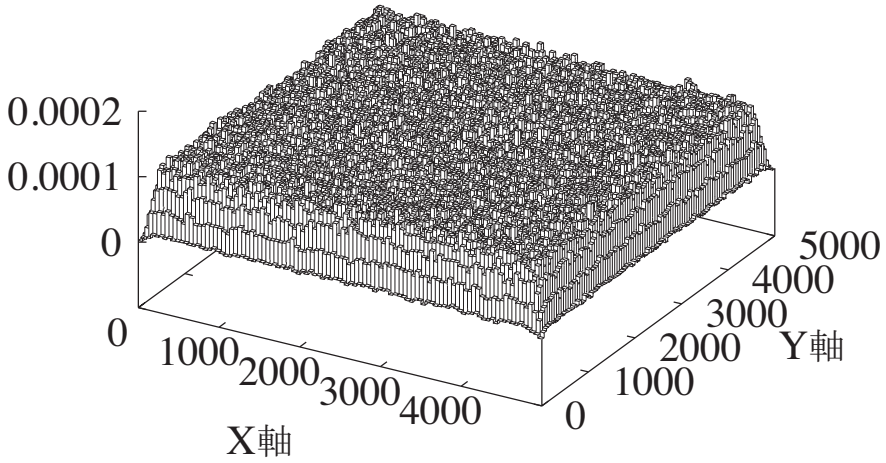


(b) パラメトリック密度推定法

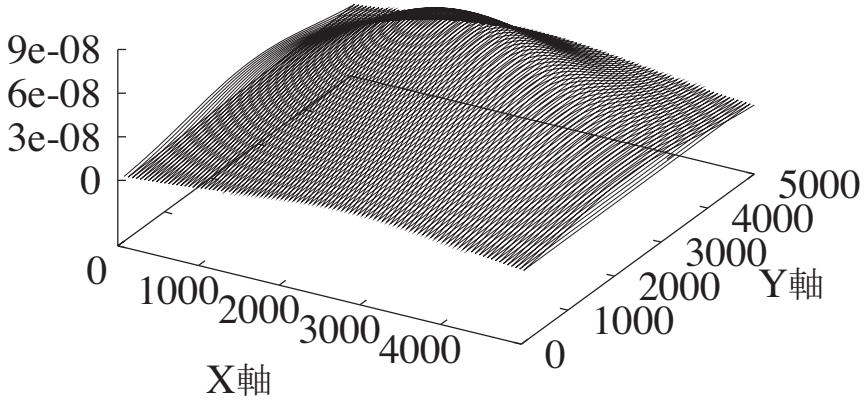


(c) ガウスカーネル密度推定法

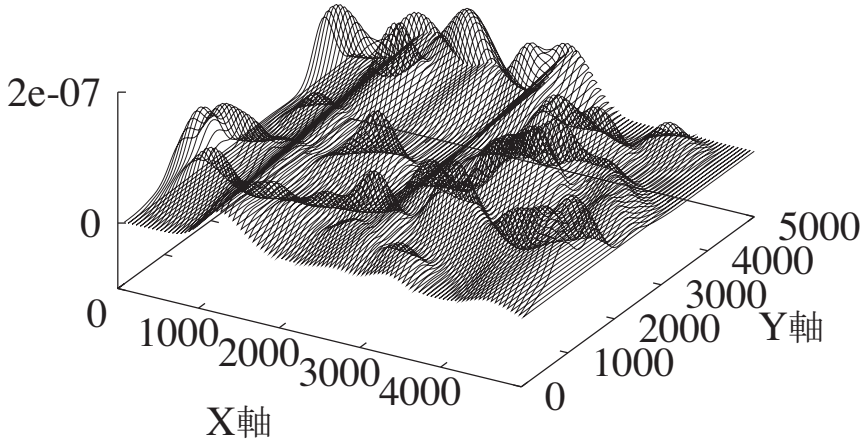
図 3.3: 密度モデルにおける端末密度分布の推定結果



(a) 真の分布（ヒストグラム法）

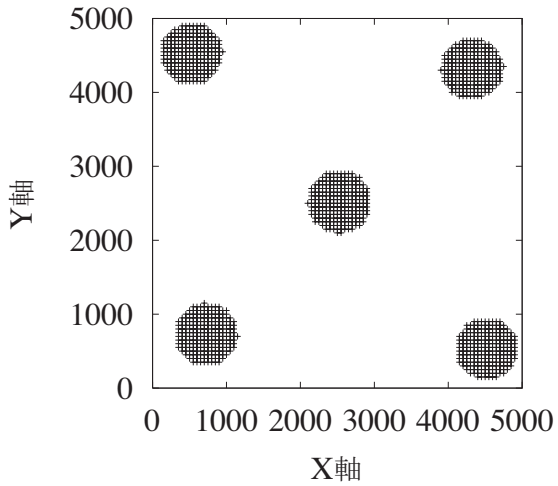


(b) パラメトリック密度推定法

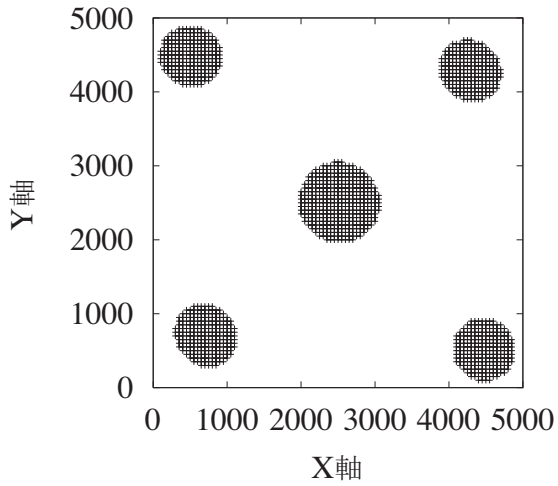


(c) ガウスカーネル密度推定法

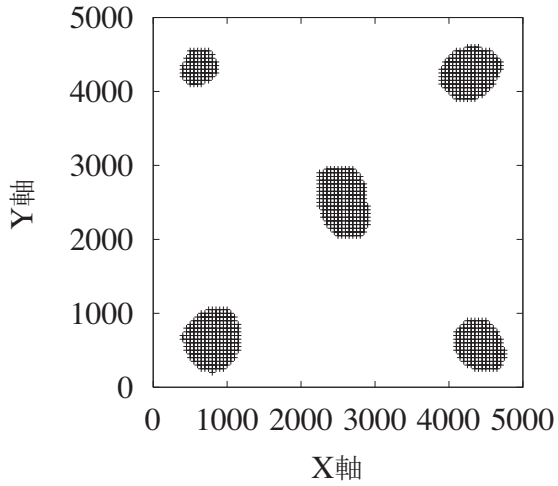
図 3.4: 均質密度モデルにおける端末密度分布の推定結果



(a) 真の分布 (ヒストグラム法)

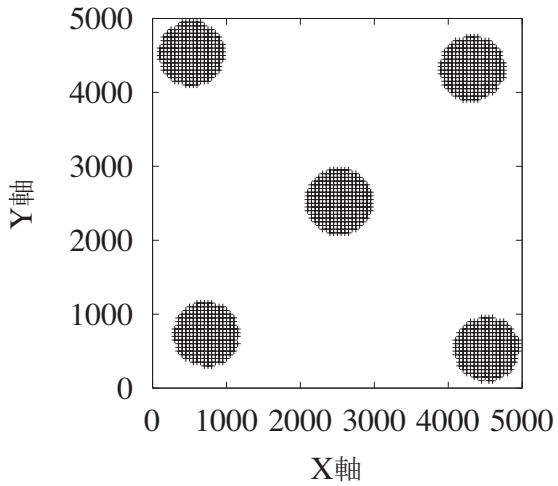


(b) パラメトリック密度推定法

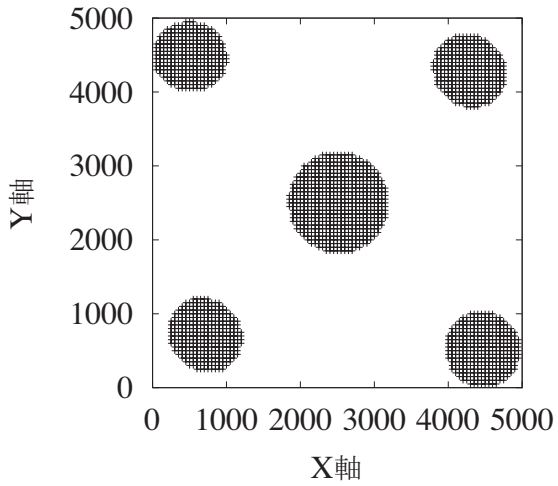


(c) ガウスカーネル密度推定法

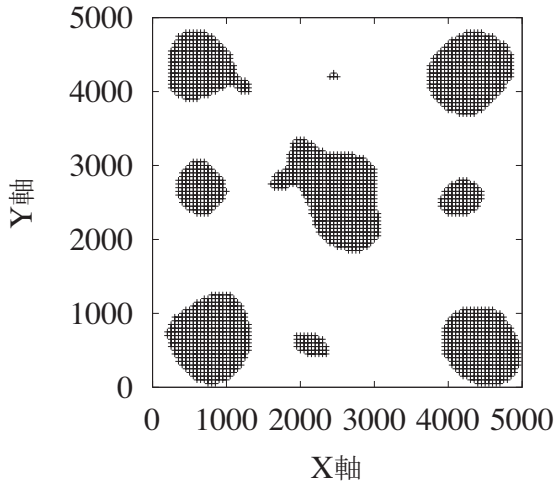
図 3.5: 非均質密度モデルにおける高密度領域 ($d_v = \mathrm{E}[I_v] + \sigma[I_v]$ の場合)



(a) 真の分布（ヒストグラム法）

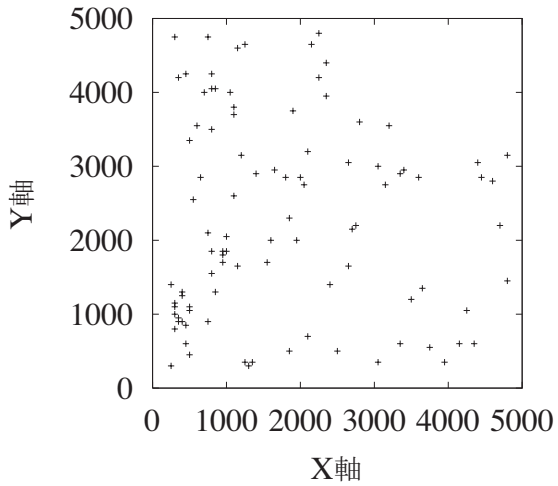


(b) パラメトリック密度推定法

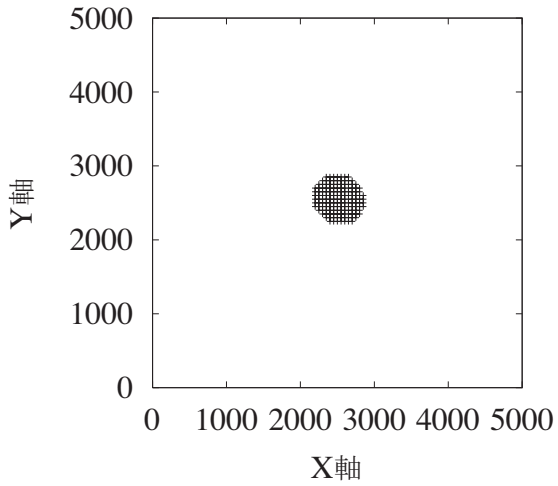


(c) ガウスカーネル密度推定法

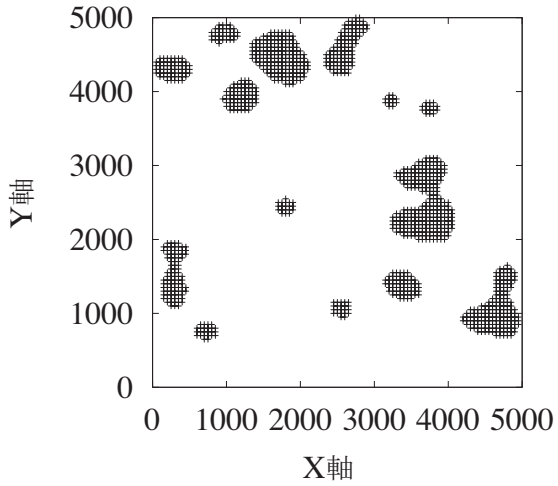
図 3.6: 非均質密度モデルにおける高密度領域 ($d_v = \mathrm{E}[I_v] + \sigma[I_v]/10$ の場合)



(a) 真の分布 (ヒストグラム法)

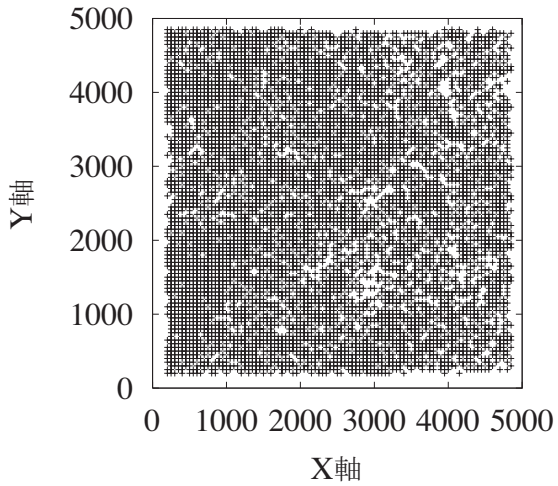


(b) パラメトリック密度推定法

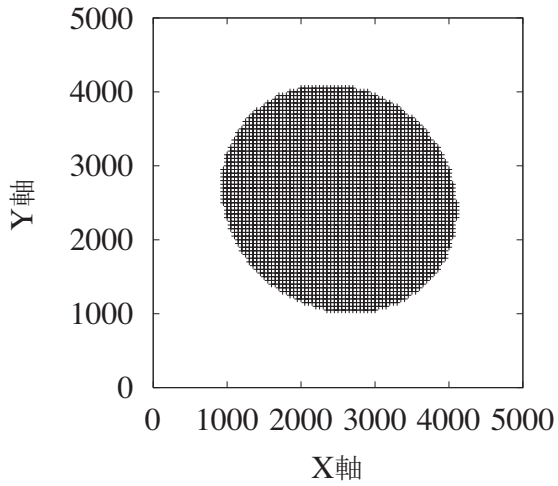


(c) ガウスカーネル密度推定法

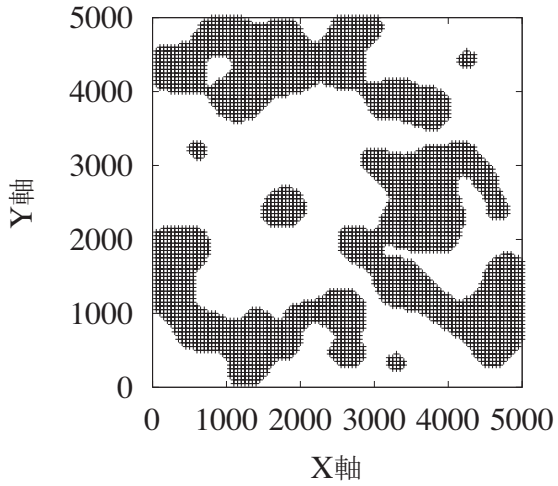
図 3.7: 均質密度モデルにおける高密度領域 ($d_v = \mathrm{E}[I_v] + \sigma[I_v]$ の場合)



(a) 真の分布 (ヒストグラム法)



(b) パラメトリック密度推定法



(c) ガウスカーネル密度推定法

図 3.8: 均質密度モデルにおける高密度領域 ($d_v = \mathrm{E}[I_v] + \sigma[I_v]/10$ の場合)

3.3.3 端末密度を考慮した確率転送方式の性能評価

本節では，端末密度を考慮した確率転送方式の性能を評価する．回復手法としては VACCINE 回復手法を採用する [28, 50]．VACCINE 回復手法では，メッセージの配送完了後に感染型ルーチング方式によってアンチパケットをブロードキャストし，不要なメッセージの複製を削除する．

各シミュレーション実験において遭遇地点情報を収集するために，ウォームアップ期間を設ける（表 3.1 を参照）．ヒストグラム法では，いずれの密度モデルの場合においてもウォームアップ期間中に約 527,000 回の遭遇が発生する．一方で，パラメトリック密度推定法とガウスカーネル密度推定法では，非均質密度モデルの場合約 58,800 回，均質密度モデルの場合 10,300 回の遭遇が起こる．パラメトリック密度推定法で用いられる $|\mathcal{L}_v|$ の平均個数は，非均質密度モデルの場合約 54,300 個であり，均質密度モデルの場合約 8800 個である．

ウォームアップ期間の終了後，30 秒毎に送信元端末の候補集合 $\mathcal{S}$ からランダムに一つの端末を選び，その端末が一つのメッセージを発生させる．各メッセージの受信先端末は，受信先端末の候補集合 $\mathcal{D}$ の中からランダムに一つの端末を選ぶ．いずれのシナリオにおいても低密度領域における確率 $p_{\mathrm{LD}} = 1$ とした．送信元候補集合 $\mathcal{S}$ と受信先候補集合 $\mathcal{D}$ が異なる複数のシミュレーションシナリオを考える．具体的な $\mathcal{S}, \mathcal{D}$ については後述する．各シミュレーションシナリオにおいて 1,000 個のメッセージを発生させた．

性能評価指標として平均配送遅延と平均複製転送回数を用いる．メッセージの配送遅延 T_D はメッセージの発生から受信先端末の配送されるまでに要する時間である．また，複製転送回数 N_D はメッセージの複製がネットワークから消滅するまでにメッセージの複製が転送された回数を表す．一般に，$\mathrm{E}[T_D]$ と $\mathrm{E}[N_D]$ の間にトレードオフの関係がある．性能比較のため，端末密度に依らず一定の確率 $p_{\mathrm{HD}} = p_{\mathrm{LD}} = q$ でメッセージの複製を転送する確率転送方式の結果についても示す．

図 3.9 と図 3.10 は非均質密度モデルにおける平均複製転送回数 $\mathrm{E}[N_D]$ に対する平均配送遅延 $\mathrm{E}[T_D]$ の変化を示している．図 3.9 では $\mathcal{S} = \mathcal{V}_1, \mathcal{D} = \cup_{i=7}^{10}\mathcal{V}_i$ とし，図 3.10 では $\mathcal{S} = \mathcal{V}_1, \mathcal{D} = \cup_{i=2}^{6}\mathcal{V}_i$ とする．いずれの場合においても，端末密度を考慮した転送方式の性能は，確率転送方式によって上から抑えられている．この結果は低密度領域における転送確率 p_F の違いによって生じる．端末密度を考慮した転送方式では $p_{\mathrm{LD}} = 1\ (\geq p_{\mathrm{HD}})$ であるのに対して，確率転送方式では $p_{\mathrm{HD}} = p_{\mathrm{LD}} = q$ である．さらに，受信先端末が低密度領域に位置する場合，端末密度を考慮した転送方式が良い性能を示している．すなわち，わずかに $\mathrm{E}[T_D]$ を増加させることで $\mathrm{E}[N_D]$ を大きく減少できる．

また，高密度領域をある程度推定できてさえいれば，密度推定の誤差がほとんど影響を与えないことが分かる．ルーチング性能の違いは，主に高密度と推定された総面積 $|\mathcal{A}_{\mathrm{high}}|$ の違いによって生じている．端末密度を考慮した転送方式の転送確率は高密度領域において 1 よりも小さく設定されるため，面積 $|\mathcal{A}_{\mathrm{high}}|$ の増加に伴って $\mathrm{E}[N_D]$ が減少する．さらに，$|\mathcal{A}_{\mathrm{high}}|$ の増加に伴って $\mathrm{E}[N_D]$ の制御範囲が広がる．低密度領域を高密度領域として誤って推定する場合，すなわち，閾値

表 3.1: ウォームアップ期間の長さ

非均質密度モデル		
	ヒストグラム法	パラメトリック推定法, ガウスカーネル推定法
期間 (秒)	900,000	100,000

均質密度モデル		
	ヒストグラム法	パラメトリック推定法, ガウスカーネル推定法
期間 (秒)	5,100,000	100,000

を $d_v = \mathrm{E}[I_v] + \sigma[I_v]/10$ と設定したガウスカーネル密度推定法の場合には性能がわずかに悪化している．

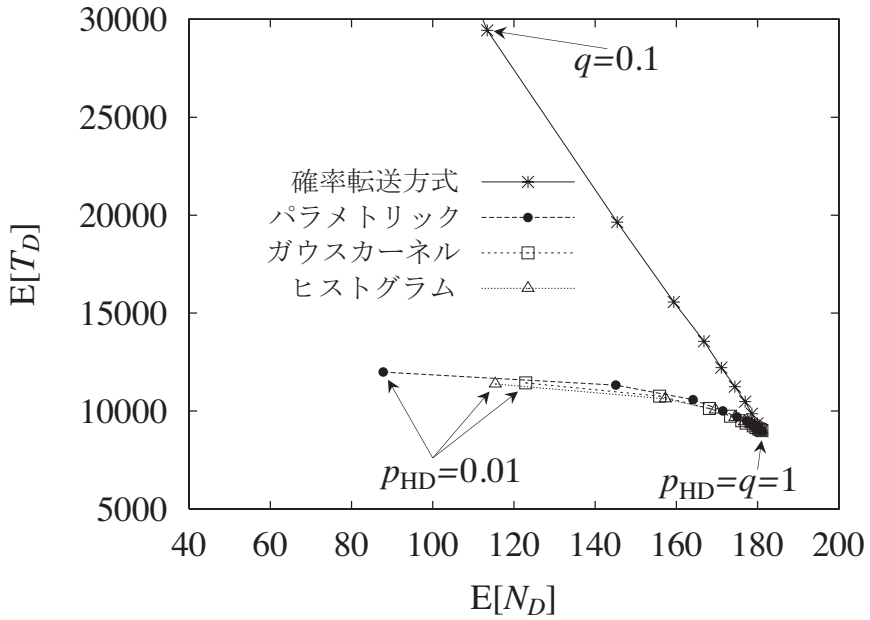
図 3.11 は均質密度モデルにおける $\mathrm{E}[N_D]$ に対する $\mathrm{E}[T_D]$ の変化を示している．端末密度を考慮した確率転送方式の全ての性能曲線が確率転送方式の曲線と重なっている．端末密度が領域全体にわたって同質である場合には，端末密度を考慮した転送確率制御は役に立たない．さらに，このような場合には端末密度を考慮した確率転送方式は確率転送方式とほとんど同等の性能であることが分かる．

3.4 結言

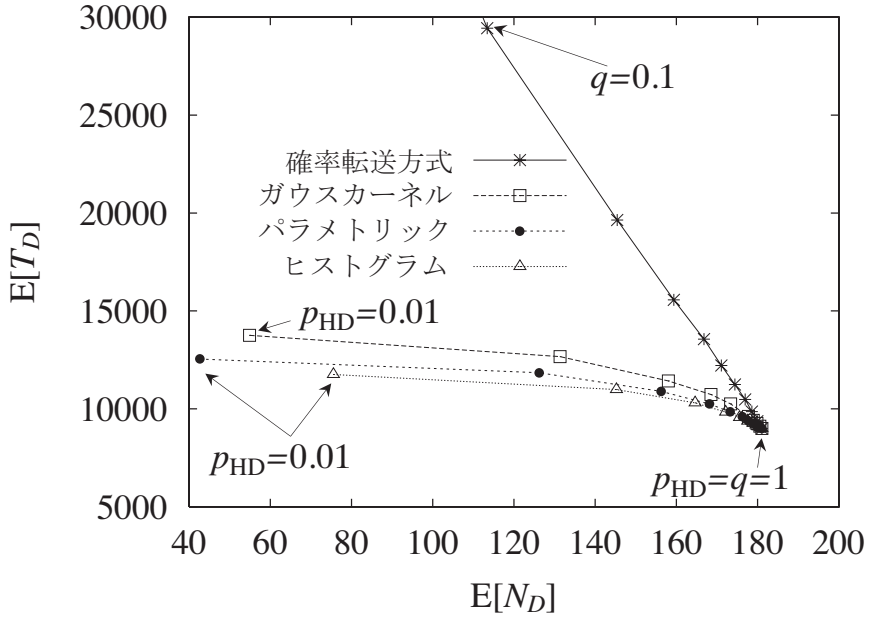
本章では，端末密度推定に基づく確率的転送方式を提案した．端末密度分布を推定するために，パラメトリック密度推定法とガウスカーネル密度推定法の二つの推定法を用いた．高密度領域を特定できてさえいれば，端末密度の推定誤差に対して提案方式がロバストであることを示した．ガウスカーネル密度推定法は推定に用いた遭遇地点情報の数が少なく端末密度の推定精度が悪いにも関わらず，パラメトリック密度推定法を用いたルーチングと同程度の効率性を達成した．さらに，受信先端末が低密度領域に位置している場合には，効率的なメッセージ配送を行えることを示した．受信先端末が高密度領域に位置する場合においても確率的転送方式よりもわずかに効率性が改善することを示した．また，提案方式にとって最悪なシナリオとして領域全体にわたって端末密度が均質な状況を想定し，このような状況において提案方式が確率的転送方式とエネルギー消費に関して同程度の効率性を達成できることを示した．これらの結果から，遭遇場所の特徴量を考慮することによって効率的なメッセージ配送を行えることが分かった．

今後の課題としては推定された端末密度分布のより有効な利用方法を考える必要がある．本論文では，最も簡単な方法である一つの閾値を用いて転送確率 p_F の制御を行った．一般に，p_F は遭遇地点 $\boldsymbol{x}$ における端末密度 $I_v(\boldsymbol{x})$ の連続関数として定義できる．また，本章では遭遇地点情報の収集や保持に消費されるネットワーク資源量については無視して議論してきたが，実際にはネットワーク資源を消費する．ガウスカーネル密度推定法では少ない情報量で端末密度を推定できるため，ネットワーク資源消費量は少ないが，達成できるメッセージ配送の効率性は低い．一方で，ヒストグラム法やパラメトリック密度推定法では推定に必要な

情報量が多いため，ネットワーク資源消費量は多いが，達成できるメッセージ配送の効率性は低い．よって，遭遇地点情報を維持に必要な資源消費量とメッセージ配送の効率性の間にはトレードオフの関係がある．このトレードオフの関係を踏まえた上で評価をする必要がある．さらに，端末密度の時間的変動が大きな状況における提案方式の性能評価についても行う必要がある．

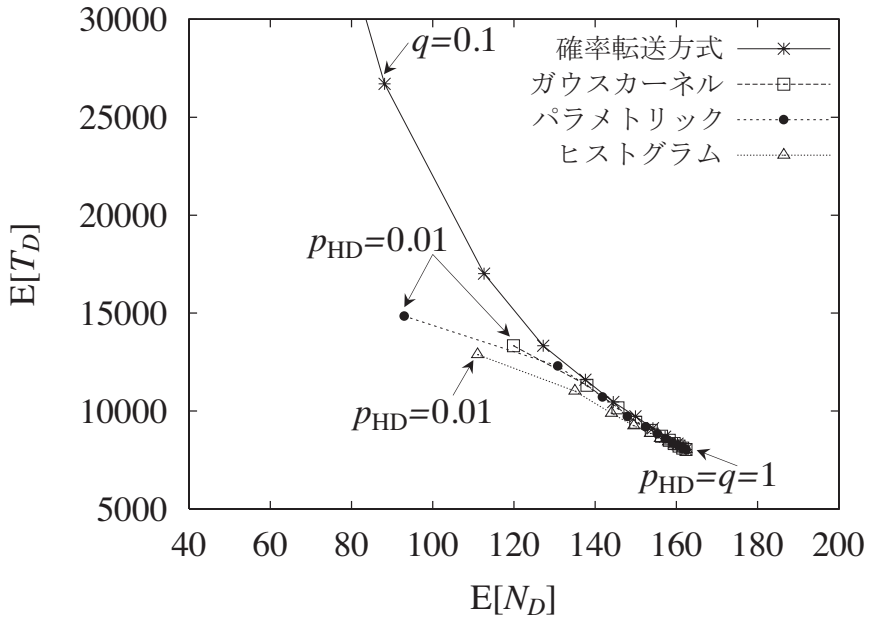


(a) $d_v = \mathrm{E}[I_v] + \sigma[I_v]$ の場合

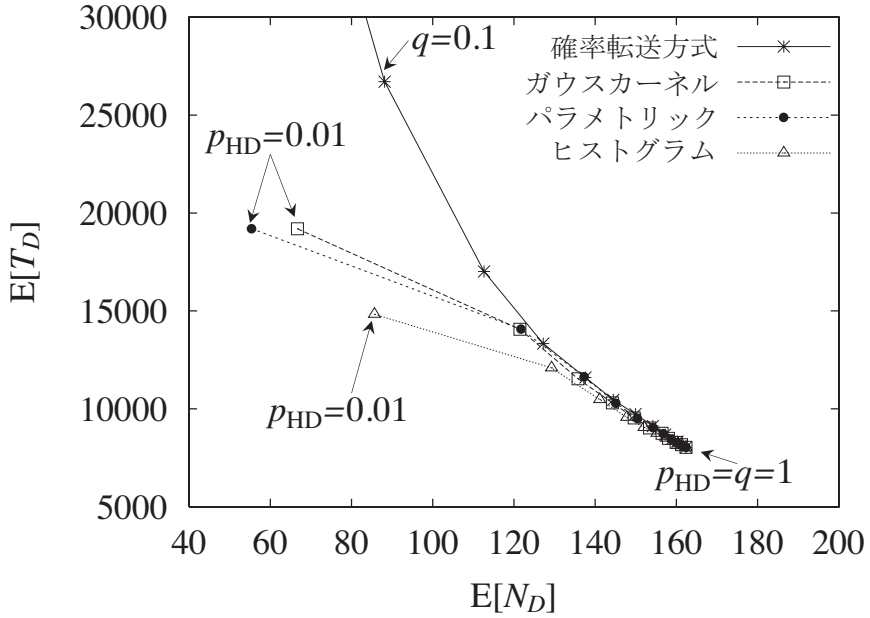


(b) $d_v = \mathrm{E}[I_v] + \sigma[I_v]/10$ の場合

図 3.9: 非均質密度モデルにおける平均複製転送回数 $\mathrm{E}[N_D]$ に対する平均配送遅延 $\mathrm{E}[T_D]$ の変化（受信先端末が低密度領域から選択される場合）

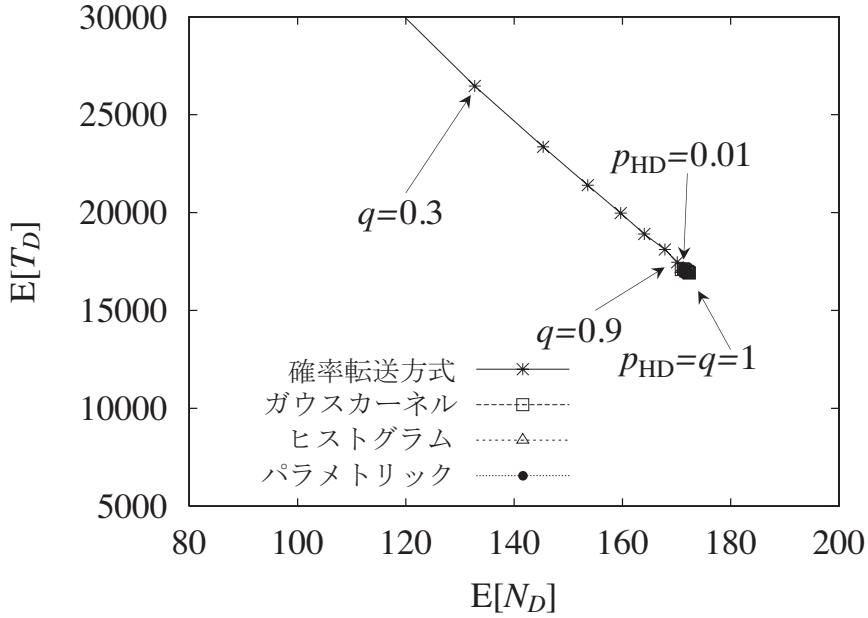


(a) $d_v = \mathrm{E}[I_v] + \sigma[I_v]$ の場合

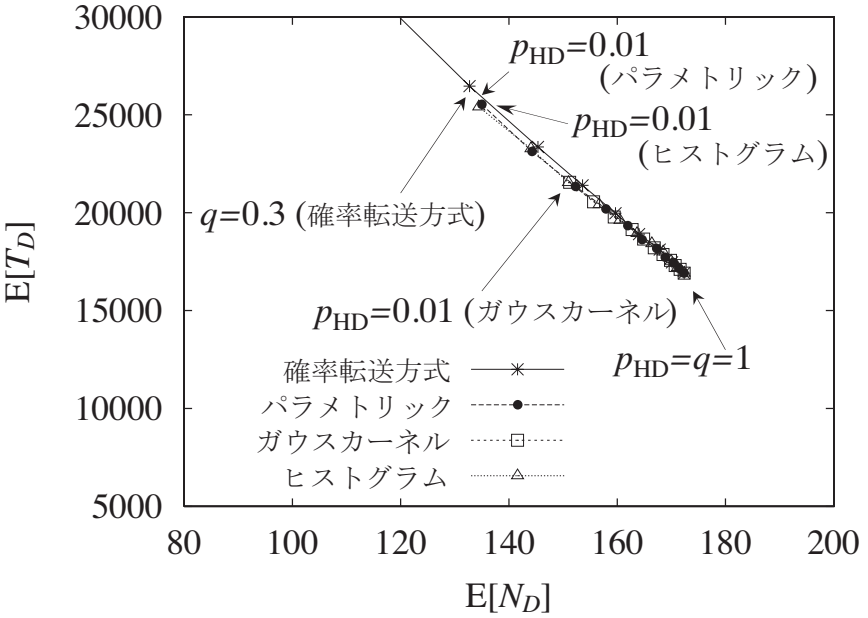


(b) $d_v = \mathrm{E}[I_v] + \sigma[I_v]/10$ の場合

図 3.10: 非均質密度モデルにおける平均複製転送回数 $\mathrm{E}[N_D]$ に対する平均配送遅延 $\mathrm{E}[T_D]$ の変化（受信先端末が高密度領域から選択される場合）



(a) $d_v = \mathrm{E}[I_v] + \sigma[I_v]$ の場合



(b) $d_v = \mathrm{E}[I_v] + \sigma[I_v]/10$ の場合

図 3.11: 均質密度モデルにおける平均複製転送回数 $\mathrm{E}[N_D]$ に対する平均配送遅延 $\mathrm{E}[T_D]$ の変化

第4章

端末密度を考慮した効用型ルーチング方式の提案とその評価

4.1 緒言

本章では，遭遇端末の特徴量である効用値と遭遇場所の特徴量である端末密度の両方を用いてメッセージの複製の転送を制御するルーチング方式を提案する．1.2.4 節で述べたように，効用値は受信先端末と遭遇する潜在的な可能性を意味する．さらに，1.2 節で述べたように，既存のルーチング方式では遭遇端末の特徴量を単独で用いてメッセージの複製の転送を制御しているものが大半である．しかしながら，遭遇端末と遭遇場所の特徴量を併用することによって効率的なメッセージ配送の実現が期待される．

提案方式では，効用型ルーチング方式の性能が端末間の遭遇率に大きく依存することに着目する．メッセージ保持端末が別の端末と頻繁に遭遇する場合，メッセージの複製を効用値の高い端末にのみ転送すれば良い．この場合，効用値の高い端末との遭遇時間間隔が短いため，効用値の低い端末に複製を転送しなくとも遅延性能にほとんど影響を与えない．一方で，メッセージを持つ端末が別の端末と遭遇する機会がほとんどない場合，メッセージの転送能力を維持するために積極的にメッセージの複製を転送した方が良い．一般に，端末間の遭遇率と端末密度には強い関連性があるため，遭遇地点の端末密度を考慮することによって効用型ルーチング方式の性能を改善できる可能性が高い．

そこで，既存の効用型ルーチング方式と 3 章で述べた端末密度を考慮した確率転送方式を組み合わせることを考える．メッセージを持つ端末が効用値の高い端末と遭遇した場合，前者は後者にメッセージの複製を必ず転送する．これは既存の効用型ルーチング方式と同じ挙動である．一方で，メッセージを持つ端末が効用値の低い端末と遭遇した場合，前者は後者にメッセージの複製を転送するか否かを遭遇地点の端末密度に基づいて決定する．具体的には，低密度領域において遭遇が起こった場合，メッセージの複製を高い確率で転送する．このように転送確率を制御することによってメッセージの転送能力が低密度領域において維持される．これは 3 章で述べた端末密度を考慮した確率転送方式の考え方に基づいている．

4.2 システムモデル

$\mathcal{V}$ をネットワーク内の端末の集合と定義する．閉領域 $\mathcal{A} \subset \mathbb{R}^2$ 内を全ての端末が移動するものとし，各端末の通信半径 R 内に別の端末が入った際には互いに通信を行えるものとする．各端末はメッセージを蓄積するための無限大のバッファを保持しているものとする．さらに，GPS (Global Positioning System) などの位置測定装置を備えており，自身の位置を把握できるものとする．また，各メッセージ m の送信元端末と受信先端末をそれぞれ $src(m) \in \mathcal{V}$ と $dst(m) \in \mathcal{V}$ と表す．本章では，オリジナルのメッセージとその複製を区別する必要がない場合にはこれらをメッセージと呼ぶことにする．

4.3 効用型ルーチング方式

本節では，1 章において述べた効用型ルーチング方式の代表例である PRoPHET ルーチング方式 [46] と Delegation 転送方式 [17] について詳細に述べる．

4.3.1 メッセージ転送の手順

$U_v(d) \in [0,1]$ を端末 $d \in \mathcal{V}$ に対する端末 $v \in \mathcal{V}$ の効用値と定義する．メッセージ m を持つ端末 v がメッセージ m を持たない端末 $w \in \mathcal{V} \setminus \{src(m), dst(m)\}$ に遭遇した状況を考える．効用型ルーチング方式では，$U_w(dst(m))$ が閾値 $\theta_v(m)$ を上回る場合，端末 v は端末 w にメッセージ m の複製を転送する．閾値 $\theta_v(m)$ はメッセージ m に依存し，端末 v によって決定される．

PRoPHET ルーチング方式では，メッセージ m と端末 v に対して，$\theta_v(m)$ は $\theta_v(m) \triangleq U_v(dst(m))$ と定義される．したがって，メッセージを持つ端末 v がその端末自身の効用値よりも大きな効用値を持つ端末と遭遇する度にメッセージの複製を転送する．一方で，Delegation 転送方式では，$\theta_v(m)$ は $\theta_v(m) \triangleq \tau_v(m)$ と定義する．ただし，$\tau_v(m)$ は端末 v が遭遇した端末の中で最大の効用値を表す．ここで，端末 v がメッセージ m を生成する場合を考える．最初に，$\tau_v(m)$ を $U_v(dst(m))$ に設定する．端末 v が端末 w に遭遇し，$\tau_v(m) < U_w(dst(m))$ の場合，$\tau_v(m)$ を $\tau_v(m) = U_w(dst(m))$ に更新し，$\tau_w(m)$ を $U_w(dst(m))$ に設定する．その後，端末 v と端末 w はこの手順を繰り返す．したがって，Delegation 転送方式は効用値の高い端末にのみメッセージの複製を転送することを狙っている．

4.3.2 効用値の計算手順

PRoPHET ルーチング方式と Delegation 転送方式のどちらの方式においても，効用値の計算法が必要となる．本論文では，PRoPHET ルーチング方式で提案されている計算手順 [46] を用いる．まず，最初に端末 $v \in \mathcal{V}$ の効用値 $U_v(k)$ $(k \in \mathcal{V} \setminus \{v\})$

を 0 に設定する．端末 v が端末 w に遭遇した際に，端末 v は次式に従って $U_v(k)$ を更新する．

$$U_v(k) := U_v(k) \cdot \gamma^t, \quad k \in \mathcal{V} \setminus \{v\}$$

ただし，$\gamma \in [0,1]$ であり，t は端末 v が別の端末と最後に遭遇した時点からの経過時間である．さらに，端末 v は $U_v(w)$ を次式によって更新を行う．

$$U_v(w) := U_v(w) + (1 - U_v(w)) \cdot P_{\mathrm{init}}$$

ただし，$P_{\mathrm{init}} \in [0,1]$ である．この更新によって端末 w と遭遇すればするほど $U_v(w)$ は大きくなるため，直接遭遇しやすい端末 w の $U_v(w)$ は大きくなる．同様にして端末 w も $U_w(k)$ $(k \in \mathcal{V} \setminus \{w\})$ を更新する．その後，端末 v と端末 w はそれぞれが保持する効用値の情報を交換する．すなわち，端末 v は端末 w に $U_v(k)$ $(k \in \mathcal{V} \setminus \{v\})$ を送信し，同様に端末 w も端末 v に情報を送信する．効用値情報の受信後，端末 v は $U_v(k)$ $(k \in \mathcal{V} \setminus \{v, w\})$ を次式によって更新する．

$$U_v(k) := U_v(k) + (1 - U_v(k)) \cdot U_v(w) \cdot U_w(k) \cdot \beta$$

ただし，$\beta \in [0,1]$ である．この更新によって，端末 v が端末 k と直接遭遇することがなかったとしても，端末 k に対する効用値の高い端末 w に遭遇すると，端末 k に対する端末 v の効用値 $U_v(k)$ が大きくなる．よって，$U_v(k)$ は直接遭遇するしやすさに加えて，間接的なメッセージ配送のしやすさについても表していることが分かる．同様にして，端末 w も $U_w(k)$ $(k \in \mathcal{V} \setminus \{v, w\})$ を更新する．

4.4 端末密度を考慮した効用型ルーチング方式

本節では，提案する端末密度を考慮した効用型ルーチング方式について述べる．提案方式では，メッセージの転送確率は遭遇端末の効用値と遭遇地点の端末密度に基づいて適応的に制御される．以降では，各端末 v が地点 $\boldsymbol{x} \in \mathcal{A}$ における端末密度 $I_v(\boldsymbol{x})$ を 3.2.2 節で述べた端末密度推定法によって推定するものとする．

4.4.1 提案方式のメッセージ転送手順

提案方式では，3 章で述べた端末密度を考慮した確率転送方式と既存の効用型ルーチング方式の転送規則を組み合わせる．まず，提案方式では，効用値の高い端末と低い端末を閾値 $\theta_v(m)$ を用いて区別する．さらに，端末密度の高い領域と低い領域を各端末 v が独立して設定する閾値 d_v を用いて区別する（3.2.3 節を参照）．もし，$\boldsymbol{x}$ における端末密度 $I_v(\boldsymbol{x})$ が d_v よりも大きい場合，端末 v は地点 $\boldsymbol{x}$ における端末密度を高いものとみなす．一般に p_F は遭遇地点 $\boldsymbol{x}$ における $I_v(\boldsymbol{x})$ の連続関数として定義することできるのだが，3 章で示したように一つの閾値 d_v を用いて p_F を制御する単純な方法であったとしても効率的なメッセージ配送に役立つ．

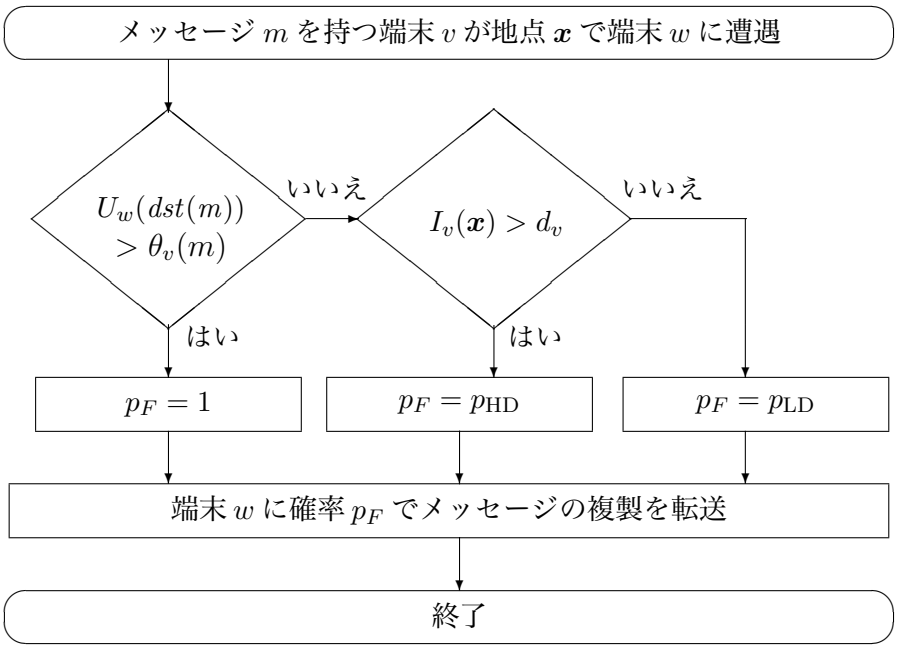


図 4.1: 端末密度を考慮した効用型ルーチング方式の手順

図 4.1 は提案方式の手順を示している．ただし，p_{LD} と p_{HD} （$p_{\mathrm{LD}} \geq p_{\mathrm{HD}}$）はそれぞれ低密度領域と高密度領域における効用値の低い端末への転送確率である．p_{LD} の増加に伴って低密度領域において効用値の低い端末がメッセージを受信する可能性が高まる．一方で，p_{HD} の増加に伴って高密度領域において効用値の低い端末がメッセージを受信する可能性が高まる．p_{LD} を大きく設定することによって効用値の高い端末だけでなく低い端末にもメッセージの複製が転送されるため，低密度領域においてメッセージの転送能力を維持することができる．また，p_{HD} を十分小さく設定することによって，高密度領域において効用値の高い端末のみがメッセージの複製を受信できる．

PRoPHET 方式の中継端末選択規則を用いた提案方式 ($\theta_v(m) \triangleq U_v(dst(m))$) を密度型 PRoPHET 方式と呼ぶ．一方で，Delegation 転送方式の中継端末選択規則を用いた提案方式（$\theta_v(m) \triangleq \tau_v(m)$）を密度型 Delegation 方式と呼ぶ．また，$p_{\mathrm{HD}} = p_{\mathrm{LD}} = 0$ の場合の密度型 PRoPHET 方式と密度型 Delegation 方式はオリジナルの PRoPHET 方式と Delegation 転送方式と一致する．$p_{\mathrm{HD}} = p_{\mathrm{LD}} = 1$ の場合，密度型 PRoPHET 方式と密度型 Delegation 方式は感染型ルーチング方式と一致する．さらに，$\theta_v = 1$ の場合，密度型 PRoPHET 方式と密度型 Delegation 方式は 3 章で述べた端末密度を考慮した確率転送方式と一致する．以降では，表記の簡単化のため，3 章で述べたルーチング方式を密度型ルーチング方式と呼ぶことにする．

4.5 性能評価

シミュレーション実験によって密度型 PRoPHET 方式と密度型 Delegation 方式の性能評価を行う．

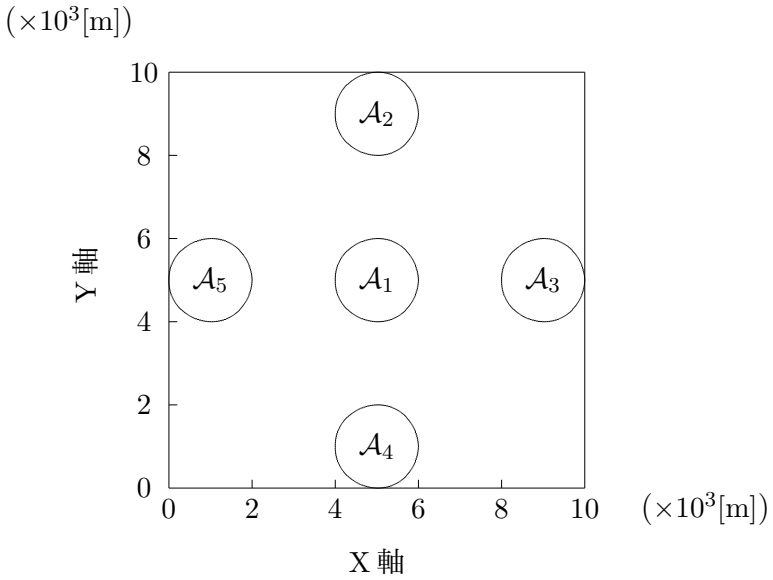


図 4.2: 領域 $\mathcal{A}_i$ $(i = 1, 2, \cdots, 5)$

4.5.1 シミュレーションモデル

10000 [m] × 10000 [m] の正方領域 $\mathcal{A}$ 内に 100 台の端末が存在するものとする (図 4.2 を参照)．領域 $\mathcal{A}$ は半径 1000 [m] の五つの円領域 $\mathcal{A}_i \subset \mathcal{A}$ $(i = 1, 2, \ldots, 5)$ を含む．また，端末集合 $\mathcal{V}$ を互いに素な五つのグループ $\mathcal{V}_i$ $(i = 1, 2, \ldots, 5)$ に分割する．グループ $\mathcal{V}_1$ 内の端末はランダムウェイポイントモデル [6] にしたがって領域 $\mathcal{A}_1$ 内のみを移動する．一方で，残りのグループ $\mathcal{V}_j$ $(j = 2, 3, 4, 5)$ 内の端末は領域 $\mathcal{A}_1$ と 領域 $\mathcal{A}_j$ を以下のモビリティに従って移動する．図 4.3 に示す 2 状態マルコフ連鎖によって目的地点が選択される．グループ $\mathcal{V}_j$ $(j = 2, 3, 4, 5)$ 内の端末は主に領域 $\mathcal{A}_j$ に滞在し，時折領域 $\mathcal{A}_1$ を訪れる．したがって，領域 $\mathcal{A}_j$ $(j = 2, 3, 4, 5)$ を $\mathcal{V}_j$ の拠点と呼ぶことにする．さらに，各メッセージ m に対する送信元端末 $src(m)$ の拠点を送信元拠点，受信先端末 $dst(m)$ の拠点を受信先拠点と呼ぶことにする．

以降では，$|\mathcal{V}_1| = 60$，$|\mathcal{V}_j| = 10$ $(j = 2, 3, 4, 5)$ とする．各端末の移動速度を 1 [m/s] とし，通信半径を $R = 50$ [m] とする．また，各端末のバッファ容量と通信帯域は十分大きいものとし，メッセージの損失は起こらないものとする．各シミュレーション実験において 50 秒毎に送信元候補集合 $\mathcal{S}$ から一つの端末を選択し，その端末がメッセージを発生させる．各メッセージの受信先端末もまた受信先候補端末集合 $\mathcal{D}$ から選択される．表 4.1 に示す $\mathcal{S}$ と $\mathcal{D}$ 異なる六つのシナリオを考える．ただし，$\mathcal{V} = \cup_{i=1}^{5} \mathcal{V}_i$ であり，“$\mathcal{X} \to \mathcal{Y}$” は $\mathcal{S} = \mathcal{X}, \mathcal{D} = \mathcal{Y}$ $(\mathcal{X}, \mathcal{Y} \subseteq \mathcal{V})$ であることを表す．いずれのシナリオにおいても 5000 個のメッセージを発生させ，これらのメッセージは密度型 PRoPHET 方式または密度型 Delegation 方式のいずれかを用いて受信先端末へ配送される．効用値を計算するために，$P_{\mathrm{init}} = 0.7$, $\beta = 0.2$, $\gamma = 0.9$ とした．4.3.2 節で述べた手順に従って効用値を更新するため，各シミュレーション実験には 10000 [s] のウォームアップ期間を設ける．

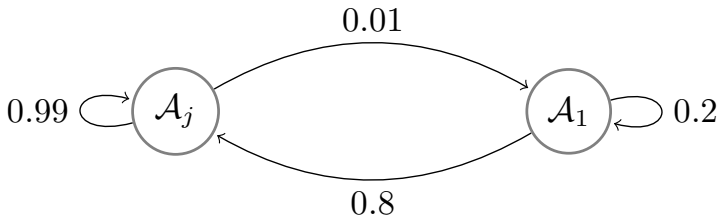


図 4.3: グループ $\mathcal{V}_j$ $(j = 2, 3, 4, 5)$ における目的地点の状態遷移図

表 4.1: シミュレーションシナリオ

シナリオ	送信元候補集合 $\mathcal{S}$	受信先候補集合 $\mathcal{D}$
$\mathcal{V}_1 \to \mathcal{V}_1$	$\mathcal{V}_1$	$\mathcal{V}_1$
$\mathcal{V}_i \to \mathcal{V}_i$	$\mathcal{V}_i$ $(i = 2, 3, 4, 5)$	$\mathcal{V}_i$
$\mathcal{V}_i \to \mathcal{V}_j$	$\mathcal{V}_i$ $(i = 2, 3, 4, 5)$	$\mathcal{V}_j$ $(j = 2, 3, 4, 5,\ j \neq i)$
$\mathcal{V}_1 \to \mathcal{V}_i$	$\mathcal{V}_1$	$\mathcal{V}_i$ $(i = 2, 3, 4, 5)$
$\mathcal{V}_i \to \mathcal{V}_1$	$\mathcal{V}_i$ $(i = 2, 3, 4, 5)$	$\mathcal{V}_1$
$\mathcal{V} \to \mathcal{V}$	$\mathcal{V}$	$\mathcal{V}$

簡単化のため，各端末 v は遭遇地点が高密度領域 $\mathcal{A}$ であるか否かを正確に把握できるものとする．3 章で端末密度推定の精度について議論した際に述べたように，高密度領域をある程度推定出来てさえいれば，多少端末密度推定の誤差があったとしても端末密度を考慮した確率的転送方式は効率的なメッセージ配送を行える．以降では，$\mathcal{A}_1$ を高密度領域，残りの部分 $\mathcal{A} \setminus \mathcal{A}_1$ を低密度領域と呼ぶ．さらに，いずれのルーチング方式においても回復手法として VACCINE 回復手法 [28] を採用する．

4.5.2 性能評価指標

密度型 PRoPHET 方式と密度型 Delegation 方式の平均配送遅延と平均複製転送回数を評価する．メッセージの配送遅延 T_D はメッセージの生成から受信先端末に配送されるまでに要する時間であり，複製転送回数 N_D はオリジナルのメッセージとその複製の全てがネットワーク内から VACCINE 回復手法によって削除されるまでにメッセージの複製を転送した回数である．一般に，$\mathrm{E}[T_D]$ と $\mathrm{E}[N_D]$ の間にはトレードオフの関係がある．提案方式では，効用値の低い端末への転送確率 p_{LD} と p_{HD} の増加に伴って，$\mathrm{E}[T_D]$ は減少するが，$\mathrm{E}[N_D]$ の増加を招く．

4.5.3 確率 p_{LD} の影響

本節では，p_{HD} をゼロに固定し，p_{LD} の増加がどのように提案方式の性能に影響を与えるかを明らかにする．図 4.4 は $\mathrm{E}[N_D]$ に対する平均配送遅延 $\mathrm{E}[T_D]$ の変化

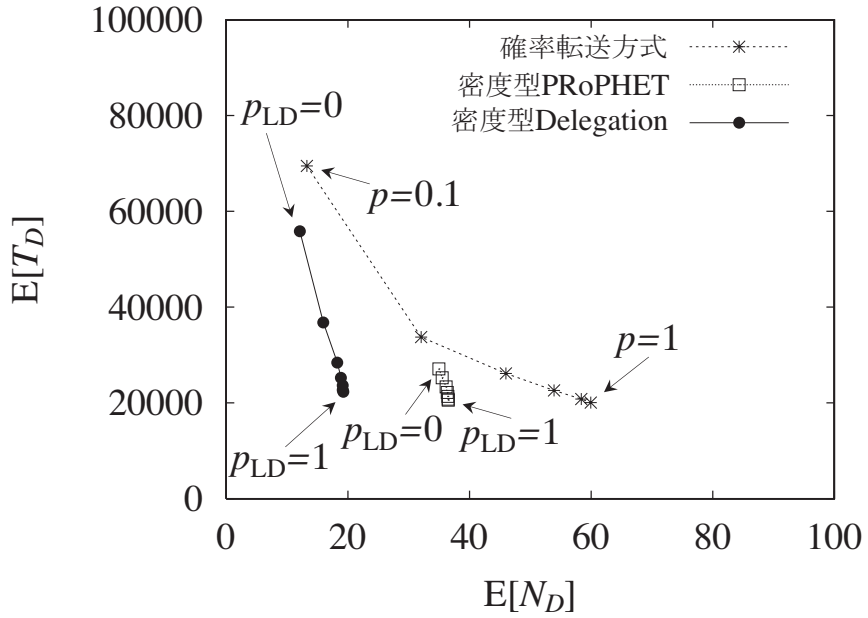
を示している．遭遇地点や遭遇端末に関わらず，同じ確率 p（$0 < p \leq 1$）でメッセージの複製を転送する確率転送方式の結果についても示す．

図 4.4(a), 図 4.4(b), 図 4.4(c) はそれぞれシナリオ $\mathcal{V}_i \to \mathcal{V}_1$, $\mathcal{V}_i \to \mathcal{V}_i$, $\mathcal{V}_i \to \mathcal{V}_j$ の結果を示している．密度型 Delegation 方式ではわずかに $\mathrm{E}[N_D]$ を増加させることで，$\mathrm{E}[T_D]$ が大きく減少する．特に，$p_{\mathrm{LD}} = 1$ の場合に良い性能を達成している．この結果から，高密度領域において中継端末を適切に選択できていれば，低密度領域においては効用値に基づく中継先端末の選択はほとんど役に立たないことが分かる．これらのシナリオにおいて性能改善が生じる理由は二つ考えられる．一つ目の理由は，送信元拠点や受信先拠点が疎な領域である場合，これらの領域において積極的にメッセージ拡散することによって，$\mathrm{E}[T_D]$ を大きく改善できるが，$\mathrm{E}[N_D]$ の増加にはほとんど影響を与えないからである．この長所は，主に密度型ルーチング方式の考えに基づいている．もう一つの理由は，送信元拠点及び受信先拠点を除く疎な領域ではメッセージ拡散が抑制されるからである．$\{\mathcal{V}_2, \mathcal{V}_3, \mathcal{V}_4, \mathcal{V}_5\} \setminus \{\mathcal{S} \cup \mathcal{D}\}$ に含まれる端末は効用値が低いため，高密度領域 $\mathcal{A}_1$ において遭遇したとしてもメッセージの転送が抑制される．この長所は，効用型ルーチング方式の考え方によるものである．このようにして，端末密度を考慮した効用型ルーチング方式はこれらのシナリオにおいて有効に働く．

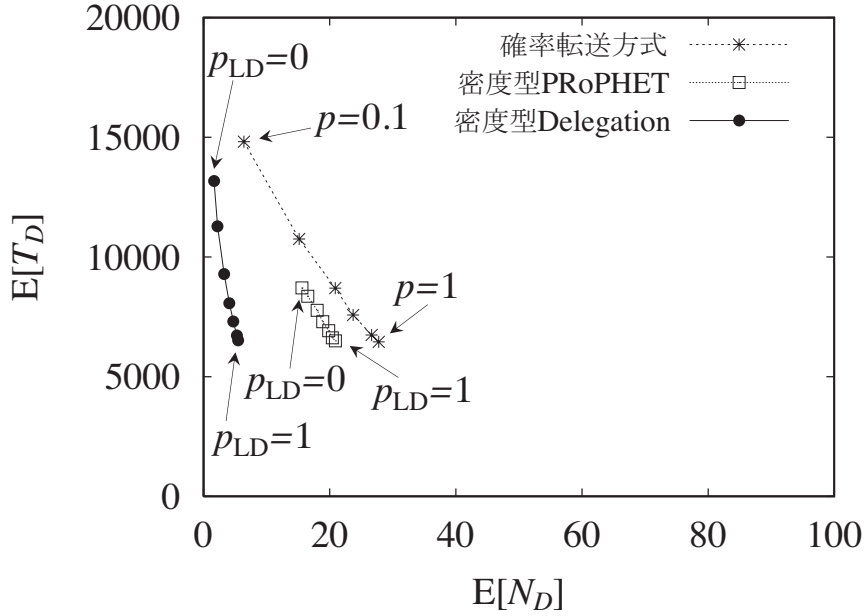
図 4.4(d) はシナリオ $\mathcal{V}_1 \to \mathcal{V}_1$ の結果を示す．p_{LD} の増加は提案方式の性能に全く影響を与えない．このシナリオでは，$\mathcal{V}_1$ 内の端末と比較して $\{\mathcal{V}_2, \mathcal{V}_3, \mathcal{V}_4, \mathcal{V}_5\}$ に含まれる端末の効用値が低い．したがって，効用型のメッセージ転送を行うことによって，疎な領域へのメッセージ拡散が抑制されている．

図 4.4(e) はシナリオ $\mathcal{V}_1 \to \mathcal{V}_i$ の結果を示している．p_{LD} の増加に伴って提案方式の $\mathrm{E}[N_D]$ が増加するが，p_{LD} に対して $\mathrm{E}[T_D]$ は感度が低い．さらに，感染型ルーチング方式（$p = 1$ の確率転送方式）と比較して密度型 Delegation 方式の $\mathrm{E}[T_D]$ は大きい．密度型 Delegation 方式では，一旦，効用値が相対的に高い $\mathcal{V}_1$ 内の端末にメッセージの複製が転送されると，その後 $\mathcal{V}_1$ 内の端末に複製の転送をほとんど行えない．一方で，密度型 PRoPHET 方式では，密な領域 $\mathcal{A}_1$ において積極的に複製を転送するため，密度型 Delegation 方式と比較して $\mathrm{E}[T_D]$ は小さくなり $\mathrm{E}[N_D]$ は大きくなる．

最後に，包括的なシナリオ $\mathcal{V} \to \mathcal{V}$ について考える．図 4.4(f) はシナリオ $\mathcal{V} \to \mathcal{V}$ の結果を示しており，表 4.2 はシナリオ $\mathcal{V} \to \mathcal{V}$ に含まれる送受信端末ペアの割合を示している．最も大きな割合は p_{LD} の増加が性能改善に寄与する場合であり，二番目に大きな割合は p_{LD} の増加が影響を与えない場合である．したがって，密度型 Delegation 方式では p_{LD} の増加が $\mathrm{E}[T_D]$ を大きく減少させるが，$\mathrm{E}[N_D]$ の増加は小さく抑えられ，低密度領域における積極的なメッセージ拡散が総合的に有効であると結論づけられる．また，密度型 PRoPHET 方式と比較して，密度型 Delegation 方式の $\mathrm{E}[T_D]$ は 2 倍以上大きく，$\mathrm{E}[N_D]$ は半分以下になっており，このシナリオではいずれのルーチング方式も同程度の効率性である．



(a) シナリオ $\mathcal{V}_i \to \mathcal{V}_1$

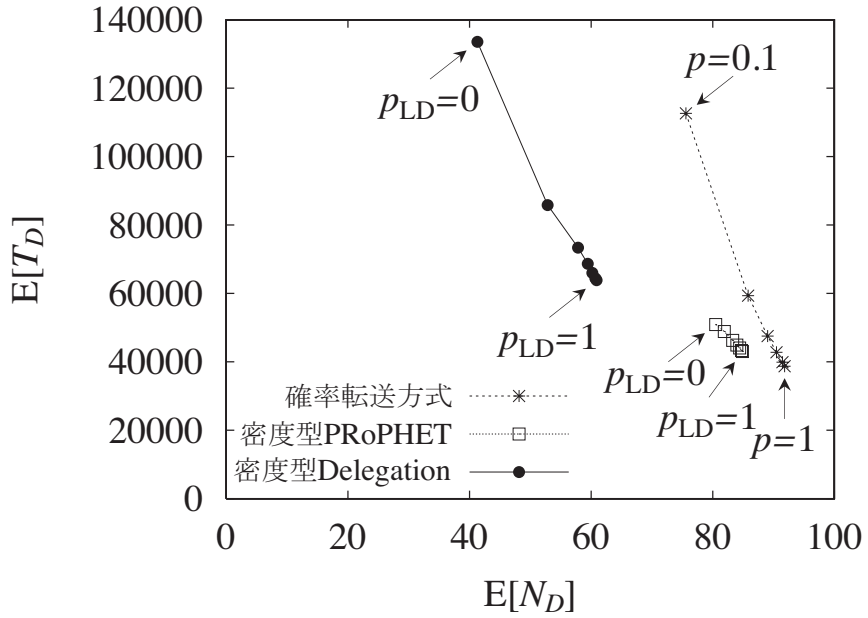


(b) シナリオ $\mathcal{V}_i \to \mathcal{V}_i$

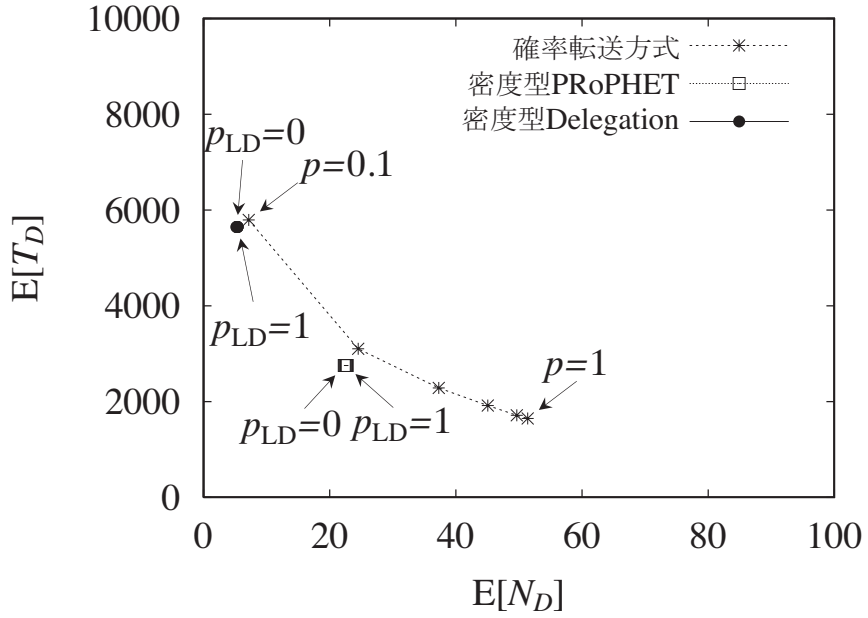
図 4.4: E[N_D] に対する E[T_D] の変化 ($p_{\mathrm{HD}} = 0$ の場合)

4.5.4 確率 p_{HD} の影響

次に，$p_{\mathrm{LD}} = 1$ の場合に，確率 p_{HD} が性能に与える影響について考える．$p_{\mathrm{HD=0}}$ の場合，シナリオ $\mathcal{V}_1 \to \mathcal{V}_1$ では密度型 Delegation 方式は高密度領域 $\mathcal{A}_1$ においてメッセージの転送を抑制しすぎ，密度型 PRoPHET 方式は積極的にメッセージ転送を行い過ぎることを示した．そこで，密度型 Delegation 方式の p_{HD} を増加させ



(c) シナリオ $\mathcal{V}_i \to \mathcal{V}_j$

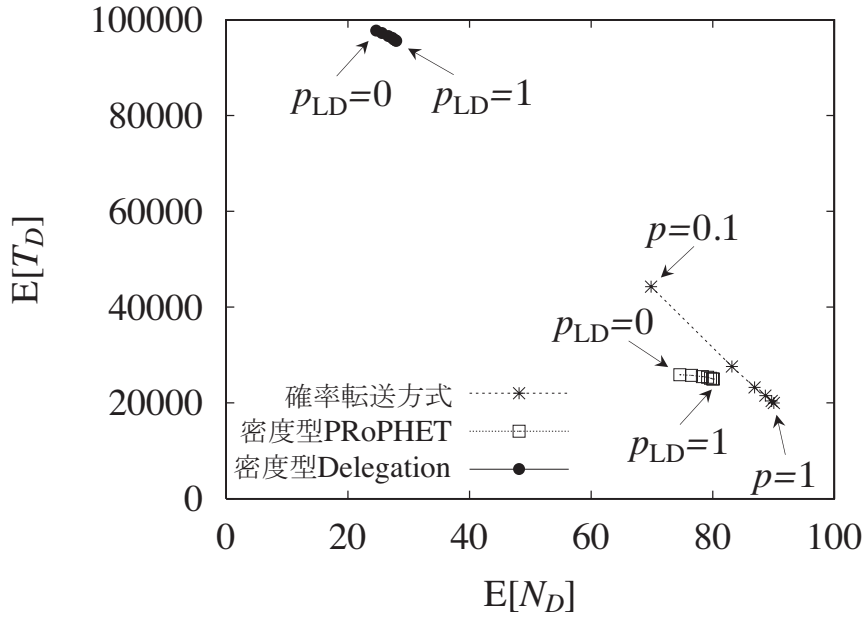


(d) シナリオ $\mathcal{V}_1 \to \mathcal{V}_1$

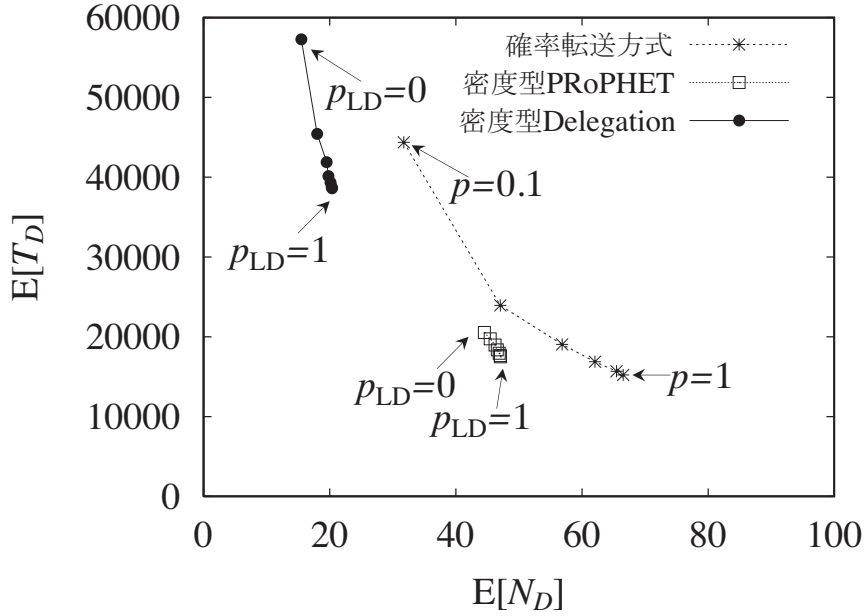
図 4.4: E[N_D] に対する E[T_D] の変化 ($p_{\mathrm{HD}} = 0$ の場合)

ることによって性能を改善できるか否かについて調べる．

図 4.5 は E[N_D] に対する平均配送遅延 E[T_D] の変化を示している．性能比較のために，確率転送方式に付け加えて，3 章で述べた端末密度を考慮した確率転送方式の結果についても示す．表記の簡単化のため，以降では端末密度を考慮した確率転送方式を密度型ルーチング方式と呼ぶことにする．密度型ルーチング方式では，転送確率 p_F は低密度領域においては 1 に，高密度領域においては p ($0 < p <\leq 1$



(e) シナリオ $\mathcal{V}_1 \to \mathcal{V}_i$



(f) シナリオ $\mathcal{V} \to \mathcal{V}$

図 4.4: E[N_D] に対する E[T_D] の変化 ($p_{\mathrm{HD}} = 0$ の場合)

) に設定される．

図 4.5(a) はシナリオ $\mathcal{V}_1 \to \mathcal{V}_1$ の結果を示している．このシナリオでは受信先端末にメッセージが配送されるとすぐに高密度領域にアンチパケットを拡散することができるため，0 よりもわずかに大きな p_{HD} が適していることが分かる．

図 4.5(b) と図 4.5(c) はシナリオ $\mathcal{V}_i \to \mathcal{V}_i$ とシナリオ $\mathcal{V}_i \to \mathcal{V}_i$ の結果を示している．p_{HD} の増加に伴って E[N_D] は増加するが，E[T_D] はほとんど変化しない．

表 4.2: シナリオ $\mathcal{V} \to \mathcal{V}$ における送受信端末ペア $(\mathcal{S}, \mathcal{D})$ の割合

ペア $(\mathcal{S}, \mathcal{D})$	$(\mathcal{V}_i, \mathcal{V}_1)$, $(\mathcal{V}_i, \mathcal{V}_i)$, $(\mathcal{V}_i, \mathcal{V}_j)$	$(\mathcal{V}_1, \mathcal{V}_1)$	$(\mathcal{V}_1, \mathcal{V}_i)$
割合 (%)	40	36	24

$\{\mathcal{V}_2, \mathcal{V}_3, \mathcal{V}_4, \mathcal{V}_5\} \setminus \mathcal{S}$ に含まれる端末へのメッセージ転送はほとんど役に立たないのだが，p_{HD} の増加に伴って高密度領域においてこれらの端末へのメッセージ転送が増加する．

図 4.5(d) と 図 4.5(e) はそれぞれシナリオ $\mathcal{V}_1 \to \mathcal{V}_i$，$\mathcal{V}_i \to \mathcal{V}_j$ の結果を示している．p_{HD} が 0 から少し大きくなると，$\mathrm{E}[T_D]$ は大きく減少するが，$\mathrm{E}[N_D]$ が大きく増加する．一般に，高密度領域においてメッセージの複製が増加すると平均配送遅延の改善に有効であるが，提案方式では p_{HD} が小さな値であったとしても過剰にメッセージ拡散が起こる．これは，高密度領域のメッセージ拡散能力が高いことに起因する．

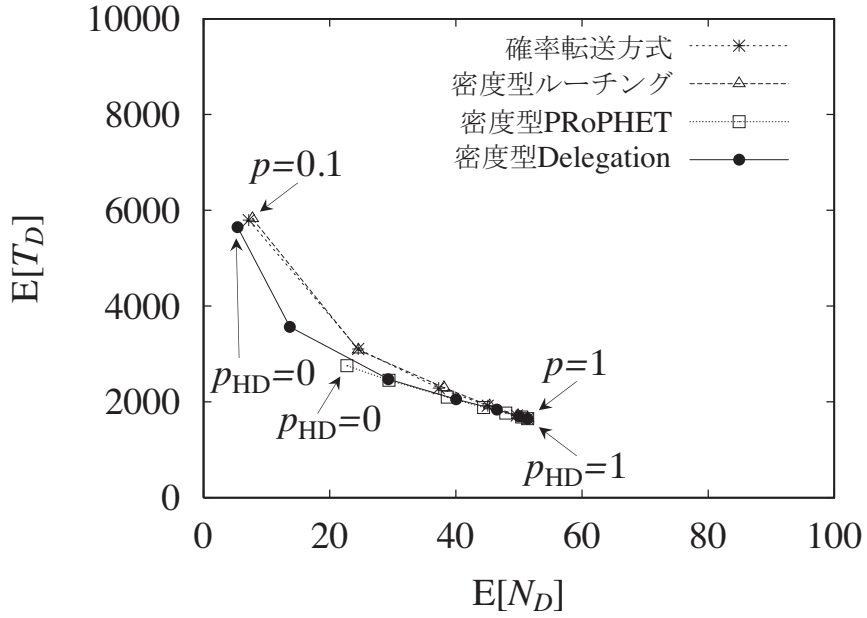
これらの観察から，高密度領域においてはより良い中継端末選択法が必要であることが分かる．特に,「各端末の拠点がどこであるか」という情報の利用が役立つ．このような拠点情報が得られたとすると，メッセージを持つ端末が受信先端末と同じ拠点である端末と遭遇した場合，積極的にメッセージの複製を転送する．一方で，拠点が異なる端末にはメッセージの複製の転送を抑制することで，受信先端末が存在する領域でのメッセージ拡散を抑制する．

最後に，送受信端末を全端末からランダムに選択するシナリオ $\mathcal{V} \to \mathcal{V}$ を考える．図 4.5(f) から，等しい $\mathrm{E}[N_D]$ に対して提案方式は確率転送方式や密度型ルーチング方式よりも $\mathrm{E}[T_D]$ が小さくなっており，効率的なメッセージ配送を行えている．さらに，密度型 Delegation 方式では $\mathrm{E}[N_D]$ の制御範囲が密度型 PRoPHET 方式よりも広い．複数の送受信端末ペアにおいて p_{HD} の増加が性能改善に貢献しないのだが，密度型 Delegation 方式は総合的に $\mathrm{E}[N_D]$ に対して有効である．

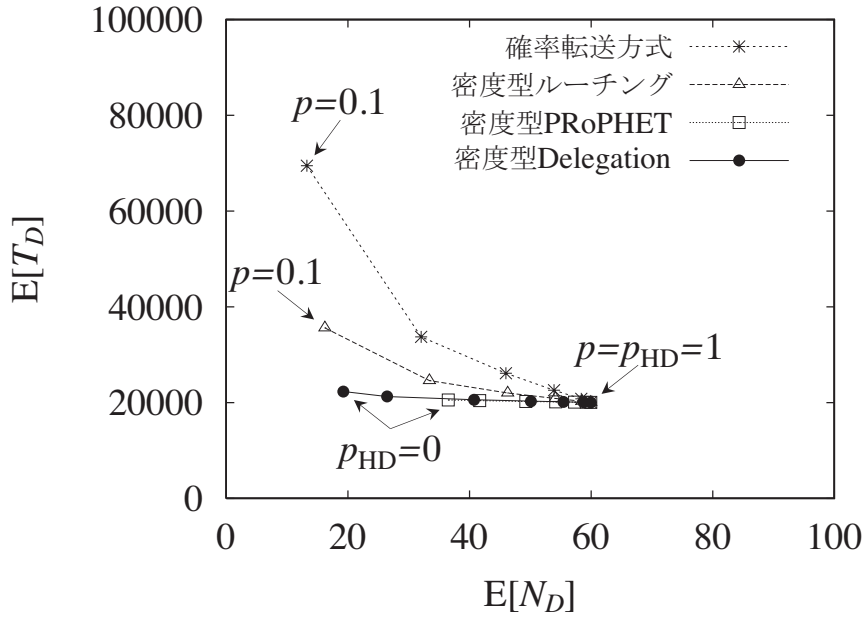
4.6 結言

本章では，端末密度を考慮をした効用型ルーチング方式を提案した．具体的には，中継先選択規則が異なる密度型 PRoPHET 方式と密度型 Delegation 方式の二つを考えた．シミュレーション実験によって高密度領域において中継端末を適切に選択していれば，低密度領域においては効用値を用いた中継先選択規則は役に立たないことを示した．さらに，提案方式が確率転送方式や密度型ルーチング方式よりも効率的なメッセージ配送を行えることを示した．よって，遭遇端末の特徴量と遭遇場所の特徴量の両方を考慮することによって効率的なメッセージ配送を行えることが分かった．しかしながら，提案方式では高密度領域において中継端末を適切に選択することができないため，拠点情報を利用した中継先選択規則などのより精巧なルーチング方式の検討が必要である．

また，本章で示したシミュレーション実験では，周辺の一つの領域と中心の領域を端末が移動する状況を想定した．これは，中心部が都市近郊であり，周辺部が郊外である状況を想定してモデル化を行った．しかしながら，実際には，ある郊外の場所から都市部を経由して別の郊外へ移動する端末が存在し得る．このような端末が含まれる場合においても端末密度と効用値の併用による性能改善が見られるか調べる必要がある．さらに，より現実的な状況においても有効であるか調べるために，リアルトレースデータを用いたシミュレーション実験を行う必要があるが，これまで遭遇地点情報を含むリアルトレースデータの収集はほとんど行われていないため，このようなトレースデータの収集が今後の課題である．

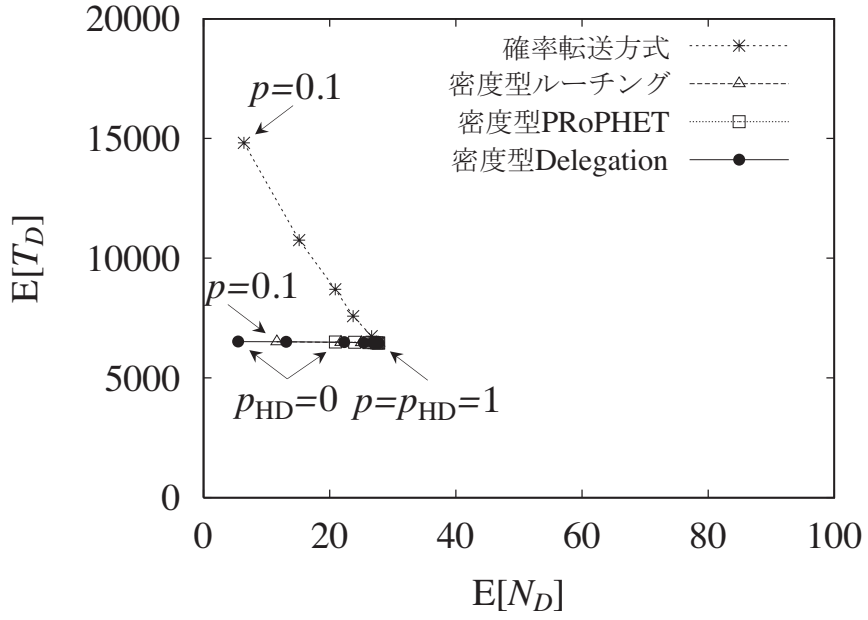


(a) シナリオ $\mathcal{V}_1 \to \mathcal{V}_1$

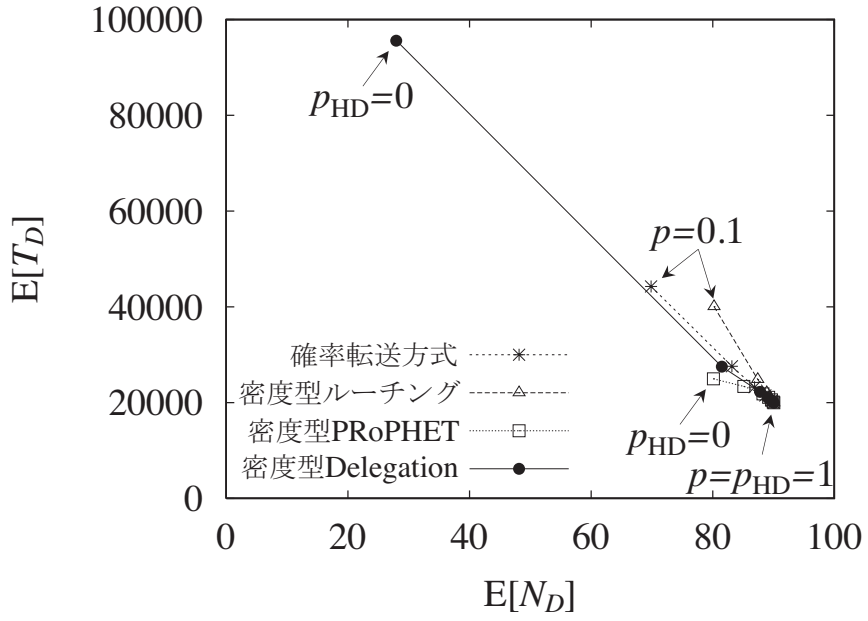


(b) シナリオ $\mathcal{V}_i \to \mathcal{V}_1$

図 4.5: E[N_D] に対する E[T_D] の変化 ($p_{\mathrm{LD}} = 1$ の場合)

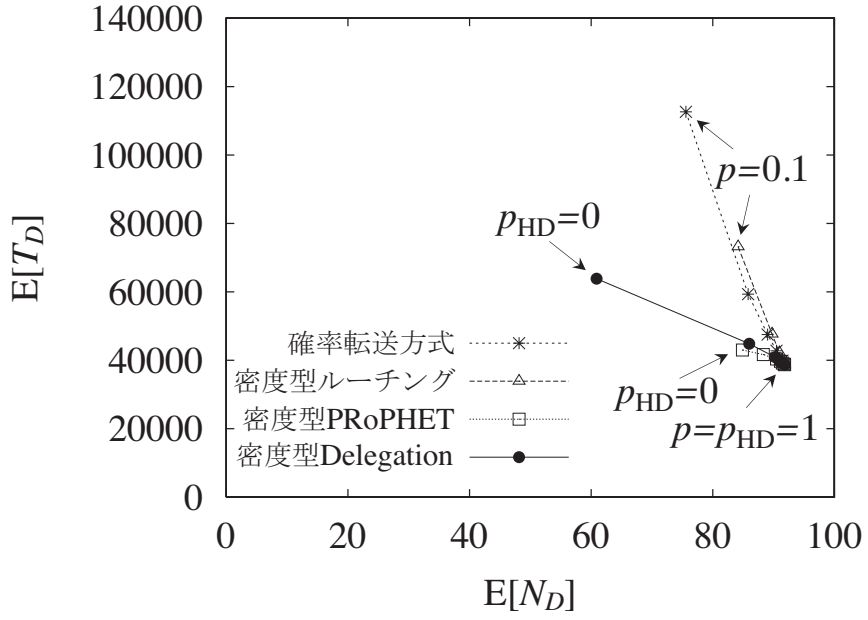


(c) シナリオ $\mathcal{V}_i \to \mathcal{V}_i$

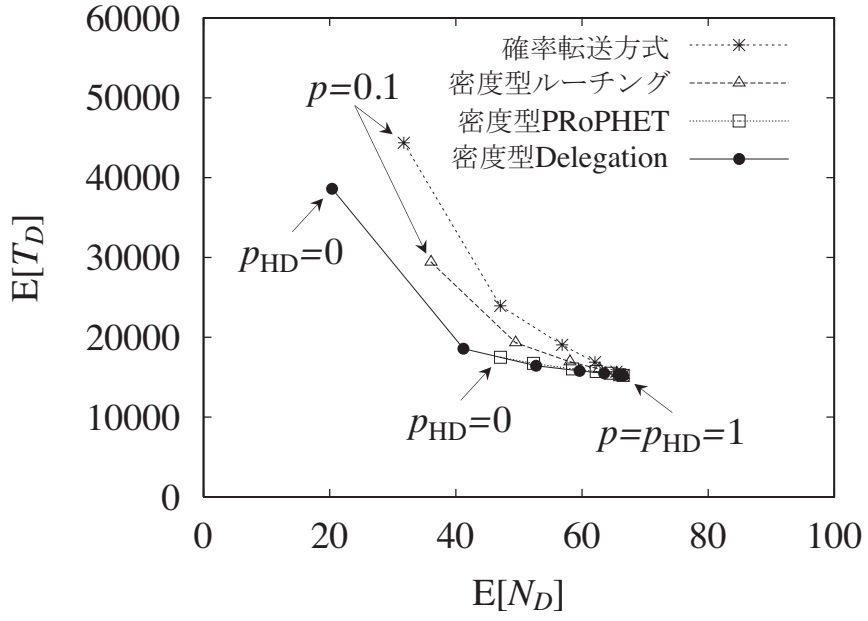


(d) シナリオ $\mathcal{V}_1 \to \mathcal{V}_i$

図 4.5: E[N_D] に対する E[T_D] の変化 ($p_{\mathrm{LD}} = 1$ の場合)



(e) シナリオ $\mathcal{V}_i \to \mathcal{V}_j$



(f) シナリオ $\mathcal{V} \to \mathcal{V}$

図 4.5: E[N_D] に対する E[T_D] の変化 ($p_{\mathrm{LD}} = 1$ の場合)

第5章

結論

本論文では，モバイルアドホックネットワークにおける蓄積運搬転送型ルーチング方式について議論してきた．本章では，これまでに行った議論や得られた結果についてまとめる．

1章では，まず，DTN 技術が利用される状況とその課題について述べた．次に，DTN 技術の代表例である蓄積運搬転送型ルーチング方式に着目し，その技術課題について述べた．さらに，蓄積運搬転送型ルーチング方式をネットワーク情報の利用の有無によって分類した．既存のネットワーク情報を用いないルーチング方式では，設定パラメタに対する感度が低く，かつ，達成可能なメッセージ配送の効率性が高いという特徴を持つものについてほとんど議論されていないことを示した．さらに，既存のネットワーク情報を用いるルーチング方式では，遭遇端末の特徴量に着目したルーチング方式が大半であり，遭遇場所の特徴量に着目したルーチング方式は少ないことを示した．

2章では，複製の生成権利を有する散布端末の最大数を制限するマルチスプレッダルーチング方式を提案した．マルチスプレッダルーチング方式は，ネットワーク情報を用いないルーチング方式に分類される．マルコフ解析及びリアルトレースデータを用いたシミュレーション実験によって，散布端末の最大数を小さく設定しておけば効率的なメッセージ配送を行うことができることを示した．よって，マルチスプレッダルーチング方式は，設定パラメタに対する感度が低く，かつ，達成可能なメッセージ配送の効率性が高いことが分かった．このような特性を持つルーチング方式は，ネットワーク情報が取得できない状況に有効である．

3章では，端末密度推定に基づく確率転送方式を提案した．各端末が遭遇地点情報を収集し，収集した情報に基づいた推定によって得られる端末密度に応じてメッセージの複製の転送確率を制御することによって効率的なメッセージ配送を行えることを示した．さらに，端末密度推定の精度に対するロバスト性について議論し，高密度領域をある程度推定できてさえいれば，提案方式はかなりロバストであることを示した．これらの結果から，遭遇場所の特徴量を考慮することによって効率的なメッセージ配送を行えることが分かった．

4章では，端末密度を考慮した効用型ルーチング方式を提案した．遭遇端末の特徴量である効用値と遭遇場所の特徴量の両方を考慮することによって，単独の情

報を利用するよりも効率的なメッセージ配送を行えることをシミュレーション実験によって示した．

各章の結果から，蓄積運搬転送型ルーチング方式の性能はネットワーク環境に大きく依存することが分かる．2 章で述べたマルチスプレッダルーチング方式では，ネットワーク環境によらず，ある程度効率の良いメッセージ配送を行える．しかしながら，ネットワーク情報を取得できる場合には，そのネットワーク環境に適した，より効率的なメッセージ配送を行えるルーチング方式が必ず存在する．

3 章や 4 章で示したルーチング方式では，想定通りのネットワーク環境であれば，効率の良いメッセージ配送を実現できる．3 章で述べたルーチング方式では遭遇が発生する場所に偏りが存在することを前提として設計されている．また，4 章で述べたルーチング方式では，遭遇場所に加えて遭遇する端末の偏りを前提として設計が行われている．しかしながら，これらの偏りがほとんどないネットワーク環境においては，ネットワーク情報を用いないルーチング方式とほとんど同じ効率性しか達成できないにも関わらず，ネットワーク情報を求める計算を行うために余計な負荷を端末にかけることとなる．

したがって，ネットワークの設計者が用いるルーチング方式を選択する際には，達成したいメッセージ配送の効率性とネットワーク情報を得るために必要な計算量を十分考慮して判断する必要がある．つまり，効率性を多少犠牲にすることになるがネットワーク情報の取得にかかる負荷を軽減するか，もしくは，効率性を重視しネットワーク情報の取得にかかる負荷を大きく被るかを選択する必要がある．効率性を多少犠牲にしても問題がない場合，マルチスプレッダルーチング方式が一つの選択候補になり得る．また，遭遇場所の偏りが明らかにある場合には，端末密度を考慮した確率的転送方式を用いることによって効率性を重視できる．さらに，遭遇端末と遭遇場所の偏りが明らかにある場合には，端末密度を考慮した効用型ルーチング方式を用いることでより効率性を重視したメッセージ配送を行える．このように各ルーチング方式の特性を十分踏まえ，それぞれのネットワーク環境に適したルーチング方式を選択することが重要である．

付 録 A

マルチスプレッダルーチング方式における状態遷移率

2 章で述べたマルチスプレッダルーチング方式における状態 $(\boldsymbol{x}, k, 0)$ と状態 $(\boldsymbol{x}, k, m)$ からの遷移率について述べる．

A.1 状態 $(\boldsymbol{x}, k, 0)$ からの遷移率

表 2.1 を得るために，状態 $(\boldsymbol{x}, k, 0)$ において二つの端末が遭遇する状況を考える．

i) カウンタ値 c_j $(j = 1, 2, \ldots, n(P) - 1)$ の散布端末が不保持端末に遭遇する場合: 遭遇後，不保持端末はカウンタ値 $\lfloor c_j/2 \rfloor$ を持つ散布端末になる．このとき同時に，散布端末のカウンタ値が $\lceil c_j/2 \rceil$ に交信される．したがって，状態 $(\boldsymbol{x}, k, 0)$ から状態 $(\boldsymbol{x}'_j, k+1, 0)$ へ遷移する．ただし，$\boldsymbol{x}'_j$ は式 (2.3) によって与えられる．状態 $(\boldsymbol{x}, k, 0)$ においては，カウンタ値 c_j の散布端末が x_j 個と不保持端末が $N - k$ 個存在するため，状態 $(\boldsymbol{x}, k, 0)$ から状態 $(\boldsymbol{x}'_j, k+1, 0)$ への遷移率は $(N - k)x_j\lambda$ である．

ii) カウンタ値 $c = 1$ の散布端末が不保持端末に遭遇する場合: 遭遇後，不保持端末が中継端末になる．したがって，状態 $(\boldsymbol{x}, k, 0)$ から状態 $(\boldsymbol{x}, k+1, 0)$ への遷移が起こる．その遷移率は $(N - k)x_{n(P)}\lambda$ で与えられる．

iii) カウンタ値 c_j $(j = 1, 2, \ldots, n(P) - 1)$ の散布端末が中継端末に遭遇する場合: 節 2.2.1 で述べたように，中継端末がカウンタ値 $\lfloor c_j/2 \rfloor$ の散布端末に変化する．このとき同時に，散布端末のカウンタ値が $\lceil c_j/2 \rceil$ に変化する．したがって，状態 $(\boldsymbol{x}, k, 0)$ から状態 $(\boldsymbol{x}'_j, k, 0)$ へ遷移する．ただし，$\boldsymbol{x}'_j$ は式 (2.3) によって与えられる．状態 $(\boldsymbol{x}, k, 0)$ において $N_R^{[\boldsymbol{x},k]}$ 個の端末が存在するため，状態 $(\boldsymbol{x}, k, 0)$ から状態 $(\boldsymbol{x}'_j, k, 0)$ への遷移率は $N_R^{[\boldsymbol{x},k]}x_j\lambda$ で与えられる．

iv) 散布端末が受信先端末と遭遇する場合: 散布端末がメッセージを受信先端末に転送するとすぐに，二つの端末はアンチパケットを生成する．したがって，散布端末が保持しているメッセージのカウンタ値は c_j であるため，状

態 $(\boldsymbol{x}, k, 0)$ から状態 $(\boldsymbol{x}-\boldsymbol{e}_j, k-1, 2)$ への遷移が起こる．受信先端末は一つであるため，状態 $(\boldsymbol{x}, k, 0)$ から状態 $(\boldsymbol{x}-\boldsymbol{e}_j, k-1, 2)$ への遷移率は $x_j\lambda$ で与えられる．

v) 中継端末が受信先端末に遭遇する場合: 中継端末が受信先端末にメッセージを転送するとすぐに，二つの端末はアンチパケットを生成する．したがって，状態 $(\boldsymbol{x}, k, 0)$ から状態 $(\boldsymbol{x}, k-1, 2)$ への遷移が起こり，その遷移率は $N_R^{[\boldsymbol{x},k]}\lambda$ で与えられる．

vi) その他の場合: 散布端末とメッセージ数に変化が生じないため，状態遷移は起こらない．

A.2 状態 $(\boldsymbol{x}, k, m)$ からの遷移率

表 2.2 を得るために，状態 $(\boldsymbol{x}, k, m)$ $(m \geq 2)$ において二つの端末が遭遇することを考える．

i) カウンタ値 c_j $(j = 1, 2, \cdots, n(P)-1)$ の散布端末が不保持端末と遭遇する場合: 不保持端末がカウンタ値 $\lfloor c_j/2 \rfloor$ の散布端末になる．このとき同時に，散布端末のカウンタ値は $\lceil c_j/2 \rceil$ に更新される．したがって，状態 $(\boldsymbol{x}, k, m)$ から状態 $(\boldsymbol{x}'_j, k+1, m)$ への遷移が起こる．状態 $(\boldsymbol{x}, k, m)$ においてカウンタ値 c_j の散布端末が x_j 個と不保持端末が $N+1-k-m$ 個存在するため，遷移率は $(N+1-k-m)x_j\lambda$ で与えられる．

ii) カウンタ値 $c_{n(P)}$ $(=1)$ の散布端末が不保持端末と遭遇する場合: 不保持端末が中継端末になる．したがって，状態 $(\boldsymbol{x}, k, m)$ から状態 $(\boldsymbol{x}, k+1, m)$ への遷移が起こる．その遷移率は $(N+1-k-m)x_{n(P)}\lambda$ と与えられる．

iii) カウンタ値 c_j $(j = 1, 2, \cdots, n(P)-1)$ の散布端末が中継端末に遭遇する場合: 中継端末がカウンタ値 $\lfloor c_j/2 \rfloor$ の散布端末に変わる．このとき同時に，散布端末のカウンタ値は $\lceil c_j/2 \rceil$ に更新される． したがって，状態 $(\boldsymbol{x}, k, m)$ から状態 $(\boldsymbol{x}'_j, k, m)$ へ遷移する．状態 $(\boldsymbol{x}, k, m)$ における中継端末の数は式 (2.2) の $N_R^{[\boldsymbol{x},k]}$ で与えられるため，遷移率は $N_R^{[\boldsymbol{x},k]}x_j\lambda$ で与えられる．

iv) 散布端末がアンチパケット保持端末に遭遇する場合: 散布端末はアンチパケットを受信し，保持するメッセージの削除を行う．したがって，散布端末がカウンタ値 c_j $(j = 1, 2, \cdots, n(P))$ のメッセージを保持する場合，状態 $(\boldsymbol{x}, k, m)$ から状態 $(\boldsymbol{x}-\boldsymbol{e}_j, k-1, m+1)$ への遷移が起こる．アンチパケット保持端末は m 個存在するため，遷移率は $mx_j\lambda$ で与えられる．

v) 中継端末がアンチパケット保持端末と遭遇する場合: 中継端末はアンチパケットを保持し，メッセージを削除する．したがって，状態 $(\boldsymbol{x}, k, m)$ から状態 $(\boldsymbol{x}, k-1, m+1)$ への遷移が起こる．その遷移率は $mN_R^{[\boldsymbol{x},k]}\lambda$ で与えられる．

vi) アンチパケット保持端末が不保持端末に遭遇する場合:　不保持端末がアンチパケットを受信する．したがって，状態 $(\boldsymbol{x}, k, m)$ から状態 $(\boldsymbol{x}, k, m+1)$ へ遷移し，その遷移率は $m(N+1-k-m)\lambda$ で与えられる．

vii) その他の場合:　散布端末数，メッセージの複製数，アンチパケットの総数が変化しないため，状態遷移が起こらない．

付 録B

マルチスプレッダルーチング方式の性能評価

2 章において述べたマルチスプレッダルーチング方式の平均メッセージ生存時間，システム内の平均メッセージ累積数，平均複製転送回数の計算手順について述べる．

B.1 平均メッセージ生存時間

$T_E(\boldsymbol{x}, k, m)$ $(m = 0, 2, 3, \ldots, N+1)$, $(\boldsymbol{x}, k, m) \in \mathcal{S}_P(m))$ を以下のように定義する．

$$T_E(\boldsymbol{x}, k, m) = \mathrm{E}[T_E \mid \boldsymbol{Z}(0) = (\boldsymbol{x}, k, m)]$$

ただし，T_E は式 (2.7) で定義される．定義から，$\mathrm{E}[T_E] = T_E(\boldsymbol{e}_1, 1, 0)$ である．表 2.1 から $T_E(\boldsymbol{x}, k, 0)$ $(N_S^{[\boldsymbol{x}]} \leq k \leq N)$ は次式を満たす．

$$\begin{aligned}
T_E(\boldsymbol{x}, k, 0) \;=\; & \frac{1}{\lambda_1(\boldsymbol{x}, k, 0)} + \sum_{j=1}^{n(P)-1} \frac{x_j N_R^{[\boldsymbol{x},k]} \lambda}{\lambda_1(\boldsymbol{x}, k, 0)} T_E(\boldsymbol{x}'_j, k, 0) \\
& + \sum_{j=1}^{n(P)-1} \frac{(N-k) x_j \lambda}{\lambda_1(\boldsymbol{x}, k, 0)} T_E(\boldsymbol{x}'_j, k+1, 0) \\
& + \sum_{j=1}^{n(P)-1} \frac{x_j \lambda}{\lambda_1(\boldsymbol{x}, k, 0)} T_E(\boldsymbol{x} - \boldsymbol{e}_j, k-1, 2) \\
& + \frac{(N-k) x_{n(P)} \lambda}{\lambda_1(\boldsymbol{x}, k, 0)} T_E(\boldsymbol{x}, k+1, 0) \\
& + \frac{x_{n(P)} \lambda}{\lambda_1(\boldsymbol{x}, k, 0)} T_E(\boldsymbol{x} - \boldsymbol{e}_{n(P)}, k-1, 2) \\
& + \frac{N_R^{[\boldsymbol{x},k]} \lambda}{\lambda_1(\boldsymbol{x}, k, 0)} T_E(\boldsymbol{x}, k-1, 2)
\end{aligned} \tag{B.1}$$

ただし，$T_E(\boldsymbol{x}, N+1, 0) = T_E(\boldsymbol{x}, 0, 0) = T_E(\boldsymbol{0}, k, 0) = 0$ である．

また，表 2.2 から $m \geq 2$ に対して状態 $(\boldsymbol{x}, k, m)$ からの遭遇率の和を $\lambda_2(\boldsymbol{x}, k, m)$ $((\boldsymbol{x}, k, m) \in \mathcal{S}_P(m))$ とする．

$$\begin{aligned}\lambda_2(\boldsymbol{x}, k, m) &= \sum_{j=1}^{n(P)-1} (N_R^{[\boldsymbol{x},k]} + N - k + 1)x_j\lambda \\ &\quad + (N-k+1)x_{n(P)}\lambda \\ &\quad + mN_R^{[\boldsymbol{x},k]}\lambda + m(N+1-k-m)\lambda\end{aligned}$$

したがって，$T_E(\boldsymbol{x}, k, m)$ $(N_S^{[\boldsymbol{x}]} \leq k \leq N-1, 2 \leq m \leq N, k+m \leq N+1)$ は次式を満たす．

$$\begin{aligned}T_E(\boldsymbol{x}, k, m) &= \frac{1}{\lambda_2(\boldsymbol{x}, k, m)} + \sum_{j=1}^{n(P)-1} \frac{N_R^{[\boldsymbol{x},k]}x_j\lambda}{\lambda_2(\boldsymbol{x}, k, m)} T_E(\boldsymbol{x}'_j, k, m) \\ &\quad + \sum_{j=1}^{n(P)-1} \frac{(N+1-k-m)x_j\lambda}{\lambda_2(\boldsymbol{x}, k, m)} T_E(\boldsymbol{x}'_j, k+1, m) \\ &\quad + \sum_{j=1}^{n(P)-1} \frac{mx_j\lambda}{\lambda_2(\boldsymbol{x}, k, m)} T_E(\boldsymbol{x} - \boldsymbol{e}_j, k-1, m+1) \\ &\quad + \frac{(N+1-k-m)x_{n(P)}\lambda}{\lambda_2(\boldsymbol{x}, k, m)} T_E(\boldsymbol{x}, k+1, m) \\ &\quad + \frac{mx_{n(P)}\lambda}{\lambda_2(\boldsymbol{x}, k, m)} T_E(\boldsymbol{x} - \boldsymbol{e}_{n(P)}, k-1, m+1) \\ &\quad + \frac{mN_R^{[\boldsymbol{x},k]}\lambda}{\lambda_2(\boldsymbol{x}, k, m)} T_E(\boldsymbol{x}, k-1, m+1) \\ &\quad + \frac{m(N+1-k-m)\lambda}{\lambda_2(\boldsymbol{x}, k, m)} T_E(\boldsymbol{x}, k, m+1) \qquad \text{(B.2)}\end{aligned}$$

ただし，$T_E(\boldsymbol{x}, 0, m) = T_E(\boldsymbol{x}, N+2-m, m) = 0$ である．各々の m（$m = 0, 2, 3, \ldots, N+1$）に対して $\mathcal{X}_P(m)$ に含まれる散布端末の状態は $N_S^{[\boldsymbol{x}]}$ の昇順に番号を付けられるため，$\mathcal{X}_P(m)$ の l（$l = 1, 2, \ldots, |\mathcal{X}_P(m)|$）番目の要素を $\boldsymbol{x}^{[l]} = (x_1^{[l]}, x_2^{[l]}, \ldots, x_{n(P)}^{[l]})$ と表す．これによって，図 B.1 に示される後進再帰手順を用いることができ，$\mathrm{E}[T_E] = T_E(\boldsymbol{e}_1, 1, 0)$ を数値計算によって求めることができる．

図 B.2 は散布端末の最大数 P に対する $\mathrm{E}[T_E]$ の変化を示している．$\mathrm{E}[T_E]$ は P に対して単調減少な凸関数であり，図 2.7 において示した平均配送遅延 $\mathrm{E}[T_D]$ の結果と同じ傾向であることが分かる．

ここで，回復手法を用いてメッセージの複製を削除するのに要する時間を $T_R = T_E - T_D$ と定義する．図 B.3 は散布端末の最大数 P に対する回復手法を用いてメッセージの複製を削除するのに要する平均時間 $\mathrm{E}[T_R]$ の変化を示している．$\mathrm{E}[T_R]$ は P に対する増加凹関数になっている．大きな P の場合，受信先端末へメッセージが配送された後にも多くのメッセージの複製が生成されるため，P の増加に伴って $\mathrm{E}[T_R]$ は増加する．しかしながら，ネットワーク内の端末数の制限によってメッ

```
入力: P, S_P(m), X_P(m) (m = 0, 2, 3, ..., N)
出力: E[T_E]
FOR m := N STEP −1 UNTIL 2 DO
  FOR l := |X_P(m)| STEP −1 UNTIL 1 DO
    IF (x^[l], N_S^[x^[l]] + 1, m) ∈ S_P(m) THEN
      FOR k := N − m STEP −1 UNTIL N_S^[x^[l]]
        式 (B.2) を用いて T_E(x^[l], k, m) を計算する.
      ENDFOR
    ELSE
      式 (B.2) を用いて T_E(x^[l], N_S^[x^[l]], m) を計算する.
    ENDIF
  ENDFOR
ENDFOR
FOR l := |X_P(0)| STEP −1 UNTIL 1 DO
  IF x_n(P)^[l] ≥ 1 THEN
    FOR k := N STEP −1 UNTIL N_S^[x^[l]]
      式 (B.1) を用いて T_E(x^[l], k, 0) を計算する.
    ENDFOR
  ELSE
    式 (B.1) を用いて T_E(x^[l], N_S^[x^[l]], 0) を計算する.
  ENDIF
ENDFOR
E[T_E] := T_E(e_1, 1, 0) とする.
```

図 B.1: $\mathrm{E}[T_E]$ を計算するための後進再帰手順

セージの複製数が制限されるため，$\mathrm{E}[T_R]$ は増加凹関数となる．さらに，平均配送遅延 $\mathrm{E}[T_D]$ と比較して，$\mathrm{E}[T_R]$ は P に対する感度が小さいことが分かる．これは，感染型ルーチング方式によってアンチパケットをネットワーク中にかなりはやく拡散できるため，パラメタ P による $\mathrm{E}[T_R]$ への影響はほとんどない．

B.2 システム内の平均メッセージ累積数

端末負荷 O_L とメッセージに関する情報の平均送信回数 $\mathrm{E}[N_I]$ は $R(\boldsymbol{e}_1, 1, 0)$ を用いて表現される．$R(\boldsymbol{e}_1, 1, 0)$ は以下の手順によって求めることができる．$R(\boldsymbol{x}, k, m)$ $(m = 0, 2, 3, \ldots, N+1,\ (\boldsymbol{x}, k, m) \in \mathcal{S}_P(m))$ を次式のように定義する．

$$R(\boldsymbol{x}, k, m) = \mathrm{E}\left[\int_0^{T_E} K(t)dt \mid \boldsymbol{Z}(0) = (\boldsymbol{x}, k, m)\right]$$

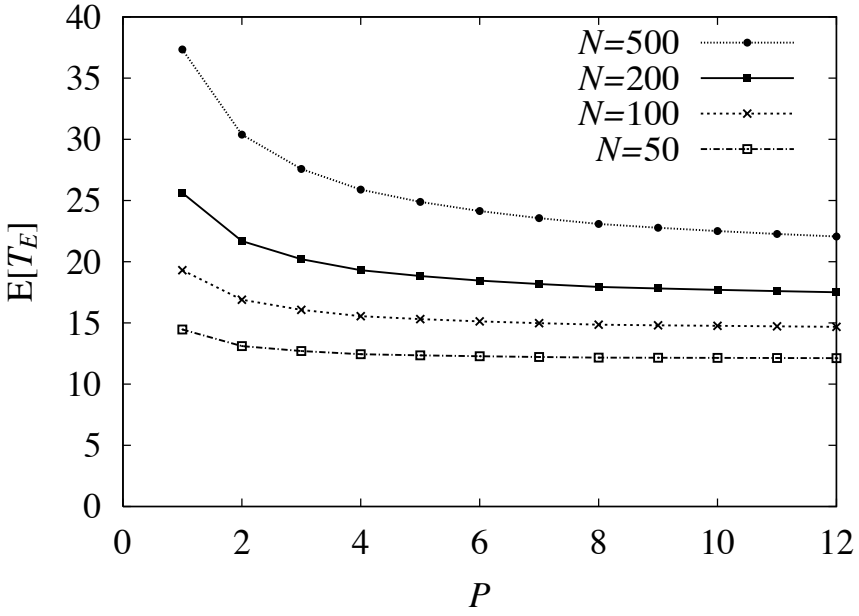


図 B.2: 平均メッセージ生存時間 E[T_E]

$T_E(\boldsymbol{x}, k, m)$ の場合と同様に，$R(\boldsymbol{x}, k, 0)$ $(N_S^{[\boldsymbol{x}]} \leq k \leq N)$ が次式を満たすことが容易に分かる．

$$
\begin{aligned}
R(\boldsymbol{x}, k, 0) \quad = \quad & \frac{k}{\lambda_1(\boldsymbol{x}, k, 0)} + \sum_{j=1}^{n(P)-1} \frac{N_R^{[\boldsymbol{x},k]} x_j \lambda}{\lambda_1(\boldsymbol{x}, k, 0)} R(\boldsymbol{x}'_j, k, 0) \\
& + \sum_{j=1}^{n(P)-1} \frac{(N-k) x_j \lambda}{\lambda_1(\boldsymbol{x}, k, 0)} R(\boldsymbol{x}'_j, k+1, 0) \\
& + \sum_{j=1}^{n(P)-1} \frac{x_j \lambda}{\lambda_1(\boldsymbol{x}, k, 0)} R(\boldsymbol{x} - \boldsymbol{e}_j, k-1, 2) \\
& + \frac{(N-k) x_{n(P)} \lambda}{\lambda_1(\boldsymbol{x}, k, 0)} R(\boldsymbol{x}, k+1, 0) \\
& + \frac{x_{n(P)} \lambda}{\lambda_1(\boldsymbol{x}, k, 0)} R(\boldsymbol{x} - \boldsymbol{e}_{n(P)}, k-1, 2) \\
& + \frac{N_R^{[\boldsymbol{x},k]} \lambda}{\lambda_1(\boldsymbol{x}, k, 0)} R(\boldsymbol{x}, k-1, 2)
\end{aligned}
$$

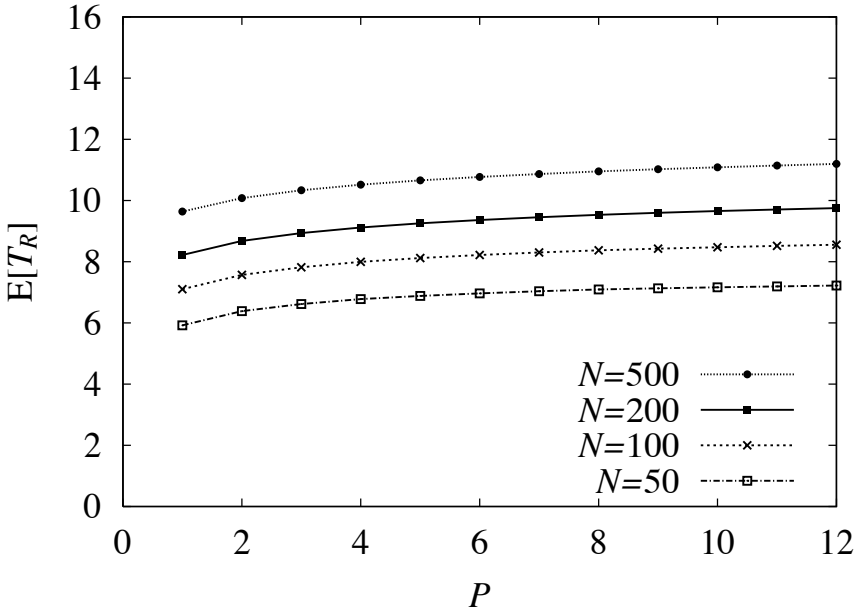


図 B.3: 回復手法を用いてメッセージを削除するのに要する平均時間 E[T_R]

ただし，$R(\boldsymbol{x}, N+1, 0) = R(\boldsymbol{x}, 0, 0) = R(\mathbf{0}, k, 0) = 0$ である．また，$R(\boldsymbol{x}, k, m)$ $(N_S^{[\boldsymbol{x}]} \leq k \leq N-1,\ 2 \leq m \leq N,\ k+m \leq N+1)$ は次式を満たす．

$$
\begin{aligned}
R(\boldsymbol{x}, k, m) \;=\; & \frac{k}{\lambda_2(\boldsymbol{x}, k, m)} + \sum_{j=1}^{n(P)-1} \frac{N_R^{[\boldsymbol{x},k]} x_j \lambda}{\lambda_2(\boldsymbol{x}, k, m)} R(\boldsymbol{x}'_j, k, m) \\
& + \sum_{j=1}^{n(P)-1} \frac{(N+1-k-m) x_j \lambda}{\lambda_2(\boldsymbol{x}, k, m)} R(\boldsymbol{x}'_j, k+1, m) \\
& + \sum_{j=1}^{n(P)-1} \frac{m x_j \lambda}{\lambda_2(\boldsymbol{x}, k, m)} R(\boldsymbol{x} - \boldsymbol{e}_j, k-1, m+1) \\
& + \frac{(N+1-k-m) x_{n(P)} \lambda}{\lambda_2(\boldsymbol{x}, k, m)} R(\boldsymbol{x}, k+1, m) \\
& + \frac{m x_{n(P)} \lambda}{\lambda_2(\boldsymbol{x}, k, m)} R(\boldsymbol{x} - \boldsymbol{e}_{n(P)}, k-1, m+1) \\
& + \frac{m N_R^{[\boldsymbol{x},k]} \lambda}{\lambda_2(\boldsymbol{x}, k, m)} R(\boldsymbol{x}, k-1, m+1) \\
& + \frac{m(N+1-k-m) \lambda}{\lambda_2(\boldsymbol{x}, k, m)} R(\boldsymbol{x}, k, m+1)
\end{aligned}
$$

ただし，$R(\boldsymbol{x}, 0, m) = R(\boldsymbol{x}, N+2-m, m) = 0$ である．したがって，$R(\boldsymbol{e}_1, 1, 0)$ は $T_E(\boldsymbol{x}^{[l]}, k, m)$ を $R(\boldsymbol{x}^{[l]}, k, m)$ に置き換えた図 B.1 の後進再帰手順による数値計算によって求めることができる．

B.3 平均複製転送回数

N_D をメッセージの複製を転送した総回数と定義する．$N_D(\boldsymbol{x}, k, m)$ $(m = 0, 2, 3, \ldots, N+1,\ (\boldsymbol{x}, k, m) \in \mathcal{S}_P(m))$ を次式で定義する．

$$N_D(\boldsymbol{x}, k, m) = \mathrm{E}[N_D \mid \boldsymbol{Z}(0) = (\boldsymbol{x}, k, m)]$$

表 2.1 から $N_D(\boldsymbol{x}, k, 0)$ $(N_S^{[\boldsymbol{x}]} \leq k \leq N)$ は次式を満たす．

$$\begin{aligned}
N_D(\boldsymbol{x}, k, 0) \;=\; & \sum_{j=1}^{n(P)-1} \frac{N_R^{[\boldsymbol{x},k]} x_j \lambda}{\lambda_1(\boldsymbol{x}, k, 0)} N_D(\boldsymbol{x}'_j, k, 0) \\
& + \sum_{j=1}^{n(P)-1} \frac{(N-k) x_j \lambda}{\lambda_1(\boldsymbol{x}, k, 0)} \{1 + N_D(\boldsymbol{x}'_j, k+1, 0)\} \\
& + \sum_{j=1}^{n(P)-1} \frac{x_j \lambda}{\lambda_1(\boldsymbol{x}, k, 0)} N_D(\boldsymbol{x} - \boldsymbol{e}_j, k-1, 2) \\
& + \frac{(N-k) x_{n(P)} \lambda}{\lambda_1(\boldsymbol{x}, k, 0)} \{1 + N_D(\boldsymbol{x}, k+1, 0)\} \\
& + \frac{x_{n(P)} \lambda}{\lambda_1(\boldsymbol{x}, k, 0)} N_D(\boldsymbol{x} - \boldsymbol{e}_{n(P)}, k-1, 2) \\
& + \frac{N_R^{[\boldsymbol{x},k]} \lambda}{\lambda_1(\boldsymbol{x}, k, 0)} N_D(\boldsymbol{x}, k-1, 2)
\end{aligned}$$

ただし，$N_D(\boldsymbol{x}, N+1, 0) = N_D(\boldsymbol{x}, 0, 0) = N_D(\boldsymbol{0}, k, 0) = 0$ である．また，表 2.2 から $N_D(\boldsymbol{x}, k, m)$ $(N_S^{[\boldsymbol{x}]} \leq k \leq N-1, 2 \leq m \leq N, k+m \leq N+1)$ が次式を満たす．

$$\begin{aligned}
N_D(\boldsymbol{x}, k, m) \;=\; & \sum_{j=1}^{n(P)-1} \frac{N_R^{[\boldsymbol{x},k]} x_j \lambda}{\lambda_2(\boldsymbol{x}, k, m)} N_D(\boldsymbol{x}'_j, k, m) \\
& + \sum_{j=1}^{n(P)-1} \frac{(N+1-k-m) x_j \lambda}{\lambda_2(\boldsymbol{x}, k, m)} \{1 + N_D(\boldsymbol{x}'_j, k+1, m)\} \\
& + \sum_{j=1}^{n(P)-1} \frac{m x_j \lambda}{\lambda_2(\boldsymbol{x}, k, m)} N_D(\boldsymbol{x} - \boldsymbol{e}_j, k-1, m+1) \\
& + \frac{(N+1-k-m) x_{n(P)} \lambda}{\lambda_2(\boldsymbol{x}, k, m)} \{1 + N_D(\boldsymbol{x}, k+1, m)\} \\
& + \frac{m x_{n(P)} \lambda}{\lambda_2(\boldsymbol{x}, k, m)} N_D(\boldsymbol{x} - \boldsymbol{e}_{n(P)}, k-1, m+1) \\
& + \frac{m N_R^{[\boldsymbol{x},k]} \lambda}{\lambda_2(\boldsymbol{x}, k, m)} N_D(\boldsymbol{x}, k-1, m+1) \\
& + \frac{m(N+1-k-m) \lambda}{\lambda_2(\boldsymbol{x}, k, m)} N_D(\boldsymbol{x}, k, m+1)
\end{aligned}$$

ただし， $N_D(\boldsymbol{x}, 0, m) = N_D(\boldsymbol{x}, N+2-m, m) = 0$ である．したがって，$\mathrm{E}[N_D] = N_D(\boldsymbol{e}_1, 1, 0)$ は $T_E(\boldsymbol{x}^{[l]}, k, m)$ を $N_D(\boldsymbol{x}^{[l]}, k, m)$ に置き換えた図 B.1 の後進再帰手順による数値計算によって求めることができる．

付 録C

端末密度分布の分散

C.1 パラメトリック密度推定法

本節では，式 (3.1) で表されるパラメトリック密度推定法における端末密度分布 $I_v(\boldsymbol{x})$ の分散 $\sigma[I_v]$ について考え，$\sigma[I_v]$ が式 (3.3) で与えられることを示す．

$$
\begin{aligned}
&\sigma^2[I_v] \\
&= \int_{\boldsymbol{x}\in\mathcal{A}} I_v^2(\boldsymbol{x})\frac{d\boldsymbol{x}}{|\mathcal{A}|} - (\mathrm{E}[I_v])^2 \\
&\approx \int_{\boldsymbol{x}\in\mathbb{R}^2} I_v^2(\boldsymbol{x})\frac{d\boldsymbol{x}}{|\mathcal{A}|} - (\mathrm{E}[I_v])^2 \\
&= \int_{\boldsymbol{x}\in\mathbb{R}^2} \sum_{i,j=1}^{K} \frac{\pi_i\pi_j}{4\pi^2|\Sigma_i|^{1/2}|\Sigma_j|^{1/2}} \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)}{2}\right) \\
&\qquad \cdot \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{\mu}_j)^T\Sigma_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_j)}{2}\right)\frac{d\boldsymbol{x}}{|\mathcal{A}|} - (\mathrm{E}[I_v])^2 \\
&= \sum_{\substack{i,j=1\\ i\neq j}}^{K} c_{i,j} \exp\left(\frac{\boldsymbol{b}_{i,j}^T H_{i,j}^{-1}\boldsymbol{b}_{i,j} - \boldsymbol{\mu}_i^T\Sigma_i^{-1}\boldsymbol{\mu}_i - \boldsymbol{\mu}_j^T\Sigma_j^{-1}\boldsymbol{\mu}_j}{2}\right) \\
&\qquad \cdot \int_{\boldsymbol{x}\in\mathbb{R}^2} \frac{1}{2\pi|H_{i,j}^{-1}|^{1/2}} \exp\left(-\frac{(\boldsymbol{x}-H_{i,j}^{-1}\boldsymbol{b}_{i,j})^T H_{i,j}(\boldsymbol{x}-H_{i,j}^{-1}\boldsymbol{b}_{i,j})}{2}\right) d\boldsymbol{x} \\
&\qquad + \sum_{i=1}^{K} \frac{\pi_i^2}{4\pi|\Sigma_i|^{1/2}|\mathcal{A}|} \int_{\boldsymbol{x}\in\mathbb{R}^2} \frac{1}{\pi|\Sigma_i|^{1/2}} \exp\left(-(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right) d\boldsymbol{x} \\
&\qquad - (\mathrm{E}[I_v])^2 \\
&= \sum_{\substack{i,j=1\\ i\neq j}}^{K} c_{i,j} \exp\left(\frac{\boldsymbol{b}_{i,j}^T H_{i,j}^{-1}\boldsymbol{b}_{i,j} - \boldsymbol{\mu}_i^T\Sigma_i^{-1}\boldsymbol{\mu}_i - \boldsymbol{\mu}_j^T\Sigma_j^{-1}\boldsymbol{\mu}_j}{2}\right) \\
&\qquad + \sum_{i=1}^{K} \frac{\pi_i^2}{4\pi|\Sigma_i|^{1/2}|\mathcal{A}|} - \left(\frac{1}{|\mathcal{A}|}\right)^2
\end{aligned}
$$

ただし，$H_{i,j}$, $\boldsymbol{b}_{i,j}$, $c_{i,j}$ はそれぞれ式 (3.4), (3.5), (3.6) で表される．上式の 2 番目の変形では，対象領域 $\mathcal{A}$ を全平面 $\mathbb{R}^2$ に置き換える近似を行っている．さらに，最後の変形では密度関数の全平面にわたる積分が 1 になることを利用している．

C.2 ノンパラメトリック密度推定法

ガウスカーネル密度推定法によって得られる端末密度分布 $I_v(\boldsymbol{x})$ $(\boldsymbol{x} \in \mathcal{A})$ の分散 $\sigma^2[I_v]$ を導出し，式（3.8）で与えられることを示す．$\boldsymbol{x} = (x(1), x(2)), \boldsymbol{y}_i = (y_i(1), y_i(2))$ $(\boldsymbol{y}_i \in \mathcal{L}_v,\ i = 1, 2, \ldots, |\mathcal{L}_v|)$ とし，$f_i(\boldsymbol{x})$ を次式で表す．

$$f_i(\boldsymbol{x}) = \frac{1}{2\pi h^2} \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{y}_i)^T(\boldsymbol{x}-\boldsymbol{y}_i)}{2h^2}\right)$$

$I_v(\boldsymbol{x})$ の対象領域 $\mathcal{A}$ 内における分散 $\sigma^2[I_v]$ は次式で与えられる．

$$\begin{aligned}
\sigma^2[I_v] &= \int_{\boldsymbol{x}\in\mathcal{A}} I_v^2(\boldsymbol{x}) \frac{dx(1)dx(2)}{|\mathcal{A}|} - (\mathrm{E}[I_v])^2 \\
&= \left(\frac{1}{|\mathcal{L}_v|}\right)^2 \left(\sum_{i=1}^{|\mathcal{L}_v|} \int_{\boldsymbol{x}\in\mathcal{A}} f_i^2(\boldsymbol{x}) \frac{dx(1)dx(2)}{|\mathcal{A}|}\right. \\
&\qquad \left. + \sum_{\substack{i,j=1 \\ i\neq j}}^{|\mathcal{L}_v|} \int_{\boldsymbol{x}\in\mathcal{A}} f_i(\boldsymbol{x}) f_j(\boldsymbol{x}) \frac{dx(1)dx(2)}{|\mathcal{A}|}\right) - (\mathrm{E}[I_v])^2 \qquad \text{(C.1)}
\end{aligned}$$

式 (C.1) の 1 番目の積分は次式で与えられる．

$$\begin{aligned}
\int_{\boldsymbol{x}\in\mathcal{A}} f_i^2(\boldsymbol{x}) \frac{dx(1)dx(2)}{|\mathcal{A}|} &\approx \int_{\boldsymbol{x}\in\mathbb{R}^2} f_i^2(\boldsymbol{x}) \frac{dx(1)dx(2)}{|\mathcal{A}|} \\
&= \int_{\boldsymbol{x}\in\mathbb{R}^2} \frac{1}{2\pi h^2} \exp\left(-\frac{(x(1)-y_i(1))^2}{h^2}\right) \\
&\qquad \cdot \frac{1}{2\pi h^2} \exp\left(-\frac{(x(2)-y_i(2))^2}{h^2}\right) \frac{dx(1)dx(2)}{|\mathcal{A}|} \\
&= \frac{1}{4\pi h^2} \int_{\boldsymbol{x}\in\mathbb{R}^2} \frac{1}{\sqrt{\pi h^2}} \exp\left(-\frac{(x(1)-y_i(1))^2}{h^2}\right) \\
&\qquad \cdot \frac{1}{\sqrt{\pi h^2}} \exp\left(-\frac{(x(2)-y_i(2))^2}{h^2}\right) \frac{dx(1)dx(2)}{|\mathcal{A}|} \\
&= \frac{1}{4\pi h^2 |\mathcal{A}|} \qquad \text{(C.2)}
\end{aligned}$$

また，式（C.1）の２番目の積分は次式で与えられる．

$$
\begin{aligned}
&\int_{\boldsymbol{x}\in\mathcal{A}} f_i(\boldsymbol{x})f_j(\boldsymbol{x})\frac{dx(1)dx(2)}{|\mathcal{A}|} \\
&\quad\approx \int_{\boldsymbol{x}\in\mathbb{R}^2} f_i(\boldsymbol{x})f_j(\boldsymbol{x})\frac{dx(1)dx(2)}{|\mathcal{A}|} \\
&\quad= \frac{1}{2\pi h^2}\int_{\boldsymbol{x}\in\mathbb{R}^2}\frac{1}{\sqrt{2\pi h^2}}\exp\left(-\frac{(x(1)-y_i(1))^2}{2h^2}\right)\exp\left(-\frac{(x(1)-y_j(1))^2}{2h^2}\right) \\
&\qquad\cdot\frac{1}{\sqrt{2\pi h^2}}\exp\left(-\frac{(x(2)-y_i(2))^2}{2h^2}\right)\exp\left(-\frac{(x(2)-y_j(2))^2}{2h^2}\right)\frac{dx(1)dx(2)}{|\mathcal{A}|} \\
&\quad= \frac{1}{2\pi h^2}\int_{\boldsymbol{x}\in\mathbb{R}^2}\frac{1}{\sqrt{2\pi h^2}}\exp\left(-\frac{(x(1)-y_i(1))^2+(x(1)-y_j(1))^2}{2h^2}\right) \\
&\qquad\cdot\frac{1}{\sqrt{2\pi h^2}}\exp\left(-\frac{(x(2)-y_i(2))^2+(x(2)-y_j(2))^2}{2h^2}\right)\frac{dx(1)dx(2)}{|\mathcal{A}|} \qquad \text{(C.3)}
\end{aligned}
$$

ここで，

$$
(x(k)-y_i(k))^2+(x(k)-y_j(k))^2 = 2\left(x(k)-\frac{y_i(k)+y_j(k)}{2}\right)^2+\frac{1}{2}(y_i(k)-y_j(k))^2
$$

に注意すると，次式を得られる．

$$
\begin{aligned}
&\frac{1}{\sqrt{2\pi h^2}}\exp\left(-\frac{(x(k)-y_i(k))^2+(x(k)-y_j(k))^2}{2h^2}\right) \\
&\quad= \frac{1}{\sqrt{2}}\exp\left(\frac{(y_i(k)-y_j(k))^2}{4h^2}\right)\frac{1}{\sqrt{\pi h^2}}\exp\left(-\frac{\left(x(k)-\frac{y_i(k)+y_j(k)}{2}\right)^2}{h^2}\right)
\end{aligned}
$$

よって，式 (C.3) は次式のように変形できる．

$$
\begin{aligned}
&\int_{\boldsymbol{x}\in\mathbb{R}^2} f_i(\boldsymbol{x})f_j(\boldsymbol{x})\frac{dx(1)dx(2)}{|\mathcal{A}|} \\
&\quad= \frac{1}{2\pi h^2}\cdot\frac{1}{\sqrt{2}}\exp\left(\frac{(y_i(1)-y_j(1))^2}{4h^2}\right)\frac{1}{\sqrt{2}}\exp\left(\frac{(y_i(2)-y_j(2))^2}{4h^2}\right) \\
&\quad= \frac{1}{4\pi h^2|\mathcal{A}|}\exp\left(\frac{\|\boldsymbol{y}_i-\boldsymbol{x}_j\|^2}{4h^2}\right) \qquad \text{(C.4)}
\end{aligned}
$$

ただし，$\|\boldsymbol{x}\|^2 = x(1)^2+x(2)^2$ である．式 (C.1), (C.2), (C.4) より，分散 $\sigma^2[I_v]$ は次式で与えられる．

$$
\begin{aligned}
\sigma^2[I_v] &= \left(\frac{1}{|\mathcal{L}_v|}\right)^2\left(\frac{|\mathcal{L}_v|}{4\pi h^2|\mathcal{A}|}+\sum_{\substack{i,j=1\\ i\neq j}}^{|\mathcal{L}_v|}\frac{1}{4\pi h^2|\mathcal{A}|}\exp\left(\frac{\|\boldsymbol{y}_i-\boldsymbol{x}_j\|^2}{4h^2}\right)\right)-(\mathrm{E}[I_v])^2 \\
&= \frac{1}{4\pi h^2|\mathcal{L}_v|^2|\mathcal{A}|}\left(|\mathcal{L}_v|+\sum_{\substack{i,j=1\\ i\neq j}}^{|\mathcal{L}_v|}\exp\left(\frac{\|\boldsymbol{y}_i-\boldsymbol{x}_j\|^2}{4h^2}\right)\right)-\left(\frac{1}{|\mathcal{A}|}\right)^2
\end{aligned}
$$

付 録D
ガウスカーネル密度推定法

3 章で述べたガウスカーネル密度推定法の遭遇地点情報リスト $\mathcal{L}_v$ の更新手順と平滑化パラメタ h の影響について述べる．

D.1　遭遇地点情報リスト $\mathcal{L}_v$ の更新手順

ガウスカーネル密度推定法はノンパラメトリック密度推定法の一種であるため，遭遇地点情報リスト $\mathcal{L}_v$ の各要素を用いて端末密度分布が与えられる．時間の経過に伴ってリストに含まれる数 $|\mathcal{L}_v|$ は単調に増加していくため，遭遇履歴情報リストの要素を削除しなければ各端末が持つ $|\mathcal{L}_v|$ が大きくなり，過剰な負荷を招く要因となる．そこで，本節では，$\mathcal{L}_v$ を更新し，古い情報を破棄する手順について述べる．

端末 $v \in \mathcal{V}$ が保持する遭遇地点情報リスト $\mathcal{L}_v = \{(w, \tau, \boldsymbol{z}) \mid w \in \mathcal{V}, \tau \in \mathbb{R}, \boldsymbol{z} \in \mathbb{R}^2\}$ は三つ組の情報 $(w, \tau, \boldsymbol{z})$ によって構成されるものとする．ただし，$w \in \mathcal{V}$ は端末の識別子，$\boldsymbol{z} \in \mathbb{R}^2$ は遭遇地点，$\tau \in \mathbb{R}$ は遭遇時刻をそれぞれ表す．図 D.1 は端末 v が別の端末に遭遇する度に $\mathcal{L}_v$ を更新する手順を示している．この手順では，$\boldsymbol{y} = (w, \tau, \boldsymbol{z}) \in \mathcal{L}_v$ に対して 関数 $f_I(\boldsymbol{y})$ と関数 $f_T(\boldsymbol{y})$ はそれぞれ $f_I(\boldsymbol{y}) = w$, $f_T(\boldsymbol{y}) = \tau$ と定義される．さらに，$\mathcal{G}(w, \mathcal{L}_v) = \{\boldsymbol{y} \mid \boldsymbol{y} \in \mathcal{L}_v, f_I(\boldsymbol{y}) = w\}$ である．端末 v が端末 u と遭遇した際には，まず，$\mathcal{L}_v$ に三つ組の情報 $(u, t_{v,u}, \boldsymbol{x}_u)$ が追加される．ただし，$t_{v,u}$ は端末 v, u が遭遇した時刻であり，$\boldsymbol{x}_u$ は端末 u の遭遇時にいる地点である．その後，$\mathcal{L}_v$ と $\mathcal{L}_u$ を一つに統合し，その統合したリストの中に同じ識別子をもつ情報が複数存在する場合には古い情報を全て削除し，最も新しい情報のみを残す．この手順によって，各端末が保持する遭遇地点情報の数を大きく減らすことができる．

D.2　平滑化パラメタ h の影響

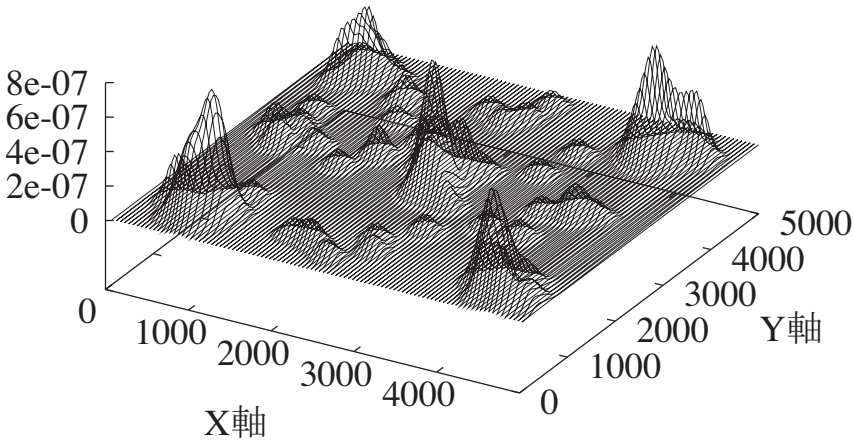
本節では，3.3 節で述べた非均質密度モデルにおいて，平滑化パラメタ h を変化させたときにどのような影響があるか議論する．図 D.2(a) と図 D.2(b) はそれぞれ $h = 100, 500$ の場合の端末密度分布の推定結果を示している．$h = 100$ の場合，

入力: v, u, $\mathcal{L}_v$, $\mathcal{L}_u$, $t_{u,v}$, $\boldsymbol{x}_u$
出力: $\mathcal{L}'_v$
$\mathcal{L}'_v := \mathcal{L}_v \cup \mathcal{L}_u \cup \{(u, t_{v,u}, \boldsymbol{x}_u)\} \backslash \mathcal{G}(u, \mathcal{L}_v) \backslash \mathcal{G}(v, \mathcal{L}_u)$
$\mathcal{J} := \{w \mid \boldsymbol{y} \in \mathcal{L}'_v, f_I(\boldsymbol{y}) = w\} \backslash \{u, v\}$
WHILE $\mathcal{J} \neq \emptyset$ DO
  $\mathcal{J}$ から任意の要素 w を選ぶ.
  IF $\mathcal{G}(w, \mathcal{L}_v) \neq \emptyset$ AND $\mathcal{G}(w, \mathcal{L}_u) \neq \emptyset$ THEN
    $\boldsymbol{y}_v := \mathcal{G}(w, \mathcal{L}_v)$
    $\boldsymbol{y}_u := \mathcal{G}(w, \mathcal{L}_u)$
    IF $f_T(\boldsymbol{y}_v) > f_T(\boldsymbol{y}_u)$ THEN
      $\mathcal{L}'_v := \mathcal{L}'_v \setminus \{\boldsymbol{y}_u\}$
    ELSE
      $\mathcal{L}'_v := \mathcal{L}'_v \setminus \{\boldsymbol{y}_v\}$
    ENDIF
  ENDIF
ENDWHILE

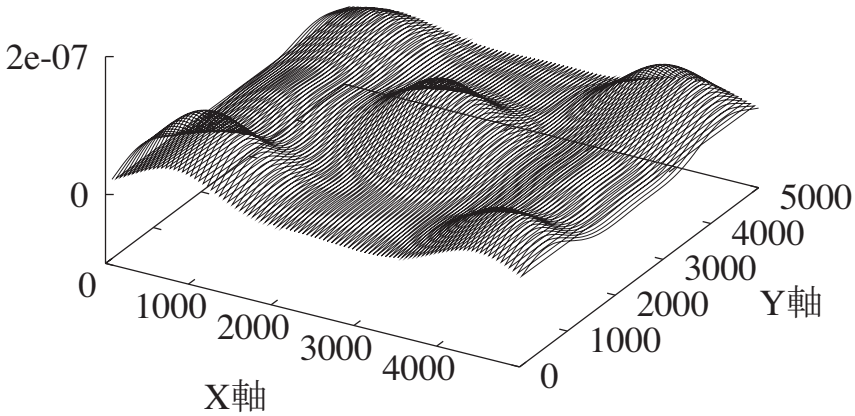
図 D.1: 遭遇履歴情報 $\mathcal{L}_v$ の更新手順

小さな山が多く存在している．一方で，$h = 500$ の場合に滑らかな大きな山のみが存在しており，小さな山は存在しない．よって，h の増加に伴って端末密度分布が滑らかになっていくことが分かる．

図 D.3(a) と図 D.3(b) はそれぞれ $h = 100, 500$ の場合に端末密度 $I_v(\boldsymbol{x})$ が d_v を上回る領域を示している．ただし，$h = 100$ の場合，$d_v = \mathrm{E}[I_v] + \sigma[I_v] \approx 1.3 \times 10^{-7}$ とし，$h = 500$ の場合，5.3×10^{-8} とした．$h = 100$ の場合，多くの領域が d_v を上回っていることが分かる．また，$h = 500$ の場合，グループ $\mathcal{V}_2$ 内の端末の移動範囲において，図 3.5(c) に示した $h = 200$ の場合と比較して d_v を上回る領域が小さくなっている．これらの観察により，h を極端に大きくしたり小さくすると密度推定の誤差が大きくなることが分かる．

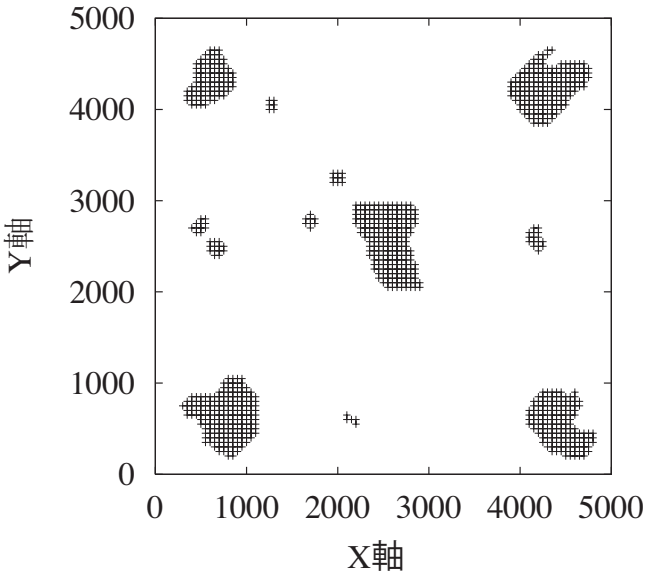


(a) $h = 100$ の場合

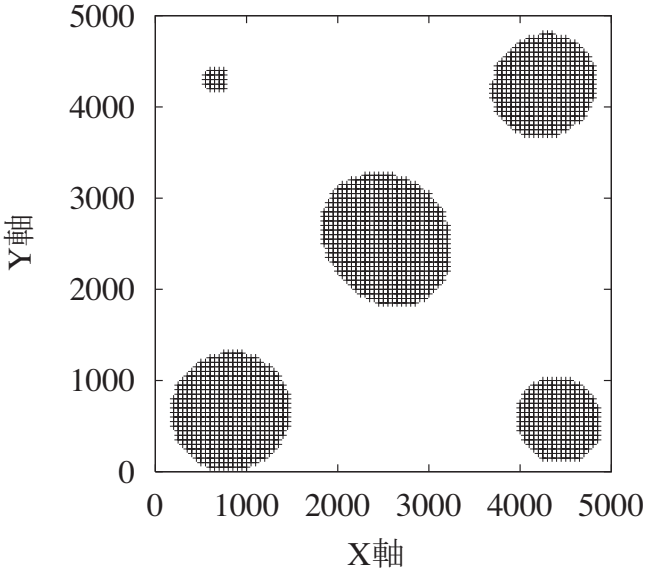


(b) $h = 500$ の場合

図 D.2: 推定された端末密度分布 $I_v(\boldsymbol{x})$ ($h = 100, 500$ の場合)



(a) $h = 100$ の場合



(b) $h = 500$ の場合

図 D.3: 閾値 d_v を上回る $I_v(\boldsymbol{x})$ の領域（$h = 100, 500$ の場合）

参考文献

[1] M. Alresaini, M. Sathiamoorthy, B. Krishnamachari, and M. J. Neely, "Back-pressure with adaptive redundancy," *Proceedings of the 31st Annual IEEE International Conference on Computer Communications (INFOCOM '12)*, pp. 2300–2308, 2012.

[2] E. Altman, T. Başar, and F. De Pellegrini, "Optimal monotone forwarding policies in delay tolerant mobile ad-hoc networks," *Performance Evaluation*, vol. 67, no. 4, pp. 299–317, 2010.

[3] Q. Ayub, M. S. M. Zahid, S. Rashid, and A. H. Abdullah, "DF++: An adaptive buffer-aware probabilistic delegation forwarding protocol for delay tolerant network," *Cluster Computing*, vol. 17, no. 4, pp. 1465–1472, 2014.

[4] A. Balasubramanian, B. Levine, and A. Venkataramani, "Replication routing in DTNs: A resource allocation approach," *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 596–609, 2010.

[5] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[6] C. Bettstetter, "Mobility modeling in wireless networks: Categorization, smooth movement, and border effects," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 3, pp. 55–66, 2001.

[7] E. Bulut, Z. Wang, and B. K. Szymanski, "Cost-effective multiperiod spraying for routing in delay-tolerant networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 5, pp. 1530–1543, 2010.

[8] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: An approach to interplanetary Internet," *IEEE Communications Magazine*, vol. 41, no. 6, pp. 128–136, 2003.

[9] J. Burgess, G. D. Bissias, M. D. Corner, and B. N. Levine, "Surviving attacks on disruption-tolerant networks without authentication," *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '07)*, pp. 61–70, 2007.

[10] Y. Cao, and Z. Sun, "Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 654–677, 2013.

[11] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay tolerant networking architecture," Internet RFC 4838, 2007.

[12] A. Chaintreau, P. Hui, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 606–620, 2007.

[13] V. Conan, J. Leguay, and T. Friedman, "Fixed point opportunistic routing in delay tolerant networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 773–782, 2008.

[14] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '07)*, pp. 32–40, 2007.

[15] A. Dvir and A. V. Vasilakos, "Backpressure-based routing protocol for DTNs," *Proceedings of the ACM SIGCOMM 2010 conference (SIGCOMM '10)*, pp. 405–406, 2010.

[16] A. Elwhishi, P. H. Ho, S. Naik, and B. Shihada, "Contention aware mobility prediction routing for intermittently connected mobile networks," *Wireless Networks*, vol. 19, no. 8, pp. 2093–2108, 2013.

[17] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation forwarding," *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 251–260, 2008.

[18] K. Fall, "A delay-tolerant network architecture for challenged Internet," *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '03)*, pp. 27–34, 2003.

[19] K. Fall and S. Farrell, "DTN: An architectural retrospective," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 828–836, 2008.

[20] S. Farrell and V. Cahill, *Delay- and Disruption-Tolerant Networking*, Boston, Artech House, 2006.

[21] S. Gaito, E. Pagani, and G. P. Rossi, "Fine-grained tracking of human mobility in dense scenarios," *Proceedings of the 6th Annual IEEE Communications*

Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops (SECON Workshops '09), pp. 40–42, 2009.

[22] S. Gaito, E. Pagani, and G. P. Rossi, "Strangers help friends to communicate in opportunistic networks," *Computer Networks*, vol. 55, no. 2, pp. 374–385, 2011.

[23] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: A social network perspective," *Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '09)*, pp. 299–308, 2009.

[24] Y. Gong, Y. Xiong, Q. Zhang, Z. Zhang, W. Wang, and Z. Xu, "Anycast routing in delay tolerant networks," *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '06)*, pp. 1–5, 2006.

[25] R. Groenevelt, *Stochastic Models for Ad Hoc Networks*, Ph.D Thesis, INRIA, 2005.

[26] R. Groenevelt, P. Nain, and G. Koole, "The message delay in mobile ad hoc networks," *Performance Evaluation*, vol. 62, no. 1, pp. 210–228, 2005.

[27] M. Grossglauser and D.N.C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, 2002.

[28] Z. Haas and T. Small, "A new networking model for biological applications of ad hoc sensor networks," *IEEE/ACM Transactions on Networking*, vol. 14, no. 1, pp. 27–40, 2006.

[29] B. Han, P. Hui, and A. Srinivasan, "Mobile data offloading in metropolitan area networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 14, no. 4, pp. 28–30, 2010.

[30] C. Hu and B. Hsieh, "A density-aware routing scheme in delay tolerant networks," *Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC '12)*, pp. 563–568, 2012.

[31] T. K. Huang, C. K. Lee, and L. J. Chen, "Prophet+: An adaptive prophet-based routing protocol for opportunistic network," *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA'10)*, pp. 112–119, 2010.

[32] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay tolerant networks," *Proceedings of the 9th ACM International*

Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08), pp. 241–250, 2008.

[33] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, 2011.

[34] R. Jathar and A. Gupta, "Probabilistic routing using contact sequencing in delay tolerant networks," *Proceeding of the 2nd International Conference on Communication Systems and Networks (COMSNETS'10)*, pp. 1–10, 2010.

[35] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, vol. 353, pp. 153–181, 1996.

[36] E. P. C. Jones and P . A. S. Ward, "Routing strategies for delay-tolerant networks," University of Waterloo, 2006. [Online]. Available: `http://ccng.uwaterloo.ca/\~{}pasward/Publications/dtn-routing-survey.pdf`

[37] P. Juang, H. Oki, Y. Wang, M. Maronosi, L. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet," *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating System (ASPLOS '02)*, pp. 96–107, 2002.

[38] K-H. Jung, W-S. Lim, J-P. Jeong, and Y-J Suh, "A link contact duration-based routing protocol in delay-tolerant networks," *Wireless Networks*, vol. 19, no. 6, pp. 1299–1316, 2013.

[39] A. Kate, G. M. Zaverucha, and U. Hengartner, "Anonymity and security in delay tolerant networks," *Proceedings of the 3rd International Conference on Security and Privacy in Communications Networks and the Workshops (SecureComm '07)*, pp. 17–21, 2007.

[40] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," *Proceedings of the 2nd International Conference on Simulation Tools and Techniques (SIMUTools'09)*, pp. 55:1–55:10, 2009.

[41] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges," *IEEE Communication & Surveys Tutorials*, vol. 14, no. 2, pp. 607–640, 2012.

[42] A. Krifa, C. Barakat, and T. Spyropoulos, "Optimal buffer management policies for delay tolerant networks," *Proceedings of the 5th Annual IEEE Commu-*

nications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '08), pp. 260–268, 2008.

[43] V. G. Kulkarni, *Modeling and Analysis of Stochastic Systems*, London, Chapman & Hall, 1995.

[44] J. LeBrun, C. N. Chuah, D. Ghosal, and M. Zhang, "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," *Proceedings of the 61st IEEE Vehicular Technology Conference (VTC '05-Spring)*, pp. 2289–2293, 2005.

[45] Y. Li, X. Li, Q. Liu, and Z. Liu, "E-PROPHET: A novel routing protocol for intermittently connected wireless networks," *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly (IWCMC '09)*, pp. 452–456, 2009.

[46] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Computing Communication Review*, vol. 7, no. 3, pp. 19–20, 2003.

[47] J. A. B. Link, N. Viol, A. Goliath, and K. Wehrle, "Simbetage: Utilizing temporal changes in social networks for pocket switched networks," *Proceedings of the 1st ACM Workshop on User-Provided Networking: Challenges and Opportunities (U-NET '09)*, pp. 13–18, 2009.

[48] Y. Lu, W. Wang, L. Chen, Z. Zhang, and A. Huang, "Opportunistic forwarding in energy harvesting mobile delay tolerant networks," *Proceeding of 2014 IEEE International Conference on Communications (ICC '14)*, pp. 526–531, 2014.

[49] Y, Lu, W. Wang, L. Chen, Z. Zhang, and A. Huang, "Distance-based energy-efficient opportunistic forwarding in mobile delay tolerant networks," *Proceedings of 2014 IEEE International Conference on Communications (ICC '14)*, pp. 3564-3569, 2014.

[50] T. Matsuda and T. Takine, "(p, q)-epidemic routing for sparsely populated mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 5, pp. 783–793, 2008.

[51] S. Medjiah and T. Ahmed, "Orion routing protocol for delay tolerant networks," *Proceedings of 2011 IEEE International Conference on Communications (ICC '11)*, pp. 1–6, 2011.

[52] A. Mtibaa, M. May, C. Diot, and M. Ammar, "Peoplerank: Social opportunistic forwarding," *Proceedings of the 29th Conference on Computer Communications (INFOCOM '10)*, pp. 1–5, 2010.

[53] J. Mukherjee and B. Ramamurthy, "Communication technologies and architectures for space network and interplanetary Internet," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 881–897, 2013.

[54] P. Nain, D. Towsley, L. Benyuan, and L. Zhen, "Properties of random direction models," *Proceedings of the 24th Conference on Computer Communications (INFOCOM '05)*, pp. 1897–1907, 2005.

[55] S. C. Nelson, M. Bakht, and R. Kravets, "Encounter-based routing in DTNs," *Proceedings of the 28th Conference on Computer Communications (INFOCOM '09)*, pp. 846–854, 2009.

[56] E. C. R. de Oliveira and C. V. N. de Albuquerque, "Nectar: A DTN routing protocol based on neighborhood contact history," *Proceedings of the 2009 ACM Symposium on Applied Computing (SAC '09)*, 2009.

[57] J. Partan, J. Kurose, B. N. Levine, "A survey of practical issues in underwater networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, pp. 23–33, 2007.

[58] P. Pereira, A. Casaca, J. Rodrigues, V. Soares, J. Triay, and C. Cervello-Pastor, "From delay-tolerant networks to vehicular delay-tolerant networks," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 1166–1182, 2012.

[59] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: Rethinking connectivity in developing nations," *IEEE Computer*, vol. 37, no. 1, pp. 78–83, 2004.

[60] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," Internet Draft, draft-ietf-manet-aodv-13.txt, 2003.

[61] A. Picu and T. Spyropoulos, "DTN-Meteo: Forecasting the performance of dtn protocols under heterogeneous mobility," *IEEE/ACM Transactions on Networking*, 2014. Digital Object Identifier 10.1109/TNET.2014.2301376.

[62] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan, "Prioritized epidemic routing for opportunistic networks," *Proceedings of the 1st International MobiSys Workshop on Mobile Opportunistic Networking (MobiOpp '07)*, pp. 62–66, 2007.

[63] F. D. Rango, M. Tropea, G. Laratta, and S. Marano, "Hop-by-hop local flow control over interplanetary networks based on DTN architecture," *Proceedings of 2008 IEEE International Conference on Communications (ICC '08)*, pp. 1920–1924, 2008.

[64] K. Sabeetha, A. V. A. Kumar, R. S. D. Wahidabanu, and W. A. M. Othman, "Encounter based fuzzy logic routing in delay tolerant networks," *Wireless Networks*, 2014. Digital Object Identifier 10.1007/s11276-014-0780-4.

[65] G. Sandulescu and S. Nadjm-Tehrani, "Opportunistic DTN routing with window-aware adaptive replication," *Proceedings of the 4th Asian Conference on Internet Engineering (AINTEC '08)*, pp. 103–112, 2008.

[66] K. Scott and S. Burleigh, "Bundle protocol specification," Internet RFC 5050, 2007.

[67] A. Seth and S. Keshav, "Practical security for disconnected nodes," *Proceedings of the 1st IEEE ICNP Workshop on Secure Network Protocols (NPSec '05)*, pp. 31–36, 2005.

[68] Y. Shibata, N. Uchida, and N. Shiratori, "Analysis of and proposal for a disaster information network from experience of the great east Japan earthquake," *IEEE Communications Magazine*, vol. 52, no. 3, pp. 44–50, 2014.

[69] C. Singh, A. Kumar, and R. Sundaresan, "Delay and energy optimal two-hop relaying in delay tolerant networks," *Proceedings of Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '10)*, pp. 100–109, 2010.

[70] P. Sok and K. Kim, "Distance-based PRoPHET routing protocol in disruption tolerant network," *Proceedings of International Conference on ICT Convergence (ICTC '13)*, pp. 159–164, 2013.

[71] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN '05)*, pp. 252–259, 2005.

[72] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops '07)*, pp. 79–85, 2007.

[73] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient routing in intermittently connected mobile networks: The single-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 63–76, 2008.

[74] 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.

[75] T. Spyropoulos, R. N. B. Rais, T. Thierry, K. Obraczka, and A. V. Vasilakos, "Routing for disruption tolerant networks: Taxonomy and design," *Wireless Networks*, vol. 16, no. 8, pp. 2349–2370, 2010.

[76] S. Srinivasa and S. Krishnamurthy, "CREST: An opportunistic forwarding protocol based on conditional residual time," *Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'09)*, pp. 1–9, 2009.

[77] S. Symington, S. Farrell, H. Weiss, and P. Lovel, "Bundle security protocol specification" Internet RFC 6257, 2011.

[78] L. Tang, Q. Zheng, J. Liu, and X. Hong, "Smart: A selective controlled flooding routing for delay tolerant networks," *Proceedings of the 4th International Conference on Broadband Communications, Networks and Systems (BROADNETS '07)*, pp. 356–365, 2007.

[79] P. U. Tournoux, J. Leguay, F. Benbadis, J. Whitbeck, V. Conan, and M. Dias de Amorim, "Density-aware routing in highly dynamic DTNs: The rollernet case," *IEEE Transactions on Mobile Computing*, vol. 10, no. 12, pp. 1755–1768, 2011.

[80] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," *Duke Technical Report*, CS-2000-06, 2000.

[81] A. V. Vasilakos, Z. Zhang, and T. Spyropoulos, *Delay Tolerant Networks: Protocols and Applications*, New York, CRC Press, 2012.

[82] Q. Yuan, I. Cardei, and J. Wu, "Predict and relay: An efficient routing in disruption-tolerant networks," *Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '09)*, pp. 95–104, 2009.

[83] X. Wang, Y. Shu, Z. Jin, Q. Pan, and B. S. Lee, "Adaptive randomized epidemic routing for disruption tolerant networks," *Proceedings of the 5th International Conference on Mobile Ad Hoc and Sensor Networks (MSN'09)*, pp. 424–429, 2009.

[84] L. Wood and P. Holliday, "Using http for delivery in delay/disruption tolerant networks," Internet Draft, draft-wood-dtnrg-http-dtn-delivery-7.txt, 2011.

[85] Y. Zeng, K. Xiang, D. Li, and A. V. Vasilakos, "Directional routing and scheduling for green vehicular delay tolerant networks," *Wireless Networks*, vol. 19, no. 2, pp. 161–173, 2013.

[86] W. Zhao, M. Ammar, and E. Zegura *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '04)*, pp. 187–198, 2004.

[87] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," *Computer Networks*, vol. 51, no. 10, pp. 2867–2891, 2007.

[88] X. Zhang, Z. Honggang , and Gu. Yu, "Impact of source counter on DTN routing control under resource constraints," *Proceedings of the 2nd International Workshop on Mobile Opportunistic Networking (MobiOpp'10)*, pp. 41–50, 2010.

[89] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 1, pp. 24–37, 2006.

[90] contacts.Exp6.dat, [Online]. Available: `http://crawdad.cs.dartmouth.edu/meta.php?name=cambridge/haggle/imote/infocom2006`

[91] pmtr.txt, [Online]. Available: `http://crawdad.cs.dartmouth.edu/unimi/pmtr`

研究業績

雑誌論文 (査読あり)

⟨1⟩ T. Kimura, T. Matsuda, and T. Takine,
"Multi-spreader routing for sparsely populated mobile ad hoc networks," *Wireless Networks*, vol. 20, no. 1, pp. 155–175, Jan. 2014.

⟨2⟩ T. Kimura, T. Matsuda, and T. Takine,
"Location-aware store-carry-forward routing based on node density estimation," *IEICE Transactions on Communications*, vol. E98-B, no. 1, pp. 99–106, Jan. 2015.

国際会議 (査読あり)

⟨1⟩ T. Kimura, T. Matsuda, and T. Takine,
"Probabilistic store-carry-forward message delivery based on node density estimation," *Proceedings of the 11th Annual IEEE Consumer Communications and Networking Conference (CCNC '14)*, pp. 1350–1354, Las Vegas, NV, Jan. 2014.

⟨2⟩ T. Kimura, T. Matsuura, M. Sasabe, T. Matsuda, and T. Takine,
"Location-aware utility-based routing for store-carry-forward message delivery," *Proceedings of the 29th International Conference on Information Networking 2015 (ICOIN '15)*, pp. 194–199, Siem Reap, Cambodia, Jan. 2015.

研究会 (査読なし)

⟨1⟩ 木村共孝, 松田崇弘, 滝根哲哉,
"疎密度モバイルアドホックネットワークにおける Multi-Spreader ルーチング方式," 信学技報, vol. 107, No. 525, IN2007–183, pp. 145–150, 2008 年 3 月.

⟨2⟩ 木村共孝, 滝根哲哉,
"疎密度モバイルアドホック網におけるバッファ管理方式," 信学技報, vol. 108, no. 204, IN2008-53, pp. 57–62, 2008 年 9 月.

⟨3⟩ 木村共孝, 滝根哲哉,
“疎密度モバイルアドホック網におけるマルチキャスト通信,” 信学技報, vol. 108, no. 458, IN2008-179, pp. 279–284, 2009 年 3 月.

⟨4⟩ 木村共孝, 滝根哲哉,
“蓄積運搬転送型ルーチングにおける中継先端末選択法,”
信学技報, vol. 111, no. 469, IN2011-201, pp. 383–388, 2012 年 3 月.

⟨5⟩ 木村共孝, 松田崇弘, 滝根哲哉,
“端末密度推定を用いた蓄積運搬転送型ルーチング,”
電子情報通信学会ソサイエティ大会, B-6-71, 2013 年 9 月.

⟨6⟩ 木村共孝, 松田崇弘, 滝根哲哉,
“端末密度推定を用いた確率的蓄積運搬転送型ルーチング,” 信学技報, vol. 113, no. 208, CQ2013-34, pp. 35–40, 2013 年 9 月.

⟨7⟩ 木村共孝, 松田崇弘, 滝根哲哉,
“蓄積運搬転送型ルーチングのための端末密度推定法,” 信学技報, vol. 113, no. 471, CQ2013-105, pp. 95–100, 2014 年 3 月.

⟨8⟩ 木村共孝, 松浦翼, 笹部昌弘, 松田崇弘, 滝根哲哉,
“ネットワーク環境に適応した蓄積運搬転送型ルーチング,”
電子情報通信学会総合大会, DS-2-2, 2014 年 3 月.

⟨9⟩ 木村共孝, 松浦翼, 笹部昌弘, 松田崇弘, 滝根哲哉,
“遭遇端末と遭遇地点の特性に基づく蓄積運搬転送型ルーチング,” 信学技報, vol. 114, no. 131, CQ2014-15, pp. 1–6, 2014 年 7 月.

⟨10⟩ 城之内達郎, 木村共孝, 松田崇弘, 滝根哲哉,
“端末密度を用いた蓄積運搬転送型ルーチングの転送確率制御,”
電子情報通信学会ソサイエティ大会, B-11-34, 2014 年 9 月.

⟨11⟩ 城之内達郎, 木村共孝, 松田崇弘, 滝根哲哉,
“蓄積運搬転送型ルーチングにおける端末密度に基づく適応的確率制御,”
信学技報, vol. 114, no. 298, CQ2014-79, pp. 47–52, 2014 年 11 月.

受賞

⟨1⟩ Best Paper Award of the 29th International Conference on Information Networking 2015 (ICOIN ’15), January 12-14, 2015.