現実的なシミュレーションシナリオが記述可能な無線ネットワークシミュレータ MobiREAL

前田 久美子[†] 小西 一樹[†] 佐藤 和基[†]
山口 弘純[†] 安本 慶一[††] 東野 輝夫[†]

本論文では現実環境における MANET アプリケーションの性能評価のために，現実的な地理領域内での移動端末（ノード）の現実的な行動を記述しシミュレートするための手法を提案するとともに，その手法を用いたネットワークシミュレータ MobiREAL の設計，および開発について述べる．ノードの行動記述には，確率つきイベント発生モデルを採用する．提案手法により，ノードがその位置や周辺情報（障害物や近隣ノード），MANET アプリケーションから得られた情報などに基づき，目的地や移動経路，速度，移動方向を変更するといった現実的な行動パターンを記述することが可能となる．MobiREAL シミュレータは，現実的な行動に基づいたノードから構成される様々なモバイルアドホックネットワークのプロトコルやアプリケーションをシミュレートすることができる．いくつかの実験例を通し，MANET アプリケーションの開発における MobiREAL の有用性を示す．

MobiREAL Simulator Achieving Realistic Performance Evaluation of MANET

Kumiko Maeda,[†] Kazuki Konishi,[†] Kazuki Sato,[†]
Hirozumi Yamaguchi,[†] Keiichi Yasumoto[††]
and Teruo Higashino[†]

In this paper, we present a new methodology to model and simulate realistic mobility of nodes for accurately evaluating performance of MANET applications. We have designed and developed a network simulator called MobiREAL based on the proposed methodology. We adopt a probabilistic rule-based model to describe behavior of mobile nodes. The proposed model allows us to describe how mobile nodes change their destinations, routes and speeds/directions based on their positions, surroundings (obstacles and neighboring nodes), information obtained from applications, and so on. Our MobiREAL simulator can simulate various protocols and applications on mobile ad-hoc networks composed of mobile nodes with realistic movements. Through several case studies, we show that it is important to simulate MANET applications in realistic environments, and that MobiREAL can effectively be used for such a purpose.

1. はじめに

モバイルアドホックネットワーク（MANET）はユビキタス社会を実現するための有用かつ重要な基盤となることが予想されている．MANET のシミュレーションには，たとえば Random Way Point（RWP）モデル[1] のような単純なノード移動モデル（モビリティモデル）がよく用いられる．しかしながら，モビリティモデルが MANET プロトコルやアプリケーションの性能に影響を与えることは以前から知られている[2)~4)]．したがって，実際の環境における MANET プロトコルやアプリケーションの性能を正しく評価するためには，より現実的なモビリティモデルのもとでのシミュレーションを行う必要がある．また，シミュレーションにかかるコストの観点からより現実的でかつ十分にシンプルなモビリティモデルが求められている．そのような背景から，近年，現実的なモビリティモデルに関する研究がさかんに行われている[5)~7)]．

たとえば，文献 5) では建物（障害物）による電波の遮断を再現し，建物を回避するモビリティモデルを

† 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
†† 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology

導入している．また，文献 6) ではシミュレーション領域を住宅地やビジネス街など特徴に応じていくつかのゾーンに区分し，ゾーン単位でのノード種別ごとの密度の変遷を，既存の交通計画手法を用いて推定している．これらの研究では現実に近いノードの動きをある程度再現できるが，ノードが周辺環境やネットワークシステムの実行結果に依存してその行動を動的に変化させる様子をモデル化する枠組みは提供されていない．

本論文では，ノードが自身の周辺環境やネットワークシステムの振舞いに応じて行動を動的に変化させる挙動を記述可能な確率つきイベント発生モデル（Condition Probability Event Model，以下 CPE モデル）を提案する．さらに CPE モデルに基づいたノード移動のシミュレートと，その動きに基づいたネットワークシステムのシミュレートが可能なネットワークシミュレータ MobiREAL の設計および実装を行う．MobiREAL を用いてシミュレーションを行うことにより，ネットワークシステムとノードの行動が相互に影響を与える様子をシミュレートでき，現実により近い環境でシステムの性能を評価できると考えられる．

2. MobiREAL の構成

図 1 に MobiREAL シミュレータの構成図を示す．MobiREAL シミュレータはネットワークシステムをシミュレートするネットワークシミュレータと，ノードの行動をシミュレートする行動シミュレータからなる．汎用性を高めるため，行動をシミュレートする行動シミュレータはネットワークシミュレータから独立したプログラムとしている．これにより，ns-2 や GloMoSim などの様々なネットワークシミュレータに対しても，少しの改造を加えることで行動シミュレータが利用可能となり，現実的なノード移動を前提としたネットワークシミュレーション機能を提供することができる．

MobiREAL では，ネットワークシミュレータと行動シミュレータを独立したプログラムとしたため，以下の設計方針を採用する．まず，ノードの位置や速度はそれぞれ独立に管理させる．また，シミュレーション進行をシミュレーションクロックで同期させ，ノード行動のシミュレーション実行結果（ノードの位置や移動方向）を行動シミュレータからネットワークシミュレータに対し随時通知させる．ネットワークシミュレータはこれを受けて自身のノードの位置情報を更新し，シミュレーションを続行する．逆に，ネットワークシステムのシミュレーション実行結果（アプリケーションからユーザへのデータ出力）をネットワークシミュレータから行動シミュレータに対し随時通知させる．行動シミュレータはこれを受けて自身のノードの振舞いに反映させ，シミュレーションを続行する．

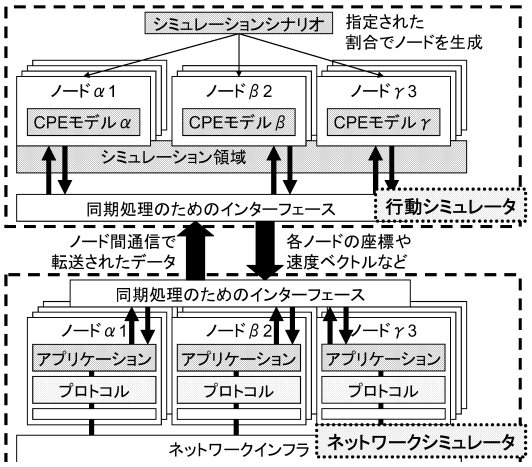


図 1 MobiREAL シミュレータ概略図
Fig. 1 MobiREAL simulator overview.

図 1 における色付きのコンポーネントはシミュレータのユーザによりその内容が指定されることを示す．行動シミュレータの入力には，シミュレーション領域，シミュレーションシナリオ，CPE 記述の 3 つがあり，シミュレーション領域では建物や道路などの情報を指定する．CPE 記述では，ノードの行動種別（たとえば通勤者，買い物客など）ごとに，周囲の環境やアプリケーションから得た情報に応じた動的な行動変化規則などを変数を利用した汎用的な形式で記述する（図 1 の CPE モデル α，β，γ に相当）．シミュレーションシナリオでは，ノードがたどる経路などの具体的な変数値および，ノードの生成割合を指定し，それに従って各 CPE 記述に基づいて行動するノードインスタンスを生成する．ネットワークシミュレータに対しては，単体でネットワークシミュレータを利用するときと同様に，ネットワークシステム（アプリケーションやトランスポート層/ネットワーク層プロトコルなど）の記述を指定する．アプリケーションには，CPE 記述で指定され行動シミュレータから通知される，ユーザノードからのデータ入力に対する処理も記述する．

以上のことから，ネットワークシミュレータには行動シミュレータと通信して情報交換を行う機能，行動シミュレータから受け取ったノード位置，速度を自身のノードに反映させる機能，シミュレータユーザが行動シミュレータへの出力や行動シミュレータからの入力に対する振舞いを記述するためのインタフェース，の 3 つを実装する必要がある．

3. 行動シミュレータとノードのモデル化

本章では，現実世界の環境やノードの行動をモデル化する方法を提案し，それに基づく行動シミュレータの設計と実装について述べる．

3.1 シミュレーション領域のモデル化

建物，広場などの自由に通行できない閉空間は閉多角形（たとえば**図 2** における色付きの領域）で指定し，進入可能な閉空間については図 2 の E や J のようにポイントを閉多角形の辺上に設定することで出入り口を指定する．ノードは指定された出入り口を除いては，閉空間の外側から内側へ，内側から外側へ移動することはできない．その他の領域は道路などの自由移動領域とする．ただし交差点や領域の出入り口に設定されたポイントを線分で結んだ仮想グラフを合わせて指定することで，道路の論理的な構造情報を与え，目的地の指定や経路計算を簡単化している．

3.2 ノードの行動のモデル化

一般にノード（携帯端末ユーザ）は自身の周辺の情報（障害物や近隣ノードの行動）や自身の携帯端末上で動作するネットワークシステムの出力に基づいて目的地などを動的に変更する場合がある．たとえば混雑している目的地は後で訪問する，あるいは近隣の店舗を検索するナビゲーションシステムに購入したい商品の情報を入力し，システムが提示した店舗に向かう，といった行動変更が考えられる．また，ノードは目的地や経路，到着予定時刻や滞在時間などをある程度計画して行動することが多い一方，行動が確率的である場合も多い．

以上のような観点から，本研究では，ノードの動的な行動変化規則を変数を用いて記述する確率つきイベント発生モデル（CPE モデル）を提案し，さらにその CPE モデルを用いたノードの行動のモデル化手法を提案する．

CPE モデルは，ノードのプリミティブな行動とその実行条件および実行確率の組である “ルール” の並びと，任意に定義可能な内部変数の集合，および外部変数の集合として定義する．外部変数は CPE モデルの外部から更新，あるいは参照可能な変数を表し，シミュレーション時刻 T，周辺ノードの位置情報 E，ネットワークシステムからのデータ出力 AO，ネットワークシステムへのデータ入力 AI，自身の現在位置 P および速度ベクトル V の 6 つが存在する．実行条件には外部変数や内部変数を用いた論理式を，行動にはそれらの変数への代入文の集合を指定する．確率には 0 から 1 までの定数，もしくは 0 から 1 までの値をとる数式を指定する．なお，よく用いられる式はライブラリ関数として定義し，提供している．CPE モデルは実行可能なモデルであり，ルールの並びを先頭から検索し，実行条件および実行確率が満足される最初のルールの行動を実行する操作を繰り返すことで実行される．

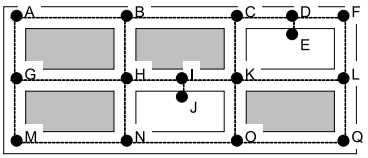


図 2 シミュレーション領域のモデル化
Fig. 2 Modeling of simulation field.

以下，IEEE802.11 などの近距離無線通信デバイスを装備した情報端末を保持する歩行者がいくつかの店舗を買い物などの目的で訪れ，店舗に関して入手した情報（たとえばセール情報）を，移動中に遭遇した他の歩行者に配布することで有用な情報の共有を行う MANET 上のネットワークシステムを考える．

このような歩行者の CPE 記述例を**図 3** に示す．この CPE 記述には大きく分けて，歩行者が目的地を巡回する行動，目的地の到着予定時刻に遅れそうな場合の行動，配布したい店舗情報を入力する行動，店舗に関する情報を周囲の歩行者から受け取った場合の行動，混雑地帯の迂回，の 5 つの行動に関する記述が含まれている．

歩行者が到着予定時刻を考えながら目的地を巡回する行動を実現するため，この CPE 記述では，目的地情報（目的地の位置を示すポイント名 p，現在地から目的地までの経路を表すポイント列 r，到着予定時刻 t，滞在予定時間 s）を保持する構造体を内部変数として利用しており，現在の目的地情報を変数 dst に，その後訪れる予定の目的地群を訪問順に，目的地情報のリストとして $Dlist$ に格納している．目的地での滞在を終え，次の目的地へと向かうときは $Dlist$ から先頭の要素を取り出し，新たな目的地として dst に代入する（ルール E7，E9）．目的地情報 dst は，ルール E7，E8，E9 のように目的地での滞在を終了する時刻を判定するのに用いられるほか，ルール E2，E3，E4 のように目的地の到着予定時刻に遅れそうな場合の行動の記述にも利用されている．

歩行者は目的地における滞在予定時刻を過ぎたとき，0.8 の確率でその目的地を出発し（ルール E7），ルール E7 が選択されなかったときは滞在を延長する（ルール E8）．滞在を延長した場合は，滞在を延長した店舗の位置 $dst.p$ を周辺の歩行者に散布（ブロードキャス

node

[CPE Description]

	Condition	Prob.	Action
E1	crowded(P,dst.r,E) 周辺ノードの位置情報 E から進行方向（現在地 P，目的地までの経路 dst.r より計算）のノード密度が高い（混雑している）と判断	N(6min, 2min)	dst.r=detour_path(E,P,dst.p); 混雑している道路を除いたグラフで P から現在の目的地 dst.p までの経路を再計算
E2	late(T,P,V,dsr.r,dst.t) 目的地の到着予想時刻（現在時刻 T, 現在地 P, 速度ベクトル V, 目的地までの経路 dst.r より計算）が到着予定時刻よりも少し遅い	1.00	V=fast(); 早歩き程度の速度になるよう V を再計算
E3	miss(T,P,V,dst.r,dst.t) 目的地の到着予想時刻が到着予定時刻よりもかなり遅い	0.20	dst.t+=10min; V=fast(); 到着予定時刻 dst.t を 10 分遅らせ，早歩き程度の速度になるよう V を再計算
E4	miss(T,P,V,dst.r,dst.t) 目的地の到着予想時刻が到着予定時刻よりもかなり遅い	1.00	dst=pop(Dlist); dst.r=shortest_path(P,dst.p); 目的地リストから次の目的地を取り出し（現在の目的地へ行くのはあきらめ），目的地までの経路を設定
E5	receive_from_net(AO,new_dst) 周辺の歩行者から店舗の位置を取得した	0.30	dst.p=new_dst; dst.r=shortest_path(P,dst.p); dst.p を取得した店舗 new_dst に変更，目的地への経路を設定
E6	¬stay ∧ ¬overstay ∧ reach(P,dst.p) 滞在（延長）状態 (over)stay ではなく，P と dst.p が一致（目的地に到着）	1.00	sst=T; stay=true; V=stop(); 到着した時刻を sst に記録，その場で停止，滞在状態 stay を設定
E7	stay ∧ (T−sst)≥dst.s 滞在状態でかつ到着時刻 sst から滞在予定時間 dst.s 以上経過	0.80	stay=false;dst=pop(Dlist);dst.r=shortest_path(P,dst.p);V=norm(); 滞在状態を解除，次の目的地を dst に取り出し，目的地までの経路を計算，V を通常の歩行速度に設定（移動を開始）
E8	stay ∧ (T−sst)≥dst.s 滞在状態でかつ到着時刻 sst から滞在予定時間 dst.s 以上経過	1.00	stay=false; overstay=true; dst.s+=rand(5min,10min); dst.s を 5 分〜10 分遅らせ，滞在状態を解除し，滞在延長状態 overstay を設定

E9	overstay ∧ (T−sst)≥ dst.s
	
	滞在延長状態でかつ到着時刻 sst から延長した滞在予定時間 dst.s 以上経過
		1.00	send_to_net(AI,dst.p); overstay=false; dst=pop(Dlist); V=norm();
			dst.r=shortest_path(P,dst.p);
			周辺の歩行者に滞在中の店舗の位置 dst.p を散布，滞在延長状態を解除，E7 と同様に目的地の設定を行い移動を開始

[Initial Values of Internal Variables]

stay = false
overstay = false
sst = 0

dst	Dlist[0]	Dlist[1]	Dlist[2]...
.p = J .r = {H, I, J} .t = 3:10 .s = 30 min	.p = E .r = {J, I, K, C, D, E} .t = 3:50 .s = 20 min	.p = F .r = {E, D, F} .t = 4:30 .s = 0 min	

AO, T, E ↑ ↓ P, V, AI

図 3 CPE モデルによる歩行者の行動記述例
Fig. 3 Example behavior description of a pedestrian in CPE model.

ト）する（ルール E9）．また，周辺の歩行者から店舗の位置 *new_dst* を入手した場合，0.3 の確率で現在の目的地をその店舗に変更する（ルール E5）．

ルール E1 では混雑地帯を迂回する行動を記述しており，条件を満たしてからルールが実行されるまでの時間は平均 6 分，標準偏差 2 分の正規分布 $N(6min, 2min)$ に従う．ただし，ルールが実行されるべき時刻までに条件を満たさなくなるとルールは実行されない．確率関数には任意の関数を定義し利用することが可能であり，ある条件が成り立つ状態での行動を起こすまでの時間的なばらつきや，行動を起こす頻度を指定することができる．

一般に歩行者は接近してくる歩行者や小さな障害物などを回避しながら歩行することから一時的に移動方向や移動速度を変えることがある．行動シミュレータではこのような衝突回避を実現するため，各ノードについて，進行方向に存在するノードの座標や速度をみて斜めに方向転換したり減速，停止したりするなど，CPE 記述で定められた値を基準として速度ベクトル V を走査周期ごとに自動で調整する機能を提供している．速度ベクトル調整の具体的な手法は文献 8) を参考にした．この衝突回避により，たとえば前方に歩行者が大勢いるため思うように移動することができない状態，すなわち混雑を再現することが可能である．

3.3 シミュレーションシナリオ

シミュレーションシナリオではシミュレーション実行時間などを設定するとともに，指定された生成タイミングに従い，各 CPE 記述に対し，目的地リストなどの変数の初期値が設定されたノードインスタンスの生成を行う．**図 4** にシミュレーションシナリオ記述例を示す．

我々はシミュレーションシナリオのパラメータを現実に即して定めることができるよう，ノード密度の観測値に基づき，ノードが目的地を巡回するような歩

```
SIMULATION_TIME = 600*sec;   //シミュレーション時間 = 600 秒
CPE_DELTA = 0.2*sec;         //CPE 走査間隔 = 0.2 秒
INTERACTION_DELTA = 1.0*sec; //インタラクション間隔 = 1 秒
    //ファイル customer.cpe に記述されている CPE を読み込む
CPE customer = readCPE("./customer.cpe");
    //10 秒間隔でノードの生成を行う
for(t=0; t<SIMULATION_TIME; t+=10*sec){
      //以下 CPE に用いる変数の初期値の設定を行う
      //速度 V を 1.0\,m/秒〜1.8\,m/秒の間でランダムで設定
  V=random(1.0*m_per_sec, 1.8*m_per_sec);
      //初期位置 P を交差点 A か交差点 Q に 3:7 の割合で設定
  if(prob(30)) P = point(A);
  else         P = point(Q);
      //一つ目の目的地の設定
  dst.p = point(E);
      //req_time は移動にかかる時間を予測する関数
  dst.t = t + req_time(point(P),point(E));
  dst.s = 30*min;
  Dlist.push_back(dst);
      //二つ目の目的地の設定
  dst.p = J;
  dst.t = t + req_time(point(E),point(J));
  dst.s = 20*min;
  Dlist.push_back(dst);
      //customer.cpe の CPE 記述に従うノードを t 秒に生成する.
  gen_node(t, customer.gen(V,P,Dlist));
}
    //以下繰り返し
for(t=0; t<SIMULATION_TIME; t+=15*sec){
    …
}
```

図 4 シミュレーションシナリオ
Fig. 4 Simulation scenario.

行者の流れを導出する手法，Urban Pedestrian Flow（UPF）を提案している[4]．UPF では歩行者の予想経路，およびいくつかの道路における歩行者の平均密度を入力とし，各経路をたどる歩行者の単位時間あたりの生成頻度を，線形計画法を用いて密度の観測値との誤差が最小になるように求めることで，現実の環境に近い歩行者の密度と流れの再現を可能としている．提供するシナリオ生成支援ツールにより，GUI を通して入力を与え，提案手法に基づいて導出されたノードの生成頻度を用い，シナリオを自動で作成することが可能である．

4. MobiREAL の実装

4.1 シミュレータ間の同期

MobiREAL の行動シミュレータとネットワークシミュレータは，2 章で述べたとおり，互いのシミュレーション結果を交換する処理を行う必要がある．我々の実装では，実行効率や実装の容易さなどの観点から t シミュレーション時間間隔で定期的に交換処理を行うことにした．通信には TCP を用いており，これにより，2 つのシミュレータを異なった計算機で実行し負荷分散を行うことができる．各シミュレータは t 時間分のシミュレーションを行った後，シミュレーション結果を送信し，相手からシミュレーション結果を受信するまで待機する．結果を受信すれば，あるいはすでに受信していれば受信した内容を自身の状態に反映させ，次の t 時間分のシミュレーションを開始する．なお，t を可能な限り小さい値に設定すれば，行動シミュレータが決定するノードの位置や速度などがネットワークシミュレータ側にほぼ完全に反映された正確なシミュレーションが可能であるが，その分通信頻度が増加し，シミュレーション実行時間を多く要する．t の値の決定は，シミュレーションに求める精度と実行時間とのトレードオフに基づいて行う．t の値とシミュレーションの実行時間に対する評価は 5.3 節で述べる．

4.2 ネットワークシミュレータ

MobiREAL のネットワークシミュレータにはネットワークシステムをシミュレートする基本機能に加え，2 章で述べたように，ノードの位置や速度，アプリケーション入出力を行動シミュレータと交換し，自身に反映させる機能が必要となる．我々は，米ジョージア工科大学で開発された GTNetS[9] に上記機能などを追加，修正することで MobiREAL のネットワークシミュレータを開発している．GTNetS はスケーラビリティ向上を主眼として設計されており，標準で有線ネットワークを対象としたシミュレーションの並列実行環境を備え，無線ネットワークシミュレーションに関しても AODV や DSR などのルーティングプロトコルや IEEE 802.11 などの標準的な物理層および MAC 層シミュレーションを提供している．

一般に，GTNetS を含む多くのネットワークシミュレータは，シミュレーション実行中の新規ノードの追加や既存ノードの削除を想定せずに実装されている．これに対し我々は，GTNetS のノード管理プロセスの修正を行うことにより，動的なノードの生成や削除を実現している．また，シミュレーションの現実性を向上させるため，障害物の影響を考慮した電波伝搬を計算できるように物理層シミュレーションモジュールの拡張を行っている．障害物の座標などの情報はシミュレーション開始時に，同期を行うための通信チャネルを経由して行動シミュレータから受け取るようにしている．

4.3 アニメータ

シミュレーション結果の視覚化はネットワークシミュレータにとってきわめて重要である．特に，ノードの動きを可視化することは，ネットワークシステムがノードの行動に与える影響を確認するために不可欠である．そこで MobiREAL シミュレータでは，シミュレーション結果の視覚化を行うアニメータを提供している．

[CPE Description]

[共通ルール] 各実験共通でノードは次の 2 つのルールを利用する（内部変数 dst, sst, stay は図 3 と共通とする）

Condition	Prob.	Action
¬stay ∧ reach(P,dst.p) 滞在状態ではなく，P と dst.p が一致（目的地に到着）	1.0	sst=T; stay=true; V=stop(); 到着した時刻を sst に記録，その場で停止，滞在状態 stay を設定
stay ∧ (T−sst)≥dst.s 滞在状態でかつ到着時刻 sst から滞在予定時間 dst.s 以上経過	1.0	stay=false; dst=pop(Dlist); dst.r=shortest_path(P,dst.p); V=norm(); 滞在状態を解除，次の目的地を dst に取り出し，目的地までの経路を計算，V を通常の歩行速度に設定（移動を開始）

[実験 1 DSR プロトコル] 通信を行うノード（0 番，1 番ノード）のうち 0 番ノードのみ以下のルールを先頭に追加する

Condition	Prob.	Action
T==240*sec シミュレーション時間が 240 秒である	1.0	send_to_net(AI,“connect 1”); 1 番ノードと通信を開始する

[実験 2 情報配布] 以下のルールを先頭に追加する（内部変数 receive の初期値は false に設定する）

Condition	Prob.	Action
¬receive ∧ receive_from_net(AO) 受信済みでなく，システムから情報を得た	0.8	dst.p=shopA; dst.r=shortest_path(P,dst.p); receive=true; 目的地を店 A に変更，経路を計算，受信済みフラグを立てる

図 5 CPE モデルによる行動記述例
Fig. 5 Example behavior description in CPE model.

アニメータは Windows 上で動作し，ノードやリンク，無線到達半径，パケット伝搬の様子などをアニメーションで表示できる．これらは項目ごとに表示か非表示かの選択が可能であり，パケットは種類（データパケットや制御パケットなど）別に色表示できる．またノードの色表示はシミュレーション対象となるネットワークシステムのプログラム内で自由に指定可能である．たとえばシステムから特定の情報を受け取ったノードを赤で，それ以外のノードを青で表示するといったことが可能となる．アニメータ実行時の静止画像と動画像は MobiREAL のウェブページ[10] で公開している．

5. シミュレーション事例と実験結果

MANET アプリケーションに対する我々の提案手法の有用性を評価するために，典型的な MANET ルーティングプロトコル（DSR）とアプリケーション（情報配布）について，MobiREAL シミュレータを用いて性能評価を行った．

5.1 DSR プロトコル

1 つ目の実験では DSR プロトコルについて，**図 5** のように目的地を巡回する動作を記述した CPE モデルおよび 3.3 節で述べた UPF 手法を用いて生成したモビリティ（以下 UPF）を用いた場合と，Random Way Point モビリティを拡張した障害物つき RWP モビリティ（以下 RWP/ob）を用いた場合について性能を計測し，比較を行った．この実験により，モビリティモデルがネットワークシステムに与える影響を評価し，現実的なモビリティのもとでシミュレーションを行うことの必要性を示す．

UPF では 3.3 節で述べたとおり，道路構造，経路，

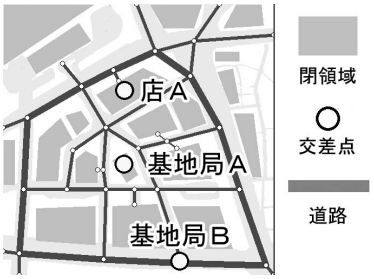
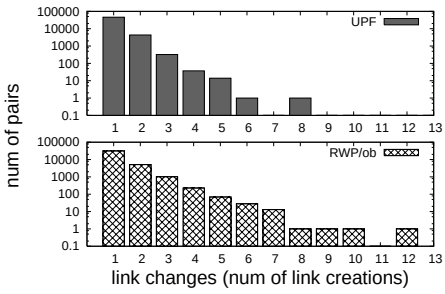


図 6 大阪駅前の地図
Fig. 6 Downtown Osaka City.

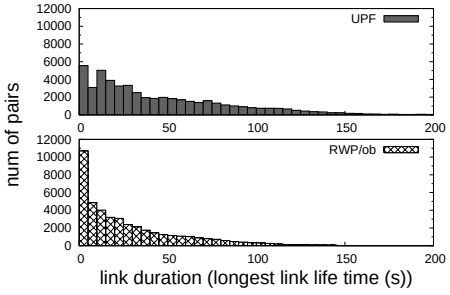
歩行者密度の観測値が与えられると，各経路の歩行者の流れが自動で導出される．シミュレーション領域は JR 大阪駅前の 500 m×500 m の領域をモデル化した．モデル化した領域とその道路構造を**図 6** に示す．歩行者密度は実際に大阪駅前で測定し，得られたデータを用いた．また経路は，主要な建物や地図の端にあたる頂点など，目的地となりうる頂点の間のすべての最短経路を計算し利用した．

RWP/ob とは，グラフとして与えられた道路上を一定速度でランダムに移動するモデルである．各ノードは，ランダムに選択された頂点で発生し，各頂点で次の目的地として隣接する頂点をランダムに選択し移動する．そして，頂点を一定数通過したら消滅する．この実験では両モビリティで同じ道路構造を用い，また，ノードの移動速度や，領域上に存在するノードの総数もほぼ同じとなるように設定した．

実験では，シミュレーション領域の外周道路（図 6 において太線で表される道路）に沿って 400 m 程度離れて移動している 2 人のアプリケーションユーザ間で通信を行った．シミュレーション時間は 720 秒，通信は 240 秒から 720 秒まで行った．通信には UDP

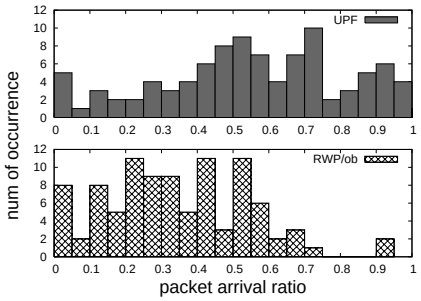


(a) リンク変更回数の分布

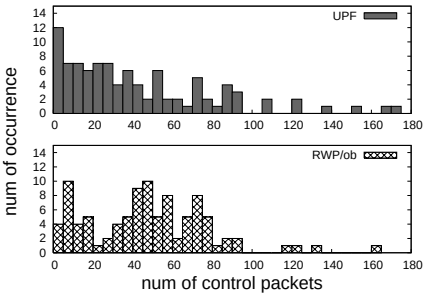


(b) リンクの最長生存時間の分布

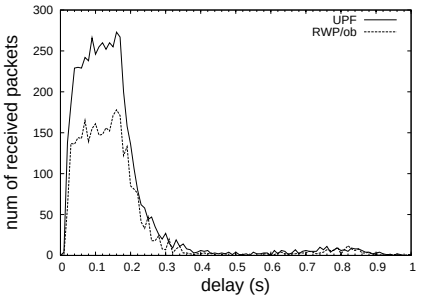
図 7 リンクの接続性に関する評価指標
Fig. 7 Link connectivity metrics.



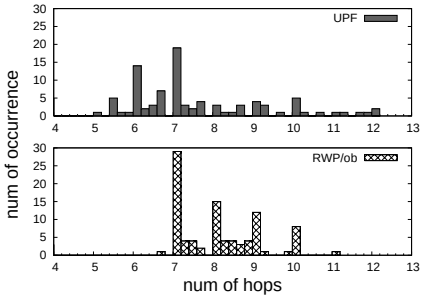
(a) パケット到着率分布（5 秒ごとの平均値）



(b) 制御パケット数分布（5 秒ごとの総発生数）



(c) パケット遅延分布



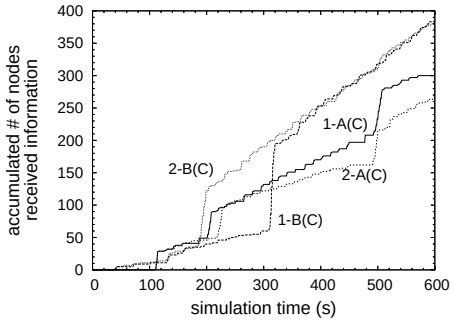
(d) ホップ数分布（5 秒ごとの平均値）

図 8 DSR の性能評価
Fig. 8 DSR performance.

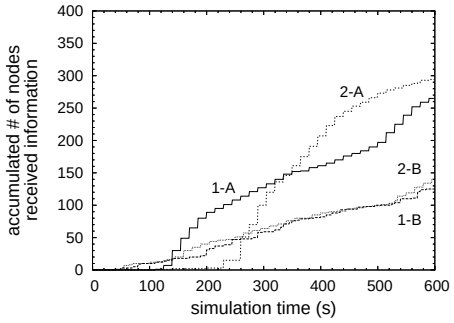
を用い，2 人のアプリケーションユーザ間で双方向，10 kbps の CBR トラフィックを生成した．MAC 層は IEEE 802.11 DCF + RTS/CTS を用い，各ノードの無線伝播距離は 100 m と設定した．総ノード数は約 2000，通信端末を所持するノードは全体の 20%とした．また，ノードの移動速度は 1.1〜1.7 m/s とした．

まず，ネットワークグラフの接続性を示す以下の 2 つの指標値，リンク変更回数（2 つのノード間でリンクが生成された数），リンクの最長生存時間（2 つのノードが互いに連続して通信可能であった時間の最大値），を計測した（これらの指標値の詳細については文献 3) 参照）．計測結果を**図 7** に示す．また，DSR の性能に関する計測を行った．その結果を**図 8** に示す．

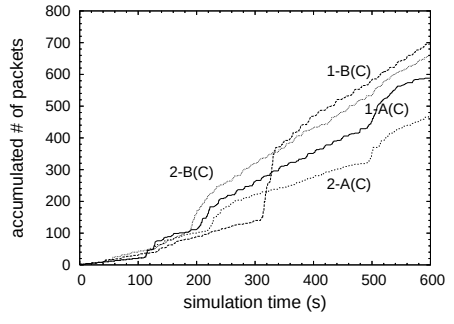
RWP/ob は UPF に比べてややリンク変更回数が多い方に偏っている．RWP/ob では各交差点でノードが方向転換する可能性が高いため，UPF よりも近隣ノードの入れ替わりが激しくなる．そのため，RWP/ob ではノードが互いの通信範囲外に移動することによるリンクの切断，再構築が UPF よりも頻繁に起こり，経路が不安定となる．このことは，図 7 (b)，図 8 (a)，(b) にも表れている．経路が不安定であればリンク持続時間は短くなり，また切断が頻繁に起こるためパケット



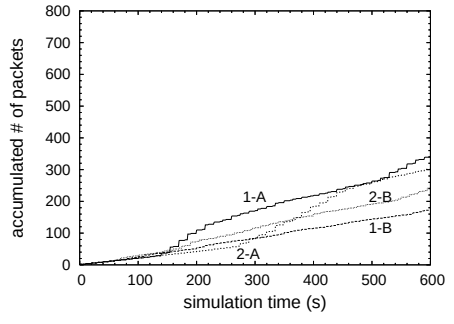
(a) 情報受信ノード数（行動変化あり）



(b) 情報受信ノード数（行動変化なし）



(c) メッセージ総数（行動変化あり）



(d) メッセージ総数（行動変化なし）

図 9 情報配布システムの性能評価
Fig. 9 Performance of infomation diffusion system.

到着率は低く，制御パケット数は多くなる．

一方，図 8 (c) では，UPF と RWP/ob は似たような特性を示している．これは両ケースにおいて，2 つのノード間の距離はほぼ同じであるためと考えられる．また，図 8 (d) では，UPF に比べて RWP/ob のホップ数の分布は中央付近に集中している．これは，RWP/ob では道路上の歩行者密度はすべての道においておおよそ同じであるが，UPF では道路によって密度に偏りが生じ，そのため UPF の方がホップ数は広く分散したと考えられる．

5.2 MANET 上での情報配布システム

次に，MANET 上で広告情報などを配布する位置依存データ配信システムのシミュレーションを行った．この実験では，ネットワークアプリケーションから得られた情報に基づき動的な行動変化を行う場合と行わない場合とを比較することにより，動的な行動変化によるネットワークの性能評価への影響を測定する．

情報の配布は非同期型のマルチキャスト方式で行われ，簡易基地局や近隣ノードから配布された情報を受信したノードは配布された情報をキャッシュし，15 秒間隔でその情報を再配布すべきかどうかを，自分の位置 P から店 A（図 6）の位置 S までの距離 $d(P,S)(m)$ を用いた確率関数（配布ポリシ）に従って判定する．ノードは一度情報を配布するかキャッシュしてから 60 秒経過した時点でキャッシュした情報を破棄する．

配布ポリシは，(1) 配布確率 $= 1-(d(P,S)/500\sqrt{2})$（店 A から近いほど高確率で配布），(2) 配布確率 $= d(P,S)/500\sqrt{2}$（店 A から遠いほど高確率で配布），の 2 つを，基地局はフィールドの中心付近に配置された基地局 A と南に配置された基地局 B の 2 つを用意し，それぞれの組合せで計 4 通りのシミュレーションを行った．以下，配布ポリシ i，基地局 x のシミュレーションをシナリオ *i-x* で表す．また，情報を受信すれば目的地を店 A に変更するルール（図 5 参照）を CPE 記述に加えて各シナリオ *i-x* を用いたシミュレーションを行った．このシナリオを *i-x*(C) で表す．シミュレーション時間は 600 秒，簡易基地局からの情報配信は 180 秒経過後から開始し，以降は 2 秒間隔で行う．ノード数などその他の設定は 5.1 節と同じとする．

図 9 に情報を受信したノード数とメッセージ総数を示す．基地局 B のシナリオで行動変化を行う場合と行わない場合では，情報受信ノード数に大きな差が見られる．行動変化を行う場合，情報を受け取ったノー

表 1 行動，ネットワークシミュレータ間の同期処理に要する時間
Table 1 Time for interaction between network and behavior simulators.

同期間隔 T（秒）	0.4	1.0	2.0	5.0	10.0	30.0
シミュレーションに要した時間（秒）	79.39	77.20	74.42	74.87	73.32	65.59
同期処理に要した時間（秒）	2.49	0.995	0.525	0.29	0.15	0.06
同期処理の占める割合（%）	3.14	1.29	0.71	0.39	0.20	0.09
2VT（V=1.7 m/秒）（m）	1.36	3.40	6.80	17.00	34.00	102.00
ノード座標の誤差の最大値（m）	0.77	2.33	3.80	8.30	15.72	45.40
ノード座標の誤差の平均（m）	0.53	1.33	2.66	6.62	13.20	38.98
ノード座標の誤差の標準偏差	0.052	0.130	0.267	0.667	1.364	4.549

ドは店 A の周辺に集まる．そのため，基地局から離れていて情報が伝わりにくかった図 6 の上方を歩いているノードにも情報が伝達しやすくなったのではないかと考えられる．また，情報が伝わったノードが増加すれば，再配布を行うノード，ひいてはメッセージ総数も増加する（図 9 (c)，(d)）．

この事例で示したようなアプリケーション出力に対する動的な行動変化は，シミュレーションを行う前に行動を変化させる人数を予測するのが困難であるため，あらかじめノードの行動のみをシミュレートした結果を利用してネットワークシミュレーションを行う手法では，正確にシミュレートできない．また行動変化を行うノードの割合が多ければネットワークシステムにも大きな影響が出る．このような相互関係をより正確にシミュレートするためには，提案方式のようなアーキテクチャは不可欠であると考えられる．

5.3 同期間隔とシミュレーション実行時間

最後に DSR プロトコルのシナリオを用いて，行動，ネットワークシミュレータ間の同期間隔を変え，シミュレーションと同期処理に要する時間および，行動シミュレータとネットワークシミュレータ間のノード座標の誤差に関する測定を行った．その結果を**表 1** に示す．ノード座標の誤差は，各ノードについて，行動シミュレータが保持するノード座標と，行動シミュレータから最新の位置情報などを受け取る直前のネットワークシミュレータが保持するノード座標との距離を計算することにより導出した．ノードの移動速度は 1.1 m/s〜1.7 m/s とし，ノード数は約 250 であった．

ノード数が多くなるほど同期処理にかかる時間も長くなるが，同時にシミュレーション全体にかかる時間も長くなるため，同期処理の占める割合はそれほど変わらないことが分かっている．表 1 によると同期処理の占める割合は同期間隔が最も短い場合でも数%程度であり，同期処理にかかるオーバヘッドは十分小さいといえる．

また，ノード座標の誤差のとりうる値はノードの移動速度に依存する．具体的には，ノードの最高移動速度を V，同期間隔を T とすると，誤差は $2VT$ 以下となる．だが現実には，速度が急激に変化することは稀であるため誤差はもっと小さくなる（表 1）．よって同期間隔は，シミュレーションに求める精度にもよるが，$2VT$ が数 m 程度に抑えられるよう設定すれば十分であると思われる．我々の実験では同期間隔を 1 秒に設定してシミュレーションを行っている．

6. ま と め

本論文ではモバイルアドホックネットワークのシミュレーションにおいて，ノードの現実的なモビリティを記述するための新たな手法を提案し，その手法を用いたネットワークシミュレータ MobiREAL の設計，開発について述べた．モバイルアドホック通信では，シングルホップ通信によるデータ配布，マルチホップ通信における AODV や GPS を用いた位置依存ルーティングプロトコルなどいくつかの通信プロトコルが考えられるが，それらの性能はノードのモビリティモデルに大きく影響を受ける．特に MobiREAL では，ノード行動とネットワークシステムの依存関係に着目し，それらが互いに影響を及ぼしあうようなケースを考慮することで，より現実に即したシミュレーションを可能としている．シミュレータとしての利便性を考慮し，ライブラリの充実を図ることなどが現在の課題である．

参 考 文 献

1) Broch, J., Maltz, D., Johnson, D., Hu, Y.-C. and Jetcheva, J.: A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, *Proc. ACM/IEEE MobiCom*, pp.85–97 (1998).
2) Camp, T., Boleng, J. and Davies, V.: A Survey of Mobility Models for Ad Hoc Network Research, *Wireless Communications & Mobile Computing* (*WCMC*)*: Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, pp.483–502 (2002).
3) Bai, F., Sadagopan, N. and Helmy, A.: The IMPORTANT Framework for Analyzing the

Impact of Mobility on Performance of Routing for Ad Hoc Networks, *AdHoc Networks Journal*, pp.383–403 (2003).
4) Maeda, K., Sato, K., Konishi, K., Yamasaki, A., Uchiyama, A., Yamaguchi, H., Yasumoto, K. and Higashino, T.: Getting Urban Pedestrian Flow from Simple Observation: Realistic Mobility Generation in Wireless Network Simulation, *Proc. 8th ACM/IEEE Int. Symp. on Modeling, Analysis, and Simulation of Wireless and Mobile Systems* (*MSWiM*), pp.151–158 (2005).
5) Jardosh, A., Belding-Royer, E.M., Almeroth, K.C. and Suri, S.: Towards Realistic Mobility Models For Mobile Ad hoc Networks, *Proc. ACM MobiCom*, pp.217–229 (2003).
6) Hollick, M., Krop, T., Schmitt, J., Huth, H.-P. and Steinmetz, R.: Modeling Mobility and Workload for Wireless Metropolitan Area Networks, *Computer Communications*, pp.751–761 (2004).
7) Musolesi, M., Hailes, S. and Mascolo, C.: An Ad Hoc Mobility Model Founded on Social Network Theory, *Proc. ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems* (*MSWiM*), pp.20–24 (2004).
8) 岡田公孝，和田　剛，高橋幸雄：個人行動をベースにした歩行モデルと歩行流シミュレーション，日本オペレーションズ・リサーチ学会，春季研究発表会 (2003).
9) Riley, G.F.: The Georgia Tech Network Simulator, *Proc. ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research*, pp.5–12 (2003).
10) MobiREAL Simulator Web Page. http://www.mobireal.net

(平成 17 年 5 月 23 日受付)
(平成 17 年 11 月 1 日採録)

前田久美子（学生会員）
平成 16 年大阪大学基礎工学部情報科学科飛び級のため中退，現在，大阪大学大学院情報科学研究科博士前期課程在学中．アドホックネットワークに関する研究に従事．

小西　一樹
平成 16 年大阪大学基礎工学部情報科学科飛び級のため中退，現在，大阪大学大学院情報科学研究科博士前期課程在学中．アドホックネットワークに関する研究に従事．

佐藤　和基（学生会員）
平成 16 年大阪大学基礎工学部情報科学科飛び級のため中退，現在，大阪大学大学院情報科学研究科博士前期課程在学中．アドホックネットワークに関する研究に従事．

山口　弘純（正会員）
平成 6 年大阪大学基礎工学部情報工学科卒業．平成 10 年大阪大学大学院基礎工学研究科博士後期課程修了．同年オタワ大学客員研究員．平成 11 年大阪大学大学院基礎工学研究科助手．平成 14 年より同大学院情報科学研究科助手．博士（工学）．分散システムや通信プロトコルの設計および実装に関する研究に従事．

安本　慶一（正会員）
平成 3 年大阪大学基礎工学部情報工学科卒業．平成 7 年大阪大学大学院基礎工学研究科博士後期課程退学後，滋賀大学経済学部助手．平成 9 年モントリオール大学客員研究員．平成 14 年より奈良先端科学技術大学院大学情報科学研究科助教授．博士（工学）．分散システム，マルチメディア通信システムに関する研究に従事．IEEE/CS 会員．

東野　輝夫（正会員）
昭和 54 年大阪大学基礎工学部情報工学科卒業．昭和 59 年大阪大学大学院基礎工学研究科博士課程修了．同年同大学助手．現在，同大学大学院情報科学研究科教授，工学博士．分散システム，通信プロトコル，モバイルコンピューティング等の研究に従事．電子情報通信学会，ACM 各会員．IEEE Senior Member．