Available online at www.sciencedirect.com
ScienceDirect
Ad Hoc Networks 7 (2009) 153–170
Ad Hoc Networks
www.elsevier.com/locate/adhoc

Urban pedestrian mobility for mobile wireless network simulation

Kumiko Maeda [a,*], Akira Uchiyama [a], Takaaki Umedu [a], Hirozumi Yamaguchi [a], Keiichi Yasumoto [b], Teruo Higashino [a]

[a] Graduate School of Information Science and Technology, Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
[b] Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara 630-0192, Japan

Received 5 March 2007; received in revised form 29 December 2007; accepted 2 January 2008
Available online 17 January 2008

Abstract

In order to perform precise evaluation of MANET applications in the real world, realistic mobility models are needed in wireless network simulation. In this paper, we propose a new method to create urban pedestrian flows (UPF) mobility scenarios, which reproduce the walking behavior of pedestrians in urban areas. From given densities of pedestrians observed at several points, our method derives a UPF mobility scenario that reproduces the walking behavior of pedestrians consistent with the observed densities, using linear programming techniques. We have developed a network simulator MobiREAL to design and evaluate MANET protocols and applications with this realistic mobility model. MobiREAL provides various functions and tools including a mobility model to describe the behavior of individual nodes, a GUI to assist with automatic generation of UPF mobility scenarios and a visualization tool. We have conducted some experiments using the MobiREAL simulator. Through the experiments, we have investigated the influence of node mobility on the performance of MANET protocols and have shown the usefulness of our method and the MobiREAL simulator.

Keywords: Mobile ad hoc network; Simulation; Mobility model; Performance evaluation; Design support

1. Introduction

Mobile ad hoc networks (MANETs) are expected to be very useful and important infrastructure for achieving a future ubiquitous society. However, designing MANET protocols and applications is a very complicated task since it is hardly possible to build large-scale and realistic testbeds in the real world for performance evaluation. Thus there are always demands for methodologies that allow us to design, analyze and validate given applications in simple and inexpensive ways. Nowadays network simulators are mainly used for such a purpose. Since the constituents of MANETs are not stationary, mobility models greatly affect the performance of MANET systems [1–5]. Therefore, in order to evaluate the performance of MANET protocols and applications more precisely in the real world, we need more realistic mobility models. However, there is a tradeoff between the reality of mobility models

* Corresponding author. Tel.: +81 6 6879 4557; fax: +81 6 6879 4559.
E-mail address: k-maeda@ist.osaka-u.ac.jp (K. Maeda).

1570-8705/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2008.01.002

and the cost for their complexity. In particular, modeling movement of real nodes (e.g., pedestrians) with high fidelity for the evaluation of town-wide deployed networks requires detailed observation or survey of those people moving from place to place along streets. Obviously such observation cannot be easily carried out due to cost, privacy and other reasons.

In this paper, we propose a method to create urban pedestrian flow (UPF) mobility scenarios. The method targets the reproduction of the walking behavior of pedestrians in city sections, shopping malls and so on. Given the average densities of pedestrians on certain streets, which can be easily obtained by fixed point observations, and a set of walking paths pedestrians are likely to follow, the method determines flows of pedestrians using linear programming (LP) techniques. Also, the maximum error between the observed density and the derived density is minimized to reproduce realistic movement of pedestrians. In our experiment, two persons have measured the average densities of pedestrians on 33 streets in a 500 m $\times$ 500 m area in front of a large train station in Osaka for about a half hour. The maximum error between the observed densities and the derived densities was only 9.09%. We have developed a network simulator called MobiREAL for simulating MANET systems with UPF mobility scenarios.

MobiREAL has been developed by extending the network simulator GTNetS [6], developed at the Georgia Institute of Technology. The main features of MobiREAL are twofold. First, MobiREAL introduces an original model called the condition probability event (CPE) model to describe the dynamic behavior of pedestrians, such as adjusting walking speeds and directions to avoid collision with neighbors and obstacles, and stopping at a traffic signal. By the CPE model, we can also describe the interaction between pedestrians and networks, e.g., we can describe a scenario that a mobile node makes a detour when it receives traffic jam information through MANETs or cellular networks. MobiREAL is developed by integrating the framework to enable the interaction between mobile nodes and network applications. By incorporating this CPE model into UPF mobility scenarios, the reality of simulations is considerably improved. Secondly, MobiREAL provides a suite of useful tools. With a visualization tool called an animator, the results of simulations can be analyzed easily and intuitively. The animator can visually animate the movement of nodes, network topology, packet propagation and so on, and can also show statistical information like node density and the packet error rate observed in each sub-region. Also, the UPF scenario generation tool helps to input geographic information like obstacles and street structure, and automatically generates UPF scenarios. We note that the UPF scenario generator and the mobility simulator part of MobiREAL can be used independently of the network simulator part of MobiREAL. Therefore, it can easily be applied to creating trace-based mobility scenarios supported by many other network simulators. This feature allows users of the other simulators to receive the benefit of realistic mobility scenarios generated by our toolset.

We have conducted some simulation experiments using the MobiREAL simulator. Through the experiments, we have investigated the influence of node mobility on the performance of MANET protocols and have shown the usefulness of the proposed method.

2. Related work

We often use simple mobility models such as the random way point (RWP) model [7] in free space for simulating MANET protocols. This is because such simple mobility models are commonly available in many simulators and therefore can be a common environment for comparative experiments. Several analytical results have been presented for the RWP model and its variants [8–18]. In [8], Bettstetter presents a variant of RWP where nodes can change their directions smoothly, keeping the same analytical properties as RWP. In [9], Chu and Nikolaidis address that the node density of RWP is non-uniform and that the non-uniformity depends on the speed of nodes. Rojas et al. [10] show that the Cauchy distributions and the chi-square distributions can model the deployment of locations and residual pause times of waypoints more accurately than uniform distributions and exponential distributions, respectively. Hyytia et al. [11] derive an expression that represents the nodes' position distribution of RWP in an arbitrary convex domain and propose the RWPB model that forms contrastive distribution with RWP. Yoon et al. [12] focus on the velocities of nodes in RWP and prove that the harmful effects are observed in the performance evaluation before reaching the steady-state. Yoon et al. also present the "sound mobility model" based

on the RWP model in [13]. This model is designed to maintain the average velocities of nodes to exclude the harmful effects. McGuire [17] derives the stationary distribution of nodes for a general class of mobility models. Nain et al. [18] analyze the stationary distribution of nodes in the random direction model and describe its usefulness compared with RWP.

There are many methodologies that synthesize mobility models from observed data and geography information to pursue reality [19–24]. In [19], Jardosh et al. introduce obstacles (i.e., buildings) and pathways in a given simulation field. A pathway between any two obstacles can automatically be generated using a Voronoi graph computation algorithm. The research group also provides plug-ins for GloMoSim [25] and ns-2 [26] simulators to utilize the proposed mobility. Hollick et al. [20] propose a macroscopic mobility model for wireless metropolitan area networks, where a simulation field is divided into multiple zones with different attributes such as workplace, commercial and recreation zones. Also, each mobile node has an attribute of resident, worker, consumer or student. Given trips with destinations for user nodes, an existing urban transportation planning technique is used to estimate the user density in each zone. Hsu et al. [21] present the WWP (weighted way point) model. The WWP model defines a set of crowded regions such as cafeterias and buildings at a university. Given a distribution of pause times for each region and a transition probability of nodes for each pair of regions, it uses a Markov model to model nodes' movement between these regions. For vehicular ad hoc networks, a technique to reproduce realistic mobility of vehicles in ns-2 is proposed in [22]. Group-based mobility models also capture some aspects of realistic movement. Musolesi et al. [23] propose a group-based mobility model based on social networks of individual users. Also, several group-based mobility models are proposed in [24].

There is some research analyzing the relationship between network traffic and the behavior of users by tracing user behaviors on wireless networks. Kotz and Essien [27] analyze user behavior patterns by collecting traces of 2000 users for 11 weeks at 476 wireless APs distributed in 161 buildings at Dartmouth College. They found interesting characteristics of traffic depending on time and location. The authors of [28–30] have also collected similar traces from wireless network users in a metropolitan area wireless network at the SIGCOMM2001 conference location and in three large corporate buildings, respectively. Thajchayapong and Peha [31] obtained traces from the IEEE 802.11-based system at the Carnegie Mellon University campus. Contrary to most researchers' expectations, the cell residence times do not form an exponential distribution. McNett and Voelker [32] analyze the mobility patterns of users of wireless hand-held PDAs in a campus wireless network and synthesize the "campus waypoint model" that serves as a trace-based complement to the RWP model. Also, Kim et al. extract user mobility characteristics from wireless network traces and present a mobility model based on the characteristics in [33]. These research efforts aim at getting knowledge on capacity planning and AP arrangement for building next-generation large-scale mobile network infrastructures. However, they require some wireless network infrastructures and a very large amount of information collection to produce a mobility scenario.

Many realistic mobility models including our proposed methodology have intended to model the environment in the real world. Our proposed methodology is closest to the work presented in [19–21], which tries to reproduce the real world's geography and movement of nodes. These methodologies require some observation of user behaviors in a target region, but the simplicity of observation is an important factor because a large amount of human and system resources are required to collect complete information about users' behavior. Therefore, we propose a methodology to reproduce urban pedestrian flows from the density of nodes. Densities of pedestrians can be obtained by fixed-point observation using cameras and therefore it is easier than measuring transition probabilities among hotspots, which is assumed by the methods presented in [20,21]. The UPF model considers the node density to reproduce the nodes' moving flows over streets, while many existing models use the node density at the hotspots (i.e., major points) to reproduce the nodes' transitions among them. The UPF model possibly fails to accurately reproduce the transition probability among the major points, but can reproduce realistic flows on streets in the city section, which are abstracted in those existing models.

2.1. Our contributions

One of the advantages of the MobiREAL is that we can dynamically change the behavior of nodes depending on the information received through the

network. The dynamic control of pedestrians' behavior can be written simply using the CPE model. Another advantage is the powerful GUI tools (Animator and UPF scenario generator). From simple fixed-point observations on roads, we can automatically calculate the flow rate of each pedestrian flow from a starting point to a destination. As a total, multiple pedestrian flows reproduce rather realistic pedestrian flows. We can also use QualNet or OPNet as a network simulation part in the proposed framework. Details are described in Section 5.4.

Compared with our preliminary studies [5,34] that presented the basic concept of the UPF mobility generation method and the basic design of the CPE model, respectively, this paper gives a more formal and precise description of their design and implementation. Also, we added two experiments that evaluated the accuracy of mobility (Section 6.2) and two case studies that evaluated realistic applications and protocols with UPF mobility scenarios using the MobiREAL simulator (Section 6.3). Through the case studies, we could confirm that realistic (thus non-uniform) node movement and position distribution have a significant influence on performance of mobile wireless network systems. This fact encourages us to address the effectiveness of our UPF mobility and MobiREAL toolset presented in this paper. Also, we made several minor but important modifications to the MobiREAL simulator to enhance its capability. For example, we refined and simplified the CPE model to be harmonized with the UPF mobility scenarios and enhanced the facility of the MobiREAL animator to support statistical data representation. As a consequence, the contribution of this paper is that we present a complete toolset for the development of protocols and applications and prove its usefulness through case studies.

3. Deriving urban pedestrian flows from simple observations

To generate a UPF scenario, we give a street map of the simulation field as a graph called a *street graph*, and possible paths of pedestrians on the street graph. We also give average densities of pedestrians on some streets. These densities can be simply obtained by fixed-point observations and so on. Then for each path, we calculate the number of pedestrians that come into the path per unit of time (this number is called *flow rate*) using a linear programming (LP) technique, where the maximum error between the density obtained from the observation and that derived from the LP solution is minimized. Using the derived flow rates, the UPF scenario generation tool generates a UPF scenario that can be used in the MobiREAL simulator. This will be described later.

3.1. User inputs

3.1.1. Simulation field

A simulation field may contain polygons that represent obstacles such as buildings and an undirected street graph $G = (V, E)$ where V and E represent geographic points and streets, respectively (Fig. 1). The geographic points are set at intersections and entrances to buildings, stations, terminals, underpasses, shopping centers and so on. For each edge $e_{ij} \in E$, the width W_{ij} of the street is also given.

3.1.2. Pedestrians' paths

In urban areas, many types of pedestrians are walking around for different purposes. Commuters usually get off trains and go straight to their transfer points or offices. Shoppers visit their favorite shops and remain for a while. We estimate the possible walking paths of pedestrians and represent them as a set R of acyclic paths on street graph G. Each path starts and ends at vertices that can be starting points and destinations like stations and buildings. We note that if a path contains a loop, we may regard this loop as another path and separate it from the original path. Then we can avoid loops even though the given paths contain them.

It may be difficult to enumerate all of the possible paths. Therefore, we may use the following simple enumeration. In general, pedestrians are considered to walk along the shortest paths. So, we specify some points where many people appear or disappear (e.g., stations and shops) and compute the shortest path for every pair of these points. For each path computation, we can choose one of several

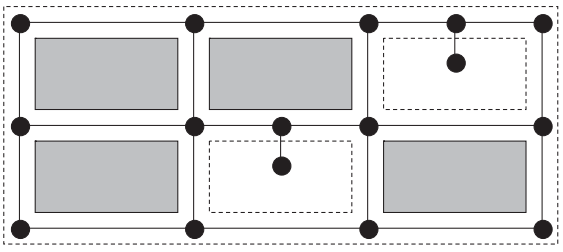

Fig. 1. Simulation field and street graph.

options case by case. For example, considering the fact that people tend to walk along major streets, we put some weights that prioritize these streets in the shortest path construction algorithm.

3.1.3. Observed density of pedestrians

Next, we have to determine a flow rate of each path, that is, the number of pedestrians that come into the path per unit of time. We give the densities on some streets measured in the real world and derive flow rates based on these values. We let $E_M(\subseteq E)$ denote the set of edges on which densities of pedestrians are observed. For each edge $e_{ij} \in E_M$, we let D_{ij} denote the observed density on the edge.

In general, for the node density d on an edge, the pedestrian rate g on the edge (i.e., the number of pedestrians going into the edge per second), the speed v of pedestrians on the edge and the width w of the edge, the following equation holds:

$$d_{(\text{person/m}^2)} = \frac{g_{(\text{person/s})}}{w_{(\text{m})} \cdot v_{(\text{m/s})}}. \tag{1}$$

It is known that the speed of a pedestrian follows the negative increase of density. Therefore,

$$v = -k \cdot d + v_0, \tag{2}$$

where k and v_0 are positive constants. Here, v_0 is the average speed of the pedestrian in a sparse area. Also on any (very) crowded street, a particular density (say $D_{\max}$) was observed when the speeds of pedestrians are very close to 0. Therefore, from (2), we obtain

$$-k \cdot D_{\max} + v_0 = 0. \tag{3}$$

Finally, from Eqs. (1)–(3), we obtain the following pedestrian rate calculation function G from the density d and width w of the edge:

$$g = G(d, w) = v_0 \cdot w \cdot d \cdot \left(1 - \frac{d}{D_{\max}}\right). \tag{4}$$

We should select streets to observe by considering the characteristics of the field geography and the desired preciseness of the reproduction. The solution will become more precise if we obtain more information about densities. We evaluate the relationship between the number of streets and the preciseness of the result in Section 6.2.1.

3.2. Determining flow rates

Next, we describe the algorithm to derive flow rates from the observed pedestrian density. Given an undirected graph G, the width W_{ij} of each edge $e_{ij} \in E$, a set R of paths, and the observed density D_{ij} on each edge $e_{ij} \in E_M$, the algorithm derives the flow rate of each path in R, minimizing the maximum error between the derived and observed densities on the edges.

To formulate this problem as an LP problem, we introduce a variable f_k representing the flow rate of path $r_k \in R$ and a variable g_{ij} representing the pedestrian rate on edge $e_{ij} \in E$. We also introduce a variable δ, which means the maximum error of the derived pedestrian rate from the corresponding pedestrian rate given by Eq. (4).

For each edge e_{ij}, its pedestrian rate is the sum of flow rates on e_{ij}. Therefore, the following equation holds:

$$\forall e_{ij} \in E; \quad g_{ij} = \sum_{r_k \in R \wedge e_{ij} \in r_k} f_k. \tag{5}$$

The following inequality limits the errors of derived pedestrian rates by δ, where function G is given by Eq. (4):

$$\forall e_{ij} \in E_M; \quad -\delta \leqslant \frac{G(D_{ij}, W_{ij}) - g_{ij}}{G(D_{ij}, W_{ij})} \leqslant \delta. \tag{6}$$

Also pedestrian rate g_{ij} has an upper limit derived from the nature of function G. From the definition of G in (4),

$$\begin{aligned} g_{ij} &= G(d_{ij}, W_{ij}), \\ &= v_0 \cdot W_{ij} \cdot d_{ij} \cdot \left(1 - \frac{d_{ij}}{D_{\max}}\right), \\ &= -\frac{v_0 \cdot W_{ij}}{D_{\max}} d_{ij}^2 + v_0 \cdot W_{ij} \cdot d_{ij}, \\ &= -\frac{v_0 \cdot W_{ij}}{D_{\max}} \left(d_{ij} - \frac{D_{\max}}{2}\right)^2 + \frac{v_0 \cdot W_{ij} \cdot D_{\max}}{4} \end{aligned}$$

is obtained. Knowing that v_0, W_{ij} and $D_{\max}$ are all positive constants, the following inequality is obtained:

$$g_{ij} \leqslant \frac{v_0 \cdot W_{ij} \cdot D_{\max}}{4}. \tag{7}$$

Our objective is to minimize the maximum error δ. Under constraints (5)–(7), the objective function is defined as follows:

$$\min \delta. \tag{8}$$

By solving this LP problem, we obtain the flow rate f_k for each path $r_k \in R$ minimizing the maximum error δ of the derived pedestrian rates. Using the derived flow rates, a UPF mobility scenario that can

be used in MobiREAL is generated automatically by the UPF scenario generation tool. The details of this tool are given in Section 5.1. In the scenario, we generate node instances following the derived flow rate f_k, and the nodes walk along the path r_k. A path with flow rate 0 means that there is no pedestrian flow along the path.

4. Condition probability event model

The UPF model focuses on macroscopic movement of nodes, while the condition probability event (CPE) model focuses on the microscopic behavior of the individual nodes. The CPE model allows us to dynamically and more precisely control the behavior of nodes on pedestrian flows determined by the UPF model. With this model, we may describe such behavior that a mobile node makes a detour (changes to another pre-scheduled route) when it receives traffic jam information through MANETs or cellular networks.

We would like to note that general-purpose programming languages such as C or Perl allow a wide variety of descriptions, and therefore can be used to describe the detailed behavior of nodes. However, this generality may confuse designers. Thus we propose the CPE model for simulator users to simply describe the behavior of nodes (simulation scenarios). In the CPE model, the next action of a node is determined by its current status (position, direction, velocity, scheduled route and so on) and environmental factors (global time, information from network systems and so on), and this decision can be probabilistic. This model captures well humans' decision-process of actions (action is probabilistically determined by current status and environment), allowing reasonable abstraction of unnecessary behavior for the fidelity of modeling. Thus it has much better applicability for behavior modeling than general languages.

We depict our concept of modeling pedestrian behavior in Fig. 2. Formally, the CPE model consists of a list of rules where each rule is a tuple of a *condition*, a *probability* and an *action*, and *node state variables*, which represent the status of each node and are introduced later. The rules can refer to and update node state variables like velocity vector and position of the node. We may specify the same set of rules to all nodes or a different set of rules to each group of nodes. We specify a logical formula as a condition, a value in the range of $[0,1]$ as a probability and a set of substitution statements that may update values of the node state variables as an action. For each increment of the simulation clock (referred to as T), all rules satisfying the given conditions are executed one-by-one from the top of the list with the specified probabilities.

Each node has six node state variables: current position P, scheduled route ROUTE (sequence of vertices), base velocity V_f, actual velocity vector $\vec{V}$, input data AI of the node to the network system (application input) and output data AO from the network system to the node (application output). These variables are referred to and updated by the rules. When a node is generated according to the UPF scenario derived in Section 3, an instance of the CPE model of the node is initialized as follows. The path $r_k \in R$ is set to the initial value of route ROUTE, and the velocity v_0 in Exp. (2) is set to V_f, which represents the velocity in a sparse area. Then normal descriptions in the CPE model contain such rules that update ROUTE and V_f followed by the rules that update current position P and actual velocity vector $\vec{V}$ accordingly. These rules determine the basic behavior of nodes that walk along the

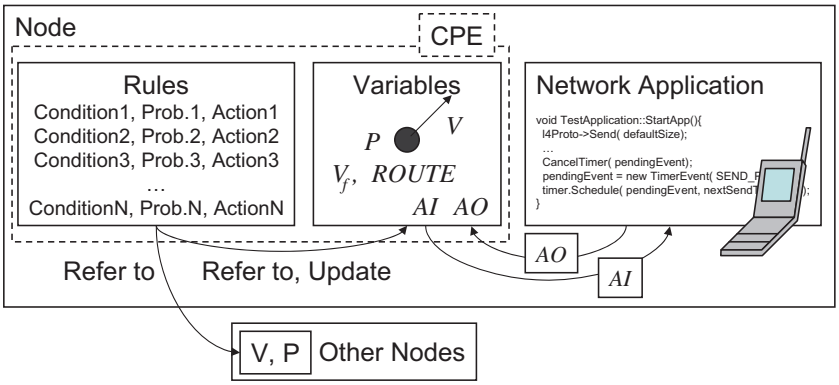

Fig. 2. Interaction between CPE instance and network application.

routes. The actual velocity $|\vec{V}|$ cannot be greater than the base velocity V_f, but may be smaller when the node cannot walk smoothly in a crowd.

The CPE model assumes that an application program of the network simulator is notified of the value of application input AI, and updates the value of application output AO. In the rules of the CPE model, we can set any value to application input AI but can only read application output AO. The variables AI and AO enable simulator users to describe interactive behavior of nodes with the application program. We note that in order to enable the application program and the CPE instance to interpret AI and AO, several libraries are offered to mitigate the developers' effort.

We offer many pre-defined functions to help the developers to specify realistic behavior of nodes in the CPE model. For example, we provide a function that returns the positions and actual velocity vectors of nodes within eyesight. A part of them are presented in the following example.

4.1. Example description in CPE model

Here, we consider the following MANET system. Shoppers holding information terminals with short range radio communication devices exchange useful information (e.g., sales information) when they encounter each other.

We present a description written in the CPE model for the shoppers in Table 1. Rule E1 describes the behavior that changes the scheduled route ROUTE dynamically, according to an application output AO. If the shopper receives information "SHOP_A" given by the application, he/she changes his/her destination to the vertex "SHOP_A" with probability 0.3. This is done by updating the ROUTE variable to the shortest path from current position P to the vertex specified by Point(SHOP_A). Rule E2 describes the behavior that gives an input to the application depending on the position of the given node. More concretely, the node gives an application input "SHOP_A" to variable AI with probability 0.1 when he/she reaches the vertex specified by Point(SHOP_A). We note that the application is programmed to distribute the given input to the neighbor nodes. Rule E3 implements the behavior that follows a traffic signal at intersection INT_1. A function "stop( )" used in the action part sets 0 to the node's base velocity V_f temporarily. The condition of E3 becomes true if the simulation time T (min.) is even, which means the period of the red light.

Rules E4 and E5 define fundamental behavior based on the UPF mobility. Rule E4 calculates velocity vector $\vec{V}$ from current position P, route ROUTE, base velocity V_f and the position of the neighbor nodes neighbors(P, V_f). We note that we have modeled evasive actions (turn or slow down) against the approaching people. The UPFvector( ) in Rule E4 calculates velocity vector $\vec{V}$ based on this model to make the behavior of the node more realistic. Rule E5 updates current position P based on velocity vector $\vec{V}$ calculated by Rule E4.

5. MobiREAL

We have developed a network simulator called MobiREAL to simulate MANET systems with the realistic behavior of nodes based on the UPF model and the CPE model.

The architecture of the MobiREAL simulator is shown in Fig. 3. MobiREAL is composed of a network simulator that simulates network systems, a behavior simulator that simulates the movement of nodes, a UPF scenario generator and an animator. The behavior simulator takes three inputs, a simulation field, a UPF scenario and a set of CPE

Table 1
Example behavior description of shoppers in CPE model

	Condition	Prob.	Action
E1[exp.]	AO == "SHOP_A"[receives string "SHOP_A" from the application]	0.30	ROUTE = shortest_path(P, Point(SHOP_A)); [change destination to vertex SHOP_A]
E2	P == Point(SHOP_A) [arrives at vertex SHOP_A]	0.10	AI = "SHOP_A"; [give an application input "SHOP_A"]
E3	(P == Point(INT_1)) ∧ ((T%2 min) < 1 min) [stops at vertex INT_1 with traffic light which changes every min.]	1.00	stop( ); [wait until the light changes to green]
E4	– [Execute always]	1.00	V = UPFvector(V_f, P, ROUTE, neighbors (P, V_f)); [calculate velocity vector from state various variables]
E5	– [Execute always]	1.00	P = UpdateLocation(P, V); [update position]

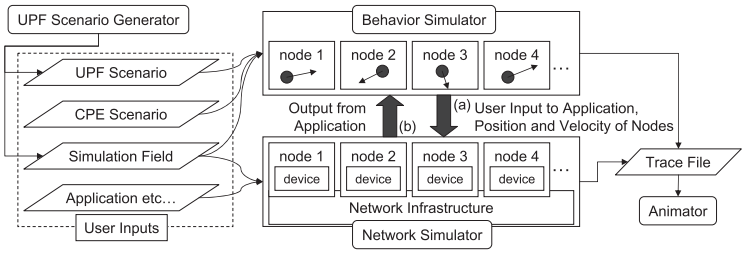

Fig. 3. MobiREAL simulator overview.

scenarios. As a simulation field, we specify the structure of streets and buildings that interfere with radio propagation. As a UPF scenario, we specify the routes of nodes and their node generation rates (flow rates). According to the UPF scenario, MobiREAL generates node instances that move based on the CPE scenarios. With the UPF scenario generator, we can generate a simulation field and the corresponding UPF scenario graphically and intuitively. For the network simulator, we specify application programs. In each application component running on a mobile node, interaction with the corresponding CPE instance may be described.

Since the simulator part of MobiREAL is composed of two independent programs (i.e., behavior simulator and network simulator), we have implemented MobiREAL according to the following concept. First, both simulators hold the positions and speeds of nodes independently and synchronize the progress of simulations. The behavior simulator calculates the latest positions, directions and speeds of nodes, and sends them to the network simulator (arrow (a) in Fig. 3). In response, the network simulator updates its information about the nodes according to the received data, sends outputs of the application to the behavior simulator (arrow (b) in Fig. 3) and continues the simulation. The behavior simulator can perform dynamic behavior change according to this information.

5.1. UPF scenario generator

We have developed the MobiREAL UPF scenario generator. Showing a BMP graphic map file like Fig. 4, simulator users can specify a street graph through the GUI with their mouse devices. We specify two attributes (width and observed node density) for the street. This tool also assists in generating potential candidate routes as described in Section 3.1.2. Then this tool generates a program code of the UPF scenario that can be run in MobiREAL

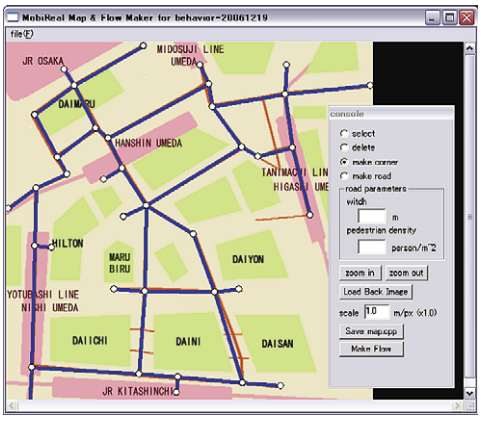

Fig. 4. GUI support for map and UPF scenario generation.

where the linear programming problem solver lp_solve [35] is used to calculate the flow rate on each route.

5.2. Network simulator

The network simulator needs to synchronize with the behavior simulator to exchange information, update the positions and the speeds to follow the behavior simulator, and process application inputs and outputs. We have developed the network simulator by extending and modifying GTNetS [6] developed at the Georgia Institute of Technology. GTNetS aims at improving the scalability of large-scale simulation, so it has a mechanism for parallel simulation of wired networks. For simulation of wireless networks, GTNetS also supports routing protocols like DSR, AODV and NVR (wireless form of Nix-Vector routing) and standard MAC and physical layer protocols like IEEE 802.11.

In GTNetS and many other simulators, the movement of nodes is bounded by boundaries, or a node moving through a boundary appears from the other side. This means that the number of nodes is fixed throughout simulations. On the other hand,

our mobility requires dynamic generation and deletion of nodes. In our MobiREAL network simulator, we have modified the node management function of GTNetS to implement the dynamic generation and deletion of nodes. Also, we have modified the simulation module of the physical layer to implement radio propagation attenuation under the existence of obstacles in order to achieve more realistic wireless network simulation. In this radio propagation model, attenuation when the radio penetrates obstacles is considered. For this purpose, simulator users can specify an attenuation coefficient for each obstacle.

5.3. Animator

The MobiREAL Animator visualizes the traces of the simulation. Visualization of the simulation results is very useful for designing, debugging and presenting network systems. In particular, visualizing mobility of nodes is mandatory to confirm their realistic behavior.

MobiREAL Animator runs on the Microsoft Windows platform, and has been developed using DirectX. Animator can animate the movement of nodes, the topology of wireless links, packet propagation and so on. We can choose whether these objects are to be displayed or not. Each mobile node is represented as a small circle, and a semi-transparent concentric circle represents its radio range. Each packet transmission is drawn by concentrically growing circles with colors. First, a circle with 1 pixel radius is drawn when a packet is sent. Then the radius of the circle gradually grows until it reaches the radio range of the node sending the packet. The color of nodes can be freely set in a network application program. For example, Animator can show the nodes that received certain packets in red and others in blue. A snapshot from Animator is shown in Fig. 5a.

Animator can also show some measured metrics like node density and packet loss rate as shown in Fig. 5b. In this function, a simulation field is separated into grid cells, and they can be color-coded according to the measured values. Thus users can analyze, for example, the influence of geography on the density of nodes, and the relationship between the density of nodes and network performance.

For interested readers, several movies and snapshots of Animator are presented on the MobiREAL web page [36].

5.4. Tool support for existing simulators

The MobiREAL behavior simulator, MobiREAL Animator and MobiREAL UPF scenario generator can be used as independent programs. Therefore, instead of GTNetS, other simulators such as ns-2, Qualnet and GloMoSim may be used as the network simulator part of MobiREAL simulator. If the interaction between the behavior simulator and the network simulator is disabled, we can use the trace file generated by the behavior simulator by converting it into the corresponding inputs for the trace mobility of those simulators. But in this case, we cannot use application input/output (AI/AO in Section 4 in the CPE model.

If we wish to let the behavior simulator interact with those network simulators, we must add such an interaction mechanism to the network simulators. Usually, wireless network simulators including GTNetS have a "mobility class", in which behavior of mobile nodes is described. In the case of the MobiREAL simulator, we have enhanced the mobility class of GTNetS so that the network simulator can periodically replace the speeds and

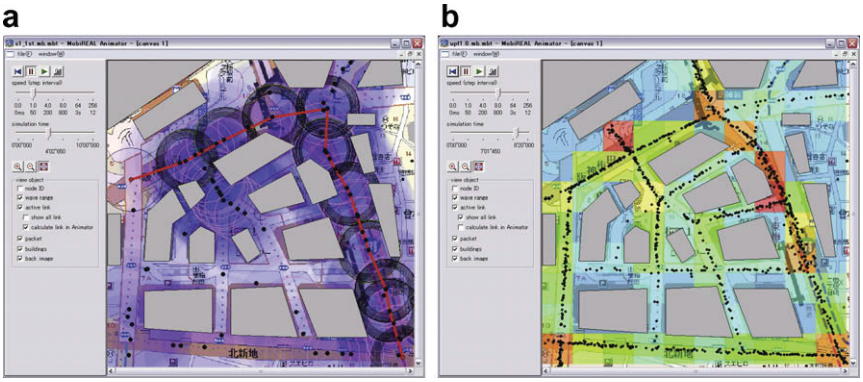

Fig. 5. Snapshots from MobiREAL Animator.

directions of nodes with the latest ones received from the behavior simulator. According to this concept, we have accomplished cooperative execution of GTNetS and the MobiREAL behavior simulator. In the case of GloMoSim, MobilityTrace class can be modified in the same way. Similarly, MobileNode class and PositionHandler class should be enhanced in the case of ns-2.

6. Experiments

We have carried out several experiments. Our objective is threefold. First, we would like to confirm that our UPF methodology and the related MobiREAL toolset can model the pedestrians' flows in the real world with high fidelity, without a great deal of manpower. For this purpose, we conducted fixed-point observation in downtown Osaka and derived the pedestrians' flows. The result of this experiment given in Section 6.1 has shown that our scheme could reproduce a position distribution very similar to the observed ones. Second, we would like to evaluate the trade-off between the fidelity and the computation cost of mobility by two experiments in Section 6.2. One evaluation is for the impact of the amount of given density information in the derivation of the UPF scenario, and another one is for the impact of the synchronization overhead between the behavior simulator part and the network simulator part of MobiREAL on the accuracy of simulation. Finally, we would like to show that the non-uniform distribution of the node movement and position affects performance of mobile wireless network systems. For example, in the real world, the location of wireless base stations considerably affects the quality of multi-hop wireless network services while it will have little impact with uniform movement and position distribution of nodes. This fact encourages us to assert the effectiveness of our UPF model presented in this paper. Thus, to prove the fact, we conducted performance evaluation of the following application and protocol: (i) real-time data streaming from a stationary base station to a mobile client via MANET; and (ii) a mobility management protocol for wireless mesh networks using the mobility prediction as in [37,38]. The results are given in Section 6.3.

6.1. Fidelity of reproduced pedestrian flows

We have measured the node density on each street in downtown Osaka. Using the obtained data, we have determined pedestrian flows based on our UPF technique described in Section 3 to see the similarity of the derived and observed density of pedestrians.

Two students observed the node density on the streets beginning at 14:00 on Sunday. At each observation point, we just took a photo with a digital camera, so the observation period itself was an instant. Next, we measured the width and length of the street captured in the photo. Finally, we cal-

Table 2
Measured node density on streets in downtown Osaka

Edge	Width	Density
9–8	7	0.033
8–7	8	0.059
7–34	8	0.041
34–5	8	0.041
5–4	4	0.025
4–2	10	0.013
15–13	6	0.005
13–10	6	0.005
10–6	8	0.030
14–15	12	0.008
15–16	8	0.006
16–32	12	0.012
32–17	12	0.012
17–18	12	0.013
24–19	12	0.032
19–20	12	0.035
20–21	12	0.023
21–22	12	0.027
8–14	8	0.045
14–19	12	0.026
19–23	8	0.058
3–7	14	0.060
7–10	6	0.065
10–31	8	0.046
31–16	8	0.045
16–20	12	0.008
10–12	12	0.015
18–22	12	0.026
12–17	12	0.012
17–21	12	0.012
11–30	8	0.040
30–12	8	0.040
0–5	14	0.034
5–6	7	0.050
6–11	12	0.060
11–18	12	0.021
1–4	12	0.002
4–6	5	0.030
30–26	3	0.013
30–27	3	0.066
28–31	3	0.025
32–29	3	0.012
34–33	3	0.053

width: m, density: person/m^2.

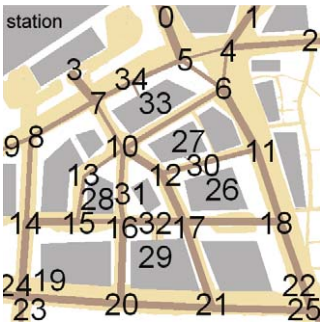

Fig. 6. Downtown Osaka.

culated the density of pedestrians using those values and the number of pedestrians in the photo. The obtained data is shown in Table 2. The map of downtown Osaka with the corresponding street graph is shown in Fig. 6. The size of the field is 500 m × 500 m and the graph includes 35 vertices and 44 edges. It took two students only about a half hour to obtain the average densities on the 43 streets.

As candidates for nodes' source or sink points (destination points), we selected 21 points located near shops, major intersections with subway entrances and the border of the field. We calculated the shortest path for each pair of the 21 destination points, and thus 210 routes were found as a total. We assumed that $v_0 = 1.39$ m/s and $D_{max} = 1.4$ person/m^2. Then we constructed 130 linear constraints that included 253 parameters (variables). lp_solve [35] was used to solve the given LP problem and the solving time was less than 1 s.

We show the result in Table 3. When we obtained the solution of the LP problem, the maximum error, which was minimized in the LP problem was 9.09% (0.102 person/s) and the mean error was 8.74% (0.030 person/s). We would like to note that since there is no absolute benchmark to compare with, there is no way to thoroughly show the fidelity of this result. However, in the general consensus we can say that this error is in the acceptable range.

These simulation settings and the obtained UPF mobility were also used in the experiments described in the following subsection.

Table 3
Derived flow rates

Routes	Flow rate
g0_11	0.375
g0_20	0.121
g0_24	0.036
g0_3	0.069
g1_18	0.026
g1_23	0.010
g1_33	0.005
g2_3	0.110
g2_11	0.057
g3_9	0.286
g3_11	0.105
g3_16	0.046
g3_23	0.326
g3_28	0.078
g4_11	0.108
g6_11	0.176
g6_16	0.258
g6_18	0.090
g6_23	0.048
g9_16	0.053
g9_25	0.006
g11_16	0.146
g14_21	0.002
g18_21	0.174
g18_23	0.027
g18_25	0.398
g18_27	0.145
g18_33	0.052
g20_23	0.101
g21_23	0.026
g23_24	0.027
g24_25	0.396
g24_28	0.002
g24_33	0.022
g26_29	0.029
g26_33	0.020
g27_33	0.096
g28_29	0.016

Flow rate: person/s, flow rates of all the other routes were zero.

6.2. Accuracy of mobility

Modeling mobility is to pursue the balance between fidelity and simplicity. In this section, we will show two experiments that evaluate the trade-off between accuracy and the computation cost of mobility. In Section 6.2.1, we evaluated the impact of the amount of given density information on the accuracy of UPF mobility. In Section 6.2.2, we evaluated the impact of the synchronization interval between the network simulator part and the behavior simulator part of MobiREAL (i.e., simulation overhead) on the accuracy of the simulation. We hope our experimental results are helpful for modeling work.

6.2.1. Density specified ratio and reproduction quality of derived flows

In this experiment, we evaluated the impact of the amount of given density information on the

Table 4
Density specified ratio and error of pedestrian rate

Number of specified streets		43	39	34	30	26	22	rwalk/ob
Number of constraints		130	122	112	104	96	88	–
Specified (ave.)	Person/s (%)	0.0304 (8.74)	0.0273 (7.81)	0.0204 (5.91)	0.0139 (3.98)	0.0112 (3.25)	0.00688 (2.04)	–
Specified (max)	Person/s (%)	0.102 (9.09)	0.102 (9.09)	0.102 (9.09)	0.101 (9.00)	0.0984 (8.82)	0.0938 (8.82)	–
All (ave.)	Person/s (%)	0.0304 (8.74)	0.0445 (13.7)	0.0645 (21.7)	0.0861 (31.2)	0.111 (43.7)	0.144 (59.5)	0.270 (253)
All (max)	Person/s (%)	0.102 (9.09)	0.449 (135)	0.626 (265)	0.738 (408)	0.801 (622)	0.933 (901)	0.825 (2170)

accuracy of UPF mobility. We observed the node densities on the 43 streets as mentioned in Section 6.1. In this section, we used the part of the observed 43 density information (from 22 up to 39 chosen randomly from the 43 density information) as an input of LP and evaluated the error of the density on the derived UPF flows. Table 4 shows the average/maximum error between the observed pedestrian rate and the derived one (|observed−derived|) and its ratio ($\frac{|\text{observed}-\text{derived}|}{\text{observed}} \times 100$). Here, "specified" is the error only on the streets where density was given, but "all" is the error on all streets. For the comparison we also simulated a modified version of the random walk mobility called random walk with obstacles (rwalk/ob) and counted the pedestrian rate of the model. In rwalk/ob, each node moves along the given street graph. At each corner, the node selects the neighboring corner at random and moves toward it. We used the same street graph in both UPF and rwalk/ob. We also show the error between the pedestrian rate of the rwalk/ob and the observed rate.

The number of LP constraints decreases as the amount of given density information (the number of specified streets) decreases, but the number of LP variables is the same for all cases. The error on the specified streets decreases as the number of specified streets decreases because the constraints are relaxed, but the error on the unspecified streets increases. The error on the unspecified streets varies from 0 to 900. The result shows that losing little density information tremendously increases errors at specific points. With more detailed analysis, we found that the observed density that extremely differs from the neighbors tends to cause large error if such density information is not given to the LP problem. Consequently, we should observe as many streets as possible, being especially careful about the streets with high or low node densities. As future work, we will study a method to fill in the unknown densities from the known densities by using empirical knowledge.

6.2.2. Time for interaction and position error

In this experiment, we evaluated the trade-off between the accuracy of the node position and the synchronization cost. In the simulation of MobiREAL, the behavior simulator holds accurate positions of nodes. The network simulator periodically synchronizes with the behavior simulator, receives the positions and velocity vectors of nodes, and calculates current nodes' positions using received values until the next synchronization. If the synchronization interval is too long, the velocity vector changed by the behavior simulator may cause a large error between the positions held by two simulators. However, if the synchronization interval is too short, the computation cost for the synchronization increases.

We used the mobility scenario derived in Section 6.1. The speed of the node followed uniform distribution between 1.0 m/s and 2.0 m/s. The simulation time was 100 s and the simulation granularity of the behavior simulator was set to 0.2 s. Results are shown in Table 5. Here, the error of the node position is defined as the error between the position which the behavior simulator holds and that which the network simulator holds. The error was sampled just before the network simulator updated its node position by the synchronization.

Obviously, there is a trade-off between the accuracy of the node position in the network simulator and the total computation time for the synchronization ("synchronization time" in Table 5). The error reaches zero when the synchronization interval is equal to the simulation granularity of the behavior

Table 5
Time for interaction and position error between network and behavior simulators

Synchronization interval (s)	0.2	0.4	1.0	2.0	4.0	10.0	20.0
Synchronization time (s)	100.3	58.5	26.2	13.5	7.0	3.2	1.9
Average error (m)	0	0.004	0.028	0.090	0.250	1.005	3.114
Maximum error (m)	0	0.789	2.816	5.722	13.497	35.765	66.372
Theoretical maximum (m)	0.8	1.6	4.0	8.0	16.0	40.0	80.0

simulator. For the maximum speed V of nodes and the synchronization interval T, the error must be less than $2VT$ (theoretical limit of the maximal error). So the synchronization interval is considered short enough if $2VT$ is far smaller than the communication range of the node. In our simulations, we set the synchronization interval to 1 s.

6.3. Performance evaluation of mobile wireless networks with UPF mobility

6.3.1. Application example 1: data streaming on MANETs

In this experiment, we carried out simulation of real-time data streaming with the assistance of MANETs. In the simulation scenario, real-time data such as movies are delivered by multi-hop relay from a short-range base station such as an Internet gateway to mobile clients. A route is established by the DSR protocol from the base station to each client. Through this scenario, we examined our expectation that different placements of base stations cause considerable performance distinction if realistic (thus non-uniform) position distribution is assumed, but little difference under uniform ones. If this is true, this fact emphasizes the necessity of realistic mobility.

Fig. 7a shows the simulation field. In the simulation, each client receives a 56 kbps movie for 180 s. We chose 120 clients at random at every 180-s interval. We compared our UPF mobility scenario modeled in Section 6.1 and random walk with obstacles (rwalk/ob). We used the same street graph in both UPF and rwalk/ob. We used IEEE802.11 DCF with the RTS/CTS mechanism as the MAC protocol. The radio range was set to 100 m. In the simulation, we used a simple radio attenuation considering line-off-sight propagation only. The obstacles were placed as shown by the grayed polygons in Fig. 7a.

We chose three candidate points for the location of the base station where node densities are quite different from each other in the UPF model. To measure the node density in the field, we may exploit the capability of the Animator, which can show the node density in each square section as shown in Fig. 7b. In this figure, the depth of gray in each section represents the degree of node concentration (i.e., deeper gray means higher density). Then we selected the three points A, B and C in Fig. 7a as the candidate positions for a base station placement. A, B and C correspond to high-, low- and middle-density sections, respectively. Then we measured the packet arrival ratio and the average path length to see the performance difference.

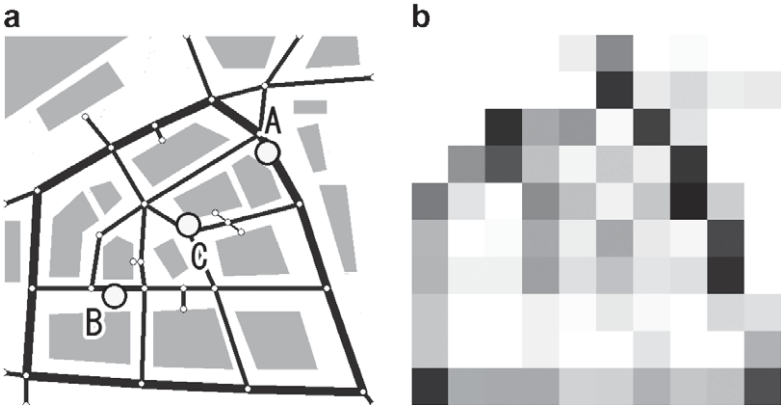

Fig. 7. (a) Simulation field and candidate points for location of base station. (b) Graphical presentation of node density (snapshot from the MobiREAL animator).

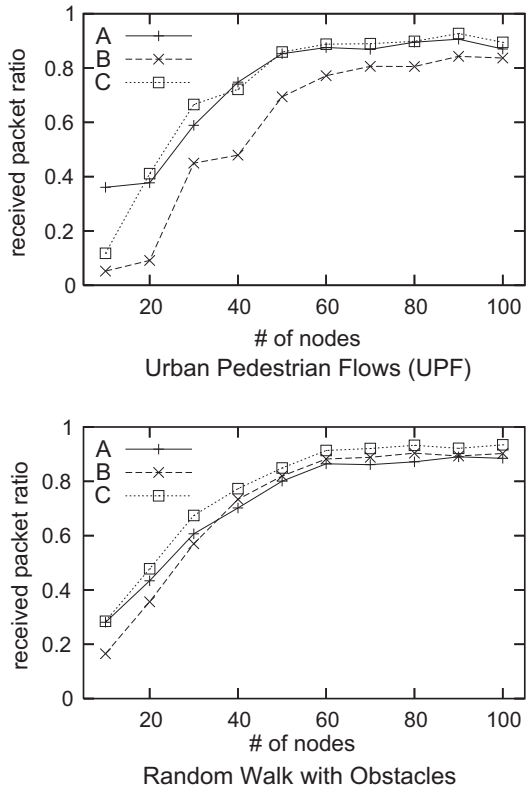

Fig. 8. Packet arrival ratio.

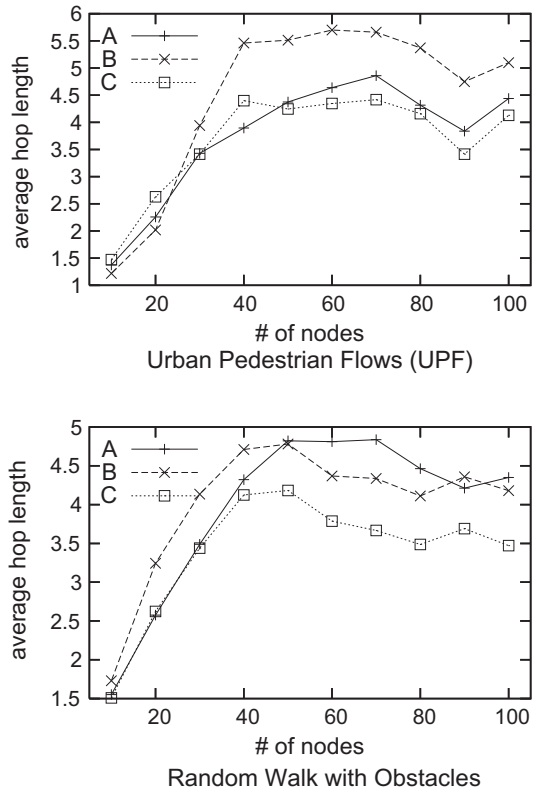

Fig. 9. Average hop length of DSR route.

Fig. 8 shows packet arrival ratio at the destination, and Fig. 9 shows the average hop length to the destination. As expected, the packet arrival ratio of each case with rwalk/ob is almost the same. On the other hand, we can see the difference in UPF mobility. The difference becomes small as the number of nodes increases, but still remains in the case of 80–90 nodes. The average of the hop length in Fig. 9 is small when the number of nodes is also small because it is hard to establish a route with many hops. When the number of nodes is larger than 50, base station C placed at the center of the field provides the shortest length of all the placements in rwalk/ob. Although, in UPF, there is a little difference between the hop length in the case of the base station C and that of the base station A placed at the upper right. It is known that as the density becomes higher, a route is shorter because one hop length can be longer in high node density situation. In addition, there are many nodes near the base station A so that the randomly chosen client walks near the base station A with a higher probability in UPF than in rwalk/ob. From these results, we found that base station C in the middle-density section could achieve almost the same packet arrival ratio as that of base station A in the high-density section, and that base station A could provide similar hop length with base station C in the center of the field.

Consequently, distribution of the density may influence the performance of the location-dependent applications such as location planning of base stations. It is necessary to use a mobility model that can reproduce realistic position distribution like the UPF model to evaluate the performance considering the characteristics of the target field.

6.3.2. Application example 2: prediction-based mobility management in wireless mesh network

As another example, we deal with mobility management for wireless mesh networks composed of a set of base stations. Each client connects to one of the base stations (mesh routers) to utilize the network. The radio ranges of a client and a router were 100 and 200 m, respectively. The placement of mesh routers is shown in Fig. 10 where polygons represent obstacles and others are walkways. The mesh rou-

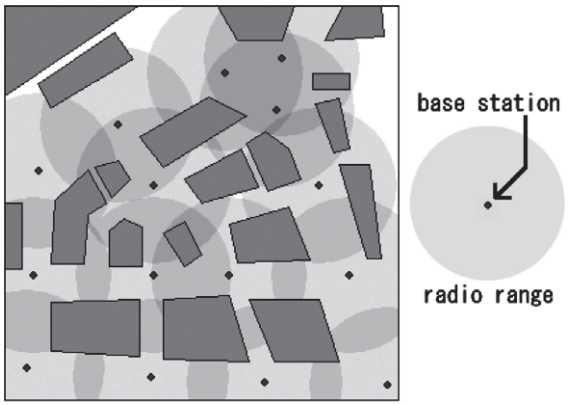

Fig. 10. Placement of mesh routers.

ters were placed to cover almost all the pedestrians' walkways.

The basic concept of mobility management schemes developed for cellular or mobile IP networks can be applied to wireless mesh networks [39]. In this experiment, we evaluate the efficiency of the prediction-based mobility management protocol that reduces the location updates by predicting the current location of mobile nodes from past locations. There are many protocols that utilize mobility prediction for mobility management [37,40,41]. The protocol of this experiment predicts mobility by the approach based on data mining like [38].

Each client holds its movement log as a sequence of neighbor routers like "$r_1, r_2, \ldots, r_n$". We assume that the client can recognize the neighbor router by a hello message or traffic for other clients. We also assume that each mesh router has enough movement logs of clients collected beforehand in its database. In this protocol, the client sends its movement log to the router in the location update process. The router predicts the movement of the client using its database. The client and the router utilize the prediction result for the next location update and paging, respectively.

The location update initiated by the client consists of the following steps:

1. First, the client that executes the location update sends a location update packet including its movement log to the neighbor router.
2. The router that receives the location update packet holds the client ID (e.g., IP address) and the current time. Then, the router searches for its holding movement logs that contain the same sequence as that in the received packet. Next, the router selects up to K routers to which the client likely moves (called candidate routers and K is given in advance) and sends the IDs of the routers to the client. The details of the router selection algorithm are shown in Fig. 11.
3. The client holds the received candidate router IDs.

If the client recognizes that he is entering an area covered by a non-candidate router, then the client initiates a location update. The router also holds the candidate routers of the client. If a request for a call was received, then all the areas of the candidate routers of the callee are paged at almost the same time.

In the experiment, we simulated the performance of this protocol in the UPF mobility and in the

```
a set of router sequence HLIST = φ;
for( each movement log A in database ){
  if( A includes same sequence of client's log ){
    HLIST = HLIST ∪ {post part of the sequence in A};
  }
}

a set of router CR, TMP = φ;
do{
  CR = CR ∪ TMP;
  TMP = φ;
  for( each A in HLIST ){
    TMP = TMP∪{pop_front(A)};
  }
}while( #(CR ∪ TMP) ≤ K ){

the candidate routers = CR;
```

Fig. 11. Algorithm to select candidate routers.

random walk with obstacles (rwalk/ob). We adjusted the node lifetime of rwalk/ob so that the distribution of the lifetime can almost be the same as that of UPF mobility. Other simulation settings are the same as in Section 6.3.1.

We calculated the total cost per call arrival. Here, U is the cost for paging the area of a router and V is the cost for a location update. Call-to-mobility ratio (CMR) represents the relative frequency of call rate and mobility and is defined as "an average stay time in the area that a router covers" divided by "an average interval of the call arrival". Fig. 12a shows the cost per call for $U = 10$, $V = 1$ and CMR $= 0.1$ as applied in [37,42]. The cost is smaller in UPF than in rwalk/ob for $K \leqslant 4$. The possible flow of the node is limited in the UPF so we can limit candidate routers. On the other hand, rwalk/ob has smaller cost for $K \geqslant 5$. Nodes in UPF go far away in a short time because they walk along the shortest path, but nodes in rwalk/ob decide movement randomly so some of them may return to the same point. This difference seems to affect the total number of the location update. Fig. 12b shows the cost per call for $U = 5$, $V = 1$, CMR $= 0.5$. K (the maximum number of the candidate routers) that has the smallest cost differs in the two mobility models.

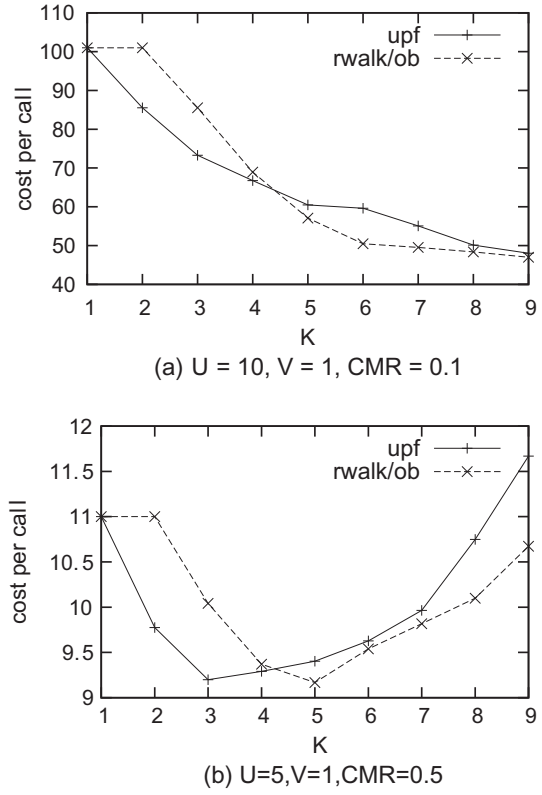

Fig. 12. Cost per call arrival.

Consequently, from these results, the movement routes of the nodes have an influence that may not be negligible in the performance evaluation, especially of the mobility-aware protocols. We should carefully choose the mobility model considering this influence.

7. Conclusion

In this paper, we have proposed a realistic mobility model called the UPF model and a behavior description model called the CPE model, and presented the facility of the MANET simulator MobiREAL, which uses our models for realistic simulation of MANET applications and protocols. By the UPF model, we can reproduce realistic pedestrian flows from data obtained by simple observation such as fixed-point observation using digital cameras or web cameras. To our best knowledge, no existing method considers this research direction. Also, through some experiments, we showed that we could design several MANET protocols and applications and verify their performance on the network topology, which was created by the UPF model and had never appeared in the existing random-based mobility models. In addition, using the CPE model, we can enhance the reality of the microscopic mobility such as stopping at traffic signals and slowing down due to congestion. The discussion about the reality and influence of this mobility still remains for future research, though we believe that our UPF/CPE framework provides reasonable fidelity of mobility modeling to conduct the performance evaluation of MANET systems in realistic environments.

Our future work includes more detailed validation of the influence and fidelity of our mobility models, comparison of our mobility models and real traces, and the evaluation of several mobile ad hoc communication protocols using the UPF/CPE mobility.

References

[1] T. Camp, J. Boleng, V. Davies, A survey of mobility models for ad hoc network research, Wireless Communications and Mobile Computing (WCMC) 2 (5) (2002) 483–502 (Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications).

[2] F. Bai, N. Sadagopan, A. Helmy, The IMPORTANT framework for analyzing the impact of mobility on perfor-

mance of routing for ad hoc networks, Ad Hoc Networks Journal 1 (4) (2003) 383–403.
[3] N. Sadagopan, F. Bai, B. Krishnamachari, A. Helmy, PATHS: analysis of PATH duration statistics and their impact on reactive MANET routing protocols, in: Proc. ACM MobiHoc, 2003, pp. 245–256.
[4] B. Ishibashi, R. Boutaba, Topology and mobility considerations in mobile ad hoc networks, Ad Hoc Networks 3 (6) (2005) 762–776.
[5] K. Maeda, K. Sato, K. Konishi, A. Yamasaki, A. Uchiyama, H. Yamaguchi, K. Yasumoto, T. Higashino, Getting urban pedestrian flow from simple observation: realistic mobility generation in wireless network simulation, in: Proc. 8th ACM/IEEE Int. Symp. on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), 2005, pp. 151–158.
[6] G.F. Riley, The Georgia tech network simulator, in: Proc. ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research, 2003, pp. 5–12.
[7] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in: Proc. ACM/IEEE MobiCom, 1998, pp. 85–97.
[8] C. Bettstetter, Mobility modeling in wireless networks: categorization, smooth movement, and border effects, ACM SIGMOBILE Mobile Computing and Communications Review 5 (3) (2001) 55–67.
[9] T. Chu, I. Nikolaidis, Node density and connectivity properties of the random waypoint model, Computer Communications 27 (10) (2004) 914–922.
[10] A. Rojas, P. Branch, G. Armitage, Experimental validation of the random waypoint mobility model through a real world mobility trace for large geographical areas, in: Proc. 8th ACM/IEEE Int. Symp. on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), 2005, pp. 174–177.
[11] E. Hyytia, P. Lassila, J. Virtamo, Spatial node distribution of the random waypoint mobility model with applications, IEEE Transactions on Mobile Computing 5 (6) (2006) 680–694.
[12] J. Yoon, M. Liu, B. Noble, Random waypoint considered harmful, in: Proc. IEEE Infocom, vol. 2, 2003, pp. 1312–1321.
[13] J. Yoon, M. Liu, B. Noble, Sound mobility models, in: Proc. ACM MobiCom, 2003, pp. 205–216.
[14] J. Yoon, M. Liu, B. Noble, A general framework to construct stationary mobility models for the simulation of mobile networks, IEEE Transactions on Mobile Computing 5 (7) (2006) 860–871.
[15] G. Lin, G. Noubir, R. Rajaraman, Mobility models for ad hoc network simulation, in: Proc. IEEE Infocom, vol. 1, 2004, pp. 454–463.
[16] W. Navidi, T. Camp, Stationary distributions for the random waypoint mobility model, IEEE Transactions on Mobile Computing 3 (1) (2004) 99–108.
[17] M. McGuire, Stationary distributions of random walk mobility models for wireless ad hoc networks, in: MobiHoc'05: Proceedings of the sixth ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2005, pp. 90–98.
[18] P. Nain, D. Towsley, B. Liu, Z. Liu, Properties of random direction models, in: Proc. IEEE Infocom, vol. 3, 2005, pp. 1897–1907.
[19] A. Jardosh, E.M. BeldingRoyer, K.C. Almeroth, S. Suri, Towards realistic mobility models for mobile ad hoc networks, in: Proc. ACM MobiCom, 2003, pp. 217–229.
[20] M. Hollick, T. Krop, J. Schmitt, H.-P. Huth, R. Steinmetz, Modeling mobility and workload for wireless metropolitan area networks, Computer Communications 27 (8) (2004) 751–761.
[21] W.-J. Hsu, K. Merchant, H.-W. Shu, C.-H. Hsu, A. Helmy, Weighted waypoint mobility model and its impact on ad hoc networks, ACM SIGMOBILE Mobile Computing and Communications Review 9 (1) (2005) 59–63.
[22] A.K. Saha, D.B. Johnson, Modeling mobility for vehicular ad-hoc networks, in: Proc. 1st ACM Workshop on Vehicular Ad Hoc Networks (VANET 2004), 2004, pp. 91–92 (poster paper).
[23] M. Musolesi, S. Hailes, C. Mascolo, An ad hoc mobility model founded on social network theory, in: Proc. ACM/IEEE Int. Symp. on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), 2004, pp. 20–24.
[24] X. Hong, M. Gerla, G. Pei, C.-C. Chiang, A group mobility model for ad hoc wireless networks, in: Proc. ACM/IEEE Int. Symp. on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), 1999, pp. 53–60.
[25] X. Zeng, R. Bagrodia, M. Gerla, GloMoSim: a library for the parallel simulation of large-scale wireless networks, in: Proc. ACM Parallel and Distributeal Simulation (PADS'98), 1998, pp. 154–161.
[26] ns-2, http://www.isi.edu/nsnam/.
[27] D. Kotz, K. Essien, Analysis of a campus-wide wireless network, in: Proc. ACM MobiCom, 2002, pp. 107–118.
[28] D. Tang, M. Baker, Analysis of a metropolitan-area wireless network, Wireless Networks 8 (2/3) (2002) 107–120.
[29] A. Balachandran, G.M. Voelker, P. Bahl, P.V. Rangan, Characterizing user behavior and network performance in a public wireless LAN, ACM SIGMETRICS Performance Evaluation Review 30 (1) (2002) 195–205.
[30] M. Balazinska, P. Castro, Characterizing mobility and network usage in a corporate wireless local-area network, in: Proc. ACM MobiSys, 2003, pp. 303–316.
[31] S. Thajchayapong, J.M. Peha, Mobility patterns in microcellular wireless networks, IEEE Transactions on Mobile Computing 5 (1) (2006) 52–63.
[32] M. McNett, G.M. Voelker, Access and mobility of wireless PDA users, ACM SIGMOBILE Mobile Computing and Communications Review 9 (2) (2005) 40–55.
[33] M. Kim, D. Kotz, S. Kim, Extracting a mobility model from real user traces, in: Proc. IEEE Infocom, 2006. URL http://www.cs.dartmouth.edu/dfk/papers/kim:mobility.pd%f.
[34] K. Konishi, K. Maeda, K. Sato, A. Yamasaki, H. Yamaguchi, K. Yasumoto, T. Higashino, MobiREAL simulator – evaluating MANET applications in real environments, in: Proc. 13th IEEE Int. Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), 2005, pp. 499–502.
[35] lp_solve, http://groups.yahoo.com/group/lp_solve/.
[36] MobiREAL Simulator Web Page, http://www.mobireal.net/.
[37] I.F. Akyildiz, J.S.M. Ho, Y.-B. Lin, Movement-based location update and selective paging for pcs networks, IEEE/ACM Transactions on Networking 4 (4) (1996) 629–638.
[38] G. Yavas, D. Katsaros, Özgür Ulusoy, Y. Manolopoulos, A data mining approach for location prediction in mobile

environments, Data Knowledge Engineering 54 (2) (2005) 121–146.
[39] I.F. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Computer Networks 47 (4) (2005) 445–487.
[40] T. Liu, P. Bahl, I. Chlamtac, Mobility modeling location tracking and trajectory prediction in wireless networks, IEEE Journal on Selected Areas in Communications 16 (6) (1998) 922–936.
[41] W.-S. Soh, H.S. Kim, Qos provisioning in cellular networks based on mobility prediction techniques, IEEE Communications Magazine 41 (1) (2003) 86–92.
[42] Y.W. Chung, D.K. Sung, A.H. Aghvami, Effect of uncertainty of the position of mobile terminals on the paging cost of an improved movement-based registration scheme, IEICE Transactions on Communications E86-B (2) (2003) 859–861.

Kumiko Maeda received her ME degree in Information and Computer Sciences from Osaka University, Japan in 2006. She is currently a Ph.D. student in the Department of Information Networking, Graduate School of Information Science and Technology, Osaka University. Her current research interests include mobile ad hoc networks, network simulation and mobility models. She is a student member of Information Processing Society of Japan (IPSJ).

Akira Uchiyama received his ME degree in Information and Computer Sciences from Osaka University, Japan in 2005. He is currently a Ph.D. student in the Department of Information Networking, Graduate School of Information Science and Technology, Osaka University. His current research interests include network security and localization in mobile ad hoc networks. He is a student member of IEEE and Information Processing Society of Japan (IPSJ).

Takaaki Umedu received his ME degree in Informatics and Mathematical Sciences and the Ph.D. degree in Information Science from Osaka University, Japan in 2001 and 2004, respectively. He is currently an Assistant Professor at Osaka University. His current research interests include design and implementation of distributed systems and mobile systems. He is a member of Information Processing Society of Japan (IPSJ).

Hirozumi Yamaguchi received his BE, ME and Ph.D. degrees in Information and Computer Sciences from Osaka University, Japan. He is currently an Associate Professor at Osaka University. His current research interests include design and implementation of distributed systems and communication protocols, especially protocols and services on mobile ad hoc networks and application layer multicast. Also, he is working on some issues of wireless network simulations. He is a member of IEEE.

Keiichi Yasumoto received the BE, ME and Ph.D. degrees in Information and Computer Sciences from Osaka University, Osaka, Japan, in 1991, 1993 and 1996, respectively. He joined the faculty of Shiga University in 1995. Since 2002 he has been an Associate Professor of Graduate School of Information Science at Nara Institute of Science and Technology. His current research interests include mobile computing, ubiquitous computing, and distributed systems. He is a member of IPSJ, ACM and IEEE/CS.

Teruo Higashino received his BS, MS and Ph.D. degree in Information and Computer Sciences from Osaka University, Japan in 1979, 1981 and 1984, respectively. He joined the faculty of Osaka University in 1984. Since 2002, he has been a Professor in Graduate School of Information Science and Technology at Osaka University. His current research interests include design and analysis of distributed systems, communication protocol and mobile computing. He is a senior member of IEEE, a fellow of Information Processing Society of Japan (IPSJ), and a member of ACM and IEICE of Japan.