Getting Urban Pedestrian Flow from Simple Observation:
Realistic Mobility Generation in Wireless Network Simulation

Kumiko Maeda, Kazuki Sato, Kazuki Konishi, Akiko Yamasaki, Akira Uchiyama,
Hirozumi Yamaguchi, Keiichi Yasumoto† and Teruo Higashino

Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan

{k-maeda,kz-satou,k-konisi,a-yamasaki,utiyama,h-yamagu,higashino}@ist.osaka-u.ac.jp

†Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, Japan

yasumoto@is.naist.jp

ABSTRACT

In order for precise evaluation of MANET applications, more realistic mobility models are needed in wireless network simulations. In this paper, we focus on the behavior of pedestrians in urban areas and propose a new method to generate a mobility scenario called *Urban Pedestrian Flows* (UPF). In the proposed method, we classify pedestrians in a simulation field into multiple groups by their similar behavior patterns (simply called flows hereafter, which indicate how they move around geographic points). Given the observed road density in the target field, we derive using linear programming techniques how many pedestrians per minute follow each flow. Using the derived flows, we generate a UPF scenario which can be used in network simulators. In particular, we have enhanced a network simulator called *MobiREAL*, which has been developed in our research group, so that we can generate and use the UPF scenario. MobiREAL simulator has three main facilities: the behavior simulator, network simulator and animator. The behavior simulator can generate/delete mobile nodes according to the UPF scenario. The network simulator can simulate MANET protocols and applications. The animator offers elegant visualization of simulation traces as well as graphical user interfaces for facilitating derivation of UPF scenarios. Through several case studies, we show similarity of the derived flows to the observed ones, as well as the metrics that characterize the mobility of the scenario.

Categories and Subject Descriptors

I.6.5 [**Simulation and Modeling**]: Model Development

General Terms

Performance, Design, Experimentation

Keywords

Mobile Ad-hoc Network, Simulation, Mobility Model

MSWiM'05, October 10–13, 2005, Montreal, Quebec, Canada.

1. INTRODUCTION

Designing effective MANET protocols and applications for given environments is a very complicated task since it is hardly possible to build large-scale and realistic testbeds in the real world for performance evaluation. Instead, network simulators are mainly used nowadays. Recently, in the research area of MANET simulation, reality of simulation is considered to be very important, and several mobility models of mobile nodes have been proposed (see [1–3] for survey).

There are three approaches toward realistic mobility models. The first approach is remodeling of simple mobility models such as the Random Way Point (RWP) model [4]. Such simple mobility models can commonly be available in many network simulators and make the analysis of target MANET applications rather simple. They can be used for traffic modeling and performance evaluation by specifying mobility metrics such as density and average speed. Ref. [1] presents a variant where nodes can change their directions smoothly, keeping analytical properties. Ref. [5] presents "sound mobility model" based on the RWP model. This model is designed to keep the average velocities of nodes to exclude harmful effects to simulation results.

The second approach is synthesizing realistic mobility models [6–11]. Ref. [6] introduces obstacles (i.e., buildings) and pathways between those obstacles in a given simulation field. Any polygon can be placed in the space and a realistic pathway between any two obstacles can automatically be generated using a Voronoi graph computation algorithm. The research group also provides plug-ins for GloMoSim and ns-2 simulators for the utilization of the proposed mobility. Ref. [7] proposes a macroscopic mobility model for wireless metropolitan area networks, where a simulation space 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. This paper mainly focuses on calculating density of each zone. It does not consider how nodes move in and between these zones. Ref. [8] presents 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 in order to model nodes' movement between these regions. Because the main goal of this method is to model realistic "density" in these regions which are close to each other, pedestrian flows between these regions are really simplified, that is, the real "behavior", which means movement of pedestrians around regions through different pathways, cannot be modeled.

The third approach is using real traces of mobile users [12–15]. In the work of this category, user behavior patterns were analyzed by collecting real traces of users. In Ref. [15], authors have collected traces from wireless network at the ACM SIGCOMM2001 conference place, and given several analysis results. Similarly, Ref. [14] has collected traces from wireless network users in a metropolitan area wireless network. Also, Ref. [12, 13] give trace results and analysis in a campus-wide wireless network. Those approaches can simulate the detailed real world. However, they require a very large amount of information collection to produce a mobility scenario.

Here, our aim is to *reproduce the pedestrian flows of the real field in the simulation*. For such a purpose, we need the full trace of behavior as done in Ref. [12–15]. However, we think that the investigation of behavior patterns in the real world should be simple enough due to cost, privacy and some other reasons. Thus we need a method to reproduce pedestrian flows from simple observation such as density observation on streets, which can be taken by fixed point observations using web cameras, several (not so many) volunteers and so on.

In this paper, we propose a new method to generate a mobility scenario called *Urban Pedestrian Flows* (UPF) to simulate real movement of pedestrians in urban areas in wireless network simulation. We classify pedestrians in the field and specify their typical behavior patterns as their routes (hereafter, we call these routes *flows*). Also we assume that average densities of pedestrians on a part of streets in the field are given. These can be simply obtained by fixed point observations and so on. Then using linear programming (LP) techniques, for each flow we calculate how many pedestrians follow the flow per unit of time, minimizing the maximum error between the road density observed and the road density derived from the LP solution. If the maximum error is small, we decide that the given behavior patterns can roughly represent the pedestrian flows in the target field.

Using the derived flows, we generate a UPF scenario which can be used in network simulators. In particular, we have enhanced a network simulator called *MobiREAL* [16,17], which has been developed in our research group, so that it can generate and use the UPF scenario. MobiREAL simulator has three main facilities: the behavior simulator, network simulator and animator. The behavior simulator can generate/delete mobile nodes according to the UPF scenario. The network simulator can simulate MANET protocols and applications. The animator offers elegant graphical view of simulation traces as well as offering graphical user interfaces for facilitating derivation of UPF scenarios. Through several case studies, we show similarity of the derived flows to the observed ones, as well as the metrics that characterize the mobility of the scenario.

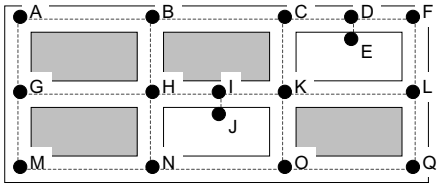


Figure 1: Simulation field graph.

This paper is organized as follows. Section 2 presents our method to generate a UPF scenario. Section 3 describes the design and implementation of our MobiREAL simulator to incorporate UPF scenario generation facility. The possibility to integrate the facility into the other network simulators is also discussed. Section 4 gives experimental results, and Section 5 concludes the paper.

2. GENERATING URBAN PEDESTRIAN FLOWS

In this section, we focus on the behavior of pedestrians in urban areas and propose a new method to generate an Urban Pedestrian Flow scenario.

2.1 Inputs

2.1.1 Simulation Field

A simulation field contains polygons that represent obstacles such as buildings. Between these polygons, an undirected graph $G = (V, E)$ is specified where geographic points and streets are represented as nodes and edges, respectively (Fig. 1). The geographic points include intersections, entrances of buildings, stations, terminals, shopping centers and so on. For each edge $e_{ij} \in E$, the width W_{ij} is also given.

We also give a partial set $V'(\subseteq V)$ of geographic points. These points are called *specific points* where persons may appear or disappear. Thus the specific points are regarded as starting points or destination points of pedestrian flows explained below. These points include, for example, entrances of buildings, stations, terminals, underpasses and shopping centers.

2.1.2 Pedestrian Behavior Patterns

In an urban area, many types of pedestrians are walking around for different purposes. Commuters usually get off trains and go straight to their transfer stations or to their offices. Shoppers visit their favorite shops and stay those shops for a while. People with enough time may stop by and stay for a while at a shopping center. Based on the above observation, we classify their behavior patterns into multiple groups. Then for each group, we estimate their behavior patterns as a route (this is called a *flow*). For example, persons in a group of shoppers may usually start from the station and visit a shopping center, and if they still have time, they may visit another department store near the shopping center. Actually, we give a set P of flows on G. Each flow is an acyclic sequence of geographic points in V where the both ends must be specific points in V'.

However, it may sometimes be difficult to estimate the behavior patterns. For such a purpose, P can be calculated by a known algorithm such as Bellman-Ford. The Bellman-Ford algorithm calculates the shortest paths between points, but actually persons may not follow the shortest path to move from point to point. For example, shopping customers may walk on the large streets or shopping streets to enjoy walking, rather than narrow streets which are the shortest to the destination. In such a case, the results of Bellman-Ford algorithm can be used to compute different routes by concatenating two or more shortest paths.

2.1.3 Observed Density

We only need to investigate average density of pedestrians on part of streets (edges) in the simulation field. We do not need the detailed survey reports from pedestrians about their routes. We let $E_M(\subseteq E)$ denote the set of edges whose densities are observed. For each edge $e_{ij} \in E_M$, we let D_{ij} denote the observed density on the edge.

2.2 Flow Rate Derivation Algorithm

Given an undirected graph G, the width W_{ij} of each edge $e_{ij} \in E$, the set V' of specific points, a set P of flows between specific points, and the observed density D_{ij} of each edge $e_{ij} \in E_M$, the algorithm derives the *flow rate* of each flow in P, minimizing the maximum error between the derived and observed densities on the edges. Here the flow rate is defined as the generation rate of nodes per unit of time at the starting point of the flow.

In general, for the density d of a street, the pedestrian rate g *on the street* (i.e., the number of pedestrians going into the street per second), the speed v of persons on the street and the width w of the street, the following equation holds.

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

From the research results in Ref. [18], the speed v of a pedestrian on a street follows the negative increase of the density on the street. Therefore,

$$v = k * d + v_0 \tag{2}$$

where k and v_0 are negative and positive constants, respectively. Also from Ref. [18], on any (very) crowded street, almost the same density (say D_{max}) was observed when the speeds v of pedestrians are very close to 0. Therefore, from (2), we obtain

$$k * D_{max} + v_0 = 0 \tag{3}$$

Finally, from inequalities (1), (2) and (3), we obtain the following pedestrian rate calculation function G on a street from the density d and width w of the street.

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

To formulate our problem as an LP problem, we introduce a variable f_k which represents the flow rate of each flow $p_k \in P$ and a variable g_{ij} which represents the pedestrian rate on each edge $e_{ij} \in E$. Here, the flow p_k is represented as a sequence of edges. We also introduce a variable δ, which means the maximum error between the pedestrian rate calculated from the observed density and the width of a street and the derived pedestrian rate.

2.2.1 Objective Function

Our objective is to minimize the maximum error δ.

$$\min \delta \tag{5}$$

2.2.2 Constraints

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

$$\forall e_{ij} \in E;\ \ g_{ij} = \sum_{p_k \in P \wedge e_{ij} \in p_k} f_k \tag{6}$$

The following inequality limits the errors of pedestrian rates between the observed and derived ones on each edge up to δ, where the function G was given in (4).

$$\forall e_{ij} \in E_M; -\delta \leq \frac{G(D_{ij}, W_{ij}) - g_{ij}}{G(D_{ij}, W_{ij})} \leq \delta \tag{7}$$

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

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

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

$$g_{ij} \leq \frac{v_0 * W_{ij} * D_{max}}{4} \tag{8}$$

By solving this LP problem, we obtain the flow rate f_k for each flow $p_k \in P$ that minimizes the maximum error δ between the observed and derived pedestrian rates of the streets. Using the derived flow rates, a UPF scenario can automatically be generated.

We conducted simulation experiments to see how accurately our generation method can generate realistic pedestrian flow rates from the density actually observed in Osaka city downtown. The result will be given in Section 4.3.

3. TOOL SUPPORT

We have implemented our scenario generation function into MobiREAL simulator [17]. MobiREAL simulator is briefly outlined in Section 3.1. We have implemented the scenario generator in C++ language which co-works with MobiREAL simulator. This is explained in Section 3.2. We also state the possibility for applying this tool support to the other existing simulators to make our mobility model widely available for researchers and developers in this area. This is discussed in Section 3.3.

3.1 Overview of MobiREAL Simulator

MobiREAL simulator is composed of three main parts called *MobiREAL behavior simulator*, *MobiREAL network simulator* and *MobiREAL animator*. We have extended the network simulator GTNetS [19] developed in Georgia Institute of Technology where dynamic node generation and deletion are supported, and node positions, speeds and directions can be updated from the behavior simulator. For simulation of network and MAC layer protocols, our MobiREAL

network simulator relies on GTNetS where DSR, AODV and NVR (wireless form of Nix-Vector routing) are supported in the network layer and the standard IEEE802.11 DCF (CSMA/CA) with RTS/CTS is supported in the MAC layer. We have enhanced the signal propagation model in the original GTNetS so that obstacles are taken into account. It has a similar propagation model to Ref. [6] where ground reflection is considered as zero propagation and line-of-sight is considered.

MobiREAL animator visualizes trace of simulation. Visualization of simulation results is one of the most important facilities for network simulators. In particular, visualizing mobility of nodes is mandatory to convince simulator users that the realistic behavior of mobile nodes is achieved. Our MobiREAL animator works on the Microsoft Windows platforms, and has been developed using DirectX 9. Each mobile node is represented as a small circle, and a semi-transparent concentric circle represents its radio range. Packet transmission is animated by concentrically growing circles with colors. To see an example demonstration of MobiREAL animator, readers may refer to our MobiREAL web page [16].

The behavior and network simulators are two independent programs that periodically exchange necessary data through a TCP channel. When they are initialized, they hold the simulation field and node objects independently. The behavior simulator calculates the latest positions, directions and speeds of nodes, and send them to the network simulator. Messages exchanged among nodes are also sent to the network simulator. The node objects in the network simulator hold their positions, directions and speeds. The network simulator conducts network simulation, and packet exchange among those node objects is simulated based on the mobile nodes' signal propagation model.

3.2 Tool Support for MobiREAL Simulator

As the inputs to MobiREAL behavior simulator, it requires (i) a simulation field model, (ii) a set of behavior models of mobile nodes (microscopic models) and (iii) a simulation scenario. (i) is compatible with our simulation field model in Section 2.1, and the detailed description of (ii) and (iii) are given in Ref. [17]. We generate a UPF scenario as (iii).

We have enhanced our MobiREAL animator so that developers can interactively specify simulation fields. This is a very sophisticated function which well supports the developers. As we stated earlier, a simulation field has polygons that represent obstacles and a graph that represents points (vertexes) such as intersections and destinations, and streets (edges). Showing a BMP graphic map file, the developers can specify, by their mouse devices, a sequence of points that form a polygon with signal propagation ratio through the obstacle, and a set of points as graph vertexes. Two adjacent points can be selected to make a street between them, with two attributes (the width of the street and the observed density of nodes needed for scenario generation).

Using the field model with the observed density generated through MobiREAL animator and the observed density on the streets, MobiREAL scenario generator, which has been newly developed for this work, generates the simulation scenario according to our method presented in Section 2. MobiREAL scenario generator works as a preprocessor of MobiREAL behavior simulator, and generates a scenario using a linear programming problem solver *lp_solve* [20]. MobiREAL behavior simulator and MobiREAL network simulator make use of the field model and the UPF scenario.

3.3 Tool Support for Existing Simulators

MobiREAL behavior simulator, MobiREAL animator and MobiREAL scenario generator can be used as independent programs. Therefore, instead of GTNetS, other simulators such as ns-2 and GloMoSim may be used as the network simulator part of MobiREAL simulator. Here we will describe modification to be made to generic discrete event simulators in order to make them work as the MobiREAL network simulator part. We note that these simulators like ns-2 and GloMoSim have trace mobility where we can specify trace of node positions. This can be used for "static" mobility where node positions can be known in advance. However, node behavior in MobiREAL simulator may be "dynamic", *i.e.* the behavior model allows nodes to change their behavior dependently on the environment (*e.g.*, nodes may decrease their speeds in crowded space) [16,17]. Thus periodical update of node positions, speeds and directions during the execution of scenarios is mandatory in our mobility model.

Usually, wireless network simulators including GTNetS have a "mobility class", in which behavior of mobile nodes is described. In the case of MobiREAL simulator, we have enhanced the mobility class of GTNetS so that the network simulator can replace the speeds and directions of nodes with the latest ones received from the behavior simulator. Once the network simulator receives this information, it updates the positions/speeds/directions of nodes with this information and use them for simulation until it receives new information from the behavior simulator. According to this concept, we have accomplished cooperative execution of GTNetS and MobiREAL behavior simulator. Readers who are interested in the detailed design may refer to [16,17].

In the case of GloMoSim *MobilityTrace* class can be modified in the same way as we described above. Similarly, *MobileNode* class and *PositionHandler* class should be enhanced in the case of ns-2.

4. EXPERIMENTAL RESULTS

In order to see the impact of our mobility model, we present performance evaluation of two typical MANET protocols/applications (DSR and information dissemination application) using MobiREAL simulator. These applications have been also used in Ref. [17] to validate the usefulness of MobiREAL simulator. Enhancing from the work in Ref. [17], in this paper, we carried out new experiments to measure the metrics of two categories.

The metrics in the first category are those presented in Ref. [21], which characterize graph connectivity and mobility. For example, link changes (*i.e.* the number of link creations between two nodes) and link duration (*i.e.* the longest time interval during which two nodes are in the transmission range of each other) directly affect the performance of DSR, and the metrics such as degree of spatial dependence (*i.e.* similarity of velocities of two nodes), degree of temporal dependence (*i.e.* similarity of velocities of two time slots of a node), and relative speed of two nodes affect the performance of information dissemination application since nodes in the application store information and forward it to neighbors according to certain policies, like information dissemination in vehicular ad hoc networks [22].

The metrics in the second category are introduced to mea-

sure the "reality" of our UPF mobility scenario. Here "reality" means how the derived pedestrian rates are similar with the ones observed on the real streets. We have watched and observed, for a part of streets in the simulation field, the densities and widths of the streets. These values are used in the experiments of this category to see the reality of UPF scenarios.

4.1 Simulation Settings and Applications

As the common settings, we have modeled a real 500m × 500m region including buildings in downtown Osaka city. Then we have observed the pedestrians on a part of the streets for about a half of one hour to obtain the average density of pedestrians. The generic pedestrian model presented in our previous work [17] was used as the behavior model of each node. Using these values and models, we have solved the linear programming model presented in Section 2 and the obtained UPF scenario was used in the simulation.

We have measured the performance of two applications on the UPF mobility and a modified version of RWP mobility called RWP with obstacles (denoted as RWP/ob). In RWP/ob, each node moves between intersections. At each intersection, the node randomly decides the next direction except the backward direction. The node density of RWP/ob model is controlled so that it has almost the same number of nodes as our UPF scenario. The applications are (i) DSR routing and (ii) information dissemination.

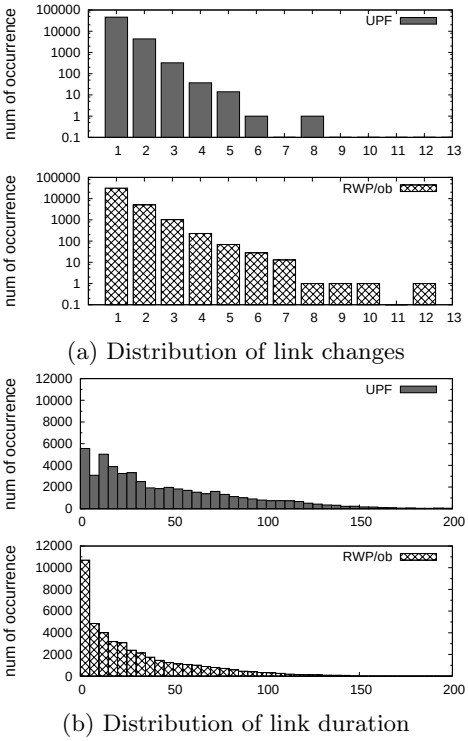
In DSR, the simulation time was 720 seconds. The communication was done by UDP from time 240 sec. to time 720 sec. The initial speeds of pedestrians were 2m/s. After the connection was established, 10 kbps traffic was generated interactively between the two application users to simulate interactive real-time communication. We used IEEE 802.11 DCF with RTS/CTS as the MAC protocol. The radio range was set to 100 meters.

In the information dissemination, we have used the implementation of the following application running on each mobile node; (i) if a mobile node receives information broadcasted by a short-range base-station or a neighboring node, it caches the information, (ii) for every 15 seconds, it checks whether it should broadcast the cached information to its neighbors or not, according to the predefined probability function $Prob(P, S)$ called *diffusion policy* (explained later) where P and S denote the positions of the node and base station, respectively, and (iii) it discards the cached information after 60 seconds or once it broadcasts the information (therefore, each node has at most 4 chances to broadcast information). As the diffusion policies, we provide the following three $Prob_i$ $(i = 1, 2, 3)$;

1. $Prob_1(P, S) = \frac{d_{max} - d(P,S)}{d_{max}}$ where $d(P, S)$ is Euclid distance between P and S and d_{max} is the maximum Euclid distance of the field (in our simulation field $d_{max} = 500\sqrt{2}$). This diffusion policy indicates that if the node is located closer to the base station, it broadcasts the information with higher probability.

2. $Prob_2(P, S) = \frac{d(P,S)}{d_{max}}$. This is the opposite of the diffusion policy $Prob_1$.

3. $Prob_3(P, S) = 0.5 * C(d(P, S) \leq d_{const})$. The value of $C(t)$ is 1 only if the condition t is true, otherwise 0. We have set $d_{const} = 150$m. This diffusion policy means that the node broadcasts the information with probability 0.5 if the distance $d(P, S)$ is no longer than 150m.



(a) Distribution of link changes

(b) Distribution of link duration

Figure 2: Graph connectivity metrics.

The base station was placed near the center of the field. The simulation time was 600 seconds, and the application was executed between time 180sec. and 600sec. Any other setting basically follows the first example.

4.2 Graph Connectivity and Mobility Characteristics

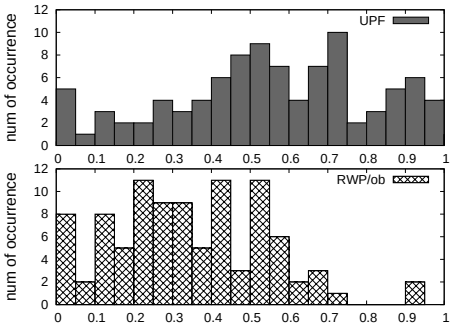
4.2.1 Graph Connectivity Metrics

At first, we have measured graph connectivity metrics; (i) link changes (the number of link creations between two nodes) and (ii) link duration (the longest time interval during which two nodes are in the transmission range of each other). The distributions of these metrics are shown in Fig. 2.

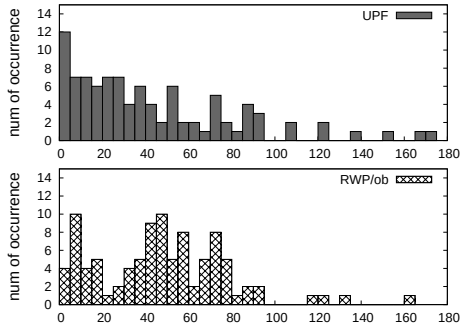
RWP/ob model has several cases with larger link changes compared with our UPF mobility. This is natural because in the UPF mobility, more neighboring nodes are going to the same destination than RWP mobility and thus link changes do not occur many times. This observation is endorsed by the link duration result, where UPF has longer durations clearly.

In the same experiments, we have measured several metrics that show the performance of DSR. We have selected 2 application users in the same pedestrian flow but away from each other. The results are shown in Fig. 3.

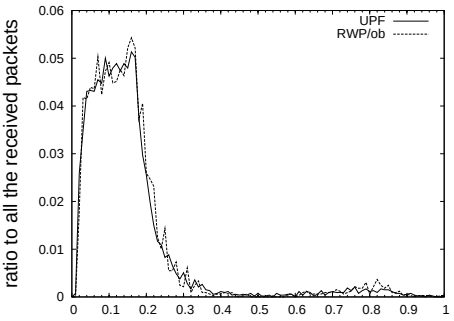
In the UPF scenario, a DSR route was established along the flow. Therefore, the route is stabler than RWP/ob. This observation is endorsed by Fig. 3(a) and Fig. 3(b). The packet arrival ratio becomes lower and the number of



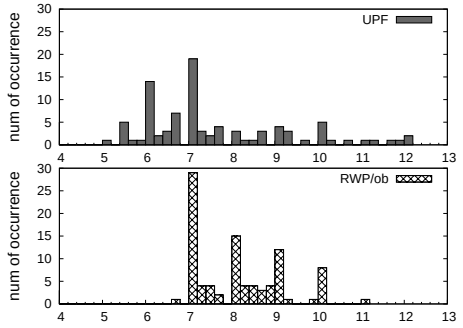
(a) Distribution of packet arrival ratio (captured during every 5s and the averages per second are shown)



(b) Distribution of number of DSR control packets (captured during every 5s)



(c) Distribution of packet delay



(d) Distribution of DSR hops (captured during every 5s and the averages per second are shown)

Figure 3: DSR performance.

control packets is larger if the route is instable, since there are many disconnections of the route. In Fig. 3(c), both UPF and RWP/ob have similar characteristics, since the distances between two application nodes were almost the same in both scenarios. Finally in Fig. 3(d), RWP/ob has similar hops in many cases, since the density on any street was almost the same among all streets. On the other hand, UPF has different density among streets, and therefore they are distributed widely.

4.2.2 Metrics Characterizing Mobility

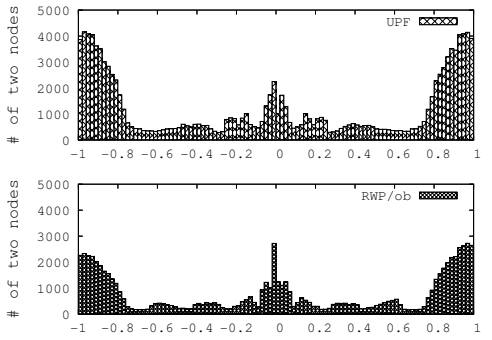
Secondly, we have measured the mobility characteristics metrics using the information dissemination scenario; (i) degree of spatial dependence (difference of velocities between two nodes whose distance is less than 50m), (ii) degree of temporal dependence (difference of velocities between two time slots of a node), and (iii) relative speed of two nodes whose distance is less than 50m. The results are shown in Fig. 4. In Fig. 4(a) and Fig. 4(c), the results of UPF and RWP/ob are very similar to each other. This is because behavior is almost the same while nodes are moving between intersections in both cases. However, total distribution of RWP/ob is less than that of UPF because of the difference in the density of nodes. There is a deviation on the density of nodes depending on places in UPF while it is average in RWP/ob. As a deviation increases, more pairs of nodes become the target of measurement. On the other hand, in Fig. 4(b), there is big difference. In UPF, each node does not change his/her direction at intersections because each node goes to his/her destination through the shortest path. Consequently, its dependence becomes very high though it is not in RWP/ob.

We have also measured the information diffusion ratio of the information dissemination application, which means how many nodes obtained the information. Fig. 5 shows the result. We can see the effect of mobility models to such an application.

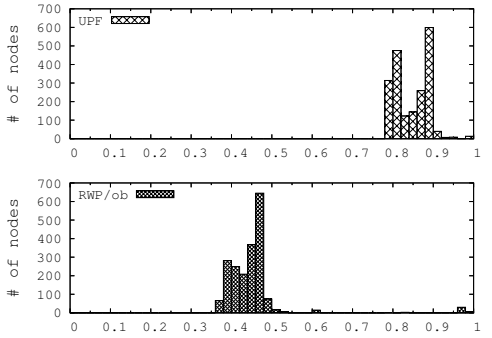
4.3 Similarity with Observed Density

We measured density of each street in downtown Osaka city. With the measured density, we generated pedestrian flows based on our technique in Section 2. The map of downtown Osaka city and the corresponding graph are shown in Fig. 6. The graph includes 35 vertexes and 44 edges. We took photos of the streets with digital cameras, and calculated the density of each street from the number of pedestrians in the street, the width and the length of the street. The measured data is shown in Fig. 6.

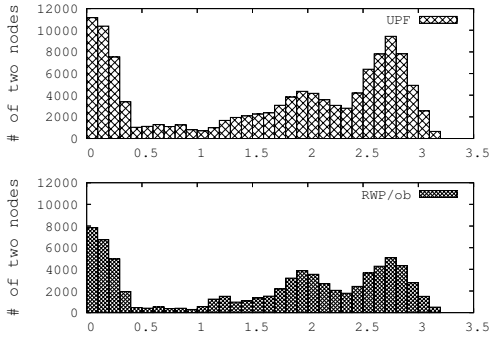
In order to decide possible routes of pedestrians, first we selected candidate destination points. As candidates, we selected points located near the edges of the map, points corresponding to entrances of shops and popular intersections with entrances to subways. We calculated the shortest path between each pair of those selected points. The calculated routes were used as the candidate routes. We selected 20 candidate destination points. So the number of all candidate routes was 190. According to Ref. [18], we assumed that $v_0 = 1.39m/s$ and $D_{max} = 1.4person/m^2$. With the proposed technique, we constructed the LP which has 192 parameters (variables) and 130 constraints. lp_solve [20] was used to solve the given LP. In our experiment, we could obtain the solution of the LP in a moment. When we obtained



(a) Degree of spatial dependence



(b) Degree of temporal dependence



(c) Relative speed of two nodes

Figure 4: Mobility metrics.

the solution of the LP, the maximum error, that is equal to the objective function of the LP, was 35.4352 % (0.398 person/sec). Since this error is too big, we modified the objective function and part of constraints as follows.

$$\min \sum_{E} \delta_{ij} \tag{9}$$

$$\forall e_{ij} \in E_M : -\delta_{ij} \leq \frac{G(D_{ij}, W_{ij}) - g_{ij}}{G(D_{ij}, W_{ij})} \leq \delta_{ij} \tag{10}$$

We can identify the bottleneck edges (i.e., edges with much larger errors than the average) by using this objective function which just sums up errors in all edges. When we solved the modified version of the LP, we obtained errors in edges as shown in Table 1 (left side). In this case, the value of the objective function was 2.24428, and the maximum error was 0.341 person/sec (68.2%) at edge $e_{10,31}$. We see that the reproduced density values on edges $e_{7,10}$, $e_{6,10}$, $e_{10,31}$ and $e_{31,16}$ were smaller than the measured ones.

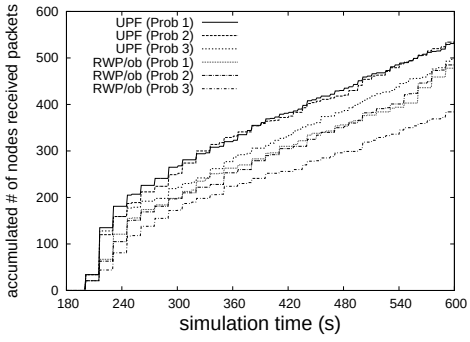


Figure 5: Application performance (information diffusion ratio vs. simulation time).

Table 1: Error between Measured and Reproduced Pedestrian Rate

	first trial	second trial
value of objective function	2.24428	0.279183
maximum error (person/s)	0.341 (68.2%)	0.062 (16.0%)
average error (pearson/s)	0.024 (5.2%)	0.002 (0.6%)

We heard from a local person that there is a popular entrance to the underground city near vertex v_{16}. So, we added v_{16} to candidate destination points. Then, the number of candidate destination points and the candidate routes became 21 and 210, respectively. The constructed LP includes 253 parameters. We show the derived errors in Table 1 (right side).

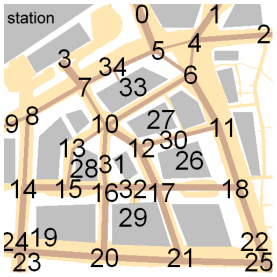
As a result, the value of the objective function was 0.279183. The maximum error value was 0.062 person/sec at edge $e_{7,10}$ and the maximum error in percentage was 16 % at $e_{14,15}$. In this case, however, the larger errors appear in specific edges. To avoid this situation, we solved the LP with the objective function (5) and the constraint (7).

We show the obtained result in Table 2. Here, the maximum error was reduced to 8.67925 % (0.097 person/sec).

5. CONCLUSION

In this paper, we have proposed a new method to generate a mobility scenario called Urban Pedestrian Flows (UPF). Given pedestrian flows in a simulation field and the observed density on the streets, we derive using linear programming techniques the rates of these pedestrian flows (how many pedestrians follow the flows per unit of time). Using the derived flow rates, we generate a UPF scenario which can be used in network simulators. In particular, we have enhanced our wireless network simulator MobiREAL so that the inputs to generate the UPF scenario can be specified through GUI and that the scenario can be automatically generated. The UPF scenario for MobiREAL simulator and sample animations of the simulation traces will be shown in our MobiREAL WWW pages [16].

Our major contribution is summarized as follows. (i) We present a new method to reproduce, in a simulation scenario, real pedestrian flows from the observed data. These data can be 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



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

edge	w.	density
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

edge	w.	density
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
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

edge	w.	density
6-11	12	0.060
11-18	12	0.063
18-22	12	0.026
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

Figure 6: Measured Density of Streets in Downtown Osaka City

Table 2: Result of Solving LP

path	flow rate
g11_4	0.111786
g24_11	0.039672
g24_16	0.054595
g24_25	0.357494
g24_33	0.006352
g29_11	0.007421
g29_18	0.038062
g29_33	0.008857

path	flow rate
g18_14	0.029088
g18_20	0.100903
g6_11	0.078003
g6_14	0.008147
g6_16	0.172850
g6_18	0.698762
g6_21	0.010313
g11_21	0.159075

path	flow rate
g27_11	0.096343
g27_16	0.005869
g27_33	0.140701
g4_14	0.072984
g9_11	0.046333
g9_18	0.034679
g9_23	0.052936
g25_11	0.349758

path	flow rate
g25_16	0.047970
g25_26	0.050226
g2_20	0.020553
g2_28	0.095887
g2_33	0.048179
g0_1	0.004860
g0_3	0.253643
g0_6	0.429208

path	flow rate
g1_2	0.036438
g23_20	0.169426
g23_24	0.028626
g3_9	0.210565
g0_33	0.030659
g3_16	0.243385
g3_23	0.317026
other gi_j	0

flow rate : person/sec.

direction. We gave detailed comparison with the related existing efforts in Section 1. (ii) We provide a full simulation toolset which can incorporate the UPF scenario. The toolset well supports developers who may want to conduct wireless network simulation using these UPF scenarios. (iii) We have characterized the UPF mobility compared with the RWP with obstacles mobility through experiments. Also we have endorsed that our method can reproduce pedestrian flows which are similar enough with real observation. We believe that the UPF scenarios are helpful to performance evaluation of ad-hoc communication in real environments.

More detailed evaluation of several mobile ad-hoc communication protocols is part of our future work.

6. REFERENCES

[1] C. Bettstetter. Mobility modeling in wireless networks: Categorization, smooth movement, and border effects. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(3):55–67, July 2001.

[2] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, editors. *Mobile Ad Hoc Networking – in Section 14; Simulation and Modeling of Wireless, Mobile, and Ad Hoc Networks*. IEEE Press, 2004.

[3] T. Camp, J. Boleng, and V. Davies. 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*, pages 483–502, 2002.

[4] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proc. of ACM/IEEE MobiCom*, pages 85–97, 1998.

[5] J. Yoon, M. Liu, and B. Noble. Sound mobility models. In *Proc. of ACM MobiCom*, 2003.

[6] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri. Towards realistic mobility models for mobile ad hoc networks. In *Proc. of ACM MobiCom*, pages 217–229, 2003.

[7] M. Hollick, T. Krop, J. Schmitt, H.-P. Huth, and R. Steinmetz. Modeling mobility and workload for wireless metropolitan area networks. *Computer Communications*, pages 751–761, 2004.

[8] W.-J. Hsu, K. Merchant, H.-W. Shu, C.-H. Hsu, and A. Helmy. Weighted waypoint mobility model and its impact on ad hoc networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, pages 59–63, 2005.

[9] A. K. Saha and D. B. Johnson. Modeling mobility for vehicular ad-hoc networks. In *Proc. of the 1st ACM Workshop on Vehicular Ad Hoc Networks (VANET 2004)*, pages 91–92, 2004. (poster paper).

[10] M. Musolesi, S. Hailes, and C. Mascolo. An ad hoc mobility model founded on social network theory. In *Proc. of ACM/IEEE MSWiM*, pages 20–24, 2004.

[11] X. Hong, M. Gerla, G. Pei, and C. Chiang. A group mobility model for ad hoc wireless networks. In *Proc. of ACM/IEEE MSWiM*, pages 53–60, 1999.

[12] D. Kotz and K. Essien. Analysis of a campus-wide wireless network. In *Proc. of ACM MobiCom*, pages 107–118, 2002.

[13] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless network. In *Proc. of ACM MobiCom*, pages 187–201, 2004.

[14] D. Tang and M. Baker. Analysis of a metropolitan-area wireless network. *Wireless Networks*, pages 107–120, 2002.

[15] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan. Characterizing user behavior and network performance in a public wireless lan. *ACM SIGMETRICS Performance Evaluation Review*, pages 195–205, 2002.

[16] MobiREAL web page. `http://www.mobireal.net`.

[17] K. Konishi, K. Maeda, K. Sato, A. Yamasaki, H. Yamaguchi, K. Yasumoto, and T. Higashino. MobiREAL simulator – evaluating MANET applications in real environments –. In *Proc. of 13th IEEE Int. Symp. on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2005. (to appear).

[18] Y. Yamamoto, T. Yashiro, H. Shigeno, and K. Okada. The acquisition and offer method of the real time congestion information on the road for pedestrians. Technical Report 2004-MBL-29(7), IPSJ SIG Technical Reports, 2004.

[19] G. F. Riley. The Georgia Tech network simulator. In *Proc. of the ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research*, pages 5 – 12, 2003.

[20] *lp_solve*. `http://groups.yahoo.com/group/lp_solve/`.

[21] F. Bai, N. Sadagopan, and A. Helmy. The IMPORTANT framework for analyzing the impact of mobility on performance of routing for ad hoc networks. *AdHoc Networks Journal*, pages 383–403, November 2003.

[22] L. Briesemeister and G. Hommel. Role-based multicast in highly mobile but sparsely connected ad hoc networks. In *Proc. of ACM MobiHoc*, pages 45–50, 2000.