ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Issue 4, April 2016

A CLUSTER BASED DESTINATION AND PATH IDENTIFICATION SCHEME

R.SANGEETHA[1], V.BALAMURUGAN[2]
(Anna Univ Affiliated) Master of Engineering, Department of Computer Science and Engineering[1],
(Anna Univ Affiliated) Associate Professor,Department of Computer Science and Engineering[2]
Mohamed Sathak Engineering College,
Kilakarai, Ramanathapuram dist., TamilNadu
[1]rgeetha145@gmail.com

[2]vbalram78@gmail.com

Abstract*— In mobile networks multimedia services are plays an important role since all the mobile users in movement. In mobile networks understanding of such network traffic behavior needs more planning and traffic management which introduces considerable overhead for network bandwidth resources. In this paper we propose a concept for path prediction for mobile nodes and also our proposed approach can able to find out the destination that the source node wants to communicate. In existing paper they first applied probability and Dempster–Shafer processes for predicting the likelihood of the next destination. After that they applied the second-order Markov chain process for predicting the given the path from the trip origin to that specific road junction and the direction to the destination. But it did not produce better result by means of time that means the processes were too long which increase energy consumption so we move our research to the next level.In our proposed paper we are going to modify the procedures which were used in the existing paper. The Destination Prediction model technique is applied as usual in existing paper that identifies the destination node's location and which corresponding cluster holds this node. After that we propose an optimal path selection protocol for transmitting packets from the source to destination. In this paper we use Ad hoc on demand Distance Vector (AODV) routing protocol for selecting optimal path. Our simulation result shows that our proposed approach reduces energy consumption and reduces processing time.

Keywords— **Cellular network, destination prediction, mobilitymodel, mobility pattern, mobility prediction, path prediction**

I. Introduction

A cellular network or mobile network is a communications network where the last link is wireless. The network is distributed over land areas called cells, each served by at least one fixed-location transceiver, known as a cell site or base station. In a cellular network, each cell uses a different set of frequencies from neighboring cells, to avoid interference and provide guaranteed bandwidth within each cell. When joined together these cells provide radio coverage over a wide geographic area. This enables a large number of portable transceivers (e.g., mobile phones, pagers, etc.) to communicate with each other and with fixed transceivers and telephones anywhere in the network, via base stations, even if some of the transceivers are moving through more than one cell during transmission. Cellular networks offer a number of desirable features:

- More capacity than a single large transmitter, since the same frequency can be used for multiple links as long as they are in different cells
- Mobile devices use less power than with a single transmitter or satellite since the cell towers are closer
- Larger coverage area than a single terrestrial transmitter, since additional cell towers can be added indefinitely and are not limited by the horizon

Major telecommunications providers have deployed voice and data cellular networks over most of the inhabited land area of the Earth. This allows mobile phones and mobile computing devices to be connected to the public switched telephone network and public Internet. Private cellular networks can be used for research.

Cell Signal Encoding

To distinguish signals from several different transmitters, frequency division multiple access (FDMA) and code division multiple access (CDMA) were developed. With FDMA, the transmitting and receiving frequencies used in each cell are different from the frequencies used in each neighboring cell. In a simple taxi system, the taxi driver manually tuned to a frequency of a chosen cell to obtain a strong signal and to avoid interference from signals from other cells.

The principle of CDMA is more complex, but achieves the same result; the distributed transceivers can select one cell and listen to it. Other available methods of multiplexing such as polarization division multiple access (PDMA) and time division multiple access (TDMA) cannot be used to separate signals from one cell to the next since the effects of both vary with position and this would make signal separation practically impossible. Time division multiple access, however, is used in combination with either FDMA or CDMA in a number of systems to give multiple channels within the coverage area of a single cell.

Structure of the mobile phone cellular network

A simple view of the cellular mobile-radio network consists of the following:

- A network of radio base stations forming the base station subsystem
- The core circuit switched network for handling voice calls and text
- A packet switched network for handling mobile data
- The public switched telephone network to connect subscribers to the wider telephony network

This network is the foundation of the GSM system network. There are many functions that are performed by this network in order to make sure customers get the desired service including mobility management, registration, call set up, and handover.

Any phone connects to the network via an RBS (Radio Base Station) at a corner of the corresponding cell which in turn connects to the Mobile switching center (MSC). The MSC provides a connection to the public switched telephone network (PSTN). The link from a phone to the RBS is called an uplink while the other way is termed downlink.

Radio channels effectively use the transmission medium through the use of the following multiplexing and access schemes: frequency division multiple access (FDMA), time division multiple access (TDMA), code division multiple access (CDMA), and space division multiple access (SDMA).

Mobility model

Mobility models represent the movement of mobile user, and how their location, velocity and acceleration change over time. Such models are frequently used for simulation purposes when new communication or navigation techniques are investigated. For mobility modeling, the behavior or activity of a user’s movement can be described using both analytical and simulation models. The input to analytical mobility models are simplifying assumptions regarding the movement behaviors of users. Such models can provide performance parameters for simple cases through mathematical calculations. In contrast, simulation models consider more detailed and realistic mobility scenarios. Such models can derive valuable solutions for more complex cases.

Typical mobility models include

- Brownian model
- Random waypoint model
- Random walk model
- Random direction model
- Random Gauss-Markov model
- Markovian model
- Incremental model,
- Mobility vector model

Quality of Service

Quality of service (QoS) is the overall performance of a telephony or computer network, particularly the performance seen by the users of the network. To quantitatively measure quality of service, several related aspects of the network service are often considered, such as error rates, bit rate, throughput, transmission delay, availability, jitter, etc. Quality of service is particularly important for the transport of traffic with special requirements. In particular, much technology has been developed to allow computer networks to become as useful as telephone networks for audio conversations, as well as supporting new applications with even stricter service demands.

Quality of service comprises requirements on all the aspects of a connection, such as service response time, loss, signal-to-noise ratio, crosstalk, echo, interrupts, frequency response, loudness levels, and so on. A subset of telephony QoS is grade of service (GoS) requirements, which comprises aspects of a connection relating to capacity and coverage of a network, for example guaranteed maximum blocking probability and outage probability. In the field of computer networking and other packet-switched telecommunication networks, the traffic engineering term refers to resource reservation control mechanisms rather than the achieved service quality. Quality of service is the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance to a data flow. For example, a required bit rate, delay, jitter, packet dropping probability and/or bit error rate may be guaranteed.

Quality of service guarantees are important if the network capacity is insufficient, especially for real-time streaming multimedia applications such as voice over IP, online games and IP-TV, since these often require fixed bit rate and are delay sensitive, and in networks where the capacity is a limited resource, for example in cellular data communication. A network or protocol that supports QoS may agree on a traffic contract with the application software and reserve capacity in the network nodes, for example during a session establishment phase. During the session it may monitor the achieved level of performance, for example the data rate and delay, and dynamically control scheduling priorities in the network nodes. It may release the reserved capacity during a tear down phase.

A best-effort network or service does not support quality of service. An alternative to complex QoS control mechanisms is to provide high quality communication over a best-effort network by over-provisioning the capacity so that it is sufficient for the expected peak traffic load. The resulting absence of network congestion eliminates the need for QoS mechanisms. QoS is sometimes used as a quality measure, with many alternative definitions, rather than referring to the ability to reserve resources.

Qualities of traffic

In packet-switched networks, quality of service is affected by various factors, which can be divided into “human” and “technical” factors. Human factors include: stability of service, availability of service, delays, user information. Technical factors include: reliability, scalability, effectiveness, maintainability, grade of service, etc. Many things can happen to packets as they travel from origin to destination, resulting in the following problems as seen from the point of view of the sender and receiver:

Low throughput

Due to varying load from disparate users sharing the same network resources, the bit rate (the maximum throughput) that can be provided to a certain data stream may be too low for real-time multimedia services if all data streams get the same scheduling priority.

Dropped packets

The routers might fail to deliver (drop) some packets if their data loads are corrupted, or the packets arrive when the router buffers are already full. The receiving application may ask for this information to be retransmitted, possibly causing severe delays in the overall transmission.

Errors

Sometimes packets are corrupted due to bit errors caused by noise and interference, especially in wireless communications and long copper wires. The receiver has to detect this and, just as if the packet was dropped, may ask for this information to be retransmitted.

Latency

It might take a long time for each packet to reach its destination, because it gets held up in long queues, or it takes a less direct route to avoid congestion. This is different from throughput, as the delay can build up over time, even if the throughput is almost normal. In some cases, excessive latency can render an application such as VoIP or online gaming unusable.

Jitter

Packets from the source will reach the destination with different delays. A packet's delay varies with its position in the queues of the routers along the path between source and destination and this position can vary unpredictably. This variation in delay is known as jitter and can seriously affect the quality of streaming audio and/or video.

Out-of-order delivery

When a collection of related packets is routed through a network, different packets may take different routes, each resulting in a different delay. The result is that the packets arrive in a different order than they were sent. This problem requires special additional protocols responsible for rearranging out-of-order packets to an isochronous state once they reach their destination. This is especially important for video and VoIP streams where quality is dramatically affected by both latency and lack of sequence.

destination cluster is the next destination cluster making use of filtered historical movement pattern; the filtering process is based on the day and the time of the day to increase accuracy; and 2) DPM builds on the work proposed in, wherein mo-bility prediction is based on evidential reasoning of Dempster– Shafer's theory making use of the user's contextual knowledge. DPM gives different weights to each method according to the number of days in the considered historical movement pattern.

Based on the computed destination cluster, we propose PPM that aims to estimate the path (i.e., subsequent transitions of road segments toward the destination) a user would take during his movement from the current location toward the destination within

time period d_t. PPM takes into account 1) the user's habits in terms of the frequency of using road segments to reach a specific destination (e.g., estimated destination cluster), 2) the direction from the current location to that specific destination, 3) the current trajectory/path (i.e., subsequence transitions of road segments from the movement origin to the current lo-cation), and 4) spatial conceptual maps. More specifically, at each road junction/intersection, PPM determines the next road segment a user will likely use during his movement toward the destination; indeed, PPM selects the potential next road segments among the adjacent road segments of the considered/ current road junction according to the current direction (i.e., direction from the last crossed road junction to the current road junction) deviation compared with the estimated destination cluster. Then, making use of filtered historical movement trace, PPM computes, based on an extended second-order Markov chain, the probabilities of all selected potential next road seg-ments given current trajectory/path and destination; the road segment among the potential next road segments with the highest probability is then selected as the next road segment. Here, the historical movement trace filtering process is based on the day of the week (e.g., weekend, holiday, and Labor Day) and the time of the day (e.g., morning, noon, afternoon, and night). PPM repeats the same process until the selection of the last road segment to the destination cluster. The predicted user's path consists of the list of the selected road segments. To the best of our knowledge, this is the first work that takes into account both the user's habits and the user's contextual knowledge to estimate the user's path to the destination. In addition, this work is the first to consider destination clustering to reduce errors using historical and contextual knowledge. More importantly, this work presents one of the few schemes that allow for predicting, without a restrictive assumption (e.g., known specific user pattern), the whole path from the origin to the destination. In this paper, we do not take into account energy consumption of user equipment (UE); indeed, we do believe that energy consumption is not an important constraint for vehicles, and the impact on their batteries is expected to be negligible. For users using smart phones on board vehicles, they can always consider charging them while being on the move. In case charging is not possible on board vehicles, users shall be given the flexibility to manually disable DAMP. DAMP can be also automatically disabled if a UE device has battery below a predefined threshold, e.g., 20% of the battery. It is worth noting that users who are not in motion do not run DAMP; thus, they do not use UE energy for the mobility prediction process. Making use of this contextual information, they generate bodies of evidence applying the concepts of the Dempster–Shafer theory; Making use of this contextual information, they generate bodies of evidence applying the concepts of the Dempster–Shafer theory; automatically disabled when a UE device is moving at a speed lower than a predetermined threshold, e.g., an equivalent of a general pedestrian speed.

II. RELATED WORK

III.

Mobility is an important problem in mobile networks (MN). So, the mobility management in mobile networks is to be considered and the services provided by us to meet the communication reliability. So the proposed algorithm should meet all the requirements what we discussed above. Then the path prediction in mobile networks also an important characteristic since all nodes are mobile in nature. For that in this paper we proposed accurate mobility prediction scheme, which is called Destination and Mobility path Prediction (DAMP). It detects final or intermediate destination and then find out the path between the source and destination nodes. The path selection between mobile users may be based on 1) the trip origin and current location, 2) current and future directions of the mobile users, 3) the current and history of the trajectories followed by the users, and 4) the information on the users' contextual knowledge.

This proposed DAMP consists of two important models, namely, Destination Prediction Model (DPM) and Path Prediction Model (PPM). The main aim of DPM is to estimate the user's final destination or intermediate destination within a time period. It takes the following consideration for selecting the destinations which are: 1) the user's habits in terms of frequently visited locations, 2) the direction from the movement origin to the current location, and 3) the user's contextual knowledge. During path selection PPM takes the following considerations: 1) the user's habits in terms of the frequency i.e. estimated destination cluster, 2) the direction from the current location to that specific destination, 3) the current path and 4) spatial conceptual maps.

DISADVANTAGES

- It consumes more energy
- The PPM takes more time to process

Mobility modeling has been extensively studied in many types of wireless networks during the past ten years . Mobility model analysis can be used to create models for predicting user mobility. User mobility prediction allows estimating/predicting the location and trajectory of the user in the future. The commonly used mobility models are random walk, random waypoint, fluid flow, Markovian, and activity-based mobility models. The simplest of these models are the random walk and random waypoint models; they were originally proposed to emulate the unpredictable mobility of particles in physics. The other models are used for prediction, such as path prediction. Indeed, based on the regularity of user mobility, a conditional probability distribution of next moves is defined considering movement direction and time; the move with the highest value is predicted as the next move. In other words, the cell that was most frequently visited according to the current location, the current movement direction, and the time of the day is predicted as the next cell.

Recently, a considerable amount of work has been done on developing users' mobility prediction models. Many of these models heavily rely on the availability of prior information on the users' mobility history. Whereas the continuous tracking of mobile users may lead to better predictions in terms of movement, such mod-els suffer from the large overhead accrued due to constant monitoring; obviously, this requires a more detailed analysis of the users' mobility history and the application of advanced data mining and knowledge discovery techniques; unfortunately, these models are limited to predicting only where a user is likely to move (i.e., user's final destination) instead of the path to reach this final destination. For example, a scheme that incorporates geographic maps with identifiable landmark objects (e.g., schools, malls, gyms, and libraries) into the users' mobility prediction models has been proposed in; more specifically, the mobility prediction architecture gathers the necessary information for the prediction process and analyzes this information using the Dempster–Shafer theory to predict future locations of the mobile user. The process of the location prediction is carried out in three main phases: infor-mation gathering, evidence extraction, and decision making. Information gathering is concerned with capturing the neces-sary contextual information that includes environment context (places of interest for users and road segments) and user context (user's interests, user's tasks and goals, and user's schedule). Making use of this contextual information, they generate bodies of evidence applying the concepts of the Dempster–Shafer theory for example, the evidence suggesting that a student with a high interest in exercising would be going to the gym is eliminated if his schedule does not allow enough spare time. Then, they compute the belief mass of each body of evidence (i.e., the probability that the body of evidence occurs). Finally, to determine the user's future predicted location, they combine each pair of hypothesis-belief mass using the Dempster rule of combination; a hypothesis represents a location or a sequence of locations. Indeed, they compute the belief function Bel(Hi) of hypothesis Hi. Bel(Hi) quantitatively describes all the rea-sons to believe in hypothesis Hi. The location with the highest belief value is the predicted future location of the user. How-ever, this technique (i.e., destination prediction model) is used in certain path prediction models [7], [8] to improve prediction accuracy by eliminating or affirming certain paths according to the predicted destination. However, such models require a vast amount of information (e.g., user's preferences, user's goals, and user's schedules) to be collected and processed and may not perform very well with temporary changes in the surrounding infrastructure.

A few models consider using both mobility historical data and current conditions in the network. One example is the model proposed in, which considers both the current trajectory of mobile users (i.e., ordered set of cells already transited) and time-of-day, as well as historical data, to predict the likelihood of single-cell transition and N -cell transition for an arbitrary user in wireless networks; the prediction model performed quite well for lower values of N (e.g., two cells). In , a short-term prediction model that employs mobility history for predicting the future location of a mobile user while considering the mobile user's current trajectory within the predefined navigation zone was proposed; this model is limited to only next-cell prediction. In , a long-term mobility pre-diction model that considers both the current trajectory and the movement direction, as well as historical data, was proposed; however, the model requires an immense amount of mobility history and massive processing load.

In the following, we briefly overview our previous contributions that are most related to the proposed model. In, we proposed a method to estimate a user's future destination based on the use of filtered user's mobility history and con-textual knowledge; the filter is based on the type-of-day (e.g., working, holiday, and weekend) and the time-of-day (morning, noon, afternoon, evening, and busy hours); the proposed model also takes into account the movement direction. The drawbacks of this model are as follows: 1) The automatic identification of frequently visited locations (FVLs) has not been taken into account; 2) the databases are not updated according to the user's predictability level, which is the degree to which a correct prediction of a user's mobility can be made; this degree is related to the frequency of visited places and transited roads according to the day-of-week and the time-of-day; 3) the frequency function of FVL is not explicitly defined; and 4) the weighted sum of the belief and probability functions is not related to the user's predictability level. In [8], we proposed an approach that predicts the path the user will use within a time period during his movement from the trip origin to the destination; the approach makes use of filtered users' mobility history, current movement data (e.g., trip origin and current location), and spatial conceptual maps while assuming *a priori* knowledge of the destination. More specifically, at each road junction (starting from the location where the user first accesses the network), the next road junction the user will likely use during his movement toward the trip destination, is determined. The drawbacks of this approach [8] are as follows: 1) The deviation function takes into account the trip origin instead of the previous road junction, and 2) common conditional probability is used instead of second-order Markov chain that is more appropriate in this type of situations.

In this paper, we propose a model that proposes solutions to overcome these limitations. The proposed DAMP scheme considers several criteria, namely, trip origin to current location, current and future directions, user contextual (UC) knowledge, day-of-week, and time-of-day, in predicting paths; it is a long-term users' mobility prediction model. To limit the impact of using user mobility history (that may change), DAMP considers user knowledge and regular spatial and temporal patterns for predicting the mobility of users.

III. DAMP: Destination and Mobility Path Prediction Model

Here, we present the details of the proposed scheme, called DAMP. More specifically, we present 1) the process used by DAMP to collect data of interest for the prediction procedure and the structure of the database used to store these data; 2) the semi-Markov process used by DAMP to derive DPM and PPM; 3) the DPM that estimates the mobile user destination; and 4) the PPM that predicts, given the destination computed

TABLE I
UC Information Structure

Personal Context (PC)	Frequently Visited Location Context (FVLC)	Task Context (TC)	Interest Context (IC)	Calendar Context (CC)	Day Type (DT)
• User ID • Name • Age • Predictability	• Location (NM node) • Location name • Preferable day • Earliest preferable time • Latest preferable time • Duration • Characteristics • Importance • Frequency	• Task name • Earliest time • Deadline • Duration • Characteristics • Importance • Frequency	• Interest name • Preferable day • Earliest preferable time • Latest preferable time • Duration • Characteristics • Importance • Frequency	• Location (NM node) • Date • Time • Characteristics	• From (date) • To (date) • Workday • No workday

by DPM, the path the user would take during his movement from the current location to the destination.

A. User Mobility Patterns

1) Assumptions: In this paper, we assume that the road topology consists of several roads and junctions while the entire network space is assumed to be divided into cells. We refer to the location frequently visited by a user (e.g., home, school, shop, mall, and office) as an FVL. We assume that a road junc-tion or an FVL is represented by a node; each node is identified by a node ID that is related to its geographic coordinates (i.e., latitude and longitude); we refer to data about visited nodes (e.g., time, date, and node ID) as mobility data. We refer to the road between two nodes a and b as a road segment and identify it using a road segment ID that is represented by the node pair (a, b), where $a \rightarrow b$ _= $b \rightarrow a$. A user's location is identified by his geographic coordinates. The movement of a mobile user through the network can be described by a list that represents the sequence of road segments that was visited by the user throughout the trip. A user's mobility pattern from the network's perspective is determined by the user's terminal (e.g., mobile phone) mobility pattern. The users' mobility history patterns can be periodically recorded using node ID (road junction and FVL).

The mobility history can either be recorded for each user or collectively for all users into a single history profile per location. The latter method is more suitable for situations where all users generally exhibit similar behavior at a given navigation zone and are also not significantly impacted by erratic behaviors from one or more users. Although different groups of users have different mobility patterns, it can be difficult to address every type of group behavior in a single mobility model. To derive DPM, we need contextual knowledge about users; we assume that UC information is organized into six categories, as shown in Table I. The UC database can be built 1) by having users fill out a questionnaire and explicitly express their interests with regard to different places within their living areas or 2) by hav-ing users "continuously" registering both their tasks and sched-uled appointments. To implement mobility data collection, we assume that 1) the UE maintains a database that records data about the user movements and his living area; 2) static data about geographic maps (topology/map of roads), called a nav-igation map (NM), are readily available; and 3) the UE em-beds technology, such as a tachometer and Global Positioning System (GPS), that samples user velocity and coordinates of places visited by the user, along with the day and the time of the visits. It is also assumed that an NM database contains ge-ographic coordinates of nodes (e.g., road junctions and FVLs). An FVL is extracted from the UC database shown in Table I or automatically inserted. Indeed, when a user's velocity is 0 and the current location is not a road segment or road junction, we assume that the current location may be a new visited place and insert it in an FVL database. A user movement trace (UMT) database contains user ID, date d, time t, and node ID (an FVL or a road junction) that represents user location at date d and time t. A user frequently visited location trace (UFVLT) database contains user ID, date d, arrival time t_a, departure time t_d, and node ID (an FVL) that represents user location at date d from arrival time t_a to departure time t_d; in other words, arrival time t_a denotes the time when the user reaches the location, whereas departure time denotes the time when she leaves it.

To limit the size of UMT and UFVLT databases, each entry/ record in UMT and UFVLT databases is deleted after a certain number of days, called record lifetime (RL), that is closely re-lated to the user's predictability level. Effectively, each user has a predictability level stored in UC; RL decreases when the pre-dictability level increases; a "high predictability" level means the user is more predictable (i.e., it is easier to predict user's movements). For example, for a "high (resp., intermediate/ low) predictability" level, the user's RL can be set to two (resp., three/four) weeks. Static databases Node, Road junction, FVL, User, UC, and NM are updated every six months and also whenever an update becomes required.

The following section presents in detail the data collection process.

2) Data Collection Process: Algorithm 1 presents the pseudocode, executed by UE, for recording data. At every time unit t (e.g., 1 s), the current velocity of the user and geographic coordinates (*latitude and longitude*) of his current location are measured. When the user's current location is a location (a road segment or a road junction), which is stored in the NM database (Line 3), it is inserted together with the current timestamp (date and time) into the UMT database (Line 4). When the user's velocity decreases and falls to 0, the user is deemed not moving (Line 5). Thus, his current location is inserted together with the current timestamp (current date and current time) into the UFVLT database (Lines 6) when the current location is not a road junction or a road segment (Line 5); it is worth noting that a road junction or a road segment visited by a user cannot be designated as an FVL. The attribute "current time" of current timestamp is assigned to the field "arrival time" in the UFVLT database (Line 6). Indeed, arrival time corresponds to the time when velocity falls to 0, whereas departure time corresponds to the time when velocity starts increasing; in this case, current velocity is different from 0 (Line 8).

Algorithm 1: Pseudocode for movement data gathering.

```
Input: User_ID, NM, FVL
Variable: X (Boolean with initial value = true)
Output: UMT, UFVLT, and FVL
1.  each $t$ sec {
2.  Measure current_velocity and current_location
3.  If (current_location is road junction or road segment)
4.  Put in $UMT$ the 4-tuple (user_ID, current_date, current_time, current_location_ID)
5.  If [(current_velocity = 0) and (current_location is not road junction or road segment) and (X = true)] {
6.  Put in $UFVLT$ the 5-tuple (user_ID, current_date, current_time, 0, current_location_ID)
7.  X = false
8.  } elseif [(current_velocity _= 0) and (current_location is not road junction or road segment) and (X = false)] {
9.  Update the last record of $UFVLT$ set $departure\_time$ = current_time
10. X = true
11. If (current_location does not exist in $FVL$)
12. Put in $FVL$ the 1-tuple (current_location_ID)
13. }
14. }
```

Therefore, when the current velocity is different from 0 and the current location is not a road junction or a road segment (Line 8), the departure time of the last inserted location in the UFVLT database is updated, making use of the current time (Line 9). When the last inserted location in the UFVLT database (i.e., current location) does not exist in the FVL database (Line 11), it is inserted into the FVL database (Line 12). The time complexity of Algorithm 1 is $O(n)$, where n is the number of time units t during data gathering.

B. Semi-Markov Process

The mobility behavior can be modeled as a semi-Markov process and can be applied for predicting the transition that an arbitrary user makes from its current location within time period d_t. The model assumes knowledge of the transition probabili-ties; these probabilities are computed using the mobility history that is collected by each user. To avoid an opportunistic location as an FVL, we derive the frequency function $f()$ that is defined as follows:

$$f(l) = \frac{n_l}{n_d} \tag{1}$$

where n_l and n_d denote the number of times location l is recorded in the UFVLT database and the number of recorded days in the UFVLT database, respectively. Thus, location l is considered FVL if and only if $f(l)$ exceeds a predefined threshold f_{th}.

Each mobile user records his mobility history (i.e., UMT and UFVLT databases); this allows for the computation of road segment (resp., FVL) transition probabilities $P_{i\longrightarrow j}$ [i.e., transition from road segment (resp., FVL) i to road segment (resp., FVL) j]. The prediction accuracy of road segment (resp., FVL) transitions can be improved by additionally considering prior road segment transitions of the user before the transition into the current road segment. In this case, road segment (resp., FVL) transition probability can be modified to $P_{h,i\longrightarrow j}$, which is a second-order Markov chain, where h is the subsequence transitions of road segments from the trip origin to the road segment i (resp., from FVL i to the current road segment). The accuracy of the prediction can also be improved by additionally considering the type of the day and the time of the day. For example, a user who works from Tuesday to Thursday at a factory, from Friday to Saturday at a school and does not work on Monday will have three types of days: factory day, school day, and rest day.

Second-order Markov chain is derived from a semi-Markov process where the successive state occupancies are governed by the transition probabilities $P_{i\longrightarrow j}$; the semi-Markov process depends on both the current state and the next state transition. The semi-Markov process for a time-homogeneous process is given by $Q_{i,j}(d_t)$, i.e.,

$$P_{i\longrightarrow j} = Q_{i,j}(d_t) = \Pr\{X_{n+1} = j,\ T_{n=1} - T_n \le d_t | X_n = i\} \tag{2}$$

where X_n and X_{n+1} represent the state of the system after the nth and $(n + 1)$th transitions, respectively, with T_n and T_{n+1} being the times at which the nth and $(n + 1)$th transitions occur, respectively. $Q_{i,j}(d_t)$ denotes the probability that, immediately after making the transition into state i, the process makes a transition into state j within t units of time. Thus, the probability $Q_{i,j}$ () [see (2)] can be computed to evaluate the predictions of an arbitrary user making a transition to a next location (e.g., FVL). However, the semi-Markov process for mobility prediction can also be extended to the case where times-of-day $T\ ioD$ (e.g., morning, noon, afternoon, and night), types-of-day $T\ yoD$ (e.g., weekend, labor day, holiday, and vacation day), and the user's previous locations h are considered in the mobility pattern, i.e., extending $Q_{i,j}(d_t)$ to $Q^{T ioD,T yoD}_{h,i,j}(d_t,d)$, which is defined as follows:

$$\begin{aligned} P^{T ioD,T yoD}_{h,i\to j} &= Q^{T ioD,T yoD}_{h,i,j}(d_t,d) \\ &= \Pr\{X_{n+1}=j,\ T_{n+1}-T_n \le d_t | X_n = i \\ &\quad X_{n-1}=h,\ T_{n+1}\in T ioD,\ T_n \in T ioD \\ &\quad d \in T yoD\} \\ &= \frac{n_{h,i,j}(T ioD,\ T yoD)}{\sum_{k=1}^{N} n_{h,i,k}(T ioD,\ T yoD)} \end{aligned} \qquad (3)$$

where d is the current date that is used to define $T yoD$ at which the nth and (n + 1)th transitions occur, $n_{h,i,j}$ ($T ioD$, $T yoD$) is the number of times the transition from FVL i to FVL j has occurred within times-of-day $T ioD$ and types-of-day $T yoD$ after crossing h, and N is the cardinality of the set of possible states j.

For the road segment transition prediction, accuracy can be improved by additionally considering the next destination (i.e., FVL to be reached after the current trip) of the user; in this case, the probability $Q_{i,j}$ () defined in (2) can be modified to $P_{h,i,j,D}$, where D is the estimated destination (an FVL) of the trip, and h is the set of transited road segments before entering road segment i. Thus, the semi-Markov process can also be extended to the case where the user's estimated destination is considered in the mobility prediction, i.e., extending $Q_{i,j}(d_t)$ to $Q^{T ioD,T yoD}_{h,i,j,D}(d_t,d)$, which is defined as follows:

$$\begin{aligned} P^{T ioD,T yoD}_{h,i\to j,D} &= Q^{T ioD,T yoD}_{h,i,j,D}(d_t,d) \\ &= \Pr\{X_{n+1}=j,\ T_{n+1}-T_n \le d_t | X_n = i \\ &\quad X_{n-1}=h,\ X_{last}=D\ T_{n+1}\in T ioD \\ &\quad T_n \in T ioD,\ d \in T yoD\} \\ &= \frac{n_{h,i,j,D}(T ioD,\ T yoD)}{\sum_{k=1}^{N} n_{h,i,k,D}(T ioD,\ T yoD)} \end{aligned} \qquad (4)$$

where X_{last} represents the destination of the current trip, $n_{h,i,j,D}$($T ioD$, $T yoD$) is the number of times the transition from road segment i to road segment j toward FVL D has occurred within times-of-day $T ioD$ and types-of-day $T yoD$ after crossing h, and N is the cardinality of the set of adjacent road segments of road segment i. In the following section, we describe how our proposed model makes use of NM, UFVLT, and UC databases to predict the user's destination.

C. DPM

The proposed DPM, as part of DAMP, makes use of UC, NM, and UFVLT databases to predict the user's destination (i.e., the next FVL to be visited). Fig. 1 shows the interactions between UC, NM, and UFVLT databases and DPM functions (e.g., f (), o(), $clustering$, $P_{h,i,j}$ (), Bel(), b(), and ws()).

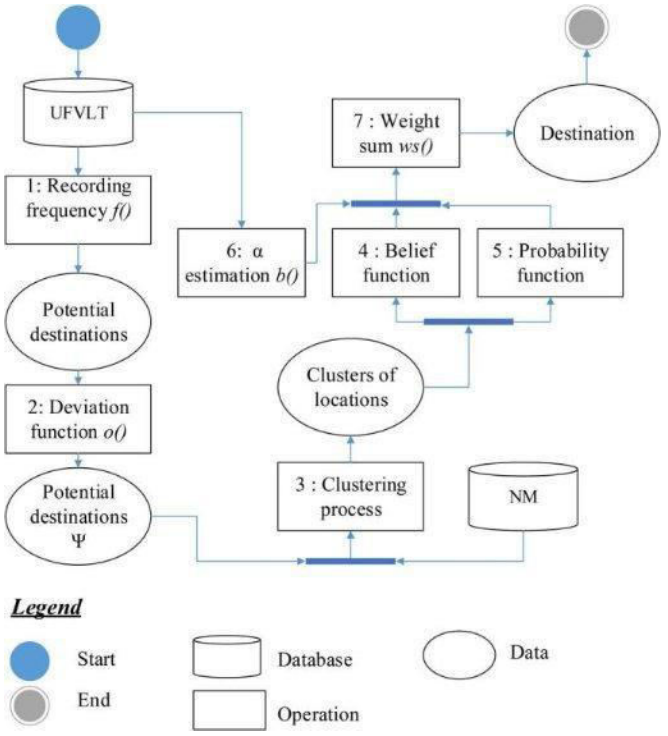


Fig. 1. DPM processes.

Indeed, the locations stored in UC and UFVLT databases represent the user's potential destinations (i.e., FVLs). DPM does not predict a single destination but a cluster of destinations that likely includes the user's destination; we define a cluster of destinations as a set of FVLs that can be visited/reached by a user using the same portion of path (set of road segments) within time period d_t. Before the clustering process, we make use of the deviation function o to select potential destinations Ψ; the deviation function o measures the deviation rate of an angle (always smaller than 180) and is defined as follows:

$$o : [0,180] \to [0,1]$$

$$o(\theta) = 1 - \frac{\theta}{180}.$$

Indeed, we measure the deviation rate $o(\theta^C_l)$ of each FVL l, which is selected in the UFVLT database, where $f(l) \geq f_{th}$ using (1). θ^C_l denotes the angle formed by the current movement direction (i.e., vector from the trip origin to the current location) and the movement direction toward FVL l (i.e., vector from the current location to FVL l). For better understanding, let us consider the example shown in Fig. 2; the angle formed by the current movement direction (vector SC) and the movement direction toward FVL $L4$ is the angle θ^C_{L4}. Notice that the maximum value of θ^C_l is 180° (case of U-turn).

We define Ψ as follows:

$$\Psi = \{l / o(\theta^C_l) \geq \varepsilon\}_l \qquad (5)$$

where $\varepsilon = o(\theta_{th1})$, and θ_{th1} denotes a predefined threshold. In our proposed model, θ_{th1} is set to 90° to consider FVLs located in front of the current location according to the current movement direction.

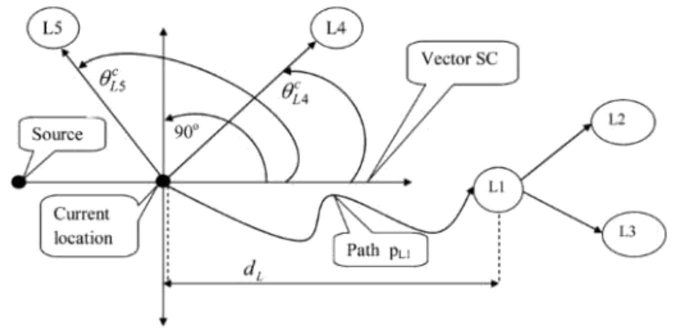


Fig. 2. Destination clustering example.

Using the example shown in Fig. 2, we compute $\Psi = \{L1, L2, L3, L4\}$ using (5); in this case, $\varepsilon = 1/2$, assuming $\theta_{th1} = 90$. L5 is not an element of Ψ because $o(\theta^c_{L5}) < \varepsilon = 1/2$.

After computing Ψ, we execute the clustering process. In-deed, all the elements of Ψ that may be reached using the same portion of the path within a predefined time period dt of travel form a single cluster; based on the NM database, we derive a directed graph G whose edges correspond to road segments and whose vertices correspond to nodes; the road segment length represents the weight of the corresponding edge. Making use of graph G, we determine the shortest path to reach each element of Ψ using Dijkstra's algorithm; then, making use of the length of each computed path and the maximum permitted velocity of road segments, we determine the portion of that computed path after d_t time of travel; finally, the elements of Ψ, which have the same portion of the path, form a single cluster. For better understanding, let us consider the example shown in Fig. 2. In this example, L1, L2, and L3 may be reached using the same portion of path P_{L1} within travel time dL. Hence, L1, L2, and L3 form a single cluster. Intuitively, the prediction becomes more accurate when the size of clusters decreases.

DPM predicts the user's destination (or rather the FVL cluster) using a combination of belief and probability functions. DPM uses the belief function, i.e., $Bel()$, adopted in [28]. $Bel()$ is based on the utilization of the mathematical theory of evidence as a tool of reasoning to investigate the user's behavior concerning his decisions about his future location (i.e., FVL). The theory is based on two ideas: the idea of obtaining degrees of belief for a related hypothesis and the idea of applying Dempster's rule for combining such degrees when they are based on different bodies of evidence; more details about this theory can be found in [28]. The net effect of Dempster's rule is that concordant bodies of evidence reinforce each other, whereas conflicting bodies of evidence erode each other. The main advantage of the underlying theory of evidence over other approaches [17], [35], [36] is its ability to model the narrowing of a hypothesis with the accumulation of evidence and to explic-itly represent uncertainty in the form of ignorance or reservation of judgment. $Bel()$ makes use of contextual information, stored in the UC database, to compute the belief level, i.e., $Bel(C_i)$, of each formed cluster C_i to be the destination cluster. DPM also computes the probability $P^{T ioD, T yoD}_{h,cl\to Ci}$ that the formed cluster C_i is the destination using (3), where cl is the current location;it is worth noting that the set of type-of-day (i.e., $T\ yoD$) is different for each user. DPM uses a weighted sum of the belief and probability functions to compute the destination; the sum is defined as follows:

$$ws(c_i) = \alpha Bel(C_i) + (1 - \alpha)P^{T ioD,T yoD}_{h,cl\to Ci} \qquad (6)$$

where α is computed as follows:

$$b : [0\ +\infty] \to [0\ 1]$$

$$\alpha = b(n) = \begin{cases} 1 - \frac{n}{RL} & \text{if } 0 \le n \le RL \\ 0 & \text{if } n \ge RL \end{cases} \qquad (7)$$

where n denotes the number of days used to learn the user's habits, and RL is the user's record lifetime. Equation (6) shows that as the number of learning days (of the users' habits) increases, the influence of the belief function $Bel()$ decreases while the influence of the probability function $P^{TioD,T yoD}_{h,cl,Ci}$ in-creases. DPM selects the cluster with the largest value of $ws()$ as the destination cluster. DPM, which is a semi-Markov process, calculates m state transition probabilities; each state refers to a discrete FVL, thus adding $O(m)$ space and time complexity.

D. PPM

PPM assumes *a priori* knowledge of the destination due to DPM. The operation of PPM consists in choosing a road seg-ment (among one or more road segments) at each road junction toward the destination. The selection process starts from the current location (i.e., the road junction, immediately after the road segment where the prediction starts) and is repeated within a predefined time period dt of travel or until the destination (an FVL) is reached; at each occurrence of the selection process, the previous road segment that has been selected becomes the current road segment, and the road junction immediately after that current road segment becomes the current road junction. Indeed, the process terminates when a list of road segments that constitutes a path from the current location to the destination cluster (i.e., cluster of FVLs that may be reached using the same portion of the path within a predefined time period dt of travel) is computed. Fig. 3 shows the PPM operation. At each road junction (e.g., the current road junction), PPM starts by a preselection process choosing a set of road segments Ω among the adjacent road segments to the current road junction; Ω represents the set of potential next road segments to be visited; the preselection process aims to reduce the size of the set of adjacent road segments used for the selection process; it is performed by making use of a deviation function r, which mea-sures the deviation rate of an angle (always smaller than 180); $r()$ is defined as follows:

$$r : [0\ 180] \to [0\ 1]$$

$$r(\theta) = \begin{cases} 1 - \frac{\theta}{\Theta} & \text{if } 0 \le \theta \le \Theta \le 180 \\ 0 & \text{if } \Theta \le \theta \le 180 \end{cases}$$

where Θ is the angle formed by $\vec{A}$ and $\vec{B}$, $\vec{A}$ is the correspond-ing vector of the road segment in the opposite direction to the previous road segment, and $\vec{B}$ is the vector from the current junction to the destination cluster.

Then, making use of the deviation function $r()$, we measure the deviation rate $r(\theta_j)$ of each adjacent road segment $j$to

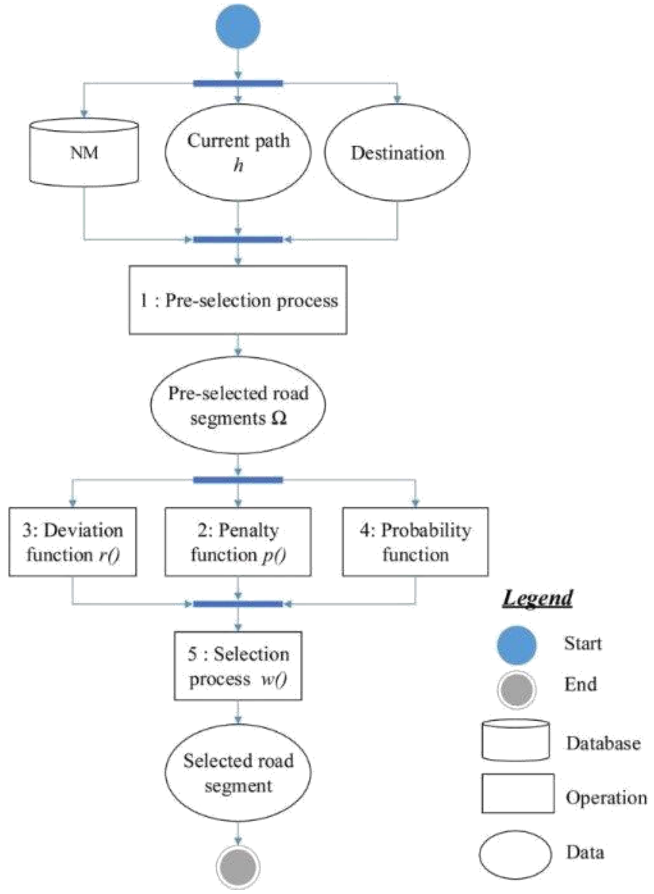


Fig. 3. PPM processes.

the current road junction; θ_j denotes the angle formed by the corresponding vector of adjacent road segment j and $\overrightarrow{B}$. The preselected adjacent road segments j are those that belong to the following set:

$$\Omega = \{j \mid r(\theta_j) \geq \phi\}_j \tag{8}$$

where $\phi = r(\theta_{th2})$, and θ_{th2} denotes a predefined threshold. For the sake of better understanding, let us consider the example shown in Fig. 4. Let $i \to C$ be the previous selected road segment and vector $\overrightarrow{iC}$ be the corresponding vector, C be the current junction, D be the destination cluster, and $C \to j$ be an element of the set of adjacent segments to the current junction $\theta^{\overrightarrow{CD}}_{\overrightarrow{Cj}}$ be the angle formed by vector $\overrightarrow{CD}$ and vector $\overrightarrow{Cj}$; notice that $\overrightarrow{Ci}$ is the vector representing the road segment $C \to i$, which is in the opposite direction to the previous road segment $i \to C$ that has been selected by PPM; selecting the road segment $C \to i$ as the next road segment represents a U-turn.

Using the example shown in Fig. 4, we compute $\Omega = \{1, 2\}$ using (8); in this case, $\phi = r(45)$, assuming $\theta_{th2} = 45$. The selection of a road segment from Ω as a next road segment is performed using the product of the following: 1) the transition probability $P^{T_{ioD},T_{yoD}}_{h,C\to j,D}$ [see (4)]; 2) the deviation function $r()$; and 3) the penalty function $p()$, which returns one when the considered road segment may be used to reach the destination

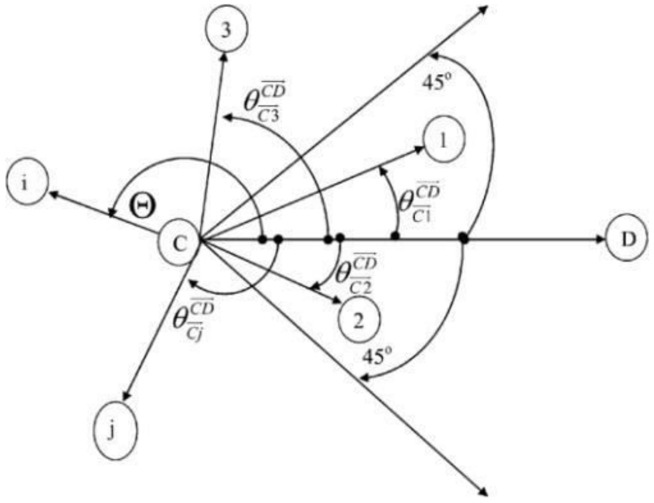

3
45°
$\theta^{\overline{CD}}_{\overline{C3}}$
1
i
Θ
C
$\theta^{\overline{CD}}_{\overline{C1}}$
D
$\theta^{\overline{CD}}_{\overline{C2}}$
$\theta^{\overline{CD}}_{\overline{Cj}}$
2
45°
j

Fig. 4. Example of the preselection process.

cluster; otherwise, it returns null; the penalty function is defined as follows:

$$p(C \to j) = \begin{cases} 1, & \text{if possible to reach } D \text{ through } C \to j \\ 0, & \text{if nonpossible to reach } D \text{ through } C \to j. \end{cases} \tag{9}$$

Indeed, we compute the product function $w(C \to j)$ of each road segment $C \to j$, in Ω, using (10) and choose the road segment with the largest value of $w(C \to j)$ as the next road segment. The product function of $C \to j$, in Ω, is defined as follows:

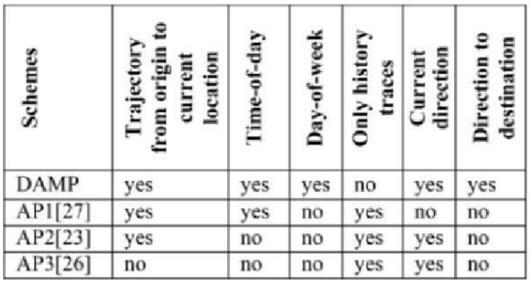
$$w(C \to j) = P^{T_{ioD},T_{yoD}}_{ct,C\to j,D} \times r\left(\theta^{\overrightarrow{CD}}_{\overrightarrow{Cj}}\right) \times p(C \to j). \tag{10}$$

We make use of the deviation function $r()$ when computing the product function $w()$ to give priority to road segments whose directions are more oriented toward the destination cluster D (i.e., the rationale behind using angle $\theta^{\overrightarrow{CD}}_{\overrightarrow{Ci}}$ to define the deviation function). The penalty function $p()$ is used to assign 0 to the product function $w()$ when the considered road segment is not an option (e.g., dead-end road) to reach destination cluster D. In case of lack of historical data, it will not be possible to calculate the transition probability $P^{T_{ioD},T_{yoD}}_{h,C\to j,D}$ [see (4)]; (10) cannot be applied to compute the product function $w()$. Thus, the selection of a road segment from Ω as a next road segment is performed using the product of the deviation function $r()$ and the penalty function $p()$. In this case, the product function of $C \to j$, in Ω, is defined as follows:

$$w(C \to j) = r\left(\theta^{\overrightarrow{CD}}_{\overrightarrow{Cj}}\right) \times p(C \to j). \tag{11}$$

The selected road segment is added to the list of previous selected road segments; this list constitutes the predicted path from the current location to the destination cluster. PPM, which is a semi-Markov process, calculates g state transition probabil-ities. Each state refers to a discrete road segment, thus adding $O(g)$ space and time complexity at each road junction and $O(g^2)$ for the operations of PPM.

TABLE II
PREDICTION SCHEMES FOR COMPARISON



Schemes	Trajectory from origin to current location	Time-of-day	Day-of-week	Only history traces	Current direction	Direction to destination
DAMP	yes	yes	yes	no	yes	yes
AP1[27]	yes	yes	no	yes	no	no
AP2[23]	yes	no	no	yes	yes	no
AP3[26]	no	no	no	yes	yes	no

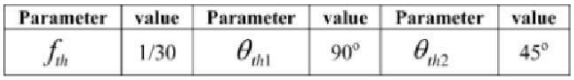
IV. PERFORMANCE EVALUATION

Here, we evaluate, via simulations, the performance of DAMP. Similar to [23], [26], [28], and [33], we define one evaluation parameter that is the prediction accuracy (i.e., path similarity). As comparison terms, we use the schemes described in [23], [26], and [27], which are referred to as AP1, AP2, and AP3, respectively. AP1, AP2, and AP3 were selected because, to the best of our knowledge, they represent the most recent work related to mobility prediction in MNs that outperform existing approaches (e.g., [12], [24], [29], and [33]). Table II shows the characteristics of DAMP, AP1, AP2, and AP3.

A. Simulation Setup

To evaluate DAMP, we use real mobile user traces (GPS trajectories), acquired from the Microsoft Research Asia laboratory's database available in the context of the GeoLife project [38]. This GPS trajectory data set was collected in a period of over three years (from April 2007 to August 2012). A GPS trajectory of this data set is represented by a sequence of timestamped points, each of which contains the informa-tion of latitude, longitude, altitude, date, and time. This data set contains 17 621 trajectories with a total distance of about 1.2 million kilometers and a total duration of 48 000 hours. These trajectories were recorded by different GPS loggers and GPS phones and have a variety of sampling rates; 91% of the trajectories are logged every 1–5 s or every 5–10 m per point. This data set recorded a broad range of users' outdoor move-ments, including not only life routines such as going home or to work but also some entertainment and sports activities, such as shopping, sightseeing, dining, hiking, and cycling. According to GPS trajectories, we identify three groups: 1) subjects whose mobility is unpredictable; 2) subjects whose mobility is mod-erately predictable; and 3) subjects whose mobility is highly predictable. Converting GPS coordinates to Cartesian coordi-nates, we identify the roads by displaying all the Cartesian coordinates in a map. Algorithm 1 is used to identify the FVLs; based on the FVLs and the sequences of timestamped points, we extract the UC information. DAMP may require a large number of contextual information to be collected and processed by a UE device. However, the new-generation UE devices have sufficient storage space; for example, in our simulation, the file (PLT file) to be maintained/used by a user is about 2.82 MB (for two months of GPS trace collection). The recent mobile devices (e.g., Samsung galaxy) can use .XML or .TXT files (instead of a database management system); these types of files do not require large storage space. Indeed, for a mobile device of 16 GB of storage space, DAMP will use 0.002% of this storage space, which is negligible. Table III shows the val-ues of the parameters used in our simulations; these parameters are selected according to the road topology of the prediction area (i.e., navigation zone). For example, the parameters in Table III are more appropriated for a Manhattan model (i.e., a 2-D environment with the roads arranged in a mesh shape).

TABLE III
SIMULATION PARAMETERS



Parameter	value	Parameter	value	Parameter	value
f_{th}	1/30	θ_{th1}	90°	θ_{th2}	45°

We define one parameter, which is denoted by A_p, to evaluate the performance of DAMP in terms of path similarity. In the literature [26], the distance error is used to measure error for predictive path queries. However, in some cases, the distance error is small while the predicted path is very different from the actual path; this justifies taking into account path similarity to measure the performance of DAMP. Let L_{act} be the actual location of the user after travel time d_t (which will become known only in the future), L_{pred} be the location of the user in the predicted path after travel time d_t (returned by path prediction model), E_{act} be the set of road segments that the actual path from path prediction origin to L_{act} contains, and E_{pred} be the set of road segments that the predicted path from path prediction origin to L_{pred} contains. Similar to [26], we measure path similarity A_p, which is defined as follows:

$$A_p(E_{act}, E_{pred}) = \frac{2.|E_{act} \cap E_{pred}|}{|E_{act}| = |E_{pred}|}. \qquad (12)$$

In the remainder of this paper, the terms path similarity and accuracy will be interchangeably used. Unless stated otherwise, in all simulation scenarios, we use the two months of the Microsoft Research Asia laboratory's data set (June–July 2012) to learn users' habits (in this case, we state that the length of the learning phase is 60 days); the learning phase denotes the period between the time at which mobility data collection is started and the time at which prediction is performed; the prediction phase comes after this phase; we use the last month of this data set (August 2012) as the prediction phase; this period is also used to compare the actual and predicted path similarity. We also assume that the path from the trip origin to the current location (where the prediction process is executed) corresponds to 500 m of the path from the trip origin to the destination, and the prediction length d_t is 2 h.

To compare DPM with the approach proposed in [28], we compute the accuracy of destination prediction Ad, which is defined as follows:

$$A_d = \frac{n_{bp}}{n_{tp}} \qquad (13)$$

where n_{bp} and n_{tp} denote the number of correct estimates (i.e., L_{act} and estimated destination are the same) and the total number of estimates, respectively.

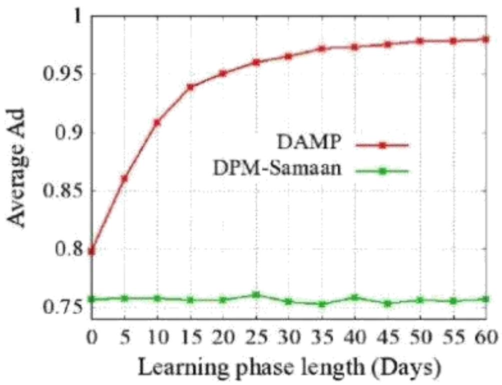


Fig. 5. Average destination prediction accuracy versus learning phase length variation.

B. Results Analysis

Simulation results are averaged over multiple runs; indeed, the simulation program is run *500* times; one run of the simulation program provides *ten* prediction units; a prediction unit contains a destination and the path toward this destination. For each run, we compute Ap (resp., Ad) using (12) [resp., (13)]; thus, to obtain the simulation results shown in Figs. 5–8, we compute the average of the *500* runs.

Fig. 5 shows the average accuracy of destination prediction when varying the length of the learning phase. We observe that for DAMP (resp., DPM-Samaan [28]), the average accuracy of destination prediction increases (resp., remains constant) with the length of the learning phase. This can be explained by the fact that DAMP uses historical data to perform destination prediction; indeed, the longer the length of the learning phase, the better the knowledge about users' mobility habits, and, ultimately, the higher the prediction accuracy; in contrast, DPM-Samaan uses only the user context (his goals and interests). Fig. 5 also shows that DAMP outperforms DPM-Samaan; DAMP provides an average accuracy that exceeds 0.85 for five days of learning phase length, whereas DPM-Samaan provides an average of 0.75, regardless of the number of days of learning phase length; overall, the average relative improvement (defined as [average Ad of DAMP—average Ad of DPM-Samaan]) of DAMP compared with DPM-Samaan is about 4% for 0 day of learning phase length and over 20% for 25 days or more of learning phase length. At 0 day of learning phase length (i.e., in case of lack of historical data), DAMP uses only the belief function of DPM-Samaan according to (6) and (7); yet, DAMP and DPM-Samman do not provide the same accuracy; this can be explained by the fact that DAMP performs the destination selection process after the preselection and clustering processes. At 30 days of learning phase length, DAMP accuracy is about 96%; according to (6) and (7), at 30 days of learning phase length, DAMP does not make use of the belief function proposed in [28]; indeed, for RL = 30 days (the maximum value of RL) and n = 30 days (which represents 30 days of learning phase), (6) becomes $ws(c_j) = P_{h,cl \rightarrow Ci}$; in this case, DAMP uses only the probability function.

Fig. 6 shows the average accuracy when varying the length of the learning phase. We observe that, for the four schemes, the average accuracy increases with the length of the learning phase. This is expected since the longer the length of the learning phase, the better the knowledge about users' mobil-ity habits, and, ultimately, the higher the prediction accuracy becomes when historical data are used to perform prediction. Fig. 6 also shows that DAMP outperforms AP1, AP2, and AP3; for example, DAMP provides an average accuracy of 0.55 for five days of learning phase length, whereas AP1 (more efficient than AP2 and AP3 in this scenario) provides an average of 0.20 for five days of learning phase length; overall, the average rela-tive improvement (defined as [average Ap of DAMP—average Ap of AP1]) of DAMP compared with AP1 is about 35% for five days of learning phase length. At ten days of learning phase length, DAMP accuracy is about 72%, whereas AP1 requires 60 days of learning phase length to provide the same accuracy; this means that DAMP requires a smaller learning phase length (one day compared with six days for AP1) to perform prediction with similar accuracy. This can be explained by the fact that 1) DAMP filters data taking into account the type-of-day (e.g., labor day, weekend, and weekday) and that 2) DAMP makes use of user context in addition to the mobility history traces. Indeed, user context along with historical data helps improve prediction accuracy. We also observe that AP1 outperforms AP2 and AP3; this can be explained by the fact that AP1, for prediction purposes, uses data filter (i.e., time-of-day); thus, when the length of the learning phase increases, the prediction accuracy increases; in this case, AP1 becomes more accurate than AP2 and AP3 when the length of the learning phase exceeds five days. In case of lack of historical data (i.e., learning phase length is equal to 0 day), DAMP accuracy is about 38%, whereas AP1, AP2, and AP3 accuracy is 0%; this can be explained by the fact that DAMP makes use of the direction toward the destination when there is no mobility history traces [see (11)]; AP1, AP2, and AP3 are only based on historical data; hence, without mobility history traces, they cannot perform a prediction.

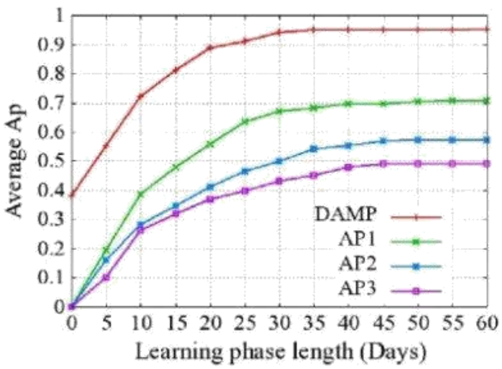


Fig. 6. Average prediction accuracy versus learning phase length variation.

Fig. 7 shows the average accuracy when varying the length of the path from the trip origin to the current location. We observe that for DAMP, AP1, and AP2 (resp., AP3), the average accuracy increases (resp., remains constant) with the length of the path from the trip origin to the current location (i.e., path already traveled by the user toward the destination); this can be explained by the fact that DAMP, AP1, and AP2 use the portion of the path already traveled by the user to compute the remaining path to the destination; in contrast, AP3 uses only the last

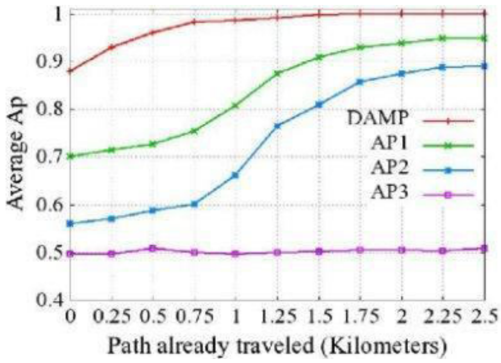


Fig. 7. Average prediction accuracy versus path-already-traveled variation.

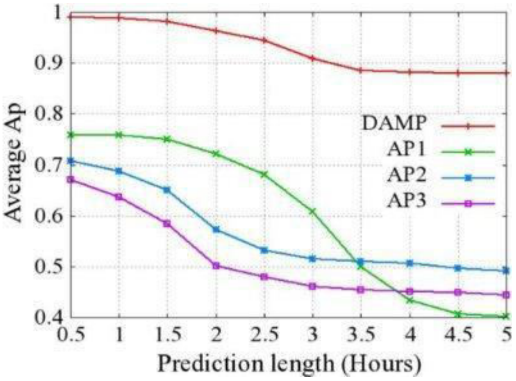


Fig. 8. Average prediction accuracy versus prediction length variation.

crossed location; indeed, DAMP, AP1, and AP2 try to match the current path (from the trip origin to the current location) to paths stored in their databases of the user's trajectory history. Thus, when the size of the current path increases, the prediction accuracy increases. Fig. 7 also shows that DAMP outperforms AP1, AP2, and AP3; DAMP provides an average accuracy of 0.89 for 0 km of path already traveled, whereas AP1 (more efficient than AP2 and AP3 in this scenario) provides an average of 0.70 for 0 km of path already traveled; the average relative improvement of DAMP compared with AP1 is about 19% for 0 km of path already traveled. We also observe that DAMP's (resp., AP1's) average accuracy increases more rapidly between 0 and 0.5 km (resp., between 0.5 and 1.5 km) of path already traveled by the user. This means that DAMP requires a smaller path already traveled by the user (about 0.5 km compared with 1.5 km for AP1) to predict the path with better accuracy. We also observe that, at 0 km of path already traveled (i.e., the prediction process is executed at the trip origin), DAMP performance is about 96%, whereas AP1 requires 1.25 km of path already traveled to provide the same performance. This means that AP1, compared with DAMP, requires that the user be located more closely to the destination to predict the path with better accuracy. This can be explained by the fact that 1) DAMP uses the direction from the current location to the destination to compute, at each location, the potential next location, and

2)DAMP filters historical data based on the type of the day. Fig. 8 shows the average accuracy prediction when varying the prediction length (i.e., d_t). We observe that, for the four schemes, the average accuracy decreases with the length of the prediction. This is expected since when the prediction length

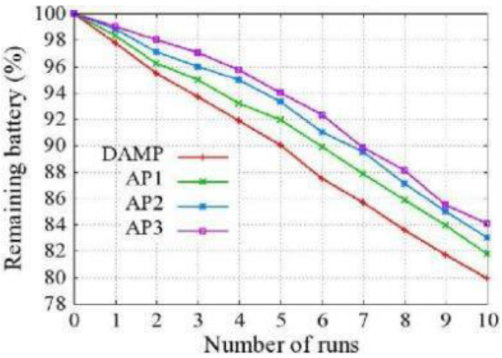


Fig. 9. UE energy consumption versus varying numbers of runs.

increases, the number of possible paths increases, and thus, the prediction accuracy decreases. Fig. 8 also shows that DAMP outperforms AP1, AP2, and AP3; this is mainly due to the fact that DAMP predicts a destination cluster (in opposition to a single destination) and makes use of the movement direction toward the destination cluster during the prediction process.

In particular, destination clustering allows the grouping of probable destinations (as a single destination: a cluster) when the length of prediction increases. This grouping reduces the number of probable destinations and increases the path predic-tion accuracy toward these probable destinations. DAMP pro-vides an average accuracy of 0.99 for 0.5 h of prediction length, whereas AP1 (more efficient than AP2 and AP3 in this sce-nario) provides an average of 0.76 for 0.5 h of prediction length; overall, the average relative improvement of DAMP compared with AP1 is about 23% for 0.5 h of prediction length. We also observe that AP2 (resp., AP3) outperforms AP1 around 3.5 (resp., 4) h of prediction length. This can be explained by the fact that AP2 and AP3, in contrast to AP1, for predic-tion purposes, consider the user's current direction (i.e., vec-tor/direction from the trip origin to the current location). Thus, when the prediction length increases, the prediction accuracy values of AP2 and AP3 are less impacted compared with those of AP1; indeed, the usage of the current direction reduces the number of probable paths and increases the path prediction accuracy; in this case, AP2 (resp., AP3) becomes more accurate than AP1 when the length of the prediction exceeds 3.5 (resp., 4) h. Although AP2 and AP3 use similar techniques as DAMP (e.g., user's current direction), they do not make use of the direction toward the destination. Indeed, when the length of the prediction is 3.5 (resp., 4) h, DAMP provides an average of 0.89 (resp., 0.89), whereas AP2 (resp., AP3) provides an average of 0.51 (resp., 0.45); overall, the average relative improvement of DAMP compared with AP2 (resp., AP3) is about 38% (resp., 44%). This means that DAMP predicts the remaining path to the destination with better accuracy despite the expansion of length of prediction. This can be explained by the fact that DAMP uses the direction from the current location to the destination cluster.

Fig. 9 shows the UE energy consumption when varying the number of runs (i.e., one run of the simulation program provides ten prediction units, and each prediction unit contains a destination and the path toward this destination).

We observe that, for the four schemes, the remaining battery energy decreases with the number of runs. This is expected since when the number of runs increases, the energy

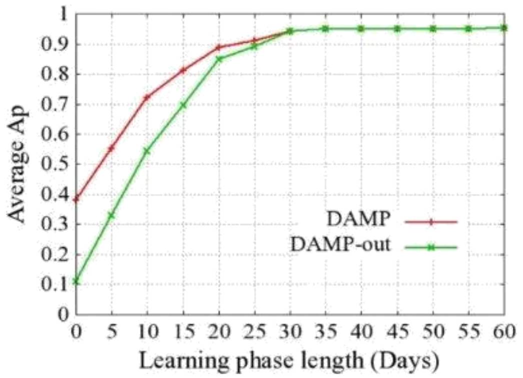


Fig. 10. Impact of *Bel*() on DAMP accuracy versus learning phase length variation.

consumption increases, and the remaining battery energy decreases. Fig. 9 also shows that AP3 consumes an average of 1.27% per run of battery energy, whereas DAMP consumes an average of 2.08% per run of battery energy; overall, the average relative improvement of AP3 compared with DAMP is about 0.81% per run of battery energy. The 0.81% battery energy consumption increase is a small price to pay for better path prediction accuracy. In this vein, it shall be noted that for mobile users with energy consumption constraints, some energy-aware settings can be envisioned in a way that the proposed solution is automatically disabled when the batteries of their devices go below a certain threshold. Furthermore, if the proposed solution is efficiently used for users without much constraint in energy consumption, the optimization and savings achieved in the network resources can be used to accommodate more mobile users with energy consumption constraints.

Fig. 10 shows the average accuracy when varying the length of the learning phase.

In the figure, the DAMP version not integrating the belief function, i.e., *Bel*(), adopted in [28] is referred to as DAMP-out. Fig. 10 shows that DAMP outperforms DAMP-out. Indeed, DAMP provides an average accuracy of 0.55 for five days of learning phase length, whereas DAMP-out provides an average of 0.32 for five days of learning phase length; overall, the aver-age relative improvement of DAMP compared with DAMP-out is about 23% for five days of learning phase length. According to (6) and (7), at 30 days of learning phase length, DAMP does not make use of the belief function proposed in [28]; DAMP and DAMP-out exhibit similar performances in terms of accuracy at 30 days of learning phase length.

In summary, the analysis of the simulation results shows that schemes that use data (e.g., user context) in addition to historical mobility traces outperform schemes that are limited to mobility traces when the size of historical mobility traces is not large enough (i.e., the length of the learning phase is not long enough). Likewise, historical mobility traces filtering increases the accuracy when the length of the learning phase increases. We also observe that schemes that consider the path from the trip origin to the current location outperform others when the length of path already traveled increases, whereas the schemes that consider the current movement direction (i.e., vector/direction from the trip origin to the current location) out-perform others when the length of prediction increases. Finally, taking into account the direction from the current location to the destination allows for improving performance despite the expansion of prediction length. We summarize DAMP evaluation findings as follows: 1) DAMP uses the path from the trip origin to the current location in the prediction process; although AP1 and AP2 use the path from the trip origin to the current location, they require a long path from the trip origin to the current location to predict, with better accuracy, the path from the current location to the destination; 2) DAMP uses movement direction in the prediction process; although AP2 and AP3 use movement direction, they do not consider the direction toward the destination; 3) DAMP uses user context and filters historical data based on the type of the day and the time of the day; this helps increase accuracy. Although AP1 uses similar data filter, it is limited to the days-of-week, and thus, it requires a long time (four times more than DAMP) to predict, with better accuracy, the path from the current location to the destination.

The communication complexity of DAMP is 0; indeed, DAMP does not communicate with the network system to predict users' mobility; all the databases and the processes are maintained/run by the UE. Admittedly, some communication overhead may become required if UE devices have to respec-tively report their predicted paths to the MN system so that their communications over the MN get optimized, e.g., as in [6]. The computational complexity of DAMP is $O(m) + O(g^2)$, where m is the number of FVLs, and g is the number of road segments of the NM. Indeed, DPM calculates m transition probabilities of FVL, whereas PPM calculates g^2 transition probabilities of road segment. We evaluated the computational complexity as a computation time on a Samsung Galaxy S4 (2 GB of RAM, 4*1.9 GHz of processor speed). For each run, we use a different mobility trace. The obtained results show that the computation time is smaller than 1 s for most users; this time can be greatly improved with optimal programming/implementation of DAMP in smart phones.

V. Conclusion

In this paper, we have introduced a destination prediction model, which is called DPM, and mobility path prediction model, which is called Ad hoc on demand routing protocol for predicting subsequent transitions of nodes across the mobility of users within a predefined time period. During destination prediction process the DPM selecting the destinations based on the following conditions: 1) the user's habits in terms of frequently visited locations, 2) the direction from the movement origin to the current location, and 3) the user's contextual knowledge. Then we used Ad hoc on Demand Vector Routing (AODV) protocol for selecting path between the source and destination nodes. The reason for choosing AODV is each node maintains a routing table for maintains the topology. So our proposed AODV chooses only the nodes with latest update in the routing table. Finally we can get better result on simulation by means of energy consumption and accuracy.

REFERENCES

1. T.-S. Chen, Y.-S. Chou, and T.-C. Chen, "Mining user movement behavior patterns in a mobile service environment," IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, vol. 42, no. 1, pp. 87–101, Jan. 2012.
2. M. Boc, M. Dias de Amorim, and A. Fladenmuller, "Near-zero triangular location through time-slotted mobility prediction," Wireless Netw. J.,vol. 17, no. 2, pp. 465–478, Feb. 2011.
3. U. Javed et al., "Predicting handoffs in 3G networks," SIGOPS Oper. Syst. Rev., vol. 45, no. 3, pp. 65–70, Dec. 2011.
4. Z. Becvar, P. Mach, and B. Simak, "Improvement of handover prediction in mobile WiMAX by using two thresholds," Comput. Netw., vol. 55, no. 16, pp. 3759–3773, Nov. 2011.
5. T. Taleb, A. Hafid, and A. Nadembega, "Mobility-aware streaming rate recommendation system," in Proc. IEEE GLOBECOM, Houston, TX, USA, Dec. 2011, pp. 1–5.
6. A. Nadembega, A. Hafid, and T. Taleb, "A destination prediction model based on historical data, contextual knowledge and spatial conceptual maps," in Proc. IEEE ICC, Ottawa, ON, Canada, Jun. 2012, pp. 1416–1420.
7. P. Salvador and A. Nogueira, "Markov modulated bi-variate Gaussian processes for mobility modeling and location prediction," in NETWORKING, vol. 6640, J. Domingo-Pascual, P. Manzoni, S. Palazzo, A. Pont, and C. Scoglio, Eds. Berlin, Germany: Springer-Verlag, 2011, pp. 227–240.
8. U. Rathnayake, M. Ott, and A. Seneviratne, "Network availability prediction with hidden context," Perform. Eval., vol. 68, no. 9, pp. 916–926, Sep. 2011.
9. A. Vassilya and A. Isik, "Predictive mobile-oriented channel reservation schemes in wireless cellular networks," Wireless Netw., vol. 17, no. 1, pp. 149–166, Jan. 2011.
10. Y. Li and I.-R. Chen, "Design and performance analysis of mobility management schemes based on pointer forwarding for wireless mesh networks," IEEE Trans. Mobile Comput., vol. 10, no. 3, pp. 349–361, Mar. 2011.