MAP Selection Algorithms Based on Future Movement Prediction Capability in Synthetic and Realistic Environment

Andrej Vilhar, Roman Novak, and Gorazd Kandus

Abstract: **Efficient mobility management involves micro-mobility principles. The performance of the Hierarchical Mobile IPv6 (HMIPv6) protocol, a representative micro-mobility approach, is affected by the Mobility Anchor Point (MAP) selection. In this paper, we propose a new selection method based on a prediction of the future movements of Mobile Nodes (MNs). The proposed algorithms exploit the information about the future availability of MAPs and choose those MAPs that assure a better service. An improvement to the evaluation methodology is also proposed. The algorithms are compared to each other not only in synthetic but also in realistic internet topologies, which has not been a practice in the past. The simulation results show promising improvements in terms of distance from chosen MAPs and frequency of MAP changes. Moreover, we showed that, for perceivable improvement of MAP selection, absolute accuracy of movement prediction is not required. As pioneers in the mobility management analysis in realistic environment, we ascertain that offering MAP services over more than one Autonomous System (AS) proves beneficial.**

***Index terms*: Hierarchical Mobile IPv6 (HMIPv6), Mobility Anchor Point (MAP), MAP selection algorithm, movement prediction, realistic topologies**

I. INTRODUCTION

In modern society, ubiquitous radio access to the Internet is becoming a reality. In such an environment, the mobility of communication devices is well supported from the physical point of view. In addition, to achieve continuous reachability and seamless handovers, efficient higher layer mobility management has to be introduced [1]. One of the most promising mobility management protocols at the macro-level is Mobile IPv6 [2], proposed by the IETF community, which has to be supplemented with micro-mobility solutions [3].

The most notable micro-mobility approach, Hierarchical Mobile IPv6 (HMIPv6), is defined in an RFC 4140 document [4]. Besides Home Agent (HA), which is also present in basic Mobile IPv6, HMIPv6 contains an additional entity called Mobility Anchor Point (MAP). In principle, HA manages the location of Mobile Node (MN) on the macro-level, while MAP keeps track of the precise location of MN on the micro-level. As MAP is always present in the routing path and MN may choose among different MAPs, the MAP selection technique affects the efficiency of the protocol significantly.

The most basic MAP selection algorithm, proposed by the authors of HMIPv6 protocol, always selects the MAP that is furthest in terms of routing hops. The intention of such an approach is to minimize the number of required MAP changes, thus reducing signalling overhead and handover delays. The two most commonly recognized drawbacks of furthest MAP selection are high load burden on the most distant MAPs and unnecessary signalling delays for the MNs which move in the scope of nearer MAPs. To overcome these drawbacks, algorithms have been proposed [5 - 12]. To evaluate their solutions, the researchers have used regular synthetic topologies, typically of tree-like shape. The actual improvement of the proposed MAP selection algorithms in the real Internet has not been studied.

This paper is original in two aspects. First, we propose a novel way of selecting MAPs. For the paths travelled by MNs that repeat frequently, MNs may predict which MAPs will be available in the future. We suggest the use of this information to improve MAP selection efficiency. Using simulations, we have evaluated the proposed algorithm and compared it to other existing solutions. The results indicate a notable improvement if any information about the future location of MAPs is available.

The second contribution of the paper is the analysis of the proposed algorithm in the realistic internet topology. The analysis confirms the algorithm's improved performance and opens a new research direction in the field. Although HMIPv6 specification [4] and some other studies indirectly suggest that MAP coverage area spreads within a single operator's network, no explicit prohibition of disregarding this suggestion can be found. We believe that MAP domains, which spread over more than one Autonomous System (AS),

Manuscript received February, 2008 and revised June, 2008.

A part of this Paper was presented at the International Conference on Software, telecommunications and Computer Networks (SoftCOM) 2007.

Authors are with Jožef Stefan Institute, Department of Communication Systems, Ljubljana, Slovenia (e-mail: {andrej.vilhar, roman.novak, gorazd.kandus}@ijs.si)

are beneficial, because they combine access diversity, low signalling costs and low signalling delays. The simulation results are in favour of this belief.

The paper is structured as follows. Section 2 gives an overview of related algorithms proposed by the research community. In section 3, we present our idea of predictive MAP selection and describe the operation of each algorithm analyzed in the paper. The descriptions of the algorithms are corroborated by pseudocodes. The simulation models and simulation results for synthetic topologies and realistic topologies are discussed in section 4 and section 5. Section 6 concludes the paper.

II. OVERVIEW OF RELATED ALGORITHMS

MAP selection algorithms can be classified into three distinct groups: speed-based algorithms, history-based algorithms and adaptive algorithms.

Algorithms based on the speed of an MN, measured in handovers per unit time, were suggested in [5, 6]. Faster MNs select more distant MAPs, as it is believed that faster movement leads to a larger moving area. In [7] the use of a speed-based algorithm in synthetic meshed networks was studied.

The authors in [8, 9] argue that the speed of an MN is not necessarily directly related to the size of a moving area, and suggested a new, history-based approach. Basically, their algorithm keeps track of available MAPs in the previous interval and selects the one that is nearest and was available for the whole interval. In [10] the full availability of an MAP is not required. Instead, a certain threshold of required availability has to be reached.

Two novelties are introduced in the adaptive algorithms [11, 12]. First, besides the MN speed, they take into account MN's activity in terms of communication traffic. The ratio of the two values, the session-to-mobility ratio, affects the MAP selection. MNs with higher session-to-mobility ratio tend to choose nearer MAPs. The second novelty is the introduction of so-called cost functions, which try to take into account all the parameters important for optimal MAP selection. The MN continually calculates the cost functions and selects MAPs for which the cost functions are lowest.

A comparison of the above MAP selection algorithms is available in [13].

III. PREDICTIVE MAP SELECTION ALGORITHM

A. Basic Idea

The majority of existing proposals assume complete randomness of MN movement. However, MNs can move according to certain repeating patterns. The frequency of repetitions may vary in the order of hours or days, up to weeks, months or even years. Illustrative examples of repeating patterns of an average person are travel paths to work, sport activities, weekend trips, vacations, etc. In some cases, the movement patterns may be well determined. Public transport vehicles such as buses, trains and airplanes move along very deterministic travel paths. It is expected that a concept of network mobility [14] with Mobile Routers (MRs) will be deployed on such vehicles. MRs will play the role of an intervener which enables internet connection to all MNs inside a vehicle. As MRs will manage the mobility on behalf of MNs, the information about MAP availability on repeating journeys may be acquired and prediction of future moves is possible to some extent.

The MAP selection procedures can be implemented on any device that manages mobility, i.e. either MR or MN. For clarity, and in compliance with terminology in the field, we refer to both MR and MN as MN in the following.

Our proposal is based on the assumption that the knowledge of MAP availability in the future can be used to improve MAP selection. The proposals in closest relation to our algorithms are history-based selection algorithms [8 - 10]. They base MAP selection on the knowledge of MN movement in the recent past. Their approach is an attempt to consider the moving pattern of MNs, but leads to selection of MAPs that are optimal in the parts of travel paths that have already happened.

B. Description of Analyzed Algorithms

In order to estimate the benefits offered by the knowledge of future movement, we compared three existing reference approaches to three types of our predictive MAP selection algorithm. All six algorithms are described in the following. Proposed predictive algorithms are further accompanied by pseudocodes (Appendixes B, C, D) in order to enable thorough study to other researchers in the field. For the reference, a pseudocode for basic furthest approach is also given (Appendix A), while pseudocodes for speed-based and history-based algorithms are left out due to space shortage. The definitions of classes, used by the algorithms are given in Appendix E. Note that for every algorithm, MN keeps its registration to the previously selected MAP as long as possible. Another MAP is selected after MN leaves the coverage area of the previous MAP.

The existing reference approaches used were the furthest, speed-based and history-based MAP selection algorithms, designated as *'furthest'*, *'speed'* and *'history & nearest'*, respectively. *'Furthest'* is the algorithm suggested in [4]. It selects at random an MAP from the group of most distant MAPs (Appendix A). In *'speed'* the fastest MNs select the most distant MAPs and vice versa. The exact distribution function of speed interval in relation to MAP distance is defined in advance. During the simulations, the distribution remains unchanged. Note that the pseudocode for *'speed'* corresponds to a subcode of the pseudocode in Appendix B (lines 0-12 and 26-29). In *'history & nearest'*, MNs observe the MAPs available in the past movement. The length of observation interval, measured in number of Access Router (AR) changes, is predetermined. MN randomly chooses the nearest MAP that is currently available and has been available for the whole observation interval. If such an MAP does not exist, the MN selects the nearest MAP from those that were available for the most of the observed time.

The first two types of our proposed predictive algorithm are based on *'speed'* with extension of prediction capability. By using *'speed'*, the algorithms first determine a group of MAPs with appropriate hop distance from an MN. In the next step, the two algorithms select from this group an MAP which is predicted to be available for the longest period of time. The two algorithms differ mainly in prediction accuracy. The first, designated as *'speed & future'*, has complete information about the future availability of MAPs. It always selects the one that will be available for the longest period of time (Appendix C). The second algorithm, designated as *'speed & direction'* only has information about the direction in which the MN is going to move in the long term (Appendix B). As the MN's temporal movement is generally random, *'speed & direction'* may not always select the optimal MAP.

The last version of our predictive algorithm, designated as *'future & nearest'*, functions in a similar way to *'history & nearest'*. The main difference is, that *'future & nearest'* uses a future oriented selection interval instead of a past observation interval (Appendix D). The length of the interval is predetermined. The algorithm selects the nearest MAP that will be available for the whole selection interval. The information about the future availability of MAPs is complete. If no MAPs are available for the duration of the selection interval, the MN selects the nearest MAP from those that will be available for most of the time. Longer selection intervals lead to less frequent MAP changes but result in more distant MAPs and vice versa. In practice, the actual value will depend on the current conditions and MN preferences.

IV. PERFORMANCE ANALYSIS FOR SYNTHETIC TOPOLOGIES

A. Simulation Model

For the purpose of performance evaluation we used a simulation model, implemented in C#. Unlike the majority of proposed solutions, we used a non-tree-like synthetic topology, similar to that in [7], achieving better simulation of random MAP coverage area overlaps. We simulated 100 MAPs, distributed over 233 ARs, each AR covering a single radio cell. A 36 radio cell cut-out of an example topology is depicted in Fig. 1. The topology consists of five levels of hierarchy. For clarity, the ARs located at the lowest level are not shown in the picture. MAPs are positioned in four higher levels. Each higher level is further away from the radio cell in terms of routing hops, and consequently in transmission delay. MAPs are distributed randomly according to the predetermined distribution probability. Higher MAPs are fewer but cover a larger number of radio cells. MAPs at the levels 2, 3, 4 and 5 cover 4, 8, 12 and 16 radio cells respectively.

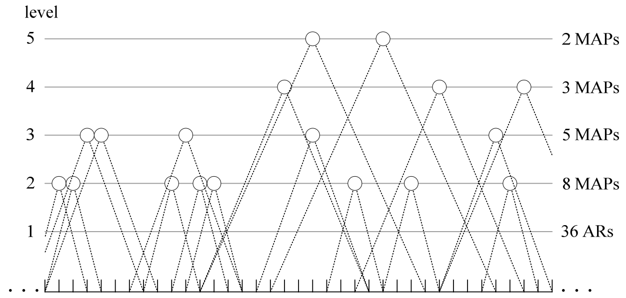

Fig. 1. A 36 radio cell cut-out of an example topology

Each MN starts its journey from the centre of the topology. During the simulation time, MNs travel between 0 and 100 cells, according to their selected constant speed. The speed interval is distributed evenly over the MNs. The direction of MN movement is chosen for each individual AR change. The probability of selecting a particular direction is determined in advance. The topology is large enough to be virtually endless for MNs, meaning that, even for fastest MNs which move solely in one direction, the border conditions do not differ from those at the starting point.

We repeated simulation runs 10.000 times for each tested algorithm and each selected probability of forward movement (right direction). For each simulation run, a new random topology was created and a new MN speed selected. The average MAP distribution, with accompanying standard deviation values in relation to distance from radio cells, is given in Table I.

TABLE I
AVERAGE MAP DISTRIBUTION IN RELATION TO DISTANCE FROM RADIO CELLS

level	number of MAPs
5	13,91 ± 0,03
4	19,69 ± 0,03
3	27,68 ± 0,03
2	38,71 ± 0,04

B. Evaluation Metrics

Several communication parameters are important for evaluating algorithm performance. The most notable examples are signalling delay, processing power and bandwidth usage. These values depend heavily on chosen characteristics of the simulation topology. In addition, the relative importance of one parameter to another is not clear, which leads to difficulties in the evaluation process.

In our study, we used a similar approach to that introduced in [10]. We observed two easily measurable metrics, average distance from the chosen MAP and average number of MAP changes during the simulation time. The former is tightly related to signalling delay, intra-domain route optimality [15] and load balance, while the average number of MAP changes directly influences the amount of signalling overhead and additional signalling delays. Both metrics should be as low as possible for better performance of the MAP selection algorithm.

Again, the importance of one metric relative to another is not straightforward and may depend on MN's preferences. However, if the value of one metric is the same for the two analyzed algorithms, the comparison of the second metric indicates unambiguously which algorithm is better. The same conclusion holds if the values of both metrics are lower for one of the compared algorithms. In both cases, the algorithms are said to be *comparable*. If the value of one evaluation metric is higher and the value of another is lower, a straightforward conclusion about optimality of the compared algorithms is not possible. In this case, the algorithms are *not comparable*.

C. Simulation Results for Synthetic Topologies

For every algorithm described in this paper, except for *'furthest'*, the relation of the measured metrics can be changed by varying the algorithm parameters. For example, one set of parameters results in shorter distances from the chosen MAP, at the cost of more frequent MAP changes, while another set leads to more distant MAPs with less frequent changes. The analysis of simulation results is given in two parts. In the first part, we selected parameters in such a way as to make the analyzed algorithms fully comparable to *'furthest'*. In the second part, precedence is given to comparison of the algorithms with parameters that lead to selection of nearer MAPs.

In order to achieve comparability of *'furthest'* and *'speed'*, we adjusted the speed distribution in *'speed'* in such a way that MNs tended to choose more distant MAPs. We kept raising the probability of choosing higher level MAPs until the number of required MAP changes was similar for the two algorithms. As shown in Fig. 2, the values match very well, regardless of the probability of MN's forward movement. For the same parameters, *'speed'* chooses MAPs that are nearer on average, as depicted in Fig. 3. Therefore, *'speed'* is the better algorithm, which confirms findings of related studies.

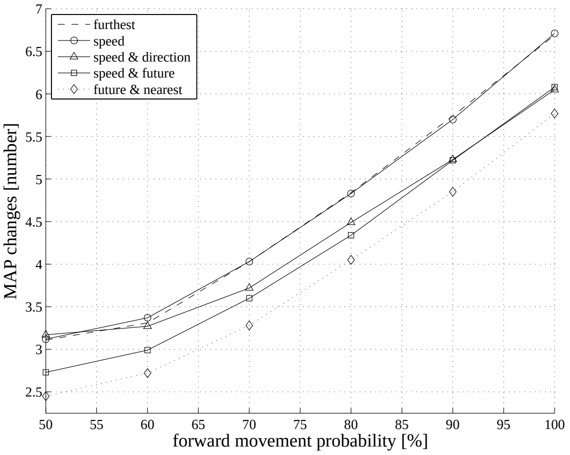

Fig. 2 Average number of MAP changes (1)

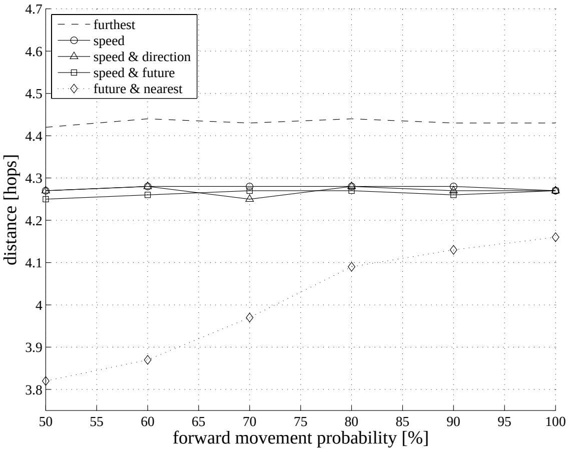

Fig. 3 Average distance from chosen MAP (1)

Next, we studied the performance of our solutions *'speed & future'* and *'speed & direction'* in relation to the existing *'speed'* solution, keeping the same MN speed distribution. As the speed distribution is the same, the average distances from chosen MAPs are very similar for each of the three algorithms (Fig. 3). However, for our algorithms the average number of MAP changes is lower by up to 12,5% than with *'speed'*. This implies that by just adding information about future movement of MNs, MAP selection is improved. Furthermore, a closer inspection of the curves in Fig. 2 reveals the relation between the accuracy of prediction and algorithm improvement capability. At the left border of the x-axis, the prediction that MN will move forward is correct in 50% of cases. Implicitly, this means that no useful information about the future movement of MN is available for *'speed & direction'*. Hence, at this point, *'speed & direction'* gives the same results as *'speed'*, which provides no prediction at all. At the opposite extreme, the right hand border of the x-axis, the prediction that MN will move forwards is always correct. In this case, *'speed & direction'* leads to the same results as *'speed & future'*, which has full information about future movement of MN. In between these two extreme points, the performance of *'speed & direction'* is poorer than *'speed & future'* but better than that of *'speed'*. The actual improvement depends on the level of prediction accuracy.

The duration of selection interval in *'future & nearest'* was set to be infinite in the first part of the analysis. *'Future & nearest'* performs even better than *'speed & future'* (Fig. 2 and Fig. 3). As both algorithms have full information about the future movement of MNs, the performance of predictive algorithms obviously depends also on the manner of use of future movement information. In this particular case, choosing nearest MAPs from those available in the future proves to be better than choosing MAPs according to MN's speed.

In the second part of the analysis, we focus on the algorithms with accompanying parameters that tend to choose nearer MAPs, and change them more frequently. As shown in

Fig. 4 and Fig. 5, these algorithms are no longer comparable to *'furthest'*, but are comparable to one another. The observation interval in *'history & nearest'* and the selection interval in *'future & nearest'* were both set to 4. The speed intervals in *'speed'* were distributed evenly in relation to the distance from MAPs.

The performance of *'history & nearest'* is better than that of *'speed'* in the case where MN moves forwards and backwards with equal probability. With increasing probability of movement in one direction, the number of MAP changes with *'history & nearest'* grows markedly. At 60% and 70% probability of forward movement, the algorithms are not comparable. At 80%, where the algorithms are comparable again, *'speed'* already outperforms *'history & nearest'*. When the forward movement probability reaches 100%, *'speed'* becomes the significantly better algorithm.

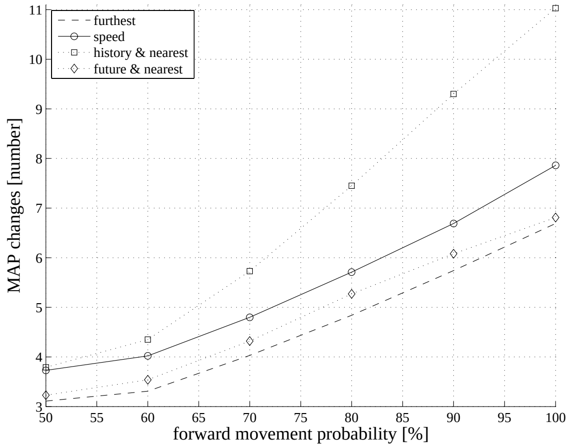

Fig. 4 Average number of MAP changes (2)

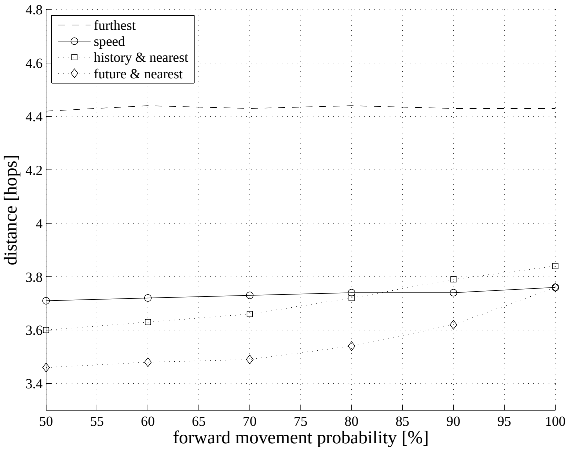

Fig. 5 Average distance from chosen MAP (2)

The reason for these results is that *'history & nearest'* chooses only from those MAPs that were available long enough in the past interval. This rule eliminates some of the MAPs that are available for a long period in the future. The method proves to be relatively effective only if the patterns of movement repeat frequently during the same journey.

According to Fig. 4 and Fig. 5, our *'future & nearest'* always outperforms the existing solutions *'history & nearest'* and *'speed'*. Exact comparison to *'furthest'* is not possible, but an intuitive estimation strongly indicates better performance of *'future & nearest'*, especially for 50% and 100% forward movement probability.

Overall, the results imply that, of the analyzed algorithms, *'future & nearest'* performs best.

V. PERFORMANCE ANALYSIS FOR REALISTIC TOPOLOGIES

Researchers of MAP selection algorithms have performed simulations on regular synthetic topologies based on some form of hierarchy. To the best of our knowledge, none of the past studies attempted to apply realistic internet topologies, which we believe could lead to higher reliability and relevance of the results. This paper appears to be the first such attempt.

A. Annotation and Usage of Realistic Topologies

Three requirements have to be satisfied in studied topologies to be adequate for mobility management analysis:

i) distinction of access nodes from non-access nodes,
ii) distinction of MAP nodes from non-MAP nodes,
iii) and determination of the MAP coverage areas.

The topology in Fig. 6a represents a tree-like shape, which is typically used by researchers in the field. Following simple hierarchical pattern of layers, access nodes, MAP nodes and MAP coverage areas are determined. MAPs are positioned in the higher layers, while access nodes are at the lowest layer. Arrows point in the direction of MAP advertisement propagation. In realistic topologies, the hierarchy is not exactly defined. Consequently, more sophisticated and systematic annotation procedure is required.

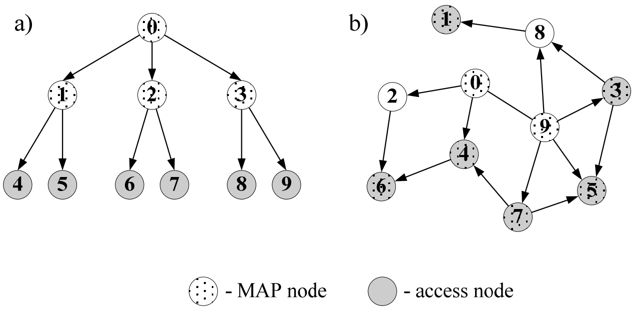

Fig. 6 Example 10-node topologies:
a) tree, b) random topology.

The real Internet exhibits two fundamental levels of granularity, the router level and the Autonomous System (AS) level, where an AS is a collection of networks and routers with a common routing policy. The hierarchical patterns are better investigated at the AS level, which is one of the reasons we picked the AS level for further analysis.

To satisfy requirements i) and ii), we use hierarchical decomposition of ASs to five hierarchical levels as suggested in [16]. The lowest in hierarchy are the customers, which carry no transit traffic. Together with small ISPs, customers represent the edge of the Internet, where the last-hop access is expected to be offered. Thus, we select customers and small ISPs to be access nodes. The rest of the ASs are part of the core. They are further subdivided into three levels, dense core, transit core and outer core, where the dense core represents the highest level, carrying the most transit traffic. In contrast to wireless access provision, MAP services could be extended into the core, because the infrastructural requirements are low while MAP coverage area spreads significantly. In our analysis, we experimented with two scenarios of MAP node designation. In the first, MAPs are positioned only on the edge of the Internet, in customers and small ISPs. In the second, MAP service is offered also from the core.

In Fig. 6b, an example of small AS-level topology is depicted. Each AS is represented as a node. Note that nodes may take both functions, i.e. access function and MAP function (nodes 1, 3, 4, 5, 6 and 7), while some nodes may not take any function (nodes 2 and 8).

Requirement iii) demands an annotation of links in the topology. Between pairs of ASs, different types of business relationships exist naturally, provider-to-customer and peer-to-peer being the most frequent representatives [17, 18]. We chose to mark the latter as undirected links, while provider-to-customer relationships are labelled as directed links. Providers are usually larger in size and sell network access to their customers. Intuitively, MAP services will also be offered in this direction, from providers to customers. Thus, we point the directed links towards the customer. The MAP advertisements are propagated along the direction of links and can traverse any number of nodes. Also, we allow the propagation of MAP advertisements over the undirected peer-to-peer links, but only if the link represents the first hop in the propagation path.

We illustrate the MAP advertisement propagation by the following examples in Fig. 6b. The MAP at node 7 is advertised by access nodes 6, 4, 5. It is advertised also by itself, as node 7 is both MAP and access node. The MAP at node 3 is seen at access nodes 1, 3 and 5. The node 8 forwards the advertisement to node 1, but does not advertise MAP by itself, because it is not an access node. The MAP at node 5 is only advertised by node 5 itself, while the MAPs at nodes 0 and 9 are advertised by all access nodes in the topology.

B. Simulation Model and Metrics

In this part of the analysis we focus on two algorithms, the *'future & nearest'*, which performed best at synthetic topologies, and the *'furthest'* as a baseline reference.

The simulations are based on the measured topologies with known AS relationships, provided by the Cooperative Association for Internet Data Analysis (CAIDA) [19]. The CAIDA researchers extract existing AS links from RouteViews snapshots [20], taken at 8-hour intervals over a 5-day period. The types of AS relationships are then inferred using the algorithm described in [18]. Each week, a newly derived Internet topology is available. We apply our annotation procedure to the selected topologies. As a result, given realistic Internet topology satisfies all three requirements, i), ii) and iii).

For each particular set of parameters, simulation run was repeated 10.000 times. For each run, MN changed 500 access nodes. We assume that the probability of selecting particular access node as next on the path is proportional to node's topological proximity to the previous access node. By default, we chose probability movement factor 3, meaning that one hop closer access node is selected with a three times higher probability.

We observed the average distance of access node to MAP node and the average number of changed access nodes per MAP change (ASs/MAP). The latter metric is analogous to the average number of MAP changes, but enables additional analysis. If this metric is much higher than 1, the MAP services over multiple ASs are well justified. Note that higher value means better performance of the algorithm.

C. Simulation Results for Realistic Topologies

We performed simulations for topologies, obtained by CAIDA during different periods of time. The differences with respect to time are minor. In the following we present the results for September 2007 topology.

In Table II, the results are shown that apply to MAPs positioned only in customers and small ISPs. Regardless of the MN's probability movement factor, the value of ASs/MAP is practically the same for both algorithms, while the distance is slightly shorter for the *'future & nearest'*. The basic *'furthest'* algorithm obviously suffices in this scenario, as even the *'future & nearest'* does not show significantly better performance. Despite that, offering MAP services in distant ASs is well justified, especially for highly localized movement of MNs with movement factor 10 or more.

TABLE II
COMPARISON OF THE ALGORITHMS
(MAPS ARE LOCATED IN CUSTOMERS AND SMALL ISPS)

mov. factor	metric	*'furthest'*	*'future&nearest'*
3	distance	0.65	0.47
	ASs/MAP	1.83	1.85
10	distance	0.56	0.50
	ASs/MAP	3.43	3.44

The results in Table III are valid for the scenario with MAP services offered also from the core. In this case, the difference in the performance between the algorithms becomes evident. *'Future & nearest'* performs much better, especially in terms of ASs/MAP. However, the selected MAPs are mostly positioned in the dense core, i.e. 90% of the MN's movement time (Fig. 7). This does not seem very realistic, because it is not likely that dense core ASs will actually offer MAP services to this extent. If the distance between access nodes and MAP nodes is limited to maximum 2 hops, the time share drops to approximately 50%, whereas limiting the distance to maximum 1 hop, leads to 30% time share or less (Fig. 7). Even with this strict limit of maximum 1 hop, which leads to more realistic results, the difference between the algorithms remains relevant and the usage of MAPs outside the scope of their ASs remains well justified.

TABLE III
COMPARISON OF THE ALGORITHMS
(MAPS ARE LOCATED AT ANY LEVEL OF THE HIERARCHY)

max. dist.	metric	*'furthest'*	*'future&nearest'*
∞	distance	2.56	2.41
	ASs/MAP	40.52	58.00
2	distance	1.73	1.49
	ASs/MAP	4.56	7.54
1	distance	0.87	0.68
	ASs/MAP	1.87	2.66

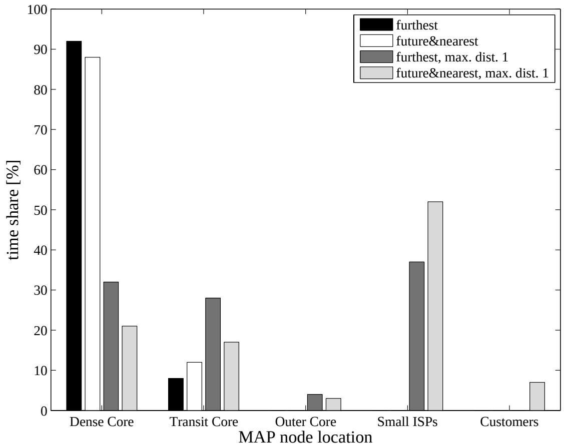

Fig. 7 Time shares of MAPs, selected at certain levels of hierarchy

VI. CONCLUSION

In this paper, we implemented a novel idea for improving MAP selection algorithms by using information about the future movement of MNs. The proposed solution shows significant improvement in terms of distance from chosen MAPs and frequency of required MAP changes. Implicitly, this means lower intra-domain signalling delays, closer to optimal intra-domain routing paths, better load balancing on MAPs, lower amounts of inter-domain signalling overhead and less frequent inter-domain signalling delays. Moreover, it was shown that, even if the prediction of future movement is not completely accurate, the benefits of the predictive algorithm remain. The grade of predictive algorithm improvement turns out to be directly related to the completeness of future movement information.

The findings were verified using both synthetic and realistic internet topologies. The latter were obtained by annotating the measured topologies of the real Internet. We introduced this novelty in order to increase the relevance of the results and strengthen the motivation for the search of improved MAP selection algorithms. Interestingly, even if our algorithm always outperforms the basic furthest algorithm, the results show that under certain realistic scenarios the differences may be smaller. Obviously, the topology structure affects the potential of MAP selection algorithms to improve the basic furthest MAP selection. As a derivative of the study, we showed that offering MAP services over multiple ASs proves beneficial for the end-users.

We are aware that there is still a space for improving the described model of realistic topologies. As the micro-mobility principles in the Internet are nowadays not in general use, patterns of mobility management relations cannot be inferred from the real environment. Therefore, the search for better topology models is a challenging open issue.

APPENDIX A

ALGORITHM: *'furthest'*
INPUT: previous mobility anchor point (anchor_point PreviousMAP), current access point (access_point CurrentAR)
OUTPUT: next mobility anchor point (anchor_point NextMAP)

ALGORITHM *'furthest'*

```
IF PreviousMAP is not in CurrentAR.ListOfMAPsSeen THEN

    SET NextMAP to first occurrence in CurrentAR.ListOfMAPsSeen
    FOREACH anchor_point MAP in CurrentAR.ListOfMAPsSeen
        IF MAP.DistanceToMN > NextMAP.DistanceToMN THEN
            SET NextMAP to MAP
        ENDIF
    ENDFOREACH

    RETURN NextMAP
ELSE
    RETURN PreviousMAP
ENDIF

ENDALGORITHM
```

APPENDIX B

ALGORITHM: *'speed & direction'*
INPUT: previous mobility anchor point (anchor_point PreviousMAP), current access point (access_point CurrentAR), mobile node (mobile_node MobileNode)
OUTPUT: next mobility anchor point (anchor_point NextMAP)

ALGORITHM *'speed & direction'*

```
IF PreviousMAP is not in CurrentAR.ListOfMAPsSeen THEN

    SET ChosenDistance to 0
    WHILE MobileNode.Speed >
    CurrentAR.SpeedDistribution[ChosenDistance]
        INCREMENT ChosenDistance
    ENDWHILE

    FOREACH anchor_point MAP in CurrentAR.ListOfMAPsSeen
        IF MAP.DistanceToMN equals ChosenDistance THEN
            SET NextMAP to MAP
            BREAK
        ENDIF
    ENDFOREACH
```

```

    FOREACH anchor_point MAP in CurrentAR.ListOfMAPsSeen
        IF ForwardProbability > 50 THEN
            IF MAP.DistanceToMN equals ChosenDistance AND
               MAP.RightBorder > NextMAP.RightBorder THEN
                SET NextMAP to MAP
            ENDIF
        ELSE
            IF MAP.DistanceToMN equals ChosenDistance AND
               MAP.LeftBorder < NextMAP.LeftBorder THEN
                SET NextMAP to MAP
            ENDIF
        ENDIF
    ENDFOREACH

    RETURN NextMAP
ELSE
    RETURN PreviousMAP
ENDIF

ENDALGORITHM
```

APPENDIX C

ALGORITHM: *'speed & future'*
INPUT: previous mobility anchor point (anchor_point PreviousMAP), current access point (access_point CurrentAR), mobile node (mobile_node MobileNode)
OUTPUT: next mobility anchor point (anchor_point NextMAP)

ALGORITHM *'speed & future'*

```
IF PreviousMAP is not in CurrentAR.ListOfMAPsSeen THEN

    SET ChosenDistance to 0
    WHILE MobileNode.Speed >
    CurrentAR.SpeedDistribution[ChosenDistance]
        INCREMENT ChosenDistance
    ENDWHILE

    SET AvailableMAPsList to empty
    FOREACH anchor_point MAP in CurrentAR.ListOfMAPsSeen
        IF MAP.DistanceToMN equals ChosenDistance THEN
            SET MAP.TimeSeen = 1
            ADD MAP to AvailableMAPsList
        ENDIF
    ENDFOREACH

    FOREACH access_point AR in MobileNode.ObservedARs
        FOREACH anchor_point MAP in AR.ListOfMAPsSeen
            IF MAP is in AvailableMAPsList AND MAP.TimeSeen
            equals position of AR in MobileNode.ObservedARs
            THEN
                INCREMENT MAP.TimeSeen
            ENDIF
        ENDFOREACH
    ENDFOREACH

    SET MaxPresence to 0
    FOREACH anchor_point MAP in AvailableMAPsList
        IF MAP.TimeSeen > MaxPresence THEN
            SET MaxPresence to MAP.TimeSeen
        ENDIF
    ENDFOREACH

    FOREACH anchor_point MAP in AvailableMAPsList
        IF MAP.TimeSeen equals MaxPresence THEN
            SET NextMAP to MAP
            BREAK
        ENDIF
    ENDFOREACH

    RETURN NextMAP
ELSE
    RETURN PreviousMAP
ENDIF

ENDALGORITHM
```

APPENDIX D

ALGORITHM: *'future & nearest'*
INPUT: previous mobility anchor point (anchor_point PreviousMAP), current access point (access_point CurrentAR), mobile node (mobile_node MobileNode)
OUTPUT: next mobility anchor point (anchor_point NextMAP)

ALGORITHM *'future & nearest'*

```
IF PreviousMAP is not in CurrentAR.ListOfMAPsSeen THEN

    SET AvailableMAPsList to empty
    FOREACH anchor_point MAP in CurrentAR.ListOfMAPsSeen
        SET MAP.TimeSeen = 1
        ADD MAP to AvailableMAPsList
    ENDFOREACH

    FOREACH access_point AR in MobileNode.ObservedARs
        FOREACH anchor_point MAP in AR.ListOfMAPsSeen
            IF MAP is in AvailableMAPsList AND MAP.TimeSeen
            equals position of AR in MobileNode.ObservedARs
            THEN
                INCREMENT MAP.TimeSeen
            ENDIF
        ENDFOREACH
    ENDFOREACH

    SET MaxPresence to 0
    SET AppropriateMAPsList to empty
    FOREACH anchor_point MAP in AvailableMAPsList
        IF MAP.TimeSeen >= MobileNode.PresenceRequired THEN
            ADD MAP to AppropriateMAPsList
        ELSEIF MAP.TimeSeen > MaxPresence THEN
            SET MaxPresence to MAP.TimeSeen
        ENDIF
    ENDFOREACH

    IF AppropriateMAPsList is empty THEN
        FOREACH anchor_point MAP in AvailableMAPsList
            IF MAP.TimeSeen equals MaxPresence THEN
                ADD MAP to AppropriateMAPsList
            ENDIF
        ENDFOREACH
    ENDIF

    SET NextMAP to first occurrence in AppropriateMAPsList
    FOREACH anchor_point MAP in AppropriateMAPsList
        IF MAP.DistanceToMN < NextMAP.DistanceToMN THEN
            SET NextMAP to MAP
        ELSEIF MAP.DistanceToMN equals NextMAP.DistanceToMN
        AND MAP.TimeSeen > NextMAP.TimeSeen THEN
            SET NextMAP to MAP
        ENDIF
    ENDFOREACH

    RETURN NextMAP
ELSE
    RETURN PreviousMAP
ENDIF

ENDALGORITHM
```

APPENDIX E

```
CLASS anchor_point
        DistanceToMN
        TimeSeen
        LeftBorder
        RightBorder
ENDCLASS

CLASS access_point
        ListOfMAPsSeen
        SpeedDistribution
ENDCLASS

CLASS mobile_node
        Speed
        ForwardProbability
        ObservedARs
        PresenceRequired
ENDCLASS
```

REFERENCES

[1] D. Saha, A. Mukherjee, I. S. Misra, M. Chakraborty: "Mobility support in IP: A survey of related protocols", IEEE Network Magazine, vol.18, no.6, pp.34-40, November/December 2004.

[2] D. Johnson, C. Perkins, J. Arkko: "Mobility support in IPv6", RFC 3775, June 2004.

[3] A.T. Campbell, J. Gomez, S. Kim, Z. Turanyi, C.-Y. Wan, Z.R. Turányi, A. Valkó: "Comparison of IP micro-mobility protocols", IEEE Wireless Commun. Magazine, vol.9, no.1, pp.72-82, February 2002.

[4] H. Soliman, C. Castelluccia, K. El Malki, L. Bellier: "Hierarchical mobile IPv6 mobility management (HMIPv6)", RFC 4140, August 2005.

[5] A. Huszak, S. Imre: "Agent selection algorithm in hierarchical mobile networks", Proc. 5th Conf. on Telecommun. ConfTele, Tomar, Portugal, April 2005.

[6] K. Kawano, K. Kinoshita, K. Murakami: "A mobility-based terminal management in IPv6 networks", IEICE Trans. Commun., vol.E85-B, no.10, pp.2090-2099, October 2002.

[7] K. Kawano, K. Kinoshita, K. Murakami: "Multilevel hierarchical mobility management in densely meshed networks", IEICE Trans. Commun., vol.E89-B, no.7, pp.2002 2011, July 2006.

[8] Y. Xu, H.C.J. Lee, V.L.L. Thing: "A local mobility agent selection algorithm for mobile networks", Proc. IEEE Int. Conf. on Commun. ICC, vol.2, pp.1074-1079, Anchorage, Alaska, USA, May 2003.

[9] M.-H. Liu, C.-C. Yang: "A multicast extension to HMIPv6 with efficient MAP selection", Proc. IEEE 62nd Semiannual Vehicular Technology Conf. VTC, vol.2, pp.816-820, Dallas, Texas, USA, September 2005.

[10] T. Kumagai, T. Asaka, T. Takahashi: "Location management using mobile history for hierarchical mobile IPv6 networks", IEICE Trans. Commun., vol.87-B, no.9, pp.2567-2575, September 2004.

[11] X. Hu, J. Song, M. Song: "An adaptive mobility anchor point selection algorithm for hierarchical mobile IPv6", Proc. IEEE Int. Symposium on Commun. and Inf. Technology ISCIT, vol.2, pp.1148-1151, Beijing, China, October 2005.

[12] S. Pack, M. Nam, T. Kwon, Y. Choi: "An adaptive mobility anchor point selection scheme in hierarchical mobile IPv6 networks", Elsevier Computer Commun., vol.29, no.16, pp.3066-3078, October 2006.

[13] S. Pack, T. Kwon, Y. Choi: "A performance comparison of mobility anchor point selection schemes in hierarchical mobile IPv6 networks", Elsevier Computer Networks, vol.51, no.6, pp.1630-1642, April 2007.

[14] V. Devarapalli, R. Wakikawa, A. Petrescu, P. Thubert: "Network mobility (NEMO) basic support protocol", RFC 3963, January 2005.

[15] R. Novak: "Proxy MAP for intra-domain route optimization in hierarchical mobile IP", IEICE Trans. Commun., vol.E89-B, no.2, pp.472-481, February 2006.

[16] L. Subramanian, S. Agarwal, J. Rexford, R. Katz: "Characterizing the Internet hierarchy from multiple vantage points", Proc. IEEE INFOCOM, vol.2, pp.618-627, New York, USA, 2002.

[17] L. Gao: "On inferring autonomous system relationships in the Internet", IEEE/ACM Trans. on Networking, vol.9, no.6, pp.733-745, 2001.

[18] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, kc claffy, G. Riley: "AS relationships: Inference and validation", ACM SIGCOMM Computer Commun. Review, vol.37, no.1, pp.29-40, 2007.

[19] "The CAIDA AS Relationships Dataset 2004–2007." http://www.caida.org/data/active/as-relationships/.

[20] University of Oregon route views project (2004).

Andrej Vilhar is a Ph.D. student at the Department of Communication Systems of the Jozef Stefan Institute in Ljubljana, Slovenia. He received his B.Sc. in electrical engineering with major in telecommunications from the University of Ljubljana in 2004. His research interests include Internet topology modelling, modelling of relationships between autonomous systems (AS) and mobility management, with emphasis on the algorithms for mobility anchor point (MAP) selection in the hierarchical Mobile IP protocol.

Roman Novak works as a researcher at the Department of Communication Systems of the Jozef Stefan Institute in Ljubljana, Slovenia. He is also an assistant professor at the Jozef Stefan International Postgraduate School. He received his M.Sc. and Ph.D. in computer science from the University of Ljubljana in 1995 and 1998, respectively. His Ph.D. research was in the field of routing in communication networks.. He participated in several projects related to the field of design and development of telecommunication systems. His research interests include routing in communication networks, with special emphasis on Mobile IP, information system security and efficient implementation of algorithms.

Gorazd Kandus received B.Sc., M.Sc. and Ph.D. degrees in Electrical Engineering from University of Ljubljana, Slovenia, in 1971, 1974 and 1991, respectively. He is currently the head of the Department of Digital Communications and Networks at Jozef Stefan Institute and Associate Professor at Faculty of Electrical Engineering, Computer Science and Information Technology, University of Maribor. He spent one year at Worchester Polytechnic Institute, Worchester, MA, as a Fulbright Fellow and 5 months as a Visiting Scientist at the University of Karlsruhe, Germany. His main research interests include design and simulation of mobile and wireless communication systems and development of new telecommunication services. He is a member of IEEE and Upsilon Pi Epsilon.