Classifying Users Based on Their Mobility Behaviour in LTE Networks

Bart Sas, Kathleen Spaey, Chris Blondia
iMinds/University of Antwerp
Antwerp, Belgium
Emails: {bart.sas, kathleen.spaey, chris.blondia}@uantwerpen.be

***Abstract*—When, in cellular networks like Long Term Evolution (LTE), there is a dense deployment of cells and/or users move at high velocities, handovers will occur frequently. This will have a severe impact on the Quality-of-Service (QoS) experienced by the users as there will be frequent call drops and a low throughput due to a service interruption time that is relatively long compared to the cell stay time. In order to mitigate these problems, users that experience a so called high mobility should be steered appropriately (e.g., to a cell on which they can be camped for a longer time). The first prerequisite for steering users that experience high mobility is identifying these users and predicting their future behaviour. In this paper, we present an algorithm that identifies users that follow similar trajectories through a cell. To perform this identification, a modified version of the Dynamic Time Warping (DTW) algorithm is used. Results show that the developed algorithm is able to detect users which follow similar trajectories adequately and is also able to distinguish between users that follow different trajectories through a cell.**

Keywords–High Mobility; Dynamic Time Warping; Traffic Steering; Handover; LTE

I. INTRODUCTION

In this paper, we focus on high mobility users in LTE networks, i.e., users that tend to make frequent handovers and for which the average amount of time they stay in a cell (time-of-stay) is low (order of magnitude of 10 seconds). For such users there will be a noticeable impact on user and network performance. This impact may be seen in a reduced QoS experienced by the users with high mobility due to a long data outage period relative to the cell stay time, an increased number of call drops and an increased signalling overhead in the core network due to handover signalling. The velocity of a user and the path it follows through a cell are the key factors that determine whether a user is subject to high mobility. Short time-of-stay can occur in two real-life situations: (1) when cell sizes are so small that even users with a low velocity perform frequent handovers and (2) when users move at a high velocity. In both situations, handovers will occur frequently and the time between entering a cell and leaving it will be small. Situation 1 might occur when there is a dense deployment of small cells, for instance in a shopping street/mall where users move at a low pace (for instance pedestrians). The cell inter-site distances in this case will be rather low (10-30 m) as will be the speed at which the users are travelling (2-3 km/h). Situation 2 might occur in macro cells along a busy highway or high-speed railroad: although the cells themselves are relatively large, the high pace of the users will cause cell stay times to be low. There do not necessarily have to be many cells involved, the aforementioned situations might also occur in isolated cases with only a few cells that have users with high mobility.

In this paper, a Self-Organising Network (SON) function [1] that aims at optimising the handover behaviour of high mobility users is presented, called the High Mobility SON function. The goal of this SON function is to classify users according to their mobility behaviour and, based on this classification, steer them in an appropriate way such that the handover rate is reduced while the QoS is maintained or possibly improved. The focus of this paper is on the design and evaluation of a method to classify users according to the trajectory they follow through a cell. By assuming that users that follow similar trajectories (i.e., similar geographical paths at similar velocities) through a cell will have similar behaviour regarding mobility and handovers, the future behaviour of a user can be predicted based on observations made from past users that had similar trajectories. These predictions can then at a later stage be used to decide which handovers are useful and which are not and to decide what the best destination cell for a handover is such that the amount of handovers can be reduced by avoiding unnecessary and suboptimal handovers.

The remainder of this paper is structured as follows: Section II gives an overview of work that is related to the topic of this paper. Section III describes the general architecture of the SON function that we designed to find users that have high mobility and to steer them such that the negative effects of the frequent handovers are mitigated. The specific component of the SON function that is the focus of this paper, the Trajectory Classifier, will use standardised measurements; these are explained in Section IV. Section V describes the classical DTW algorithm and the modifications that were made to it in order for it to be able to identify users that follow similar trajectory through a cell. Section VI describes the simulation studies that were performed in order to assess the ability of the algorithm to classify users based on their trajectories. Finally, Section VII concludes this paper and lists the future work.

II. RELATED WORK

Dynamic Time Warping is a well-known technique for finding similarities in time-series. It finds the most optimal alignment between two time series and is able to deal with slight variations in time and speeds. It is often used in automatic speech recognition for finding patterns, furthermore it is also used in speaker recognition and signature recognition [2][3]. DTW is more robust than more simple techniques like the Euclidean distance as DTW is able to deal with slight variations in the input like accelerations and decelerations.

 ISBN: 978-1-61208-347-6

Traffic steering is a technique whereby mobile users are intelligently steered towards certain base stations. It can be applied for various reasons like improving QoS, balancing the load among different base stations, satisfying certain capability needs, etc. Traffic steering can be performed between similar cells, between different layers of the same technology [4] or even between different technologies [5][6][7]. In this paper, we will consider steering users between cells of the same technology, namely LTE. The algorithms that are introduced in this paper could however be extended to other cellular technologies as well.

Also in [8][9][10], the problem of high mobility is considered. Fei and Fan [8] describe two position-assisted handover schemes: one that aims to reduce the handover delay and one that aims to reduce the handover frequency and improve the handover success rate. Both schemes do however rely on the geographical locations of the users, which are not always known. The goal of the schemes is also not targeted at users with high mobility, but instead to high velocity users. Papathanasiou et al. [9], and Cheng and Fang [10] are concerned with applying scanning narrow beams to LTE networks that contain fast moving mobiles.

III. General Architecture

The general architecture of the proposed High Mobility SON function is shown in Figure 1. It consists of a number of components, which are described below. The *Trajectory Classifier* is the core of the SON function. It is responsible for classifying users according to the trajectory that they follow through a cell. Users that follow similar trajectories than other past users will be identified by it. This information can then be used to predict the future behaviour of currently active users. This component plays an important role in the SON function and will be the focus of this paper.

The set of trajectories that are available to the Trajectory Classifier to map an active user on is determined by the *Trajectory Identifier*. This component decides which trajectories are distinct and useful enough to be considered for matching users to by the Trajectory Classifier.

The *Mobility Classifier* is responsible for determining the mobility type of the users. The mobility type of a user can for instance be vehicular user, pedestrian, stationary user, etc. Knowing what the mobility type of the users is, is important for knowing how they will behave in the future. Pedestrians might for instance regularly stop or go slower or faster in an unpredictable way while vehicular users will more likely move at a predictable pace, only stopping at intersections and traffic lights.

Finally, the *Traffic Steerer* is responsible for the actual decision of when and to which target cell the user should be handed over. Based on the trajectory on which an active user has been mapped and its mobility type the Traffic Steerer will estimate when certain events like call drops, the throughput or some other QoS Key Performance Indicator (KPI) becoming too low, etc. will occur; and which action should be taken and at what time in order to steer the users as appropriate as possible.

The decisions that are taken by the Traffic Steerer are communicated to the *Handover algorithm*. This algorithm is

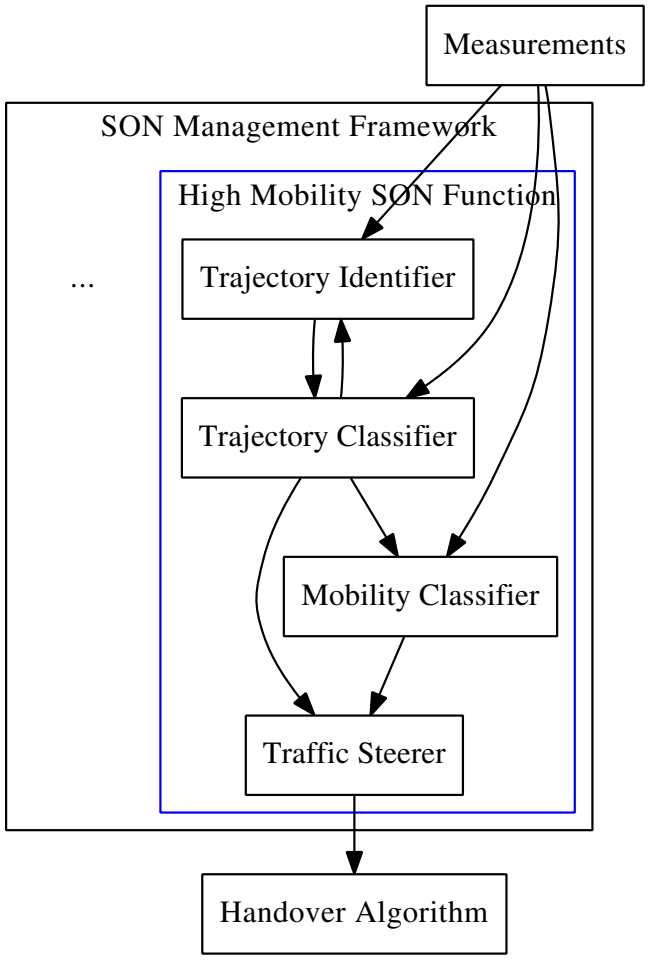

Figure 1. The general architecture of the proposed High Mobility SON function

responsible for performing the actual handovers. Apart from executing the commands coming from the Traffic Steerer, the Handover (HO) algorithm is also is responsible for handing over users for which the SON function did not provide instructions as the SON function will only deal with these users for which it sees an opportunity to optimise the handover behaviour.

Typically, a SON function will be part of a larger SON management framework like for example the SEMAFOUR integrated SON management framework [11]. This framework integrates existing and future SON functions across several radio access technologies and is in charge of functions like policy transformation/supervision and conflict detection/resolution among multiple SON functions.

The High Mobility SON function will be placed at the eNodeB and will operate on a per-cell level. This allows the SON function to be gradually deployed in the network and makes it scalable.

IV. Measurement Reports

As mentioned before, the Trajectory Classifier is the component of the SON function that is responsible for classifying users based on the trajectory they follow through a cell. It performs a key role in the SON function as it provides information that

 ISBN: 978-1-61208-347-6

serves as input for most of the other components. In order to distinguish between users that follow similar trajectories and other users we need some kind of measurements that are made by the User Equipment (UE) and sent to its Serving eNodeB (SeNB), which can then use them. It is important that these measurements are sent sufficiently frequent in order for the Trajectory Classifier to make actual decisions; furthermore they must allow distinguishing between users that travel along different trajectories through a cell. The measurement reports that will be used by our algorithm are the measurement reports a UE sends to its SeNB as part of the LTE handover process. Both these measurement reports as well as the measurement report triggering have been standardised [12]. Active users monitor the Reference Signal Received Power (RSRP) and/or Reference Signal Received Quality (RSRQ) of their SeNB and its surrounding Neighbouring eNodeBs (NeNBs). Whenever a certain condition involving the RSRP/RSRQ holds and sometimes also when it no longer holds, the UE sends a measurement report to its SeNB containing the list of NeNBs for which the condition (still) holds at that time as well as the RSRP/RSRQ measured for the SeNB and the NeNBs in the list. The triggering of measurement reports at the UE is configured by the SeNB. There are a number of possible event types that can be configured, named event A1 through A6. All these events specify an entering and leaving condition. These conditions are equations that involve the RSRP/RSRQ of the SeNB and/or the NeNB; and/or event-specific thresholds. In order to prevent measurement reports from being triggered too often due to small fluctuations in the RSRP/RSRQ measurements, the triggering conditions also specify a hysteresis value that specifies an extra offset that is added to the equation. Furthermore, the conditions also have to hold for a certain amount of time called the Time-to-Trigger (TTT) before the events are fired. The event-specific thresholds, hysteresis and TTT of an event can all be specified for each individual event that is configured at the UE. In this paper, event A4 is exclusively used as it provides information about the RSRPs of the NeNBs in comparison to a fixed threshold. The other events provide information about the RSRPs relative to the SeNB or about the SeNB itself, which is less usable as the information is more volatile or less exhaustive. An A4 event is triggered when the RSRP/RSRQ of a user becomes better than a certain threshold. By configuring a number of these events the SeNB of a UE will at every point in time know the signal levels of its NeNBs that are observed by the UE within an upper and a lower bound, because whenever the observed RSRP/RSRQ of a NeNB changes and crosses one of the thresholds an update is sent to the SeNB. As multiple events are often triggered at the same time, because of sudden changes in the RSRP/RSRQ, events that occur within a short amount of time (< 0.03 s) are aggregated into a single measurement. This results in a time series where every element represents an update of the RSRP/RSRQ level of the NeNBs. Using this information users will be classified based on their trajectory. The rationale behind this approach is that users that pass through the same parts of a cell will measure similar signal strengths from the same surrounding eNodeBs.

In the following section, the algorithm that matches users that follow similar trajectories based on the collected measurements is presented.

V. Dynamic Time Warping

The key to handing over users to the most suited target cell is being able to identify users that follow similar trajectories and to distinguish between users that follow different trajectories.

Measurements that are sent by users can slightly differ due to slight deviations in the trajectory that is followed, slightly different user velocities, time variations in fading, etc. Because of this, it is not just possible to compare the measurements of both the active and reference users by looking for, for instance, the longest common sub-string. In order to make this matching more resilient against slight variations in the measurement data, a modified version of the DTW algorithm is used.

A. Classical Dynamic Time Warping

The DTW algorithm [2] is used in signal processing to find an optimal alignment between two time series (like the measurements coming from the users). Dynamic Time Warping determines the distance between two time series $X = (x_1, \ldots, x_M)$ and $Y = (y_1, \ldots, y_N)$ by determining the so called optimal warping path through the cost matrix $C \in R^{M \times N}$ whose elements $C_{m,n}$ express the distance $c(x_m, y_n)$ between the elements x_m and y_n of the respective series. A warping path through this cost matrix is a series $p = (p_1, \ldots, p_L)$ with $p_l = (m_l, n_l) \in \{1, \ldots, M\} \times \{1, \ldots, N\}$, satisfying the restrictions that $p_1 = (1, 1)$, $p_L = (M, N)$ and $p_l = (m_l, n_l)$ for $l \in \{2, \ldots, L\}$ can only be reached from $p_{l-1} \in \{(m_l - 1, n_l), (m_l, n_l - 1), (m_l - 1, n_l - 1)\}$, i.e., a warping path is a path through the cost matrix that starts at $(1, 1)$ and goes to (M, N) by either increasing the row index, increasing the column index or increasing both at the same time. The cost or distance of a warping path is given by the total cost of the elements along the path:

$$\sum_{l=1}^{L} c(x_{m_l}, y_{n_l}) \tag{1}$$

The optimal warping path is the warping path that has the lowest total cost of all possible warping paths. The optimal warping path can be efficiently determined by constructing an $M \times N$ matrix D in which each element (m, n) contains the minimal total cost to match the prefix of length m of X with the prefix of length n of Y. Each element (m, n) of this matrix (except from the ones on the first row and column) is calculated by adding the cost $c(x_m, y_n)$ to the minimum of $D[m-1, n]$, $D[m, n-1]$ and $D[m-1, n-1]$. The values of the elements of the first row and column are calculated by adding the cost to the value of the elements to the left or above respectively. Pseudocode for the dynamic time warping algorithm is given in Figure 2. In this code, an additional row and column are added to the matrix D. The elements in the first row and column of this matrix are initialised to infinity, except for the element in the upper left corner which is set to 0. By doing this, the elements in the remainder of the matrix can all be treated the same.

B. Modified Dynamic Time Warping

The major shortcoming of the classical DTW algorithm for our purpose is that it matches two time series entirely. This is however not desirable. First, the more recent past of the active user is more interesting as its behaviour in the future

 ISBN: 978-1-61208-347-6

```
1:  D := array[0..M, 0..N]
2:  D[0, 0] := 0
3:  for m := 1 → M do
4:      D[m, 0] := ∞
5:  end for
6:  for n := 1 → N do
7:      D[0, n] := ∞
8:  end for
9:  for m := 1 → M do
10:     for n := 1 → N do
11:         D[m, n] := c(X[m], Y[n]) + min(
12:                D[m − 1, n],
13:                D[m, n − 1],
14:                D[m − 1, n − 1])
15:     end for
16: end for
```

Figure 2. The classical Dynamic Time Warping algorithm.

will be influenced more by this than by its earlier behaviour. Furthermore, it should be possible to match an active user with a reference user without having to match the latest measurement of the active user with the last reference measurement as this usually is when the reference user left the cell and we want to be able to proactively make decisions before the active user also leaves the cell.

In order to do so, the DTW algorithm is adapted as follows: the matrix that is used to calculate the intermediate distances is filled backwards, i.e., the DTW algorithm is applied to the reverse series. By doing this each element of the matrix contains the minimal total cost, when only matching the suffixes of both measurement series up to that point. During construction of the matrix, a current best match is kept and updated each time the value of an element is calculated. By doing this, by the time the upper left corner of the matrix (the element that corresponds to the first elements in both series) is reached the best matching suffixes of the measurement series of the active and reference user has been found.

In order to match a suffix of the measurement series of the active user with any interval of the measurements of the reference user, the end of the measurement series of the active user is shifted along the elements of the measurement series of the reference user. The Modified Dynamic Time Warping (MDTW) algorithm is applied to the entire active series and the sub-series of the reference user starting from the first measurement up to the measurement that is aligned with the end of the measurement series of the active user, as is illustrated in Figure 3. As the MDTW algorithm is able to find matches of sub-series of different lengths, it is important that the length of the warping path is taken into account when determining the optimal warping paths. In order to do this, costs will be represented using tuples (v, w). The component v represents the value of the cost and is a value between 0 and 1. When v has value 0 this means that there is an exact match between two measurements. When v has value 1 this means that two measurements are completely different. The component w represents the importance or weight of the value. The exact meaning of the importance depends on how the cost between measurements is defined, but as a general rule the weight should be higher when the cost is based on more information, for instance a longer warping path. Two operations

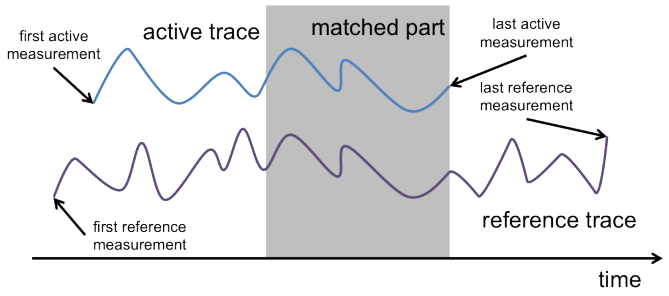

Figure 3. A suffix of the active trace is matched with an interval of the reference trace

are needed by the dynamic time warping algorithm: the addition of costs and the comparison of costs. The addition of two costs is defined as follows:

$$(v_1, w_1) + (v_2, w_2) = \left(\frac{v_1 w_1 + v_2 w_2}{w_1 + w_2}, w_1 + w_2 \right) \quad (2)$$

As can be seen from (2), the value of the sum is equal to the weighed average of both addends while the weight of the sum is equal to the sum of the weights. Two costs are compared by ignoring the weights and just comparing their values:

$$(v_1, w_1) < (v_2, w_2) \iff v_1 < v_2 \quad (3)$$

Due to the way that addition is implemented, tuples will be compared based on the average cost of the elements along the warping path.

Considering every possible suffix of both series has a disadvantage: shorter suffixes will be favoured over longer suffixes as it is more likely to find a short perfect match than it is to find a longer one. In order to mitigate this problem, the initial cost is set to $(1, x)$ with $x \geq 0$. Since after a series of additions, the resulting value of the tuple is the weighted mean of all added values, the importance of this initial cost will become less important as more values are added to it. By using a tuple with non-zero weight as the initial cost, longer matches will be favoured over shorter matches. Note that adding (v, w) to $(1, 0)$, i.e., $x = 0$, will result in (v, w) removing the influence of the initial cost altogether.

Another problem that arises when allowing the algorithm to match every combination of suffixes is that the cost matrix can be traversed mainly horizontally or vertically, as this produces warping paths of the same length as going diagonally. Going mainly vertically or horizontally is however less favourable as this will match shorter portions of either one of the compared traces for the same length of warping path. This is not a problem with the classical DTW algorithm as it matches the entire series. In order to mitigate this problem an additional cost can be added when traversing the cost matrix in the vertical or horizontal directions, but not when going diagonally. This will 'encourage' the algorithm to match longer portions of both series. Like the initial cost, this additional cost will have the form $(1, x)$ with $x \geq 0$. Pseudocode for this is given in (4).

$$\begin{aligned} D[i, j] := c + \min (& D[i - 1, j] + \text{extraCost}, \\ & D[i, j - 1] + \text{extraCost}, \\ & D[i - 1, j - 1]) \end{aligned} \quad (4)$$

 ISBN: 978-1-61208-347-6

```
1:  bestMatch := None
2:  bestDistance := ∞
3:  for start := 1 → m do
4:      distances := array[1..start + 1, 1..n + 1]
5:      for i := 1 → start do
6:          distances[i, n + 1] := ∞
7:      end for
8:      for j := 1 → n do
9:          distances[start + 1, j] := ∞
10:     end for
11:     distances[start + 1, n + 1] := initialCost
12:     for i := start → 1 do
13:         for j := n → 1 do
14:             cost := calculateDistance(
15:                 referenceMeasurements[i],
16:                 activeMeasurements[j])
17:             distances[i, j] := cost + min(
18:                 distances[i + 1, j] + extraCost,
19:                 distances[i, j + 1] + extraCost,
20:                 distances[i + 1, j + 1])
21:             if distances[i, j] < bestDistance then
22:                 bestMatch := (start, i, j)
23:                 bestDistance := distances[i, j]
24:             end if
25:         end for
26:     end for
27: end for
```

Figure 4. The modified Dynamic Time Warping algorithm

The pseudocode for the MDTW algorithm is given in Figure 4. Note from this code that the MDTW algorithm will always return a match, together with the distance of that match.

VI. Evaluation

In order to assess the ability of the MDTW algorithm to identify users that follow similar trajectories and to distinguish between users that follow different trajectories, simulations were performed using a simulator that is based on the OMNeT++ simulation library [13]. In these simulations, a user moves around producing reference traces. Afterwards, the user starts to produce active traces, which are then matched to the reference traces. Each active trace is compared to all reference traces and the reference trace for which the MDTW algorithm finds the match with the lowest 'bestDistance' is considered as the reference trace to which the active trace is matched. The accuracy of this match will be assessed using a performance metric (explained in Section VI-B), which is based on the geographical locations that are visited by the user. These geographical locations are collected during the simulations specifically for the purpose of being able to assess the accuracy of the matches made. In reality these geographical locations will not necessarily be available; therefore it is important that it is possible to specify criteria based on known information that determine whether a match made by the MDTW algorithm can be trusted or not.

A. Simulation Setup

The simulation area is a rectangle measuring 1732 by 2000 metres featuring wrap-around. It contains 16 three-sectored cells with directed antennas, placed at regular distances, as is depicted in Figure 5. Pathloss calculations are performed using the Okumura-Hata model for large urban areas. Furthermore, shadow fading that is both auto-correlated in time and cross correlated with the shadow fading of other antennas is considered [14]. Two different scenarios are considered. The goal of

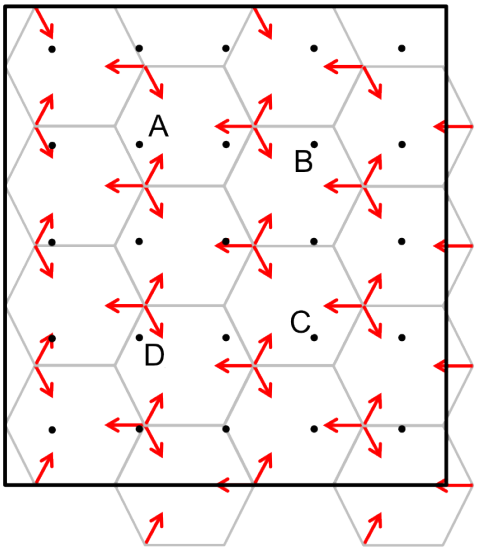

Figure 5. Overview of the simulation area

the first scenario is to assess whether the MDTW algorithm is actually able to find good matches in case it is a certainty that good matches exist. In this scenario a single user follows a rectangular path through the simulation area, passing through the same locations multiple times. The path that is followed by the user is the path between the points A, B, C and D in Figure 5. The simulations consist of two phases: in the first phase reference traces are collected and stored by the different SeNBs the user is connected to. After a certain amount of time (10000 s), the second phase starts. In this phase, the SeNBs start to match the measurements that are generated by the active user with reference traces from the past. These simulations are carried out a number of times. Each time with a different interval from which the user chooses its velocity. The user chooses a different velocity at each corner of its rectangular path. These intervals are $[5 - \frac{i}{2}\ \mathrm{m/s}; 5 + \frac{i}{2}\ \mathrm{m/s}]$ with $i \in \{0, 1, 2, 3, 4, 5\}$. By doing this, also the ability of the MDTW algorithm to deal with slight variations in the velocity of the users can be assessed.

In the second scenario, the user no longer moves along a fixed path. Instead, it moves from point to point (the black points in Figure 5) in a rectangular grid of points that are equally distributed across the simulation area, i.e., so-called Manhattan mobility [15]. Each time a user reaches a point, it randomly chooses to go either left, right or forward with the probability of going forward being twice as large as the probability of going either left or right. It then travels along the chosen direction with a velocity that is chosen from a certain velocity interval. Similar as in the first scenario this interval is different between simulations. Furthermore, the user not just travels between the the grid points, but instead the points it travels between are chosen uniformly within a circle centred around the grid points. The radius r of this circle is either 0 m, 1 m, 5 m or 10 m.

 ISBN: 978-1-61208-347-6

When the radius is 0 m the user travels directly between the grid points.

Nineteen different A4 events were configured. Their thresholds range from -100 dBm to -10 dBm in steps of 5 dB. The hysteresis of all these events is set to 2 dB and the TTT to 480 ms. The 'initialCost' and 'extraCost' parameters of the MDTW algorithm are set to 0.1 and 0.01 respectively.

An overview of the simulation parameters is given in Table I.

Table I. OVERVIEW OF THE SIMULATION PARAMETERS

Parameter	Value
Simulation area	1732×2000 m
Pathloss model	Okumura-Hata (large urban)
User height	1.8 m
Base station height	30 m
Site-to-site distance	500 m
Carrier frequency	2.6 GHz
Antenna model	NGMN [16]
Number of sectors	48
Mean user velocity	5 m/s
Event A4 hysteresis	2 dB
Event A4 time-to-trigger	480 ms
Simulation duration	20000 s
Initial cost (initialCost)	0.1
Extra cost (extraCost)	0.01

B. Performance Metric

In order to assess whether the matches that are made by the MDTW algorithm are accurate, a metric is needed that assigns an accuracy to the matches. The accuracy metric should reflect how well the algorithm is able to identify the part of the trajectory where the reference and active user followed a similar geographical path at a similar velocity through the cell. It is important that this metric takes into account both the part of the trajectory that was matched by the algorithm, but that does not overlap in reality and the part that does overlap in reality but was not matched by the algorithm, as it is important that the MDTW algorithm is able to identify the exact part of the trajectory that overlaps.

As described in Section VI-A, the users move through the simulation area along certain paths. For each match that is made, the portion of the paths of the active and reference users that overlap geographically is determined. In case there is no geographical overlap, the accuracy metric will be set equal to 0. From this overlapping part, the start and end times at which the active user visited the overlapping part are determined. Suppose o_s is the time on which the active user enters the overlapping part and o_e the time on which it leaves it. Similarly, m_s and m_e are the respective begin and end times of the match made by the MDTW algorithm. The accuracy metric is then given by (5).

$$\text{Accuracy} = \frac{\max(0, \min(o_e, m_e) - \max(o_s, m_s))}{\max(o_e, m_e) - \min(o_s, m_s)} \quad (5)$$

This formula will yield 0 when the interval of which the MDTW decided that the users followed the same trajectory and the interval corresponding to the geographical overlapping parts are completely disjoint and 1 when they are exactly the same, i.e., there is a perfect match. An illustration of this is given in Figure 6. In this figure, the accuracy metric is the ratio of the common part of both intervals (the part between m_s and o_e) and the width between the leftmost and rightmost ends of the intervals (the part between o_s and m_e).

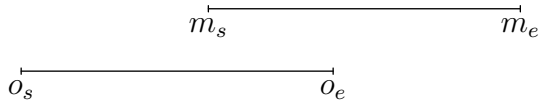

Figure 6. The accuracy metric is calculated as the ratio between the length of the interval $[m_s; o_e]$ and the length of the interval $[o_s; m_e]$

C. Results

Figure 7 shows the results obtained with the first scenario. The x-axis contains the number of active measurements on which the match is based. The y-axis contains the average accuracy of all matches with the corresponding number of measurements on the x-axis. The different curves are for the different velocity intervals. As it can be seen in this figure, in

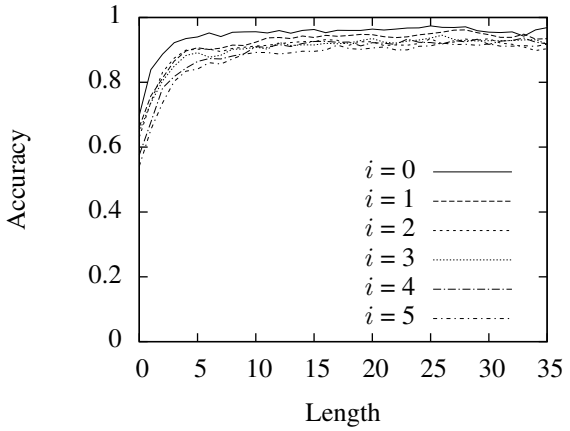

Figure 7. The accuracy metric approaches 1 (perfect match) as the number of measurements on which the match is based increases

case it should be possible to find accurate matches ($i = 0$, velocity interval $[5\ \text{m/s}; 5\ \text{m/s}]$) the accuracy of the matches becomes high (> 0.9) when the matches are based on 5 or more measurements. These matches are not necessarily perfect, which is due to small deviations on when measurements are generated.

When the velocity interval becomes larger, the accuracy of the MDTW algorithm becomes slightly worse. This is to be expected as different velocities will cause measurements to be triggered on different locations, which will make it more difficult for the MDTW algorithm to make accurate matches. The obtained results are, however, still very good and accurate enough to make predictions. Note that when the differences in velocity would become even more pronounced it is no longer desired that good matches are made as users with pronounced different velocities should be treated differently anyway.

The results of scenario 1 show that, in case it is certain that good matches exist (because of how the scenario is constructed), these are found by the algorithm, except if they are based on too few measurements.

Figure 8 and Figure 9 show the results obtained with the second scenario. Note that in this scenario good matches will

 ISBN: 978-1-61208-347-6

not necessarily exist, because it is possible that the active user follows a path that is different from all earlier collected reference paths. On the y-axis of Figure 8 and Figure 9, again, the accuracy of the matches is shown. The x-axis now contains the score that is associated with the matches by the MDTW algorithm (i.e., the value of the lowest 'bestDistance'). In these figures, results for matches based on less than 10 measurements have been left out. Figure 8 shows the results obtained with

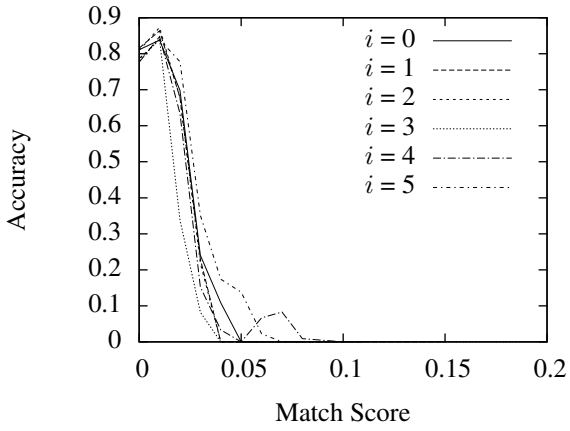

Figure 8. The MDTW algorithm is able to assign low scores to accurate matches and higher scores to inaccurate matches for different velocity intervals

$r = 0$ (i.e., the user travels directly between the grid points of the scenario), for different velocity intervals. This figure shows that there is a strong correspondence between the score that is assigned to a match by the MDTW algorithm and the actual accuracy. Furthermore, there is a clear drop of the match score around 0.03. This drop occurs around the same value for all curves, which means that it is independent from the velocity of the users.

Figure 9 shows the results obtained with $i = 0$ (i.e., the user always travels at a fixed velocity of 5 m/s) but for different radiuses r around the grid points of the Manhattan mobility model. The result is similar as for Figure 8; when the score that is assigned by the MDTW algorithm is low, the accuracy of the match is high and it decreases steeply around a match score of about 0.03. From the results obtained with scenario 2 it is clear that the match score of the MDTW algorithm reflects the accuracy of the matches made: if the accuracy of the match is high (i.e., a match we want to trust) then the match score is low (below 0.03). The match score together with the number of measurements on which the match is based can thus be used as criteria that determine whether a match made by the MDTW algorithm can be trusted or not.

VII. Conclusions and Way Forward

In this paper, an algorithm that identifies users that follow similar trajectories through a cell and distinguishes between users that follow different trajectories was discussed. This algorithm is based on the DTW algorithm that is used in signal processing. The DTW algorithm was adapted such that it is able to find the best match of any suffix of one time series

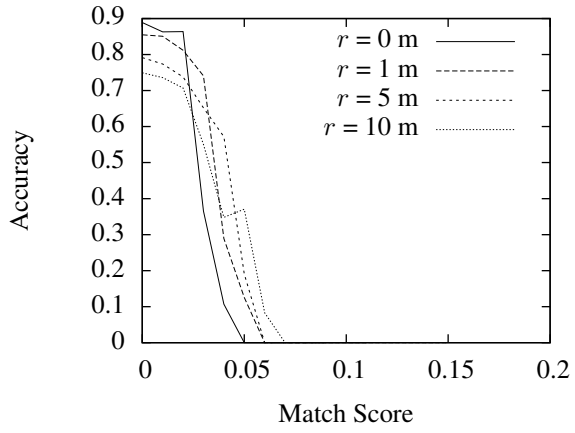

Figure 9. The MDTW algorithm is able to assign low scores to accurate matches and high scores to inaccurate matches for different deviations

(measurements from an active user) with any interval of another series (measurements from a reference user).

The ability of the algorithm to match users that follow similar trajectories was assessed by simulations. First a scenario was considered in which it is certain that good matches should exist and then a scenario was examined in which good matches do not necessarily exist. The results show that the algorithm is actually able to identify users that follow similar trajectories through a cell except if there are too few measurements. The algorithm is also able to deal with small variations in the input. This is important as users will never behave exactly the same: there will always be small differences in velocity and the trajectory that the users follows. From the results, it became also clear that the match score the MDTW algorithm assigns to a match reflects the real accuracy of the match: when the accuracy of the match is high the match score is very low. So, the match score together with the number of measurements on which the match is based can be used as criteria for determining whether a match made by the MDTW algorithm can be trusted or not.

In the future, this algorithm will be used as part of a larger SON function that aims at steering users that have a certain mobility behaviour more appropriately in order to reduce the number of call drops and improve the QoS of the users while reducing the signalling overhead in the core network. The applicability of the SON function in other cellular technologies could also be tested as this paper only focused on LTE.

Acknowledgement

The research leading to these results has received funding from the European Union, Seventh Framework Programme (FP7/2007-2013) under grant agreement n°316384.

References

[1] S. Hämäläinen, H. Sanneck, and C. Sartori, LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency, 1st ed. Wiley Publishing, 2012.

[2] Muller and Meinard, Information Retrieval for Music and Motion. Springer, 2007.

 ISBN: 978-1-61208-347-6

[3] C. Ratanamahatana, J. Lin, D. Gunopulos, E. Keogh, M. Vlachos, and G. Das, "Mining time series data," in Data Mining and Knowledge Discovery Handbook, O. Maimon and L. Rokach, Eds. Springer US, 2010, pp. 1049–1077.

[4] P. Fotiadis et al., "Multi-layer mobility load balancing in a heterogeneous LTE network," in Vehicular Technology Conference (VTC Fall), 2012 IEEE, Sept 2012, pp. 1–5.

[5] S. Frei, W. Fuhrmann, A. Rinkel, and B. Ghita, "Prospects for WLAN in the evolved packet core environment," in New Technologies, Mobility and Security (NTMS), 2012 5th International Conference on, May 2012, pp. 1–5.

[6] M. D. Nisar, V. Pauli, and E. Seidel, "Multi-RAT traffic steering - why, when, and how could it be beneficial?" Nomor white paper, December 2011.

[7] N. Jorgensen, D. Laselva, and J. Wigard, "On the potentials of traffic steering techniques between HSDPA and LTE," in Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd, May 2011, pp. 1–5.

[8] M. Fei and P. Fan, "Position-assisted fast handover schemes for LTE-advanced network under high mobility scenarios," Journal of Modern Transportation, vol. 20, no. 4, 2012, pp. 268–273.

[9] C. Papathanasiou, N. Dimitriou, and L. Tassiulas, "On the applicability of steerable beams in LTE-advanced networks with high user mobility," EURASIP Journal on Wireless Communications and Networking, vol. 2012, no. 1, 2012, p. 234.

[10] M. Cheng and X. Fang, "Location information-assisted opportunistic beamforming in LTE system for high-speed railway," EURASIP Journal on Wireless Communications and Networking, vol. 2012, no. 1, 2012, p. 210.

[11] A. Eisenblätter et al., "Integrated self-management for future radio access networks: Vision and key challenges," in Future Network and Mobile Summit (FutureNetworkSummit), 2013, July 2013, pp. 1–10.

[12] "Evolved universal terrestrial radio access (E-UTRA) and evolved universal terrestrial radio access network (E-UTRAN); overall description; stage 2 (release 11)," 3rd Generation Partnership Project, Tech. Rep. 3GPP TS 36.300 v11.3.0, September 2012.

[13] OMNeT++ Network Simulation Framework. [Online]. Available: http://www.omnetpp.org/

[14] R. Fraile, J. F. Monserrat, J. Gozálvez, and N. Cardona, "Wireless systems mobile radio bi-dimensional large-scale fading modelling with site-to-site cross-correlation," European Transactions on Telecommunications, vol. 19, no. 1, 2008, pp. 101–106.

[15] F. Bai and A. Helmy, Wireless Ad Hoc and Sensor Networks. Springer, October 2006, ch. A Survey of Mobility Modeling and Analysis in Wireless Adhoc Networks.

[16] "Radio access performance evaluation methodology," Next Generation Mobile Networks, Tech. Rep., January 2008.

 ISBN: 978-1-61208-347-6