Institute of Communication Networks and Computer Engineering
University of Stuttgart
Pfaffenwaldring 47, D-70569 Stuttgart, Germany
Phone: ++49-711-685-68026, Fax: ++49-711-685-67983
Email: mail@ikr.uni-stuttgart.de, http://www.ikr.uni-stuttgart.de

A realistic reference simulation environment for typical urban areas

Thomas Jansen
Institute for Communications Technology
Technische Universität Braunschweig
Braunschweig, Germany
Email: jansen@ifn.ing.tu-bs.de

Ulrich Tuerke
atesio GmbH
Berlin, Germany
Email:tuerke@atesio.de

Christian M. Mueller and Thomas Werthmann
Institute of Communication Networks
and Computer Engineering
University of Stuttgart, Germany
Email: [firstname.lastname]@ikr.uni-stuttgart.de

***Abstract*—The increasing complexity in nowadays networks enforces the need for automation and a better understanding of radio network modeling. To ensure comparability of simulation results we propose a common simulation environment for the IC 1004. The aim of this paper is to actuate discussions on the desirable content of a common simulation environment.**

I. Introduction

The demand for mobile broadband access has significantly increased in the last years. On the one hand smartphones and the huge amount of applications that are developed by companies and private people introduce new services to mobile radio networks. On the other hand mobile network operators launch new network layers like femto-cells to encounter the increasing data traffic in their networks. The outcome of this is an increasing complexity in nowadays networks which enforces the need for automation and the development of optimization algorithms. Thus a better understanding of the radio networks and its modeling is required.

One of the challenges for scientific research in radio network simulations is to properly model the important parts of a mobile communication system. This is both time consuming and error-prone. Furthermore it would be beneficial for comparability of the simulation results if a common simulation environment would be used by different research groups. This is of course even more important if the simulation outcomes are presented to larger communities as the IC 1004.

In this paper we propose a simulation environment for the IC 1004 that will be publicly available. The idea is to use common simulation scenarios in the future that serve the needs of the individual partners. In this paper we describe the current status and our ideas for the scenario layout, the propagation model, the radio network design, the user mobility and traffic model. The aim of this paper is to actuate discussions on the desirable content of a simulation environment for the IC 1004.

II. Scenario layout

The proposed reference simulation scenario is based on an artificial macro-cell network in the northern German city of Hanover. The scenario comprises an area of 5 km x 7 km. Within this area a total number of 81 sectors belonging to 26 individual base stations are located. The positions of the base stations are defined taking into account the EMF database of the German regulatory authority [1] which provides the positions of all radio plants in Germany and the building height in the area of interest for a base station to select a high building if possible. The density of the base stations is aligned with the network density of todays UMTS networks. For each sector the following information is provided:

- Relative x-y-coordinates to a point of origin in meters
- Height in meters
- Azimuth in degree
- Tilt in degree
- Antenna type

The point of origin will be given in Gauß-Krüger 4 coordinates as well to support geodetic visualization. Figure 1 shows the layout of the network. The majority of the base stations has 3 sectors. However 3 base stations show 2, 4 and 6 sectors respectively. The scenario covers dense urban and urban areas as well as some open spaces with vegetation. The network is designed for a transmission frequency of 2 GHz.

It is planned to extend the scenario area to encounter corner effects introduced by the missing interference from base stations outside the current scenario. We did not decided on the final scenario area so far. However the modeling for the outer area would be less detailed and the relevant area for the analysis of the system behavior will remain as described above.

III. Wireless propagation model

As a deterministic channel model we use a 3D ray-optical model that has initially been developed for car-to-car communication purposes. The model (described in [2]) has been extended by a diffraction model for the development of this reference scenario. The following types of rays are considered in the model.

- Direct path
- Specular reflections up to n-th order (2nd order in this case)
- Diffuse scattering
- Diffraction

The direct path between the transmitter and receiver is identified by a line-of-sight check using 3D polygons of the buildings of Hanover. The specular reflections are calculated

978-1-4673-4455-5/12/$31.00 ©2012 IEEE

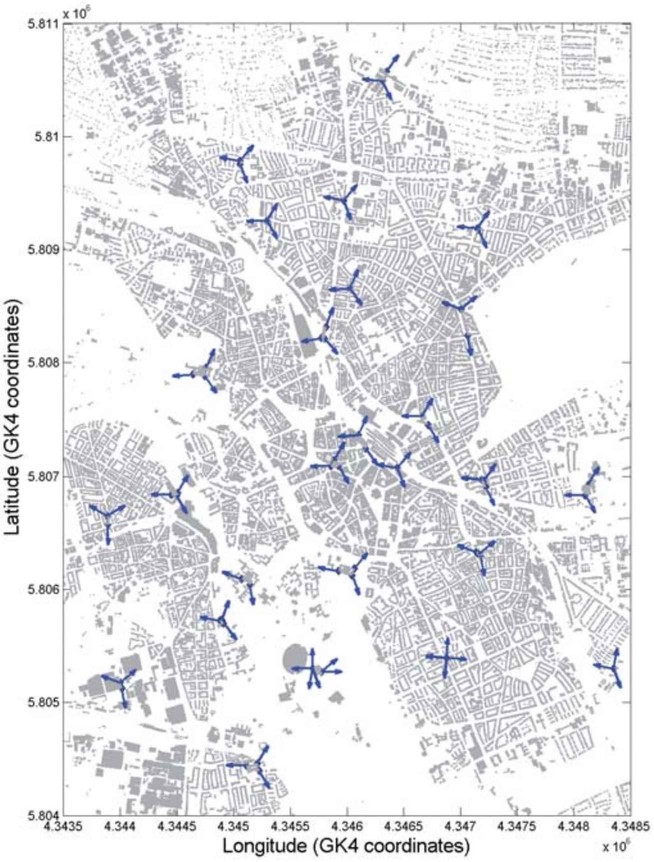

Figure 1: The network layout

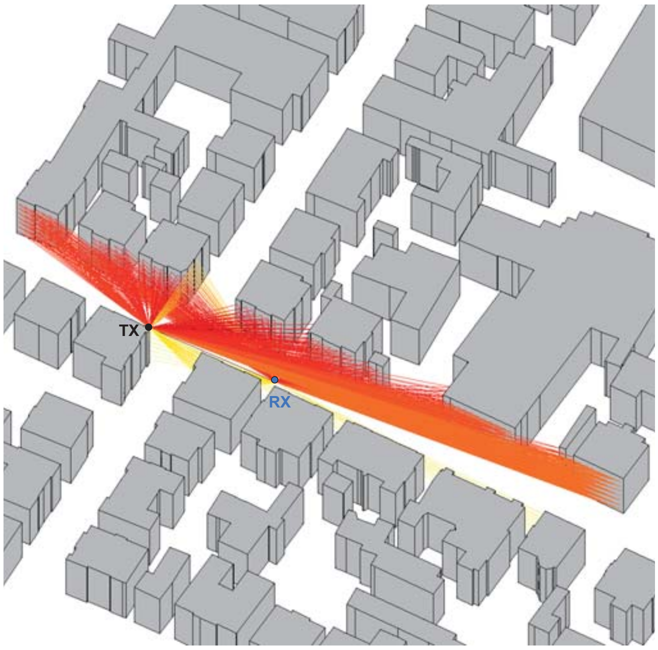

Figure 2: 3D-ray-optical model

based on the image method [3]. Due to the high computation time only 2nd order reflections are considered. On surfaces seen by both the transmitter and receiver diffuse scattering is taken into account. Diffraction is considered using the knife-edge model. Figure 2 shows an example of the identified rays between the transmitter (TX) on a rooftop of a building and receiver (RX) located on the street level.

IV. Radio network design

A realistic radio network design shall be used in the simulation environment. In absence of publicly available real-world network configuration data, a realistic network configuration has been developed. For this, a state-of-the-art planning process, comprising the three components grid design, site design, and cell configuration has been applied.

In *grid design,* proper site locations are selected. Taking the availability of locations (and in real-world planning scenarios monetary cost) into consideration, the high-level goal of grid design is to select locations such that coverage and capacity requirements can be satisfied at minimal cost. Additional information, e.g. about the potential for future extensions may be taken into account. For the particular (small) exercise considered here, the grid design has been carried out manually, taking local knowledge of the involved persons into account.

During *site design,* sectorization, the positioning, height as well as orientation of the antennas are decided on. For the Hanover scenario, this has been carried out as semi-automatic process. Sectorization, antenna locations and antenna heights are configured manually, while the antenna orientation, i.e. mechnical tilt and azimuth angles, are optimized by means of an automated process. The objective of this process is to minimize the interference coupling among cells, while constraining the minimal degree of coverage.

During *cell configuration* several remaining cell parameters, such as transmit power and handover configuration, are determined. Those parameters can typically be set through the OSS. The simulation environment presented in this paper, focuses on the realistic modeling of propagation conditions and leaves the modeling of most of these parameters to the scenario users. In consequence, the only cell configuration parameter contained in the data set is the electrical antenna tilt. This has been predefined during site design.

V. Users and user mobility model

In the current version the reference scenario provides data for two kinds of users - immobile outdoor and indoor users and mobile users (cars) traveling on the streets. For every user the following information is provided:

- User ID
- Time step in seconds
- Relative x-y-coordinates to a point of origin in meters
- Pathloss to the N strongest sectors (e.g. N = 20)
- Height in meters
- Data traffic

For the mobile users the information is updated if any changes appear as long as the user does not leave the scenario area. For the immobile users only the data traffic changes are updated. The amount of pathloss data can be selected in discrete steps based on the individual simulation capacities.

The mobility is introduced to the reference scenario using the open source microscopic road traffic simulator SUMO (Simulation of Urban MObility) [4] [5]. SUMO supplies space-continuous and time-discrete vehicle movements with adjustable sub-second vehicle-based outputs, multi-lane streets with lane changing, right-of-way rules, tunable traffic lights and acceleration and braking of the vehicles. It is optimized for fast execution speed (up to 100.000 vehicle updates/s on a 1GHz machine) and manages several 10.000 edges (streets). The user density in the network has been verified using a traffic density map showing the vehicle traffic within 24 hours from the city authority of Hanover.

The road network is based on OpenStreetMap. The traffic signal timing of the traffic lights has been tuned by hand for every driving direction to deal with the vehicle traffic on the streets. The turning ratios at the road intersections have to be defined for every junction in the road network. We use the turning ratios from the MOMENTUM mobility model specified in [6] for the different road intersection types. In the current proposal the user positions are updated every 100 ms.

We plan to extend the user mobility by a passenger and bicycle model that is currently in development. These users will travel alongside the streets and in the open areas of the scenario. Furthermore we intend to introduce indoor user mobility at a later stage.

VI. Data traffic model

We provide a model for the data traffic in the downlink of a cellular network. Our traffic model comprises data traffic only, voice traffic is not included in the model (there is a profound base of voice traffic models available in literature, we e. g.refer to [7]). We base our model on recently published measurements of cellular data traffic and user behavior [8], [9]. The current model consists of a single user type only.

Our model consists of two parts, which we describe in detail in the following subsections. The *user activity model* describes the macroscopic aspects, i. e., when a user becomes active and how much data a user transfers. The *traffic characteristics model* specifies the microscopic aspects, i. e., how does the traffic look like on packet level. For the traffic characteristics model, we envisage to provide models with different level-of-detail to suit the requirements of different kinds of evaluations. As an example, one might want to use a rather coarse-grained model when investigating possible basestation activation/suspend modes in a diurnal load cycle, whereas one might want to use a more fine-grained model when investigating the bearer idle time thresholds.

A. User activity model

1) Model blueprint: The user activity model specifies which user sends data at what time and what the corresponding data volume is. This trace is generated based on a distribution

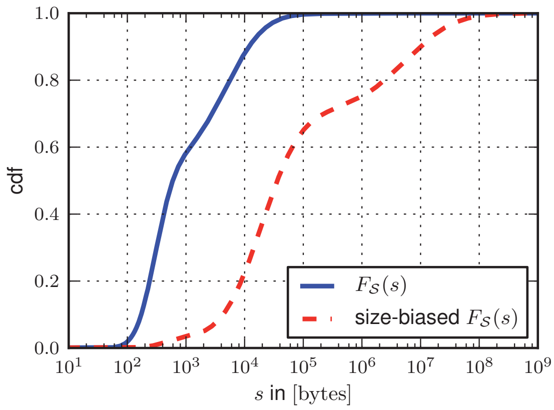

Figure 3: Object size CDF

of the object size, a distribution of the number of objects per day and user and a diurnal cell load profile. Object size distributions of Internet data traffic and diurnal load profiles can be found in literature and will be further detailed below. The number of objects per user and day in a cellular network is not known, but some information is available on the overall data volume sent per user and day. We thus chose a distribution of the number of objects per user and day such that the overall data volume matches the measured data. We further make the following assumptions:

1) The number of objects sent per user and day is independent and identically distributed (i.i.d.) and follows a lognormal distribution.
2) The objects' sizes are i.i.d.
3) The time of day the transmission of an object starts is independent from its size and other objects.

If we consider a total of U users, every user transmits n_i objects per day, with $n_i \in \mathcal{N}$ and $i \in [1, U]$. $\mathcal{N}$ has the distribution function $F_{\mathcal{N}}(x)$. We denote an object's size by $s_{i,j} \in \mathcal{S}$, where j is the index of the object sent by user i. The object size has the distribution function $F_{\mathcal{S}}(x)$. . The data volume generated by user i over one day, $v_i \in \mathcal{V}$, is then

$$v_i = \sum_{j=1}^{n_i} s_{i,j} \tag{1}$$

From this sum, we get a distribution function $\hat{F}_{\mathcal{V}}(x)$ of the daily data volume per user. Because $F_{\mathcal{S}}(x)$ and $F_{\mathcal{V}}(x)$ are known, we have to construct a function $F_{\mathcal{N}}(x)$ such that $\hat{F}_{\mathcal{V}}(x) \approx F_{\mathcal{V}}(x)$.

Finally, for every object $s_{i,j}$, we draw a random transmission time $t(s_{i,j})$ from the distribution function $F_{\mathcal{T}}(x)$, which models the diurnal network load profile.

2) Input data: For the object size distribution, we only consider data being sent over HTTP and HTTPS, since most applications on mobile devices use HTTP and HTTPS for data transfer [8], [10]. This comprises web surfing using a browser, so-called smartphone apps, and application markets. Please note that this also includes mobile video applications such as YouTube. In total, HTTP and HTTPS based applications

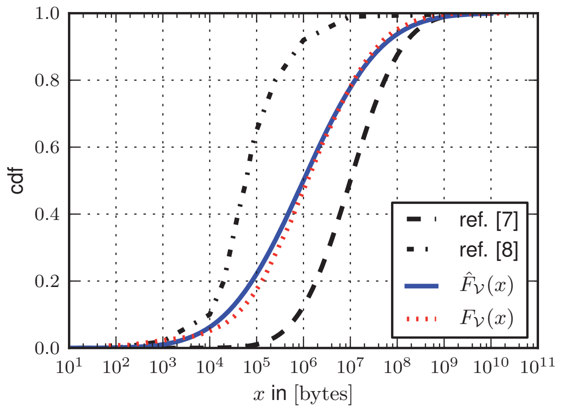

Figure 4: Daily per user volume

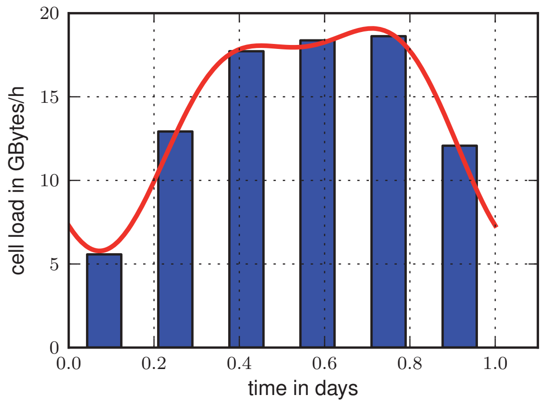

Figure 5: Diurnal load profile

account for more than 80% of the traffic volume on mobile devices [8], [10]. We therefore restrict our model to objects sent over HTTP(S). The term object here refers to application layer data units such as websites, videos etc. More specifically, an object is an HTTP(S) response message.

The distribution of HTTP object sizes is heavy-tailed and only slightly differs between fixed and wireless networks [10], [11]. In our model, we use the HTTP response size distribution published in [11], which provides a representation of the empirical object size distribution function as a mixture of three lognormal distribution functions. Figure 3 shows the cumulative distribution function (cdf) of the object size. It also shows the size-biased object size distribution function, which illustrates the amount of data a certain object size contributes to the overall traffic volume. While more than 80% of the objects have less than 10^4 bytes, they only account for about 20% of the traffic volume.

The authors of [9] and [8] published data on the data volume generated per user and day. [9] analyzed a one week trace from 2007 of all data traffic in a life nation-wide cellular network. [8] analyzed more recent data collected from 33 Android smartphone users over a period of several months. We extracted the data from both papers and reproduced the corresponding cumulative distribution functions in Figure 4. There is a difference of two orders of magnitude between the measurements of [9] and [8]. Reasons for this large discrepancy lie in the differences of user populations, devices and pricing schemes of the two studies. The sample in [9] contains many low volume data users, whereas [8] only contains a small set of moderate to high volume smartphone users. We design our reference scenario to lie in between these two cases, shown as the solid curve in Figure 4.

Given the distributions $F_{\mathcal{S}}(x)$ and $F_{\mathcal{V}}(x)$, we did experiments with different distribution functions to find a function $F_{\mathcal{N}}(x)$ such that the distribution of v_i matches $F_{\mathcal{V}}(x)$. For the input data described here, a lognormal distribution with parameters $\mu = \ln(250)$ and $\sigma = 2.5$ provides a reasonable fit. Here, μ and σ denote the mean and the standard deviation of the associated normal distribution. Figure 4 depicts $F_{\mathcal{V}}(x)$ as well as the distribution function $\hat{F}_{\mathcal{V}}(x)$, which is the result of the sum over all objects.

The authors of [9] also provide data about the diurnal and weekly load cycles of the network. We do not model the weekly cycle, but use the data on the diurnal load distribution to produce a diurnal load profile for our traffic traces. As the authors only provide six sampling points, we fit a function to the given values. We assume that the same load cycle repeats every day and therefore use a linear combination of sine and cosine functions. Although this fitting function is a bit more complex than a piecewise linear fit, with this function we can easily extend the load profile to multiple days without getting unwanted discontinuities. The function

$$\begin{aligned} r(t) = 56.87 - 21.37\cos(2\pi t) - 12.27\sin(2\pi t) \\ - 6.296\cos(4\pi t) - 6.692\sin(4\pi t) \end{aligned} \quad (2)$$

models the diurnal load profile. Here, r is the aggregated network load in gigabytes per hour and t is the time in days. Figure 5 display the diurnal load profile from [9] together with the fitted function $r(t)$. Integration of this function and normalization to a probability sum equal to one results in the following cdf:

$$\begin{aligned} F_{\mathcal{T}}(t) = -0.04370 + t + 0.03434\cos(2\pi t) \\ - 0.05981\sin(2\pi t) \\ + 0.009365\cos(4\pi t) \\ - 0.008810\sin(4\pi t) \end{aligned} \quad (3)$$

For each object which is transmitted in the network, we draw a random number from $F_{\mathcal{T}}(t)$ to determine the time the object arrives at the base station. Under the assumption that every object is immediately transmitted, the resulting load profile corresponds to the one of [9].

B. *Traffic characteristics model*

To be able to provide a simple trace file, we assume that object arrivals are independent of object transmissions, i. e., there is no feedback loop. This means that objects can accumulate in a user's buffer, in case a user's throughput is insufficient for the offered load. Care has to be taken to avoid

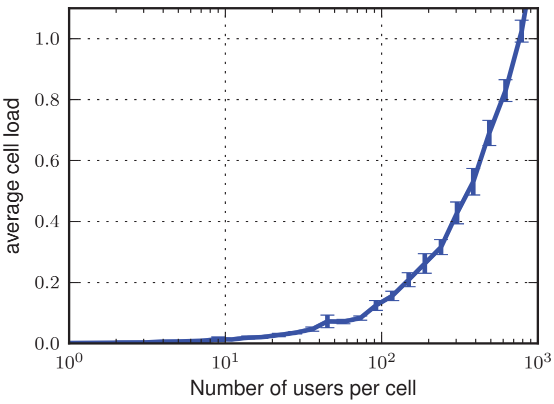

Figure 6: Average busy time cell load

configurations which cause a system to overload, as this might lead to instationary simulation. To suit the requirements of different simulation studies, we envisage to provide two traffic characteristics models with different level-of-detail:

In the *bulk transfer model*, objects arrive at the basestation as a whole and we assume the basestation immediately starts transsmitting. The transmission duration depends on the user's channel quality, the scheduling discipline, the amount of radio resources allocated to the user and maybe other system properties that are specific to the simulation.

The *packet-level model* includes a detailed TCP model and can thus adapt to a user's throughput in the simulation. TCP is a sliding window protocol, where the window size is continuously adapted to the properties of the communication channel. This adaptive behavior leads to a close interaction between physical and MAC layer mechanisms and the transport protocol, which has to be taken into account when studies are conducted on packet-level. Because TCP consists of a set of algorithms to control the behavior of sender and receiver, it's behavior is complex and it is difficult to model it in an abstract way. Therefore, we recommend to use a real world TCP implementations in the simulation. One way to integrate the TCP/IP stack of the Linux kernel is to use the *Network Simulation Cradle (NSC)* [12]. Another way is the use of virtual machines as described in [13].

For the results of different studies to be comparable when using a real world TCP/IP stack, we need to agree on the following parameters in addition to the parameters of the bulk transfer model: maximum transfer unit size, one-way-delay from server to client and from client to server, uplink throughput per user, size of the sender's transmit buffer, size of the receiver's input buffer, length of the drop-tail queue at the basestation, packet drop probabilities, TCP flavor and kernel version. The reference configuration together with reference results will be provided together with the final traces.

C. Illustration

For illustration, we provide a few results on cell load and user activity in a sample scenario. We considered a single 5 MHz LTE cell with a variable number of users. We used the Shannon capacity formula as PHY layer abstraction (with SINR clipped at 22 dB) and assumed that all objects are transmitted sequentially and none of the objects gets dropped (this corresponds to a rate-fair resource allocation). A user's SINR is assumed constant over the whole duration of an object transfer. The distribution of the SINR corresponds to the distribution for the whole reference scenario, assuming that every cell always produces interference.

Figure 6 depicts the average cell load for this setup for the busy time (from 0.6 to 0.8 in Figure 5). The activity factor of a single user in this setup is 0.11%. This means that only a small subset of the users in the cell is active at the same time. Up to around 100 users, the cell is only slightly loaded. For more users, the average cell load increases quickly.

VII. Data format

We are going to provide trace files which include all data described by this model. This section gives an example for the data format we plan to use. Although the data format is still preliminary, it allows you to easily apprehend which information is stored in the trace files.

We will provide 4 separate trace files for 4 types of information, i.e. user positions, channel quality, data traffic and user state. This allows to read the files in different parts of a modularized simulator. All files follow a common format:

We chose to repesent the data in an ASCII format (CSV with digits). This is platform independent and allows easy debugging, because it is human readable. In addition, it is possible to handle these files with standard linux command line tools. To reduce the disc footprint, the files can be zipped.

Each line contains a single information update. The lines are sorted by time, which allows to read the files sequentially. To allow to reference the same users from all files, we use a zero-based integer ID to represent each user. (It is not defined whether the IDs are consecutive.) Lines are delimited by linefeed (ASCII 0x0A), fields by a semicolon (";").

The first two columns of all files contain the time in seconds since simulation start and the user ID. This data format introduces a certain overhead, as time and user IDs follow a repeating pattern. Nevertheless, each line is self-contained, which allows to easily filter out a set of users or a slice in time.

Positions, channel quality values and the user state are sampled with a frequency which can be specified upon generation of the trace file. In addition, it can be specified whether a line in those files is omitted when the respective information didn't change since the last update (e.g. the user is nomadic or waiting in front of a traffic light). Omitting these lines reduces the amount of data and is well suited for event-driven simulations. Repeating this lines is better suited for tools which read the whole file as a matrix.

A. Position data

The position data file describes the coordinates of the users. The file has 5 colums. The first two contain time and user ID as described above. The following three columns define the user's

x, y, and z coordinates in meters (rounded to centimeters). The reference for the coordinate system is the lower left corner of the scenario ($x = 0$, $y = 0$). The zero level in z direction is defined arbitrarily but matches that of the base station height definition. The following lines give an example of the position file:

```
0;   0; 256.13;  49.73; 1.5
0;   1;  24.47; 319.62; 1.5
0;   2; 140.57; 161.00; 1.5
...
0.1; 0; 256.27;  50.01; 1.5
0.1; 1;  24.47; 319.62; 1.5
0.1; 2; 140.43; 161.33; 1.5
...
```

The sampling frequency of the positions is a parameter which can be specified upon generation of the trace file. In addition, it can be specified whether a line is omitted when the respective user didn't move since the last position update (e.g. he is nomadic or waiting in front of a traffic light).

B. Channel data

The channel data file describes the channel quality between mobiles and sectors, for the serving link as well as for the interfering links. To reduce the file size, weak links are omitted. The accuracy of the data and the number n of strongest links which are written to the file can be configured.

The file has $2+2n$ columns. The first two columns contain time and user ID as described above. The following $2n$ columns contain n pairs of sector IDs (zero-based integer) and pathloss in dB (positive value). The following lines give an example of the channel data file.

```
0; 0; 13; 66.81; 12; 60.74; 76; 62.29
0; 1; 42; 69.67; 16; 63.02; 35; 64.05
0; 2; 74; 67.56; 17; 71.10; 78; 70.78
...
```

C. Traffic data

The traffic data file describes the arrival of downlink objects at the sender (server or base station). The file has 3 columns. The first two contain time and user ID as described above. The following column contains the object size in bytes. In general the arrival time of these objects is not slotted, but in the trace file the times are rounded to milliseconds:

```
0.009; 3; 22801
0.018; 0; 13868
0.054; 4; 2919
...
```

D. User state

Users can become inactive when they leave the scenario. We encode this in a special file because

- We want to be able to omit position updates for users who do not move, so we cannot encode the inactive state as not updating the positions.
- Our data traffic file does only describe object arrivals, so a user without new arrivals can still transmit previous objects. Therefore we cannot encode the inactive state as not having new objects.

The user state file has 3 columns. The first two contain time and user ID as described above. The following column "1" when the user is active in the respective slot, otherwise "0". Repeating values of the same state can be omitted as described above. The following excerpt gives an example of the user state file:

```
0;   0; 1
0;   1; 1
...
0.1; 0; 1
0.1; 1; 0
...
```

Running data transmissons of users who become inactive are paused (freezed) until the respective users become active again. Transmissions starting for inactive users are ignored, so that users entering the scenario have the same probability of having an active transmission than users leaving the scenario.

References

[1] *EMF database*, German Federal Network Angency Std. [Online]. Available: http://emf2.bundesnetzagentur.de/karte.html

[2] J. Nuckelt, M. Schack, and T. Kürner, "Deterministic and Stochastic Channel Models Implemented in a Physical Layer Simulator for Car-to-X Communications," *Kleinheubacher Berichte 2011 – Advances in Radio Science*, 2011.

[3] J. McKown and R. Hamilton Jr., "Ray tracing as a design tool for radio networks," *Network, IEEE*, vol. 5, pp. 27–30, 1991.

[4] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, "Sumo (simulation of urban mobility); an open-source traffic simulation," in *4th Middle East Symposium on Simulation and Modelling*, Sharjah, United Arab Emirates, September 2002.

[5] *Simulation of Urban MObility*, Institute of Transportation Systems at the German Aerospace Center Std., microscopic road traffic simulation package. [Online]. Available: http://sumo.sourceforge.net

[6] L. Ferreira, L. M. Correia, D. Xavier, I. Vasconcelos, and E. R. Fledderus, "Final report on traffic estimation and services characterisation," IST-2000-28088 MOMENTUM, Tech. Rep. D1.4, 2003.

[7] P. Orlik and S. Rappaport, "A model for teletraffic performance and channel holding time characterization in wireless cellular communication with general session and dwell time distributions," *Selected Areas in Communications, IEEE Journal on DOI - 10.1109/49.700913*, vol. 16, no. 5, pp. 788–803, 1998.

[8] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin, "A first look at traffic on smartphones," in *Internet Measurements Conference (IMC)*, Melbourne, Australia, November 2010.

[9] U. Paul, A. Subramanian, M. Buddhikot, and S. Das, "Understanding traffic dynamics in cellular data networks," in *INFOCOM, 2011 Proceedings IEEE DOI - 10.1109/INFCOM.2011.5935313*, 2011, pp. 882–890.

[10] G. Maier, F. Schneider, and A. Feldmann, "A first look at mobile handheld device traffic," in *11th International Conference on Passive and Active Network Measurement*. Springer, 2010.

[11] F. Hernandez-Campos, J. Marron, G. Samorodnitsky, and F. Smith, "Variable heavy tails in internet traffic," *Performance Evaluation*, vol. 58, no. 2-3, Nov. 2004.

[12] S. Jansen and A. McGregor, "Simulation with real world network stacks," in *Simulation Conference, 2005 Proceedings of the Winter*, 2005.

[13] T. Werthmann, M. Kaschub, C. Blankenhorn, and C. M. Mueller, "Approaches for evaluating the application performance of future mobile networks," IC 1004, TD(11)01038, June 2011.