Combining Mobility Models with Arrival Processes

Jan Kriege
TU Dortmund, Germany
jan.kriege@tu-dortmund.de

Author prepared version of a paper published in

Measurement, Modelling and Evaluation of Dependable Computer and Communication Systems

Volume 9629 of the series Lecture Notes in Computer Science, 2016

The original publication is available at www.springerlink.com

http://dx.doi.org/10.1007/978-3-319-31559-1_10

Combining Mobility Models with Arrival Processes

Jan Kriege

TU Dortmund, Germany

jan.kriege@tu-dortmund.de

Abstract

The realistic modeling of mobile networks makes it necessary to find adequate models to mimic the movement of mobile nodes. In the past various such mobility models have been proposed, that either create synthetic movement patterns or are based on real-world observations. These models usually assume a constant number of mobility nodes for the simulation. Although in real-world scenarios new nodes will arrive and other nodes will leave the simulation area, only little attention has been paid to modeling these arrivals and departures of nodes.

In this paper we present an approach to easily extend mobility models to support the generation of arrivals and departures. For three standard mobility models the effect of this extension on the performance measures of a simple mobile network is shown.

Keywords: Mobility models, Scenario generation, Arrival Processes, ARTA Processes

1 Introduction

The adequate and realistic modeling of the traffic load is a crucial step when building stochastic models of computer and communication networks. For wired networks it is well known that packet interarrival times are correlated and that neglecting this correlation might have significant impact on performance measures [16]. With the increased availability of mobile devices performance evaluation of wireless networks has become more important. For a realistic load modeling the user mobility has to be considered additionally in wireless networks. To mimic movement patterns of users (or mobile nodes) in a wireless scenario mobility models are used. Mobility models basically consist of some rules that define how the nodes of a wireless network move. On an abstract level the mobility of a node consists of a spatial component, that defines to what destination a node is moving, and a temporal component, that defines when and at what speed the node is moving. It is well known, that unrealistic mobility models may lead to wrong assumptions on the performance of the system that is analyzed [21].

In the past various mobility models have been proposed and the overview we can give here is by far not complete. These models can be divided according to different criteria [3]. The most distinctive criteria are probably whether the mobile model synthetically creates movements or is based on real-world observations and whether it treats movements of single nodes or groups of nodes.

An overview of models creating synthetic movements can be found in [10]. Those models are easy to implement, can be easily integrated into simulation models and do not need additional information and are therefore widely used. Classical examples are the Random Waypoint, Random Direction and Random Walk models, where a direction or destination and the speed of a node is randomly determined. While these models are memoryless, i.e. they do not use information from the past to determine the next destination or speed, the Gauss-Markov model [17] chooses these values depending on previous values. The QoS-RWP model [19] is based on the Random Waypoint model, but divides the nodes into two classes. One class moves according to the Random Waypoint model, while the second class is stationary unless their quality of service drops beyond a given threshold. Aside from these very general models, approaches like the City Section model [10], that aims at representing the topology of streets, exist for special applications.

More recent approaches use real world-observations as basis for the mobility models, because the synthetically generated movements might differ from real patterns and require an idealized, free simulation area [22], which could lead to wrong assumptions for performance measures [18].

For example the model from [14] considers buildings and obstacles by using Voronoi-diagrams. [22] constructs a list of trips from real data that consist of visited access points and in combination with a map realistic routes can be obtained for the mobile nodes. In [15] a matrix of transition probabilities for different locations and the distributions for pause times and speed values are estimated from real-world observations.

A different approach was chosen in [12] where a network of queues corresponding to the different access points was used to model the wireless network on a more abstract level.

In contrast, little attention has been paid at modeling of arrivals and departures of users in wireless scenarios. Usually the number of mobile nodes is set to a fixed value at the beginning of a simulation and does not change during simulation, because no nodes leave or enter the area. For short simulations these assumptions might be justified, though as pointed out in [21] the common short simulation times are not sufficient for modeling the mobility in WiFi networks. However, for longer simulation runs it is very likely that the number of nodes varies. Typical scenarios are an airport terminal or a shopping mall where there should be a large throughput of mobile nodes. These considerations clearly motivate that mobility models should also be able to account for a varying number of nodes. However, this has hardly been treated in the literature, yet. [4] used queueing networks to model a wireless network and considered external arrival rates. In [2] the arrivals of participants of a conference were analyzed and modeled as a Markov-Modulated-Poisson Process (MMPP) but not combined with a mobility model. As the authors state, a MMPP is sufficient for modeling the arrivals at a conference with phases of many arrivals (start of a session) and few arrivals (during a session), but is probably not adequate for other scenarios with more complicated arrival patterns.

In this paper we propose a general approach to enhance mobility models to account for arrivals and departures of nodes resulting in a varying number of nodes during the simulation of the model. Since the arrivals and departures are likely to exhibit correlation we propose a combination of mobility models with stochastic processes. The effect of incorporating these arrival patterns into the mobility models is systematically assessed by measuring the traffic load generated by the models in a wireless network scenario.

The paper is structured as follows. In Sect. 2 we briefly introduce the mobility models and stochastic processes used in our experimental analysis. Sect. 3 describes our approach to combine mobility models with arrivals and departures. In Sect. 4

we experimentally evaluate the effect of the added arrival patterns. The paper ends with the conclusions in Sect. 5.

2 Background and Notations

As already mentioned in Sect. 1 there exist various mobility models for different applications and requirements, though they usually assume a fixed number of nodes. In the following we will introduce three basic mobility models in more detail that are later used for our experiments. Additionally we present the theoretical background on ARTA processes that we will use to generate arrivals and departures.

2.1 Random Walk Mobility Model

In the Random Walk mobility model nodes change their location by randomly choosing the direction and the speed to travel. The model is parametrized by the bounds for the speed ($[v_{min}, v_{max}]$) and either a time interval t or a distance d. Each movement then either takes t time units or covers the distance d. The direction is chosen from $[0, 2\pi]$. At the end of a movement a new speed and direction are randomly determined. Nodes that reach the border of the simulation area are reflected.

The Random Walk model is memoryless, since no information about past locations or speeds is used when determining the next speed and direction values. This might lead to unrealistic movements. Nevertheless the Random Walk is a widely used mobility model [10].

2.2 Random Waypoint Mobility Model

Nodes following the Random Waypoint model switch between pause periods and movements, i.e. they stay at a location for a randomly determined time and then randomly choose a speed between $[v_{min}, v_{max}]$ and a random destination in the simulation area. Having reached the destination the node pauses again and so on [10].

2.3 Random Direction Mobility Model

The Random Direction model [10] is similar to the Random Walk as the node also randomly chooses a speed from $[v_{min}, v_{max}]$ and a direction between 0 and 180 degrees. But in contrast to the Random Walk the node always moves to the border of the simulation area. Here the node pauses for a randomly determined time and after that chooses a new direction and a new speed.

The Random Direction model has the advantage to overcome so called density waves, i.e. a clustering of nodes in one part of the simulation area, that for example the Random Waypoint model suffers from [20].

2.4 Scenario Generation

As mentioned above a mobility model describes the behavior of a single node or a group of nodes by some formal definition. One common way to use mobility models in a simulation is to generate a mobility scenario. A mobility scenario contains the movement patterns of nodes that follow the definition from a mobility model, i.e. the scenario contains realizations of the mobility model. Scenario generators like BonnMotion [1] can create scenarios from a large list of mobility models that can then be loaded by simulation tools like OMNeT++ [13] to be used in a larger simulation model.

Without loss of generality we identify nodes by a number $i \in \mathbb{N}_{>0}$. We assume in the following that a scenario consists of waypoints that define at what time t a node i is at location (x, y), i.e. a waypoint is a tuple (i, t, x, y). The possible values for a location (x, y) are restricted by the size of the simulation area $\mathcal{C}$, i.e. we require $(x, y) \in \mathcal{C}$.

Then $\mathcal{S}^{(i)} = ((i, t_{i,1}, x_{i,1}, y_{i,1}), (i, t_{i,2}, x_{i,2}, y_{i,2}), \cdots, (i, t_{i,l}, x_{i,l}, y_{i,l}))$ contains all waypoints of a single node until the end of the simulation. The first waypoint, i.e. the initial location is usually chosen randomly. Further waypoints are always necessary when a node changes direction or speed, either explicitly by selecting a new destination, direction or speed randomly or implicitly when bouncing off the boundaries of the simulation area as for the Random Walk model. This implies for two consecutive waypoints $(i, t_{i,1}, x_{i,1}, y_{i,1})$ and $(i, t_{i,2}, x_{i,2}, y_{i,2})$ that in the time interval $[t_{i,1}, t_{i,2}]$ node i moves with constant speed from $(x_{i,1}, y_{i,1})$ to $(x_{i,2}, y_{i,2})$. For pause times the locations of two consecutive waypoints are identical.

The complete scenario for n nodes is then given by $\mathcal{S} = \left(\mathcal{S}^{(1)}, \mathcal{S}^{(2)}, \cdots, \mathcal{S}^{(n)}\right)$.

2.5 Autoregressive-To-Anything Processes

Autoregressive-To-Anything (ARTA) Processes [11] combine an autoregressive process of order p, denoted $AR(p)$, with an arbitrary marginal distribution F_Y. The $AR(p)$ is given by [7]

$$Z_t = \alpha_1 Z_{t-1} + \alpha_2 Z_{t-2} + \ldots + \alpha_p Z_{t-p} + \epsilon_t$$

where the α_i are autoregressive coefficients and the values ϵ_t, denoted as innovations, are normally distributed with zero mean and variance σ_ϵ^2. The ARTA process is then defined as a sequence

$$Y_t = F_Y^{-1}[\Phi(Z_t)], t = 1, 2, \ldots$$

where F_Y is the marginal distribution, Φ is the standard normal cumulative distribution function and $\{Z_t; t = 1, 2, \ldots\}$ is a stationary Gaussian $AR(p)$ process as described above.

ARTA processes can model correlated input processes with a wide variety of shapes for the distribution. The approach works for any distribution F_Y for which F_Y^{-1} can be computed, either by a closed-form expression or by numerical methods. Since the autocorrelations of the background $AR(p)$ process and the ARTA process are directly related and autoregressive processes are very flexible in modeling autocorrelation, the ARTA process inherits this property from the $AR(p)$ process. In addition, there are approaches available to construct ARTA processes from measured observations from a real system [11, 5].

3 Mobility Models with Arrivals and Departures

There are basically two possible approaches that can be used to extend mobility models such that they account for arrivals and departures. Of course, one can modify the definition of the mobility model itself to include the generation of new nodes and the deletion of departing nodes at runtime. Though, depending on the complexity of the mobility model this can be complicated and it has to be done for every mobility model that should be supported. Alternatively, one can leave the mobility model untouched and add arrivals and departures to the generated scenarios. Since for arrival and departure generation only the scenario is used, no knowledge on the mobility model that generated this scenario is required. We will follow this idea that is sketched in Fig. 1. Our model consists of three parts: The (unmodified) mobility model generates

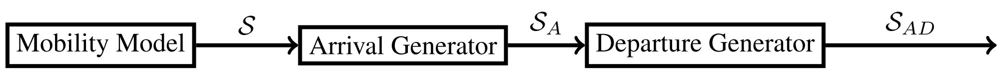


Figure 1: Scenario generation with arrivals and departures

a scenario $\mathcal{S}$ as described in Sect. 2. The *Arrival Generator* then adds additional nodes to the scenario resulting in a new scenario $\mathcal{S}_A$ and the *Departure Generator* modifies the scenario such that nodes leave the simulation area. The approach in Fig. 1 is very modular, as we have no real restrictions on the choice of the mobility model or the generators for arrivals and departures. We have already introduced three mobility models in Sect. 2 that we used later in our experiments. But of course the approach works for the other models mentioned in Sect. 1 as well.

In the following we describe the two generators from Fig. 1 in more detail and present an algorithm for scenario generation with arrivals and departures of nodes.

3.1 Arrival and Departure Generators

The Arrival and the Departure Generator work in a similar way, i.e. they have to (randomly) determine the time of an arrival or departure, the location where the node enters or leaves the simulation area and in case of departures also which node should leave. Therefore, the generators basically consist of probability distributions and stochastic processes to draw those random numbers.

In addition they have to utilize a set of entry coordinates $\mathcal{C}_{entry}$ and exit coordinates $\mathcal{C}_{exit}$, respectively. Of course, we have that $\mathcal{C}_{entry} \subset \mathcal{C}$ and $\mathcal{C}_{exit} \subset \mathcal{C}$ and additionally $\mathcal{C}_{entry}$ and $\mathcal{C}_{exit}$ should only consist of points at the boundary of $\mathcal{C}$. $\mathcal{C}_{entry}$ and $\mathcal{C}_{exit}$ can either be a discrete number of coordinates $(x_i, y_i), i = 1, \cdots, k$ or a continuous region $\{(x, y)|(x, y) \in \mathcal{C}\}$. Of course, also combinations of these two definitions are possible.

We define two sets here, because the entry and exit coordinates are not necessarily identical. If we model a part of an airport terminal, e.g. the route to the gates is only an exit point but not an entry point. In other scenarios like an university campus we might of course have that $\mathcal{C}_{entry} = \mathcal{C}_{exit}$, though.

The choice of the probability distributions or stochastic processes for random number generation of course depends on the system we want to model. In the simplest form one could just use standard distributions, e.g. an exponential distribution for interarrival and interdeparture times and an uniform distribution for the selection of entry and exit coordinates.

If the arrivals or departures should exhibit autocorrelation a stochastic process is required. We already introduced ARTA processes in Sect. 2 that are suitable for this task and that we used for our experiments. An alternative to this are Markovian Arrival Processes [9] that are more prominent for models that should be analyzed numerically, but can also be used in simulation.

It is of course also possible to use more elaborate stochastic processes like Marked MAPs [8] or Vector ARTA processes [6] that can generate interevent time and entry/exit coordinates in one step and can additionally express correlation between those two values.

For the Departure Generator a further distribution for the selection of the nodes has to be specified. Possible candidates are a discrete uniform distribution or a geometric distribution, that could be used to make the selection of a node with a small number (i.e. a node that is in the system for a long time) more likely.

Assume, that we have n initial nodes in the scenario without arrivals and departures. Then, more formally, the arrival generator creates a sequence

$$\Big((n+1, t_{n+1}^{(entry)}, x_{n+1}^{(entry)}, y_{n+1}^{(entry)}), (n+2, t_{n+2}^{(entry)}, x_{n+2}^{(entry)}, y_{n+2}^{(entry)}), \cdots\Big)$$

where $(x_{n+i}^{(entry)}, y_{n+i}^{(entry)}) \in \mathcal{C}_{entry}$. The i-th tuple is the first waypoint of node $n+i$. The remaining waypoints are then determined by the mobility model $\mathcal{M}$, i.e. we obtain

$$\mathcal{S}^{(n+i)} = \Big((n+i, t_{n+i}^{(entry)}, x_{n+i}^{(entry)}, y_{n+i}^{(entry)}), (n+i, t_{n+i,2}, x_{n+i,2}, y_{n+i,2}), \cdots\Big).$$

In a similar way, the departure generator creates a sequence

$$\Big((i_1, t_{i_1}^{(exit)}, x_{i_1}^{(exit)}, y_{i_1}^{(exit)}), (i_2, t_{i_2}^{(exit)}, x_{i_2}^{(exit)}, y_{i_2}^{(exit)}), \cdots\Big)$$

of nodes i_j that should leave the simulation area at location $(x_{i_j}^{(exit)}, y_{i_j}^{(exit)}) \in \mathcal{C}_{exit}$ at time $t_{i_j}^{(exit)}$. Assume that there are n initial nodes in the scenario and the arrival generator created l additional nodes. Let $\mathcal{N} \subseteq \{1, 2, \cdots, n+l\}$ denote all the nodes that exist in the scenario at a departure time $t^{(exit)}$. Then of course, the node i_j that should leave the area may only be drawn from $\mathcal{N}$.

In the final step the departure generator has to modify $\mathcal{S}^{(i_j)}$, i.e. a new waypoint $(i_j, t^{(reroute)}, x^{(reroute)}, y^{(reroute)})$ has to be determined. All waypoints $(i_j, t, x, y) \in \mathcal{S}^{(i_j)}$ with $t > t^{(reroute)}$ are discarded and $(i_j, t^{(reroute)}, x^{(reroute)}, y^{(reroute)})$ and $(i_j, t^{(exit)}, x^{(exit)}, y^{(exit)})$ are added as new waypoints. We will explain in the next section, where the scenario generation is described, how this new waypoint can be determined.

3.2 Scenario Generation

The algorithm for a scenario generation that includes arrivals and departures is sketched in Fig. 2. As already mentioned, we are using a modular approach and consequently the algorithm consists of three parts: The creation of the scenario from the mobility model without arrivals and departures (line 1), extending the scenario with arrivals (lines 2 - 10) and the addition of departures (lines 11-21). As inputs the algorithm takes the mobility model $\mathcal{M}$, the size or coordinates of the simulation

Input: Mobility Model $\mathcal{M}$, Coordinates of simulation area $\mathcal{C}$;
Input: Arrival Generator $\mathcal{A}$, Entry coordinates $\mathcal{C}_{entry}$;
Input: Departure Generator $\mathcal{D}$, Exit coordinates $\mathcal{C}_{exit}$;
Input: simtime, offset, n
Output: Mobility scenario $\mathcal{S}_{AD}$;

generate $\mathcal{S} = \left(\mathcal{S}^{(1)}, \mathcal{S}^{(2)}, \cdots, \mathcal{S}^{(n)}\right)$ from $\mathcal{M}$;
t = offset;
c = n+1;
repeat
  draw t_{sample} according to $\mathcal{A}$; $t_c^{(entry)} = t + t_{sample}$;
  draw $(x_c^{(entry)}, y_c^{(entry)})$ from $\mathcal{C}_{entry}$ according to $\mathcal{A}$;
  generate $\mathcal{S}^{(c)} = \left((c, t_c^{(entry)}, x_c^{(entry)}, y_c^{(entry)}), \cdots\right)$ from $\mathcal{M}$;
  t = $t_c^{(entry)}$; c = c+1;
  $\mathcal{S} = \mathcal{S} \cup \mathcal{S}^{(c)}$;
until t $>$ simtime
t = offset;
repeat
  draw t_{sample} according to $\mathcal{D}$; $t^{(exit)} = t + t_{sample}$;
  draw $(x^{(exit)}, y^{(exit)})$ from $\mathcal{C}_{exit}$ according to $\mathcal{D}$;
  determine nodes $\mathcal{N}$ that are available at time t_{exit};
  draw i from $\mathcal{N}$;
  compute $(t^{(reroute)}, x^{(reroute)}, y^{(reroute)})$ from $\mathcal{S}^{(i)}$;
  $\mathcal{S}'^{(i)} = ((i, t_{i1}, x_{i1}, y_{i1}), \cdots, (i, t_{ij}, x_{ij}, y_{ij}),$
    $(i, t^{(reroute)}, x^{(reroute)}, y^{(reroute)}, (i, t^{(exit)}, x^{(exit)}, y^{(exit)})\big)$;
  $\mathcal{S} = \mathcal{S} \setminus \mathcal{S}^{(i)} \cup \mathcal{S}'^{(i)}$;
  t = $t^{(exit)}$;
until t $>$ simtime
$\mathcal{S}_{AD} = \mathcal{S}$;

Figure 2: Algorithm for scenario generation

area $\mathcal{C}$, the simulation time and the number of nodes n that populate the area in the beginning. Further inputs are related to the arrivals and departures, i.e. we need a list of entry and exit coordinates, an arrival generator $\mathcal{A}$ and a departure generator $\mathcal{D}$, that are basically probability distributions or stochastic processes we can sample from. The *offset* indicates when arrival and departure generation should start, i.e. we simulate the initial n mobile nodes only for *offset* time units before arrivals and departures start.

First, the algorithm generates a scenario $\mathcal{S}$ that contains the movement patterns for the n initial nodes according to the mobility model $\mathcal{M}$ in line 1, i.e. it calls a subroutine for an existing mobility model like Random Waypoint or Random Direction. In the second step arrivals are added. We sample the next arrival time $t_c(arrival)$ from the arrival generator (line 5) and determine the entry coordinates from $\mathcal{C}_{entry}$ (line 6). In the algorithm the arrival time and the entry coordinates are determined independent of each other. Once the entry point and the arrival time are known we use the mobility model $\mathcal{M}$ to generate the movement patterns for the new node (line 7). Thus, the movements of the initial nodes and the generated nodes basically differ in the generation of the first waypoint. While the initial nodes start at $t = 0$ at some random point of the simulation area, nodes created by the arrival generator start at an entry point at some time during the simulation. After that they behave similar according to mobility model $\mathcal{M}$. Finally, the movement patterns of the newly generated node is added to the scenario and the time is increased.

The last part of the scenario generation consists of the computation of departures. The first steps are similar to the arrival generation, i.e. we draw the departure time $t^{(exit)}$ and the exit coordinates (lines 13 and 14). In addition to this information we also have to determine which node should leave the simulation area (lines 15 and 16). Note, that our scenario $\mathcal{S}$ contains the movement of all nodes and some of them did probably not exist at time $t^{(exit)}$. Hence, we collect in $\mathcal{N}$ all nodes that inhabit the simulation area at $t^{(exit)}$. Recall, that we used an *offset* for the beginning of the arrival and departure generation. It is advisable to use an offset here as well, i.e. only nodes that have existed for at least *offset* time units in the model at time $t^{(exit)}$ are collected in $\mathcal{N}$. The reasons for this offset will become obvious later when we describe how the node is routed to the exit point.

From $\mathcal{N}$ we randomly determine one node for departure. In lines 17 and 18 new waypoints for this node are computed, i.e. we identify a time $t^{(reroute)}$ and a corresponding location $(x^{(reroute)}, y^{(reroute)})$ where the existing movements of the node are interrupted and from where it is rerouted to the exit point. We will explain below how this is done exactly. Finally, the old waypoints for the departing node are deleted from the scenario and replaced by new waypoints including the departure.

In accordance with Fig. 1 line 1 of the algorithm describes the scenario generation by the mobility model, lines 2 - 10 constitute the arrival generator and lines 11-21 describe the departure generator.

Fig. 3 depicts how the rerouting of mobile nodes for departure works. The blue lines starting at (a) and ending at (b) are the original movement patterns as generated from the mobility model. (e) is the exit point where the node is supposed to depart at time $t^{(exit)}$. First we compute the location of the node at time $t^{(exit)}$ according to the original movement pattern. This location is labeled with (c) in Fig. 3. The remaining part of the original movement pattern, i.e. the dashed line between (c) and (b) is discarded. Starting from (c) we process the node's movement backwards until we have found a location $(x^{(reroute)}, y^{(reroute)})$ and the corresponding time $t^{(reroute)}$, such that the node can cover the distance between $(x^{(reroute)}, y^{(reroute)})$ and the exit point in time $t^{(exit)} - t^{(reroute)}$ with an appropriate speed, i.e. a speed that is for example

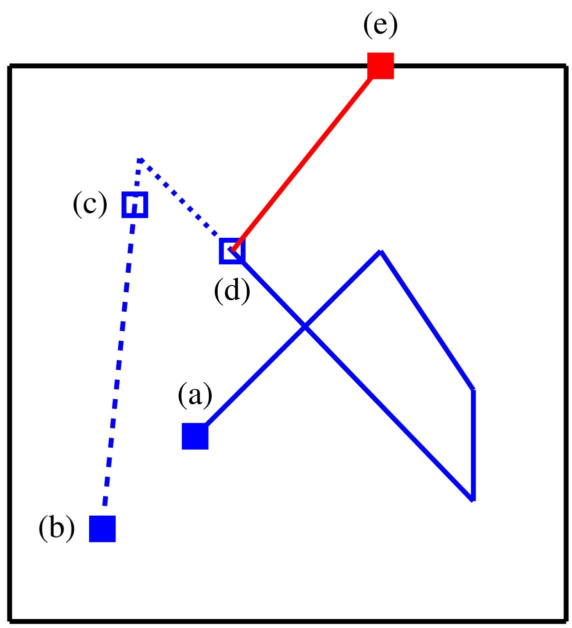


Figure 3: Rerouting for departure

drawn from the speed range for that mobility model or corresponds to the node's mean speed. This location is labeled (d) in Fig. 3. The movements between the locations (d) and (c) are also discarded and the node gets new waypoints for the locations (d) and (e).

From Fig. 3 it becomes obvious why the *offset* introduced in the algorithm in Fig. 2 is helpful. If a departure is due at the very beginning of the lifetime of a node it might not be possible to find a suitable location to reroute the node, since it has hardly moved yet. The *offset* ensures that all nodes applicable for departure have existing movement patterns at the time of the departure.

Of course, the rerouting introduces some overhead when generating the scenario, because parts of the already generated movement patterns are discarded again. However, it has the important advantage that it can be used for any mobility model, since it works only on the generated movement patterns and no knowledge about the mobility model or changes to the mobility model are required.

4 Experimental Evaluation

To systematically assess the effect of arrivals and departures on the performance of a wireless network we combined mobility models with different arrival and departure generators with varying rates and correlation for the creation and deletion of nodes.

Although the approach presented in Sect. 3 is very general and not specific to certain mobility models, we conducted our experiments with three basic random mobility models that are well known and understood, in particular the Random Walk, Random Direction and Random Waypoint models.

4.1 Experiment Setup

Our experiments were performed using OMNeT++ [13] and the INET framework, that supports mobility scenarios in the form described in Sect. 2. The extension with arrivals and departures required slight modifications of the standard modules from OMNeT++ to allow for nodes to become active (i.e. arrive) or inactive (i.e. leave) during the simulation run.

For the experiments we used a simple quadratic simulation area of $100 \times 100m^2$. There are four access points that cover the area as shown in Fig. 4. We added nine entry and exit points ($\mathcal{C}_{entry} = \mathcal{C}_{exit}$) evenly to the border of the area and

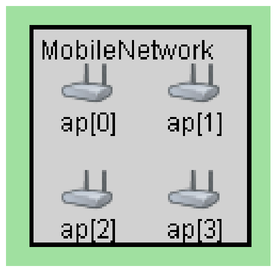


Figure 4: Simulation area with access points

generated various mobility scenarios using the algorithm from Fig. 2 for Random Walk, Random Direction and Random Waypoint models that differed in the number of initial nodes, the rate of arrivals and departures or the correlation of arrivals and departures. In all models we assumed that the mean arrival rate and the mean departure rate are equal to keep the mean number of nodes equal to the initial number of nodes. The speed of the mobile nodes lies within the interval $v_{min} = 3km/h$ and $v_{max} = 8km/h$. If the model supports pause times they are between 0 and 30 seconds. As *offset* we used 100 seconds, i.e. the model is simulated for 100 seconds before arrival and departure generation starts. In addition we required that a node has to exist for at least 100 seconds before it may be selected for departure. Each scenario was simulated for 180 minutes.

At randomly chosen times the mobile nodes generate traffic. To keep the model simple and allow for a better control of the generated data volume, we modeled traffic generation at an abstract level without including all the network layers. The access points are basically servers with a buffer size of 50 that handle the traffic randomly generated by the nodes that are close to them, i.e. each node generates load for the access point that is closest to its current location. When a node is moving the nearest access point might of course change during the simulation. To be able to assess whether the different scenarios have an effect on the performance we measured the queue length distribution in the four access points.

4.2 Experimental Results

Before we present the results of the queue length distribution we visualize the effect of arrivals and departures using the spatial node distribution. The spatial node distribution shows the probability that nodes are at the different locations of the simulation area. Fig. 5 shows the spatial node distribution for a Random Waypoint model with an initial number of $n = 30$

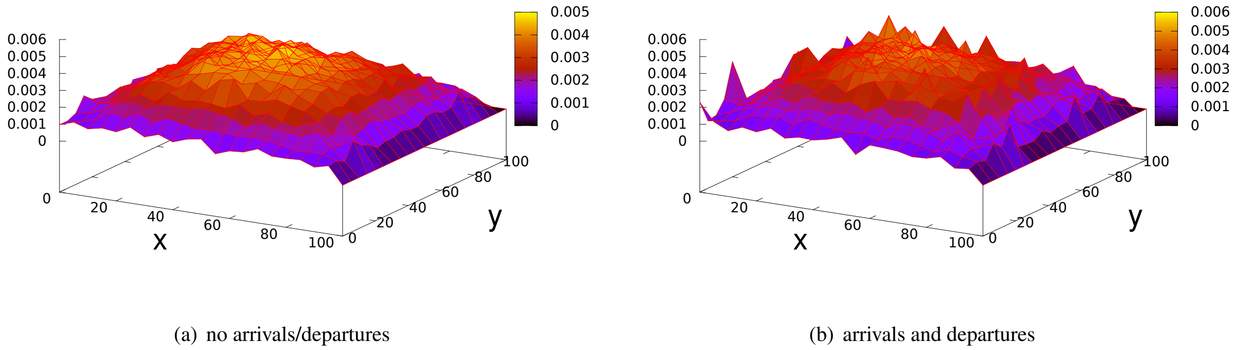


Figure 5: Spatial Node Distribution of Random Waypoint Model

nodes with and without arrivals and departures. As we can see in Fig. 5(b) the probability that nodes are at the border of the simulation area where the entry and exit points are increases, while the distribution remains similar in other parts of the area. Fig. 6 shows the number of nodes and the average number of nodes that are present in the simulation area for a Random

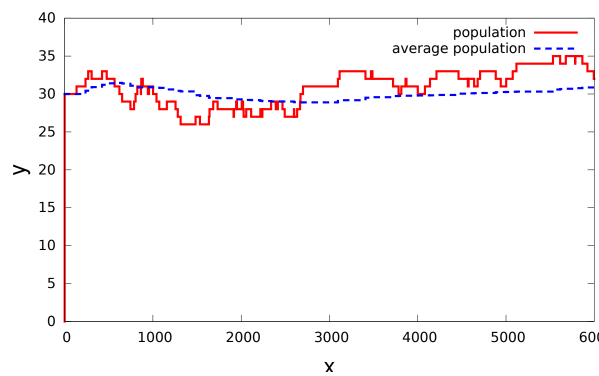
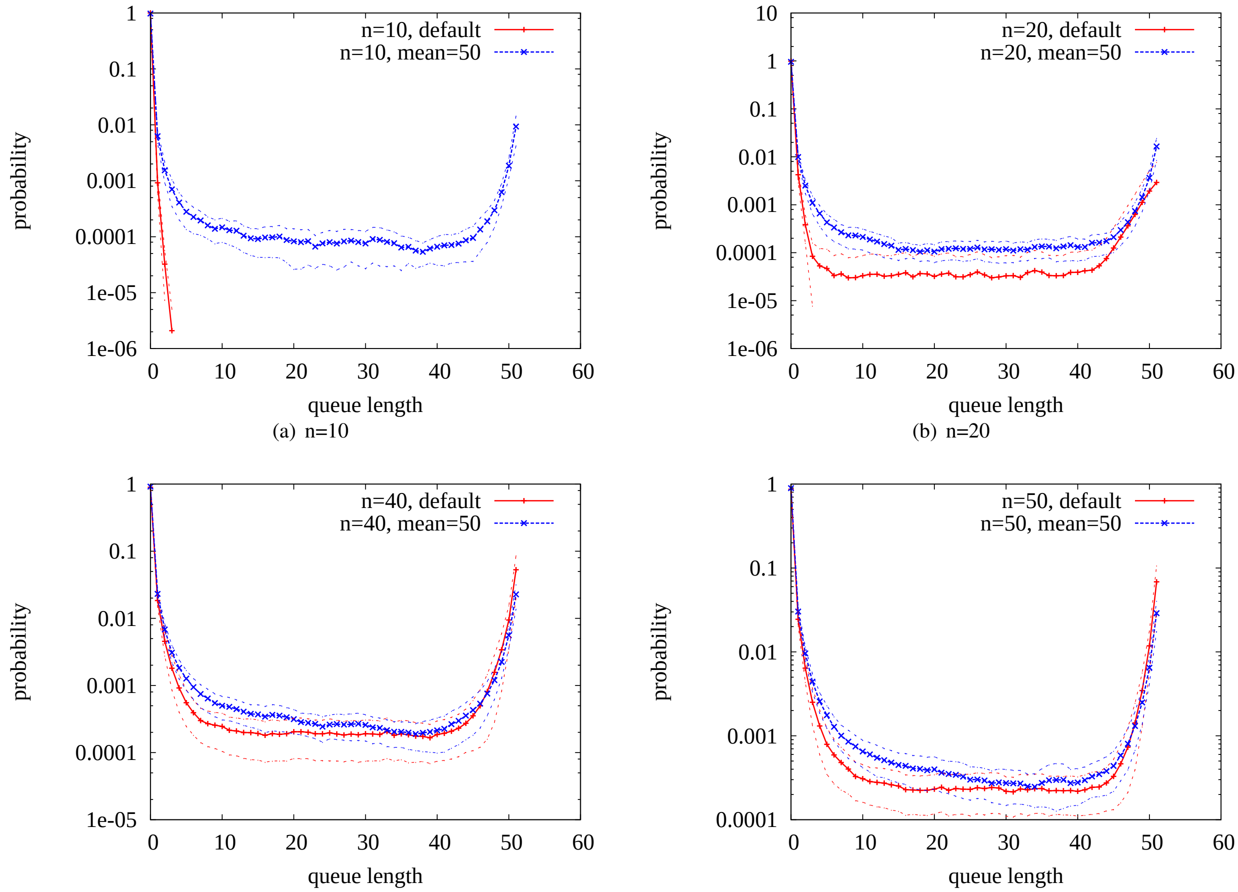


Figure 6: Number of nodes present in the simulation area

Waypoint model with $n = 30$ and arrivals and departures for the first 6000 seconds. As we can see the number of nodes varies around $n = 30$ (while the average number of nodes remains almost constant), implying that there are periods with a higher load for the access points and periods with a lower load. The effect of these periods on the access points is evaluated in the following.

The following simulation results are all obtained from 30 replications of the simulation model. If applicable we also present 90% confidence intervals for the results, though for plots with a larger number of curves we omitted them to keep the plots accessible. In a first series of experiments we compared the effect of arrival and departures for different numbers of initial nodes n. As a reference value we simulated the original default scenarios with a constant number of nodes and



(a) n=10 (b) n=20 (c) n=40 (d) n=50

Figure 7: Queue length distribution for different numbers of initial nodes and the Random Direction model (thin dashed lines denote the 90% confidence intervals)

compared it with scenarios where arrivals and departures occur according to an exponential distribution with mean 50. The entry and exit nodes are drawn independently from an uniform distribution. Fig. 7 shows the queue length distribution at the first access point for the Random Direction model for an increasing number of initial nodes. The results for the other access points and mobility models are similar. As we can see arrivals and departures have a large effect for the smaller node numbers but the effect diminishes if we increase the number of nodes (i.e. the difference in the mean values becomes smaller and the confidence intervals start to overlap). Obviously, this is because fluctuations in the node number caused by arrivals and departures have a larger influence if the initial number of nodes is relatively small compared to the size of the fluctuation, i.e. three additional nodes are easily noticeable if there are 10 nodes present but the effect disappears if there are 50 nodes.

For the next experiments we kept the initial number of nodes fixed and varied the arrival rate. Results for the Random Waypoint model are shown in Fig. 8. The plot shows the curves for the default model without arrivals and departures and for models where the arrivals and departures follow an exponential distribution with mean 10 and 50, respectively. As we can see, the queue length increases slightly with smaller mean values (i.e. larger rates).

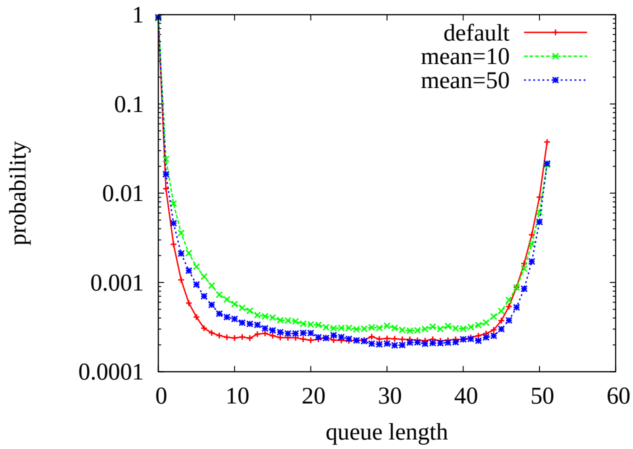
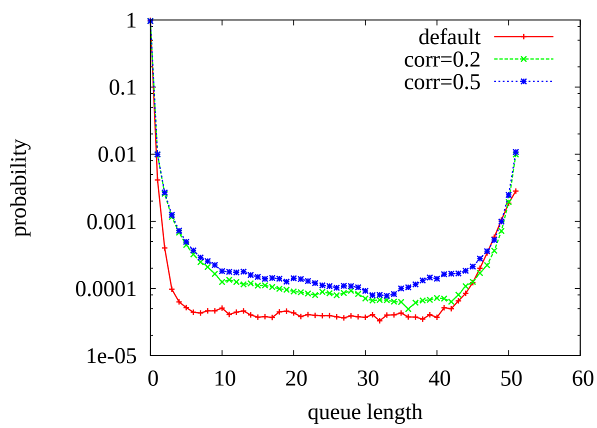
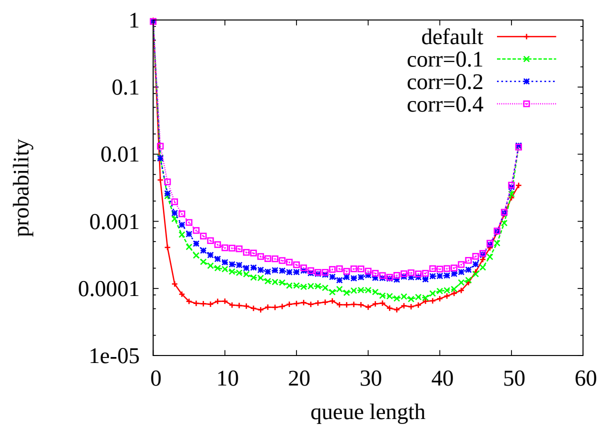


Figure 8: Queue length distribution for different arrival rates for the Random Waypoint model and $n = 30$

As mentioned before, it is likely that arrivals and/or departures are correlated in some real world scenarios. We already



(a) Random Walk model, correlated arrivals



(b) Random Waypoint model, correlated arrivals and departures

Figure 9: Queue length distribution for different levels of lag-1 autocorrelation for $n = 20$

introduced ARTA processes in Sect. 2 that can serve to model autocorrelated interarrival or interdeparture times. In the last experiments we evaluated the effect of autocorrelation on the mobile network. Fig. 9(a) shows the results for a Random Walk model where the arrivals are generated according to an ARTA process with exponential distribution and different levels of autocorrelation, while the departures follow an exponential distribution and are uncorrelated. As we can see, an increased autocorrelation also results in a larger queue length. Similar results can be observed in Fig. 9(b) where both, arrivals and departures, are generated by identical ARTA processes and thus, are correlated.

The experimental results clearly indicate, that arrivals and departures can have a significant effect on the performance of a wireless network. We have also seen that this effect becomes larger if the variation in the number of nodes is relatively large compared to the mean number of nodes, which can be caused by a higher arrival rate or correlated arrivals.

5 Conclusions

We have presented an approach to combine mobility models with stochastic processes to account for the arrival and departure of nodes during simulation. The approach works on the generated scenarios and thus, can easily be combined with any mobility model. Arrivals and departures of mobility nodes occur in many real world scenarios (like airports, shopping centers, parts of an university campus) and our experimental study suggests, that modeling of arrivals and departures can have a significant effect on the performance results.

Of course, the results presented here can only serve as a first step towards more realistic mobility models. We only used completely synthetically generated mobility scenarios in our study. Mobility models based on real-world observations naturally qualify for an extension with arrival and departure generators since the observations already contain information about nodes that newly arrive or leave the area, but are subject to further research.

References

[1] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn. BonnMotion - a Mobility Scenario Generation and Analysis Tool. In *Proc. of SIMUTools*, 2010.

[2] A. Balachandran, G. Voelker, P. Bahl, and P. Rangan. Characterizing user behavior and network performance in a public wireless LAN. In *SIGMETRICS*, 2002.

[3] C. Bettstetter. Smooth is better than sharp: a random mobility model for simulation of wireless networks. In *Proc. of MSWIM*, 2002.

[4] H. Bhatia, R. B. Lenin, A. Munjal, S. Ramaswamy, and S. Srivastava. A Queuing-theoretic Framework for Modeling and Analysis of Mobility in WSNs. In *Proc. of PerMIS*, 2008.

[5] B. Biller and B. Nelson. Fitting time-series input processes for simulation. *Oper. Res.*, 53(3):549–559, 2005.

[6] B. Biller and B. L. Nelson. Modeling and Generating Multivariate Time-Series Input Processes using a Vector Autoregressive Technique. *ACM Transactions on Modeling and Computer Simulation*, 13(3):211–237, 2003.

[7] G. Box and G. Jenkins. *Time Series Analysis - forecasting and control*. Holden-Day, 1970.

[8] P. Buchholz, P. Kemper, and J. Kriege. Multi-class Markovian arrival processes and their parameter fitting. *Performance Evaluation*, 67(11):1092–1106, 2010.

[9] P. Buchholz, J. Kriege, and I. Felko. *Input Modeling with Phase-Type Distributions and Markov Models - Theory and Applications*. Springer, 2014.

[10] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications & Mobile Computing*, 2002.

[11] M. Cario and B. Nelson. Autoregressive to anything: Time-series input processes for simulation. *Operations Research Letters*, 19(2):51–58, 1996.

[12] Y. Chen, J. Kurose, and D. Towsley. A mixed queueing network model of mobility in a campus wireless network. In *Proc. of INFOCOM*, 2012.

[13] R. Hornig and A. Varga. An Overview of the OMNeT++ Simulation Environment. In *Proc. of SIMUTools*, 2008.

[14] A. Jardosh, E. Belding-Royer, K. Almeroth, and S. Suri. Towards realistic mobility models for mobile ad hoc networks. In *Proc. of MobiCom*, 2003.

[15] M. Kim, D. Kotz, and S. Kim. Extracting a Mobility Model from Real User Traces. In *Proc. of INFOCOM*, 2006.

[16] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. Netw.*, 1994.

[17] B. Liang and Z. Haas. Predictive distance-based mobility management for PCS networks. In *Proc. of INFOCOM*, 1999.

[18] W. Navidi and T. Camp. Stationary distributions for random waypoint models. *IEEE Transactions on Mobile Computing*, 2004.

[19] G. Resta and P. Santi. The QoS-RWP mobility and user behavior model for public area wireless networks. In *Proc. MSWiM*, 2006.

[20] E. Royer, P. Melliar-Smith, and L. Moser. An analysis of the optimum node density for ad hoc mobile networks. In *Proc. of the IEEE ICC*, 2001.

[21] C. Tuduce and T. Gross. A mobility model based on WLAN traces and its validation. In *Proc. of INFOCOM*, 2005.

[22] J. Yoon, B. Noble, M. Liu, and M. Kim. Building realistic mobility models from coarse-grained traces. In *Proc. of MobiSys*, 2006.