Model Complexity vs. Better Parameter Value Estimation—Comparing four Topography-independent Radio Models

Jochen Koberstein, Sören Witt, Norbert Luttenberger
Research Group for Communication Systems
Dept. of Computer Science
University of Kiel, Germany
{jko|swi|nl}@informatik.uni-kiel.de

ABSTRACT

In this case study we show how four topography-independent radio models (models not regarding individual obstacles in the operations area) can be "tuned" such that the outcomes of an event-driven network simulator come as close as possible to reference data obtained in a real-world radio network experiment. An evolutionary optimization process was applied to find near optimal parameter values four these models. We describe the optimization process, the error function that guides it and discuss the results.

Categories and Subject Descriptors

I.6.4 [**Computing Methodologies**]: Simulation and Modeling—*Model Validation and Analysis*

Keywords

Radio Model, Comparison, Real World, Optimization, Calibration, Simulation Model Validation

1. INTRODUCTION

It is a commonly accepted assumption that the accuracy of radio network simulation depends to a large degree on the quality of the underlying radio model. The network application designer has to choose from a great variety of such models, ranging from a "simple range model", assuming constant radio signal strength in a sharp-edged circular area, to very elaborate models that may also take into account the respective area's topography, including descriptions of individual obstacles in that area and formulas for the related radio signal attenuation. Each of these models brings with it a set of parameters for which values need to be provided—in the above mentioned simple range model e.g. the range, i.e. the diameter of the implied disc. After having overcome the burden to choose an appropriate radio model, the network application designer may thus feel trapped for a second time: What are appropriate parameter values? We call the process of identifying "appropriate" parameter values also "calibrating the model".

In this paper, we present some preliminary results that reveal that simulation results of nearly equal accuracy can be obtained for a set of topography-independent radio models if the per-model required parameter set values are chosen carefully. By the term "topography-independent radio models" we denote models that do not encompass a detailed knowledge of individual obstacles in the intended radio network operations area. In contrast to these models, topography-oriented models come with a very high degree of complexity and have their value e.g. in mobile network planning. This is beyond the scope that we have in mind; we are interested in general purpose simulation for e.g. the pre-deployment analysis of wireless sensor networks (WSN).

An important question in the raised context is: How to define and to determine the "accuracy of simulation results"? In our case study, we decided for an approach that compares event traces generated by an event-driven network simulator with event traces measured during a real-world experiment of a dedicated simple radio network. We equipped the network simulator in sequence with four different topography-independent radio models, and each time tried to systematically determine the "best" parameter values for the respective radio modes, i.e. parameter values that produced a replay of the real-world experiment as close to perfection as possible.

In our approach, traces are not compared in total ("event-by-event"), but according to a focus that can be chosen by the network application designer. This focus is defined by a so-called error function. The error function expresses some functional interrelation between sets of events. A simplistic example for this could be: the duration between related start and stop events. For systematic variation of parameter values we employed evolutionary algorithms. The search of the parameter value space conducted by these algorithms is guided by the error function.

The paper is organized as follows: In the following section we introduce the pre-requisites needed for the applied method for comparison which is described in section 3. Section 4 presents the found results and comprises an according interpretation before section 5 points out some related work. Section 6 concludes the contribution.

SIMUTools/SCENES Workshop 2009, Rome, Italy

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5629
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5629

2. PRE-REQUISITES

2.1 Simulator

For this case study we use the OMNeT++ simulator [16] accompanied by the INET framework [1] and the SEE framework [6] (Simulator Extension for Operations Environment Models). While OMNeT++ provides basic distributed event simulation the INET Framework adds node mobility and radio simulation. The SEE approach adds a hardware abstraction layer which enables us to use identical implementations of the application software on real hardware and in the simulation. To replay node movements captured during a real-world experiment in the simulator, we use a graph based mobility model as presented in [7].

2.2 Reference Data Acquisition

As common baseline for comparing the four different radio models introduced below, we acquired reference data from a small-scale, real-world WSN deployment. This network consisted of four nodes, which are based on modified Linksys WRT WiFi-Routers running *OpenWRT*[12]. To make this platform usable for WSN applications it was extended by a GPS-module and a sensor interface. The WRT provides enough resources to record communication and motion events. The nodes were attached to cars, bicycles or carried by pedestrians which moved inside a defined urban area of about $1km^2$, following arbitrary routes. The recorded GPS positions are used to create the graphs for node mobility in the simulator.

The "application" software for reference data acquisition running on these nodes periodically emits beacons with a certain frequency and records every beacon received from other nodes. The mentioned SEE framework allowed a seamless portage of the application software from the simulator to the real world device and back.

2.3 Radio Models

We chose to study four radio models with different levels of complexity. They actually differ in the way the attenuation of radio signal power, i.e. the path loss, is modeled and if bit errors regarding potentially received packets are calculated or not. Note that these are models for physical phenomenons. Related models e.g. regarding the MAC layer are protocol issues and are left unstudied since we suppose that these might be identical in the real world and in the simulation.

Right from the beginning, the reader should note that this paper does not try to contribute to the design of enhanced and more accurate radio models, e.g. according to [13]; instead it focusses on the question, what level of accuracy with respect to the network behavior in total can be achieved by simulation, when some simple radio models are equipped with parameter values that are carefully estimated by an optimization procedure. To put it to an extreme: As soon as we can define a parameter configuration for a radio model that yields reasonable accuracy according, we don't care too much for the model's internal properties.

2.3.1 Simple Range Model

The simple range model assumes a sharp-edged circular area in which a message may be received. The receive power within this range equals the transmitter power while it is zero outside. Hence we have a hard cut between receiving at full signal strength and not receiving at the defined distance as sketched in figure 1.

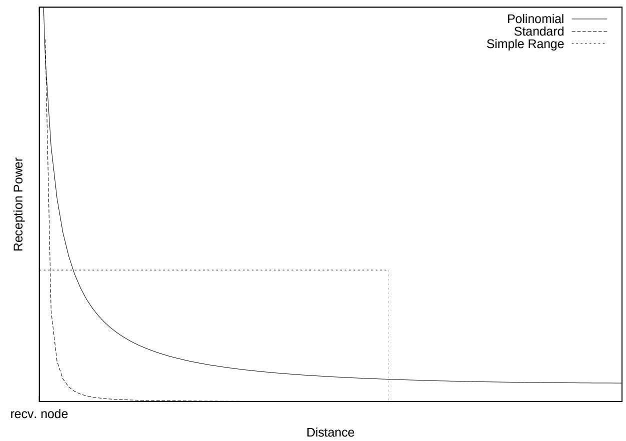

Figure 1: Four models for radio attenuation (Sketch)

A calculation of bit errors in received packets is omitted. The only possible reason for not receiving a packet while in range are collisions with other transmissions. The model takes only a single parameter, which is the radio range in meters.

2.3.2 Polynomial Model

The polynomial model shapes the radio path loss in an abstract way. It calculates the reception power P_{rec} according to

$$P_{\text{rec}}^{\text{polynomial}} = \frac{P_{\text{trans}}}{c_0 + c_1 \cdot d + c_2 \cdot d^2} \quad (1)$$

with transmitter power P_{trans} and the distance d between sending and receiving node. This approach is borrowed from a model for light attenuation in computer graphics as e.g. used in OpenGL [14, 2]. Hence the coefficients c_0 to c_2 do not have any physical "meaning" but can be seen as abstract regulating screws.

Additionally to the more complex radio attenuation this model introduces the calculation of bit errors which may occur more often in situations of poor reception, e.g. due to other transmissions nearby or a generally noisy environment. The calculation of bit errors is also part of the standard and the urban model described below.

Four additional parameters are introduced (besides c_0 to c_2), namely the transmitter power, the general noise level, the signal to noise interference threshold and the sensitivity of the radio. All of these influence the decision, if a packet may be received successfully at all or if it falls victim to noise level or interference with other transmissions. The former two also influence the bit error calculation.

2.3.3 Standard Model

The standard model also assumes purely radial dissemination as the models described above. Furthermore it calculates bit errors in the same way as the polynomial model. The only difference lies in the calculation of the receive power. It considers the wavelength $\lambda = \frac{\text{speed of light}}{\text{carrier frequency}}$ as physical value and the geometry of an assumed spheric dissemination (compare to Friis transmission equation). It further introduces the path loss exponent α instead of abstract

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5629
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5629

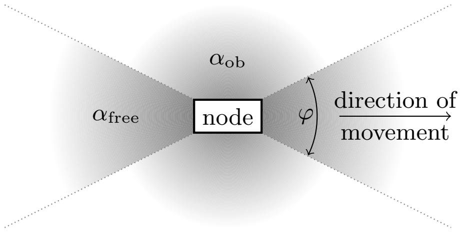

Figure 2: The unit-disc radio model is extended by the definition of sectors of different path loss exponents.

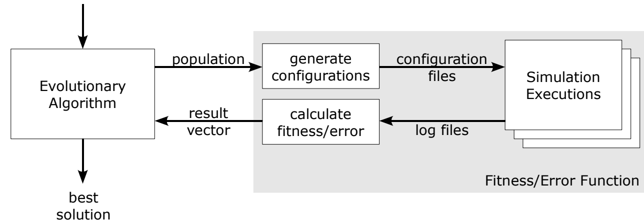

Figure 3: Basic layout of the optimizer

polynomial coefficients.

$$P_{\mathrm{rec}}^{\mathrm{standard}} = \frac{P_{\mathrm{trans}}}{4\pi \cdot d^{\alpha}} \cdot \frac{\lambda^2}{4\pi} \qquad (2)$$

The set of parameters remains the same as for the polynomial model except that c_0 to c_2 are exchanged by α. Note that there are dozens of path loss based attenuation models like this one. We call it the "standard" model since it is the standard model used for IEEE 802.11 radios in the OMNeT++/INET framework.

2.3.4 Simple Urban Model

To advance the precision towards the urban scenario, we extended the standard model by areas of different attenuation. This is expected to account for obstacles like buildings. To keep the model simple, we assume to have streets of houses leading to areas of expected low attenuation along the direction of a node's movement and high attenuation in the orthogonal direction. This is modelled by cones as depicted in figure 2 and discussed in [7].

Accordingly the model introduces an additional parameter α_{ob} regarding the path loss for obstructed propagation and a parameter for the angle φ. Note that this model is still topography-independent since we do not capture any map influencing the radio but use the orientation of the nodes only. The other parameters are identical to the standard model.

3. METHOD FOR COMPARISON

The radio models introduced in the previous section have different sets of parameters, that define their bahavior. As these models are an abstraction of reality, one must not expect that a model parameter, which is set to a value taken from the reality, will make the model behave realistic. E.g. the transmitter power actually emitted by a radio in reality (including antenna etc.) does not necessarily match the value which was configured. Hence we can at most estimate the correct value to be used in simulation.

Thus before one can decide which simulated radio model is adequate, each of these models need to be configured such that they behave as realistic as possible. Therefore a metric must be found, that measures one or more characteristic aspects of the real world, and which can be acquired by simulation in the same way too. Furthermore it needs to be applicable to all models which are going to be compared.

3.1 Evolutionary Approach

Evolutionary optimization algorithms are a well known meth-od for determining parameter values in areas where analytical methods are inefficient, unknown or hard to proof. As numerous articles are available on evolutionary algorithms (EAs), we discuss them here only briefly.

In [8] we introduced a method and framework, that runs an EA with a fitness function that is actually based on simulation. The EA generates a set of vectors x_i which is called a population. Each vector (called an individual in the scope of EAs) contains a fixed number of p values corresponding to p parameters of the simulation model under inspection. Hence an individual reflects a candidate solution in the p-dimensional search space. The EA passes this population to the fitness/error function f and "expects" it to return one fitness/error value per individual, measuring the quality of the according configuration.

The EA scans f for a global minimum (error function) or maximum (fitness function) and generates the following populations. This generation process works not (completely) randomly but e.g. follows tendencies in the landscape of resulting fitness or error values. As we are aiming to minimize the difference between simulation and real world behavior, we are talking about an error function in the following. Until now we are using Genetic Algorithms and CMA-ES [5] as EAs—other are possible.

Evaluating the error function means in our framework to first convert the individuals x_i into configuration files which are recognized by the simulator (see figure 3). Running according simulations produces a set of log-files which are interpreted to calculate the regarding error values as described in the following section.

Naturally the number of parameters may differ heavily between studied models. The larger p is, the longer an optimization process will take. To keep the search space not only finite but also reasonably small, a range of allowed values needs to be specified for each parameter. Some ranges may already be limited by some boundary conditions given by the scenario, e.g. a negative radio range is not feasible. Beyond this, choosing such ranges means to find a balance between acceptable execution time and the risk to miss possibly good solutions, e.g. the radio range for a typical WLAN-radio will not exceed 500m. Table 1 shows all parameters and their limits used for the currently discussed radio models.

3.2 Error Function

The error function for our case study ranks a single candidate solution by quantifying the difference between real world and simulation behavior. This can be achieved by comparing measurements from real world experiments with measurements from simulation executions, that perform a replay of the real world experiment.

We decided to take an application oriented perspective for

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5629
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5629

Parameter	Simple Range	Polynomial	Standard	Simple Urban
(Free Air) Path Loss Exp.	–	–	1.0 to 4.0	1.0 to 4.0
Obstructed Path Loss Exp.	–	–	–	2.0 to 6.0
Free Air Angle	–	–	–	0.1 to 20.0
Thermal Noise [dBm]	–	-100.0 to -105.0	-100.0 to -105.0	-100.0 to -105.0
Radio Sensitivity [dBm]	–	-70.0 to -100.0	-70.0 to -100.0	-70.0 to -100.0
Transmitter Power [mW]	–	50.0 to 150.0	50.0 to 150.0	50.0 to 150.0
SNIR Threshold [dBm]	–	2.0 to 20.0	2.0 to 20.0	2.0 to 20.0
all polynomial coefficients	–	-50.0 to 50.0	–	–
Distance [m]	10.0 to 500.0	–	–	–

Table 1: Overview of parameter ranges scanned by the optimization algorithm

defining the error function, i.e. a position that reflects radio behavior from the point of view of the application software running on the mobile nodes. The error function resulting from this approach measures the "encounters" between mobile nodes. "Encounter" is defined as time span during which two or more mobile are able to exchange data. Application software experiences encounters as a set of successful receive operations.

Let the network consist of a set of n nodes, then each node may possibly receive beacons from $n-1$ other nodes. Hence we can suppose for each node r a set of $n-1$ chronological sequences regarding beacons received from one other sending node s. (A receiver cannot be the sender at the same time.) Regarding the whole network, we have $n \cdot (n-1)$ of such chronological sequences describing the observed encounter pattern. We are partitioning these sequences into timeslots of one second each. Given a frequency of 8 Hz for sending beacons, each slot has a count c_i with $0 \leq c_i \leq 8$ of received beacons. The time considered in the real-world experiment is derived from the GPS time, which is synchronous over the whole network.

Discriminating 'good' from 'bad' configurations of the radio model requires to compare the encounter pattern resulting from a simulation run to the reference encounter pattern determined from real-world log files. For all sender/receiver pairs and all encounters in both encounter patterns, we can calculate per time slot the difference $c_i^{\text{ref}} - c_i^{\text{sim}}$. A negative value for this difference shows how many packets are indicated as received by the simulator in that time slot in excess to what happened in the real-world experiment, a positive value shows how many packets were not received by a simulated node in that timeslot. Summing up the absolute values of these differences lets us express in a single value the amount of vain and missing packets for one chronological sequence of k seconds regarding a sending node s and a receiving node r:

$$\Delta_{r,s} = \sum_{i=0}^{k} \left| c_i^{\text{ref}} - c_i^{\text{sim}} \right| \tag{3}$$

Let N be the set of all nodes, we finally compute the mismatch for the whole encounter pattern as

$$E = \sum_{r \in N} \sum_{s \in N, r \neq s} \Delta_{r,s} \tag{4}$$

E is an error value, expressing the number of misplaced and missing beacons in the simulated network. Thus E has to be minimized using the optimizer.

Model	minimal error (← more accurate)
Simple Range	11425
Polynomial	11414
Standard	11328
Urban	10406

Table 2: Achievable error values for different models

Model	simsec/sec (faster →)
Simple Range	162.13
Polynomial	161.17
Standard	135.72
Urban	133.98

Table 3: Processing effort for different models

4. RESULTS

The results presented below are all obtained running the optimization process as outlined. Depending on the number of parameters the search space varies significantly in size. The number of simulation runs needed to find relevant results varies accordingly. The termination condition depends on the algorithm in use: For the genetic algorithm we chose to abort if no better solution was found for more than 50 generations. CMA-ES has some internal termination conditions which respond to flat error landscapes, i.e. if no further gradient was found which may lead to an other minimum.

To validate the method, we exemplarily generated artificial instances of reference data by executing simulations using sample model configurations. Optimizing against these artificial sets of reference data led to configurations which match the previously chosen sample configurations very closely. In the following we do not discuss this validation process in detail, but focus on the results achieved applying the reference data acquired in the real world.

4.1 Simple Range Model

The simple range model considers only one parameter–the distance between sender and receiver. Hence no parameter dependencies exist and we find a clear minimum in the error function as depicted in figure 4. The best results found in the search range from 10m to 500m was using a range of 105 m which leads to an error value of 11425. Note that us-

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5629
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5629

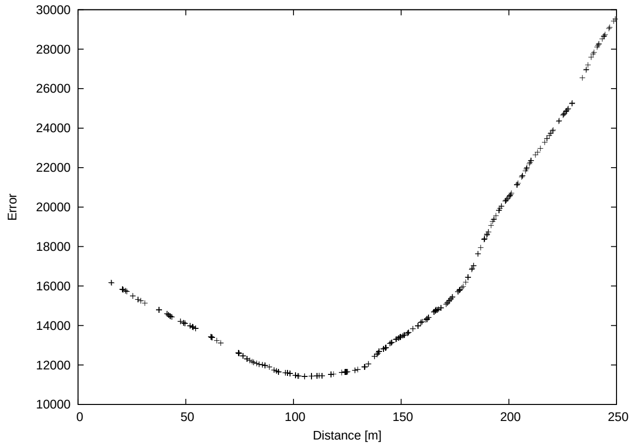

Figure 4: Error for simple range model plotted against distance

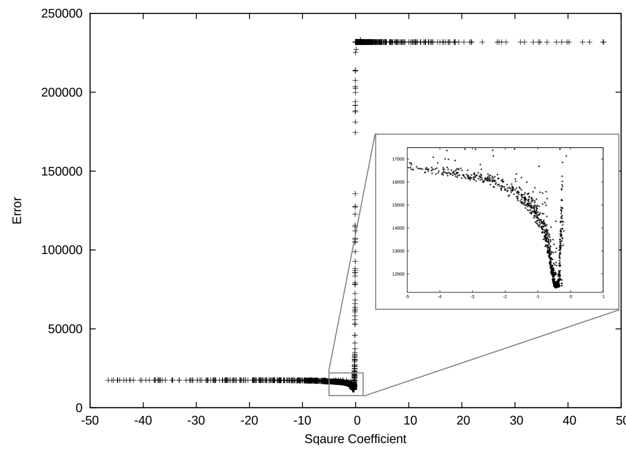

Figure 5: Error for polynomial model plotted square coefficient

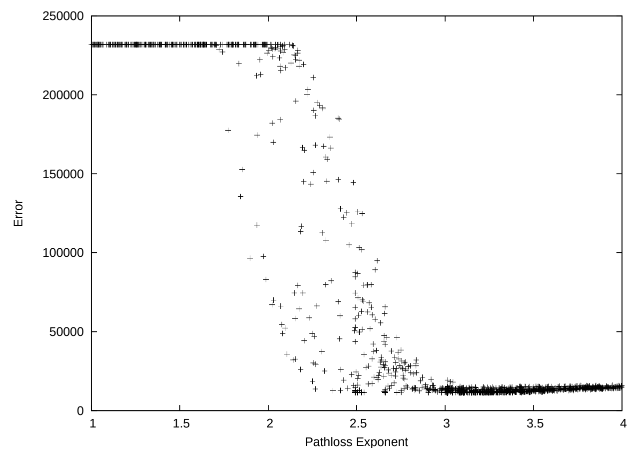

Figure 6: Error for standard model plotted against pathloss exponent

ing wrong range estimates leads to a comparably moderate increase of the error value—changing the range by ±10m does not change much. Compared to the number of approximately 17440 beacons successfully received during the real-world experiment, this implies an error of misplaced beacons of about 65.5% regarding the given error function!

4.2 Polynomial Model

The polynomial model introduces a set of seven parameters of which we found the square coefficient to be the most significant. As depicted in figure 5 reasonable results can be found around a value of approximately -0.4. The best configuration found has an error value of 11414 which also corresponds to about 65.5%. Setting the square coefficient smaller than -2.0 inhibits nearly all communication leading to an error near to the total number of beacons received during the real-world experiment. Setting it larger than 0 lets every node communicate with every other, hence the position of nodes does not matter anymore and nearly every beacon send is received successfully by all other nodes. Note that finding a comparably good configuration is actually depending on the use of an optimizer or a similar approach, since the parameters have no physical analogies and cannot be chosen e.g. based on experiences.

4.3 Standard Model

The most significant parameter regarding the standard model is the path loss exponent. Figure 6 shows that using pathloss exponents smaller than 2 leads to dramatic error values since again every node can communicate with every other. Setting it greater than 4 eliminates most communication. The best value found was 3.0.

Having a look at the other parameters plotted in figure 7 reveals a quite undifferentiated picture: There are no clear extrema or tendencies. This is a result to the correlation between these parameters. Changing e.g. the noise level may be compensated by using a higher transmitter power or by reducing the signal to noise ratio threshold. This can also be confirmed utilizing some enhanced features of the CMA-ES optimization algorithm.[1] This leads to the guess that this model might by overspecified regarding the set of parameters.

At the bottomline we slightly advanced accuracy compared to the above models, currently achieving an error of 65.0% with the best found solution. Nevertheless this is still far from being a satisfying result.

4.4 Urban Model

Similar to the standard model the path loss exponents are the most significant parameters for the urban model. This could have been expected, since this model is actually derived from the standard model. Considering figure 8 we find a greater dispersion of measurements than in figure 6 for the standard model. This is due to the additional parameters: reducing the free air angle φ gives more influence to the obstructed pathloss exponent hence enabling for results with $\alpha_{\text{free}} < 2$ and an error significantly smaller than the maximal error—and again we observe correlations between parameters.

Regarding the achievable error we obtain values of about

[1] We refrain from discussing this in depth since this needs deeper insights to the CMA-ES algorithm which is out of the scope of this contribution.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5629
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5629

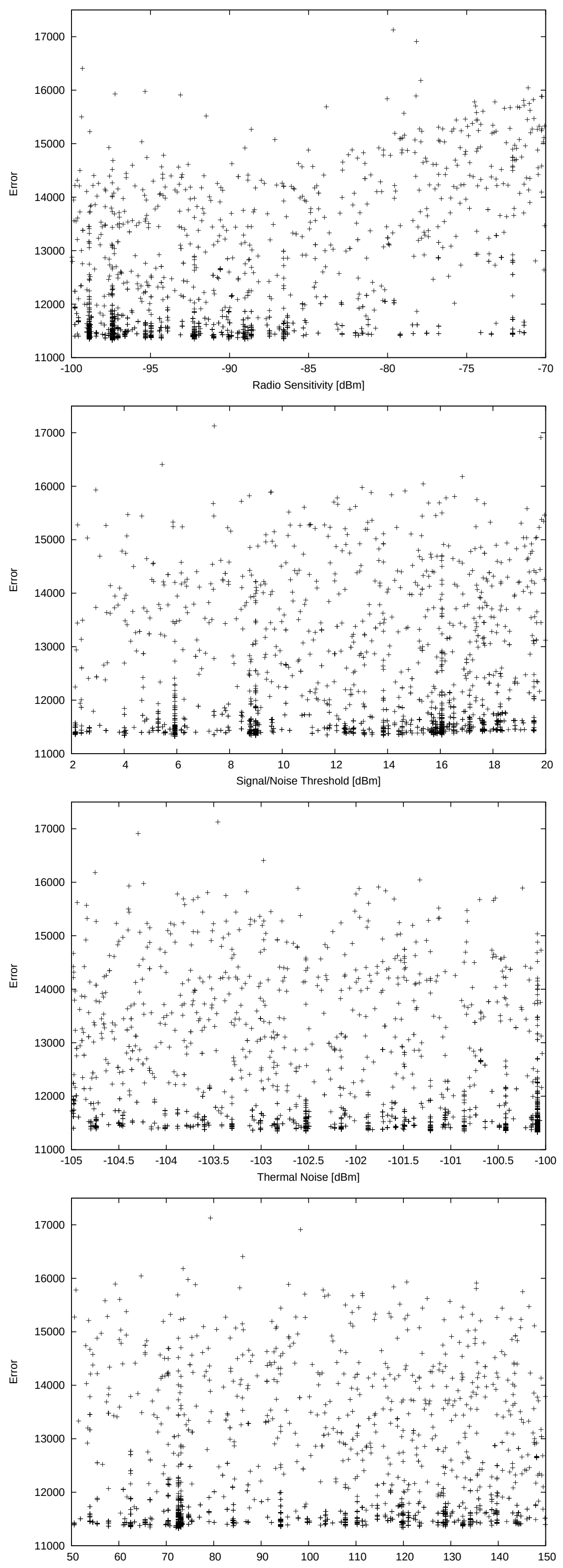

Figure 7: Less significant parameters of the standard model showing indifferent distributions (cutouts)

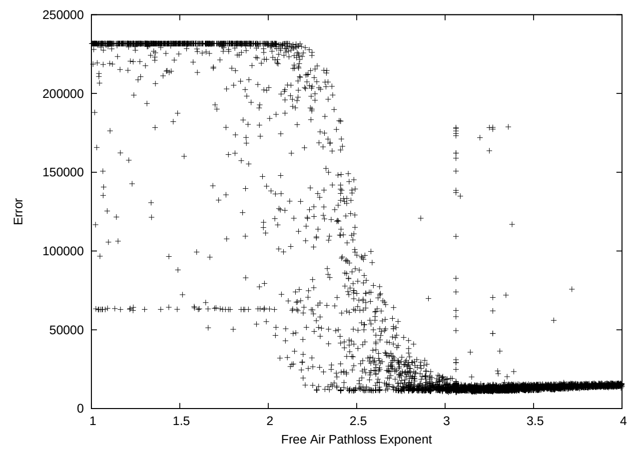

Figure 8: Error for urban model plotted against free air pathloss exponent

59.7%. This is an observably better result compared to the other models but again not satisfying.

4.5 Comparison

The achievable minimal error values are by trend lower for more complex models as shown in table 2. This behavior might have been expected, nevertheless the simple range, the polynomial and the standard model are quite close to each other.

Besides the accuracy the actual duration of executions of simulations is relevant—especially regarding large networks to be simulated. To compare the discussed models, we ran the same replay simulation as used in the optimization process 1000 times for each model. The configuration chosen for all simulations was the best configuration found respectively. From this we took the average rate of seconds simulated to seconds of execution time. Note that this value comprises also execution times of the application software and non-radio models. Hence these results must not be interpreted as absolute values but only relatively among the models. Furthermore the speedup regarding the studied radio models itself is even larger than the values found, since non-modell related parts (e.g. basic simulator modules or the application) add an offset which is expected to be approximately the same for all of the presented experiments.

For comparability reasons we configured each model with the regarding most accurate configuration found, as mentioned above. Note that this does only lead to an approximately equal number of executions of each radio model since the number of beacons received is slightly different. Nevertheless this configuration constitutes the settings to be used for further simulations. Hence this is in a way a normative operation condition.

Table 3 shows the actual differences between the models. As expected a more complex model implies more calculations and hence lower performance of the simulation executions. Nevertheless it is remarkable that the standard model is significantly slower than the simple range model while not providing a particular benefit regarding the accuracy. On the other hand using the urban model comes at nearly no (performance) costs compared to the standard model but provides an observable advance regarding the accuracy.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5629
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5629

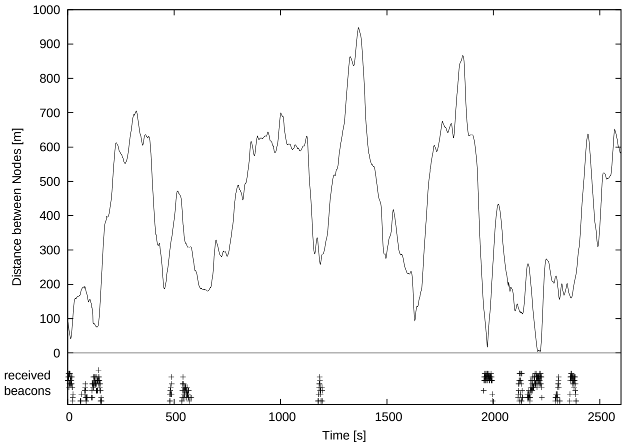

Figure 9: Example for distance between two nodes during the real-world experiment and the according number of beacons received per seconds.

4.6 Interpretation

To emphasize it once more: all presented results are based on the above defined error function and the reference data recorded in a real-world experiment. Assumed that both are suitable, we can state that no model achieves a satisfying accuracy for the given scenario—even not with a near optimal configuration. Do more enhanced and therefore more complex models as e.g. the urban model presented in [10] lead to better results? This cannot be answered on the basis of the presented studies—although it seems to be feasible. To look a bit closer into the peculiarities of the given urban scenario, figure 9 exemplarily depicts the distance of two nodes A and B during the real-world experiment (upper part of plot). The distance plot is accompanied by the number of beacons node A received from node B per second. Between e.g. 0 and 200 seconds or from 1900 to 2400 seconds communication takes place as expected: as the nodes approach each other communication starts, when the distance grows again the connectivity breaks away. Even the larger distance at approximately 2100 seconds is reflected by a gap in the recorded sequence of beacons. Now take a look at the situations at about 1200s and at about 1700s. In the first case the nodes have a distance of more than 250m and communication takes place. In the second case they get closer than 100m but no communication takes place. Hence every simply range or path loss based model will not be able to match both cases with the same configuration. Models considering more aspects of the real world, like buildings, landscape, reflection or artificial noise sources, may be a solution. This will be a part of future studies. Nevertheless most of these models are not topography-independent anymore and hence need to be specifically adapted to a given environment—possibly based on further intensive real world measurements.

It is common to all presented models, that there are regions in the space of possible configurations, which lead to very bad results in terms of the used error function. As already suggested in 4.2 configurations from these regions allow no or too much communication between the nodes. Especially the latter group will possibly produce trivial conditions for protocols and application software that will be simulated on basis of accordingly configured radio models. This may lead to simulation results for application software and protocols, that are too optimistic, finally resulting in surprisingly bad behavior in real world deployments.

Trapping into such a configuration is more probable for models that have larger parameter sets and more sophisticated interrelations between the parameters. The simple range model for example can be configured comparatively easy, by assuming a reasonable radio range. The other modules in contrast show much stronger dependencies between parameters, as for example mentioned in 4.3. These relations are often not obvious. Good documention or analyzing source code may be a solution. Alternatively using an optimization process as described can also help to identify these critical parameter settings.

Besides model related issues, these results show how crucial a suitable configuration of the model is: even small misconfigurations may lead to significantly worse accuracy compared to the use of a simpler model.

5. RELATED WORK

The accuracy of a simulation according to the real world is the most crucial issue regarding its applicableness for a (productive) development and/or deployment process. Newport et. al. present in [11] an evaluation of routing protocols first in an artificial real world deployment and second in a simulation environment using different radio models. Besides clearing up some common assumptions often made for simulation, they show how sensitive routing results are to model inaccuracies. Nevertheless they chose radio parameters on some kind of experiences without further justification.

Liu et. al. [9] argue for so called "direct-execution" which actually means to run the same code in simulation and deployment. Furthermore they point out the correlation between routing results and the chosen path loss exponent for a radio model.

Without relating their results to real world experiments Cavin et. al. [4] and Takai et. al. [15] show the influence of the simulated physical layer to the routing layer by comparing different simulators with different radio models leading to dramatically different results. Especially Takai et. al. discuss physical details of the physical layer to be reflected in the models. This is somewhat contrary to the polynomial model discussed above trying to omit physical analogies.

An overview regarding the credibility of MANET Simulations also discussing the comparability of results can be found in [3].

6. CONCLUSION

As shown in section 4 the differences between the models regarding the accuracy are smaller than it might have been expected. Inadequate parameterization in contrast has a significant effect to the results. Given the case that we have no reference data and no optimizer at hand, this possibly raises the question if it is actually wise to use complex models without being very sure about the correct configuration or if it is better to stick to simple models. These might be configured more intuitively and/or on the basis of simple real world observations, as e.g. some radio range in meters.

On the other hand, if we have the possibility to use reference data and optimization, models do not necessarily need to take physical measures as parameters, since values can be

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5629
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5629

determined as described in this contribution.

Regarding the complexity of models, the results show that more enhanced distributions of the reception power and introducing bit error models have a minor influence under the described scenario and error function. Qualitative changes as introduced with the urban radio model seem to be more promising. Hence investigating topography-oriented models in the future might lead to solutions with smaller absolute errors.

7. REFERENCES

[1] INET framework with ad-hoc routing protocols, available at `http://webpersonal.uma.es/~AARIZAQ/`.

[2] The official guide to learning OpenGL, available at `http://www.glprogramming.com/red/`.

[3] T. R. Andel and A. Yasinac. On the credibility of manet simulations. *Computer*, 39(7):48–54, 2006.

[4] D. Cavin, Y. Sasson, and A. Schiper. On the accuracy of manet simulators. In *POMC '02: Proceedings of the second ACM international workshop on Principles of mobile computing*, pages 38–43, New York, NY, USA, 2002. ACM.

[5] N. Hansen, S. Muller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11, 2003.

[6] J. Koberstein and N. Luttenberger. Data-centric wireless sensor network programming methodology. In *3rd International Conference on Networked Sensing Systems (INSS)*, 2006.

[7] J. Koberstein, H. Peters, and N. Luttenberger. Graph-based mobility model for urban areas fueled with real world datasets. In *1st International Workshop on OMNeT++*, Marseille, France, March 2008. ACM.

[8] J. Koberstein, S. Witt, J. Boetel, and N. Luttenberger. Multipurpose application of evolutionary algorithms for wsn parameterization—a case study. In *submitted to International Conference Information Processing in Sensor Networks (IPSN)*, San Francisco, USA, April 2009. ACM/IEEE.

[9] J. Liu, Y. Yuan, D. M. Nicol, R. S. Gray, C. C. Newport, D. Kotz, and L. F. Perrone. Simulation validation using direct execution of wireless ad-hoc routing protocols. In *PADS '04: Proceedings of the eighteenth workshop on Parallel and distributed simulation*, pages 7–16. ACM, 2004.

[10] R. Nagel and S. Eichler. Efficient and realistic mobility and channel modeling for vanet scenarios using omnet++ and inet-framework. In *Simutools '08: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, pages 1–8, 2008.

[11] C. Newport, D. Kotz, Y. Yuan, R. S. Gray, J. Liu, and C. Elliott. Experimental evaluation of wireless simulation assumptions. *SIMULATION: Transactions of The Society for Modeling and Simulation International*, 83(9):643–661, September 2007.

[12] OpenWrt. *available at `http://www.openwrt.org`.*

[13] T. S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 1996.

[14] M. Segal and K. Akeley. *The OpenGL Graphics System: A Specification*. December 2006.

[15] M. Takai, J. Martin, and R. Bagrodia. Effects of wireless physical layer modeling in mobile ad hoc networks. In *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 87–94, New York, NY, USA, 2001. ACM.

[16] A. Varga. The OMNeT++ discrete event simulation system. In *Proceedings of the European Simulation Multiconference (ESM)*, 2001.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5629
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5629