Effects of Random Number Generators on V2X Communication Simulation

Robert Protzmann[1], Björn Schünemann[2], and Ilja Radusch[2]

[1] Fraunhofer FOKUS, Berlin
Automotive Services and Communication Technologies
robert.protzmann@fokus.fraunhofer.de
[2] Technische Universität Berlin
OKS / Daimler Center for Automotive IT Innovations

Abstract. Detailed simulation models have to incorporate random effects. Since the generation of randomness is subject to several shortcomings, this needs to be considered for the setup and evaluation of simulations. On the basis of well-known metrics for the domain of V2X communication we will evaluate the influences of differently generated random sequences on the simulation. We will show that it is important to pay attention to avoid skewed results caused by random number generation and ensure the statistical relevance of the simulation series. It can be stated that well established random number generators are suitable. Meaningful simulation results rely rather on a sufficient number of simulation runs which in turn will depend on the applied models.

Keywords: RNG, V2X simulation verification, VSimRTI

1 Introduction

Vehicle-to-X communication is an up-and-coming technology for improvement of road safety, traffic efficiency, and infotainment applications [1]. At the present time, already field operational tests are performed to investigate the behavior of V2X communication based applications in reality. However, a good portion of research is done with the help of simulations.

For credible statements about the application behavior, it is important to ensure simulations that are able to produce qualified results. Much effort is spent for the development of preferably detailed models that include the according realworld properties. E.g. in [2] a V2X simulation environment was extended with advanced communication models. Since many processes in nature can only be modeled as random, the computational models need to rely on generators for randomness. In the V2X simulation context, examples for models using Random Number Generators (RNG) are e.g. the vehicle movement models in a traffic simulator or the radio propagation models in a communication simulator. The expected value of these models can be described to follow a known probability distribution. During the simulation run, a RNG then delivers a certain representation of the distribution or more concrete a sequence of individual samples as a

subset of the whole sample space. At this point a first important issue arises. A sufficient coverage of the sample space and finally a result close to the expected value needs multiple samples to be generated which means in turn that multiple simulation runs need to be performed. Depending on the law of large numbers, the number of simulation runs tends to infinite for a perfect result. However, due to practical reasons a reduction of simulation runs is always desirable to save computing costs and time. A second issue is to avoid skewed results. This can be ensured for individual simulation models, as the results can be cross compared with the according probability distributions. When finally the application behavior is evaluated, all aspects of the simulated models, protocols and application algorithms play together. Hence, it is not trivial to trace the aggregated result back to a certain influence or even to be sure that the whole spectrum of the sample space is represented. This may lead to wrong statements or wrong optimizations of the application algorithms, merely due to an insufficient setup of the simulation series.

In this paper we present an approach to ensure correct results in simulations based on randomness. The central question is how strong the used RNGs do affect the results. If the choice of the RNG itself or the parameterization of the RNG would have a substantial impact on the results, it would mean that great care is advisable when setting up simulations with random components. It is undesirable that RNGs affect the functioning of the simulated processes even more than certain details in the models themselves. Furthermore, we show a direction how the results converge with different numbers of simulation runs.

The paper is structured as follows. In the next Section 2 we give a short overview of important properties of RNGs and introduce three examples which are used in this work. In Section 3 we present our simulation setup and give additional details on the randomness-based models and the metrics to measure the impact of randomness. The results are given in Section 4. The findings are resumed and the paper is concluded in Section 5. Finally, we state the future work on the topic in the outlook in Section 6.

2 Random Number Generators

For simulation purposes two features of RNGs are demanded, 1) the generation of truly random and statistically credible results with a limited number of simulation runs and 2) the reproducibilty of results, which is important for debugging and comparison [3]. Due to the latter requirement only deterministic RNGs can be used. These RNGs are based on deterministic (software) algorithms and deliver, when the starting situations are equal, also equal results. As the generated sequences are not truly random anymore, they are called pseudo-random. The dimension of incorporated errors compared to true randomness depends on the quality of the used generator and its according initialization. The quality of a RNG can be described first by the distribution of the random samples which should be uniform and the individual samples should be statistically independent [4]. The statistical independence of the samples is demanded so that the

aggregated result converges to the expected value according to the law of large numbers [5]. Several simulation model implementations are based on probability distributions which are not necessarily uniform. With the method of the inverse transform sampling it is possible to generate any distribution given its cumulative distribution function (cdf) out of the uniform distribution. If the distribution is not uniform, certain number patterns are generated more often. This effect is known as falling in the planes [6]. The second characteristic to measure suitability of RNGs is the period of the random sequence, before it starts again. This limit is a result from the fact that all deterministic RNGs are finite state machines. However, a long period not necessarily ensures a high quality, but a short period is often a problem.

Actually there exist a quite high number of suitable RNGs, but in our work we have concentrated on the Linear Congruential Generator (LCG) [7], the Mersenne-Twister [8], and the Blum-Blum-Shub Generator (BBS) [9].

The importance of the **Linear Congruential Generator** stems from the fact that it is very well established for a long time. It is part of the runtime libraries of the GNU Compiler Collection and the Java SDK and consequently it is directly available for the simulation program code. The LCG is based on the linear modulo operation, described with the following Equation 1.

$$x_i = (a * x_{i-1} + b) \bmod m \tag{1}$$

A typical problem of the LCG is the occurrence of the falling-in-the-planes effect which means that the generated numbers are not distributed perfectly uniform. Furthermore, depending on the platform, the period as one important measure for the quality of RNGs, may be quite short, starting at an order of 2^{32} (or respectively 10^9 to 10^{10}) for 32 bit systems. The LCG in Java.util possesses a period of 2^{48} (ca. 10^{14}).

The **Mersenne-Twister** is a newer generator which delivers random sequences of very high quality [8]. Hence, it is very well suited for simulation purposes and already incorporated in various simulators as SUMO, OMNeT++ and JiST/SWANS. The components of the Mersenne-Twister are given in the following recurrence Equation 2. Here, the $|$ operation is a just a concatenation of the upper and lower part. The multiplication with the matrix A performs the twist transformation. Finally, $\oplus$ is the bitwise XOR which is actually an addition with modulo two.

$$x_{k+n} = x_{k+m} \oplus (x_k^u | x_{k+1}^l) A \tag{2}$$

The MT19937 is a specific implementation of the Mersenne-Twister that features an outstanding period of 2^{19937} (10^{6001}). Moreover, it generates an almost uniformly distributed sequence and passes most of the mathematical randomness tests. Due to these advantages it is commonly used for large scale simulations, e.g. according to the Monte-Carlo approach.

The **Blum-Blum-Shub** was designed for cryptographic systems. For this purpose it needs to satisfy slightly different requirements, e.g. that the random

sequence cannot be predicted. The unpredictability arises from the quadratic residue in the modulo-operation as outlined in Equation 3.

$$x_{i+1} = x_i^2 \bmod n \tag{3}$$

Whereas n (and in turn the period of the BBS) is defined as the product of two prime numbers p, q of the form $4k + 3$. For a sufficient period these primes need to be very high which in turn slows down the simulation performance. However, this fact allows different configurations for a comparison of a period that should be still sufficient for our simulations with a bad configuration.

3 Simulation Study

In our investigation we place the emphasis on the dependency of the simulated communication behavior on different RNGs. We simulate a single-access scenario and a multi-access scenario where we apply the already introduced RNGs LCG, Mersenne-Twister, and BBS with an appropriate and inappropriate configuration. We analyze three communication metrics to compare the influences of the different RNGs. To avoid side-effects we have configured the models for traffic simulation in our setup to simulate exactly the same deterministic behavior in each V2X scenario. For all V2X simulations we use the simulation environment VSimRTI [10]. VSimRTI couples discrete event simulators from different fields. In our concrete case we use JiST/SWANS for the simulation of the communication stack. Furthermore, we use the application simulator VSimRTI_app which usually embeds the logic of V2X applications, but can also be used to generate only a certain data load for the communication simulator. At last we use the traffic simulator SUMO for the vehicle movements.

3.1 Analyzed communication models with randomness

In our setup, the **IEEE 802.11p MAC layer** is the first module which incorporates a random component. When the Distributed Coordination Function (DCF) of the sender senses the radio channel as occupied upon a transmission attempt, it needs to listen to the channel to defer the transmission until the channel is free again. But when during an ongoing transmission more than one sender are waiting for the moment when the channel is free again and would directly start their own transmission, a packet collision would occur. Thus, the DCF provides the backoff procedure to minimize the probability of a concurrent transmission. For the backoff procedure one random integer value is selected out of the interval of the contention window. This means e.g. for the case of broadcasting where the initial contention window is 31 slots, every node selects a random value between 0 and 30. According to the back backoff procedure, the node has to sense the channel as free for the duration of the slot time and then decrement the previously selected backoff timer. When the backoff timer reaches the value of 0, the node can start its transmission. For a fair prioritization of the different senders, the used random variable needs to be distributed uniformly.

Furthermore, the **IEEE 802.11p PHY layer** model uses a random component to simulate a packet error rate (PER). In contrast to the random component in the MAC layer it is not a feature of the protocol according to the IEEE 802.11p standard specification, but it is more a characteristic of the simulation model itself. A packet error exists in the situation when PHY layer was able to sense at least a received signal, but the signal, or more concrete the SNIR (Signal to Noise and Interference Ratio), is to weak that the signal processing chain in the PHY is not able to decode the packet correctly. As a signal processing chain would be computationally too time-consuming, an efficient abstraction is implemented. The model calculates the PER out of the SNIR, the symbol modulation and the packet length. The PER can have a value between 0 and 1. To determine if the packet is eventually received or not, a comparison is performed with a random value which is sampled uniformly between 0 and 1. If the PER is lower than this random value, the packet is regarded as successfully received.

At last, more realistic models for the **Radio Channel** also incorporate fading effects. We use the well-established Rayleigh and Rice Fading Models for our investigation. These fading models already incorporate probability distributions which are not uniform. As the Rayleigh Fading is mathematically described by the integral cdf of the Rayleigh distribution, the inversion sampling method can be applied for the concrete implementation of the Rayleigh model. Hence, the implementation can be reduced to one random sample of the uniform distribution. The Rice Fading is mathematically based on a Bessel function which needs to be approximated numerically in the implementation and uses two random samples for one calculation.

3.2 Analyzed metrics for simulation

The **Backoff Timer** is linearly dependent on the calculated random value from the according call in the MAC module. Thus, the quality of the applied RNG should be directly assessable on the basis of the distribution of the backoff timer. In the single-access case, the sending node is always able to directly transmit its packet and no further backoff mechanism is needed. Hence, this metric is only evaluated in the multi-access case.

The **Channel Access Delay** is the time from the moment where a packet arrives in the MAC layer to the moment where it can be send to the channel. It depends in the single-access case merely on a possible simulated processing delay for the layer, yet our model considers no processing time. In the multi-access case it also depends on the selected backoff timer and furthermore on the backoff timers of the other simulated nodes. E.g. when two nodes wait for a free channel and then select their backoff timer, the node with the longer backoff timer needs to additionally defer its transmission until the node with the shorter backoff timer has finished its own transmission. The channel access delay is only evaluated in the multi-access case.

The **Packet Delivery Rate** shows the influences of the random numbers on the simulation of the radio channel fading. The PDR for a single packet can be either 0 or 1. So it needs to be evaluated in a statistical context, whereas

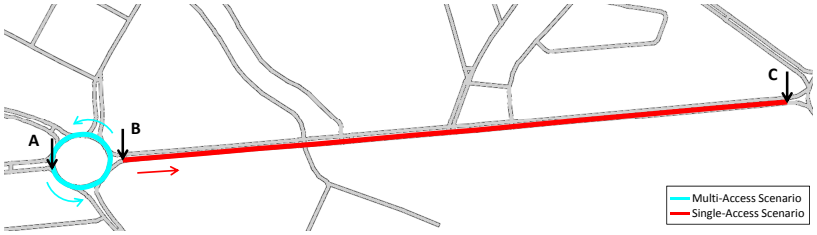

Fig. 1: Simulated scenarios multi-access (circle), single-access (line)

the arithmetic-mean aggregation of all packet sending attempts in the simulated time frame delivers the most significant result. Especially in the single-access case the PDR depends directly on the applied fading models which affect only the signal term of the SNIR for the packet error model in the PHY layer. When no other sender exists, the noise and interference term is constant and the medium access function works transparently. In the multi-access case it is not trivial to trace the result back to one single influence. Hence, this metric is only evaluated in the single-access scenario.

3.3 Simulated Scenarios

The two simulation scenarios for the evaluation are located in the Strasse des 17. Juni in Berlin (Figure 1). In the multi-access scenario especially the backoff RNG is considered. 20 vehicles communicate concurrently via periodic broadcasts with a frequency of 10 Hz. The nodes have to sense the medium and apply their backoff mechanisms to avoid possible packet collisions. However, with the given number of senders and the configured sending frequency the radio channel is still free enough that every packet can be successfully transmitted. The hidden terminal problem and consequently a malfunction of the backoff procedure does not exist in this scenario, as all vehicles are always in communication range in the cyan colored traffic circle, which has a diameter of 150m. The vehicles are introduced in the simulation with a time delay at position A and move around the circle again to point A (Fig. 1). This round is repeated until the simulation ends.

The single-access scenario is intended to be used primarily for the evaluation of fading RNGs. It consists of a stationary node, located at point B which is the receiver and a sender which starts also at point B near the receiver and then moves away on the red route with a constant speed of 10 m/s towards point C (Figure 1). The final point C of the sender is nearly 1.9 km away from receiver. Consequently the route is long enough up to the maximum distance where the connection between both nodes is finally lost. The sender emits messages with a constant period of 10 Hz. In this way communication measurements with a resolution of 1 m can be captured.

The following calculation should clarify the maximum number of random samples needed for the simulation. For every sent packet one backoff timer needs to be generated. For every packet a fading channel needs to be simulated (in the

case of Rice Fading by the use of 2 random samples). Finally, when the signal energy at receiver side is at least high enough to sense the packet, the PHY layer PER is calculated. For a traffic-efficiency scenario where $n = 400$ vehicles with a sending frequency of $f = 10Hz$ are investigated over one hour of simulation time ($t = 3600s$), the result would be the following (Eq. 7), whereas N_{BT} is the number of random samples for the backoff timer, N_{FC} for the fading channel, and N_{PER} for the packet-error check.

$$N_{BT} = n \cdot f \cdot t = 14400000 \tag{4}$$

$$N_{FC} = 2 \cdot n(n-1) \cdot f \cdot t = 11491200000 \tag{5}$$

$$N_{PER} = n(n-1) \cdot f \cdot t = 5745600000 \tag{6}$$

$$N = N_{BT} + N_{FC} + N_{PER} = 1.72512 \cdot 10^{10} \tag{7}$$

When the period of a RNG is considered as quality measure, it can be stated that it should be in the order of 10^{11} or higher to be appropriate for the scenario. The LCG and certainly the Mersenne Twister should be appropriate. The BBS in the good configuration should also exhibit a period higher than N.

4 Simulation Results

The results for three metrics are presented in the following. The backoff timer is selected, because it is known that the samples need to be distributed uniformly. Furthermore, the channel access delay is selected as it already includes the protocol behavior of the IEEE 802.11p MAC layer. Finally, the PDR is presented as a metric where the required distribution is also known in advance, but it is not uniform. Moreover, the applied models for Rayleigh and Rice Fading rely on a different number of random samples for one calculation.

4.1 Backoff-Timer

Two comparisons are carried out for the backoff timer. First, the result of each individual RNG is compared with the uniform distribution, which is important to facilitate a fair prioritization of all senders in the system. Second, the BBS is compared with the good and the bad parameterization to get an overview, if it is possible at all to set up the RNG in an insufficient way. In Figure 2 one histogram for each RNG is displayed. The histograms directly allow a visual examination of the probability distributions.

Generally, the LCG, the Mersenne-Twister, and the BBS (good) produce a fairly uniform distribution. In comparison with the first two generators, it can be noticed that distribution for the BBS (good) is slightly more uneven. The second comparison of the Blum-Blum-Shub shows that the BBS (bad) is not able to generate a uniform distribution at all, although the period of this RNG should be able to deliver more than the required number of different samples. This may be traced back to the fact that the random integer number (in the interval

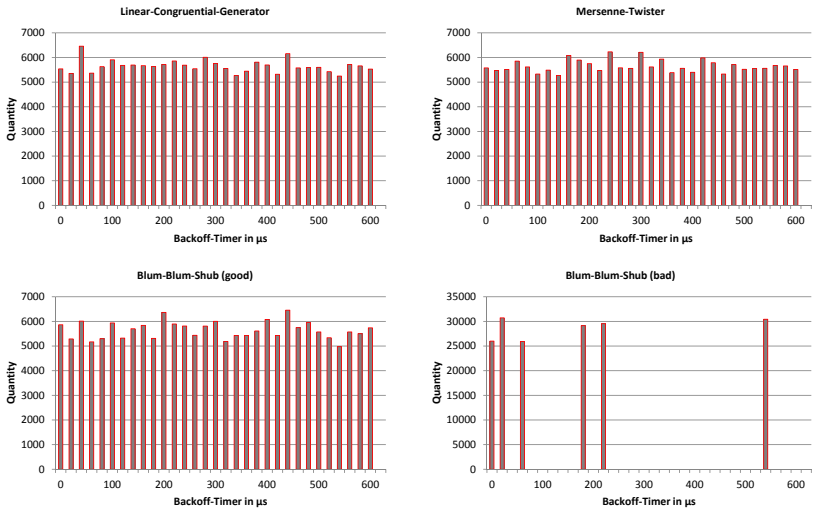

Fig. 2: Distribution of the backoff timer, different RNGs, 30 simulation runs

between 0 and 30) is generated by the use of a real number sample (between 0 and 1) and a subsequent real to integer conversion in the implementation. This conversion may intensify the effect that not the whole sample space is covered.

Additionally, we have applied the Kolmogorov-Smirnov test to measure the goodness of fit [11]. The following table 1 contains the result for the K-S test D-statistic, which indicates the maximum deviation of the empirical distribution generated by the RNGs and the hypothetical (uniform) distribution. A perfect fit would result in the D-value of 0. This test supports the previous finding that the LCG, Mersenne-Twister and BBS (good) are very close to the uniform distribution, while the BBS (bad) involved insufficient results.

RNG	**K-S test D-statistic**
LCG	0.006465
Mersenne-Twister	0.006071
Blum-Blum-Shub (good)	0.006791
Blum-Blum-Shub (bad)	0.435661

Table 1: K-S test D-statistic for the evaluated RNGs

The metric of the backoff timer especially demands a uniform distribution of random numbers. Generally, a collision occurs when two nodes select the same backoff timer and start their transmission at the same time. When certain random values are generated more frequently due to the RNG, this issue is over amplified in the simulation and leads to skewed results. Thus, a wrong selection and parameterization of the RNG builds a wrong basis of the whole simulation.

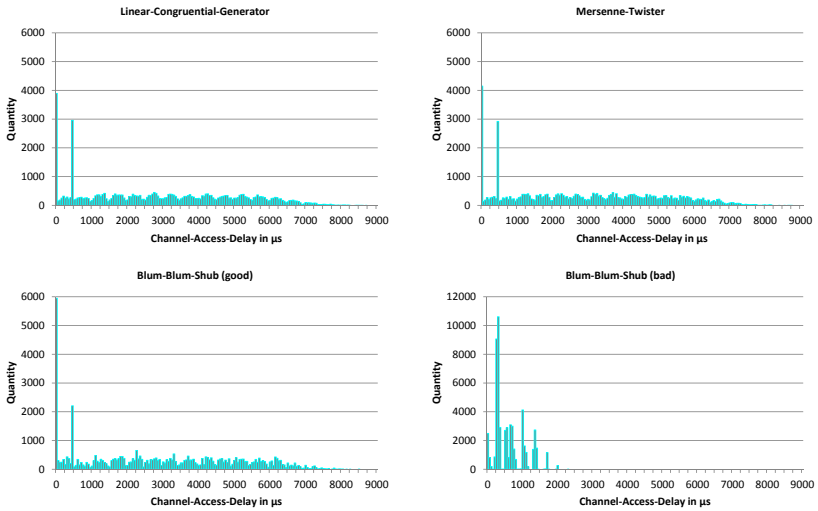

Fig. 3: Distribution of the channel access delays, different RNGs, 30 simulation runs

4.2 Channel Access Delay

The channel access delay already incorporates the protocol characteristics of the IEEE 802.11p MAC layer in the situation when multiple nodes compete for the radio channel. The diagrams (Figure 3) show that it is not trivial to compare the results with a known distribution.

For the Mersenne-Twister, the LCG and the BBS (good) there exists one high peak for the lowest delay which represents the case when one sender has selected the lowest backoff timer of all competing nodes and can be the first sender. The second highest peak may indicate nodes which have selected a higher backoff timer have to defer their transmission to be the second sender, but it also includes nodes which are the first sender and have selected a higher backoff timer and no other sender is competing for the access. The BBS (bad) again produces a different result compared to the other configurations.

The results show that errors from the backoff timer emerge to have an effect on the channel access delay. Already for this metric it is difficult to acknowledge the correctness. This means in turn that for key performance indicators on application layer it would be more complicated to prove the result. Hence, it is recommended to record metrics as the backoff timer for a direct comparison to ensure the correct simulation setup.

4.3 Packet Delivery Rate

The PDR evaluation first aims to compare how models with different number of random samples influence the result. Second, a comparison is drawn where only few simulation runs are evaluated, because few simulation runs are desired of practical reasons to save time.

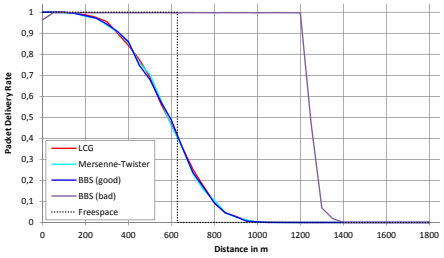

Fig. 4: PDR for Rayleigh Fading, different RNGs, all 30 simulation runs

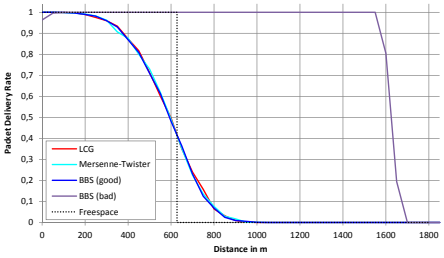

Fig. 5: PDR for Rice Fading, different RNGs, all 30 simulation runs

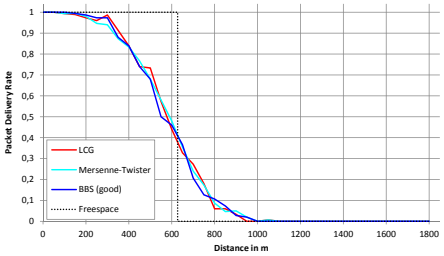

Fig. 6: PDR for Rayleigh Fading, different RNGs, only 3 simulation runs

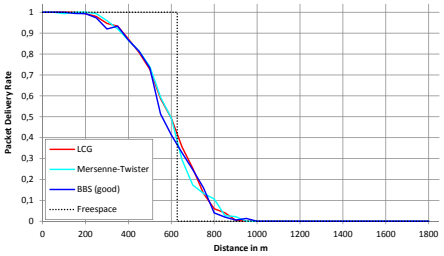

Fig. 7: PDR for Rice Fading, different RNGs, only 3 simulation runs

The results are presented for the arithmetic mean of the PDR for 30 simulations with Rayleigh (Fig. 4) and Rice Fading (Fig. 5). Additionally, the PDR graph for the Freespace Model is included as a reference. In both figures it is observable that the results for the LCG, the Mersenne-Twister and the Blum-Blum-Shub (good) are almost identical. Furthermore, a comparison with the Freespace reference shows that the graphs of the PDRs have the correct trend and start to decline at the right distance. The Blum-Blum-Shub with the bad configuration exhibits a different and certainly wrong behavior. This behavior is even worse for the evaluation of Rice Fading and can probably be traced back to the fact that Rayleigh Fading uses only one random call for one calculation, while Rice Fading uses two random realizations.

In a second evaluation again Rayleigh (Fig. 6) and Rice Fading (Fig. 7) was analyzed but only 3 simulations where selected for the PDR aggregation. Assuming that one RNG possesses a lower quality regarding the random number distribution, it would show a less smooth run. As the Blum-Blum-Shub (bad) already showed a wrong result in the first evaluation, it was omitted in the following diagrams. The graphs display that no RNG produces an especially uneven result. Moreover a comparison of Rice and Rayleigh Fading indicates that the application of two random realizations in the Rice model leads to an already smoother result. This means that the number of required simulation

runs for a satisfying coverage of the sample space depends more on the applied simulation models than on the applied RNG.

To sum up the results, we see that the LCG and the Mersenne-Twister exhibit nearly identical results for all evaluated metrics. The sufficiently configured Blum-Blum-Shub shows similar results for the PDR. The distribution of the backoff timer seems to be not completely uniform, but one can assume that for the medium scale scenario the quality of the randomness is sufficient. Yet, there also exist RNGs which lead to a completely wrong outcome, as the BBS with the configuration of insufficient p and q values. In this case the period was although selected to meet at least the requirement for number of random values, but the sequence was not uniformly distributed. The channel access delay is presented as a metric where the comparison between the different RNGs was still possible, while it is not straightforward to assess the correctness as it incorporated already several influences.

Aside from this, it is important to use known random seeds. With equal random seeds the simulation behavior can be reproduced exactly for debugging purposes. With different seeds in the multiple simulation runs the result converges to the expected value according to the law of large numbers. Whereas the number of simulation runs for a statistical significance depends more on the used models than on the used RNGs.

Finally, the approach to record and evaluate metrics with an expected result is deemed to be successful to ensure a correct simulation setup. In our case the backoff timer and the PDR are suitable, because it is known that they are demanded to follow the uniform or respectively the Rayleigh and Rice distribution. In this way it is also possible to check the statistical relevance.

5 Conclusion

In this paper we have evaluated the influences of RNGs on the simulation results of V2X communication scenarios. We have set up two scenarios for a single-access and a multi-access case. The focus of our investigations was put especially on the communication metrics of the packet delivery rate, the distribution of the backoff timer, and the channel access delay. In the simulations we have applied different RNGs, 1) the LCG which is the standard RNG in many compilers for famous programming languages as C/C++ and Java, 2) the Mersenne-Twister which is known to deliver especially high quality random sequences and for comparison the Blum-Blum-Shub in a 3) good as well as in a 4) bad configuration.

We have seen that it is straightforward to assess the behavior affected by the RNG for certain metrics where a dedicated result is expected in advance. This was the case for the backoff timer and the PDR. The metric of the channel access time was not trivial to assess as it incorporates already several aggregations. Key performance indicators for applications depend even on more influences.

For the actual comparison of the RNGs we have seen, that the LCG and the Mersenne-Twister both delivered very good results and the BBS in the good configuration is still sufficient. In contrast, the BBS in the bad configuration

always delivered completely wrong results. This means that indeed care needs to be taken for the correct choice of the RNG and its parametrization.

A comparison of the number of simulation runs to achieve a statistical significant result lead to the finding that it depends more on the simulation model which relies on the random samples than on the RNGs themselves.

Finally, as specific lessons learned it can be stated that known metrics should be recorded and evaluated to ensure the correctness and the sufficient number of simulation runs to cover the whole sample space for a statistically significant result.

6 Outlook

As already mentioned, an evaluation of known metrics along the way is important to ensure meaningful results for the actually investigated key performance indicators. Hence, in future simulation studies we want to introduce further statistical tests for the verification to our workflow. The presented K-S test was applied to check the correct distribution of the analyzed metrics. In addition, there exist other suitable tests as the runs-test according to Wald-Wolfowitz for the check of the statistical independence of the random samples.

References

1. Hartenstein, H., Laberteaux, K.: VANET: vehicular applications and internetworking technologies. Volume 1. Wiley Online Library (2010)
2. Protzmann, R., Mahler, K., Oltmann, K., Radusch, I.: Extending the v2x simulation environment vsimrti with advanced communication models. In: ITS Telecommunications (ITST), 2012 12th International Conference on, IEEE (2012) 683–688
3. Law, A.M.: Simulation Modeling and Analysis. 4rd edn. McGraw-Hill Higher Education (2007)
4. Matsumoto, M., Saito, M., Haramoto, H., Nishimura, T.: Pseudorandom number generation: Impossibility and compromise. J. UCS **12**(6) (2006) 672–690
5. Fishman, G.S.: Monte Carlo: concepts, algorithms, and applications. Springer series in operations research. Springer (1996)
6. Marsaglia, G.: Random numbers fall mainly in the planes. Proceedings of the National Academy of Sciences **61**(1) (1968) 25–28
7. Entacher, K.: A collection of classical pseudorandom number generators with linear structures: advanced version. (2000)
8. Matsumoto, M., Nishimura, T.: Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation (TOMACS) **8**(1) (1998) 3–30
9. Blum, L., Blum, M., Shub, M.: A simple unpredictable pseudo-random number generator. SIAM Journal on computing **15**(2) (1986) 364–383
10. Schünemann, B.: V2x simulation runtime infrastructure vsimrti: An assessment tool to design smart traffic management systems. Computer Networks (2011)
11. Massey Jr, F.J.: The kolmogorov-smirnov test for goodness of fit. Journal of the American statistical Association **46**(253) (1951) 68–78