An Open Source Channel Emulator for Non-Terrestrial Networks

Tim Düe, MohammadAmin Vakilifard, Carsten Bockelmann, Dirk Wübben, Armin Dekorsy
Department of Communications Engineering, *University of Bremen*, Bremen, Germany.
{duee, vakilifard, bockelmann, wuebben, dekorsy}@ant.uni-bremen.de

***Abstract*—Non-Terrestrial Networks (NTNs) are one of the key technologies to achieve the goal of ubiquitous connectivity in 6G. However, as real world data in NTNs is expensive, there is a need for accurate simulations with appropriate channel models that can be used for the development and testing communication technologies for various NTN scenarios. In this work, we present our implementation of multiple channel models for NTNs provided by the 3rd Generation Partnership Project (3GPP) in an open source framework. The framework can be integrated into the existing Python framework Sionna™ , enabling the investigations of NTNs using link-level simulations. By keeping the framework open source, we allow users to adapt it for specific use cases without needing to implement the complex underlying mathematical framework. The framework is implemented in Python as an extension to the existing Sionna™ framework, which already provides a large number of existing 5G-compliant communications components. As the models in the framework are based on Tensorflow and Keras, they are compatible with not only Sionna™ , but also many existing software solutions implemented in Tensorflow and Keras, including a significant amount of the Machine Learning (ML) related research.**

***Index Terms*—Channel Simulations, Data Generation, Link-Level Simulation, Non-Terrestrial Networks**

I. Introduction

Non-Terrestrial Networks (NTNs) are one of the key technologies for reaching the goal of ubiquitous coverage. However, they also introduce complex challenges, including significantly different channel conditions among users within the same satellite beam, long delays caused by large distances, and substantial Doppler effects from the high velocities of non-terrestrial components [1]. Therefore, a comprehensive channel model is crucial when simulating NTN channels, as relying solely on simplified models may not sufficiently capture the full complexity of the challenge sufficiently.

To tackle complexities of NTN communication channels, accurate simulations are essential. This can apply to a wide range of applications, such as radio resource management and interference estimation [2]. However, the development of these solutions requires large amounts of data, creating a demand for diverse and cost-effective datasets [3]. Unfortunately, real-world data from NTNs is limited and expensive [4].

This research was supported in part by the German Federal Ministry of Education and Research (BMBF) within the projects Open6GHub under grant number 16KISK016 and 6G-TakeOff under grant number 16KISK068 as well as the European Space Agency (ESA) under contract number 4000139559/22/UK/AL (AIComS).

Our open-source framework addresses this need by offering a tool for simulations using channel models from the 3GPP TR38.811 standard [5]. This standard includes models, that were developed based on real-world measurements, providing a foundation for a broad spectrum of communication system simulations. This enables researchers and developers to test and refine their systems under complex non-terrestrial conditions, ensuring reliable performance across various scenarios.

In this paper, we present a new open source framework for the generation of NTN channel models, which can be integrated into link level simulations, specifically the existing software framework Sionna™. The models we chose to implement are based on real-world measurements, as we expect these to most closely mimic real-world scenarios. The framework can be used as an extension of the Sionna™ framework, an existing Python toolkit designed for research in communications [6]. Sionna™ utilizes Graphics Processing Unit (GPU) acceleration and a modular, open-source design to enable flexible high-performance simulations. Our open-source framework offers easy-to-use simulations compatible with existing software implementations, including the 5G-compliant Sionna™ toolkit and existing software solutions implemented in Tensorflow and Keras. By keeping the framework open-source, we intend to offer everyone the ability to further adapt the code for specific use cases, and enable the use of these highly advanced channel models, without needing to fully implement the complex underlying mathematical model.

At the time of release of this paper, the framework can be found on Github under the name OpenNTN [7].

II. Capabilities and Applications of the Framework

Our framework allows for the definition of different adjustable channel models based in the 3GPP TR38.811 standard [5]. In this section, we will name the most important parameters users can directly adapt when using the framework.

The models provided by the standard describe multipath channels with frequency selectivity. In general, the channels are separated into the scenarios dense urban, urban, and suburban, which all describe different expected distances between users, buildings, and other objects and user velocities. Additionally, parameters such as the Line-of-Sight (LOS) probability also change based on the scenario. While users can adjust these parameters, the framework includes default parameter sets provided by 3GPP.

After selecting a scenario, the transmission parameters can be adjusted. These include the direction (uplink (UL) or downlink (DL)), carrier frequency, elevation angle (angle between the user and satellite and the horizon), the satellite height, User Terminal (UT) and satellite antennas, and the number of UTs. The carrier frequency needs to be either in S-Band (defined in this standard as 1.9 GHz-4 GHz) or Ka-Band (defined in this standard as 19 GHz-40 GHz). The elevation angle needs to be between 10 degrees and 90 degrees. The satellite height needs to be between 600 km and 36 000 km. For each simulation, the satellite and UTs can have either singular or array antennas with different gain patterns, such as omnidirectional antennas and VSAT antennas. While the satellite's antenna and UT antennas can differ, all UTs are equipped with the same antenna. Lastly, the number of UTs can be freely adjusted, whereas the number of satellites is always 1. Additionally, the satellite will always be considered as the base station.

The defined model can be used to generate Channel Impulse Responses (CIRs), which consist of the path coefficients and path delays for each simulated path for each transmitter receiver antenna pair. The generated CIRs can be used directly or processed further to create a channel matrix for a predefined Orthogonal Frequency-Division Multiplexing (OFDM) system.

Through both the use of array antennas and the use of multiple UTs, the framework enables multi-user Multiple Input Multiple Output (MIMO) simulations. The framework also provides control over the Doppler shift effect, which can be toggled on or off, depending on whether satellite-induced Doppler shifts should be considered in the simulation. This is particularly interesting, as the Doppler introduces a very significant challenge, but can sometimes be assumed to be known and compensated, if for example the exact satellite position and orbit is known [8].

The framework's modularity allows to customize scenarios further by altering parameters such as user movement, velocity, and geographic settings. These capabilities enable researchers to investigate a broad range of scenarios, from densely populated urban environments to more dispersed suburban scenarios.

By integrating with the Sionna™ ecosystem, this framework is compatible with other existing 5G components, such as Low Density Parity Check (LDPC) coding, channel estimation, and OFDM, already provided by Sionna™. The GPU-acceleration of the framework ensures fast simulations, making it possible to handle large-scale investigations. Finally, the framework's open-source design allows for further customization and adaptation for specific research needs.

III. Non-Terrestrial Channel Model

The mathematical models behind the NTN channel models are defined in the 3GPP TR38.811 standard, which is an extension of the standard for the simulations of terrestrial channels defined in 3GPP TR38.901. Additionally, other documents have been used to define the models, such as descriptions of the environmental losses provided by the International Telecommunication Union (ITU) found in [9] and further parameter and scenario descriptions found in 3GPP TR38.821 [10]. As the full modeling process is complex and defined in many different documents, this section provides a summary of the process we implemented, to equip the reader with the necessary knowledge to adapt the framework and make use of its open-source nature.

The models implemented are frequency-selective multipath models. The frameworks' input is a scenario with the parameters defined by the user and the output is a set of all paths between each transmitter receiver antenna pair, consisting of both the path coefficient, its complex valued gain, and the path delay, its time of arrival relative to the first path. The process is divided into three parts: topology and large-scale parameters, small-scale parameters, and channel coefficient generation. All parts together consist of 16 steps, which we have visualized in Fig. 1. The boxes in blue showcase steps that are either unique to NTN models or differ significantly from terrestrial channel model generations defined in 3GPP TR38.901 [11], whereas the steps in red are very similar to the process used in terrestrial simulations.

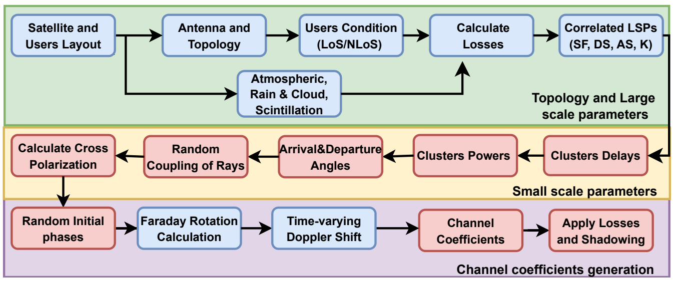

Fig. 1: Channel profile generation procedure based on the 3GPP TR38.811 standard, showing the steps involved in generating communication links for NTNs.

A. Topology and Large Scale Parameters

1) Satellite and Users Layout: First, a layout for the satellite and users is generated. This consists of the position and orientation of each UT and the satellite in a global coordinate system. Aspects such as the distance between users are randomly sampled based on the scenario, with dense urban having little distances between UTs and suburban having large distances. The distance between the UTs and the satellite depends on the elevation angle, usually called θ in the standards, and the satellites height. The UTs orientation is random, while the satellite always points directly downwards towards the Earth's surface. A visualization of the 3D coordinate system is shown in Fig. 2.

2) Antenna and Topology: All UTs and the satellite get an antenna assigned. The antenna can be either singular or an array. The available antenna patterns are:

- Satellite Antenna:

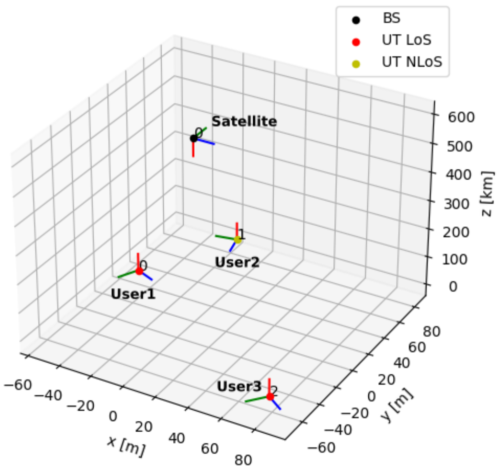

Fig. 2: Example coordinate system used in the simulation with UTs placed according to the selected dense urban scenario and an elevation angle of 90 degrees.

1) Reflector antenna with a circular aperture as described by [5]:

$$\begin{cases} 1 & \text{for } \theta = 0 \\ 4\left|\frac{\mathrm{J}_1(ka\sin\theta)}{ka\sin\theta}\right|^2 & \text{for } 0 < |\theta| \leq 90^\circ \end{cases} \tag{1}$$

where $\mathrm{J}_1(x)$ is the Bessel function of the first kind and first order, a is the radius of the antenna aperture. $k = 2\pi\frac{f}{\mathrm{c}}$ is the wave number for f as the carrier frequency and c as speed of light, and finally θ as the angle measured w.r.t the bore sight of the main beam.

2) Uniform rectangular panel array with dual linear polarization antenna pattern as defined in [11] Section 7.

- UT Antenna: For the users, we can choose 3 models:
  1) Quasi isotropic antennas for handheld UTs,
  2) Dual linear polarized patterns,
  3) very small aperture terminal (VSAT) antennas modeled the same way as the uniform rectangular panel array based on [11].

3) Propagation Condition: Each UT is assigned to be in either LOS or Non-Line-of-Sight (NLOS). The probability for each state is based on the scenario and elevation angle θ based on [5] Table 6.6.1-1. Different from terrestrial simulations, the indoor condition is not considered for satellites.

4) Atmospheric, Rain & Cloud, Scintillation: Due to the propagation of the signal between the satellite and the Earth's surface, multiple types of losses arise, which are not present in terrestrial communications. As their calculation involves many steps, we will only name them here. $\mathrm{PL_g}$, the gas loss or tropospheric loss (only considered in Ka-Band frequencies, 0 in S-Band). $\mathrm{PL_s}$, scintillation loss or ionospheric loss (only considered in S-Band frequencies, 0 in Ka-Band). Rain and cloud losses are incorporated into the other losses and not directly named.

5) Calculate Losses: We express the total loss as $\mathrm{PL_{total}}$

$$\mathrm{PL_{total}} = \mathrm{PL_b} + \mathrm{PL_g} + \mathrm{PL_s} + \mathrm{PL_e}\ . \tag{2}$$

$\mathrm{PL_e}$ is the building entry loss, which is always 0, as no UTs indoors are considered. $\mathrm{PL_b}$ marks the basic pathloss, calculated as

$$\mathrm{PL_b} = \mathrm{FSPL}\left(d, f_c\right) + \mathrm{SF} + \mathrm{CL}\left(\theta, f_c\right), \tag{3}$$

where

- FSPL is the free space pathloss defined as $\mathrm{FSPL}\left(d, f_c\right) = 32.45 + 20\log_{10}\left(f_c\right) + 20\log_{10}(d)$, d as distance between UT and satellite and f_c as carrier frequency,
- The shadow fading SF, drawn from a Gaussian distribution based on the elevation angle θ $\mathrm{SF} \sim N\left(0, \sigma_{SF}^2\right)$, based on [5] Table 6.6.2-1 to 3, and
- the clutter loss CL $\mathrm{CL}\left(\theta, f_c\right)$, a constant based on θ, which is 0 for LOS cases.

6) Correlated LSPs (SF, DS, AS, K): For each user, delay spread (DS), zenith of arrival (ZOA) and zenith of departure (ZOD), the Rician factor K are all sampled from Gaussian distributions, parameterized based on [5] Chapter 6.7.2. After the initial sampling, the parameters are correlated using a cross correlation matrix simulated according to [12] part 3.3.1.

B. Small Scale Parameters

The methodology for this part is based on [11] Chapter 7.5 and parameters based on [5]. The main concept the generation of m clusters with n rays per cluster for each transmitter receiver pair. At the end of the model generation, all rays arriving at a receiver at the same time form one path. The number of clusters depends on the elevation angle and scenario. A visualization of the idea is shown in Fig. 3, in which a satellite transmits a signal to Earth, resulting in multiple clusters, each cluster creating multiple rays to each user.

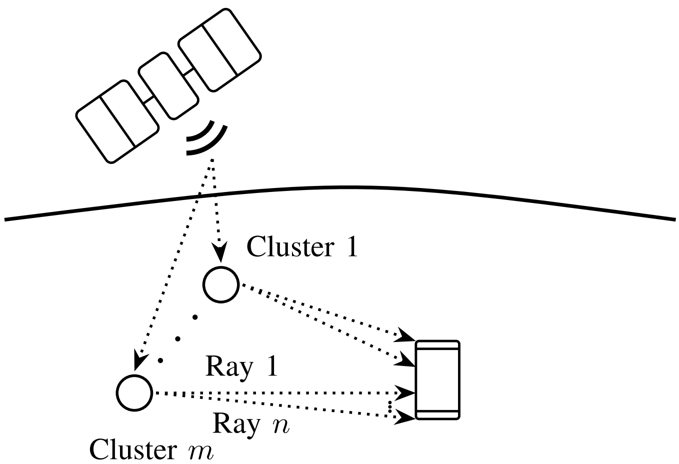

Fig. 3: Visualization of the principle clusters and rays structure in a DL example.

1) Cluster Delays: For each transmitter (satellite in DL and UT in UL), m clusters are generated for each pair of transmitter receivers. For all clusters a delay is generated, which is used as the base delay for each of the n rays of this cluster. The delays are scaled with the DS and a scenario specific delay distribution proportionality factor. For the LOS case, the delays are scaled based the Rician factor K.

2) Cluster Powers: A cluster power is assigned to each cluster, based on its DS, the delay distribution proportionality factor and a shadowing term, which depends on the scenario and is sampled from a Gaussian distribution. After creating all cluster powers, the powers are normalized, so that the sum of all cluster powers is 1. In the LOS case, the individual cluster powers are scaled using the Rician factor K.

3) Arrival & Departure Angles: For each ray, the angle of departure and arrival is calculated for both the azimuth and zenith of the antennas. The angle of arrival depends the antennas orientation and the angular spread, cluster power and a scaling factor, which depends on the total number of clusters. Additionally, all rays are grouped into sets of two and a phase offset based on the initial phase is applied.

4) Random Coupling of Rays: Here, the order of rays from the same cluster is randomly shuffled, to avoid artificial correlations, as the phase offset from the previous step are always on the same rays.

5) Calculate Cross Polarization: For each ray the cross polarization is generated, based on a value, randomly sampled from a Gaussian distribution, which is parameterized based on the scenario.

C. Channel Coefficient Generation

The last part of functional blocks generates the complex valued channel gains based on the procedure explained in [11] with non-terrestrial specific parameters and the addition of phase rotations due to Faraday Rotation and Time-Varying Doppler Shift based on [5].

1) Random Initial Phases: For each ray, an initial phase is drawn from a uniform distribution from $(-\pi, \pi)$.

2) Faraday Rotation Calculation: The Earths magnetic field and ionized medium introduce a rotation to the rays known as the Faraday rotation. This is calculated based on the carrier frequency.

3) Time-varying Doppler Shift: As the base station is considered stationary in the terrestrial scenarios, the Doppler shift is only introduced by the movement of the UTs. In the NTN case, the time-varying satellite Doppler shift is calculated in addition and both shifts are added. The Doppler shift is based on the satellites' velocity relative to the signal, which depends on the satellites' height. Additionally, a random orientation is sampled to simulate the orbital direction of the satellite. If the Doppler influence is disabled by the user, its value is set to 0 here.

4) Channel Coefficients: Using all of the calculated results, the channel coefficients are calculated for each ray. A very simplified version of the equation found in [5] is

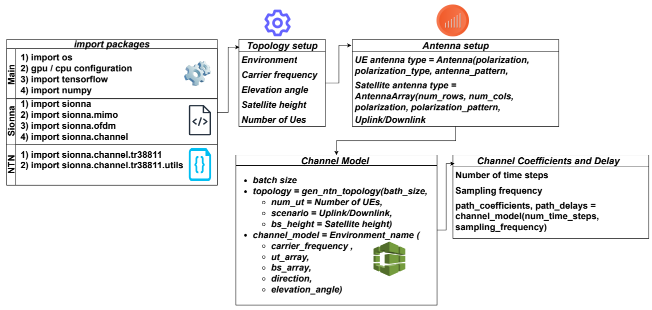

Fig. 4: Functional blocks of the user interface and the corresponding parameters required from users.

$$H_{u,s,n,m}(t) = P_{n,m} \cdot G_{\text{Rx}} \cdot \Phi \cdot \Psi_{\text{Faraday}} \cdot G_{\text{Tx}} \cdot e^{j\omega_{\text{Rx}}} \cdot e^{j\omega_{\text{Tx}}} \cdot e^{j\omega_{\text{Doppler}}}, \quad (4)$$

where

- $H_{u,s,n,m}(t)$ is the complex gain of the received signal from transmitter u to receiver s for ray n of cluster m,
- $P_{n,m}$ is the power of a ray from cluster m,
- G_{Rx} and G_{Tx} are the antenna gains (receiver and transmitter) based on their radiation patterns and the angles of arrival and departure,
- Φ is the initial phase of the ray,
- Ψ_{Faraday} represents Faraday rotation due to the Earth's magnetic field,
- $e^{j\omega_{\text{Rx}}}$ and $e^{j\omega_{\text{Tx}}}$ account for the phase shifts at the receiver and transmitter due to their position, and
- $e^{j\omega_{\text{Doppler}}}$ captures the Doppler shift due to the relative motion between the user and satellite.

5) Apply Losses and Shadowing: Finally, the generated channel coefficients values are multiplied by PL_{total} to apply the losses and shadowing. Please note that this is without considerations of the transmission power, which is currently always considered to be 1.

IV. User Interface

In this section we briefly showcase the user interface of the framework, including its setup and the inputs and outputs.

Generating the channel includes four steps of antennas, scenario, topology, and channel coefficient model as explained in Section III. Fig.4 shows the different steps of using the framework and what type of parameters are required.

Additionally, shadow fading and pathloss can be disabled. A batch of topologies is generated by defining the batch size, number of users, satellite height, and propagation scenario. If nothing else is specified, the expected distance between users, user height, and user velocities are generated based on values from the standards [5], [10]. Finally the topology is applied to the scenario.

Then, based on all the defined parameters the channel model is generated as shown in (4), including all Large Scale Parameters (LSPs), which includes path coefficients and path delays. The generated path coefficients are in a tensor with the dimensions $N_{\text{batch size}} \cdot N_{\text{path}} \cdot N_{\text{time step}} \cdot N_{\text{Transmitter}} \cdot N_{\text{Tx Antenna}} \cdot N_{\text{Reviver}} \cdot N_{\text{Rx Antenna}}$, where $N_{\text{batch size}}$ is batch size, N_{path} is the total number of paths between each transmitter and receiver, $N_{\text{time step}}$ is the number of successively simulated time steps, $N_{\text{Tx Antenna}}$, and $N_{\text{Rx Antenna}}$ are the number of antenna elements for each transmitter and receiver respectively.

In the case of DL transmission $N_{\text{Transmitter}} = 1$, with the transmitter being the satellite. The number of receivers is equal to the numbers of UTs $N_{\text{Receiver}} = N_{\text{UT}}$. This relation is the other way around in UL. The CIRs are generated out of the channel model, which can then be used to generate a desired channel. The Sionna™ framework already provides a function to use the CIRs to generate OFDM channels for both the time and the frequency domain. Both of those require a resource grid.

V. Calibration and Testing

To ensure the correctness of our implementation, we conducted a thorough testing process consisting of three steps: calibration against 3GPP standards, traditional unit tests, and performance validation through comparisons with other simulation models.

A. Calibration using the 3GPP TR38.821 standard

We validated our model using the calibration guidelines provided in the 3GPP TR38.821 standard [10]. This standard defines multiple test scenarios, including variations in satellite heights, carrier frequencies, and elevation angles, and provides corresponding benchmark values for comparisons. There are 30 of these scenarios with varying relevance, providing values from real world measurements. For each of the scenarios, the standard provides the different values of the link budget, which are also generated during the simulation. These include elements with set values and elements of which the values are generated during the simulations. The generated elements are the Free Space Pathloss (FSPL), the loss due to atmospheric effects, and the loss due to scintillation. Using these and the predefined parameters, such as antenna gain and transmission power, we can calculate the Signal to Noise Ratio (SNR).

Table I shows the most important scenarios (SC). For each of the scenarios the FSPL, atmospheric loss, and scintillation loss were calculated using the framework. Additionally, the other provided parameters were used to calculate the SNR. As the table is very large, we have only provided the results for the simulated values and the most important scenarios. For all of the scenarios in Table I the results from our implementation, given in the row Simulation, closely match the values provided by the standard [10], indicated as 3GPP. The maximum deviation is $0.6\,\text{dB}$ for the SNR in SC1 UL, which we consider an acceptable margin. Other tools, such as the C++ NTN simulation framework provided by ns3 [13], reached similar accuracies. The small deviations between the simulations and the values provided by 3GPP likely stem from the 3GPP values being real measurements instead of simulation results.

B. Unit Testing of Framework Components

The second step is the use of traditional unit tests to ensure the functionality of all components. These unit tests, available in our GitHub repository, systematically verify the functionality of system components, including parameter generation, Doppler effect handling, and antenna configurations. Running these tests provides confidence that each module works as expected and maintains correctness throughout code updates.

Their structure follows the approach of testing the input and output of all individual steps of the parameter generation process, the correct behavior in predefined edge cases, such as incorrect orbit heights or frequencies, and the correct final results for given values, such as the losses used for the link budget.

C. Comparison with Other Existing Models

Besides the validation of our models according to the standards provided by 3GPP and the traditional unit tests we can also compare our framework with different existing implementations. By doing that, we can validate the reasonability of our results. However, as the models in our framework follow distributions of which we sample specific values, full analytical verifications are hardly possible. Additionally, as the other frameworks focus on other aspects of the simulation or use different models, a certain mismatch between our results and the existing frameworks is to be expected.

Frameworks and datasets we can compare our framework with include the MATLAB implementation of the 3GPP TR38.811 standard [14], the implementation of ns3 in C++ of the 3GPP TR38.811 standard [13], the implementation of Gigayasa of the Cluster Delay Line (CDL) and Tapped Delay Line (TDL) models of the 3GPP TR38.811 standard [15], and a dataset provided by the university of Luxembourg, containing channel coefficients for an NTN scenario [16]. Tools like the link level simulator provided by Gigayasa can be used to validate full end-to-end simulations, whereas the system level simulations of ns3 can be used to compare our link budgets.

VI. Outlook and Conclusion

In this work, we presented an open-source framework for simulating NTNs based on the 3GPP TR38.811 standard. Our framework allows users to simulate a wide variety of scenarios and environments, including dense urban, urban, and suburban scenarios. Through thorough validation, including calibration against 3GPP standards, unit testing, and performance comparisons, we have ensured the accuracy and robustness of the implementation.

For future work, we plan to expand the capabilities of the framework in several directions. First, we aim to incorporate analytical models from the 3GPP TR38.811 standard, once the full parameter sets are available. This will enable more precise simulations across different elevation angles, removing the current dependence on scaling parameters for non-standard angles. As new standards are released by 3GPP and other

TABLE I: Link Budget Calibration Results Based on 3GPP TR38.821 [10]

Scenario	Origin	FSPL [dB]	Atmospheric [dB]	Scintillation [dB]	SNR [dB]
SC1 DL	Simulation	210.6	1.3	1.2	11.4
	3GPP	210.6	1.2	1.1	11.6
SC1 UL	Simulation	214.2	1.3	1.4	-0.1
	3GPP	214.1	1.1	1.1	0.5
SC6 DL	Simulation	179.1	0.5	0.4	8.4
	3GPP	179.1	0.5	0.3	8.5
SC6 UL	Simulation	182.6	0.6	0.5	18.1
	3GPP	182.6	0.5	0.3	18.4
SC9 DL	Simulation	159.1	0.1	2.2	6.6
	3GPP	159.1	0.1	2.2	6.6
SC9 UL	Simulation	159.1	0.1	2.2	2.3
	3GPP	159.1	0.1	2.2	2.8
SC14 DL	Simulation	164.5	0.1	2.2	7.3
	3GPP	164.5	0.1	2.2	7.2
SC14 UL	Simulation	164.5	0.1	2.2	-3.1
	3GPP	164.5	0.1	2.2	-2.6
SC16 DL	Simulation	210.4	0.8	0.7	4.6
	3GPP	210.4	0.8	0.5	4.8
SC16 UL	Simulation	213.9	0.8	0.7	-6.8
	3GPP	213.9	0.7	0.5	-6.3

organizations, we will regularly update the framework to stay aligned with the latest developments.

Beyond the current 3GPP based models, we plan to incorporate additional models from organizations like the ITU to cover additional aspects, such as satellite constellations and advanced atmospheric conditions. We plan to explore extending our drop-based simulations to time-dependent topologies that simulate dynamic topology changes during the simulation.

In conclusion, this framework provides a versatile and robust platform for both traditional and ML based NTN research. By maintaining its open-source nature and updating it regularly, we aim to foster innovation and provide researchers with tools to explore new frontiers in NTN communication systems.

References

[1] I. S. M. Hashim and A. Al-Hourani, "Satellite Visibility Window Estimation Using Doppler Measurement for IoT Applications," *IEEE Communications Letters*, vol. 27, no. 3, pp. 956–960, 01 2023.

[2] F. A. Aoudia and J. Hoydis, "Model-Free Training of End-to-End Communication Systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2503–2516, 2019.

[3] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," 2016, http://www.deeplearningbook.org.

[4] F. Fourati and M.-S. Alouini, "Artificial intelligence for satellite communication: A review," *Intelligent and Converged Networks*, vol. 2, no. 3, pp. 213–243, 2021. [Online]. Available: https://www.sciopen.com/article/10.23919/ICN.2021.0015

[5] 3GPP, "Study on New Radio (NR) to support non-terrestrial networks," 3rd Generation Partnership Project (3GPP), Technical report (TR) 38.811, 10 2020, version 15.4.0.

[6] J. Hoydis, S. Cammerer, F. A. Aoudia, A. Vem, N. Binder, G. Marcus, and A. Keller, "Sionna: An open-source library for next-generation physical layer research," 03 2022.

[7] M. V. Tim Düe, "OpenNTN, open source channel models for NTN simulations," Available at https://github.com/ant-uni-bremen/OpenNTN (2025/01/21).

[8] M. Röper, B. Matthiesen, D. Wübben, P. Popovski, and A. Dekorsy, "Position Based Transceiver Design for Multiple Satellite to VSAT Downlink," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 7022–7040, 10 2024.

[9] ITU, "Ionospheric propagation data and prediction methods required for the design of satellite services and systems," 9 2016.

[10] 3GPP, "Solutions for NR to support Non-Terrestrial Networks (NTN)," 3rd Generation Partnership Project (3GPP), Technical report (TR) 38.821, 04 2023, version 16.2.0.

[11] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," 3rd Generation Partnership Project (3GPP), Technical report (TR) 38.901, 04 2024, version 18.0.0.

[12] P. Kyösti, J. Meinilä et. al, "WINNER II channel models," *IST-4-027756 WINNER II, D. 1. 1. 2 V1. 2*, 2007.

[13] G. Ferreira, "ns-3-ntn," Available at https://gitlab.com/mattiasandri/ns-3-ntn (2022/10/13).

[14] The MathWorks, Inc., "Model NR NTN Channel," Available at https://de.mathworks.com/help/satcom/ug/model-nr-ntn-channel.html (2024/08/09).

[15] Gigayasa Wireless Private Limited, "Gigayasa 5G toolkit," Available at https://gigayasa.com/5g-toolkit/ (2024/11/19).

[16] H. A.-H. Madyan Alsenwi, Eva Lagunas and S. Chatzinotas, "CVaR-based Robust Beamforming Framework for Massive MIMO LEO Satellite Communications," in *IEEE Global Communications Conference (IEEE GlobeCom)*, Kuala Lumpur, Malaysia, 2023.