Integration and Validation of LEO NTN-TDL Channel Models in OpenAirInterface5G

Sumit Kumar, Ion Turcanu
Luxembourg Institute of Science and Technology, Luxembourg
{sumit.kumar, ion.turcanu}@list.lu

***Abstract*—This paper presents the implementation and validation of 3GPP TR 38.811-aligned Non-Terrestrial Network tapped-delay-line (NTN-TDL) channel models in the OpenAirInterface5G (OAI) radio-frequency simulator (RFsim) for Low Earth Orbit (LEO) satellite scenarios. The proposed implementation supports standardized NTN-TDL A, B, C, and D multipath profiles together with key LEO channel dynamics, including time-varying propagation delay, bulk Doppler evolution, elevation-dependent delay-spread, and per-tap Doppler extension. The implementation is validated through code-level consistency checks and controlled OAI RFsim experiments designed for reproducibility. The results confirm correct tap delays, normalized powers, and Ricean (K)-factors against TR 38.811 reference tables. They further verify coherent delay and Doppler evolution, phase continuity across block updates, environment-dependent delay-spread execution across dense urban, urban, suburban, and rural settings, and PHY-level execution across NTN-TDL A–D profiles. Performance measurements further quantify the runtime and memory overhead of the optional per-tap extension. The implementation provides a reproducible OAI-based realization of LEO NTN-TDL models for channel-level and PHY experimentation, supporting future studies of link adaptation and robustness under realistic LEO channel dynamics.**

I. Introduction

Non-terrestrial networks (NTNs) are an important part of the 5G coverage-expansion roadmap, especially for sparse terrestrial coverage and wide-area resilient connectivity [1]. Among NTN options, LEO satellite systems are attractive because of their lower propagation delay than GEO systems [2], but they also exhibit strongly time-varying channel conditions driven by rapid satellite motion. Meaningful LEO radio-access experimentation therefore requires channel models that capture both frequency-selective multipath structure and pass-dependent delay, Doppler, and large-scale evolution. 3GPP TR 38.811 provides the reference basis for NTN radio-channel modeling in NR, including tapped-delay-line (TDL) profiles and elevation-dependent large-scale effects. However, practical open-source 5G NR platforms still offer only limited implementations of these models inside their RF-channel execution paths.

OpenAirInterface5G (OAI) [2]–[4] is a widely used open-source platform for academic 5G NR research. Its existing NTN support and baseline LEO RFsim modes provide a template for further development. However, the current LEO channel modes in OAI are based on dynamic one-tap links and do not provide 3GPP TR 38.811-aligned frequency-selective NTN-TDL channels with joint multipath, delay, Doppler, and large-scale evolution. Adding such models would therefore give researchers, particularly those working through simulation, a practical and reproducible environment for evaluating NTN algorithms.

This paper addresses that gap by integrating a TR 38.811-aligned LEO NTN-TDL channel model into the OAI RF simulator together with dedicated validation tools. Our current implementation supports transparent payloads and targets channel and PHY level experimentation, while regenerative payload support remains future work; integration into the OAI `develop` branch is in preparation. The main contributions of this paper are as follows:

1) We integrate standardized NTN-TDL A/B/C/D profile support into OAI RFsim for frequency-selective LEO channel emulation.
2) We extend the native LEO RFsim path with orbit-driven delay evolution, bulk Doppler updates, elevation-dependent delay-spread, and deterministic per-tap Doppler extension.
3) We provide standalone tools for validating channel construction, impulse response, dynamic evolution, environment dependent delay-spread behavior, NTN-TDL A–D PHY performance, and runtime overhead.

II. Related Work and OAI Baseline

The discussion below focuses on the software realization of NTN channel models within simulators, emulators, and protocol stacks. Analytical and measurement-based channel characterization is excluded, as our contribution targets implementation-level integration rather than channel modeling theory. Within this scope, prior work spans three main directions: reference or link-level environments exposing TR 38.811 tap delays, powers, and Ricean factors for controlled descriptor checks [5], [6]; network and system level simulators modeling satellite motion, elevation-dependent loss, shadowing, clutter, and constellation effects for large-scale evaluation [7]–[9]; and practical NTN platform integration in OAI, srsRAN, and commercial systems such as Amarisoft [10], [11]. In the OAI ecosystem, prior work established a basis for 5G-NTN experimentation [4], demonstrated GEO NTN operation in in-lab and over-the-air settings [12], [13], reported experimental GEO and LEO NTN demonstrations over OAI-based setups [2], [14], and

© 2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS 2026)

compared 5G NTN against DVB-S2/RCS2 in a user-centric evaluation [15].

Baseline LEO NTN Channel Models in OAI

Prior to this work, OAI RFsim provided simplified LEO NTN support through two separate resources. The first is a pair of native models, `SAT_LEO_TRANS` and `SAT_LEO_REGEN` [16], which apply only a bulk, single-tap propagation delay for transparent and regenerative satellite paths, respectively, over a fixed, non-configurable orbit geometry, without multipath dispersion or fading. The second is a set of generic terrestrial TDL models [17], which provide standard multipath structures (e.g., TDL-A/B/C/D/E) but are not NTN-specific and do not capture LEO propagation dynamics such as elevation-dependent delay evolution or satellite-motion-induced Doppler.

Neither resource, taken alone or in combination, provided a TR 38.811-aligned LEO NTN-TDL model that combines the standardized NTN-TDL A/B/C/D multipath structure with LEO-specific delay, Doppler, and large-scale channel evolution within a single, frequency-selective realization. OAI likewise lacked compiled validators for channel descriptor conformance, impulse-response/multipath-path correctness, dynamic-state (delay/Doppler) evolution, and runtime-overhead characterization. Our work addresses these gaps as discussed in Section III and Section IV.

III. Implementation Details

This section explains how the LEO NTN-TDL model is integrated into the OAI RFsim channel-processing path.

A. RFsim Integration

The LEO NTN-TDL model is integrated directly into the existing OAI RFsim channel-processing path rather than implemented as an external emulator. Following the native OAI architecture, descriptor construction is handled in `random_channel.c`, orbit-driven runtime updates in `apply_channelmod.c`, and validation through compiled tools under `openair1/SIMULATION/TOOLS`. This integration preserves compatibility with the existing RFsim architecture and enables validation at multiple levels, including descriptor construction, impulse response, dynamic channel evolution, and runtime overhead. Fig. 1 summarizes the implementation flow followed in this work.

B. LEO NTN-TDL Channel Descriptor Construction

The small-scale LEO NTN-TDL profiles are instantiated during channel descriptor creation in `random_channel.c`. The implementation adds the `SAT_LEO_NTN_TDL_A/B/C/D` descriptors corresponding to the TR 38.811 NTN-TDL A–D profiles (see Table I). Each descriptor combines the NTN-TDL fading structure with LEO time evolution, using a default altitude of (600) km unless overridden. For each profile, the standardized delay-power parameters are mapped to OAI's internal channel representation, including the tap count, normalized tap powers, and Ricean (K)-factor for the LOS-oriented profiles C and D.

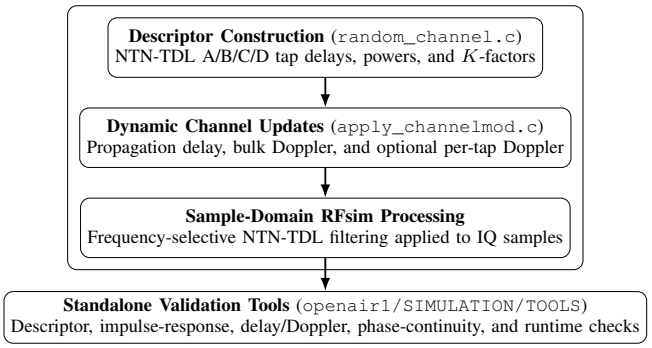


Fig. 1. Integration of LEO NTN-TDL channel model into OAI RFsim

TABLE I
INTEGRATED NTN-TDL PROFILE MAPPING FROM 3GPP TR 38.811

Prof.	Type	Taps	K	Descriptor	TR ref.
A	NLOS	3	N	`SAT_LEO_NTN_TDL_A`	Tbl. 6.9.2-1
B	NLOS	4	N	`SAT_LEO_NTN_TDL_B`	Tbl. 6.9.2-2
C	LOS	2	Y	`SAT_LEO_NTN_TDL_C`	Tbl. 6.9.2-3
D	LOS	3	Y	`SAT_LEO_NTN_TDL_D`	Tbl. 6.9.2-4

C. Dynamic Delay and Bulk Doppler Update

This subsection details how the LEO NTN-TDL runtime path computes the time-varying propagation delay and bulk Doppler shift that drive the channel's evolution over a satellite pass. We extend OAI `update_channel_model()` function in `apply_channelmod.c` to implement runtime delay, Doppler, and large-scale evolution. The implemented LEO channel model assumes a circular orbit with satellite radius $R_{\text{sat}} = R_E + h$, where R_E is the Earth radius and h is the satellite altitude. The orbital angular velocity is $\omega = \sqrt{\frac{\mu_E}{(R_E+h)^3}}$, where μ_E is Earth's standard gravitational parameter. The implemented two-dimensional orbit state is then expressed, for orbit time t, through the satellite coordinates $y_s(t) = (R_E+h)\sin(\omega t)$ and $z_s(t) = (R_E+h)\cos(\omega t)$, with velocity components $v_y(t) = \omega(R_E + h)\cos(\omega t)$ and $v_z(t) = -\omega(R_E + h)\sin(\omega t)$. From this geometry, an equivalent closed-form slant range as a function of elevation angle is $d_{\text{slant}}(\theta) = -R_E \sin\theta + \sqrt{(R_E + h)^2 - R_E^2 \cos^2\theta}$, where θ is the elevation angle. In the current RFsim implementation, inherited from the existing OAI baseline geometry, the UE and gNB are modeled at the same ground reference point $(0, 0, R_E)$, so $\theta(t)$ is obtained from the satellite state $(y_s(t), z_s(t))$ relative to that ground terminal. In the transparent-payload case, the two one-hop propagation distances in the delay equation are the UE-to-satellite and satellite-to-gNB distances, computed directly by the runtime path from general 3-D satellite, UE, and gNB position vectors (rather than the planar orbit state above) and only afterward specialized via the closed-form $d_{\text{slant}}(\theta)$ helper (which is instead used separately for the elevation-dependent path loss of Sec. III-D): i.e. $d_{\text{UE}\to\text{SAT}}(t) = d_{\text{SAT}\to\text{gNB}}(t) = d_{\text{slant}}(\theta(t))$.

$$\tau(t) = \frac{d_{\text{UE}\to\text{SAT}}(t) + d_{\text{SAT}\to\text{gNB}}(t)}{c}, \tag{1}$$

and the RFsim delay state is updated through the corresponding sample offset $n_\tau(t) = \lfloor \tau(t) f_s \rfloor$, where c is the speed of light and f_s is the sampling rate.

The bulk Doppler term, i.e., the common orbit-driven Doppler component applied to the channel before any tap-specific offsets, is derived from the satellite radial velocity along the active propagation path, $v_r(t) = \frac{\mathbf{v}_s(t)\cdot\mathbf{r}(t)}{\|\mathbf{r}(t)\|}$, where $\mathbf{v}_s(t)$ is the satellite velocity vector and $\mathbf{r}(t)$ is the relative-position vector directed along the link in use, i.e., from UE to satellite for the uplink Doppler term and from satellite to UE for the downlink Doppler term (the two are related by $\mathbf{r}_{\mathrm{UL}}(t) = -\mathbf{r}_{\mathrm{DL}}(t)$), which yields the implemented uplink and downlink Doppler shifts

$$f_D^{\mathrm{UL}}(t) = -\frac{v_r(t)}{c} f_c, \qquad f_D^{\mathrm{DL}}(t) = \frac{v_r(t)}{c - v_r(t)} f_c \tag{2}$$

with f_c denoting the carrier frequency. The channel runtime then converts this into the per-sample phase increment $\Delta\phi(t) = \frac{2\pi f_D(t)}{f_s}$, which is stored in `Doppler_phase_inc`, so the phase advances consistently with the pass-dependent Doppler evolution across successive sample blocks.

D. Elevation-Dependent Path Loss and Delay Spread

This part of the implementation captures elevation-dependent path loss and delay-spread evolution. During runtime updates, `apply_leo_ntn_largescale_path_loss()` computes the current elevation-dependent path-gain terms, while `apply_leo_ntn_elevation_ds()` updates the effective delay spread when the elevation-dependent NTN delay-spread mode is enabled. For the large-scale component, the implementation evaluates the free-space path loss as $FSPL_{\mathrm{dB}}(d, f_c) = 32.45 + 20\log_{10}(d_{\mathrm{km}}) + 20\log_{10}(f_{\mathrm{MHz}})$,

where d_{km} is the propagation distance in kilometers and f_{MHz} is the carrier frequency in megahertz. The implementation then uses the elevation-dependent excess term relative to the overhead case

$$\Delta FSPL_{\mathrm{dB}}(\theta) = FSPL_{\mathrm{dB}}(d_{\mathrm{slant}}(\theta), f_c) - FSPL_{\mathrm{dB}}(d_{\mathrm{slant}}(90^\circ), f_c). \tag{3}$$

Delay-spread adaptation is implemented through the newly introduced function `ntn_tdl_rescale_delay_spread()`, which rescales the tapped-delay structure according to the elevation-dependent effective delay spread while preserving the normalized NTN-TDL profile shape. Further, the runtime DS is obtained from the nearest TR 38.811 environment/elevation log-domain parameters $(\mu_{\log DS}, \sigma_{\log DS})$ using a deterministic seeded draw. If τ_i^{norm} denotes the normalized tap delay of the selected NTN-TDL profile and DS the runtime delay spread selected from the applicable elevation bin, the implemented rescaling is

$$\tau_i = \tau_i^{\mathrm{norm}}\, DS, \tag{4}$$

with the overall delay extent $T_d = \tau_{N-1}^{\mathrm{norm}}\, DS$. This preserves the relative tap structure of the TR 38.811 profile while adapting its absolute spread to the current LEO elevation condition.

E. Per-Tap Doppler Extension

The implementation also introduces an optional OAI per-tap Doppler extension. It is initialized through `ntn_init_per_tap_doppler()` and refreshed through `ntn_sync_per_tap_doppler()`. The current design provides a deterministic and reproducible per-tap Doppler realization: tap 0 carries only the bulk orbit Doppler, whereas taps $1 \ldots N-1$ are assigned fixed seed-derived frequency offsets within a configured spread. These offsets remain fixed for the lifetime of the descriptor, while the corresponding phase increments are updated whenever the bulk Doppler changes.

Accordingly, if $f_D(t)$ is the bulk Doppler and δf_i is the deterministic offset assigned to tap i, the implementation updates the tap-specific phase increment at each channel-state refresh step b as

$$\Delta\phi_i[b] = \Delta\phi_{\mathrm{bulk}}[b] + \frac{2\pi\delta f_i}{f_s}, \qquad \delta f_0 = 0, \tag{5}$$

with $\Delta\phi_{\mathrm{bulk}}[b]$ taken from the current bulk `Doppler_phase_inc`. During sample processing, the accumulated tap phase then follows the discrete recursion

$$\phi_i[n{+}1] = \phi_i[n] + \Delta\phi_i[b]. \tag{6}$$

This is an OAI deterministic extension that introduces controlled frequency-selective time variation around the common orbit Doppler, rather than a fully calibrated TR 38.811 temporal Doppler-spectrum model. It allocates additional per-tap state, including `tap_Doppler_offset_hz`, `tap_Doppler_phase_inc`, `tap_Doppler_phase_cur`, and `tap_ch`, layered on top of the bulk orbit Doppler.

The implemented runtime mechanisms can be summarized as follows. Bulk delay and bulk Doppler evolution are handled through the extended `update_channel_model()` path, which applies the geometry-driven pass-dependent time-of-flight and common orbit Doppler updates. Elevation-dependent large-scale evolution is handled by `apply_leo_ntn_largescale_path_loss()`, while elevation-dependent delay-spread adaptation is handled by `apply_leo_ntn_elevation_ds()` together with the new `ntn_tdl_rescale_delay_spread()` helper. The optional deterministic per-tap Doppler extension is initialized and refreshed through `ntn_init_per_tap_doppler()` and `ntn_sync_per_tap_doppler()`. These runtime paths combine TR 38.811-aligned profile import with executable RFsim channel evolution and are verified through the dedicated descriptor, dynamic-state, impulse-path, and performance validators described later in the paper.

F. RFsim Sample Processing Path

Sample-domain channel application is handled by `rxAddInput()` in the RFsim source file `apply_channelmod.c`. When per-tap Doppler is disabled,

the received sample is generated using the standard time-domain channel response `ch[]` and is then rotated by the bulk Doppler phase accumulator. When per-tap Doppler is enabled, the received sample is instead formed as the sum of tap-wise filtered contributions built from `tap_ch[]`, with each tap contribution rotated by its own tap-specific phase. This per-tap convolution and phase rotation path constitutes the main source of the additional runtime cost quantified later in Table IV.

For the implemented tapped-delay realization, a tap with complex gain $a_{i,q}$ on link q and delay τ_i is mapped onto the discrete sample grid as

$$h_{i,q}[\ell] = a_{i,q}\,\mathrm{sinc}(\ell - \tau_i f_s - c_{\mathrm{off}})\,, \tag{7}$$

where ℓ is the discrete channel-sample index and c_{off} denotes the implemented channel offset. With per-tap Doppler enabled, `rxAddInput()` then computes the received sample as

$$y[n] = \sum_i e^{j\phi_i[n]} \sum_{q=1}^{N_{\mathrm{tx}}} \sum_{\ell=0}^{L-1} h_{i,q}[\ell]\, x_q[n-\ell], \tag{8}$$

where L is the channel length and $x_q[\cdot]$ is the transmitted sample stream from transmit antenna q. When per-tap Doppler is disabled, the same structure reduces to a single assembled channel response followed by one bulk phase rotation.

The main implementation spans descriptor construction and per-tap state setup in `random_channel.c`, runtime channel evolution and sample processing in `apply_channelmod.c`, and compiled validators under `openair1/SIMULATION/TOOLS`.

The implemented design targets a practical and reproducible LEO NTN-TDL RFsim framework in OAI, centered on standardized NTN-TDL A–D profile integration, orbit-driven delay and Doppler evolution, elevation-dependent delay-spread handling, and an optional deterministic per-tap Doppler extension. At the current stage, NTN-CDL support, full spectral validation of the per-tap Doppler process, and exhaustive end-to-end performance evaluation across all NTN deployment conditions remain outside the present scope.

IV. Validation Methodology and Results

Validation is carried out using compiled OAI tools introduced in this work for descriptor checks, impulse-path inspection, dynamic LEO-state verification, PHY executability sweeps, and runtime-overhead benchmarking.

A. Descriptor and Impulse Validation Against TR 38.811

`validate_ntn_tdl_oai` verifies the instantiated `SAT_LEO_NTN_TDL_A/B/C/D` descriptors against the TR 38.811 Section 6.9.2 reference tap delays and relative powers, with LOS K-factor checks for the LOS-oriented profiles. It also verifies dynamic-delay and dynamic-Doppler enablement, valid satellite altitude, small-scale remapping behavior, and the elevation-dependent delay-spread lookup. Across these checks, the implemented descriptors reproduce the expected tap counts, normalized powers, tap delays, LOS K-factors, and delay-spread behavior for the integrated NTN-TDL A–D profiles. Next, `validate_ntn_tdl_impulse_oai` tool starts from an impulse input and records both the discrete output samples and an averaged discrete power-delay profile (PDP) accumulated over multiple channel realizations.

TABLE II
Basis-fit cluster-power recovery accuracy (two independent 50 000-realization runs, seeds S42/S7)

Prof.	Frac. err. (S42)	Frac. err. (S7)	Tot. power (S42)	Tot. power (S7)
A	1.44×10^{-3}	1.06×10^{-3}	1.023	1.019
B	1.11×10^{-4}	1.52×10^{-4}	1.046	0.993
C	5.5×10^{-5}	5.3×10^{-5}	0.999	0.992
D	1.8×10^{-5}	7.6×10^{-5}	0.983	1.000

Figure 2 shows a representative result for `NTN_TDL_B`. The top panel shows the expected tap locations and normalized tap powers from the instantiated descriptor, while the bottom panel shows the normalized power accumulated in each discrete sample bin at the output of the executed impulse-response path and averaged over multiple channel realizations. The same impulse-response validation was also executed successfully for `NTN_TDL_A/C/D`, with output sample-energy placement consistent with the expected tap structure of each profile.

To move beyond qualitative visual inspection, we quantify this agreement with a basis-fit power attribution: fractional-delay sinc interpolation causes closely spaced taps to overlap across neighboring sample bins (most severely for the three near-coincident early taps in `NTN_TDL_B`), so a naive hard bin-partitioning of the averaged PDP is not a valid attribution method. Instead, early/late cluster weights are fitted against the measured averaged PDP using the same sinc-interpolation kernel used by the OAI runtime path, and the recovered cluster-power fractions are compared against the TR 38.811-derived expected fractions over two independent $50\,000$-realization runs with fixed RNG seeds.

Table II reports the resulting fraction error and recovered absolute power per profile. The fraction error stays below 1.5×10^{-3} for all four profiles, confirming that the OAI discrete-time realization preserves the expected TR 38.811 cluster-power structure even in the overlap-sensitive `NTN_TDL_B` case. The recovered absolute power stays within a few percent of unity and does not trend consistently with realization count, consistent with residual Monte Carlo sampling noise rather than a runtime normalization defect.

B. Dynamic LEO-State Validation

The second validation layer uses `validate_ntn_dynamic_oai` to compare the channel state produced by `update_channel_model()` against an explicit LEO orbit reference over approach, near-overhead, and departure conditions. For uplink and downlink, it checks slant-range-driven propagation delay, bulk Doppler magnitude and sign, the expected near-zero Doppler condition around overhead, block-boundary bulk-phase continuity, and the expected increase of relative free-space path loss toward low

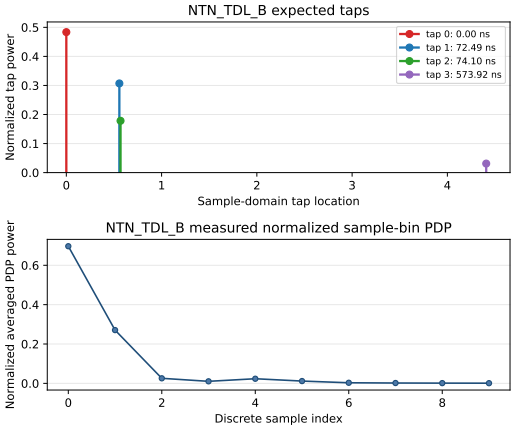


Fig. 2. Impulse-path validation for `NTN_TDL_B`. The top panel shows the expected descriptor tap locations in the sample domain and their normalized powers, while the bottom panel shows the measured normalized power in each discrete sample bin after execution of the impulse response path and averaging over multiple channel realizations.

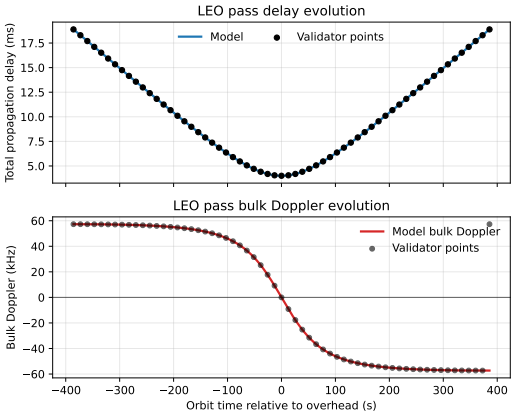


Fig. 3. Orbit-driven delay and bulk-Doppler evolution reproduced by the implemented LEO model. The curves are generated from the implemented equations, and the markers are extracted from the executed `validate_ntn_dynamic_oai` trace. The top panel shows total propagation delay over the visible pass, while the bottom panel shows bulk Doppler; at the shown scale, uplink and downlink trends are nearly coincident.

elevation. All checks pass, confirming that the orbit-driven delay and bulk-Doppler logic from Eqs. (1) and (2) behave coherently in the RFsim runtime path. Figure 3 shows the resulting pass-dependent delay and bulk Doppler evolution for the implemented 600 km LEO orbit and the representative 2.48615 GHz carrier used in this validation, for which the bulk Doppler spans approximately -57 to $+57$ kHz over the visible pass.

C. Per-Tap Doppler Validation

The optional per-tap Doppler path is validated as an OAI deterministic extension rather than as a full standardized Doppler-spectrum model. The descriptor validator checks that tap 0 carries the bulk orbit Doppler, that the remaining taps receive deterministic seed-derived offsets within the configured spread according to Eq. (5), and that the resulting per-tap state remains repeatable and internally consistent after synchronization with the bulk Doppler update. This confirms that the implemented frequency-selective time-variation path is deterministic and reproducible.

TABLE III
ENVIRONMENT AND PROFILE VALIDATION MATRIX AT 25 PRB AND 10 DB; ALL CASES PASSED SUCCESSFULLY, THE TABLE REPORTS DL/UL EFFECTIVE THROUGHPUT (EFF THROUGHPUT, $n = 300$ TRANSPORT BLOCKS PER OPERATING POINT)

Env/Prof	Runtime DS (ns)	Channel length	DL TP (%)	UL TP (%)
Dense urban	90.155	22	100.00	100.00
Urban	365.620	85	100.00	100.00
Suburban	21.571	6	100.00	100.00
Rural	0.262	1	100.00	100.00
Profile sweep at fixed $DS = 100$ ns				
NTN-TDL-A	100	6	99.83	99.67
NTN-TDL-B	100	10	99.83	99.67
NTN-TDL-C	100	24	100.00	100.00
NTN-TDL-D	100	13	100.00	100.00

D. PHY Validation Across Environments and Profiles

To validate the elevation-dependent delay-spread path beyond the urban baseline, we executed a multi-environment PHY sweep using the compiled OAI simulators `nr_dlsim` and `nr_ulsim`, verifying executable RFsim sample-chain behavior when the TR 38.811 environment lookup changes the runtime delay spread over a wide range.

The PHY follow-up used fixed 10° elevation, 25 PRB bandwidth, 10 dB SNR, a deterministic seed, and $n = 300$ transport blocks per run in `nr_dlsim`/`nr_ulsim`. For each environment, the logs record the sampled runtime delay spread and rescaled `channel_length`. As shown in Table III, the sampled delay spread spans more than three orders of magnitude, from 365.620 ns in the urban case down to 0.262 ns in the rural case, while the corresponding channel support shrinks from 85 samples to 1 sample. All four environments complete successfully in both DL and UL, confirming that the elevation-driven excess loss and delay-spread adaptation expressed by Eqs. (3) and (4) remain consistent with executable PHY processing across the tested environments.

The same PHY campaign was then extended across the standardized NTN-TDL-A/B/C/D profile families under the same 25 PRB, 10 dB, deterministic-seed `nr_dlsim`/`nr_ulsim` operating point. As also shown in Table III, all four profiles complete successfully in both directions, with measured `channel_length` values of 6, 10, 24, and 13 samples for NTN-TDL-A/B/C/D, respectively.

The same campaign was then tightened around the weakest observed corner, namely NTN-TDL-A uplink under the urban elevation-dependent delay-spread setting at low elevation. A narrow SNR sweep at $8/10/12/14$ dB for 10° and 30° showed that the `nr_ulsim` pass criterion, defined as effective throughput of at least 97% over 300 transport blocks, was

TABLE IV
MEASURED COST OF THE OPTIONAL PER-TAP DOPPLER PATH (INTEL CORE I9-10940X CPU @ 3.30 GHZ)

Case	Mode	Init μs	Block μs	Sample/Rx μs	Desc bytes
1×1	bulk only	572.818	190.360	0.046475	54020
1×1	per tap	562.511	687.254	0.167787	54788
2×2	bulk only	2187.121	532.271	0.064975	217440
2×2	per tap	2224.017	1708.631	0.208573	220224

first satisfied at 12 dB at both elevations, confirming stable executable behavior in the most demanding tested uplink case.

E. Performance Benchmark

The final validation layer measures the implementation cost of enabling the optional per-tap Doppler path. The `benchmark_ntn_tdl_perf_oai` tool repeatedly executes the RFsim `rxAddInput()` steady-state path for a `SAT_LEO_NTN_TDL_B` channel at 7.68 MHz and 2.48615 GHz, comparing bulk-Doppler-only and per-tap-Doppler modes for 1×1 and 2×2 cases using 4096-sample blocks over 2000 iterations on a single core of an Intel Core i9-10940X CPU @ 3.30 GHz.

In Table IV, enabling per-tap Doppler increases the steady-state block cost by 261.0% in the 1×1 case and by 221.0% in the 2×2 case, while the descriptor-owned memory increase remains modest. The relative overhead is smaller for 2×2 because the bulk-only baseline itself scales with antenna-pair count faster than the added per-tap cost, diluting the per-tap fraction of the total. Across both configurations, the initialization cost changes only slightly and without a consistent sign, indicating that the dominant overhead of the per-tap extension lies in runtime sample processing rather than in channel construction.

V. CONCLUSION

This paper presented the integration and validation of a TR 38.811-aligned LEO NTN-TDL channel model in the OpenAirInterface5G RF simulator, adding standardized NTN-TDL A–D profile support, orbit-driven delay and bulk-Doppler evolution, elevation-dependent delay-spread handling, and an optional deterministic per-tap Doppler extension to the original OAI NTN baseline. Compiled descriptor, impulse-path, dynamic-state, PHY, and performance validation confirm correct descriptor realization, coherent orbit-driven delay/Doppler evolution, stable delay-spread behavior across environments, and executable NTN-TDL A–D PHY operation, with the optional per-tap Doppler path mainly increasing runtime sample processing while remaining lightweight in descriptor memory. The present realization targets the transparent-payload case; ongoing work includes preparing the merge of this implementation into the OAI `develop` branch, protocol-level NTN validation, broader KPI-oriented campaigns, and refinement of the temporal fading process.

ACKNOWLEDGMENT

This work was supported by the 6G-PATH project under SNS JU Grant Agreement No. 101139172, through the WEBSAT-6G project.

REFERENCES

[1] F. Rinaldi, H.-L. Maattanen, J. Torsner, S. Pizzi, S. Andreev, A. Iera, Y. Koucheryavy, and G. Araniti, "Non-terrestrial networks in 5g & beyond: A survey," *IEEE access*, vol. 8, pp. 165 178–165 200, 2020.

[2] S. Kumar, C. K. Sheemar, J. Querol, T. Yilmaz, S. Chatzinotas, M. Hammouda, O. Kodheli, A. Astro, and F. Kaltenberger, "5G NTN LEO based demonstrator using OpenAirInterface5G," in *IET Conf. Proc.*, vol. 2023, no. 48, 2023, pp. 69–75.

[3] OpenAirInterface Software Alliance, "OpenAirInterface5G: OpenAir-Interface 5G Wireless Implementation," 2015, gitLab repository, Apache License 2.0. [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g

[4] S. Kumar, A. K. Meshram, A. Astro, J. Querol, S. Chatzinotas, T. Schlichter, G. Casati, T. Heyn, F. Völk, S. Kaya, R. T. Schwarz, A. Knopp, P. Marques, L. Pereira, R. Magueta, A. Kapovits, and F. Kaltenberger, "OpenAirInterface as a platform for 5G-NTN research and experimentation," in *Proc. IEEE Future Netw. World Forum (FNWF)*, 2022, pp. 500–506.

[5] MathWorks, "Model NR NTN Channel," Satellite Communications Toolbox documentation, 2024. [Online]. Available: https://nl.mathworks.com/help/satcom/ug/model-nr-ntn-channel.html

[6] T. Düe, M. Vakilifard, C. Bockelmann, D. Wübben, and A. Dekorsy, "An open source channel emulator for non-terrestrial networks," in *Advanced Satellite Multimedia Systems Conference/Signal Processing for Space Communications Workshop (ASMS/SPSC 2025)*, 2025.

[7] M. Sandri, M. Pagin, M. Giordani, and M. Zorzi, "Implementation of a channel model for non-terrestrial networks in ns-3," in *Proc. Workshop on ns-3 (WNS3)*, Arlington, VA, USA, Jun. 2023, arXiv:2305.05544.

[8] ns-3 Project, "Propagation models: ThreeGppNTNPropagationLossModel," ns-3 Model Library documentation, 2024. [Online]. Available: https://www.nsnam.org/docs/models/html/propagation.html

[9] Magister Solutions, "SNS3: Satellite network simulator 3," 2024, ns-3 satellite extension module developed under ESA contract. [Online]. Available: https://github.com/sns3/sns3-satellite

[10] Software Radio Systems, "srsRAN gNB for 5G NTN," srsRAN Project documentation, 2024. [Online]. Available: https://docs.srsran.com/projects/project/en/latest/tutorials/source/ntn/source/index.html

[11] Amarisoft, "5G SA NTN Tutorial," Amarisoft Tech Academy, 2025. [Online]. Available: https://tech-academy.amarisoft.com/NR_SA_NTN.html

[12] S. Kumar, A. Astro, O. Kodheli, J. Querol, S. Chatzinotas, T. Schlichter, G. Casati, T. Heyn, F. Völk, S. Kaya, R. T. Schwarz, A. Knopp, A. Kapovits, and F. Kaltenberger, "5G-NTN GEO-based in-lab demonstrator using OpenAirInterface5G," in *Proc. 11th Adv. Satellite Multimedia Syst. Conf. (ASMS)*, 2022.

[13] S. Kumar, O. Kodheli, A. Astro, J. Querol, S. Chatzinotas, G. Casati, T. Schlichter, T. Heyn, H. Cheporniuk, F. Völk, R. T. Schwarz, A. Knopp, A. Kapovits, and F. Kaltenberger, "5G-NTN GEO-based over-the-air demonstrator using OpenAirInterface," in *Proc. 39th Int. Commun. Satellite Syst. Conf. (ICSSC)*, 2022.

[14] F. Völk, T. Schlichter, S. Kumar, R. T. Schwarz, A. Knopp, M. Hammouda, T. Heyn, J. Querol, S. Chatzinotas, and A. Kapovits, "5G Non-Terrestrial Networks With OpenAirInterface: An experimental study over GEO satellites," *IEEE Access*, vol. 12, pp. 155 098–155 109, 2024.

[15] S. Kumar, J. C. Estrada-Jimenez, and I. Turcanu, "User-centric comparison of 5G NTN and DVB-S2/RCS2 using OpenAirInterface and OpenSAND," in *2025 IEEE Future Netw. World Forum (FNWF)*, 2025, pp. 1–6.

[16] OpenAirInterface Software Alliance, "LEO satellite delay and Doppler simulation and adjusted NR_UE time sync," GitLab merge request !2893, Aug. 2024.

[17] ——, "Draft: TDLA30, TDLB100 and TDLC300 channel models," GitLab merge request !1686, Aug. 2022.