Aerial Omniverse Digital Twin (AODT) Platform - New Featuers in Release 1.2

NVIDIA Telecom Product | Jan 30, 2024

AODT Release 1.2

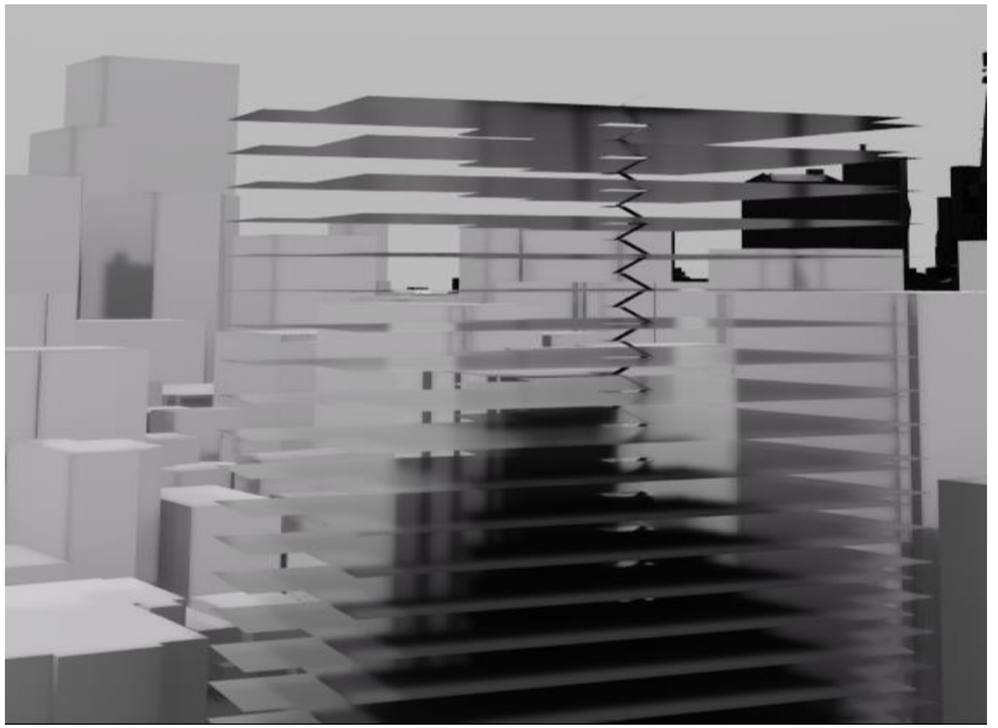
- **GIS Import:** Procedural indoor generation and Python pipeline
- **UI and UE Mobility:** Indoor/Outdoor UE allocation, UE Mobility
- **EM Solver:** Outdoor-indoor and indoor-indoor
- **RAN;** 64T64R and MU-MIMO, Different target BLERs
- **AI/ML Examples:** PUSCH and SRS Channel Estimation

HW Configuration: Single GPU

GIS Import: Indoor

Procedural generation of inter-floor and intra-floor mobility

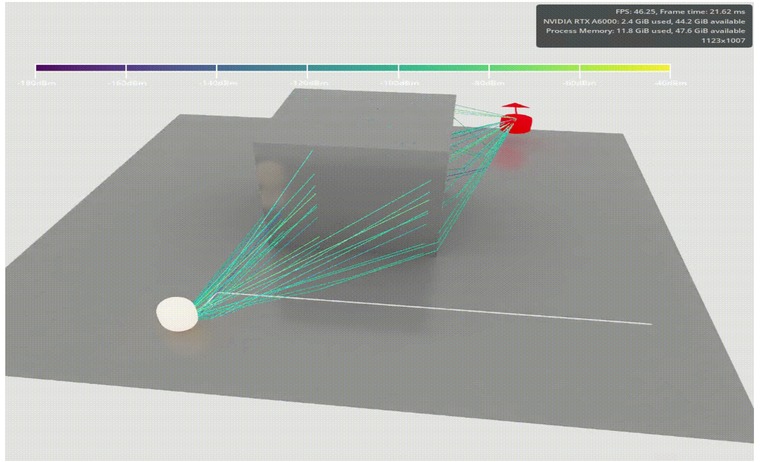
- Fully Pythonic SDK with more user-friendly example scripts
- Procedurally generated indoor spaces with any GML or OSM import.
  - Highly configurable, includes reasonably realistic indoor spaces (reasonable area, wall height, doors, etc.) with inter-floor accessibility, multiple floors where applicable.
- More tolerant mobility generation (disconnected components OK)
- Added Berlin sample map



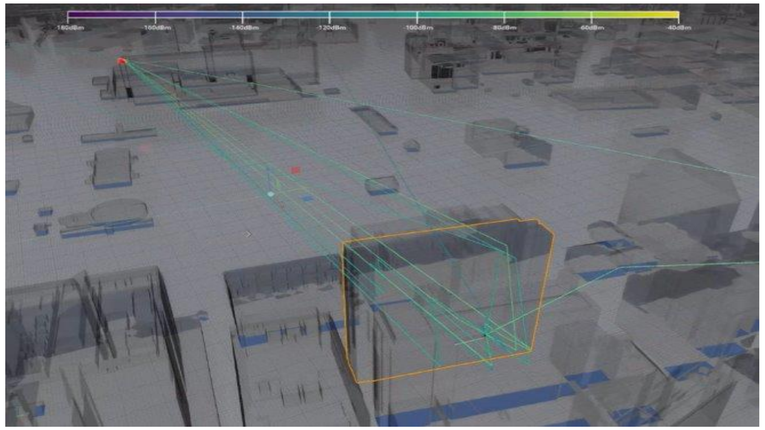
EM Solver

Key features in 1.2 release

- Transmission model:
  - Ray paths penetrating buildings (experiencing additional transmission loss)
  - Outdoor-to-indoor coverage for UEs inside a building
  - Indoor UE mobility
- Enable/Disable Diffraction
  - Users can enable or disable diffraction in their simulations, allowing for more precise control over the simulations in terms of use cases and scenarios.



EM transmission across obstacles



User Interface

Key features in 1.2 release

New Features

- Added functionality for DB replay, copy and delete.
- Enabled selection of DB tables (e.g., CFRs, CIRs, ray paths, telemetry, training results).
- Introduced indoor visualization mode.
- Supported deployment of RUs on walls with appropriate azimuth alignment.
- Implemented a minimum intra-element distance limit based on roll angles for half-wave dipoles.
- Prevented edits before worker attachment to ensure edits are created and managed during live sessions.

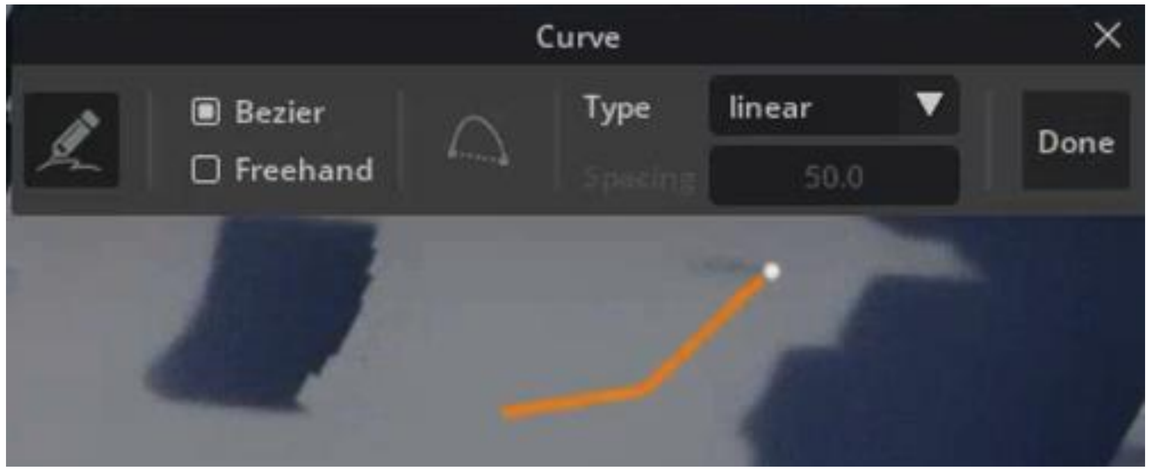
Tech Debt Reduction and Code Improvement

- Refactored attributes_widget.py (4,071 lines) into base_property_widget.py (525 lines).
- Deduplicated antenna patterns in prims and messages and established antenna unit tests.

UE Mobility

Flexible configuration of the UE mobility

- Specify a UE mobility path by clicking on the Edit Waypoints button in the UE property widget.
  - Draw a polyline defining the intended trajectory of the UE across the map.
  - Customized Speed (in meters per second)
  - Customized Azimuth Angle Offset (in degrees)
  - Customized Pause Duration (in seconds)
- Percentage of Indoor Procedural UEs controls the percentage of procedurally generated UEs that are placed indoor.
  - The ratio of indoor to outdoor UEs can now be manually tuned to match actual distribution, e.g. 80% indoor to 20% outdoor.



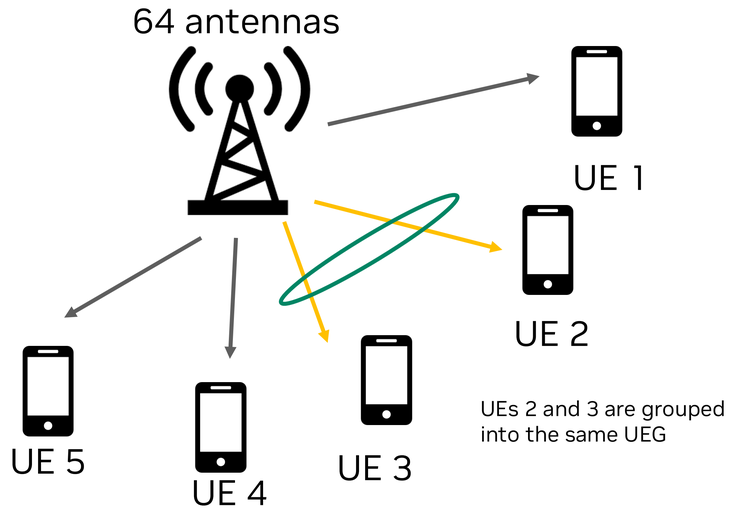
Waypoints Attributes

	Speed [m/s]	Pause duration [s]	Azimuth offset [deg]
Waypoint 0	0.0	0.0	0.0
Waypoint 1	0.0	0.0	0.0
Waypoint 2	0.0	0.0	0.0
Waypoint 3	0.0	0.0	0.0
Waypoint 4	0.0	0.0	0.0
Waypoint 5	0.0	0.0	0.0

RAN Digital Twin

MU-MIMO with 64TRx for better network efficiency

- Support 64 antennas at RU
  - The ability to support up to 64 antennas at the Radio Units (RUs) expands the scope and scale of RAN simulations.
- MU MIMO for DL and UL
  - Multi-User Multiple Input Multiple Output (MU-MIMO) in both downlink (DL) and uplink (UL) slots.

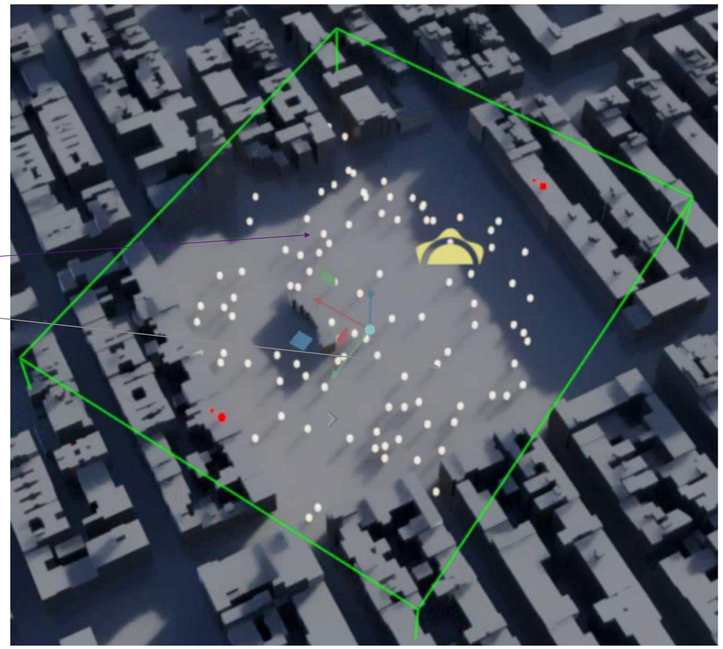


batch_id	slot_id	link	ru_id	ue_id	startPrb	nPrb	mcs	layers	tbs	rv	outcome	scs	preEqSinr	postEqSinr
...														
0	29	DL	2	1	220	52	10	3	4867	0	1	30000	9.182941	15.490848
0	29	DL	2	2	108	56	27	2	9987	0	1	30000	34.339355	37.904247
0	29	DL	2	3	108	56	27	2	9987	0	1	30000	34.548115	38.04521
0	29	DL	2	4	0	108	21	1	7172	2	1	30000	13.200821	27.504408
0	29	DL	2	5	164	56	4	2	1569	0	1	30000	9.53027	8.710031
...														
0	64	UL	2	1	228	44	5	4	3138	0	1	30000	6.5925293	7.3060503
0	64	UL	2	2	184	44	27	1	3905	0	1	30000	37.077072	39.6355
0	64	UL	2	3	184	44	27	1	3905	0	1	30000	37.04961	39.45168
0	64	UL	2	4	0	92	23	1	6913	2	1	30000	13.37422	25.350723
0	64	UL	2	5	92	92	8	2	4737	2	1	30000	9.552433	11.236434
...														

Different Target BLERs

Foundation for Multiple QoS flows

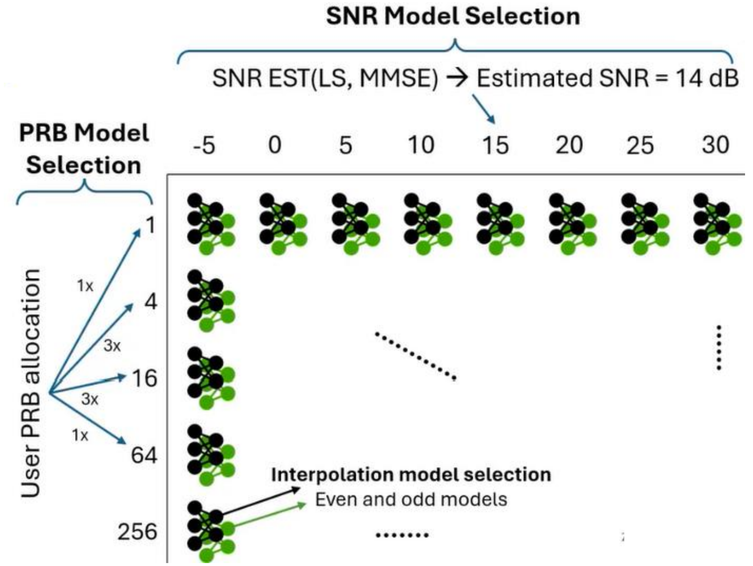
- Different target Block Error Rates (BLERs) can be set for different UEs.
  - The BLERs are used to guide adaptive modulation and coding schemes. This provides more granular control over RAN simulations, allowing for diverse and tailored performance analyses.
  - This is fundamental for multiple QoS flows with different QoS flows and allow to set some UE to 10% BLER, while others to 1%



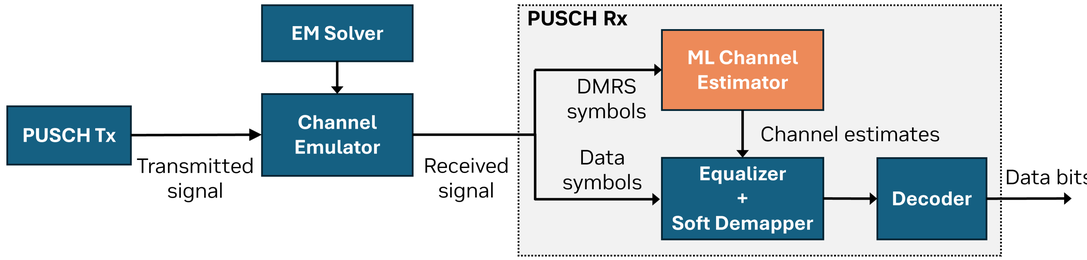
AI/ML Support

Accelerating Adv. 5G/6G Research and Development

- This release provides examples for the following AI-driven features
  - PUSCH Channel Estimation Inference:
    - Demodulation Reference Signal (DMRS) channel estimation inference demonstrates the integration of AI with AODT RAN simulation
  - SRS Channel Estimation Inference:
    - Sounding Reference Signal (SRS) channel estimation inference demonstrates the integration of AI with AODT RAN simulation
- Example workflow for the ML PUSCH Channel Estimation:
  - Step 1 - Generate the models
    - To generate the channel estimator models, use the PyAerial Channel Estimation notebook in aodt_sim/external/cuBB/pyaerial/notebooks/channel_estimation
    - This produces a different model for each one of the SNRs and PRB points
  - Step 2 - Make the models accessible to AODT
    - Once the models have been generated, they need to be exposed to aodt_sim. Configuration and data file access needs to be set properly
    - The channel estimation module will combine the models with different SNRs and PRBs to match the PRB allocation from the scheduler
  - Step 3 - Configure the RAN simulation to enable the RAN simulation to show enhanced performance



PUSCH AI-based DMRS Channel Estimation



AODT Hardware: Single GPU Support in Release 1.2

Single/Double GPU On-Prem, Public Cloud, NVCF

Rel 1.2 (Jan/Feb 25)

On Prem (Collocated) Jan 25

- Single-GPU, Ubuntu 22.04
- Double-GPUs, Front/Backend @ Window/Ubuntu 22.04
- Enterprise grade
  - Single or 2xRTX6000 Ada/48 GB
  - Ubuntu 22.04;
  - Qualified workstation: Dell 7960 with 2xRTX6000Ada; Dell760 with 2xL40
  - Target industrial and large-scale network simulation, fine-tuning and deployment

Public Cloud Jan 25

- Azure
  - Frontend
    - 12 GB+; GTX/RTX
    - RTX6000Ada, A10, L40
    - Window Server 2022
  - Backend
    - 48 GB+; GTX/RTX
    - RTX6000Ada, A100, H100, L40
    - Ubuntu 22.04

AODT-on-Cloud (NVCF) – Feb 25

- AODT running environment
- Synthetic data generation