Simulation Grid

The simulation comprised four terrestrial base stations (BSs), one macro-BS and three micro-BSs. We recreated 50 episodes of 100 pedestrian users randomly distributed in the grid (120x120 meters), registering 100 seconds of simulations with a resolution of 1 second.

The mobility pattern of the users and the distribution of the BSs in the grid guarantee recreating diverse reception conditions with a Channel Quality Indicator (CQI) range between 0 and 15. Specifically, 0 means the user is out of the coverage area of the BS, whereas 15 represents the best reception conditions.

The outputs of this simulation were used to train a deep reinforcement learning (DRL) to dynamically select the best BS to satisfy each user request. To achieve this task, we also included in the state space the QoS constraints of the considered applications, the priority of the users and the number of available RBs to attend to each service request.

Main configurations:

```
general_simulation_parameters = pd.DataFrame(
    [[
      "[120, 120]",  # "grid_xy"
      100,  # "simulation_time"
      1,  # "simulation_resolution"
      True,  # "downlink"
      False,  # "uplink"
      False,  # "d2d_link"
```

```python
        False,  # "ntn_link"
        True,  # "save_scenario_xlsx"
        True,  # "save_metrics_xlsx"
        True,  # "show_video"
        False,  # "save_video"
        "gif",  # "video_format"
        0.1,  # "video_velocity"
        True,  # "print_scenario_outputs"
        True  # "print_metrics_outputs"
    ]])

general_channel_modeling = pd.DataFrame(
    [[
        True,        # "dynamic_los"
        False,       # "dynamic_hb"
        False,       # "o2i
        "dynamic",   # "inside_what_o2i"
        "low-loss",  # "penetration_loss_model"
        True,        # "shadowing"
        True,        # "fast_fading"
        "jakes",     # "fast_fading_model"
        False,       # "atmospheric_absorption"
        "Very short"  # "desired_delay_spread"
    ]])

general_parameters = pd.DataFrame(
    [[
        -174,  # "thermal_noise"
        10,  # "h_ceiling"
        0.2,  # "block_density"
        "real",  # "channel_type"
        0.1  # "target_bler"
    ]])

bs_parameters = pd.DataFrame([
        [100, 100, 25, "tbs", "UMa", "three_sectors", "dual", "E", "B", 28, 2, 100, 20, 20, 2, 7, 15, None],
        [60, 60, 10, "tbs", "UMi", "three_sectors", "dual", "D", "A", 28, 2, 100, 10, 10, 2, 7, 15, None],
        [20, 20, 10, "tbs", "UMi", "three_sectors", "dual", "D", "A", 28, 2, 100, 10, 10, 2, 7, 15, None],
        [20, 100, 10, "tbs", "UMi", "three_sectors", "dual", "D", "A", 28, 2, 100, 10, 10, 2, 7, 15, None],
                ])

sub_groups_parameters = pd.DataFrame(
    [["pedestrian", 5000, "omni", 0, 0, 0, 7, True, True, "[1, 1]", "[60, 60]", "[0.4, 1.2]", 1, "Reference Point
Group model", 0, 50, "[1.5, 1.5]", "urban"],
        ])
```