**Communication Model Implementation in SAGIN Baseline**

**A single, general function compute\_rate\_general() in communication\_model.py is used to calculate wireless data rates between all network elements. Each communication scenario (IoT-to-UAV and UAV-to-Satellite) passes different, context-specific parameters to this function.**

**1. IoT to UAV Communication (TDMA)**

* **Approach:**  
  A Time Division Multiple Access (TDMA) scheme is used. Each UAV sequentially connects to individual IoT devices in its region; only one IoT device transmits to a UAV at any time. Once a transmission is completed, the next IoT device begins.
* **Rate Calculation:**  
  The function compute\_iot\_to\_uav\_rate() in iot\_region.py wraps compute\_rate\_general() with IoT-to-UAV-specific parameters:
  + **Bandwidth:** 5 MHz
  + **Transmit Power:** 0.1 W (IoT device)
  + **Carrier Frequency:** 2 GHz
  + **Gains:** 1 (IoT, isotropic), 10 (UAV, moderate)
  + **Noise:** 1e-9
  + **Fading:** Fixed at 1.0

**Call example:**

return compute\_rate\_general(

sender\_pos=iot\_pos + (0,), # IoT at ground level (z=0)

receiver\_pos=uav\_pos, # UAV 3D position

bandwidth=5e6,

P\_tx=0.1,

fc=2e9,

G\_tx=1,

G\_rx=10,

noise=1e-9,

fading=1.0

)

**2. UAV to Satellite Communication (OFDMA, Ku-Band)**

* **Approach:**  
  Orthogonal Frequency Division Multiple Access (OFDMA) is used in the satellite uplink. The Ku band is divided into 9 subchannels (subcarriers); each subchannel is randomly assigned to UAVs at every time slot. UAVs that are assigned a slot communicate with satellites using their allocated frequency.
* **Rate Calculation:**  
  In sagin\_env.py, the same compute\_rate\_general() function is called, but with parameters tailored for UAV-to-satellite links:
  + **Bandwidth:** 1 MHz per slot (subchannel)
  + **Transmit Power:** 10 W (UAV high-end)
  + **Carrier Frequency:** 12 GHz (Ku-band)
  + **Gains:** 10,000 (both UAV and satellite, representing 40 dB directional antennas)
  + **Noise:** 1e-13 (ultra low noise for satellite link)
  + **Fading:** Dynamic, time-varying (cycling through a list of fading states per slot)
* **Call Example:**

rate, ok = compute\_rate\_general(

sender\_pos=uav\_pos, # UAV 3D position

receiver\_pos=sat\_pos, # Satellite position

bandwidth=1e6,

P\_tx=10.0,

fc=12e9,

G\_tx=10000,

G\_rx=10000,

noise=1e-13,

fading=fading # Dynamic, per-slot fading value

)

* **Fading Model:**  
  Satellite links use a time-varying fading parameter. At each time step, a different fading value is selected (e.g., 1.0, 3.46, 5.03), simulating channel quality variations:
* self.fading\_states = [1.0, 3.46, 5.03]
* fading = self.fading\_states[timestep % len(self.fading\_states)]

**Summary Table**

| **Link** | **Method** | **Bandwidth** | **Power** | **Frequency** | **Gains** | **Noise** | **Fading** | **Access** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| IoT → UAV | TDMA | 5 MHz | 0.1 W | 2 GHz | 1/10 | 1e-9 | 1.0 (fixed) | Sequential |
| UAV → Satellite | OFDMA | 1 MHz | 10 W | 12 GHz | 10000/10000 | 1e-13 | Dynamic | Slot-based |

**Breief:** We use a unified function (compute\_rate\_general) to model all wireless communication links. For IoT-to-UAV, we simulate TDMA with modest transmit power and bandwidth, appropriate for local connections. For UAV-to-satellite, we use OFDMA in the Ku-band, high-gain antennas, and strong power, with dynamic fading to reflect real satellite channel conditions. All parameter settings can be customized for further extensions, such as introducing mobility or more advanced channel models