Tổng Quan Đề Tài và Hướng Đi Nghiên Cứu

# 1. Giới thiệu

Đề tài tập trung vào việc nghiên cứu và xây dựng hệ thống phân bổ tài nguyên trong mạng SAGSINs (Satellite-Air-Ground-Sea Integrated Networks). Mục tiêu là tối ưu hiệu quả sử dụng tài nguyên mạng trong điều kiện môi trường động và phức tạp, nơi các node (satellite, UAV, ground station, sea station, end-device) có thể thay đổi vị trí, tải trọng và điều kiện truyền thông.

# 2. Hướng đi hiện tại

- Áp dụng LLM (Large Language Model) để sinh ra các kịch bản dữ liệu động, ví dụ: di chuyển UAV/Satellite, điều kiện thời tiết, sự cố mạng, và cấu hình traffic theo thời gian/khu vực.  
- Sử dụng Reinforcement Learning (RL) để huấn luyện một AI Agent có khả năng nhận đầu vào từ môi trường (do LLM sinh) và ra quyết định phân bổ tài nguyên (băng thông, tuyến truyền, ưu tiên node).  
- Lưu lại lịch sử trạng thái môi trường, quyết định của Agent và kết quả vào MongoDB để phân tích, đánh giá và tối ưu.  
- Xây dựng client/server bằng Node.js để người dùng có thể gửi yêu cầu, quan sát trạng thái mạng, và theo dõi quyết định của AI.

# 3. Kiến trúc hệ thống

Hệ thống gồm các thành phần chính:  
1. LLM Engine: sinh ra các kịch bản mô phỏng động.  
2. RL Agent: nhận dữ liệu từ LLM, ra quyết định phân bổ tài nguyên.  
3. MongoDB: cơ sở dữ liệu lưu trạng thái, quyết định, và lịch sử huấn luyện.  
4. Node.js Client/Server: cung cấp giao diện giao tiếp với người dùng và kết nối với AI Agent.

# 4. Kế hoạch thực hiện

Giai đoạn 1: Nghiên cứu cơ sở lý thuyết  
- Tìm hiểu về kiến trúc SAGSINs, các yếu tố ảnh hưởng đến phân bổ tài nguyên.  
- Học các thuật toán RL (Q-learning, DQN, PPO, A3C, ...).  
- Hiểu cách dùng LLM để sinh dữ liệu mô phỏng.  
  
Giai đoạn 2: Xây dựng mô hình cơ bản  
- Tạo bộ class đại diện cho các node trong mạng (Satellite, UAV, Ground Station, Sea Station, End-device).  
- Tích hợp LLM để sinh dữ liệu động.  
- Xây dựng RL Agent rule-based ban đầu.  
  
Giai đoạn 3: Huấn luyện và tối ưu  
- Huấn luyện RL Agent trên dữ liệu LLM sinh.  
- Lưu kết quả vào MongoDB, phân tích log.  
  
Giai đoạn 4: Tích hợp hệ thống  
- Xây dựng Node.js server và client giao tiếp với Flask API.  
- Hiển thị trạng thái, kịch bản và quyết định của AI.  
  
Giai đoạn 5: Đánh giá và mở rộng  
- Đánh giá hiệu quả (throughput, latency, fairness).  
- Mở rộng kịch bản và độ phức tạp của môi trường.

# 5. Kết luận

Việc kết hợp LLM để sinh kịch bản động cùng RL để học chính sách phân bổ tài nguyên là một hướng đi mới mẻ và tiềm năng trong nghiên cứu SAGSINs. MongoDB đảm bảo khả năng lưu trữ và phân tích, trong khi Node.js giúp triển khai ứng dụng thực tiễn. Đề tài này vừa có giá trị học thuật vừa mang tính ứng dụng cao.

# Các thư viện Python sử dụng

* - \*\*flask\*\*: Web framework nhẹ để xây dựng API.
* - \*\*flask-cors\*\*: Hỗ trợ CORS để client Node.js gọi API từ domain khác.
* - \*\*pymongo\*\*: Kết nối và thao tác với MongoDB.
* - \*\*dnspython\*\*: Cần thiết khi dùng MongoDB Atlas (kết nối qua URI).
* - \*\*requests\*\*: Gọi API hoặc test nội bộ.
* - \*\*numpy\*\*: Xử lý dữ liệu số, vector, ma trận.
* - \*\*pandas\*\*: Phân tích và xử lý dữ liệu dạng bảng.
* - \*\*matplotlib\*\*: Vẽ biểu đồ trực quan hóa dữ liệu.
* - \*\*seaborn\*\*: Thư viện vẽ biểu đồ nâng cao, đẹp hơn matplotlib.
* - \*\*networkx\*\*: Mô phỏng và vẽ graph topology của mạng SAGSINs.
* - \*\*torch\*\*: Framework AI/ML, dùng để train RL agent hoặc LLM.
* - \*\*tensorflow\*\*: Một framework AI/ML khác (tuỳ chọn, nếu không dùng PyTorch).
* - \*\*transformers\*\*: Dùng Hugging Face Transformers để gọi hoặc fine-tune LLM.
* - \*\*stable-baselines3\*\*: Thư viện RL phổ biến, cung cấp PPO, DQN, A2C, SAC, v.v.
* - \*\*gymnasium\*\*: Xây dựng môi trường RL (state, action, reward).
* - \*\*openai\*\*: Gọi OpenAI API nếu dùng LLM online.
* - \*\*anthropic\*\*: Gọi Claude API nếu dùng Anthropic LLM.
* - \*\*sag\_env\*\*: Thư viện tuỳ chỉnh (do nhóm phát triển) định nghĩa môi trường RL cho SAGSINs (cần nếu muốn mô phỏng riêng).

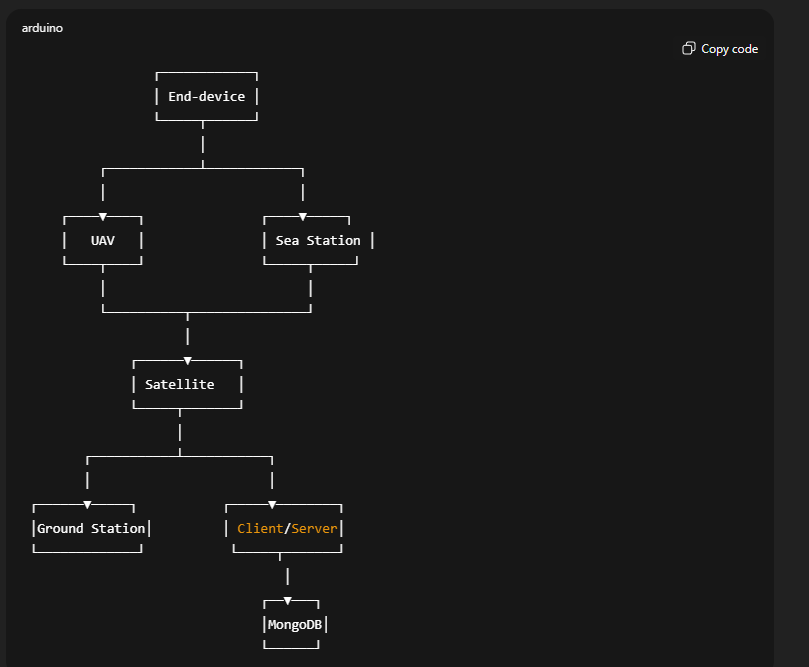
# Sơ đồ và Workflow

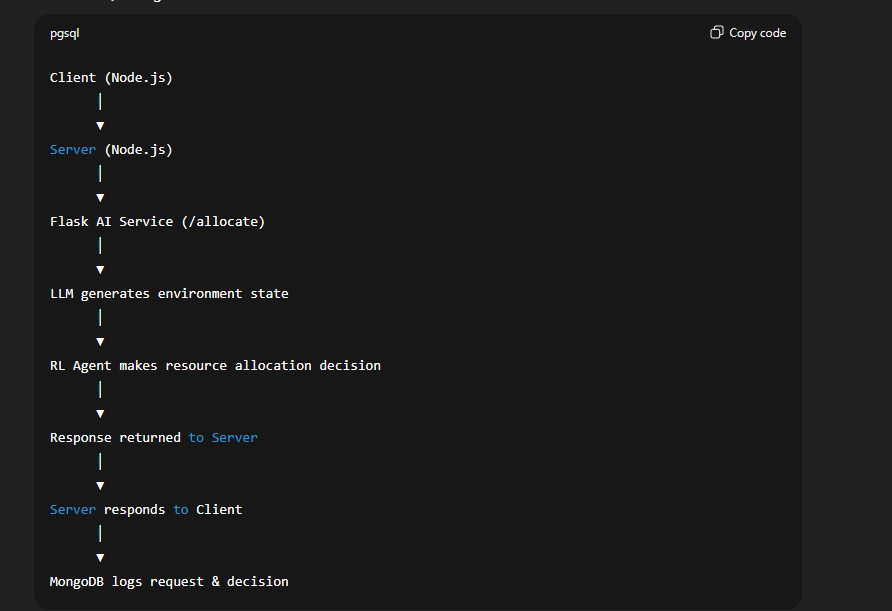
## 1. Sơ đồ tương tác giữa các đối tượng trong mạng SAGSINs

Các đối tượng chính trong mạng SAGSINs bao gồm:  
- Satellite (Vệ tinh)  
- UAV (Thiết bị bay không người lái)  
- Ground Station (Trạm mặt đất)  
- Sea Station (Trạm biển)  
- End-device (Thiết bị người dùng)  
  
Luồng tương tác điển hình:  
1. End-device gửi yêu cầu tài nguyên/kết nối.  
2. UAV và Sea Station đóng vai trò relay chuyển tiếp dữ liệu đến Satellite hoặc Ground Station.  
3. Satellite và Ground Station xử lý, định tuyến, tối ưu phân phối tài nguyên.  
4. Quyết định phân phối tài nguyên được phản hồi lại End-device.

## 2. Workflow hệ thống (LLM + RL + MongoDB + Node.js)

Workflow tổng thể của hệ thống đề xuất:  
  
1. Client (Node.js) gửi request tài nguyên (ví dụ: băng thông, latency) đến Server.  
2. Server chuyển request đến Python Flask AI Service (/allocate).  
3. LLM sinh kịch bản dữ liệu môi trường (sự kiện, traffic, điều kiện thời tiết).  
4. RL Agent sử dụng dữ liệu này để ra quyết định phân phối tài nguyên.  
5. Flask Service trả lại quyết định phân phối cho Server, từ đó phản hồi Client.  
6. MongoDB ghi log toàn bộ lịch sử request và decision.  
7. RL Agent có thể truy xuất dữ liệu lịch sử từ MongoDB để huấn luyện (training loop).





**Satellite Relay Mode (Chế độ truyền tiếp vệ tinh)** là một phương thức kết nối trong đó:

* **Người dùng (user terminals)** hoặc các thiết bị chuyên dụng sẽ kết nối đến **vệ tinh** thông qua các **trạm người dùng vệ tinh (satellite user station)**.
* Vệ tinh lúc này hoạt động như một **máy phát trung gian** (relay transmitter), chuyển tiếp dữ liệu từ người dùng đến mạng không gian (space-based network) và sau đó đến mạng mặt đất (land-based network).
* Ở phía mặt đất, dữ liệu có thể được chuyển qua các **trạm mặt đất (ground station)** hoặc tích hợp với các hệ thống **Wi-Fi** để phân phối cho nhiều người dùng khác.

Mô hình này **thích hợp cho những nơi có nhiều người dùng tập trung, hoặc có tuyến di chuyển rõ ràng** (ví dụ: đường sắt, khu khai thác, dàn khoan dầu, sự kiện tập trung đông người). Nó có thể **thay thế truyền dẫn cáp quang** trong nhiều trường hợp, đặc biệt khi việc kéo cáp không khả thi.  
Hiện nay, Satellite Relay Mode được ứng dụng nhiều trong:

* **Phủ sóng đặc thù** (vùng sâu vùng xa, biên giới, sa mạc, hải đảo).
* **Thông tin liên lạc khẩn cấp** (cứu hộ thiên tai, quân sự, hàng hải, hàng không).

### Thông tin bổ sung về ****Satellite Relay Mode****

* **Ưu điểm**:
  + Không phụ thuộc hạ tầng cáp quang.
  + Triển khai nhanh ở khu vực khó tiếp cận.
  + Khả năng kết nối ổn định trong tình huống khẩn cấp.
* **Nhược điểm**:
  + Độ trễ cao hơn so với cáp quang (do tín hiệu phải truyền lên xuống vệ tinh).
  + Chi phí thiết bị và băng thông cao.
  + Bị ảnh hưởng bởi điều kiện thời tiết (mưa, bão).

### Phân biệt ****User Station**** và ****Ground Station****

1. **User Station (Trạm người dùng)**:
   * Quy mô nhỏ, thường gắn với **thiết bị đầu cuối** của người dùng hoặc nhóm người dùng.
   * Ví dụ: anten vệ tinh gắn trên xe, tàu, máy bay, hoặc trạm VSAT (Very Small Aperture Terminal) trong doanh nghiệp.
   * Mục đích chính: **truy cập vào vệ tinh** để gửi/nhận dữ liệu.
2. **Ground Station (Trạm mặt đất)**:
   * Quy mô lớn, đặt cố định tại mặt đất, thường do **nhà mạng hoặc tổ chức vận hành vệ tinh** quản lý.
   * Chức năng: **kết nối vệ tinh với mạng lõi (core network)** hoặc Internet.
   * Đây là **cửa ngõ chính** đưa dữ liệu từ vệ tinh vào hệ thống viễn thông toàn cầu.

👉 Có thể hiểu: **User Station = điểm truy cập của người dùng**, còn **Ground Station = trung tâm điều phối kết nối vệ tinh với hạ tầng mặt đất**.

### Nếu có ****Sea Station (Trạm biển)**** thì sao?

* Trong trường hợp người dùng ở trên biển (tàu cá, du thuyền, giàn khoan):
  + **Sea Station** hoạt động tương tự **User Station**, nhưng thiết kế đặc biệt để chịu được môi trường biển (muối, gió, rung lắc).
  + Thường trang bị **ăng-ten định hướng tự động (stabilized antenna)** để luôn hướng về vệ tinh, ngay cả khi tàu di chuyển.
  + Vai trò: **trạm trung gian cho người dùng trên biển kết nối đến vệ tinh**, sau đó dữ liệu được chuyển về Ground Station hoặc các điểm kết nối khác.

👉 Tóm lại:

* **User Station** = thiết bị đầu cuối của người dùng (trên đất liền hoặc biển).
* **Ground Station** = trạm lớn trên đất liền, nối mạng vệ tinh với Internet/mạng viễn thông.
* **Sea Station** = một dạng User Station đặc biệt, dùng cho môi trường biển.

Bạn có muốn mình vẽ thêm **sơ đồ ASCII** hoặc **workflow hình ảnh** để minh họa rõ mối quan hệ giữa **User Station – Ground Station – Sea Station – Satellite** không?

## ****LEO và GEO là gì?****

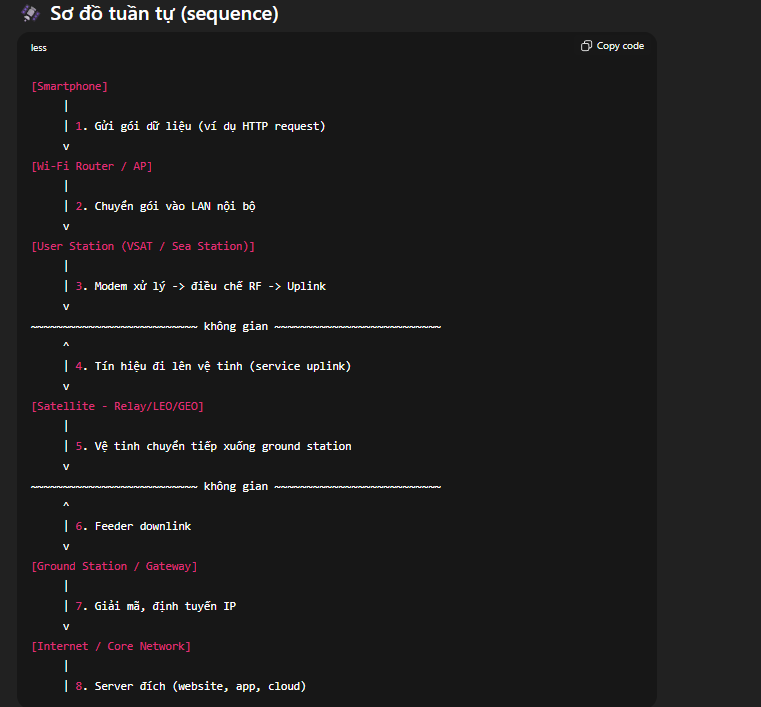
### 1. ****LEO (Low Earth Orbit – Quỹ đạo Trái Đất Thấp)****

* **Độ cao**: khoảng **200 km – 2.000 km** so với mặt đất (phổ biến ~500–1.200 km).
* **Thời gian bay quanh Trái Đất**: rất nhanh, khoảng **90–120 phút**/vòng.
* **Độ trễ (latency)**: rất thấp, chỉ ~20–50 ms (gần giống mạng mặt đất).
* **Đặc điểm**:
  + Cần **nhiều vệ tinh** (hàng trăm → hàng ngàn) để phủ sóng liên tục (ví dụ: Starlink ~4.000 vệ tinh).
  + Mỗi vệ tinh chỉ phủ sóng một vùng nhỏ, và **di chuyển liên tục**, nên user phải **handover** qua vệ tinh khác khi nó bay qua.
* **Ứng dụng**: Internet băng rộng (Starlink, OneWeb), quan sát Trái Đất, trinh sát, IoT vệ tinh.

### 2. ****GEO (Geostationary Earth Orbit – Quỹ đạo địa tĩnh)****

* **Độ cao**: khoảng **35.786 km** so với mặt đất (gấp ~60 lần LEO).
* **Đặc điểm đặc biệt**: vệ tinh quay cùng tốc độ với Trái Đất → **luôn đứng yên trên một điểm xích đạo**.
* **Độ trễ (latency)**: cao hơn nhiều, thường **~500–600 ms RTT thực tế**.
* **Đặc điểm**:
  + Chỉ cần **3 vệ tinh GEO** là phủ sóng gần như toàn cầu (trừ vùng cực).
  + Vệ tinh “cố định trên trời” nên anten người dùng có thể chĩa cố định, **không cần handover**.
* **Ứng dụng**: Truyền hình vệ tinh (Vinasat, VTV), thông tin liên lạc quân sự, phủ sóng diện rộng cho Internet vệ tinh truyền thống.







# 📝 Giải thích chi tiết từng bước

1. **Smartphone → Wi-Fi Router**
   * Smartphone chỉ thấy Wi-Fi như mạng bình thường (SSID, WPA2/3).
   * Không biết phía sau là vệ tinh hay cáp quang.
2. **Wi-Fi Router → User Station**
   * Router kết nối LAN tới modem VSAT (có thể tích hợp chung).
   * Modem xử lý dữ liệu thành dạng sóng RF uplink.
3. **User Station → Satellite**
   * Anten VSAT (cố định hoặc Sea Station) phát tín hiệu lên vệ tinh qua băng Ku/Ka/L.
4. **Satellite Relay**
   * GEO: bent-pipe (chỉ chuyển tiếp tần số).
   * LEO: có thể route qua Inter-Satellite Link trước khi xuống gateway.
5. **Satellite → Ground Station**
   * Vệ tinh phát feeder downlink đến trạm gateway trên đất liền.
6. **Ground Station → Internet**
   * Gateway giải điều chế tín hiệu, đưa vào router, NAT, firewall, sau đó kết nối backbone Internet.
7. **Internet → Server đích**
   * Server xử lý request (ví dụ: tải website).
8. **Phản hồi về Smartphone**
   * Ngược lại theo đúng chuỗi: server → gateway → vệ tinh → user station → Wi-Fi → smartphone.

📌 **Tóm lại**:  
Smartphone **chỉ nhìn thấy Wi-Fi**, nhưng thật ra gói tin của nó đi qua **router → user station → vệ tinh → ground station → Internet**.  
Với người dùng, trải nghiệm giống Wi-Fi bình thường, nhưng thực tế có **một chặng không gian (space hop)**.

Trong **mạng SAGSINS (Space–Air–Ground–Sea Integrated Networks)**, **không phải tất cả kết nối đều bắt buộc phải đi qua vệ tinh**. Vệ tinh chỉ là **một thành phần** trong nhiều lớp kết nối.

### 1. Các trường hợp ****có vệ tinh tham gia****

* **Phủ sóng vùng xa, hải đảo, đại dương** → không có cáp quang, không có trạm mặt đất gần.
* **Kết nối di động toàn cầu** (ví dụ máy bay, tàu biển, cứu hộ khẩn cấp).
* **Backhaul thay thế cáp quang** (relay mode) khi triển khai mạng tạm thời hoặc vùng đặc thù.

### 2. Các trường hợp ****không cần đi qua vệ tinh****

Trong **môi trường có hạ tầng ground station (trạm mặt đất) hoặc trạm gốc mặt đất (terrestrial base station)** thì dữ liệu có thể:

* **User → Ground Station (hoặc trạm 5G/4G/6G)** → Core Network → Internet.  
  → Đây là cách mà phần lớn smartphone/Wi-Fi hoạt động hiện nay.
* **User → Sea Station** (trạm nổi trên biển, kết nối bằng cáp ngầm hoặc microwave link) → Core Network.
* **Airborne Station (UAV, HAPS)** → Ground Station → Internet, mà không cần qua vệ tinh.

### 3. Khi nào chọn ****vệ tinh**** và khi nào chọn ****ground station trực tiếp****

* Nếu **có ground station gần** (cách vài km – vài chục km) → đi thẳng đến ground station sẽ tối ưu (độ trễ thấp, băng thông cao).
* Nếu **không có ground station trong vùng phủ sóng** → phải đi qua vệ tinh (LEO, MEO, GEO).
* Trong mô hình **SAGSINS tối ưu bằng GENAI server**, hệ thống có thể:
  + Quyết định **tuyến đi qua vệ tinh** hay **trực tiếp ground station** dựa trên độ trễ, tải mạng, QoS.
  + Luân phiên sử dụng: ví dụ ban ngày dùng vệ tinh LEO (độ trễ thấp), ban đêm dùng ground station nếu gần bờ biển.

Đúng rồi, bạn có thể coi **kết nối trực tiếp Ground Station** là một **kịch bản bổ sung (kịch bản thứ 3)** bên cạnh:

1. **Satellite Relay Mode**
   * User → User Station → Satellite (chỉ relay) → Ground Station → Internet
   * Vệ tinh chỉ là **ống dẫn trung gian**, xử lý chính vẫn ở mặt đất.
2. **Terminal – Satellite Direct Connection**
   * User Terminal (thiết bị đầu cuối) → Satellite (có khả năng xử lý tín hiệu 3GPP NTN) → Ground Station → Internet
   * Vệ tinh **chủ động xử lý và quản lý kết nối**, không cần User Station.
3. **Direct Ground Station Connection (Terrestrial Mode trong SAGSINS)**
   * User Terminal → Ground Station (hoặc trạm 5G/6G, Wi-Fi AP, Sea Station) → Core Network/Internet
   * **Không đi qua vệ tinh** → độ trễ thấp, chi phí thấp, nhưng vùng phủ hẹp (chỉ trong phạm vi hạ tầng mặt đất).

### 📊 So sánh nhanh 3 kịch bản:

| **Chế độ kết nối** | **Đặc điểm** | **Ưu điểm** | **Hạn chế** |
| --- | --- | --- | --- |
| **Satellite Relay Mode** | Vệ tinh làm “cáp quang trên trời” | Phủ sóng rộng, triển khai nhanh | Độ trễ cao hơn ground, chi phí vệ tinh |
| **Terminal–Satellite Direct** | Thiết bị kết nối trực tiếp vệ tinh | Phù hợp di động toàn cầu, không cần User Station | Cần thiết bị đặc biệt, vệ tinh phức tạp |
| **Direct Ground Station** | User kết nối thẳng trạm mặt đất | Độ trễ thấp, rẻ, phổ biến | Giới hạn trong phạm vi hạ tầng mặt đất, không phù hợp vùng xa |

📌 Kết luận: **Direct Ground Station** thực chất là chế độ **cơ bản nhất** (terrestrial), vốn chiếm phần lớn lưu lượng mạng hiện nay. Trong khi đó **Satellite Relay** và **Direct Satellite** chỉ là các giải pháp **bổ sung** khi ground station không đủ.









## 1️⃣ Vị trí đặt AI trong mạng SAGSINS

### a) ****Trung tâm GENAI Server (Data Center)****

* **Vị trí**: nằm ở **ground station hoặc cloud server** với khả năng thu thập dữ liệu toàn mạng.
* **Chức năng**:
  + Thu thập dữ liệu từ user terminals, user station, ground/sea stations, satellites.
  + Phân tích traffic, độ trễ, băng thông, mật độ người dùng.
  + Tính toán **tối ưu hóa phân bổ tài nguyên, chọn kịch bản kết nối phù hợp**.

### b) ****Edge AI (nếu cần)****

* **Vị trí**: gần trạm mặt đất (ground station / sea station) hoặc trên một số vệ tinh LEO có khả năng xử lý onboard.
* **Chức năng**:
  + Ra quyết định tức thời cho các user trong vùng phụ trách.
  + Giảm tải cho GENAI server trung tâm.
  + Ví dụ: quyết định user nào nên chuyển từ **ground station sang satellite** khi traffic tăng.

## 2️⃣ Thời điểm AI tham gia

| **Thời điểm** | **Hoạt động của AI** |
| --- | --- |
| **Trước khi kết nối (Pre-connection)** | - Chọn kịch bản tối ưu: Relay Mode, Direct Satellite, Ground Station - Chọn thiết bị / trạm để kết nối - Dự đoán băng thông cần cấp cho user |
| **Trong quá trình kết nối (Runtime)** | - Giám sát mạng: đo độ trễ, băng thông, lỗi packet - Quyết định **chuyển đổi mode (handover)** nếu cần: ví dụ từ Ground → Satellite khi mất tín hiệu |
| **Sau kết nối (Post-connection / Learning)** | - Phân tích logs để tối ưu thuật toán - Điều chỉnh dự đoán traffic cho lần kết nối sau - Cập nhật mô hình AI cho điều phối tài nguyên thông minh hơn |

## 3️⃣ Luồng dữ liệu giữa các thực thể

User Device ↔ User Station / Direct ↔ Satellite / Ground Station

│ │

└──────────► GENAI Server ◄───┘

(thu thập dữ liệu, phân tích, quyết định)

* **User Device**: gửi thông tin yêu cầu dịch vụ, vị trí, tải hiện tại.
* **User Station / Ground Station / Satellite**: gửi thông tin trạng thái mạng, băng thông, độ trễ.
* **GENAI Server**: tính toán, ra quyết định kết nối / chuyển mode / phân bổ tài nguyên.
* **User Device**: nhận chỉ dẫn (mode kết nối, tần số, beam, route).

## 4️⃣ Quy trình tối ưu hóa

1. **Thu thập dữ liệu thời gian thực**: traffic, QoS, vị trí user.
2. **Phân loại user & kịch bản khả thi**: dựa trên vị trí, thiết bị, vùng phủ sóng.
3. **Chạy thuật toán tối ưu**: chọn kịch bản kết nối (Relay, Direct, Ground), chọn beam vệ tinh, trạm gần nhất.
4. **Gửi lệnh đến terminal/trạm**: user chuyển mode, kết nối với trạm/vệ tinh được chỉ định.
5. **Giám sát & điều chỉnh liên tục**: dynamic handover nếu môi trường thay đổi.

### ✅ Kết luận

* **GENAI server là trung tâm ra quyết định**, nên đặt ở **ground data center**, kết nối tới toàn bộ network entities.
* **Tham gia ở tất cả các giai đoạn**: trước khi kết nối (chọn kịch bản), trong kết nối (hậu kiểm & chuyển mode), và sau kết nối (học & tối ưu thuật toán).
* Với dự án nhỏ, bạn có thể **chỉ cần 1 GENAI server trung tâm**, không cần triển khai Edge AI, vẫn tối ưu được lựa chọn kịch bản và phân bổ tài nguyên.

Ok, mình sẽ trình bày chi tiết về **thuật toán tối ưu và các dữ liệu cần theo dõi** trong dự án SAGSINS nhỏ, để bạn có thể triển khai GENAI server. Mình sẽ phân tách theo **dữ liệu đầu vào, xử lý, quyết định, và dữ liệu trả về**.

## 1️⃣ Dữ liệu cần theo dõi

### a) ****Thông tin từ người dùng (User Device)****

* **Vị trí địa lý**: GPS / LBS (để xác định vùng phủ sóng).
* **Thiết bị**:
  + Loại terminal (smartphone, IoT, VSAT user station).
  + Khả năng hỗ trợ Direct Satellite (NTN, 5G/6G).
* **Tải hiện tại**:
  + Traffic demand (Mbps)
  + Loại dịch vụ: video, IoT sensor, voice, cloud.
* **QoS yêu cầu**:
  + Độ trễ tối đa (latency)
  + Băng thông tối thiểu (throughput)
  + Độ tin cậy (reliability)

### b) ****Thông tin từ trạm / vệ tinh****

* **Load hiện tại**:
  + Số user đang kết nối
  + Băng thông đã sử dụng
  + Beam/vùng đang quá tải
* **Thông tin trạng thái mạng**:
  + Độ trễ trung bình
  + Packet loss, jitter
  + Tình trạng trạm (up/down)
* **Thông tin môi trường** (nếu có):
  + Thời tiết ảnh hưởng vệ tinh/ground station
  + Vùng biển / núi cản sóng

### c) ****Thông tin AI cần xử lý****

* Bảng trạng thái tất cả user, trạm, vệ tinh.
* Dự đoán traffic demand trong vài phút tới.
* Tính toán độ ưu tiên user (QoS, dịch vụ quan trọng).
* Xác định **kịch bản tối ưu**: Ground Direct / Satellite Relay / Direct Satellite.
* Quyết định **beam/tần số/băng thông** cho mỗi user.
* Ra quyết định **handover** khi user di chuyển hoặc khi load thay đổi.

## 2️⃣ Thuật toán tối ưu (gợi ý cho dự án nhỏ)

### a) ****Bước 1 – Phân loại user****

if user trong vùng phủ sóng Ground Station:

mode\_candidate = Ground Direct

elif user terminal hỗ trợ NTN:

mode\_candidate = Direct Satellite

else:

mode\_candidate = Satellite Relay

### b) ****Bước 2 – Tối ưu hóa phân bổ băng thông / beam****

* Đầu vào: load trạm, băng thông yêu cầu, QoS.
* **Thuật toán**: weighted allocation hoặc simple greedy:

for each beam/trạm:

sort user theo độ ưu tiên QoS

allocate bandwidth until beam/trạm full

remaining users -> chọn kịch bản dự phòng (handover)

### c) ****Bước 3 – Quyết định Handover****

* Kiểm tra định kỳ (ví dụ 1–5 giây):

for each user:

if mode\_current == Ground Direct and signal < threshold:

switch\_to = Satellite Relay or Direct Satellite

elif mode\_current == Satellite Relay and user vào vùng Ground Station:

switch\_to = Ground Direct

### d) ****Bước 4 – Learning / cập nhật****

* Lưu log traffic, độ trễ, handover, QoS.
* Sử dụng AI (GENAI) dự đoán traffic tiếp theo.
* Cập nhật trọng số phân bổ băng thông & beam.

## 3️⃣ Thông tin gửi đi và trả về

| **Thông tin** | **Gửi đi** | **Xử lý AI** | **Trả về** |
| --- | --- | --- | --- |
| User → AI | vị trí, thiết bị, traffic demand, QoS yêu cầu | Phân tích khả năng kết nối, ưu tiên user, dự đoán load | Kịch bản tối ưu, beam / trạm / tần số được gán, lệnh handover nếu cần |
| Trạm / Vệ tinh → AI | trạng thái trạm, load, độ trễ, packet loss, trạng thái beam | Phân bổ băng thông, dự đoán quá tải, xác định handover | Điều chỉnh beam, tần số, gán user vào trạm / vệ tinh |
| AI → User / Terminal | – | – | Lệnh kết nối: mode, trạm/vệ tinh, tần số, beam |
| AI → Trạm / Vệ tinh | – | – | Lệnh phân bổ băng thông, beam steering, priority user |

## 4️⃣ Tổng kết quy trình AI

1. **Thu thập dữ liệu**: từ user và trạm/vệ tinh.
2. **Phân loại user & kịch bản**: Ground Direct, Satellite Relay, Direct Satellite.
3. **Phân bổ tài nguyên**: beam, tần số, băng thông.
4. **Handover & điều chỉnh động**: khi user di chuyển hoặc load thay đổi.
5. **Học & cải thiện**: dự đoán traffic, tối ưu thuật toán cho lần tiếp theo.