Network System Capstone
@CS.NYCU

2025.03.13 Lab2 Beamforming with NS3

Instructor: Kate Ching-Ju Lin (林靖茹)

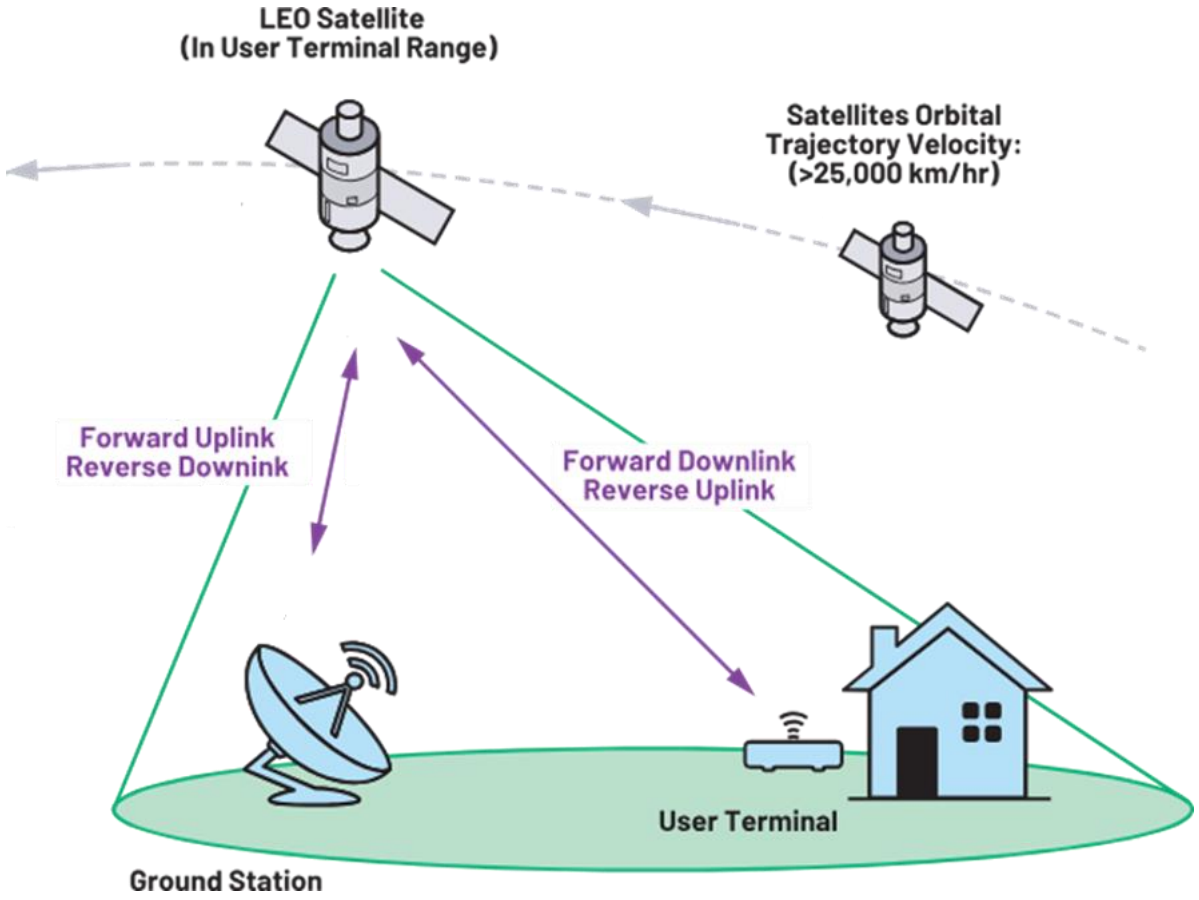
Deadline: 2025.04.03 23:59

Agenda

- Lab Overview
- Tasks
- Report & Result
- Submission

Lab Overview

- In this lab, we are going to write an NS3 program to simulate LEO communications



Lab Overview

- **Limitation of the LEO module:**
  - Constant link data rate without considering path loss and Tx Gain

- **Goal of this lab:**
  - Leverage lab 1 to find the beamforming steering vector and the corresponding Tx gain
  - Read this Tx Gain and calculate the Rx power in NS3
  - Calculate the resulting SNR and data rate
  - Set the link data rate accordingly

Lab Overview

- **Tasks (Week 1):**
  - Install Virtual Box
  - Install Ubuntu
  - Install NS3
  - Install LEO module
  - Configure and test NS3/LEO module
  - Execute the example code (calculate_delay.cc)
  - Modify the link data rate

Lab Overview

- **Tasks (Week 2-3):**
  - Output node locations
  - Execute lab1 (bf.m) to find Tx Gain
  - Read Tx gain and calculate the Rx power in NS3
  - Calculate SNR and data rate
  - Modify the link data rate

Task1: Topology Configuration

- TODO: Modify `calculate_delay.cc`

1. Set up topology
   - Tx: (6.06692, 73.0213)
   - Rx: (6.06692, 73.0213)
   - Satellite: (6.06692, 73.0213)

2. Convert to (x, y, z) coordinates
   - Use `GetObject<MobilityModel>()`

3. Output (x, y, z) to .txt file
   - Run after `initial_position()` to avoid offset

Task2: Calculate Tx Gain

- TODO: Modify `bf.m`

1. Load .txt file
   - Set Tx location as user location
   - Set Rx location as satellite location

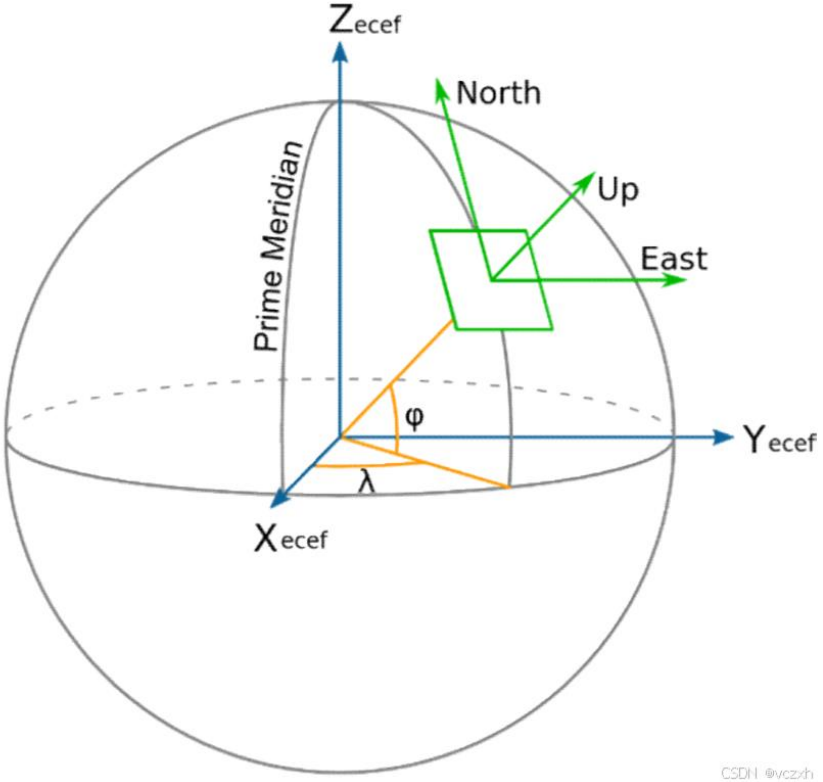
2. Change codebook to [0:5:90]
   - Convert horizontal angle to elevation angle

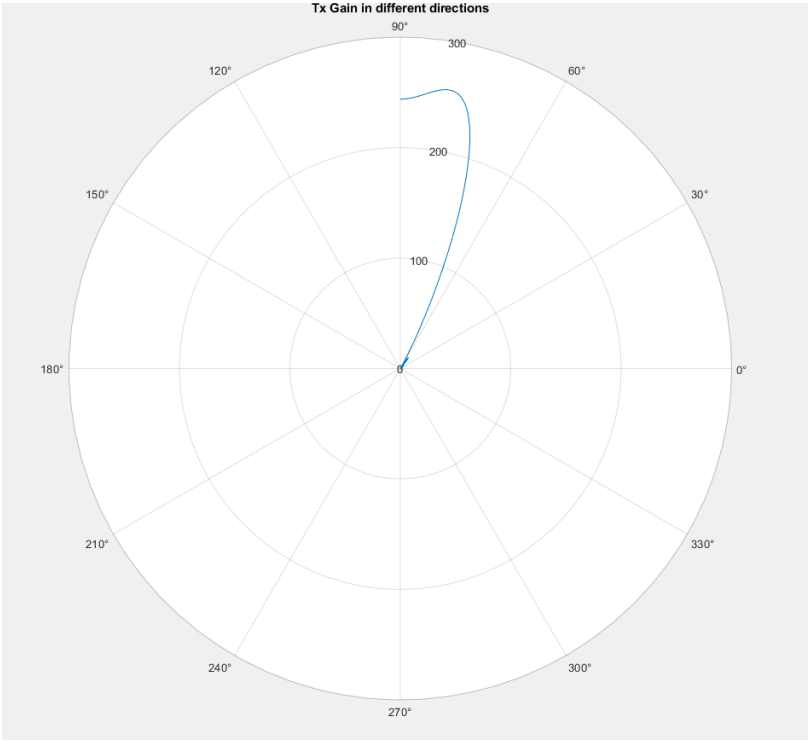
3. Update antenna phase offset
   - `psi = 2 * pi * d * sin(theta)`

4. Compute Tx gain based on elevation angle

Task2: Geodetic Misalignment

- Identical geodetic coordinates (lat/lon) do not guarantee **vertical alignment** in the Cartesian (XYZ) space





Task2: Calculate Pathloss

- TODO: Modify `bf.m`

5. Compute pathloss based on Tx gain
   - Use Friis' free space model

6. Output pathloss to .txt file

Task3: Calculate Rx Power (1/2)

- TODO: Modify `leo-propagation-loss-model.h` in `/ns-3.35/contrib/leo/model`

1. Change `DoCalcRxPower()` inheritance from private to public

- TODO: Modify `leo-propagation-loss-model.cc`

1. Load .txt file into `DoCalcRxPower()`
2. Compute Rx power
   - Replace `m_freeSpacePathLoss`

Task3: Calculate Rx Power (2/2)

- TODO: Modify `propagation-loss-model.h` in `/ns-3.35/src/propagation/model`

1. Change `DoCalcRxPower()` inheritance from private to public

- TODO: Modify `mock-channel.cc` in `/ns-3.35/contrib/leo/model`

1. Change code line 187 `rxPower < -60.0`

Task4: Transmission Configuration

- TODO: Modify `calculate_delay.cc`

1. Set up transmission configuration
   - Bandwidth = 2MHz
   - Noise = -110dBm
   - Tx power = 105.9dBm

2. Call `DoCalcRxPower()` to get Rx power

3. Calculate SNR_{dB}
   - Hint: Convert SNR_{dB} to linear scale (Reference)

4. Calculate data rate

Task5: Compute E2E Delay

- TODO: Modify `calculate_delay.cc`

1. Update DataRate with the calculated value
   - Hint: Use `setGndDeviceAttribute()`

2. Output end-to-end delay
   - Hint: Same steps as Task1 last week
   - Output format:

```
Packet average end-to-end delay is 2.5s
```

Report and Result Format

Report (50%)

- In PDF format
- Explain how you implement your lab step by step for each commit version
- Briefly explain how each answer was obtained
- Maximum of 4 pages

Result (30%)

- In PDF format
- Numerical results, figures and your observations
- Maximum of 3 pages

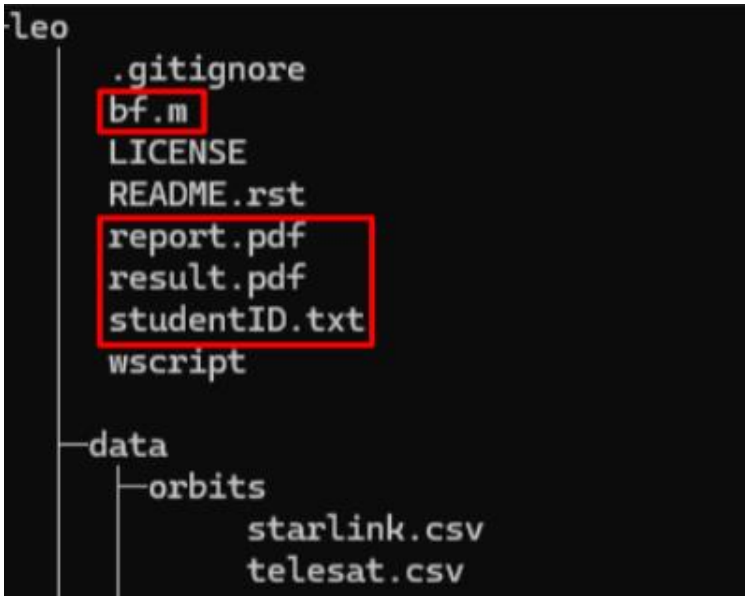
Notice: The example outputs and figures are for reference only

Result

- Given different user positions (latitude, longitude), answer the following questions:
  - User positions:
    - (20, 0)
    - (6.06692, 73.0213)
    - (-16.0634, 142.29)
  - Questions:
    - Q1: Calculate the Euclidean distance between the user and the satellite
    - Q2: Compute the path loss
    - Q3: Compare Rx power w/ and w/o beamforming
    - Q4: Evaluate the SNR & data rate
    - Q5: Estimate the end-to-end delay

Submission

- Add studentID.txt
- Add your own studentID to the file (as in lab1)
- File structure:



- Notice: You will get penalty with wrong file structure and naming

Commit to GitHub

- Add all modified and new files to the staging area

```
# git add <FILENAME>
$ git add .
```

- Record changes to the repository

```
# git commit -m "<COMMIT_MESSAGE>"
$ git commit –m "Initial commit"
```

- Upload to GitHub

```
$ git push
```

Due

- **Apr. 3 (Thu.) 23:59, 2025**
- Don’t need to submit to E3
- Commit **your files** to your GitHub repository
  - Should have at least **3 commits** (Initial, work in progress, final)
  - One version should be at least **1 day** after another
- **Notice: You will get penalty with wrong file structure and naming**

Grading Policy

- Grade
  - Code correctness – 20%
  - Report – 50%
  - Result – 30%
- Late Policy
  - (Your score) * 0.8^D, where D is the number of days overdue
- Cheating Policy
  - Academic integrity: Homework must be your own – cheaters share the score
  - Both the cheaters and the students who aided the cheater equally share the score