Network System Capstone @CS.NYCU

2025.03.13 Lab2 Beamforming with NS3

Instructor: Kate Ching-Ju Lin (林靖茹)

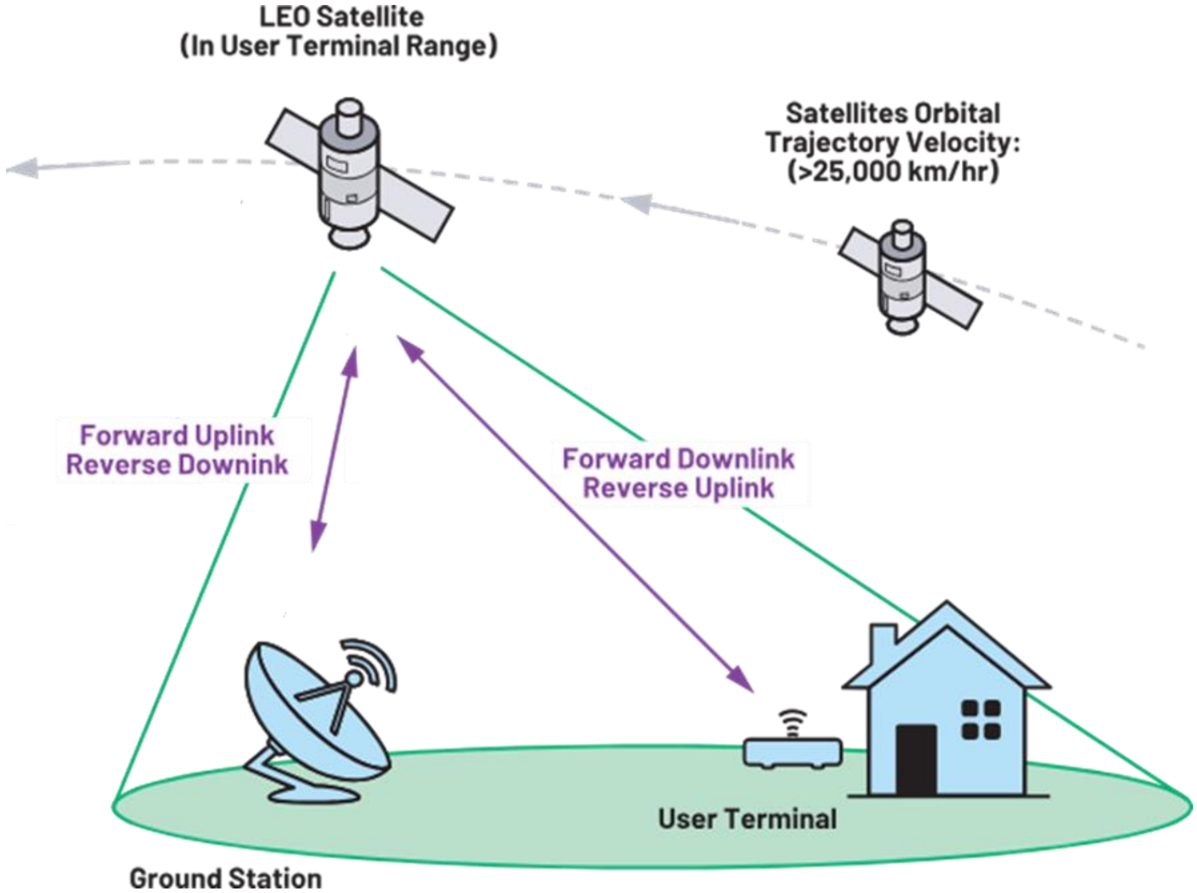
Deadline: 2025.04.10 23:59

Agenda

- Lab Overview
- Tasks 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

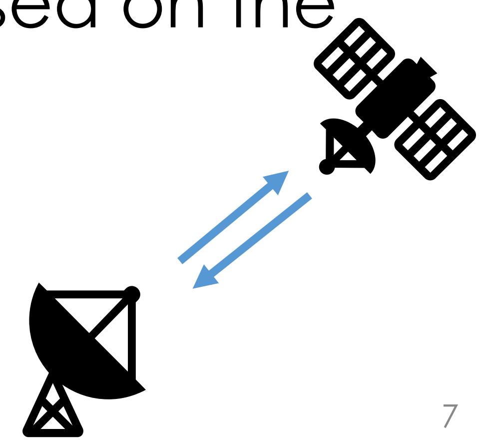
- **Tasks (Week 1):**
  - Install Virtual Box
  - Install Ubuntu
  - Install NS3
  - Install LEO module
  - Configure and test NS3/LEO module
  - Execute the example code (calculate_delay.cc)
  - Modify the link data rate

Lab Overview

- **Tasks (Week 2-4):**
  - Output node coordinate
  - Execute lab1 (bf.m) to find Tx Gain in MATLAB
  - Read pathloss and calculate the Rx power in NS3
  - Calculate SNR and data rate in NS3
  - Modify the link data rate

Tasks Overview

1. Setup topology
2. Calculate elevation angle θ
3. Find the optimal beam θ^* for θ
4. Find the Tx gain based on elevation angle θ and the identified optimal beam θ^*
5. Find the pathloss based on Tx gain and output it to the .txt file
6. Calculate the sending rate based on the receiving power
7. Calculate end-to-end latency



Task Input and Output

- Scenario: 1 ground station (GS) and 1 satellite (SAT)
  - The entire system includes multiple satellites
- Input: GS/SAT positions (latitude, longitude)
  - (20, 0)
  - (6.06692, 73.0213)
  - (-16.0634, 142.29)
- Output
  - Euclidean distance between GS and SAT
  - Path loss between GS and SAT
  - Rx power w/ and w/o beamforming
  - SNR & data rate w/ and w/o beamforming
  - End-to-end delay w/ and w/o beamforming

Task1: Topology Configuration

- TODO: Modify `calculate_delay.cc`

1. Set up node positions (latitude, longitude)
   - Source (GS): (6.06692, 73.0213)
   - Destination (GS): (6.06692, 73.0213)
   - SAT: (6.06692, 73.0213)
2. Convert (latitude, longitude) to (x, y, z) coordinates
   - Use `GetObject<MobilityModel>()`
3. Output (x, y, z) to .txt file
   - The .txt file will be read and processed by MATLAB
   - Run after `initial_position()` to avoid offset

Task2: Calculate Tx Gain

- TODO: Modify `bf.m`

1. Load .txt file to read node coordinates
   - Set Tx coordinate as GS coordinate (x_1, y_1, z_1)
   - Set Rx coordinate as SAT coordinate (x_2, y_2, z_2)

2. Use the GS/SAT coordinates to find the elevation angle θ
   - Assume GS = (x_1, y_1, z_1), SAT = (x_2, y_2, z_2)
   - Compute Horizontal Distance

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

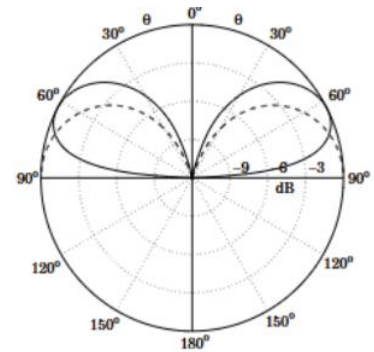
   - Compute Elevation Angle

$$\theta = \tan^{-1}\left(\frac{d}{z_2 - z_1}\right)$$

Task2: Calculate Tx Gain

- TODO: Modify `bf.m`

3. Find the optimal beam θ^* based on the elevation angle θ from codebook [0:5:90]



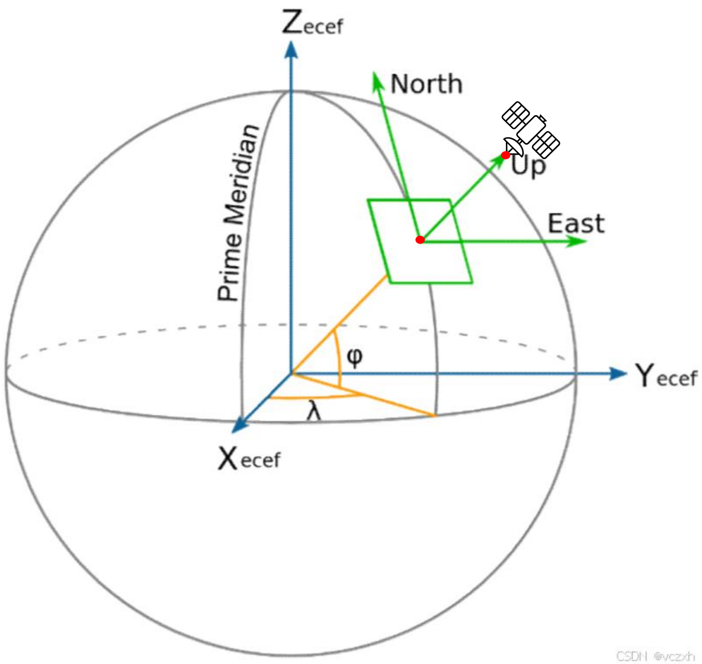
4. Update antenna phase offset
   - `psi = 2 * pi * d * sin(theta)`

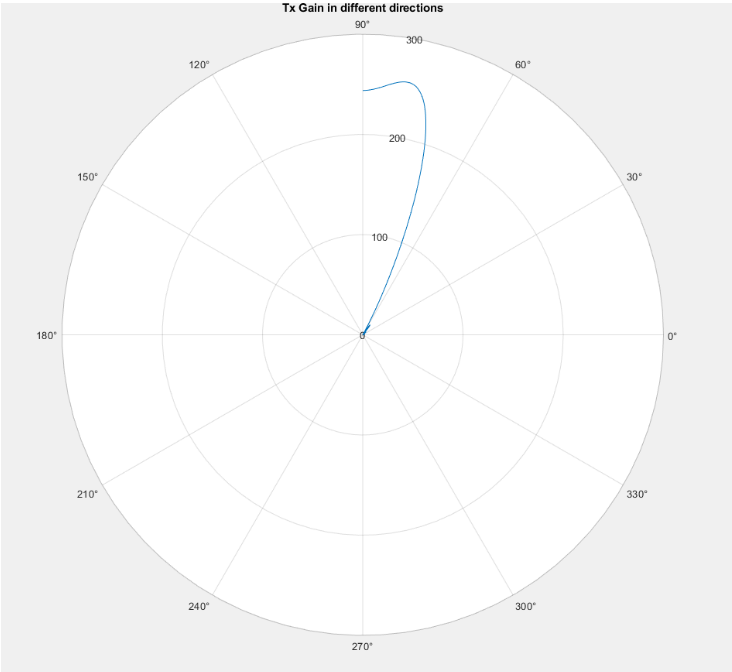
5. Find the Tx gain based on elevation angle θ and the identified optimal beam θ^*
   - Use the beam pattern of θ^* (lab1)

6. Use Tx gain to calculate pathloss and output it to .txt file

Note!

- Even if GS and SAT have the same location (lat/lon), this location will correspond to different 3D coordinates since their heights are different





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. In `DoCalcRxPower()`, read .txt file to get the pathloss
2. Compute Rx power based on the pathloss logged in .txt file
   - Hint: 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 < -120.0`

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 SNR ratio (S_{watt} / N_{watt}) (Reference)

4. Calculate data rate based on Shannon capacity

Task5: Compute E2E Delay

- TODO: Modify `calculate_delay.cc`

1. Change link data rate settings

```
utCh.SetGndDeviceAttribute("DataRate", StringValue("8kbps"));
```

Update DataRate with the calculated value

```
utNet.Get(25)->GetObject<MockNetDevice>()->SetDataRate(DataRate(oss.str()));
utNet.Get(0)->GetObject<MockNetDevice>()->SetDataRate(DataRate("1Gbps"));
```

2. Output end-to-end delay
   - Hint: Same steps as Task1 last week
   - Last received time - first transmitted time for the **same sequence number**
   - 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

Result

- Given different user positions (latitude, longitude), answer the following questions:
  - User positions:
    - (20, 0)
    - (6.06692, 73.0213)
    - (-16.0634, 142.29)
  - Questions:
    - Compare the end-to-end delay with beamforming and without beamforming
    - Link data rate setting without beamforming:

```
utCh.SetGndDeviceAttribute("DataRate", StringValue("8kbps"));
```

    - Link data rate setting with beamforming:

```
utNet.Get(25)->GetObject<MockNetDevice>()->SetDataRate(DataRate(oss.str()));
utNet.Get(0)->GetObject<MockNetDevice>()->SetDataRate(DataRate("1Gbps"));
```

Submission

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

```
leo
 │      .gitignore
 │      bf.m
 │      LICENSE
 │      README.rst
 │      report.pdf
 │      result.pdf
 │      studentID.txt
 │      wscript
 │
 ├─data
 │  ├─orbits
 │  │       starlink.csv
 │  │       telesat.csv
```

- 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. 10 (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