Network System Capstone
@CS.NYCU

2025.04.24: Lab4

Instructor: Kate Ching-Ju Lin (林靖茹)

Agenda

- **Lab Overview**
- Tasks
- Report & Result
- Submission

Lab Overview

- In this lab, we are going to write an **NS-3** program to simulate satellite-ground station association

- Goal of this lab:
  - Simulate the results obtained from lab 3
  - Set the link data rate based on network.graph and association results (network.xxx.out)
  - Transmit packets from each ground station to its associated satellite
  - Calculate the transmission time

Agenda

- Lab Overview
- **Tasks**
- Report & Result
- Submission

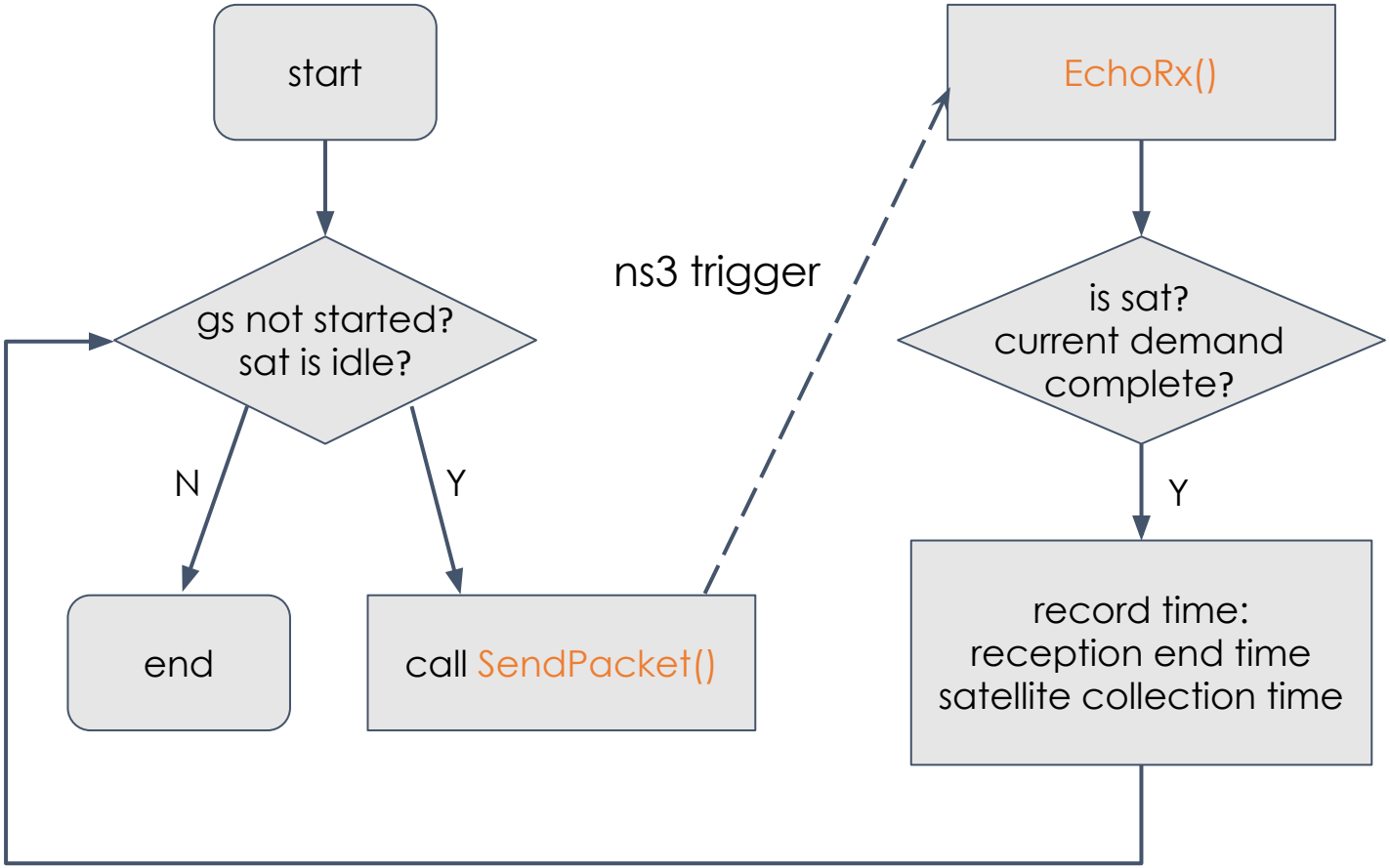
Task Overview (1/2)

- Read the output of lab3 to get link data rates and LEO-station association results
- Schedule data transmissions between ground stations and satellites
- Output total data collection time and other results

- **Notice**
  - Each satellite can receive data from only one ground station at a time
  - Each satellite switches to the next ground station only after the current transmission finishes

Task Overview (2/2)

- Schedule data transmissions according to the flowchart



Program Input / Output (1/2)

- Input
  - network.graph
  - network.ortools.out / network.greedy.out
- Output
  - lab4.ortools.out / lab4.greedy.out
  - Line 1: total collection time(second)
  - Each of next *#satellite* lines: satellite id and its collection time(second)
    - List satellite IDs in ascending order
  - Each of next *#ground station* lines: ground station id, its corresponding transmission start time(second) and reception end time(second)
    - List ground station IDs in ascending order

Program Input / Output (2/2)

- Output Format

```
total_collection_time

satellite_id collection_time
satellite_id collection_time
....

ground_station_id trans_start_time recept_end_time
ground_station_id trans_start_time recept_end_time
....
```

Pre-process (1/3)

- Modify ns-3-allinone/ns-3.35/contrib/leo/model/leo-propagation-loss-model.cc DoCalcRxPower() function
  - Comment out all the code inside this function and return 0
  - Since it is not necessary in lab4 anymore

Pre-process (2/3)

- Add following code in ns-3-allinone/ns-3.35/contrib/leo/helper/ground-node-helper.h (public)

```
void Add (NodeContainer &ground, const LeoLatLong &location);
```

- Add following code in ns-3-allinone/ns-3.35/contrib/leo/helper/ground-node-helper.cc

```
void LeoGndNodeHelper::Add (NodeContainer &ground, const
LeoLatLong &location) {
  Vector pos = GetEarthPosition (location);
  Ptr<ConstantPositionMobilityModel> mob =
CreateObject<ConstantPositionMobilityModel> ();
  mob->SetPosition (pos);
  Ptr<Node> node = m_gndNodeFactory.Create<Node> ();
  node->AggregateObject (mob);
  ground.Add (node);
}
```

Pre-process (3/3)

- Download lab4 repository
  - Includes two files: leo-lab4.cc, network.graph

```
$ cd ns-3-allinone/ns-3.35/contrib/leo/examples
$ git init
$ git remote add origin
git@github.com:NYCU-NETCAP2025/lab4-<GITHUB_ID>.git
$ git pull origin main
$ git branch -M main
```

- Generate new network.ortools.out and network.greedy.out using the provided network.graph

Compile & Run

- Compile configuration: add the following code in ns-3-allinone/ns-3.35/contrib/leo/examples/wscript

```
obj = bld.create_ns3_program('leo-lab4', ['core', 'leo',
'mobility', 'aodv'])

obj.source = 'leo-lab4.cc'
```

- Run: execute leo-lab4.cc

```
$ cd ns-3-allinone/ns-3.35
$ ./waf --run "leo-lab4 --inputFile=network.ortools.out
--outputFile=lab4.ortools.out"

$ ./waf --run "leo-lab4 --inputFile=network.greedy.out
--outputFile=lab4.greedy.out"
```

Description of Functions

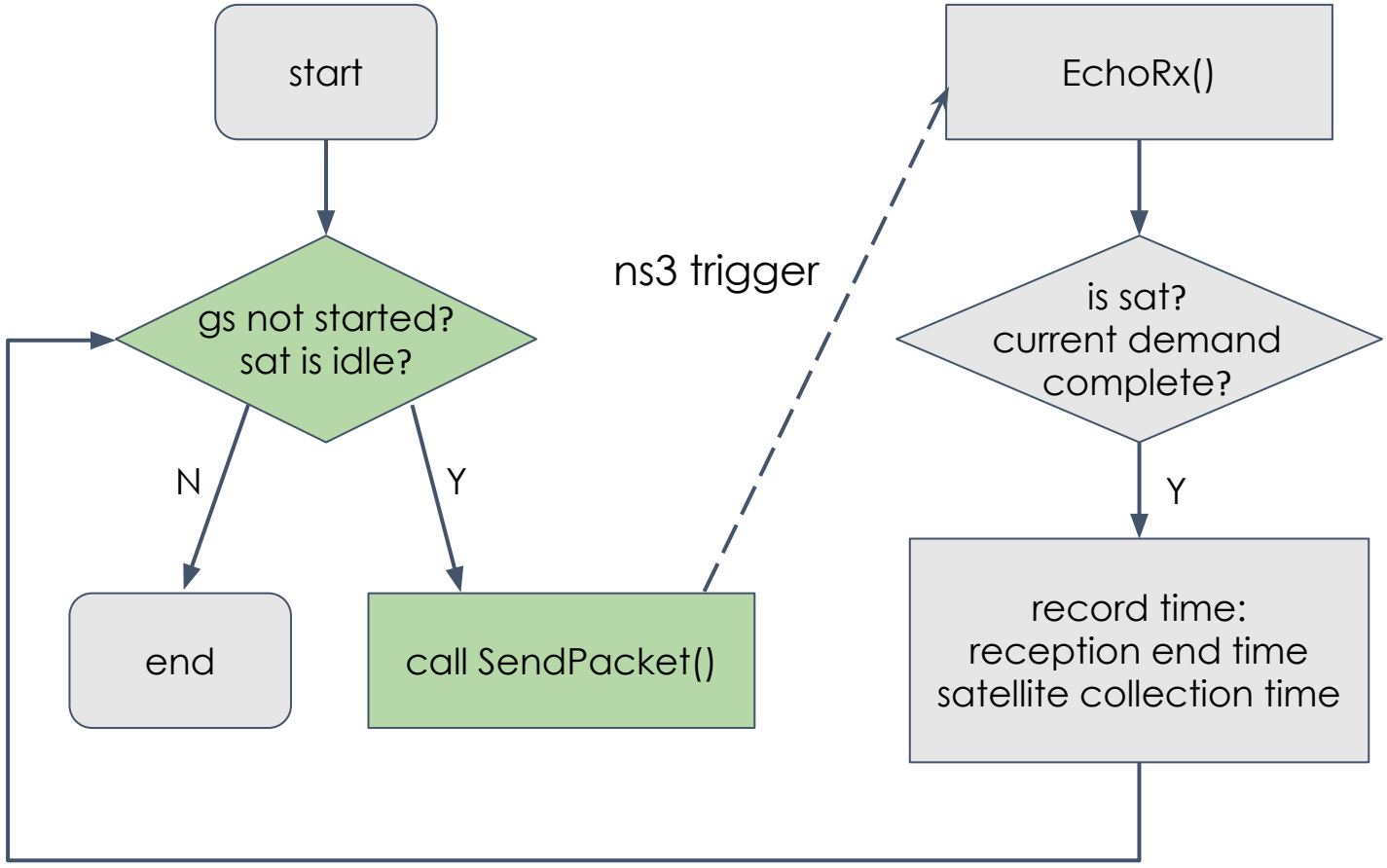
- void SendPacket(int gsId, int satId);
  - Send packet from ground station gsId to satellite satId
  - Complete this function in **Task 2**

- static void EchoRx(std::string context, const Ptr< const Packet > packet, const TcpHeader &header, const Ptr< const TcpSocketBase > socket);
  - This function will be triggered when any node receives a packet
  - Complete this function in **Task 3**

Task 1: Input File

- Get the **data rate** between each ground station and satellite based on network.graph
- Get the **association results** between ground stations and satellites based on the --inputFile provided in the command line (network.ortools.out or network.greedy.out)

Task 2: Send Packet (1/3)



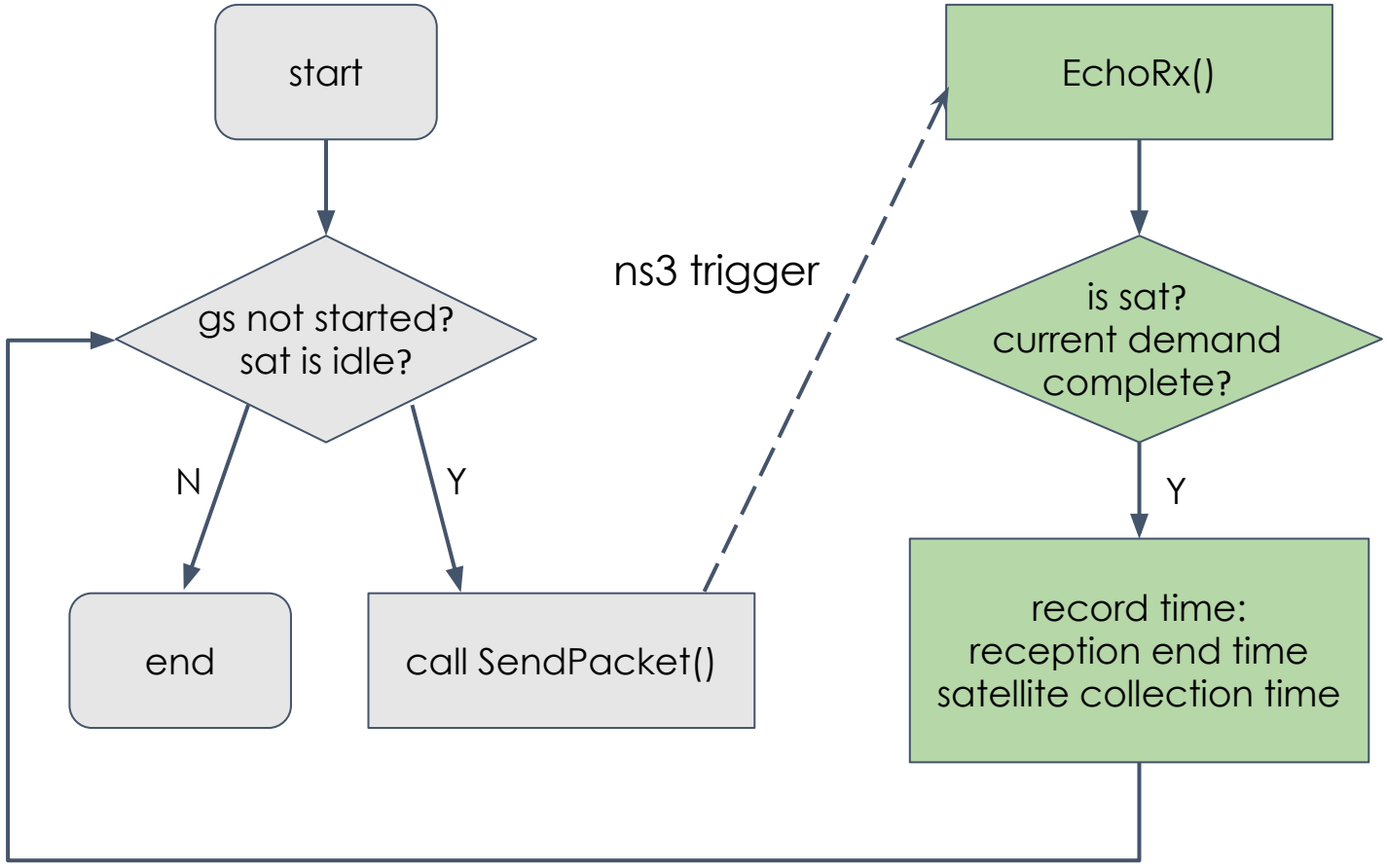
Task 2: Send Packet (2/3)

- Task 2.1: Complete SendPacket(int gsId, int satId) function
  - Set link data rate (bidirectional)
  - Get the satellite IP and use BulkSendHelper to send packet from gsId to satId
    - Use TCP protocol
    - Set MaxBytes 125000(Byte)
    - Set SendSize 512(Byte)
  - Record the transmission start time of this ground station

Task 2: Send Packet (3/3)

- Task 2.2: Call SendPacket() in main()
  - Call SendPacket() if the ground station hasn't started transmitting and the satellite is idle
  - Ground stations with smaller IDs transmit first

Task 3: EchoRx() (1/3)



Task 3: EchoRx() (2/3)

- If the node is a satellite and current transmission has finished uploading all the data
  - Record reception end time when a ground station completes transmissions
  - Record satellite collection time when all associated ground stations complete their transmissions
  - Call SendPacket() again if any ground station has not started transmitting and the satellite is idle

Task 3: EchoRx() (3/3)

- Hint
  - Use GetNodeId(context) to get node id
    - Node IDs are assigned with satellites first, then ground stations
  - Use GetTotalRx() to check how many bits the satellite has received
    - Refer to the usage in leo-bulk-send-example.cc

Task 4: Output File

- Output file: lab4.ortools.out / lab4.greedy.out
  - Check output format on page 7

- **Notice**
  - The collection time will not be the same as the solutions in network.xxx.out due to protocol overhead in the simulation
  - You don't need to follow the TA's code exactly - just make sure your output format is correct

Agenda

- Lab Overview
- Tasks
- **Report & Result**
- Submission

Report & Result

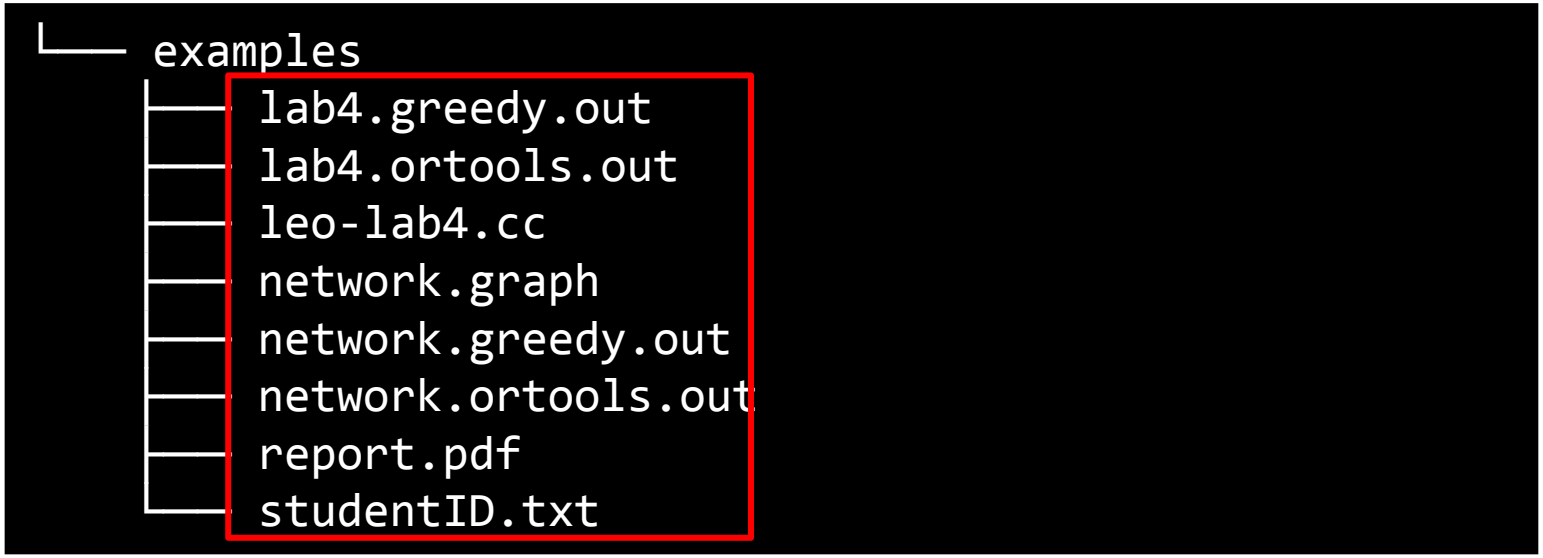
- Name as report.pdf
- Explain how you implement your lab step by step for each commit version
- Questions
  1. Compare lab4.greedy.out and lab4.ortools.out
  2. Explain why the collection time will not be the same as the solutions in network.xxx.out
  3. Explain the meaning of MaxBytes and SendSize in BulkSendHelper
  4. Adjust the value of MaxBytes and observe the changes in transmission time

Agenda

- Lab Overview
- Tasks
- Report & Result
- **Submission**

Submission

- Add your own studentID to studentID.txt (same as lab1)
- Push only the following specified files to GitHub
  - Please **do not** include any other files



Due

- **May. 8 (Thu.) 23:59, 2025**
- Don't need to submit to E3
- Commit **your flies** to your Github repository
  - Should have at least **3 commits** by **yourself** (commit by github-classroom[bot] is not included)
  - 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