Network System Capstone
@CS.NYCU

2025.05.22: Lab6

Instructor: Kate Ching-Ju Lin (林靖茹)

Agenda

- **Lab Overview**
- Tasks
- Report
- Submission

Lab Overview

- In this lab, we are going to write an **NS-3** program to simulate load balance transmissions

- Goal of this lab:
  - **Task 1**: Set the routing path (given in an input file) with *nix-vector-routing* package
  - **Task 2**: Check whether the routing path is set correctly by tracing the Mac layer TxRx
  - **Task 3**: Compare the total throughput of two different sets of paths

- **Notice**
  - You don't need to follow the TA's code exactly - just make sure your output format is correct

Routing Package in NS3

There are many routing packages in NS3

- StaticRouting
- OLSR(Optimized Link State Routing)
- **AODV(Ad Hoc On Demand Distance Vector)**
- ListRouting
- GlobalRouting
- **NixVectorRouting**

Nix-vector-routing (NVR)

Nix-Vector is a **source routing** technique used in NS-3

- Each packet carries a compact vector that encodes the path
- Use **BFS** as the default routing path algorithm
- Source code: /ns-3.35/src/nix-vector-routing/

If you are curious about why we choose this package, see the final page

Task 1 : Modify this package to specify a user-defined path

Task 1: Modify *nix-vector-routing*

Goal

Modify the *NVR* package to determine the routing based on a user-defined path (given in the input file)

Challenge

- Modify NVR to configure a user-defined path
  - Let the *NVR* read the user-defined path from an input file
  - Replace the default BFS path with the user-defined path
- *NVR* uses cache for the path with the same destination
  - Disable the cache

TODO

- Task 1.1: Modify NixVectorHelper and NixVectorRouting for setting the path file
- Task 1.2: Modify GetNixVector() for setting the path
- Task 1.3: Modify GetNixVectorInCache() and GetIpRouteInCache() for disable the cache

Task 1.1: Set the Path File

Goal: After running the following code in main(), NVR can store the paths in inputFile in its path table Table

```
InternetStackHelper stack;
Ipv4NixVectorHelper nixRouting;
nixRouting.SetPathFile(inputFile);   // todo: implement this function
stack.SetRoutingHelper(nixRouting);
stack.Install (satellites);
stack.Install (groundStations);
```

- Provide SetPathFile() in NixVectorHelper
  - store the filename
- Provide SetPaths() in NixVectorRouting
  - read the file and store the paths into Table
- Call SetPaths() when creating NixVectorRouting

Task 1.1: Set the Path File (Cont.)

- Function/variable prototype
  - NixVectorHelper
    - std::string m_pathFile
    - void SetPathFile (std::string pathFile);
  - NixVectorRouting
    - void SetPaths (std::string pathFile)
    - std::map<std::pair<int, int>, std::vector<int>> Table
      - the paths table
- Provide SetPathFile() in NixVectorHelper
  - Modify nix-vector-routing/helper/nix-vector-helper.h
    - Add member variable m_pathFile
    - Add member function SetPathFile()
  - Modify nix-vector-routing/helper/nix-vector-helper.cc
    - Implement SetPathFile(): update m_pathfile
    - Update copy constructor to also assign m_pathFile when constructing new NixVectorHelper by copy

Task 1.1: Set the Path File (Cont.)

- Provide SetPaths(pathFile) in NixVectorRouting
  - Modify nix-vector-routing/model/nix-vector-routing.h
    - Add member variable Table to store the paths
    - Add member function SetPaths(pathFile)
  - Modify nix-vector-routing/model/nix-vector-routing.cc
    - Implement SetPaths(pathFile): read the paths from the pathFile and store the paths into Table
- Call SetPaths() when creating NixVectorRouting
  - Modify nix-vector-routing/helper/nix-vector-helper.cc
    - Call SetPaths in Create function after creating NixVectorRouting object

Task 1.1: PathFile Format

- The file format for pathFile (paths1.in and paths2.in)

```
srcId dstId pathLength srcId … dstId
.
.
.
```

- Ex: A path (36, 3, 2, 5, 38)

```
36 38 5 36 3 2 5 38
```

Task 1.2: Set the Routing Path

Set the path according to the Table

Hint:

- Modify nix-vector-routing/model/nix-vector-routing.cc
- Before transmit a packet, TCP uses RouteOutput() function to find a route in NS3
- If the cache does not store the path, the GetNixVector() function will be called
- BuildNixVector() constructs the NixVector with parentVector
  - Refer to the parentVector usage in BFS()

Task 1.3: Disable the Cache

Disable the cache to prevent the SD-pairs with the same destination but different sources from using the same path

Hint:

- Modify nix-vector-routing/model/nix-vector-routing.cc
- In RouteOutput(), GetNixVectorInCache() and GetIpRouteInCache() are used for searching the routes in cache
- Set the variable/return value as the route is not in cache

Task 2: Send & Trace Packet (1/2)

Goal: Check the routing path

1. Send a packet from node 36 to node 38 based on the path in paths1.in
2. Track the packet's send/receive timestamps at each node to verify whether the path matches the one specified in paths1.in

- Task 2.1: Complete SendPacket()
  - Use BulkSendHelper and PacketSinkHelper to transmit packets from srcId to dstId
    - Use TCP protocol
    - Set MaxBytes to 512 (Byte)
    - Set SendSize to 512 (Byte)
    - Set starting simulation time to Seconds (0.0)

- Task 2.2: Call SendPacket() in main()
  - srcId=36, dstId=38

Task 2: Send & Trace Packet (2/2)

- Task 2.3: Complete EchoMacTxRx()
  - Output the send/receive timestamps on each node along the path
    - just trace the TCP packet with data payload
    - should be the same as the one in paths1.in
  - Output Format

```
<MaxTx/MaxRx> at node: <node id>, now: <time>
...
```

  - Ex

```
MacTx at node: 36, now: +1.17536e+09ns
MacRx at node: 0, now: +1.66518e+09ns
:
:
MacTx at node: 3, now: +2.17452e+09ns
MacRx at node: 38, now: +2.17852e+09ns
```

Task 3: Calculate Throughput (1/2)

Goal: Compare two different path sets (paths1.in&paths2.in)

1. Send packets between multiple SD pairs during a time period
2. Calculate the throughput

- Task 3.1: Continuously send packets
  - Set MaxBytes to 0

- Task 3.2: Call SendPacket() for 3 SD pairs in main()
  - srcId=36, dstId=38
  - srcId=37, dstId=40
  - srcId=39, dstId=41

Task 3: Calculate Throughput (2/2)

- Task 3.3: Calculate throughput
  - Use GetTotalRx() to check how many bytes the destination has received when simulation ends
  - Output Format

```
36->38: <Throughput of 36->38>
37->40: <Throughput of 37->40>
39->41: <Throughput of 39->41>
Total throughput: <Total throughput>
```

  - Ex

```
36->38: 12345
37->40: 87878
39->41: 13589
Total throughput: 113812
```

Compile & Run

- Compile configuration: add the following code in ns-3-allinone/ns-3.35/contrib/leo/examples/wscript
  - We also need nix-vector-routing package here

```
obj = bld.create_ns3_program('leo-lab6', ['core', 'leo',
'mobility', 'aodv', 'nix-vector-routing'])

obj.source = 'leo-lab6.cc'
```

- Run: execute leo-lab6.cc

```
$ cd ns-3-allinone/ns-3.35
# Task 2
$ ./waf --run="leo-lab6 --Task=2 --in="paths1.in" --out="task2.out""
# Task 3
$ ./waf --run="leo-lab6 --Task=3 --in="paths1.in" --out="task3.paths1.out""
$ ./waf --run="leo-lab6 --Task=3 --in="paths2.in" --out="task3.paths2.out""
```

Agenda

- Lab Overview
- Tasks
- **Report**
- Submission

Report

- Filename: report.pdf
- Explain how you implement your lab step by step for each commit version
- Questions
- **Q1:** Explain how parentVector in nix-vector-routing describe a path
- **Q2:** Explain why you get different total throughputs for paths1.in and paths2.in? Does congestion occurs in paths1.in and paths2.in?
- **Q3:** Please provide more experiments to clarify your answer in **Q2**. (Hint: Try to transmit each SD pair separately)

Agenda

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

Submission

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

```
.
├── leo-lab6.cc
├── nix-vector-routing
│   ├── helper
│   │   ├── nix-vector-helper.cc
│   │   └── nix-vector-helper.h
│   └── model
│       ├── nix-vector-routing.cc
│       └── nix-vector-routing.h
├── paths1.in
├── paths2.in
├── report.pdf
├── task2.out
├── task3.paths1.out
├── task3.paths2.out
└── studentID.txt
```

Due

- **June. 12th (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

How We Finish the Lab

1. How to choose suitable package to modify?

   Challenge: many routing packages in NS3

   Sol: (1) 問 ChatGpt 縮小範圍 → StaticRouting, NixVectorRouting (2) ***瘋狂*** google 找到有希望的

2. How to modify code?

   a. 有希望的 → **change BFS**: trace code to find who calls the BFS by gdb and also ctrl+f, ctrl+mouse1