Network Systems Capstone @CS.NYCU

Lab. 6: Handoff Simulation over ns3

Instructor: Kate Lin

Deadline: 2021.06.21

Objectives

In this lab, we are going to write **ns-3** program to conduct simulations and evaluate the performance of two handoff mechanisms

1. Learn how to use **ns-3**
2. Learn how to do handoff between two APs
3. Learn how to parse pcap files and analyze simulation results

TODO

1. **Handoff**: Implement threshold-based handoff for linear movement

2. **Random Walk**: Import the trajectory file to enable random walk

3. **Tracing**: Write a parser to analyze the simulation results

4. **Discussion**: Compare two algorithms according to your simulation results

Overview

NS-3 Introduction

- NS-3 is a discrete event network simulator
  - Overview of NS-3 – https://nsnam.org
  - Tutorial: https://www.nsnam.org/docs/tutorial/
- **Node**: electronic device which can connect to the network
- **Net Device**: NIC
- **Tracing**: packet logs of the simulation
  - Pcap

  -
    ```
    PcapHelperDevice::EnablePcap( filename, deviceContainer );
    ```

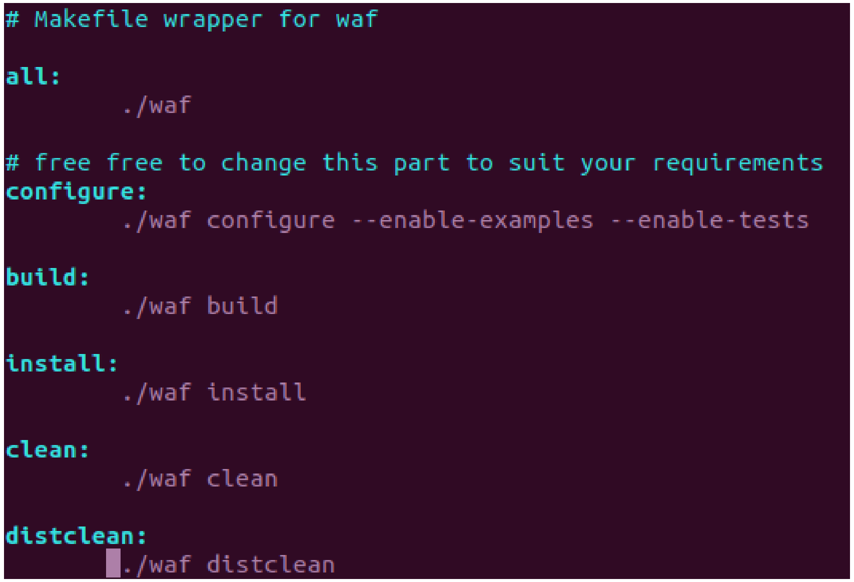
```
Example-1-0.pcap Example-2-3.pcap
```

Pcap file of 0th device on 1st node
Pcap file of 3rd device on 2nd node

Building Tool & Logger

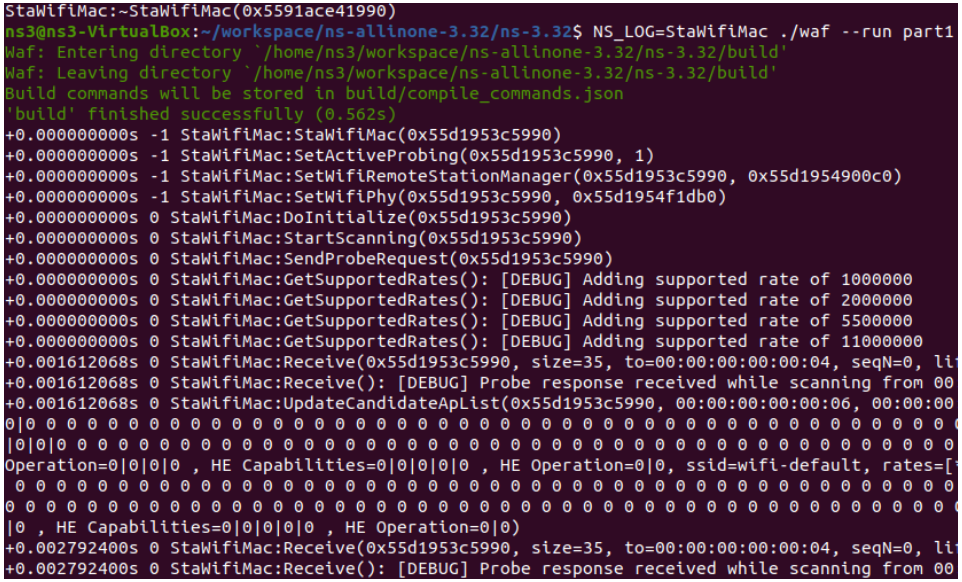
- **<u>waf</u>**: python-based building tool
- **<u>Logging module</u>**: monitor or debug the progress of simulation programs

```
$ NS_LOG=ClassName  ./waf --run YourSimulationProgram
```



```
# Makefile wrapper for waf

all:
        ./waf

# free free to change this part to suit your requirements
configure:
        ./waf configure --enable-examples --enable-tests

build:
        ./waf build

install:
        ./waf install

clean:
        ./waf clean

distclean:
        ./waf distclean
```

Using “waf” to build project



```
StaWifiMac:~StaWifiMac(0x5591ace41990)
ns3@ns3-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$ NS_LOG=StaWifiMac ./waf --run part1
Waf: Entering directory `/home/ns3/workspace/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory `/home/ns3/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.562s)
+0.000000000s -1 StaWifiMac:StaWifiMac(0x55d1953c5990)
+0.000000000s -1 StaWifiMac:SetActiveProbing(0x55d1953c5990, 1)
+0.000000000s -1 StaWifiMac:SetWifiRemoteStationManager(0x55d1953c5990, 0x55d1954900c0)
+0.000000000s -1 StaWifiMac:SetWifiPhy(0x55d1953c5990, 0x55d1954f1db0)
+0.000000000s 0 StaWifiMac:DoInitialize(0x55d1953c5990)
+0.000000000s 0 StaWifiMac:StartScanning(0x55d1953c5990)
+0.000000000s 0 StaWifiMac:SendProbeRequest(0x55d1953c5990)
+0.000000000s 0 StaWifiMac:GetSupportedRates(): [DEBUG] Adding supported rate of 1000000
+0.000000000s 0 StaWifiMac:GetSupportedRates(): [DEBUG] Adding supported rate of 2000000
+0.000000000s 0 StaWifiMac:GetSupportedRates(): [DEBUG] Adding supported rate of 5500000
+0.000000000s 0 StaWifiMac:GetSupportedRates(): [DEBUG] Adding supported rate of 11000000
+0.001612068s 0 StaWifiMac:Receive(0x55d1953c5990, size=35, to=00:00:00:00:00:04, seqN=0, li
+0.001612068s 0 StaWifiMac:Receive(): [DEBUG] Probe response received while scanning from 00
+0.001612068s 0 StaWifiMac:UpdateCandidateApList(0x55d1953c5990, 00:00:00:00:00:06, 00:00:00
0|0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
|0|0|0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Operation=0|0|0|0 , HE Capabilities=0|0|0|0|0 , HE Operation=0|0, ssid=wifi-default, rates=[
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
|0 , HE Capabilities=0|0|0|0|0 , HE Operation=0|0)
+0.002792400s 0 StaWifiMac:Receive(0x55d1953c5990, size=35, to=00:00:00:00:00:04, seqN=0, li
+0.002792400s 0 StaWifiMac:Receive(): [DEBUG] Probe response received while scanning from 00
```

Logged message would be written to stderr

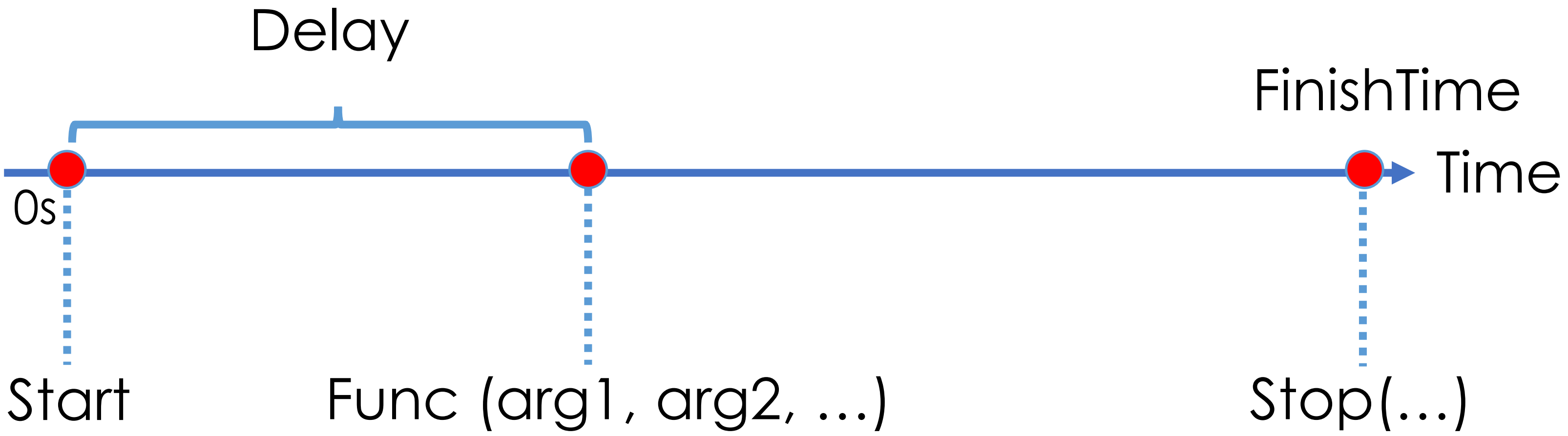
Simulator

- Schedule events during the simulation period

```
Simulator::Schedule(Delay, (*Func), arg1, arg2 ...)
```

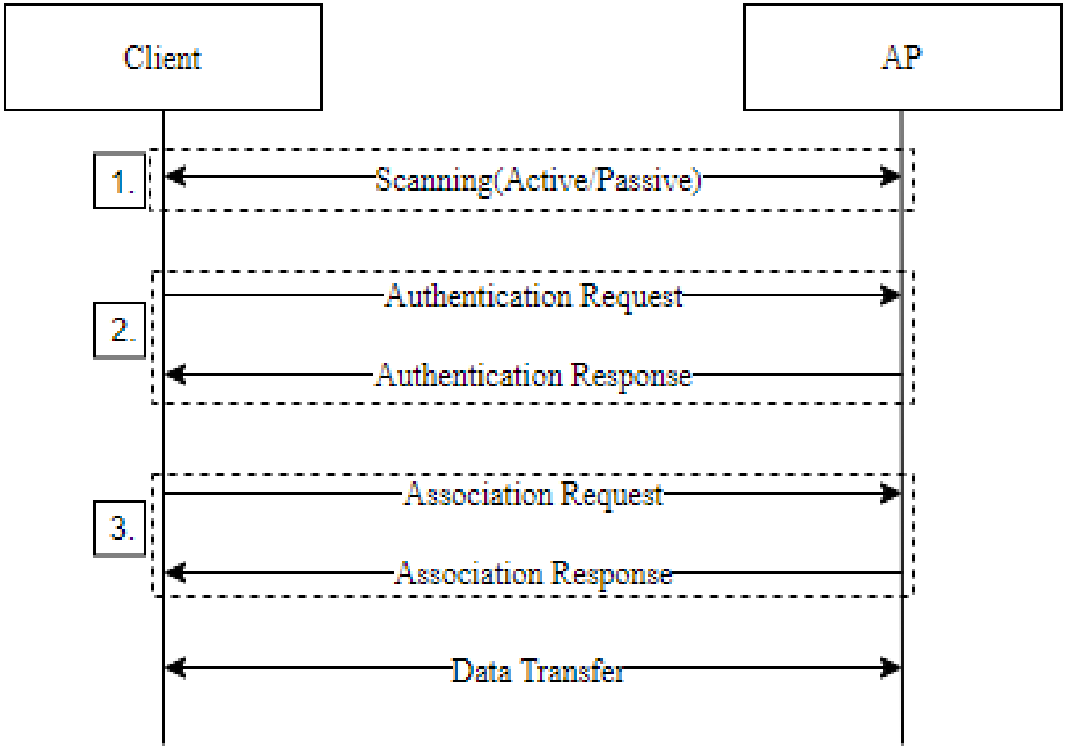
```
Simulator::Stop(FinishTime)
```

```
Simulator::Run()
```



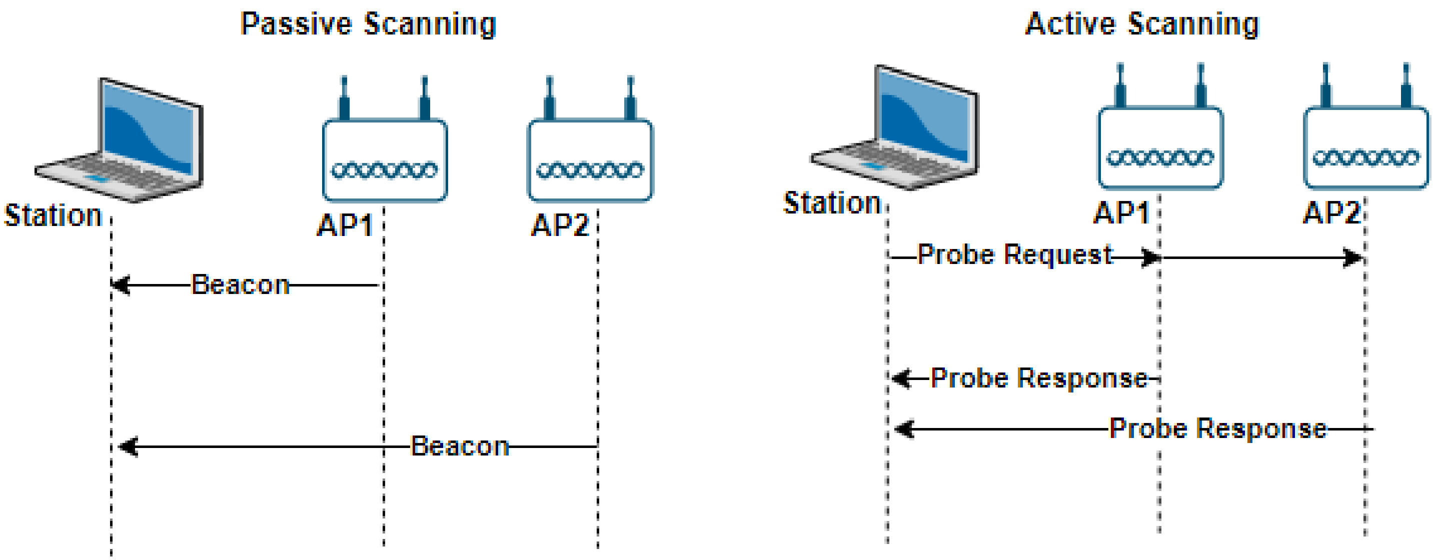
Wi-Fi Connection Process

- **Scanning**: locate visible access point
- **Authentication**: check identification, compatibility before connection
- **Association**: establish the connection



Scanning Methods

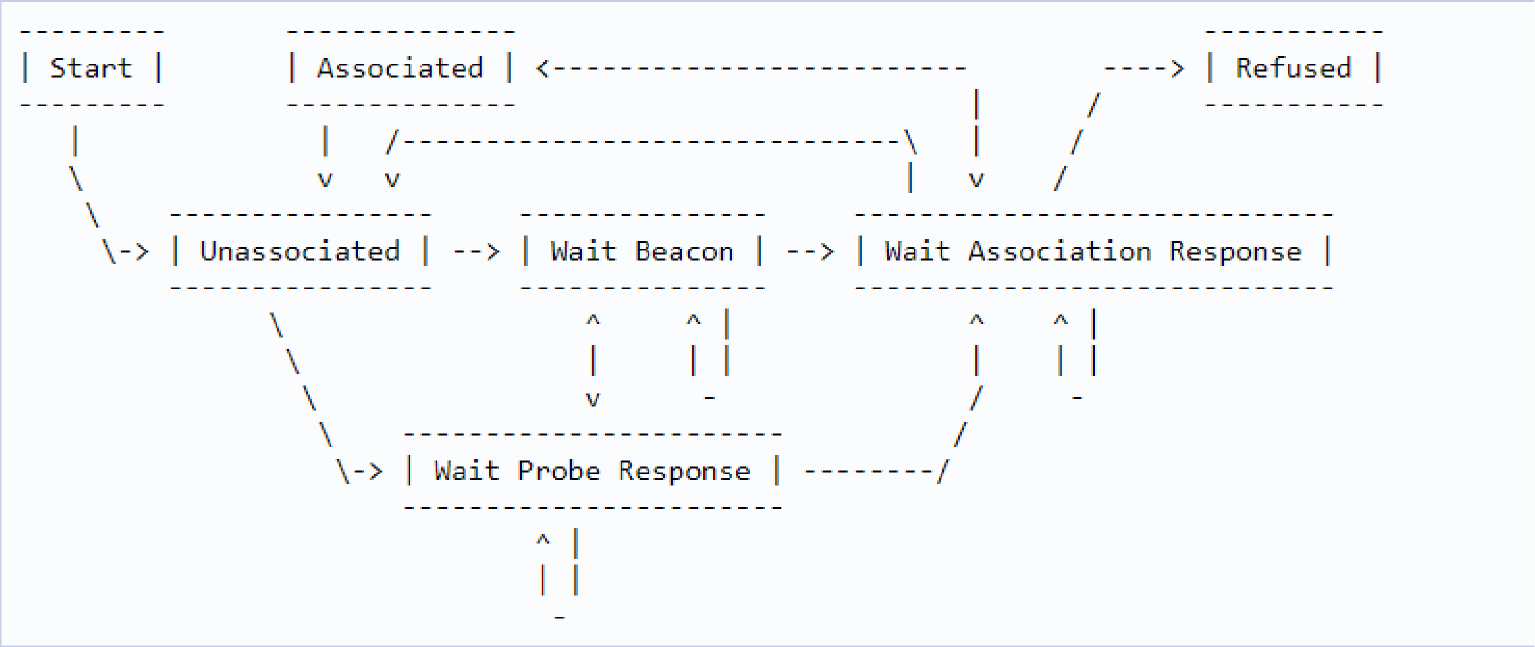
- **Passive scanning**: station passively waits for beacon message
- **Active Scanning**: station actively sends requests to APs



Reference: https://community.nxp.com/t5/Wireless-Connectivity-Knowledge/802-11-Wi-Fi-Connection-Disconnection-process/ta-p/1121148

State Diagram

- Header of beacon messages (passive scanning) and probe responses (active scanning) includes signal power
  - See sta-wifi-mac.cc for more details



```
---------       --------------                                         -----------
| Start |       | Associated | <-------------------------        ----> | Refused |
---------       --------------                           |      /      -----------
   |              |   /------------------------------\   |     /
   \              v   v                              |   v    /
    \    ----------------     ---------------     -----------------------------
     \-> | Unassociated | --> | Wait Beacon | --> | Wait Association Response |
         ----------------     ---------------     -----------------------------
               \                  ^     ^ |              ^    ^ |
                \                 |     | |              |    | |
                 \                v      -               /     -
                  \    -----------------------          /
                   \-> | Wait Probe Response | --------/
                       -----------------------
                               ^ |
                               | |
                                -
```

PCAP File

- Packet capture
  - Collect packet data and store in `.pcap` files
  - Parse file content to analyze network performance

- Tools that can import PCAP files
  - Wireshark
  - Tcpdump
  - Python module: **dpkt**, scapy, pyshark…

Tasks

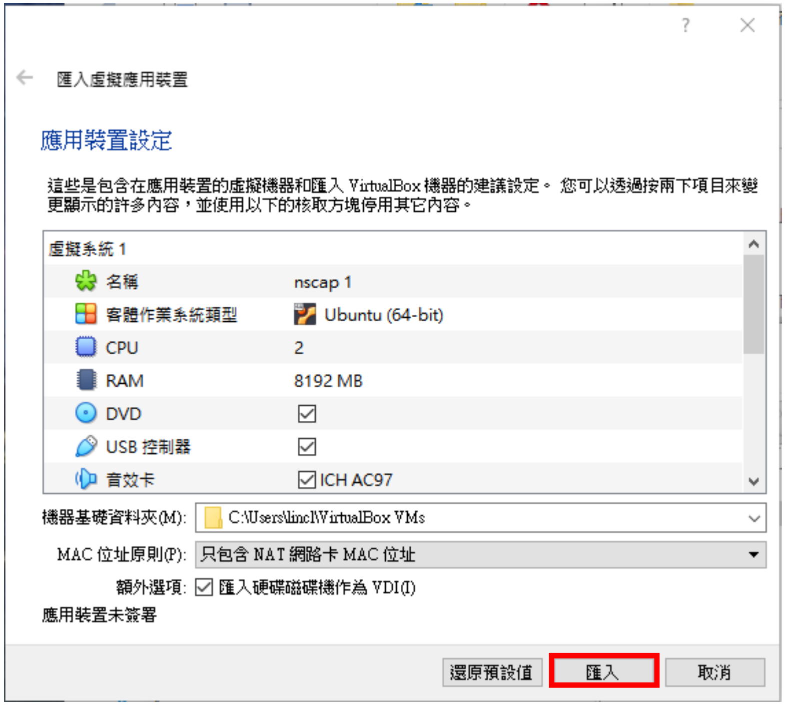
Tasks

0. Environment Setup
1. Handoff Simulation
    Modify sta-wifi-mac.cc / simulation.cc
2. Random Walk Model
    Modify HandoffSimulation.cc
3. Parse pcap file
    Python dpkt module
4. Conclude your simulation

Env Setup

Link of VM OVA file:

- https://drive.google.com/drive/folders/1PVOB3w8EKSfs1EB1SXC_-N9s6hzDhuII?usp=sharing



應用裝置設定

這些是包含在應用裝置的虛擬機器和匯入 VirtualBox 機器的建議設定。 您可以透過按兩下項目來變更顯示的許多內容，並使用以下的核取方塊停用其它內容。

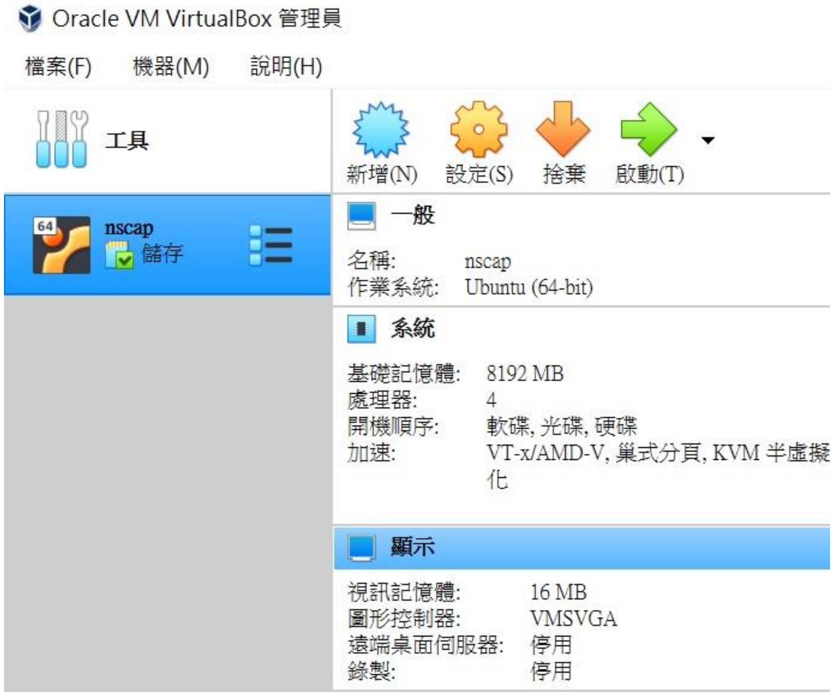
虛擬系統 1	
名稱	nscap 1
客體作業系統類型	Ubuntu (64-bit)
CPU	2
RAM	8192 MB
DVD	☑
USB 控制器	☑
音效卡	☑ ICH AC97

機器基礎資料夾(M): C:\Users\lincl\VirtualBox VMs

MAC 位址原則(P): 只包含 NAT 網路卡 MAC 位址

額外選項: ☑ 匯入硬碟磁碟機作為 VDI(I)

應用裝置未簽署



一般

名稱: nscap
作業系統: Ubuntu (64-bit)

系統

基礎記憶體: 8192 MB
處理器: 4
開機順序: 軟碟, 光碟, 硬碟
加速: VT-x/AMD-V, 巢式分頁, KVM 半虛擬化

顯示

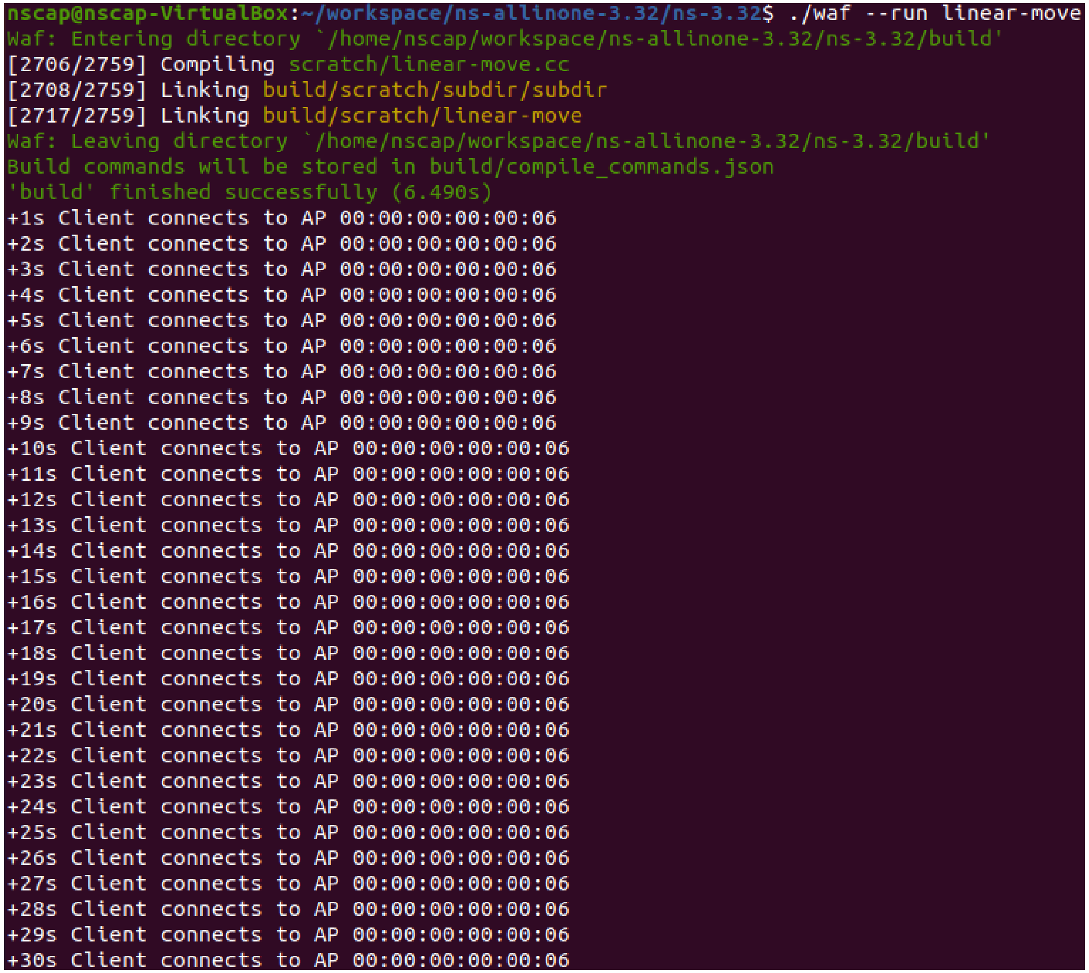
視訊記憶體: 16 MB
圖形控制器: VMSVGA
遠端桌面伺服器: 停用
錄製: 停用

Password is same as user name

Env Testing

```
$ cp task1/linear-move.cc ns-allinone-3.32/ns-3.32/scratch
```

- Expected result



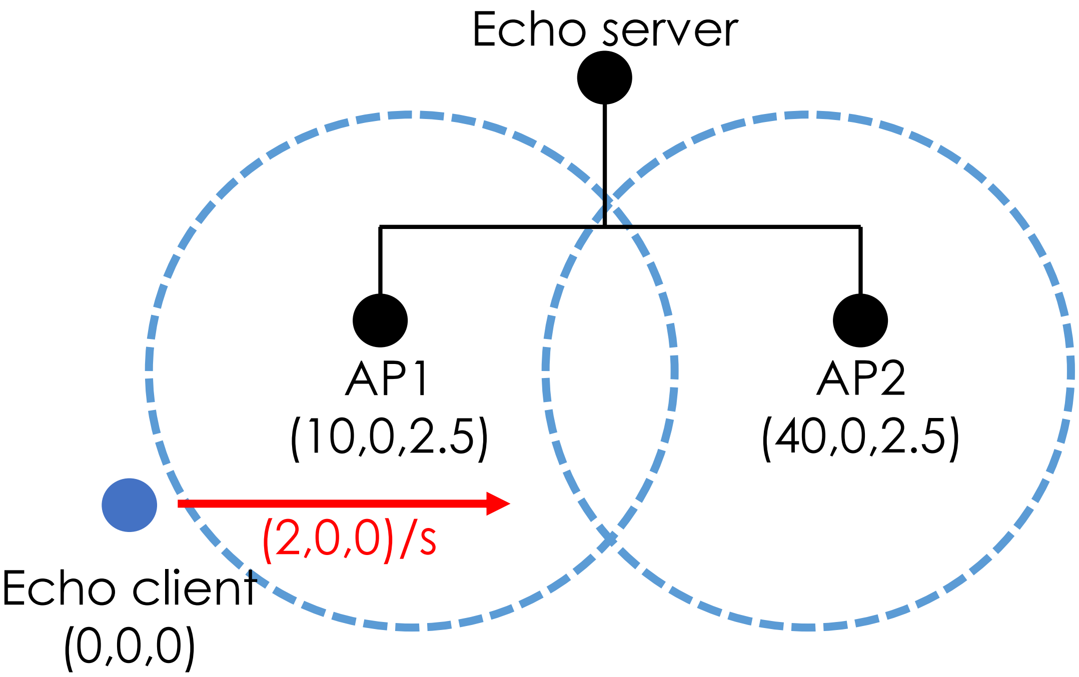
```
nscap@nscap-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run linear-move
Waf: Entering directory `/home/nscap/workspace/ns-allinone-3.32/ns-3.32/build'
[2706/2759] Compiling scratch/linear-move.cc
[2708/2759] Linking build/scratch/subdir/subdir
[2717/2759] Linking build/scratch/linear-move
Waf: Leaving directory `/home/nscap/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (6.490s)
+1s Client connects to AP 00:00:00:00:00:06
+2s Client connects to AP 00:00:00:00:00:06
+3s Client connects to AP 00:00:00:00:00:06
+4s Client connects to AP 00:00:00:00:00:06
+5s Client connects to AP 00:00:00:00:00:06
+6s Client connects to AP 00:00:00:00:00:06
+7s Client connects to AP 00:00:00:00:00:06
+8s Client connects to AP 00:00:00:00:00:06
+9s Client connects to AP 00:00:00:00:00:06
+10s Client connects to AP 00:00:00:00:00:06
+11s Client connects to AP 00:00:00:00:00:06
+12s Client connects to AP 00:00:00:00:00:06
+13s Client connects to AP 00:00:00:00:00:06
+14s Client connects to AP 00:00:00:00:00:06
+15s Client connects to AP 00:00:00:00:00:06
+16s Client connects to AP 00:00:00:00:00:06
+17s Client connects to AP 00:00:00:00:00:06
+18s Client connects to AP 00:00:00:00:00:06
+19s Client connects to AP 00:00:00:00:00:06
+20s Client connects to AP 00:00:00:00:00:06
+21s Client connects to AP 00:00:00:00:00:06
+22s Client connects to AP 00:00:00:00:00:06
+23s Client connects to AP 00:00:00:00:00:06
+24s Client connects to AP 00:00:00:00:00:06
+25s Client connects to AP 00:00:00:00:00:06
+26s Client connects to AP 00:00:00:00:00:06
+27s Client connects to AP 00:00:00:00:00:06
+28s Client connects to AP 00:00:00:00:00:06
+29s Client connects to AP 00:00:00:00:00:06
+30s Client connects to AP 00:00:00:00:00:06
```

Tasks

0. Environment Setup
1. Handoff
   - Modify `sta-wifi-mac.cc` / `linear-move.cc`
2. Random Walk
   - Modify random-walk.cc
3. Parse pcap file
   - Python dpkt module
4. Discuss the observations from your results

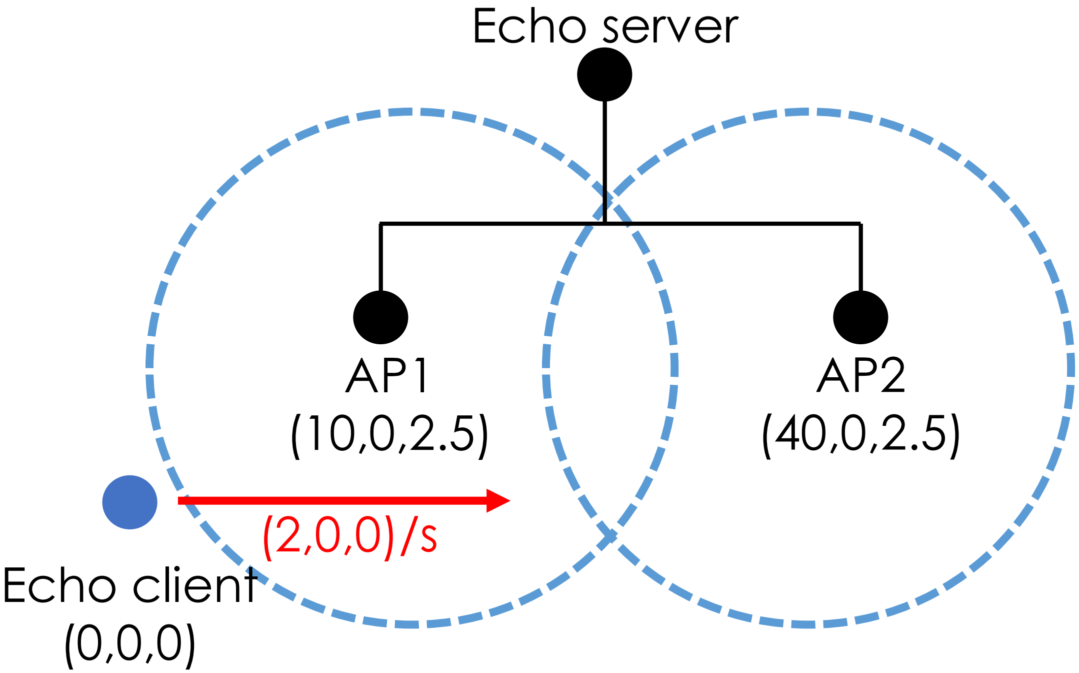
Scenario

- Simulation scenario
  - Two APs with fixed location
  - Echo client continuously sends packets to the echo server upon connecting to an AP



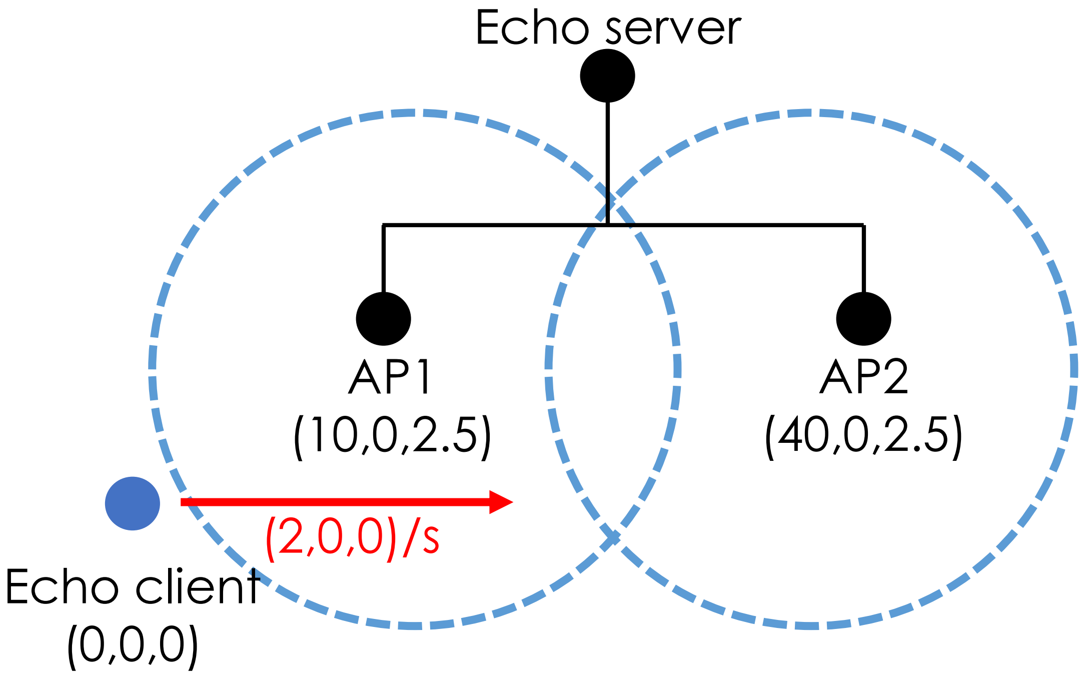
Movable Client

- Step 1: Add a function called "movement" in echo client to enable linear movement
  - A client moves with a constant speed and direction



Mobility Example

- Step 1: Add a function called "movement" in echo client to enable linear movement
  - Modify [TODO] part in `part1/linear-move.cc`

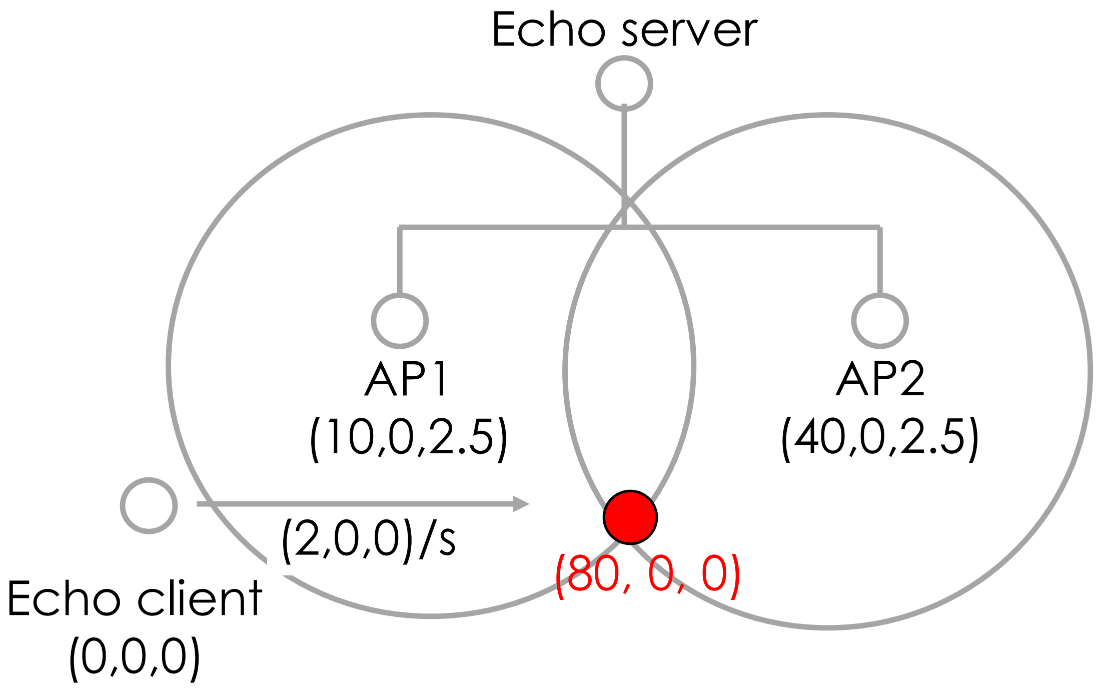


Reference

- Task 0:
  - https://www.nsnam.org/docs/tutorial/html/getting-started.html (Env setup)
  - https://www.nsnam.org/docs/tutorial/html/conceptual-overview.html#building-your-script (Build your program)
- Task1:
  - https://www.nsnam.org/docs/doxygen/wifi-rate-adaptation-distance_8cc.html (Update position)
  - https://mrncciew.com/2014/10/11/802-11-mgmt-deauth-disassociation-frames/ (Disassociation frame)
- Task3:
  - https://dpkt.readthedocs.io/en/latest/api/index.html
  (dpkt api reference)
  - http://www.radiotap.org/fields/defined
  (radiotap packet header)

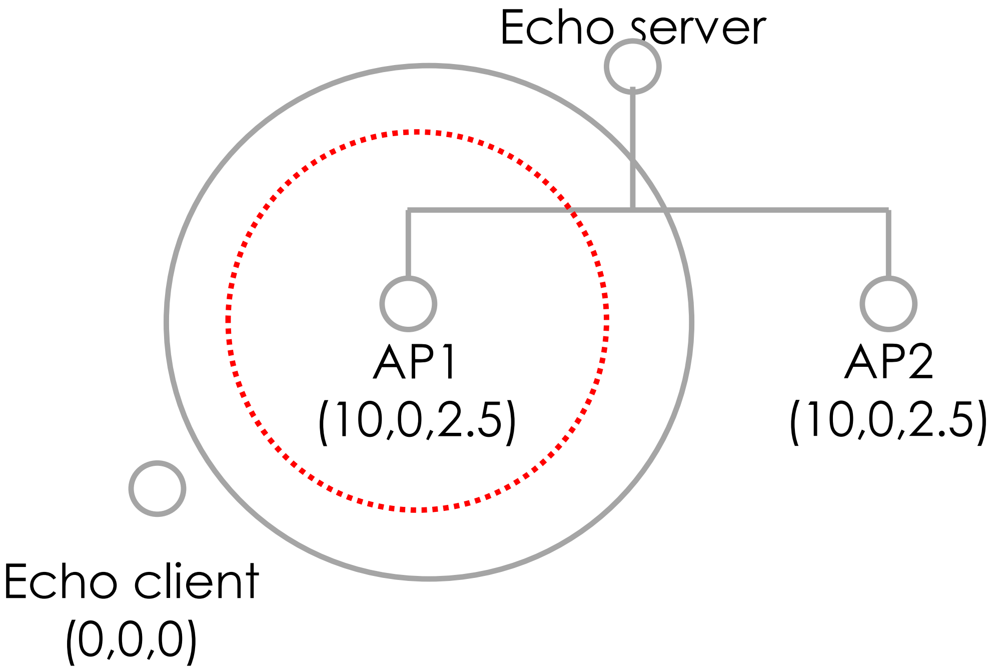
Simple Simulation

- Step 2: Run the simulation program
  - Client connects to AP1 in the beginning
  - Connection corrupts at (80, 0, 0)
  - Client establishes a new connection with AP2



Threshold-based Handoff

- Step 3: Implement threshold-based handoff algorithm
  - If (quality < threshold):
    - connection corrupts
  - Quality measurement: SNR



Detail of Implementation

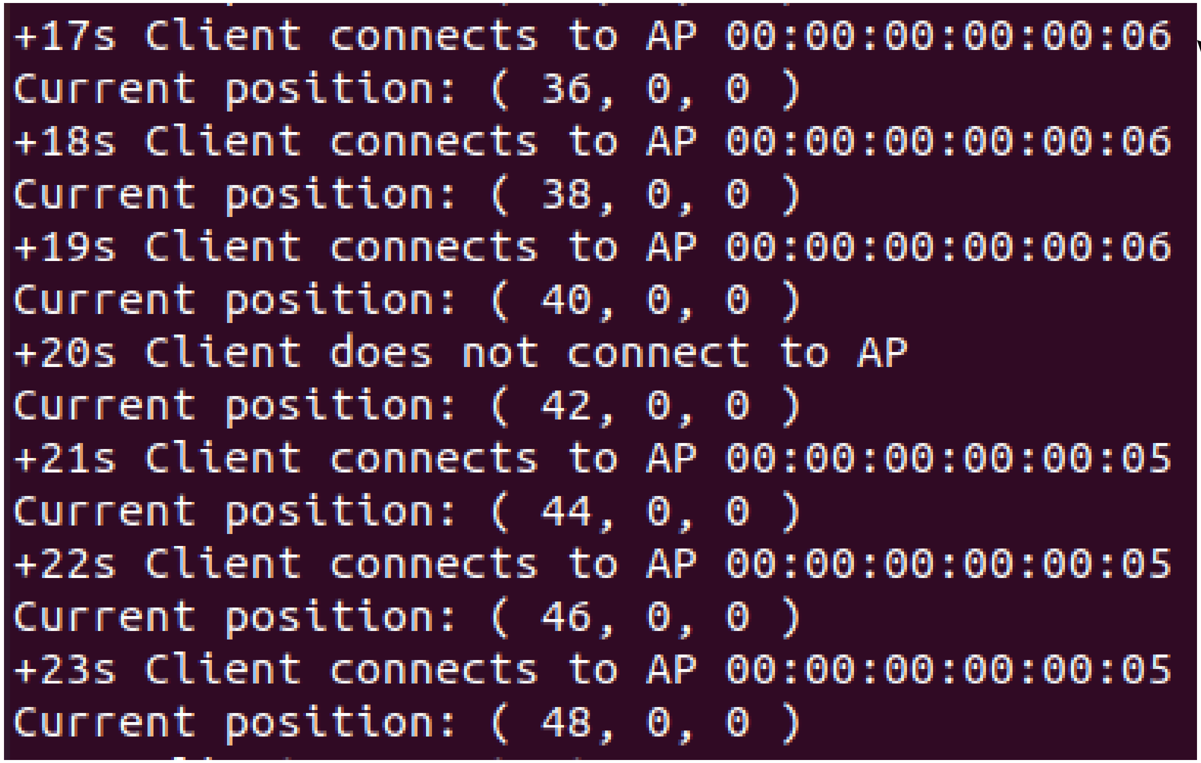
- Step 3: Implement threshold-based handoff algorithm
  - Modify [TODO] part in `sta-wifi-mac.cc`
  1. Check SNR info. in the beacon packet received from each candidate AP
  2. Print the association message to stdout

  - Hint: packet -> RemovePacketTag(SnrTag)
  - Note: please replace original files with yours

```
$ cp task1/sta-wifi-mac.cc ns-allinone-3.32/ns-3.32/src/wifi/model/
$ cp task1/sta-wifi-mac.h ns-allinone-3.32/ns-3.32/src/wifi/model/
```

Example Output

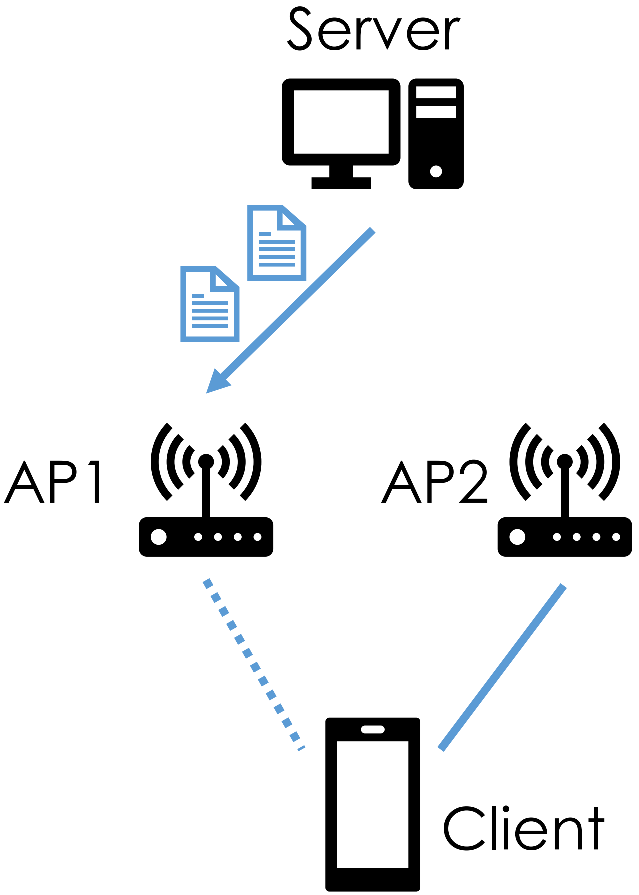
- Expected result



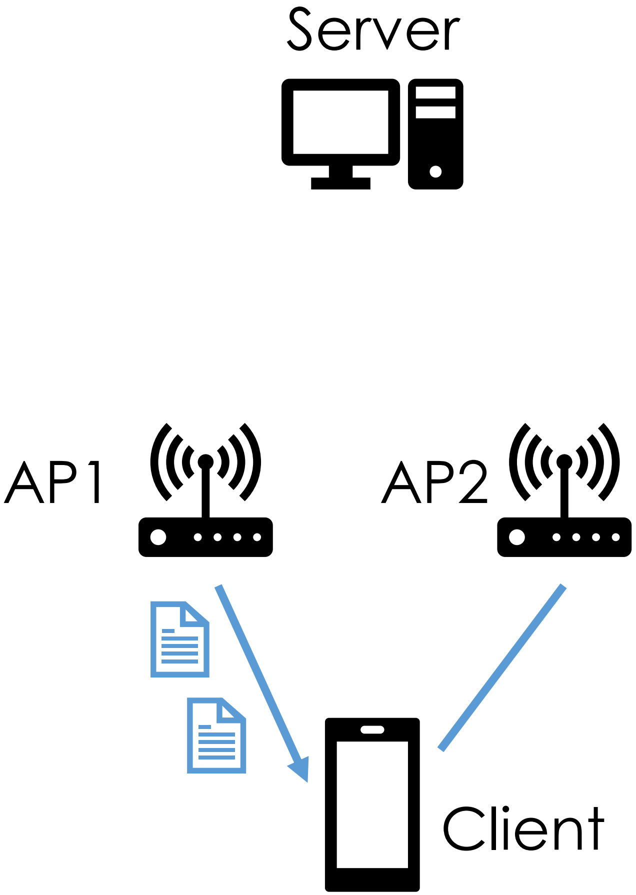
```
+17s Client connects to AP 00:00:00:00:00:06
Current position: ( 36, 0, 0 )
+18s Client connects to AP 00:00:00:00:00:06
Current position: ( 38, 0, 0 )
+19s Client connects to AP 00:00:00:00:00:06
Current position: ( 40, 0, 0 )
+20s Client does not connect to AP
Current position: ( 42, 0, 0 )
+21s Client connects to AP 00:00:00:00:00:05
Current position: ( 44, 0, 0 )
+22s Client connects to AP 00:00:00:00:00:05
Current position: ( 46, 0, 0 )
+23s Client connects to AP 00:00:00:00:00:05
Current position: ( 48, 0, 0 )
```

Association message

Disassociation Issue



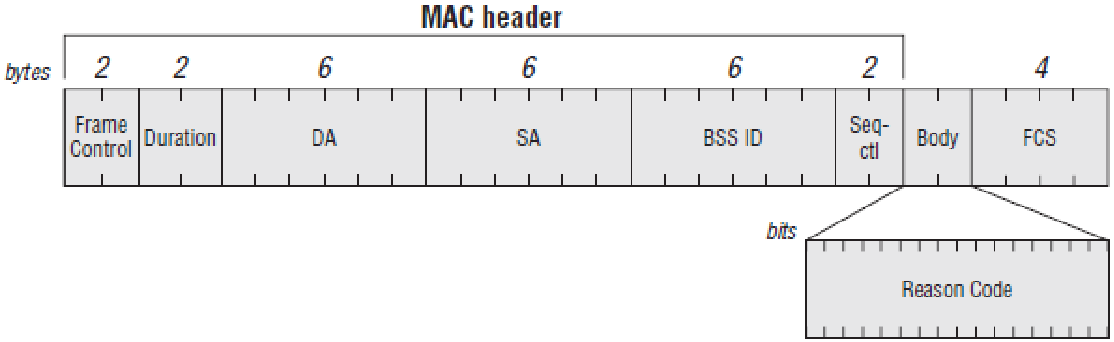
Client does handoff without notifying AP1



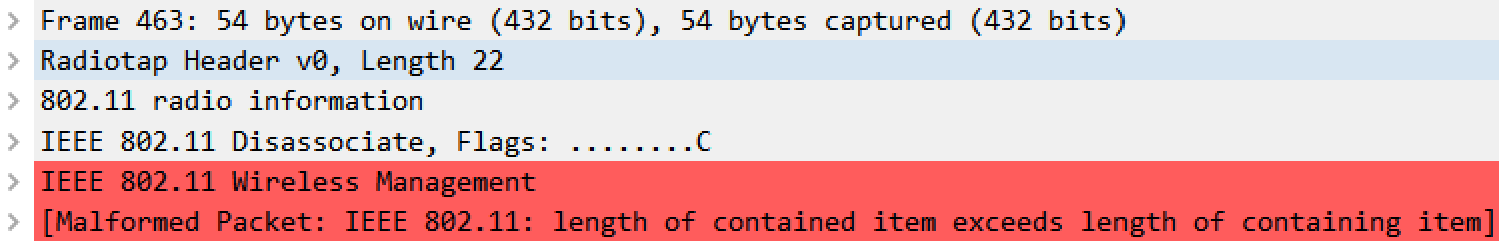
AP1 sends packets to its "served" client → Incorrect!

Disassociation Message

- [TODO]: Send a disassociation packet to AP upon handoff
  - Disassociation frame format



  - Create a well-formed packet and transmit to the AP



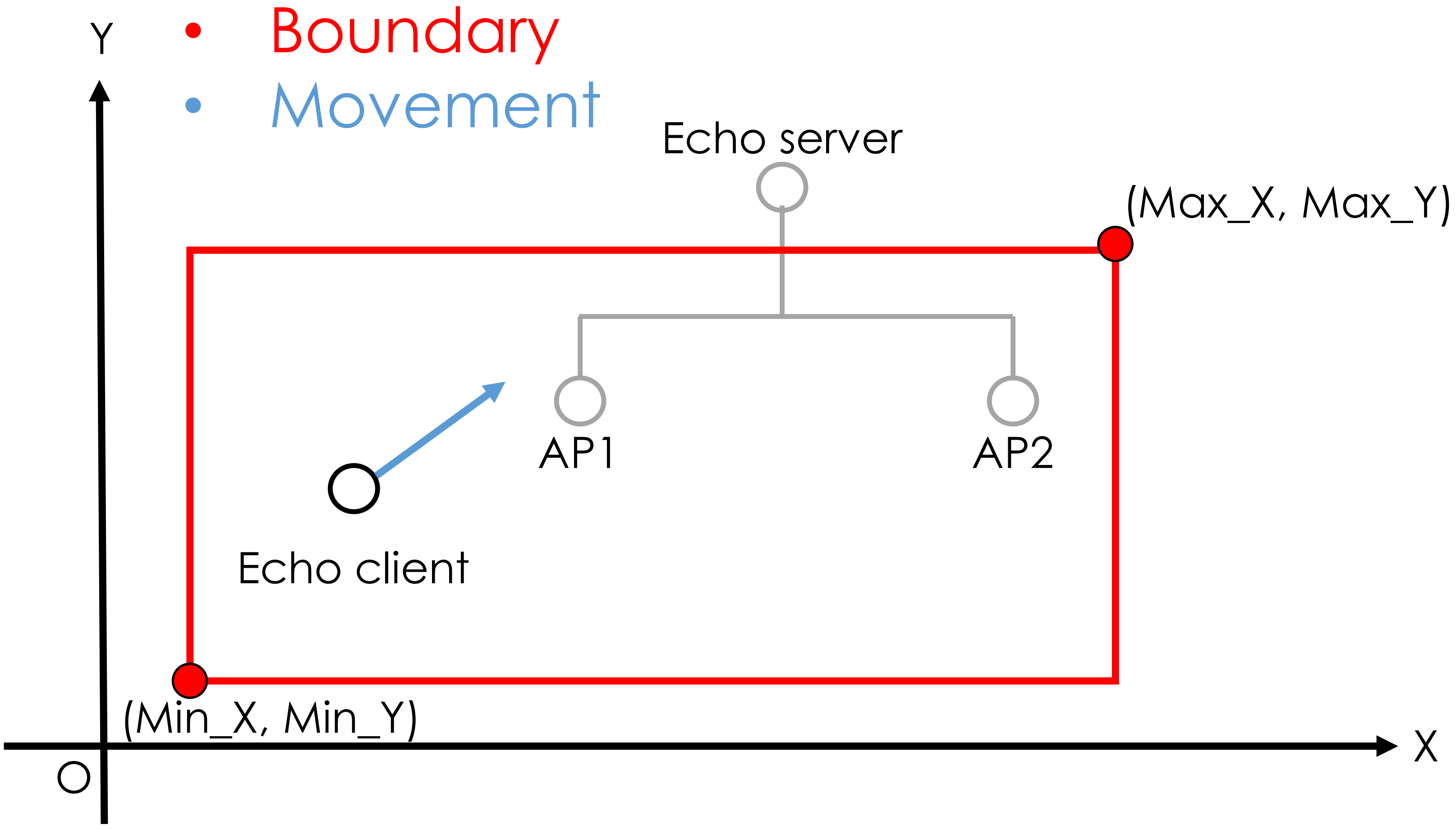
```
> Frame 463: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)
> Radiotap Header v0, Length 22
> 802.11 radio information
> IEEE 802.11 Disassociate, Flags: ........C
> IEEE 802.11 Wireless Management
> [Malformed Packet: IEEE 802.11: length of contained item exceeds length of containing item]
```

Malformed packet example

Tasks

0. Environment Setup
1. Handoff Simulation
    - Modify sta-wifi-mac.cc / linear-move.cc
2. Random Walk
    - Modify `random-walk.cc`
3. Parse pcap file
    - Python dpkt module
4. Discuss the observations from your results

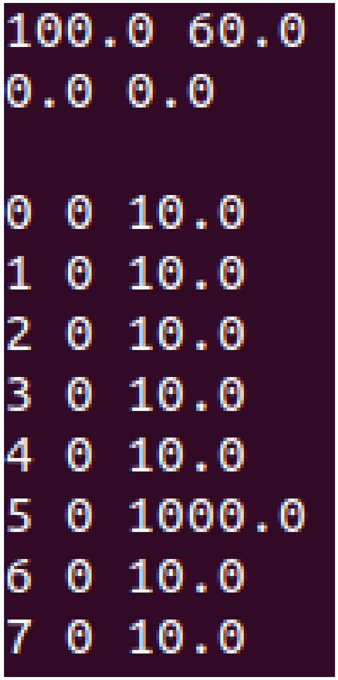
Visualize Simulation



Load Movement Data

- Step 1: Read user trajectory from “`testcase_n`”
  - Boundary: first 2 lines
    - 1st: Max_X, Max_Y ; 2nd: Min_X, Min_Y
  - Movement: index, direction(rad), speed(m/s)

Range
0 ≤ boundary ≤ 500
0 ≤ index ≤ 100
0 ≤ direction ≤ 2π
0 ≤ speed ≤ 1000



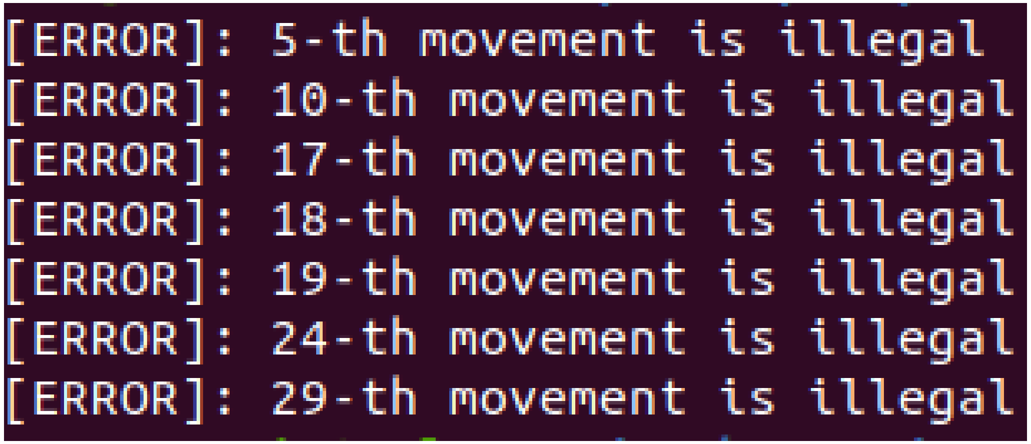
```
100.0 60.0
0.0 0.0

0 0 10.0
1 0 10.0
2 0 10.0
3 0 10.0
4 0 10.0
5 0 1000.0
6 0 10.0
7 0 10.0
```

Boundary

Movement

Boundary Check

- Step 2: Check whether the client would reach the boundary
  1. Print error message
     - [ERROR]: n-th movement is illegal
     - Write to stderr
  2. Skip this movement

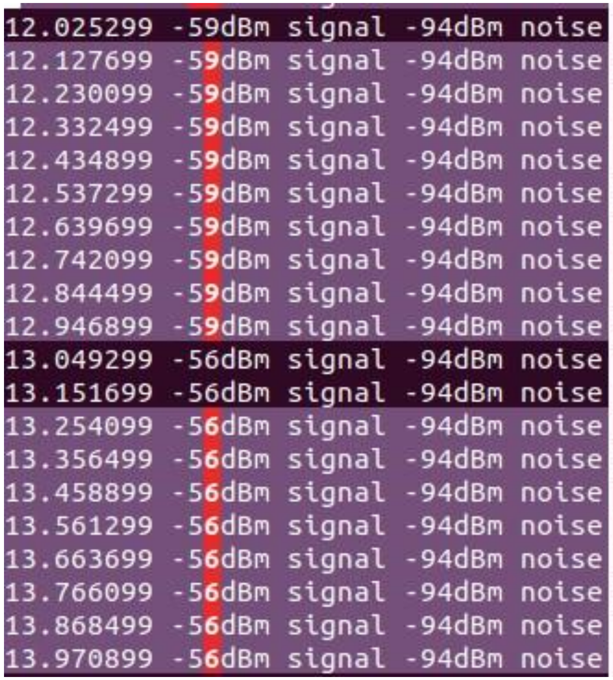


```
[ERROR]: 5-th movement is illegal
[ERROR]: 10-th movement is illegal
[ERROR]: 17-th movement is illegal
[ERROR]: 18-th movement is illegal
[ERROR]: 19-th movement is illegal
[ERROR]: 24-th movement is illegal
[ERROR]: 29-th movement is illegal
```

Example of illegal message

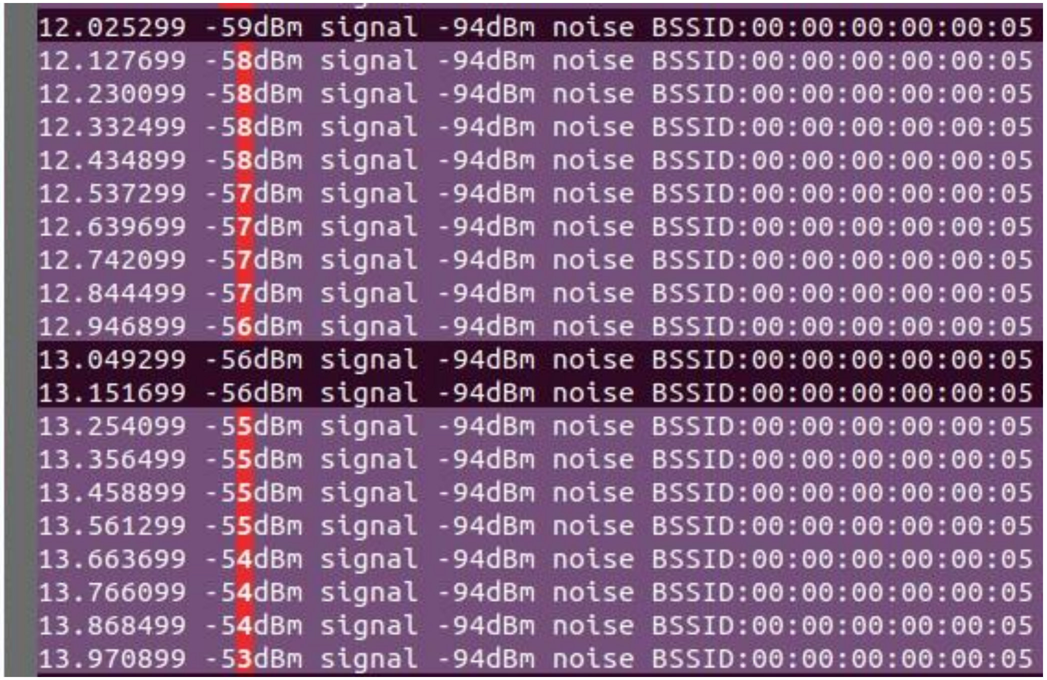
Random Walk

- Echo client walks continuously toward a specific direction
- SNR changes more smoothly than discrete movement



```
12.025299 -59dBm signal -94dBm noise
12.127699 -59dBm signal -94dBm noise
12.230099 -59dBm signal -94dBm noise
12.332499 -59dBm signal -94dBm noise
12.434899 -59dBm signal -94dBm noise
12.537299 -59dBm signal -94dBm noise
12.639699 -59dBm signal -94dBm noise
12.742099 -59dBm signal -94dBm noise
12.844499 -59dBm signal -94dBm noise
12.946899 -59dBm signal -94dBm noise
13.049299 -56dBm signal -94dBm noise
13.151699 -56dBm signal -94dBm noise
13.254099 -56dBm signal -94dBm noise
13.356499 -56dBm signal -94dBm noise
13.458899 -56dBm signal -94dBm noise
13.561299 -56dBm signal -94dBm noise
13.663699 -56dBm signal -94dBm noise
13.766099 -56dBm signal -94dBm noise
13.868499 -56dBm signal -94dBm noise
13.970899 -56dBm signal -94dBm noise
```

Discrete



```
12.025299 -59dBm signal -94dBm noise BSSID:00:00:00:00:00:05
12.127699 -58dBm signal -94dBm noise BSSID:00:00:00:00:00:05
12.230099 -58dBm signal -94dBm noise BSSID:00:00:00:00:00:05
12.332499 -58dBm signal -94dBm noise BSSID:00:00:00:00:00:05
12.434899 -58dBm signal -94dBm noise BSSID:00:00:00:00:00:05
12.537299 -57dBm signal -94dBm noise BSSID:00:00:00:00:00:05
12.639699 -57dBm signal -94dBm noise BSSID:00:00:00:00:00:05
12.742099 -57dBm signal -94dBm noise BSSID:00:00:00:00:00:05
12.844499 -57dBm signal -94dBm noise BSSID:00:00:00:00:00:05
12.946899 -56dBm signal -94dBm noise BSSID:00:00:00:00:00:05
13.049299 -56dBm signal -94dBm noise BSSID:00:00:00:00:00:05
13.151699 -56dBm signal -94dBm noise BSSID:00:00:00:00:00:05
13.254099 -55dBm signal -94dBm noise BSSID:00:00:00:00:00:05
13.356499 -55dBm signal -94dBm noise BSSID:00:00:00:00:00:05
13.458899 -55dBm signal -94dBm noise BSSID:00:00:00:00:00:05
13.561299 -55dBm signal -94dBm noise BSSID:00:00:00:00:00:05
13.663699 -54dBm signal -94dBm noise BSSID:00:00:00:00:00:05
13.766099 -54dBm signal -94dBm noise BSSID:00:00:00:00:00:05
13.868499 -54dBm signal -94dBm noise BSSID:00:00:00:00:00:05
13.970899 -53dBm signal -94dBm noise BSSID:00:00:00:00:00:05
```

Continuous

Tasks

0. Environment Setup

1. Handoff Simulation

   Modify sta-wifi-mac.cc / linear-move.cc

2. Random Walk Model

   Modify random-walk.cc

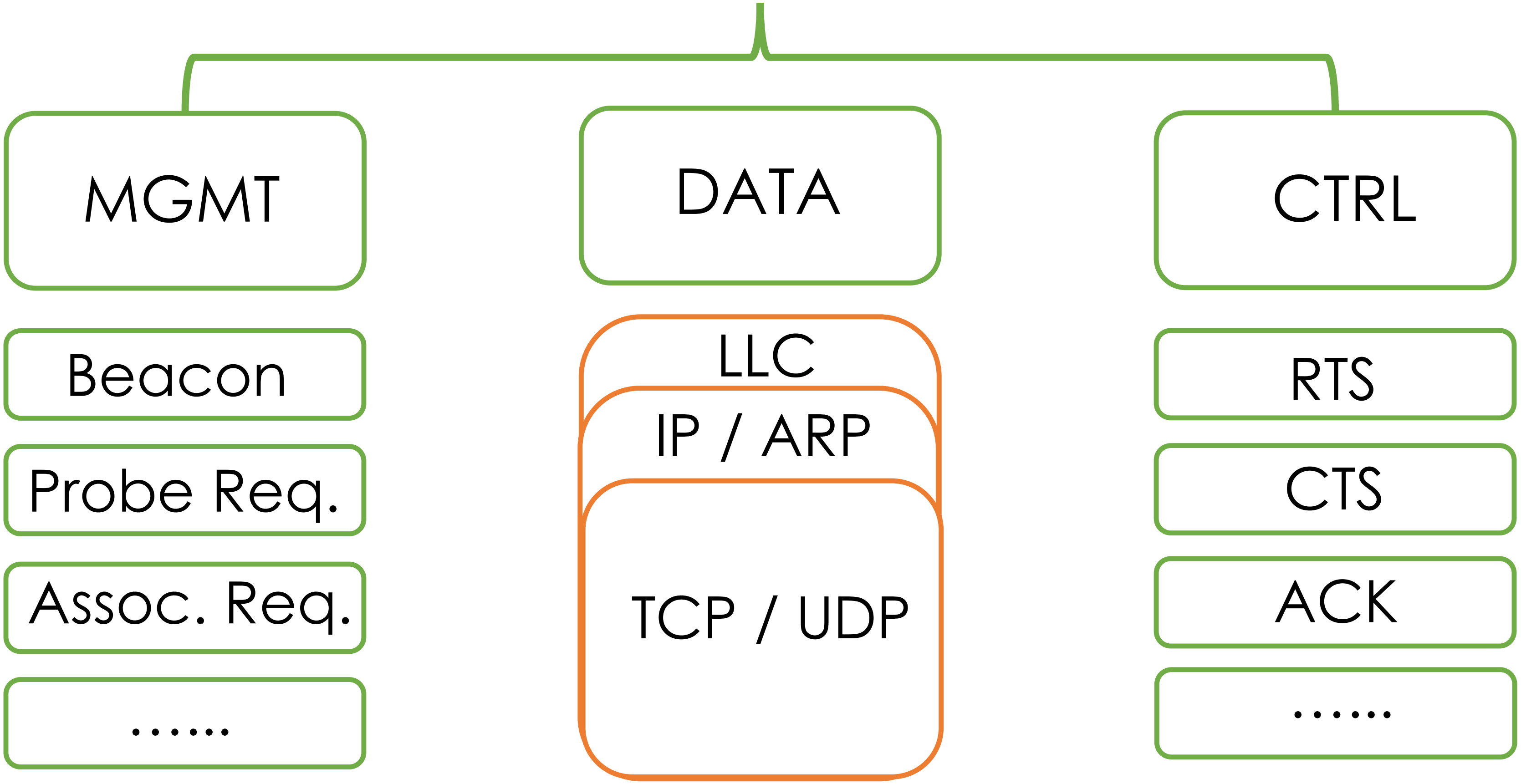
3. Parse pcap file

   Python dpkt module

4. Discuss the observations from your results

dpkt

- dpkt – a packet parsing module
- Write a parser to analyze simulation output
- 802.11: 3 Frame types, many subtypes



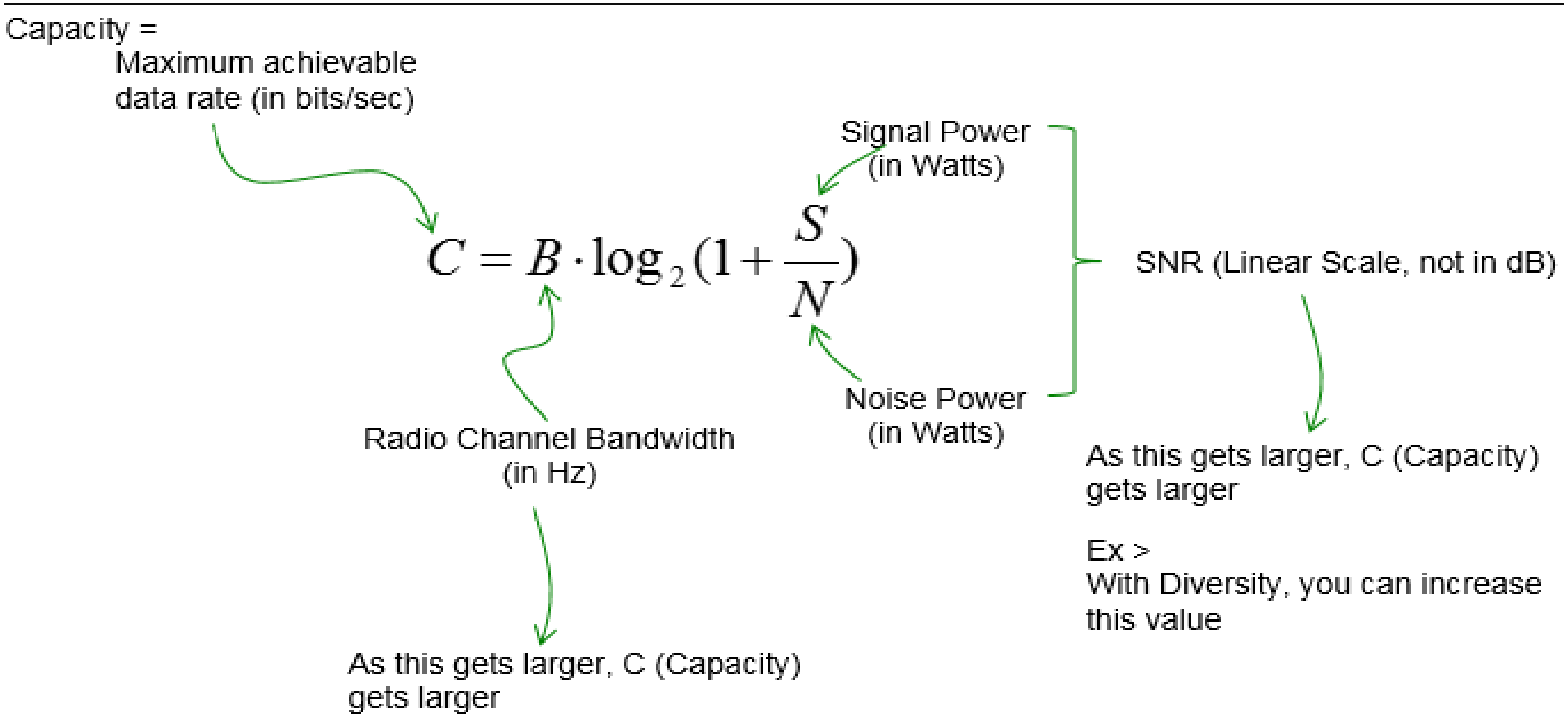
Distillation from pcap

- [TODO]

1. MGMT:
   a) Record AP's mac addr
   b) Calculate connection duration / number of handoff events
   c) SNR in beacon packet
2. DATA:
   a) Calculate total transmitted bytes
   b) Record packets' SNR

Note: As for SNR information, you only need to count downlink packets (but for all APs)

Theoretical Sum-Rate



Capacity =
Maximum achievable
data rate (in bits/sec)

$$C = B \cdot \log_2(1 + \frac{S}{N})$$

Signal Power
(in Watts)

Noise Power
(in Watts)

SNR (Linear Scale, not in dB)

Radio Channel Bandwidth
(in Hz)

As this gets larger, C (Capacity)
gets larger

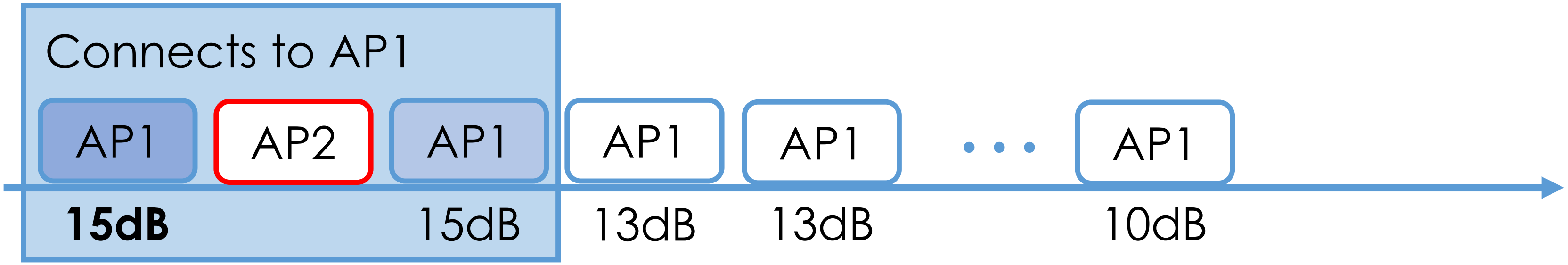
As this gets larger, C (Capacity)
gets larger

Ex >
With Diversity, you can increase
this value

Sum-rate
= Σ C(effective channel duration)/simulation duration
Capacity (Effective channel)
= beacon interval * SNR of beacon from associated AP

Sum-Rate Example

Beacon Interval (BI): 0.3 sec
Simulation duration: 60 sec



Sum-rate =

$(0.3/60) * (\text{Capacity}(\boxed{\text{AP1}}) + \text{Capacity}(\boxed{\text{AP1}}))$

$\text{Capacity} = \text{Bandwidth} * \log_2(1+10^{\mathbf{15}/10})$

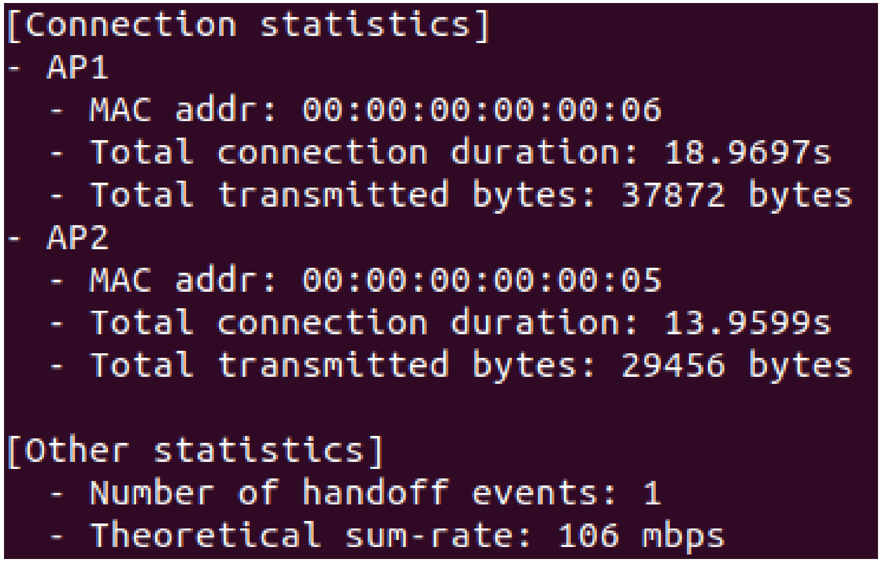
Hint: ant_sig.db / ant_noise.db in dpkt.radiotap

Output Format

- Usage:

```
$ python3 parser.py <file_name>.pcap
```

- Format:



```
[Connection statistics]
- AP1
  - MAC addr: 00:00:00:00:00:06
  - Total connection duration: 18.9697s
  - Total transmitted bytes: 37872 bytes
- AP2
  - MAC addr: 00:00:00:00:00:05
  - Total connection duration: 13.9599s
  - Total transmitted bytes: 29456 bytes

[Other statistics]
  - Number of handoff events: 1
  - Theoretical sum-rate: 106 mbps
```

- You can output results to a new file if you need

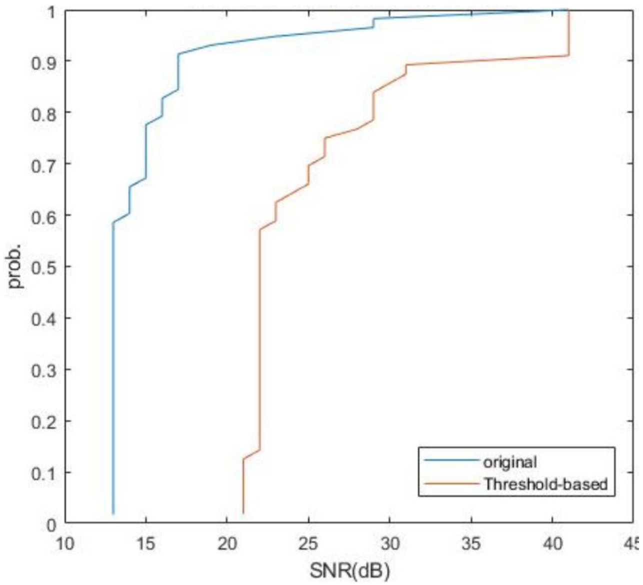
- You CANNOT import any other module

Tasks

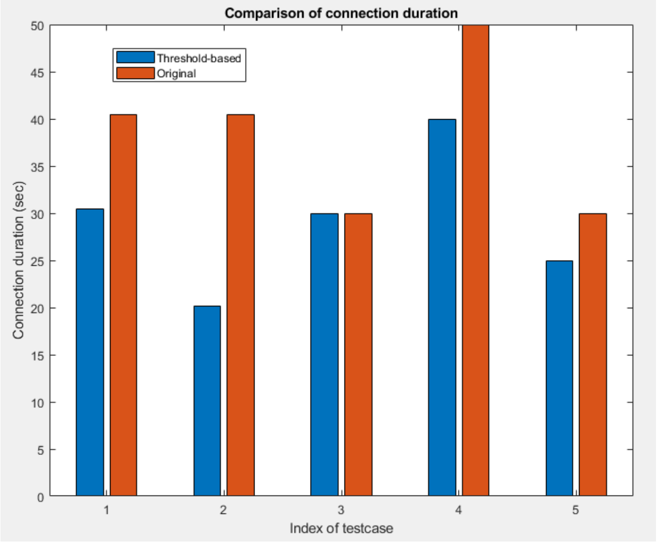
0. Environment Setup
1. Handoff Simulation
    - Modify sta-wifi-mac.cc / linear-move.cc
2. Random Walk Model
    - Modify random-walk.cc
3. Parse pcap file
    - Python dpkt module
4. Discuss the observations from your results

Visualize Result

- CDF of packet's SNR
  - 5 testcases, 5 figures



- Bar chart: comparison of
  - Connection duration
  - Number of handoff events
  - Theoretical sum-rate



Questions

- Write a short discuss to answer the following questions
  1. Compare the advantages and disadvantages of the two handoff algorithms
  2. In task2, how do you make a client walk continuously?
     (Hint: beacon interval is the smallest period of observation in the simulation)
  3. In task3, how do you know the occurrence of handoff event? And how do you get connection duration?
     (Which types of management packets did you use?)

Questions

- Please put all of your figures and answers in your report

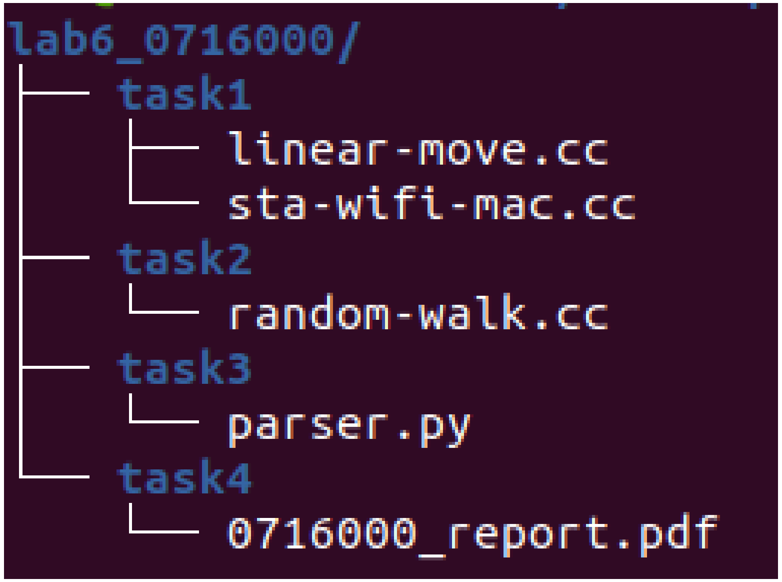
  - File name: `<student_ID>_report.pdf`

Grading Policy

- Grade
  - Task 1: linear mobility (10%) / handoff (25%)
  - Task 2: Random walk (20%)
  - Task 3: Trace analysis (20%)
  - Task 4: Report
    - Figure (10%) (pre-requisite: task1~3)
    - Question (15%)
- Cheat policy
  - Cheaters equally share the score

File Structure

- Folder name: **lab6_<student ID>**
- Folder structure:



```
lab6_0716000/
├── task1
│   ├── linear-move.cc
│   └── sta-wifi-mac.cc
├── task2
│   └── random-walk.cc
├── task3
│   └── parser.py
└── task4
    └── 0716000_report.pdf
```

- Compress your folder: **lab6_<studentID>.zip**
- **No score if filenames or the folder structure is wrong**

Submission

- Deadline: Jun. 22(Tue.) 23:59
- Submit to new E3
- Late penalty
  - 20% off within 1 week of the deadline
  - You can't submit after 06/29 23:59

Note

- We will grade by running a test script. Please follow the output format.

- Please do not modify any setting without the [TODO] mark in the example code

Reference

- Task 0:
  - https://www.nsnam.org/docs/tutorial/html/getting-started.html (Env setup)
  - https://www.nsnam.org/docs/tutorial/html/conceptual-overview.html#building-your-script (Build your program)
- Task1:
  - https://www.nsnam.org/docs/doxygen/wifi-rate-adaptation-distance_8cc.html (Update position)
  - https://mrncciew.com/2014/10/11/802-11-mgmt-deauth-disassociation-frames/ (Disassociation frame)
- Task3:
  - https://dpkt.readthedocs.io/en/latest/api/index.html

    (dpkt api reference)
  - http://www.radiotap.org/fields/defined

    (radiotap packet header)

Q&A