IETF-116
IPMON Hackathon Project

March 25~26, 2023

Champions: Jaehoon (Paul) Jeong and Yiwen (Chris) Shen

Presenter: Hyeonah Jung

Members: Bien Aime Mugabarigira and Junhee Kwon

Department of Computer Science and Engineering at SKKU

Email: {pauljeong, bienaime, hyeonah214, juun9714}@skku.edu, chrisshen@ks.ac.kr

I E T F

IP Wireless Access in Vehicular Environments (IPWAVE) Basic Protocols Project

Champion: Jaehoon (Paul) Jeong (SKKU)

IETF-116 IPWAVE Hackathon Project

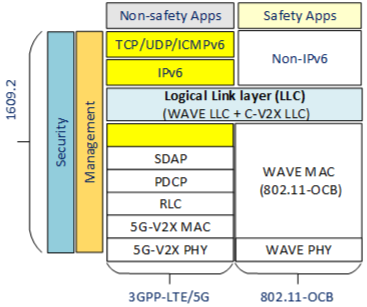
Professors:

- Jaehoon (Paul) Jeong (SKKU)
- Younghan Kim (SSU)
- Yiwen (Chris) Shen (KSU)

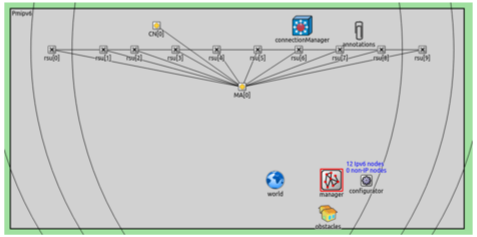
Students:

- Bien Aime Mugabarigira (SKKU)
- Hyeonah Jung (SKKU)
- Junhee Kwon (SKKU)

WAVE Protocol Stack



IPv6 ND Option



Objectives

To Demonstrate IPWAVE Basic Protocols:

- The handover based on LSTM knows the handover timing in advance and proceeds with the handover.
- Implementation of handover with the IPv6 version of 802.11-OCB.
- One-Hop V2I Vehicular ND based on IPv6 over 5G V2X for drones.

Where to get source code:

- GitHub: https://github.com/ipwave-hackathon-ietf
- YouTube: https://youtu.be/V0BoXDkIg5c

How to set up an environment:

- OS: Ubuntu 16.04/ Ubuntu 20.04
- SUMO 0.25.0/ SUMO 1.13.0
- OMNeT++ 5.4.1/ OMNeT++ 6.0
- GNU GCC 5.4 /GNU GCC 7.3
- INET 4.0/ INET 4.4

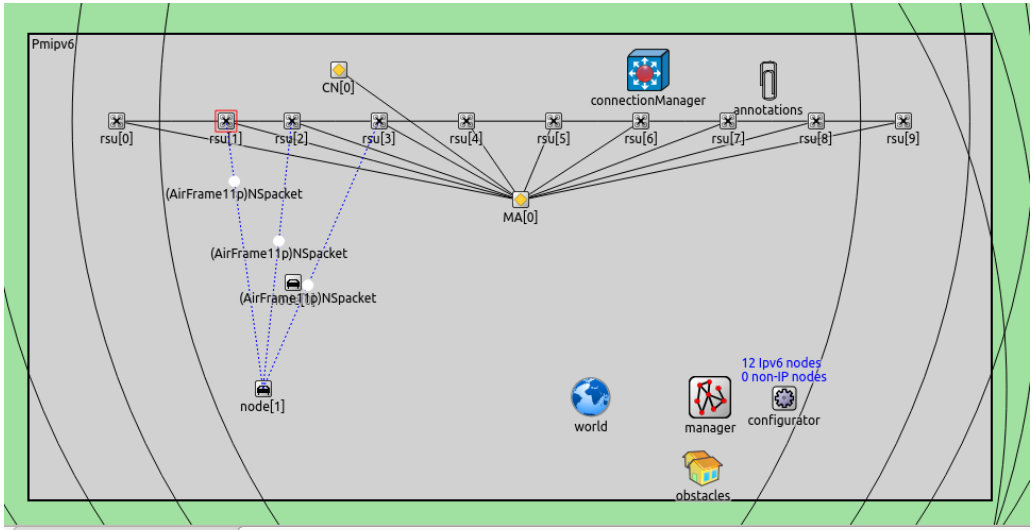
Implementation Contents:

- Handover simulation from IPv4 to IPv6.
- Implementing of IPv6 over 5G V2X for drones.

Hackathon Plan

- Draft for this Project
  - Vehicular Mobility Management for IP-Based Vehicular Networks
    - https://datatracker.ietf.org/doc/draft-jeong-ipwave-vehicular-mobility-management/
- Simulation
  - To make accurate handover timing with a low delay.
    - RSUs are deployed along a highway to give vehicles the Internet connectivity.
    - Vehicles can connect to the RSUs via Vehicular Ad Hoc Networks (VANET) as multi-hop Vehicle-to-Infrastructure (V2I).
    - To train a LSTM (Long Short-Term Memory) model to predict the handover timing, RSUs retrieve a moving vehicle's RSSI data in OMNET simulation and remove the noise from the RSSI data by Kalman filter.
    - Then, we label the filtered data in two categories, handover timing and non-handover timing. After that, we train the LSTM model to predict the handover timing well.

What got done (1/2)




```
INFO (Mac1609_4)Pmipv6.node[0].nic.mac1609_4: Passed slots after DIFS: 472003
INFO (Mac1609_4)Pmipv6.node[0].nic.mac1609_4: Updating backoff for Queue 0: 10 -> 0 PostCommit Over
** Event #854  t=24.677303351992  Pmipv6.rsu[0].nic.phy80211p (PhyLayer80211p, id=113)  on NSpacket (AirFrame11p, id=1363)
INFO (PhyLayer80211p)Pmipv6.rsu[0].nic.phy80211p: rsu[0]::PhyLayer80211p: Received new AirFrame (AirFrame11p)NSpacket from channel.
INFO (PhyLayer80211p)Pmipv6.rsu[0].nic.phy80211p: My current position is: (200,130,1.895)
INFO (PhyLayer80211p)Pmipv6.rsu[0].nic.phy80211p: rsu[0]::PhyLayer80211p: Sender's antenna gain: 0.733274
INFO (PhyLayer80211p)Pmipv6.rsu[0].nic.phy80211p: rsu[0]::PhyLayer80211p: Own (receiver's) antenna gain: 0.678599
INFO (PhyLayer80211p)Pmipv6.rsu[0].nic.phy80211p: PhyLayer(SimplePathlossModel): sqrdistance is: 41693
INFO (PhyLayer80211p)Pmipv6.rsu[0].nic.phy80211p: PhyLayer(SimplePathlossModel): wavelength is: 0.0508985
INFO (PhyLayer80211p)Pmipv6.rsu[0].nic.phy80211p: PhyLayer(SimplePathlossModel): distance factor is: 1.51886e-07
INFO (PhyLayer80211p)Pmipv6.rsu[0].nic.phy80211p: PhyLayer(SimplePathlossModel): Signal contains frequency dimension: yes
INFO (PhyLayer80211p)Pmipv6.rsu[0].nic.phy80211p: PhyLayer(SimpleObstacleShadowing): value is: 1
INFO (PhyLayer80211p)Pmipv6.rsu[0].nic.phy80211p: [Host 0] - PhyLayer(Decider): Processing AirFrame...
INFO (PhyLayer80211p)Pmipv6.rsu[0].nic.phy80211p: AirFrame: 18 with (1.56638e-09 > 1.25893e-09) -> Trying to receive AirFrame.
INFO (PhyLayer80211p)Pmipv6.rsu[0].nic.phy80211p: rsu[0]::PhyLayer80211p: Handed AirFrame with ID 18 to Decider. Next handling in 0.000144s.
** Event #855  t=24.677303351992  Pmipv6.rsu[0].nic.mac1609_4 (Mac1609_4, id=114)  on ChannelStatus (omnetpp::cMessage, id=1374)
INFO (Mac1609_4)Pmipv6.rsu[0].nic.mac1609_4: Channel turned busy: External sender
INFO (Mac1609_4)Pmipv6.rsu[0].nic.mac1609_4: Stopping Contention at 24677303351992
INFO (Mac1609_4)Pmipv6.rsu[0].nic.mac1609_4: Channel was idle for 6.136190363709
INFO (Mac1609_4)Pmipv6.rsu[0].nic.mac1609_4: Passed slots after DIFS: 472003
INFO (Mac1609_4)Pmipv6.rsu[0].nic.mac1609_4: Updating backoff for Queue 0: 7 -> 0 PostCommit Over
```

OMNeT++

What got done (2/2)

- We changed the 802.11-OCB handover simulation from IPv4 to IPv6.
  - IPv4-based handover was done in IETF-113 Hackathon in March in 2022.
  - This time we support IPv6-based handover for vehicular networks.

Open Source

URL: https://github.com/ipwave-hackathon-ietf/ipwave-hackathon-ietf-116

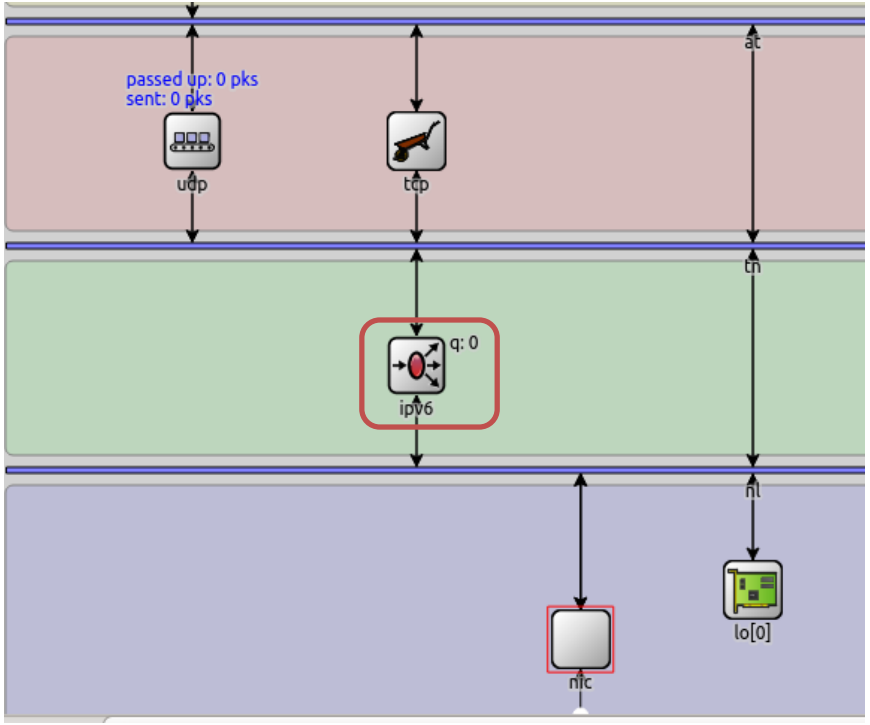
hyeonahjung IPv6 version of 802.11-OCB		3945ffd 1 minute ago	2 commits
inet	IPv6 version of 802.11-OCB		1 minute ago
veinsHW4	IPv6 version of 802.11-OCB		1 minute ago
README.md	Initial commit		8 minutes ago

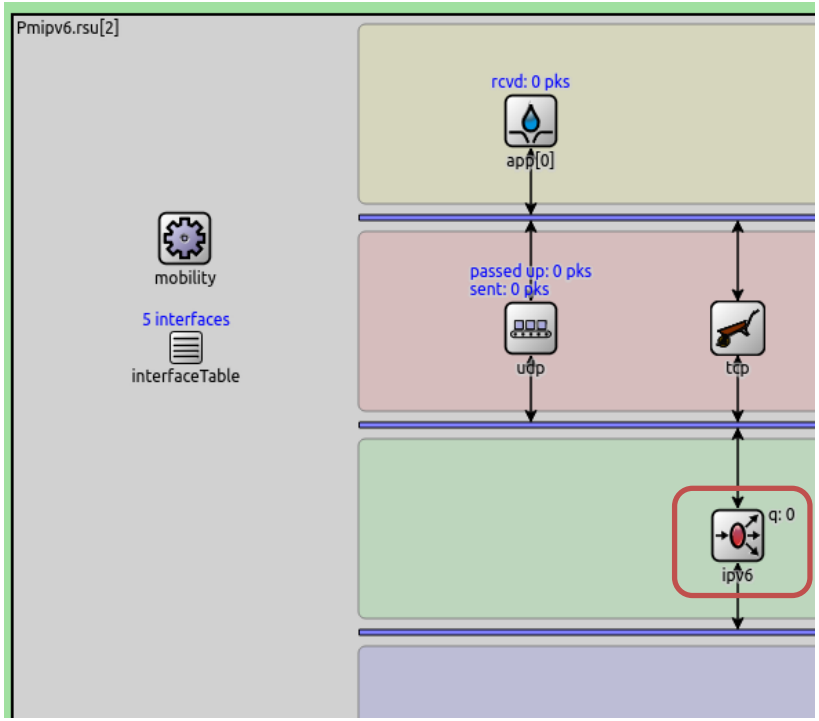
README.md

ipwave-hackathon-ietf-116

Demonstration

URL: https://youtu.be/V0BoXDkIg5c





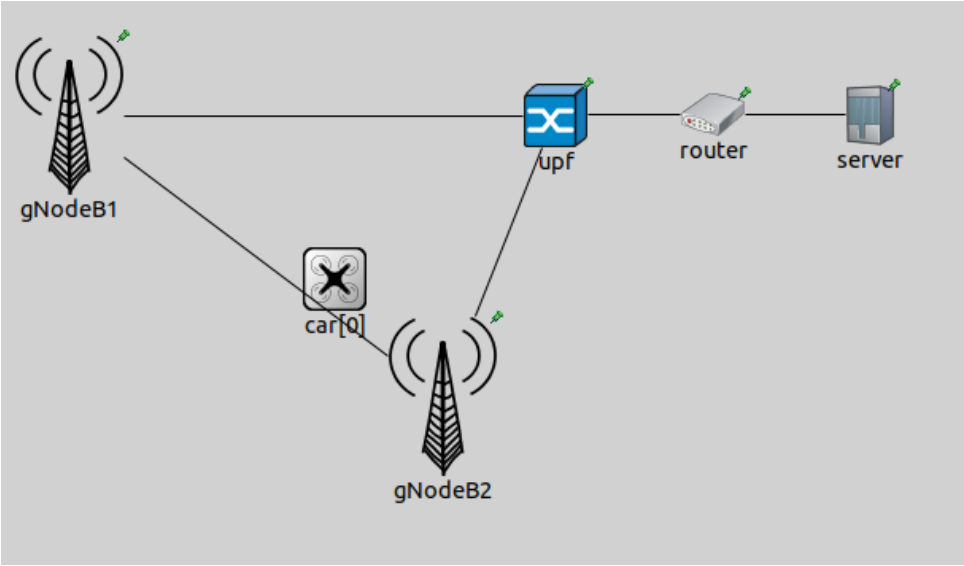
What we learned

- We changed the 802.11-OCB handover from IPv4 to IPv6 in OMNET++ simulation.

- We learned how to transform the IPv4's address structure into IPv6's address structure.

Next Step

- We will extend our handover with IPv6 multi-hop communication in 5G V2X communication.



Wrap Up

Hackathon Team

Champions:

- **Jaehoon Paul Jeong (SKKU)**
- **Yiwen (Chris) Shen**

Professor:

- **Younghan Kim (SSU)**

Researchers:

- **Jung-Soo Park (ETRI)**
- **Yunchul Choi (ETRI)**

Students:

- **Bien Aime Mugabarigira (SKKU)**
- **Hyeonah Jung (SKKU)**
- **Junhee Kwon (SKKU)**

Hackathon Team Photo

Appendix

(1) Simulation Environment Preparation Guide

(2) Implementation Environment

Simulation Environment

- OS: Ubuntu 16.04
- Simulators:
  - OMNeT++ 6.0
- GNU GCC 5.4
- Open Sources:
  - https: https://github.com/ipwave-hackathon-ietf/ipwave-hackathon-ietf-116
  - SUMO 0.25.0

Configurations

- Install OMNeT++ following the procedure in the installation manual: https://doc.omnetpp.org/omnetpp/InstallGuide.pdf
- Import projects in OMNeT++ workspace
  - Import INET by File → Import → General → Existing projects into workspace
  - Similarly, as INET, import SimuLTE

Project References

- Activate project features to ensure INET and Veins runs correctly.
- Right-click on Veins project and choose Properties Then, Project References and tick inet
- Run the scenario in veins:
  .sumo-launchd.py -v -c sumo-gui
- Run the simulation by:
  Veins → example → Pmipv6→ omnetpp and in set inifile configuration, choose Default