Implementing a mobility scenario using SDN and Ryu Framework

Iulisloi Zacarias
Janaína Schwarzrock

Instituto de Informática
Universidade Federal do Rio Grande do Sul (UFRGS)

CMP182 – Redes de Computadores I, 2016/I

Prof. Dr. Edison Pignaton de Freitas
Prof. Dr. Luciano Paschoal Gaspary

Outline

Scenario Characterization

- Proposed mobility scenario
- A stream video server
- Clients play the video streamed by the server
- Hosts can disconnect and connect from switches / access points

Scenario Diagram (Zone)

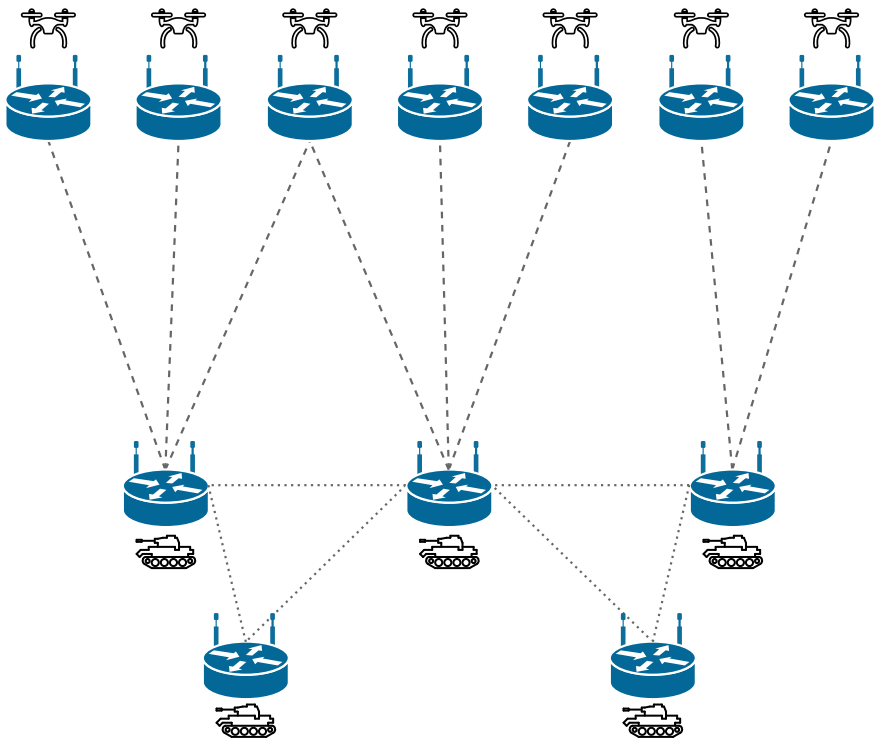

Figure: Dashed and dotted lines represent communication links among Unmanned Aerial Vehicles and Ground Vehicles

Scenario Diagram (Axis)

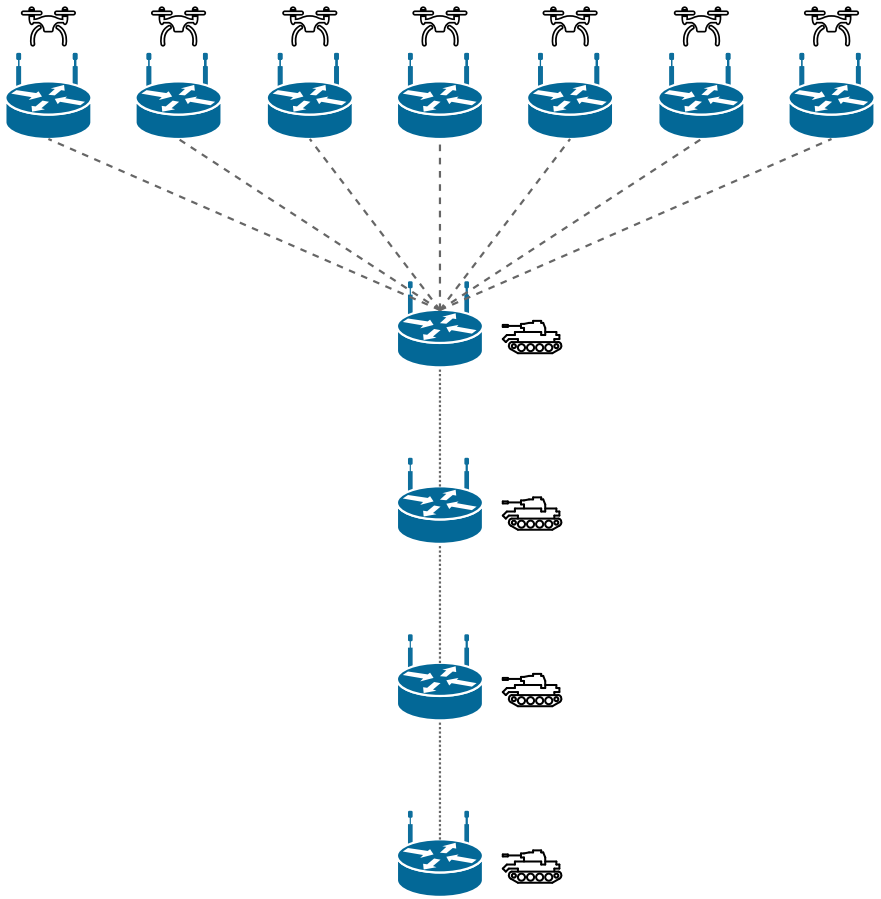

Figure: Dashed and dotted lines represent communication links among Unmanned Aerial Vehicles and Ground Vehicles

Links and Problem Characterization

- Links
  - Bandwidth (Max. throughput)
  - Packet loss expected (amount in %)
- Video and Photo format (size, resolution, format)
- Mobility of nodes (dynamic behaviour)
- Video client location (The video needs to be accessible from all vehicles?)

Flow Diagram of Topology Discover

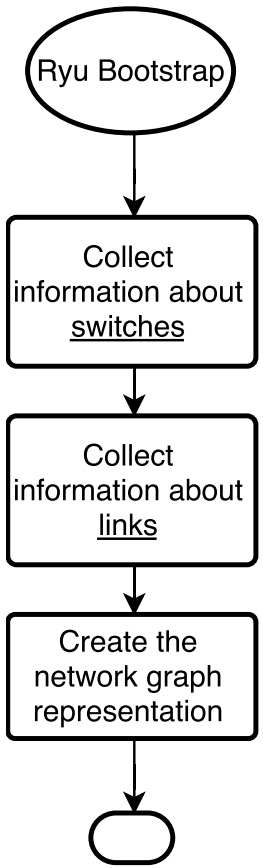

Figure: Topology discovering process

- Wait for Ryu topology start-up
- Use of LLDP Messages for discovering
- Create a network representation using NetworkX

Flow Diagram of Packet In

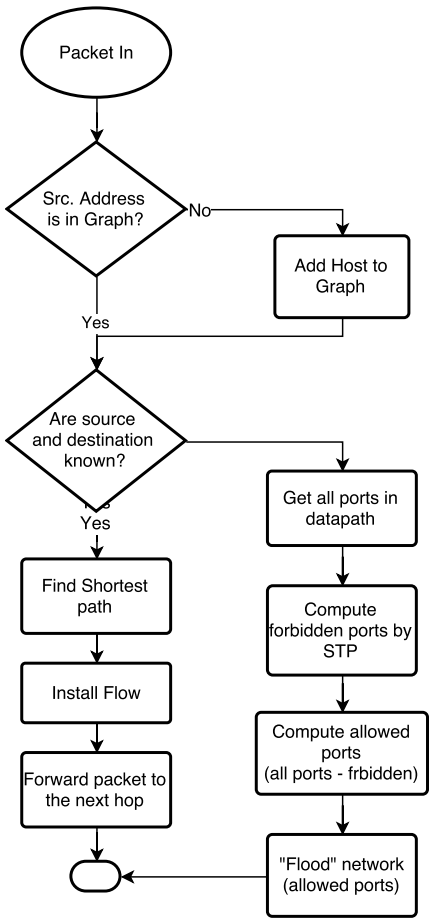

Figure: Overview of Packet In event

- Discover new hosts
- Find shortest path (known destination host)
- Discover destination using ARP messages
- Broadcast of ARP in "controlled" network path
- Discover of destination on ARP Response

Flow Diagram of Port Status Change

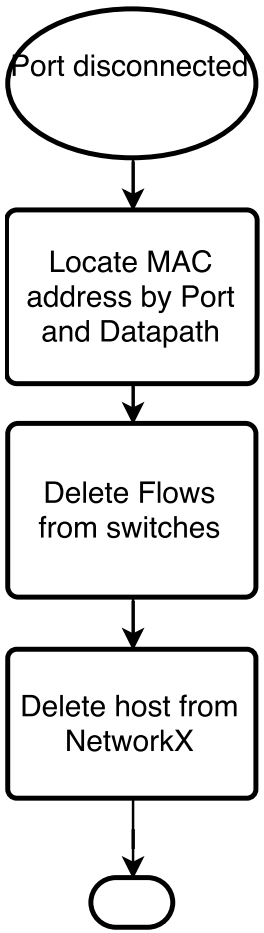

Figure: Topology discovering process

- OpenFlow sends an "Port disconnected" message
- Locate hardware address traversing the NetworkX graph
- Delete flows from switches in old path
- Delete the host disconnected from graph (NetworkX)

Tools and Libraries Employed

- "Mininet" used to simulate networks, hosts and topology
- Controller developed using Ryu Framework (v: 4.3)
- OpenFlow (v: 1.3.x) and Open vSwitch(**)
- Video playback and metrics using VLC Media Player
- NetworkX library for graph manipulation

ARP Messages forwarded

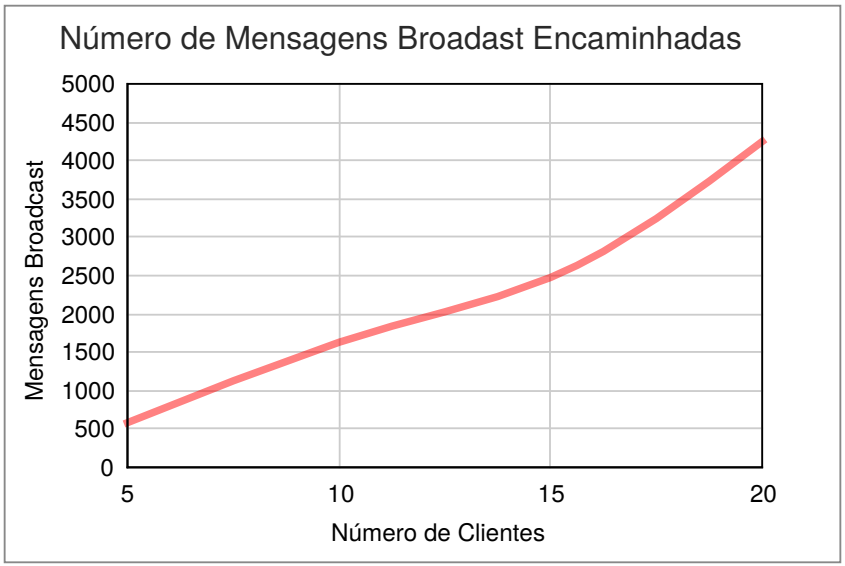

Figure: Amount of forwarded ARP Broadcast*

OpenFlow Rules

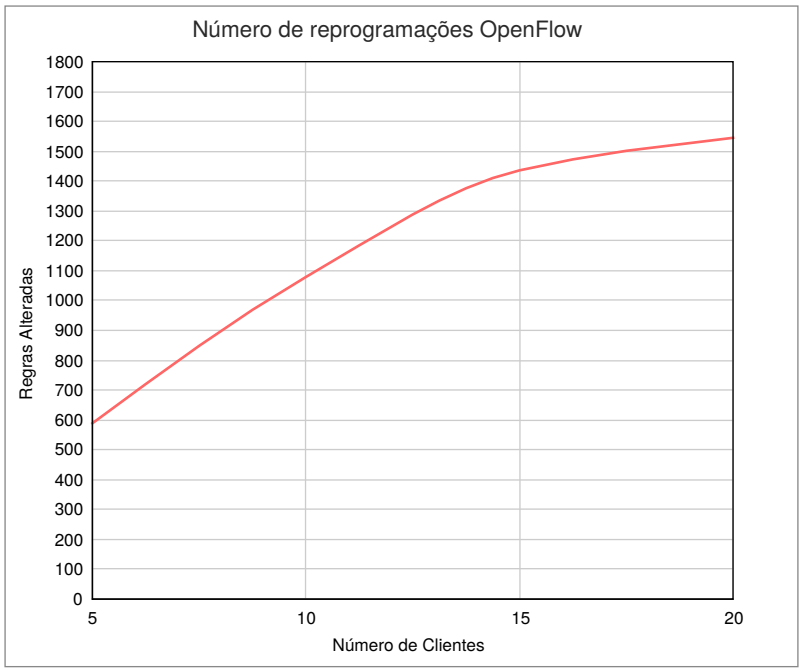

Figure: OpenFlow reprogramming rules

Conclusion

- Mobility simulation using Mininet offer some challenges
- Open vSwitch is not fully compatible with OpenFlow 1.3
- Traditional Spanning Tree Protocol (IEEE 802.1D) is not suitable for high mobility scenario

References

[1] Bob Lantz, Brandon Heller, and Nick McKeown.
Network in a Laptop: Rapid Prototyping for Software-Defined Networks.
9th ACM Workshop on Hot Topics in Networks, Oct. 20-21
Monterey, CA.

[2] Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart.
Exploring network structure, dynamics, and function using NetworkX
in Proceedings of the 7th Python in Science Conference, p. 11–15, Aug. 2008.
Pasadena, CA

[3] Nippon Telegraph and Telephone Corporation
Ryu SDN Framework
https://osrg.github.io/ryu/