This project has received funding from "HORIZON 2020" the European Union's Framework Programme for research, technological development and demonstration under grant agreement no 645220

Road-, Air- and Water-based Future Internet Experimentation

Project Acronym:	RAWFIE		
Contract Number:	**645220**		
Starting date:	**Jan 1st 2015**	**Ending date:**	**Dec 31^2t^, 2018**

Deliverable Number and Title	UNSURPASSED D1.2: First Progress Report		
Confidentiality	RAWFIE internal	**Deliverable type**[1]	R
Deliverable File	D12_UNSURPASSED.pdf	**Date**	31/1/2018
Approval Status[2]		**Version**	V1.0
Contact Person	Stavros Toumpis	**Organization**	AUEB
Phone	+30 210 8203551 +30 6974 740063	**E-Mail**	toumpis@aueb.gr

[1] Deliverable type: P(Prototype), R (Report), O (Other)
[2] Approval Status: WP leader, 1st Reviewer, 2nd Reviewer, Advisory Board

AUTHORS TABLE

Name	Company	E-Mail
Esmerald Aliai	AUEB	aliai16@aueb.gr
Georgia Dimaki	AUEB	ginadimaki135@gmail.com
Nikos Fotiou	AUEB	fotiou@aueb.gr
Petros Getsopoulos	AUEB	pgetsopoulos@hotmail.com
Stavros Toumpis	AUEB	toumpis@aueb.gr

REVIEWERS TABLE

Name	Company	E-Mail

DISTRIBUTION

Name / Role	Company	Level of confidentiality[3]	Type of deliverable

CHANGE HISTORY

Version	Date	Reason for Change	Pages/Sections Affected
1.0	31/1/2018	Original Submission	

[3] Deliverable Distribution: PU (Public, can be distributed to everyone), CO (Confidential, for use by consortium members only), RE (Restricted, available to a group specified by the Project Advisory Board).

Abstract: In this deliverable,

- we present a roadmap for integrating/embedding our work with RAWFIE (notably, showing that our planned experiments can be supported by the EDL/Kafka framework with minimal additions to it),
- we present our planned experiments using the EDL language, and, finally,
- we provide some suggestions to the developers of the EDL for future improvements.

Keywords: EDL, Experiments, UNSURPASSED

Table of Contents

Part II

List of Figures

List of Tables

Part III: Main Section

1. Introduction

Deliverable D1.1, titled "UNSURPASSED D1.1: Draft Software extensions for Tasks 1, 4" provided information on the progress of the project in the first three months. As the deliverable at hand covers only one extra month, the information found in Deliverable D1.1 on the overall progress of the project is not repeated here. Regarding our progress since the submission of Deliverable D1.1, we briefly mention:

- Another scientific publication, titled "*Asymptotics of the Packet Speed and Cost in a Mobile Wireless Network Model*" and authored by S. Toumpis together with his collaborators I. Kontoyiannis, R. Cavallari, and R. Verdone, was submitted to IEEE ISIT 2018 and is currently under review. The publication can be found appended at the end of the deliverable.
- We procured more equipment, comprised, notably, of 7 more Raspberry Pi's, allowing us to start large multihop experiments in earnest.
- We have started using and accessing the EDL Web tool, allowing us to author our first experiment drafts and also work towards developing a strategy for integrating our work in the RAWFIE platform. The results appear in the subsequent sections.

2. Integrating UNSURPASSED and RAWFIE

The integration of UNSURPASSED in the RAWFIE platform is very important to the success of UNSURPASSED. The integration must satisfy the following goals:

- Our experiments must be executed with minimal modifications to the RAWFIE platform.
- Any modifications to the RAWFIE platform should be of use to other projects as well, so that the effort put by the RAWFIE and UNSURPASSSED teams is well-spent.
- The additions to the code we provide should be available to subsequent users as an integral part of the rest of the platform.

Taking into consideration the above strategic aims, as well as the current state of development of the platform, we have decided to integrate our hardware and software in the RAWFIE platform using the already available concepts of **sensors** and **algorithms.**

NETSENSOR

Firstly, our software, whether installed on RAWFIE hardware, or additional hardware that will be added to the USVs, will appear as (virtual) sensors. This has the advantage of reusing a type of structure that is central to RAWFIE, and already well-supported. Therefore, at the start of each experiment, the initialization of our protocols will take place as the activation of a sensor, called NETSENSOR. Deactivating the NETSENSOR sensor, either at the end of the experiment, or beforehand, will trigger the transmission of statistics about the sent/routed/received load to the experiment controller, through Apache Kafka.

NETALGORITHM

Secondly, creating and routing traffic will be implemented as an *algorithm* (in the EDL parlance). The algorithm will be named NETALGORITHM and will be accepting the following arguments:

NETALGORITHM (START, END, DEST, STREAM, STAMP, ADHOC_PROT, DTN_PROT, ICN_PROT, SEC_PROT)

- START specifies the time at which the creation/routing of the traffic will start.
- END specifies the time at which the creation/routing of the traffic will stop.
- DEST specifies the destination node of the traffic.
  - If it is other than 0 and -1, it is the address of a specific other node in the network or an ICN identifier (in the case ICN routing is used).
  - If it is 0, then all other nodes are destinations, i.e., the traffic is broadcast.
  - If it is -1, then no traffic is created, and the node is only acting as a relay or destination.
- STREAM specifies the type of traffic. For example
  - STREAM=[CBR, SIZE, RATE, ENC] specifies that the node creates constant bitrate traffic with packets of a size equal to SIZE created with a rate equal to RATE. ENC specifies if the network traffic (both payload and headers) will be encrypted (ENC=1) or unencrypted (ENC=0).Furthermore:
    - If an ICN protocol is used and DEST ≠-1, then stream describes the stream of requests sent to the destination.
    - If an ICN protocol is used and DEST=-1, then STREAM describes the traffic stream sent in response to other, requesting nodes.
    - If an ICN protocol is not used and DEST≠-1, then STREAM describes the traffic stream sent to specified destinations.
  - STREAM=0 specifies that no traffic is created.
- STAMP specifies a label for the traffic stream created by this command. This argument is useful to have when there are multiple instances of the NETALGORITHM command in the same EDL script, and the gathered statistics should correspond to the right traffic stream.
- ADHOC_PROT specifies the ad hoc protocol used, and possibly parameters. For example:
  - ADHOC_PROT=0: No ad hoc protocol is used.
  - ADHOC_PROT=DIRECT: Data sources exchange traffic with their destinations when they are in direct contact, otherwise data packets are immediately dropped.
  - ADHOC_PROT=BABEL: the BABEL routing protocol is used.
  - ADHOC_PROT=BATMAN: the BATMAN routing protocol is used.
- DTN_PROT: The DTN protocol used. For example:
  - DTN_PROT=0: No DTN protocol is implemented.
  - DTN_PROT=DIRECT: Data sources exchange traffic with their destinations whenever they are in direct contact; in the meantime, packets are stored.
  - DTN_PROT=2HOP: Data sources send packets to everyone they meet; a node carrying a packet of another node will only transmit it to its destination.
  - DTN_PROT=EPIDEMIC: The epidemic protocol will be used.
- ICN_PROT specifies the ICN protocol used. For example:

    - ICN_PROT=0: no ICN protocol is used.
    - ICN_PROT=CCN: The CCN ICN protocol will be used.
- SEC_PROT specifies the security protocol used. For example:
    - SEC_PROT=0: no security protocol is used.
    - SEC_PROT= ID: Identity-based encryption is used.
    - SEC_PROT=IDPRE: Identity-based encryption with support for proxy re-encryption is used.

This high-level EDL command will be relayed, through Apache Kafka, to our installed modules, which will in turn start the creation of traffic. We note that this traffic will be on a secondary network, so that control traffic through the current network is not affected.

Using Events

In our discussions with the RAWFIE staff, we were informed of a very useful pending addition to the EDL (and the underlying network controller framework). In particular, sensor outputs can produce events that can modify the movement of nodes. This feature can be used in our experiments as well. For example, a node could be instructed to stay in one waypoint until it receives a specific data packet, signifying, e.g., that offloading is complete, at which point it moves to the next waypoint. As the functionality was not available at the time of writing of the deliverable, we did not make use of it, however we expect to make use of it at a later stage, in a subset of our experiments.

Accommodating other experimenters

The integration outlined above using NETSENSOR and NETALGORITHM satisfies the strategic goals specified at the start of the section, notably towards accommodating other experimenters.

Imagine, for example, that, in the future, another experimenter creates sensed traffic and wants to make use of the routing capabilities of our protocols. She will use the NETALGORITHM structure, with a STREAM argument that she will design, and she will also implement the underlying interfacing between her sensor and our software, without having to worry about how the traffic will be routed.

3. Descriptions of all planned experiments in the EDL

In this section we present an overview of the planned experiments, using the Experiment Description Language. The section is organized as follows: for each experiment, we first provide a description of the experiment in plain English and then in the EDL language. To help the reader understand differences between various parts of the same EDL script and subsequent scripts, some lines appear with yellow highlight.

Simplifications used

We stress that a complete and final description of the conducted experiments at this stage is impossible, since, firstly, many of the conducted experiments will depend on the outcomes of previous ones, and, secondly, we have not set up the software and hardware yet. Therefore, the experiments listed here are tentative descriptions of the first batch of experiments to be conducted; the later batches will be decided based on the outcome of the first batch. In particular, for each experiment, basic parameters (such as the locations of nodes, and the duration of the experiment, which in all cases set to 100) are chosen arbitrarily. Their precise values will be decided once the experiments start in earnest, and we have a clear idea of the conditions at the site and the hardware used (i.e., the actual range of the antennas used, etc.) The important point in this part of the deliverable is to ensure that the UNSURPASSED vision is consistent and can be embedded efficiently in the RAWFIE platform, as described in previous sections.

Also, to help the reader, we have included comments in the script, starting with the % character. We have also interpreted the Route structure to be such that

- When there is only one WP, the node remains stationary for the whole experiment duration.
- When there are two WPs, WP1 and WP2, the node travels continuously from one to the other, i.e., the node visits the following WPs, for the complete duration of the experiment:

$$WP1 \rightarrow WP2 \rightarrow WP1 \rightarrow WP2 \rightarrow WP1 \rightarrow WP2 \rightarrow \cdots$$

- When there are three or more WPs, and the first WP *coincides* with the last WP, i.e., the waypoints are WP1, WP2, …., WPN, WP1, then the node performs a continuous loop, i.e., visits the WPs as follows, for the duration of the experiment:

$$WP1 \rightarrow WP2 \rightarrow \cdots \rightarrow WPN \rightarrow WP1 \rightarrow WP2 \rightarrow \cdots \rightarrow WPN \rightarrow WP1 \rightarrow WP2 \rightarrow \cdots$$

- When there are three or more WPs, and the first WP *does not coincide* with the last, i.e., the waypoints are WP1, WP2, …., WPN, with WPN≠WP1, then the node moves along the continuous loop

$$WP1 \rightarrow WP2 \rightarrow WP3 \rightarrow \cdots \rightarrow WPN \rightarrow WP(N-1) \rightarrow WP(N-2) \rightarrow \cdots \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow \cdots$$

In practice, this functionality can be supported by inserting more WPs in the script, but we use the above convention to keep script lengths at a manageable level, for our purposes.

Finally, in order to keep the length of the deliverable at hand at a manageable level, we remove some details from the script. Notably,

- If multiple nodes behave in an identical manner, excepting their locations, we do not list the EDL code for all of them, but the code for only one of them, commenting on the rest.
- We do not include the preamble of each script, but only its part related to nodes.

Performance Metrics and parameter tuning

For all experiments, the performance metrics will be the usual ones, i.e., the achieved throughput, the packet delivery delay, and the packet delivery rate (i.e., the probability that a packet is delivered). The performance metrics will be calculated at the end of the experiment at the nodes, when the NETSENSOR of each node will be deactivated, and the results will be relayed through Apache Kafka. Also, one parameter that will be tuned in all experiments, except from those explicitly stated later on for each experiment, will be the intensity of the traffic.

Experiment 1.1: Totally connected topology (Task 1: Ad Hoc Routing)

This experiment will be used for debugging purposes, and for exploring hardware/software limitations: All 10 USVs will be roughly collocated, and made to exchange packets in pairs, so that bugs can be removed and the limits of the channel can be discovered. Parameters that will be changing will be 1) the total number of nodes, 2) the distance between them, and 3) the number of pairs. See Fig. 1 for the topology.

Table 1: Experiment 1.1

```
Node
        ID Flexus1   % This is the source of the traffic stream
                Route[
                        WP<0, +144.47, +7.06, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(0,            % Start  of traffic
                                100,              % End of traffic
                                Flexus2,          % Destination of  traffic
                                [CBR,1,100,0],  % Profile  of traffic
                                1                 %  Traffic stamp
                                DIRECT            % Ad hoc protocol used. No multihop routing is allowed
                                0, 0, 0)          % DTN, ICN, and security  protocols
                ]
~Node

Node
        ID Flexus2   % This is the destination of the traffic stream
                Route[
                        WP<0, +167.53, +19.76, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(0,            % Start  of traffic
                                  100,            % End of traffic
                                  -1,             % Destination of traffic. No traffic created
                                  0               % Profile  of traffic
                                  1               %  Traffic stamp
                                  DIRECT          % Ad hoc protocol used. No multihop routing is allowed
                                  0, 0, 0)        % DTN, ICN, and security  protocols
                ]
~Node

% Code for node pairs 3-4, 5-6, 7-8, and 9-10 is similar and so is omitted.
% Code for subsequent traffic streams is simlar and so is omitted.
```

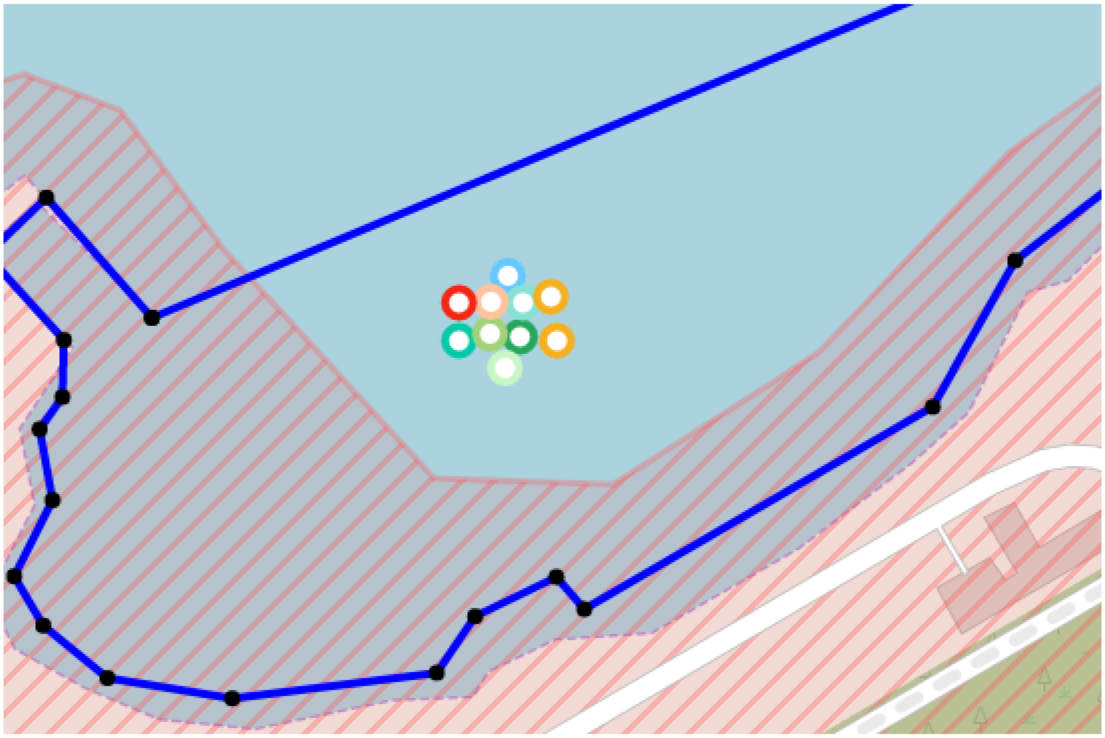

Figure 1: Experiments 1.1, 3.1, 4.1, and 4.3

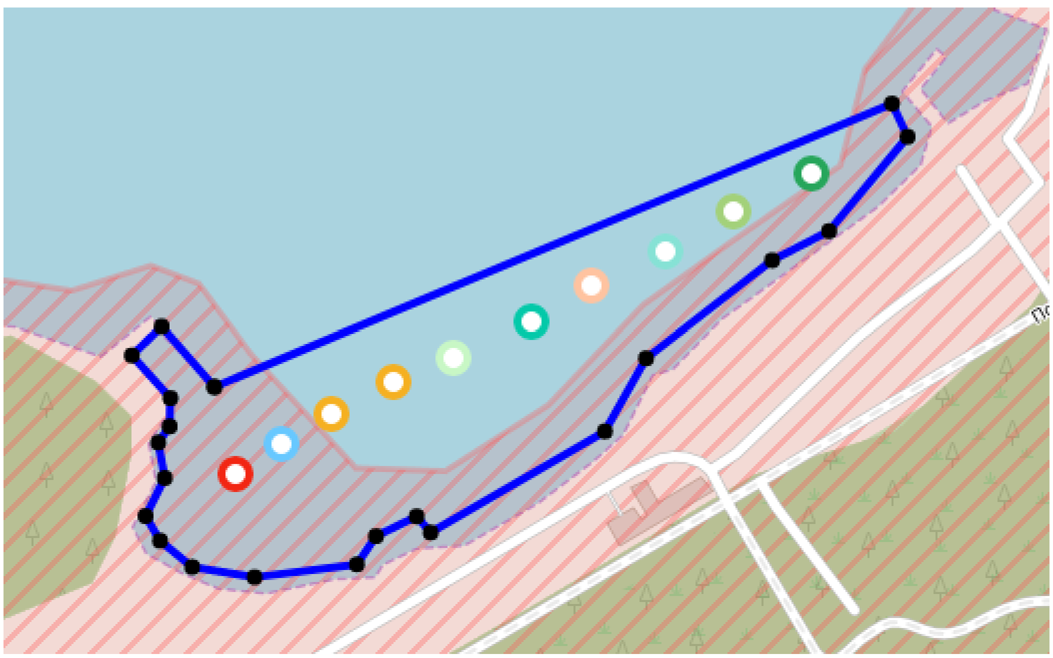

Figure 2: Experiment 1.2

Experiment 1.2: Multihop flow without mobility (Task 1: Ad Hoc Routing)

This experiment will explore the potential of the platform to support multihop traffic when the nodes do not move. The 10 USVs will be placed along a straight line and will not be moving for the duration of the experiment, and the node on the one end of the chain will be sending high-rate traffic to the other end of the chain. A parameter that will be tuned to various values will be the distance between the consecutive USVs in the chain. See Figure 2 for the topology.

Table 2: Experiment 1.2

```
Node
        ID Flexus1          % This is the source of the traffic stream
                Route[
                        WP<0, +19.76, -81.88, +0.0>
                ]
                Sensor[
                        Time 0    Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated

                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(0,          % Start  of traffic
                                                100,              % End of traffic
                                                Flexus10,         % Destination of traffic
                                                [CBR,1,100,0],    % Profile  of traffic
                                                1                 %  Traffic stamp
                                                BABEL             % Ad hoc protocol used
                                                0, 0, 0)          % DTN, ICN, and security  protocols
                ]
~Node

Node
        ID Flexus10         % This is the destination of the traffic stream
                Route[
                        WP<0, +561.88, +200.47, +0.0>
                ]
                Sensor[
                        Time 0    Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(0,        % Start  of traffic
                                                100,            % End of traffic
                                                -1,             % Destination of traffic
                                                0               % Profile  of traffic
                                                1               %  Traffic stamp
                                                BABEL           % Ad hoc protocol used
                                                0, 0, 0)        % DTN, ICN, and security  protocols
                ]
~Node

% Code for nodes 2 to 9 is similar to code of node 10 and so is omitted.
% Code for subsequent traffic streams is simlar and so is omitted.
```

Experiment 1.3: Multihop flow with mobility (Task 1: Ad Hoc Routing)

This experiment will study the effects of mobility. It will be identical to the previous one, with the exception that the nodes will be moving continuously between two locations such that the chain is maintained. Parameters that will be changing will be 1) the distance between the consecutive USVs in the chain, and 2) the speed of the nodes. See Figure 3 for the topology.

Table 3: Experiment 1.3

```
Node
        ID Flexus1              % This is the source of the single stream
                Route[
                        WP<0, +42.35, -124.24, +0.0>
                        WP<1, +0.00, -57.41, +0.0>
                ]
                Sensor[
                        Time 0    Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(0,            % Start  of traffic
                                        100              % End of traffic
                                        Flexus10,        % Destination of traffic
                                        [CBR,1,100,0],   % Profile of traffic
                                        1                % Traffic stamp
                                        BABEL            % Ad hoc protocol used
                                        0, 0, 0)         % DTN, ICN, and security  protocols
                ]
~Node
Node
        ID Flexus10             % This is the destination of the single stream
                Route[
                        WP<0, +519.53, +128.00, +0.0>
                        WP<1, +477.17, +192.00, +0.0>
                ]
                Sensor[
                        Time 0    Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(0,           % Start  of traffic
                                        100              % End of traffic
                                        -1,              % Destination of traffic
                                        0,               % Profile of traffic
                                        1                % Traffic stamp
                                        BABEL            % Ad hoc protocol used
                                        0, 0, 0)         % DTN, ICN, and security  protocols
                ]
~Node

% Code for nodes 2 to 9 is similar to code of node 10 and so is omitted.
% Code for subsequent traffic streams is similar and so is omitted.
```

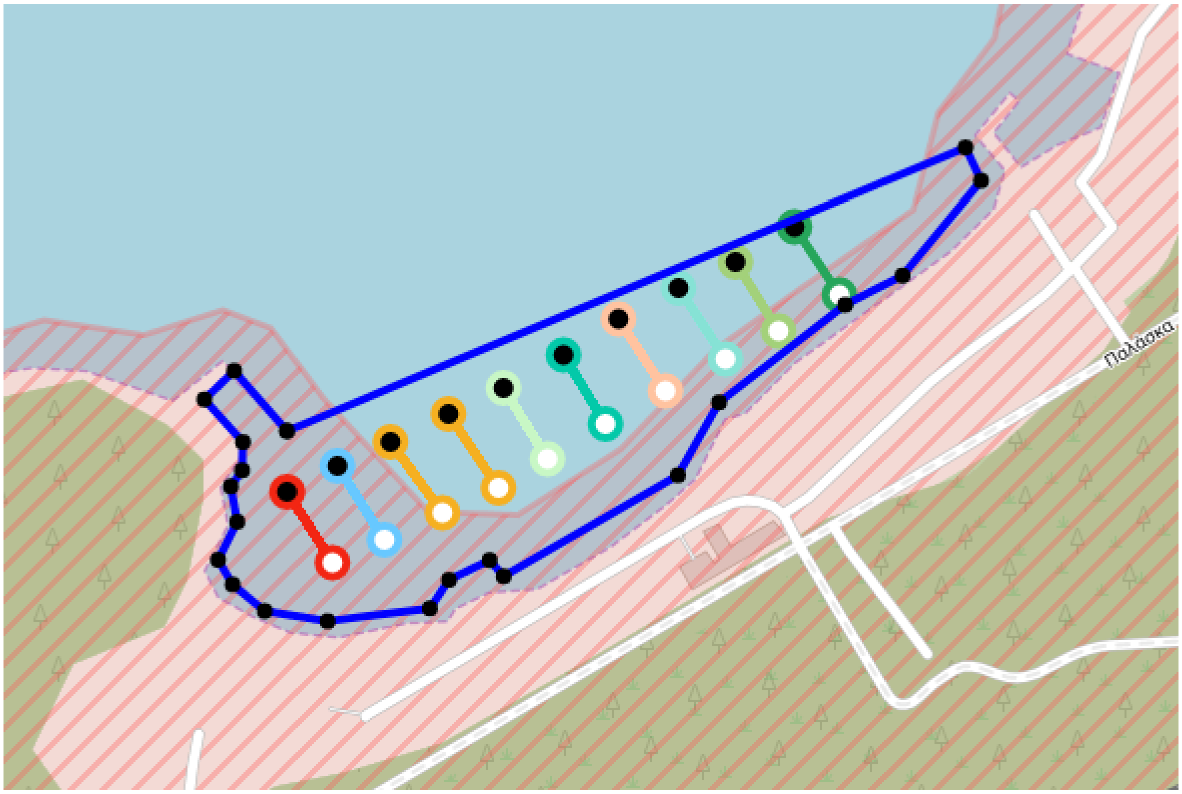

Figure 3: Experiments 1.3 and 3.2

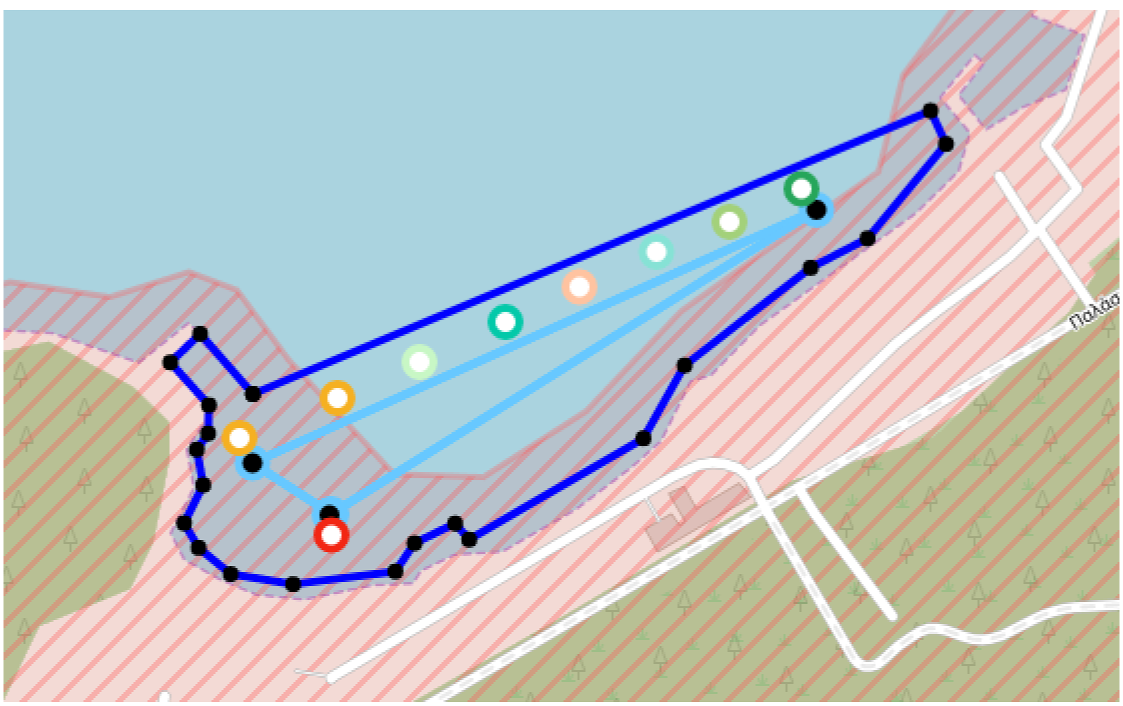

Figure 4: Experiment 2.1

Experiment 2.1: Data off-loading (Task 2: DTN Routing)

This experiment is made to simulate a WSN environment, in which a single data mule is used for the off-loading of data created by a set of sensors. 9 nodes will be placed in different locations of the network, so that none of these nodes can communicate with any other. Eight of the nodes, acting as sensors, will be creating data traffic for the ninth one, which will be acting as a base station. Finally, the 10th node, acting as a data mule, will be moving consecutively between the 9 nodes, providing connectivity. See Figure 4 for the topology. Once this experiment is successfully concluded, we will substitute the data muling USV with a UAV.

Table 4: Experiment 2.1

```
Node
        ID Flexus9  % This is the base station. It creates no traffic.
                Route[
                        WP<0, +73.88, -132.71, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(0,     % Start  of traffic
                                                100,                 % End of traffic
                                                -1,                  % Destination of traffic
                                                0,                   % Profile  of traffic
                                                1                    % Traffic stamp
                                                DIRECT,              % Ad hoc protocol used
                                                2HOP,                % DTN protocol used
                                                0, 0)                % ICN and security  protocols
                ]
~Node
Node
        ID Flexus10  % This is the data mule. Likewise, it creates no traffic.
                Route[
                        WP<0, +73.88, -112.94, +0.0> % The data mule is circling through the waypoints.
                        WP<1, -0.47, -64.94, +0.0>
                        WP<2, +530.35, +173.17, +0.0>
                        WP<0, +73.88, -112.94, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(0,     % Start  of traffic
                                                    100,         % End of traffic
                                                    -1,          % Destination of traffic
                                                    0,           % Profile  of traffic
                                                    1            %  Traffic stamp
                                                    DIRECT       % Ad hoc protocol used
                                                    2HOP         % DTN protocol used
                                                    0, 0)        % ICN and security  protocols
                ]
~Node
```

```
Node
        ID Flexus1  % This is one of the sensors, and creates traffic for the base station
                Route[
                        WP<0, +12.88, -82.71, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(0,    % Start of traffic
                                          100,                % End of traffic
                                          Flexus9,            % Destination of traffic
                                          [CBR,1,100,0],      % Profile  of traffic
                                          1                   %  Traffic stamp
                                          DIRECT              % Ad hoc protocol used
                                          2HOP                % DTN protocol used
                                          0, 0)               % ICN and security  protocols
                ]
~Node

% Code for nodes 2 to 8 is similar to code of node Flexus1 and so is omitted.
% Code for subsequent traffic streams is similar and so is omitted.
```

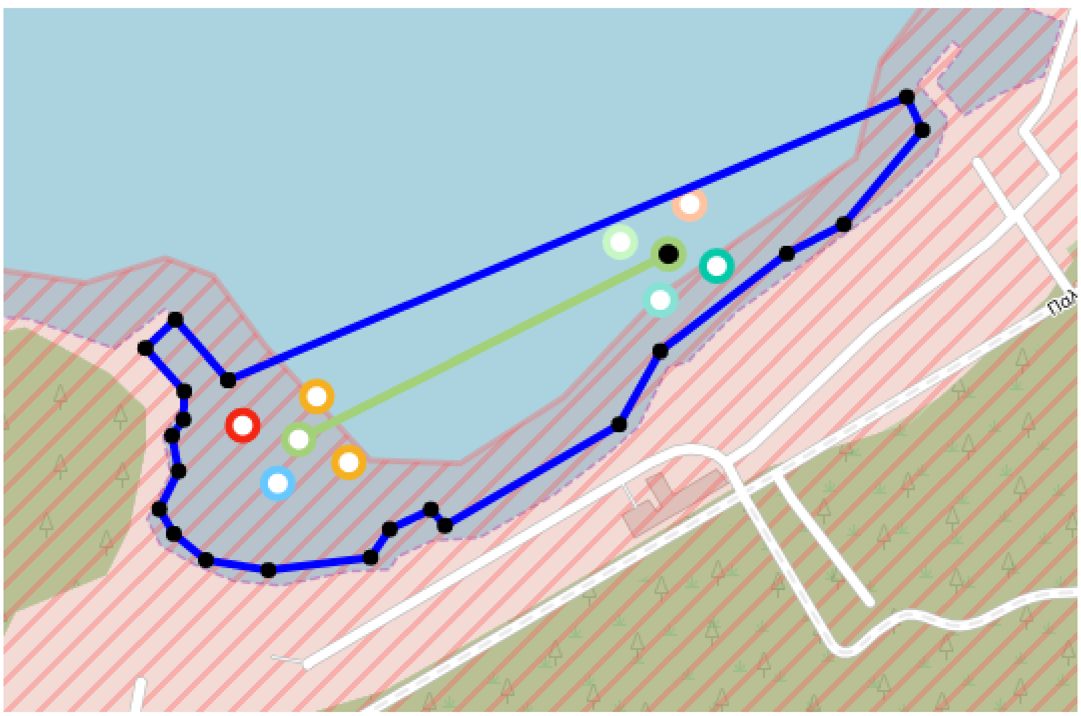

Figure 5: Experiments 2.2 and 4.2

Experiment 2.2: Multi-stream data muling (Task 2: DTN Routing)

This experiment will study the capabilities of nodes to perform data muling in a multi-destination, dense environment, which is more challenging than the previous one, of Experiment 2.1 (which is less dense, and where there is a single destination). 8 USVs will be placed in two groups, of 4 USVs each, so that the nodes of each group can directly communicate with each other but not with any node of the other group. These nodes will not be moving. Each node of the first group would like to send a distinct traffic stream to one node of the other group. A 9th USV will be moving continuously between the two groups, briefly pausing when it arrives at the center of each group, thus acting as a data mule, receiving packets from one group and passing them to the other one. Parameters that will be changing will be 1) the traffic rate, 2) the speed of the 9th node, and 3) the amount of time that the node will be pausing at the center of each group. See Figure 5 for the topology.

Table 5: Experiment 2.2

```
Node
        ID Flexus1  % This is the source of the first traffic stream
                Route[
                        WP<0, +13.70, -42.66, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(0,            % Start  of traffic
                                                1,                  % End of traffic
                                                Flexus2,            % Destination of traffic
                                                [CBR,1,100,0],      % Profile  of traffic
                                                1                   %  Traffic stamp
                                                DIRECT              % Ad hoc protocol used
                                                2HOP                % DTN protocol used
                                                0, 0)               % ICN and security  protocols
                ]
~Node
Node
        ID Flexus2  % This is the destination of the first traffic stream
                Route[
                        WP<0, +368.52, +129.57, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(0,        % Start  of traffic
                                                1,              % End of traffic
                                                -1,             % Destination of traffic
                                                0,              % Profile  of traffic
                                                1               %  Traffic stamp
                                                DIRECT          % Ad hoc protocol used
                                                 2HOP           % DTN protocol used
                                                0, 0)           % ICN and security  protocols
```

```
                    ]
~Node

Node
          ID Flexus9  % This is the data mule. Likewise, it creates no traffic
                    Route[
                            WP<0, +66.40, -55.84, +0.0>
                            WP<1, +66.40, -55.84, +0.0>         % Pause time
                            WP<2, +413.70, +118.28, +0.0>
                            WP<3, +413.70, +118.28, +0.0>     % Pause time
                    ]
                    Sensor[
                              Time 0     Name NETSENSOR set Activated
                              Time 100 Name NETSENSOR set Dectivated
                    ]
                    DataManagement[
                              Time 0 Algorithm NETALGORITHM(0,       % Start  of traffic
                                                              1,              % End of traffic
                                                              -1,             % Destination of traffic
                                                              0,              % Profile  of traffic
                                                              1               %  Traffic stamp
                                                              DIRECT       % Ad Hoc Protocol used
                                                              2HOP          % DTN protocol used
                                                              0, 0)           % ICN and security  protocols
                    ]
~Node

% Code for node pairs 3-4, 5-6, and 7-8 is similar to code for pair 1-2, and so is omitted.
% Code for subsequent traffic streams is similar and so is omitted.
```

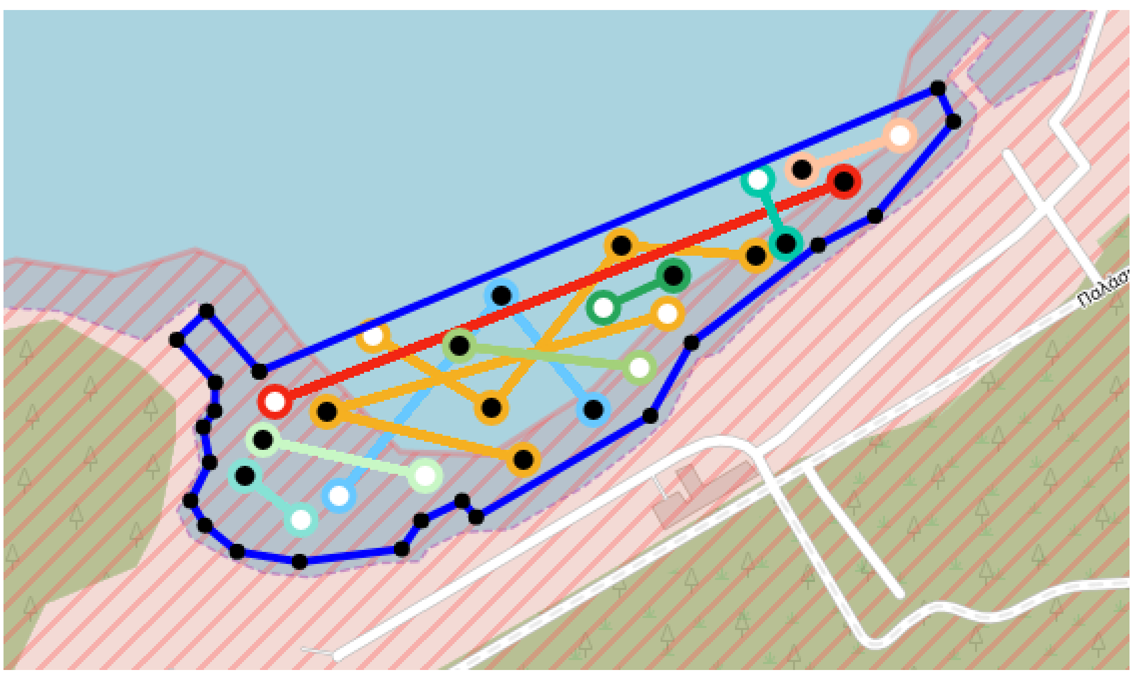

Figure 6: Experiments 2.3 and 3.3

Experiment 2.3: Epidemic Routing (Task 2: DTN Routing)

This experiment will study the performance of epidemic routing in a challenging environment with multiple moving nodes. 10 USVs will be moving continuously in a pseudorandom fashion. Each node will be creating traffic for one of the other nodes. Nodes will be exchanging data packets according to the epidemic routing protocol.

Table 6: Experiment 2.3

```
Node
        ID Flexus1  % This is the source of the first traffic stream
                Route[
                        WP<0, +14.64, -28.54, +0.0>
                        WP<1, +549.22, +178.51, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(0,        % Start  of traffic
                                                  100,              % End of traffic
                                                  Flexus2,          % Destination of traffic
                                                  [CBR,1,100,0],    % Profile  of traffic
                                                  1                 % Traffic stamp
                                                  BABEL             % Ad hoc protocol used
                                                  EPIDEMIC          % DTN protocol used
                                                  0, 0)             % ICN and security  protocols
                ]
~Node
Node
        ID Flexus2  % This is the destination of the first traffic stream
                Route[
                        WP<0, +74.88, -117.01, +0.0>
                        WP<1, +227.34, +71.22, +0.0>
                        WP<2, +313.93, -36.07, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(0,        % Start  of traffic
                                                   100,         % End of traffic
                                                   -1,          % Destination of traffic
                                                    0,          % Profile  of traffic
                                                    1           %  Traffic stamp
                                                    BABEL       % Ad hoc protocol used
                                                    EPIDEMIC % DTN protocol used
                                                    0, 0)       % ICN and security  protocols
                ]
~Node
% Code for node pairs 3-4, 5-6, and 7-8 is similar to code for pair 1-2, and so is omitted.
% Code for subsequent traffic streams is similar and so is omitted.
```

Experiment 3.1: ICN over Totally connected topology (Task 3: ICN Routing)

Similarly to Experiment 1.1, this experiment will be used for debugging purposes and for exploring the limitations of the hardware and software. All 10 USVs will be located close to each other and will be limited to direct connectivity. Five USVs will request a piece of content located in another USV. Parameters that will be changing will be 1) the total number of nodes, 2) the distance between them, and 3) the number of items each node requests. The topology depicted in Fig. 1 will be used.

Table 7: Experiment 3.1

```
Node
        ID Flexus1      % This will be the source of the stream of requests
                Route[
                        WP<0, +144.47, +7.06, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(0,               % Start  of traffic
                                100,                % End of traffic
                                Flexus2,            % Prefix of the content item name
                                [CBR,1,100,0],  % Profile  of traffic describing the requests
                                1                   % Traffic stamp
                                DIRECT              % Ad hoc protocol used. No multihop routing is allowed
                                0,                  % DTN protocol used
                                CCN,                % ICN protocol used
                                0)                  % Security protocol used
                ]
~Node
Node
        ID Flexus2       % This is the destination of the first traffic stream, so it will be providing the content
                Route[
                        WP<0, +167.53, +19.76, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(0,               % Start  of traffic
                                100,            % End of traffic
                                -1,             % Destination of traffic. No traffic created
                                [CBR,1,100,0],  % Profile  of traffic describing the provided content
                                1               % Traffic stamp
                                DIRECT          % Ad hoc protocol used
                                0,              % DTN protocol used
                                CCN,            % ICN protocol used
                                0)              % no security  protocol used
                ]
~Node
% Code for node pairs 3-4, 5-6, 7-8, and 9-10 is similar and so is omitted.
% Code for subsequent traffic streams is simlar and so is omitted.
```

Experiment 3.2: ICN over Multihop flow with mobility (Task 3: ICN Routing)

Similarly to Experiment 1.3, this experiment will study ICN performance over a multihop topology using an ad hoc routing protocol and under node mobility. In this experiment, the 10 USVs will be placed along a straight line and they will be moving continuously between two locations such that the chain is maintained. Parameters that will be changing will be 1) the distance between the consecutive USVs in the chain, and 2) the speed of the nodes. The topology of Fig. 3 will be used in this experiment.

Table 8: Experiment 3.2

```
Node
        ID Flexus1 % This will be the source of the stream of requests
                Route[
                        WP<0, +42.35, -124.24, +0.0>
                        WP<1, +0.00, -57.41, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(
                                0,                  % Start  of traffic
                                100,                % End of traffic
                                Flexus10,           % Content item name prefix
                                [CBR,1,100,0],      % Profile  of traffic
                                1                   %  Traffic stamp
                                BABEL,0,CCN,0)      % Ad hoc, DTN, ICN, and security protocols
                ]
~Node

Node
        ID Flexus10 % This is the destination of the first traffic stream, so it will be providing the content
                Route[
                        WP<0, +519.53, +128.00, +0.0>
                        WP<1, +477.17, +192.00, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(
                                0,                  % Start  of traffic
                                100,                % End of traffic
                                 -1,                % Destination of traffic. No traffic created
                                [CBR,1,100,0],      % Profile  of traffic describing the provided content
                                1                   %  Traffic stamp
                                BABEL,0,CCN,0)  % Ad hoc, DTN, ICN, and security protocols
                ]
~Node

% Code for nodes 2 to 9 is similar to code of node 10 and so is omitted.
% Code for subsequent traffic streams is similar and so is omitted.
```

Experiment 3.3: ICN over Epidemic Routing (Task 3: ICN Routing)

This experiment will study the performance of ICN in a topology where a DTN protocol (in particular, epidemic routing) is used. In this experiment 10 USVs will be moving continuously in a pseudorandom fashion. Each node will be creating traffic requesting items located in one of the other nodes. Content requests will be forwarded according to the epidemic routing protocol. The topology of this experiment is depicted in Figure 6.

Table 9: Experiment 3.3

```
Node
        ID Flexus1  % This is the source of the first traffic stream
                Route[
                        WP<0, +14.64, -28.54, +0.0>
                        WP<1, +549.22, +178.51, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(
                                0,                      % Start  of traffic
                                100,                    % End of traffic
                                Flexus2,                % Content item name prefix
                                [CBR,1,100,0],          % Profile  of traffic
                                1                       %  Traffic stamp
                                BABEL,2HOP,CCN,0)     % Ad hoc, DTN, ICN, and security protocols
                ]
~Node
Node
        ID Flexus2  % This is the destination of the first traffic stream, so it will be providing the content
                Route[
                        WP<0, +74.88, -117.01, +0.0>
                        WP<1, +227.34, +71.22, +0.0>
                        WP<2, +313.93, -36.07, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(
                                0,                      % Start  of traffic
                                100,                    % End of traffic
                                -1,                     % Content item name prefix
                                [CBR,1,100,0],          % Profile  of traffic
                                1                       % Traffic stamp
                                BABEL,2HOP,CCN,0)  % Ad hoc, DTN, ICN, and security protocols
                ]
~Node

% Code for node pairs 3-4, 5-6, and 7-8 is similar to code for pair 1-2, and so is omitted.
% Code for subsequent traffic streams is similar and so is omitted.
```

Experiment 4.1: Content Encryption (Task 4: Security)

This experiment will use a topology of totally connected nodes (as depicted in Fig.1). The experiment will be used for debugging the IBE implementations, as well as for measuring its computational and communication overhead. All 10 USVs will be located close to each other and will have direct connectivity. Five USVs will be continuously requesting pieces of content located in another node. Parameters that will be changing will be 1) the security parameters size (e.g., size of keys, size of system parameters), 2) the size of the transmitted items, and 3) the number of items each node requests.

Table 10: Experiment 4.1

```
Node
        ID Flexus1     % This will be the source of the stream of requests
                Route[
                        WP<0, +144.47, +7.06, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(
                                0,                          % Start  of traffic
                                100,                          % End of traffic
                                 Flexus2,                   % Prefix of the content item name
                                 [CBR,1,100,0],             % Profile  of traffic.
                                 1                          %  Traffic stamp
                                DIRECT, 0, CCN, ID)    % Ad hoc, DTN, ICN, and security protolcs used

               ]
~Node
Node
        ID Flexus2    % This is the destination of the first traffic stream, so it will be providing the content
                Route[
                        WP<0, +167.53, +19.76, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(
                                  0,                          % Start  of traffic
                                  100,                        % End of traffic
                                  -1,                         % Destination of traffic. No traffic created
                                  [CBR,1,100,0],              % Profile  of traffic describing the provided content
                                   0                          % Profile  of traffic
                                  1                            %  Traffic stamp
                                  DIRECT, 0, CCN, ID)    % Ad hoc, DTN, ICN, and security protolcs used
        ]
~Node

% Code for node pairs 3-4, 5-6, 7-8, and 9-10 is similar and so is omitted.
% Code for subsequent traffic streams is simlar and so is omitted.
```

Experiment 4.2: Proxy re-encryption (Task 4: Security)

This experiment will evaluate the Identity-based proxy re-encryption (IB-PRE) algorithm. For this reason, the data muling topology of Fig. 5 will be used. In particular 8 USVs will be placed in two groups, of 4 USVs each, so that the nodes of each group can directly communicate with each other but not with any node of the other group. These nodes will not be moving. Each node of the first group would like to communicate with one node of the other group. A 9th USV will be moving continuously between the two groups, briefly pausing when it arrives at the center of each group, thus acting as a data mule, receiving packets from one group and passing them to the other one. In contrast to the Experiment 2.2, the mule will pause for as much time is required to receive and transmit all packets. The latter USV will hold also the role of the "proxy", i.e., it will be responsible for re-encrypting the transferred packets. Parameters that will be changing will be 1) the size of the security parameters, 2) the size of the transmitting items.

Table 11: Experiment 4.2

```
Node
        ID Flexus1   % This will be the source of the stream of requests
                Route[
                        WP<0, +13.70, -42.66, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(
                                        0,              % Start  of traffic
                                        100,            % End of traffic
                                        Flexus2,        % Prefix of the content item name
                                        [CBR,1,100,0],  % Profile  of traffic
                                        1               %  Traffic stamp
                                        DIRECT          % Ad hoc protocol used
                                        2HOP            % DTN protocol used
                                        CCN,            % ICN protocol
                                        IDPRE)          % security  protocols
                ]
~Node
Node
        ID Flexus2  % This is the destination of the first traffic stream, so it will be providing the content
                Route[
                        WP<0, +368.52, +129.57, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(
                                    0,              % Start  of traffic
                                    100,            % End of traffic
                                    -1,             % Destination of traffic. No traffic created
                                    [CBR,1,100,0],  % Profile  of traffic describing the provided content
                                    1               %  Traffic stamp
```

```
                                          DIRECT        %  Ad hoc protocol
                                          2HOP           %  DTN protocol
                                          CCN,           %  ICN protocol
                                          IDPRE)         %  Security  protocol
                ]
~Node

Node
        ID Flexus9  % This is the data mule. It creates no traffic.
                Route[
                      WP<0, +66.40, -55.84, +0.0>
                     WP<1, +413.70, +118.28, +0.0>
                ]
                Sensor[
                      Time 0     Name NETSENSOR set Activated
                      Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                      Time 0 Algorithm NETALGORITHM(0,        % Start  of traffic
                                      100,                 % End of traffic
                                      -1,                  % Destination of traffic. No traffic created
                                       [CBR,1,100,0],     % Profile  of traffic describing the provided content
                                      1                    %  Traffic stamp
                                      DIRECT               % Ad hoc protocol
                                      2HOP                 % DTN protocol
                                      CCN,                 % ICN protocol
                                      IDPRE)               % Security  protocol
                ]
~Node
% Code for node pairs 3-4, 5-6, and 7-8 is similar to code for pair 1-2, and so is omitted.
% Code for subsequent traffic streams is similar and so is omitted.
```

Experiment 4.3: Private requests (Task 4: Security)

This experiment will utilize the fully connected ad hoc topology (Fig. 1) to evaluate private requests. All 10 USVs will be located close to each other and will have direct connectivity. Each USV will request a piece of content located in another node. The request will be encrypted using Identity-Based Encryption. For encrypting a request, the content name prefix will be used as a key. Parameters that will be changing will be 1) the total number of nodes, 2) the size of the security parameters, and 3) the number of items each node requests.

Table 12: Experiment 4.3

```
Node
        ID Flexus1
                Route[
                        WP<0, +144.47, +7.06, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(0,                % Start  of traffic
                                 1,                % End of traffic
                                 Flexus2,          % Prefix of the content item name
                                 [CBR,1,100,1],  % Profile  of traffic.
                                 1                 %  Traffic stamp
                                 DIRECT            % Ad hoc protocol used. No multihop routing is allowed.
                                 0,                % DTV
                                 CCN,              % ICN
                                 ID)               % security protocols
                ]
~Node

Node
        ID Flexus2
                Route[
                        WP<0, +167.53, +19.76, +0.0>
                ]
                Sensor[
                        Time 0     Name NETSENSOR set Activated
                        Time 100 Name NETSENSOR set Dectivated
                ]
                DataManagement[
                        Time 0 Algorithm NETALGORITHM(0,                % Start  of traffic
                                 1,                % End of traffic
                                 -1,               % Destination of traffic. No traffic created
                                 0                 % Profile  of traffic
                                 1                 %  Traffic stamp
                                 DIRECT            % Ad hoc protocol used. No multihop routing is allowed.
                                 0,                % DTV
                                 CCN,              % ICN
                                 ID)               % security  protocols
                ]
~Node

% Code for node pairs 3-4, 5-6, 7-8, and 9-10 is similar and so is omitted.
% Code for subsequent traffic streams is simlar and so is omitted.
```

4. Suggestions for enhancing the EDL

We conclude the deliverable by suggesting a number of improvements for the EDL, taking into account our experience with it.

- In case a number of nodes behave in a similar manner, it would be useful to be able to describe the behavior of all of them jointly, by describing the behavior of a generic one, using parameters, and then specifying the parameters for each of them. This will simplify the writing of the scripts, will make debugging easier, and improve the scalability of the EDL.
- It would be useful to add a FOR loop functionality, for example in specifying waypoints, so that the experiment can be described more succinctly.
- The ability to insert comments will improve the readability of the script.
- It would be useful to the developer to be able to save a script, even if it has errors.

5. Acknowledgement

We would like to gratefully acknowledge the support of the RAWFIE team, especially Dr. K. Kolomvatsos, for providing us with crucial information on the EDL language and for helping us acclimatize with the EDL environment and the underlying constraints and features of the RAWFIE testbed environment.

Asymptotics of the Packet Speed and Cost in a Mobile Wireless Network Model

Ioannis Kontoyiannis*‡, Stavros Toumpis*, Riccardo Cavallari†, and Roberto Verdone†
*Department of Informatics, Athens University of Economics and Business, Greece
‡Department of Engineering, University of Cambridge, UK
†Department of Electrical, Electronic, and Information Engineering, University of Bologna, Italy
Email: i.kontoyiannis@eng.cam.ac.uk, toumpis@aueb.gr, {riccardo.cavallari, roberto.verdone}@unibo.it

***Abstract*—An infinite number of nodes move on $\mathbb{R}^2$ according to a random waypoint model; a single packet is traveling towards a destination (located at an infinite distance away) using combinations of wireless transmissions and physical transport on the buffers of nodes. In earlier work [1] we defined two performance metrics, namely, the long-term average speed with which the packet travels towards its destination, and the rate with which transmission cost accumulates with distance covered. Analytical expressions were derived for these metrics, under specific ergodicity assumptions. In this paper we give a precise description of the induced Markov process, we show that it is indeed (uniformly) geometrically ergodic, and that the law of large numbers holds for the random variables of interest. In particular, we show that the two performance metrics are well-defined and asymptotically constant with probability one.**

***Index Terms*—Delay-Tolerant Routing, Geographic Routing, Mobile Wireless Network, Packet speed/cost.**

I. Introduction

Numerous mobile wireless networks have recently been studied, where packets travel towards their destination using both wireless transmissions and physical transport on the buffers of nodes. Examples include satellite [2], vehicular [3], and pocket-switched [4] networks.

Motivated, in part, by these and related applications, in recent work [1], we studied a stochastic/geometric model for such a network, in which an infinite number of nodes move on the infinite plane according to a random waypoint model, and a single packet is traveling towards a destination located at an infinite distance away. In this setting, we defined the packet speed and the packet cost to be the limits (as the packet trajectory length goes to infinity) of the long-term average rates with which distance is covered over time and transmission cost is accumulated over distance covered, respectively. We computed explicit expressions for these limits, but under the provision that the strong law of large numbers (SLLN) holds for a collection of random variable (RV) sequences describing the evolution of the trajectory of the system.

In this work, we develop simple, natural conditions under which we prove that, indeed, the SLLN holds, thus providing a crucial step in the calculation of the performance metrics. As an intermediate result, we clarify the conditions needed for the long-term average rates to converge to their limits as well as the rate with which this occurs, thus illuminating the applicability of our results in practical settings.

In Section II we review the relevant network model [1]; in Section III we specify approximations and define quantities computed in [1] that will be needed in this work; our new results appears in Section IV; their proofs are placed in the Appendix. The present results together with those in [1] appear in a unified preprint [5].

II. Network Model

Regarding the node mobility model, at $t = 0$ we place an infinite number of nodes on $\mathbb{R}^2$ according to a Poisson point process (PPP) with density λ. Then, each node moves along a straight line, with speed v_0, changing its travel direction at the event times of a Poisson process of rate r_0. Nodes move independently of each other and select their travel direction independently of past travel directions and according to the uniform distribution. We describe travel directions using the angle $\theta \in [-\pi, \pi)$ they form with the positive x-axis.

Regarding the transmission cost, we assume that when a packet gets transmitted from a node A to a node B, such that the vector from A to B is $\mathbf{r}$, then the transmission incurs a cost $C(\mathbf{r})$. All transmissions are instantaneous.

Regarding the traffic, we assume a single packet is created, at time $t = 0$, at some node, and needs to travel towards a destination that is located at an infinite distance away, in the direction of the positive x-axis. The packet travels towards the destination according to some routing rule (RR) that uses both wireless transmissions and extended stays at node buffers.

Observe that the trajectory of the packet is comprised of linear segments, each segment corresponding to either a wireless transmission or a sojourn on the buffer of a node while the node is not changing its travel direction. For this reason, we break the journey of the packet into stages, indexed by $i = 1, 2, \ldots$, each stage corresponding to either a wireless transmission, in which case we call it a **wireless stage (WS)**, or a sojourn along a straight line segment, in which case we call it a **buffering stage (BS)**.

We describe each stage i in terms of a number of RVs. First, let Δ_i be its duration; note that $\Delta_i = 0$ if stage i is a WS. Second, let C_i be its transmission cost; note that $C_i = 0$ if stage i is a BS. Third, let Θ_i be the travel direction of the receiver, if stage i is a WS, or the travel direction of the packet holder, if stage i is a BS. Fourth, let $X_{W,i}$ be the change in the x-coordinate of the packet due to the wireless transmission; if the stage is a BS, then $X_{W,i} = 0$. Likewise, let $X_{B,i}$ be the change in the x-coordinate of the packet due to buffering; if stage i is a WS, then $X_{W,i} = 0$. Finally, let $X_i = X_{B,i}+X_{W,i}$.

We use two metrics to describe the performance of the RR. The first metric is the **speed** V_p, defined as the limit of the long-term average speed with which the packet is traveling towards its destination,

$$V_p = \lim_{n\to\infty} \frac{\sum_{i=1}^{n} X_i}{\sum_{i=1}^{n} \Delta_i}, \quad (1)$$

provided the limit exists. The second metric is the **cost**

$$C_p = \lim_{n\to\infty} \frac{\sum_{i=1}^{n} C_i}{\sum_{i=1}^{n} X_i}, \quad (2)$$

again whenever the limit exists. In this work, we will specify a RR for which we will show that both V_p and C_p exist and are constant with probability one.

To specify the RR, first, let the forwarding region (FR) $\mathcal{F}$ be a nonempty, closed, bounded, and convex subset of $\mathbb{R}^2$ that contains $\mathbf{0}$. Second, let the potential $U(\phi, \mathbf{r}) : [-\pi, \pi) \times \mathcal{F} \to \mathbb{R}$ be a continuous function that describes the suitability of a node located at $\mathbf{r}$ with respect to the packet holder and traveling in direction $\theta \in [-\pi, \pi)$, for receiving the packet; the larger the potential, the more suitable the node. Note that the current holder's potential is $U(\theta, \mathbf{0})$, where θ is its direction of travel. For any node A, let $\mathcal{F}(A)$ be the FR shifted by its location $\mathbf{r}_A$, i.e., $\mathcal{F}(A) = \mathcal{F} + \mathbf{r}_A$.

Having $\mathcal{F}$ and $U(\theta, \mathbf{r})$, the RR is simple: the packet constantly aims to be at the node with the largest potential among all nodes in $\mathcal{F}(A)$, where A is its current holder. Therefore, if A has the largest potential, the packet stays at its buffer; if another node B is found with a higher potential (because A changed its travel direction, or B changed its direction, or B entered $\mathcal{F}(A)$), then the packet is transmitted to B; that transmission may be immediately followed by one or more transmissions.

The following assumptions are introduced for reasons of mathematical convenience.

Assumption 1: If $|\theta_1| > |\theta_2|$, then $U(\theta_2, \mathbf{r}) > U(\theta_1, \mathbf{r})$: If the travel direction improves, the potential becomes better.

Assumption 2: If $U(\theta_1, \mathbf{r}_1) > U(\theta_2, \mathbf{r}_2)$, then $U(\theta_1, \mathbf{r}_1 - \mathbf{r}_3) > U(\theta_2, \mathbf{r}_2 - \mathbf{r}_3)$. The assumption means that if a node A is better than a node B according to some node, it will also be better than B according to all other nodes in their neighborhood. This assumption prevents routing loops.

Assumption 3: Let $\mathcal{K}(\theta, \theta')$ be the subset of the FR where $U(\theta', \mathbf{r}) > U(\theta, \mathbf{0})$. Therefore, nodes entering $\mathcal{K}(\theta, \theta')$ with travel direction θ' become eligible to receive the packet. Let $\mathbf{b}(s; \theta, \theta')$ be a parametrization of the boundary of $\mathcal{K}(\theta, \theta')$ with $s \in [0, 1]$. We assume that the derivative $\mathbf{b}'(s; \theta, \theta')$ exists a.e. in $[0, 1]$ and there is a constant M_b such that, where the derivative exists, $|\mathbf{b}'|(s; \theta, \theta') < M_b$.

Assumption 4: The value of $U(-\pi, \mathbf{r})$ is equal to a constant K for all $\mathbf{r} \in \mathcal{F}$. Therefore, the direction $\theta = -\pi$ is uniformly the worst, irrespective of the location $\mathbf{r}$ of a candidate neighbor. Note, however, that the behavior of $U(\theta, \mathbf{r})$ as a function of θ can strongly depend on $\mathbf{r}$, so that 'good' locations can be favored, in terms of the potential assigned to them, as long as nodes at those locations are not traveling in direction $-\pi$.

III. Preliminary Results

We introduce two approximations and a number of quantities that have been computed in the previous work [1]. The resulting expressions and their derivations, found in [1], [5], will be used in later sections.

The following intuitive approximation is introduced for reasons of mathematical tractability. It introduces errors, however simulations show that these errors are typically quite modest, i.e., on the order of no more than 10% [1], [5].

Second Order Approximation: *When a node A receives a packet from a node B, the mobility process is restarted, except that A maintains its position and travel direction and all created nodes within the intersection $\mathcal{F}(A) \cap \mathcal{F}(B)$ with potential greater than the potential of A are removed. Also, when a node A carrying the packet changes its travel direction θ to a θ', the mobility process is restarted, except that A maintains its position and travel direction and all created nodes within $\mathcal{F}(A)$ whose potential is greater than $\max\{U(\theta, \mathbf{0}), U(\theta', \mathbf{0})\}$ are removed.*

First, consider the setting where node A, traveling with direction θ, has just received a packet from node B, such that the location of A with respect to B is $\mathbf{r}$. Let $\mathcal{G}(\mathbf{r}) = \mathcal{F}(A) \cap (\mathcal{F}(B))^c$, i.e., the new region the packet discovers upon arriving at node A and where eligible nodes can be found. Let $E(N; \theta, \mathbf{r})$ be the expected number of nodes in $\mathcal{G}(\mathbf{r})$ with potential larger than the potential of A; let $P_E(\theta, \mathbf{r})$ be the probability that $\mathcal{G}$ is empty of such nodes; let $g(\theta', \mathbf{r}'; \theta, \mathbf{r})$ be the joint density of the location $\mathbf{r}'$ and the direction θ' of an eligble node to which the packet is immediately transmitted upon its arrival at A. Expressions for these three functions appear in [1], [5].

Secondly, consider the setting where the packet has, at time $t = 0$, just started traveling with direction θ on the buffer of a node A. With a slight abuse of notation we define the following rates:

1) Let $r_{\mathcal{A}}(\theta, \theta')$ be such that the infinitesimal probability that node A will change its direction from θ to a direction in $[\theta', \theta' + d\theta']$ and a new sojourn will commence at the same node in the time interval $[0, dt]$ is $r_{\mathcal{A}}(\theta, \theta')d\theta' dt$.
2) Let $r_{\mathcal{B}}(\theta, \theta', \mathbf{r}')$ be such that the infinitesimal probability that, within the time interval $[0, dt]$, node A will change its travel direction and this will precipitate a transmission to a node B located within a region of infinitesimal area dA centered at $\mathbf{r}'$ and traveling with direction in $[\theta', \theta + d\theta']$ is $r_{\mathcal{B}}(\theta, \theta', \mathbf{r}')d\theta' dA dt$.
3) Let $r_{\mathcal{C}}(\theta, \theta', \mathbf{r}')$ be such that the infinitesimal probability that, within the time interval $[0, dt]$, a node B located within a region of infinitesimal area dA centered at $\mathbf{r}'$ and traveling with some direction θ'' will change its direction to lie in $[\theta', \theta + d\theta']$ and will thus become eligible to receive the packet, is $r_{\mathcal{C}}(\theta, \theta', \mathbf{r}')d\theta' dA dt$.

4) Let $r_{\mathcal{D}}(\theta, \theta', s)$ be such that the infinitesimal probability that within the time interval $[0, dt]$, a node B traveling with direction θ' crosses the boundary $\mathbf{b}(s; \theta, \theta')$ in the section $[s, s+ds]$ is $r_{\mathcal{D}}(\theta, \theta', s)d\theta' ds dt$.
5) Let $r(\theta)$ be such that the infinitesimal probability that any of the above events will occur in the time interval $[0, dt]$ is $r(\theta)dt$.

Expressions for the above rates appear in [1], [5].

We now introduce our second approximation:

Time Invariance Approximation: *If at time $t = 0$ the packet arrived at node A and at time $t = t_0 > 0$ the packet is still with A and A has not changed travel direction, the conditional distribution of all future events describing the end of the current stage is the same as for $t_0 = 0$.*

Intuitively, as long as node A is carrying the packet, the mobility process of all other nodes is constantly regenerated, so that the probabilities of the various stage-ending events occurring remain fixed and given by the rates we have defined.

IV. Performance Metrics

A. The Markov chain

We define the **state** S_i associated with each stage $i \geq 1$, by $S_i \triangleq (\Theta_i, (X_{W,i}, Y_{W,i}))$ if stage i is a WS, and by $S_i \triangleq (\Theta_i, (0, 0))$ if stage i is a BS.

The associated **state space** in which each S_i takes values is $\mathcal{S} \triangleq \mathcal{S}_B \cup \mathcal{S}_W$, where the **buffering state space** $\mathcal{S}_B \triangleq [-\pi, \pi) \times \{\mathbf{0}\}$, and the **wireless state space** $\mathcal{S}_W \triangleq (-\pi, \pi) \times (\mathcal{F} - \{\mathbf{0}\})$.

Observe that, due to the Second Order Approximation, the process $\{S_i,\ i = 1, 2, \ldots\}$ forms a Markov chain: If $S_i = (\theta, \mathbf{0})$, i.e., stage i is a BS, then at the start of that stage the complete mobility model was restarted, except that the carrier A kept its direction of travel θ and its FR did not contain nodes with a potential higher than that of A, i.e., $U(\theta, \mathbf{0})$. Likewise, if $S_i = (\theta_i, \mathbf{r})$ with $\mathbf{r} \neq \mathbf{0}$, i.e., in stage i the packet was transmitted from a node B to a node A located at $\mathbf{r} \in \mathcal{F}(B)$, then, at the moment A received the packet, the whole mobility model was again restarted, except that A kept its direction of travel θ and all nodes with potential higher than $U(\theta, \mathbf{r})$ were expunged from $\mathcal{F}(A) \cap \mathcal{F}(B)$. In both cases, the complete information remaining about the network is captured in the current state.

We now define one last rate function. Let $r_{\hat{\mathcal{D}}}(\theta, \theta', \mathbf{r}')$ be such that the infinitesimal probability that within the time interval $[0, dt]$, a node B located within a region of infinitesimal area dA centered at $\mathbf{r}'$ and traveling with direction θ' becomes eligible by crossing $\mathbf{b}(s; \theta, \theta')$, for some s, is $r_{\hat{\mathcal{D}}}(\theta, \theta', \mathbf{r}')d\theta' dA dt$. A simple expression for $r_{\hat{\mathcal{D}}}(\theta, \theta', \mathbf{r}')$ can be easily computed from $r_{\mathcal{D}}(\theta, \theta', \mathbf{r}')$; the details are omitted.

The distribution of the chain $\{S_i\}$ may be described as follows, using the Time Invariance Approximation. We assume that $S_1 = s \in \mathcal{S}$ is an arbitrary initial state, and for each i, given $S_i = (\theta, \mathbf{r})$, the chain moves to a state $S_{i+1} = (\theta', \mathbf{r}')$ according to the following family of conditional distributions:

1) If $\mathbf{r} = \mathbf{r}' = \mathbf{0}$, the conditional density of S_{i+1} is

$$K_{BB}(\theta; \theta') = \frac{r_{\mathcal{A}}(\theta, \theta')}{r(\theta)}.$$

2) If $\mathbf{r} = \mathbf{0}$ and $\mathbf{r}' \neq \mathbf{0}$, the conditional density of S_{i+1} is

$$K_{BW}(\theta; \theta', \mathbf{r}') = \frac{r_{\mathcal{B}}(\theta, \theta', \mathbf{r}') + r_{\mathcal{C}}(\theta, \theta', \mathbf{r}') + r_{\hat{\mathcal{D}}}(\theta, \theta', \mathbf{r}')}{r(\theta)}.$$

3) If $\mathbf{r} \neq \mathbf{0}$ and $\mathbf{r}' \neq \mathbf{0}$, the conditional density of S_{i+1} is

$$K_{WW}(\theta, \mathbf{r}; \theta', \mathbf{r}') = g(\theta', \mathbf{r}'; \theta, \mathbf{r}).$$

4) If $\mathbf{r} \neq \mathbf{0}$ and $\mathbf{r}' = \mathbf{0}$, the conditional density of S_{i+1} is

$$K_{WB}(\theta, \mathbf{r}; \theta') = \delta(\theta' - \theta) P_E(\theta, \mathbf{r}).$$

We refer to $K_{BB}(\theta; \theta')$, $K_{BW}(\theta; \theta', \mathbf{r}')$, $K_{WW}(\theta, \mathbf{r}; \theta', \mathbf{r}')$, and $K_{WB}(\theta, \mathbf{r}; \theta')$ as **kernel functions**, since they can be used to fully specify the transition kernel of the chain $\{S_i\}$.

B. Ergodicity

In this section we establish that the Markov chain $\{S_i\}$ is ergodic, with a unique invariant distribution π, to which it converges at a geometric rate.

Let $\mathcal{L}_1$ denote the Lebesgue measure on $[-\pi, \pi)$, $\mathcal{L}_2$ denote the Lebesgue measure on $\mathcal{F}$, and $\delta_{\mathbf{0}}$ be the point mass at point $\mathbf{0} = (0, 0) \in \mathbb{R}^2$. We write ψ for the measure $\psi = \mathcal{L}_1 \times \delta_{\mathbf{0}} + \mathcal{L}_1 \times \mathcal{L}_2$, defined on the state space $\mathcal{S}$, equipped with the usual Borel σ-field. Our first result describes the long-term behavior of the chain $\{S_i\}$, and its consequences are stated in detail after that; see [6] for some relevant background on Markov chains. Theorem 1 is proved in the Appendix.

Theorem 1: The Markov chain is ψ-irreducible, aperiodic, and uniformly ergodic on the state space $\mathcal{S}$, with a unique invariant measure π to which it converges uniformly geometrically fast. In particular:

1) There are constants $B < \infty$ and $\rho \in (0, 1)$ such that, for any initial state $s \in \mathcal{S}$,

$$|P(S_n \in A | S_1 = s) - \pi(A)| \leq B\rho^n,$$

for all $n \geq 1$ and any (measurable) set $A \subset \mathcal{S}$.

2) For any (measurable) function $F : \mathcal{S} \to \mathbb{R}$ with $E_\pi[|F(S)|] < \infty$, as $n \to \infty$, with probability one,

$$\frac{1}{n} \sum_{i=1}^{n} F(S_i) \to E_\pi[F(S)],$$

for any initial state $s \in \mathcal{S}$, where $S \sim \pi$.

An important ingredient in the proof of Theorem 1 is the following domination condition, which will be verified in the Appendix. Intuitively, Lemma 1 says that, irrespective of the current state, with probability at least ϵ the chain will be in a uniformly distributed buffering state after two time steps.

Lemma 1: (Doeblin condition) Let μ denote the measure $\mathcal{L}_1 \times \delta_{\mathbf{0}}$ on $\mathcal{S}$. There is an $\epsilon > 0$ such that, for any (measurable) $A \subset \mathcal{S}$ and any $s \in \mathcal{S}$, we have:

$$P(S_{i+2} \in A | S_i = s) \geq \epsilon \mu(A).$$

Another ingredient of the proof of the ψ-irreducibility part of Theorem 1 is provided by the following one-step reachability bound. Lemma 2 is proved in the Appendix.

Lemma 2: Let μ' denote the measure $\mathcal{L}_1 \times \mathcal{L}_2$ on $\mathcal{S}_W$. For any (measurable) $A \subset \mathcal{S}_W$ with $\mu'(A) > 0$ there are $-\pi \le \theta_1' < \theta_2' < \pi$ such that,

$$P(S_{i+1} \in A | S_i = (\theta, \mathbf{0})) > 0, \quad \text{for all } \theta \in (\theta_1', \theta_2'). \tag{3}$$

The main implications of Theorem 1 for our results are stated in the following corollary, which is proved in the Appendix. In order to state it we need some additional definitions. Given an arbitrary state $S_1 = s = (\theta, (x_W, y_W))$ in $\mathcal{S}$, let Δ_1 be exponentially distributed with rate $r(\theta)$ if $(x_W, y_W) = \mathbf{0}$, and $\Delta_1 = 0$ otherwise. Similarly, for each $i \ge 2$, given $(S_1, \ldots, S_{i-1}, S_i = (\theta, (x_W, y_W)))$ and $(\Delta_1, \ldots, \Delta_{i-1})$, let Δ_i have the same distribution as Δ_1 given $(\theta, (x_W, y_W))$. Then $\{\bar{S}_i = (\Theta_i, (X_{W,i}, Y_{W,i}), \Delta_i)\}$ defines a new Markov chain, on the state space:

$$\bar{\mathcal{S}} = \big([-\pi, \pi) \times \{\mathbf{0}\} \times [0, \infty)\big) \cup \big((-\pi, \pi) \times (\mathcal{F} - \{\mathbf{0}\}) \times \{0\}\big).$$

Now suppose $S = (\Theta, (X_W, Y_W))$ has distribution π and let Δ be defined as before, conditional on S. Write $\bar{\pi}$ for the induced joint distribution of $\bar{S} = (\Theta, (X_W, Y_W), \Delta)$ on $\bar{\mathcal{S}}$.

Corollary 1: For any initial state $S_1 = s$, $\Delta_1 = \delta$, the following ergodic theorems hold with probability one,

$$\begin{aligned}
\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} X_{W,i} &= E_\pi(X_W), \\
\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} C_i &= E_\pi(C) \;=\; E_\pi(C(X_W, Y_W)), \\
\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \Delta_i &= E_{\bar{\pi}}(\Delta), \\
\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} X_{B,i} &= E_{\bar{\pi}}(X_B) \;=\; v_0 E_{\bar{\pi}}(\Delta \cos \Theta),
\end{aligned}$$

where $(\Theta, (X_W, Y_W), \Delta) \sim \bar{\pi}$ so that $(\Theta, (X_W, Y_W)) \sim \pi$.

As the final step of our analysis, we provide expressions for the performance metrics defined in Section II. The following results are immediate consequences of Corollary 1.

Corollary 2: For any initial state $S_1 = s$, $\Delta_1 = \delta$, the limits defining the performance metrics V_p and C_p in (1) and (2), respectively, exist with probability one, and are given by:

$$V_p = \frac{E_{\bar{\pi}}(X_W + v_0 \Delta \cos \Theta)}{E_{\bar{\pi}}(\Delta)}, \tag{4}$$

$$C_p = \frac{E_{\bar{\pi}}(C(X_W, Y_W))}{E_{\bar{\pi}}(X_W + v_0 \Delta \cos \Theta)}, \tag{5}$$

where $(\Theta, (X_W, Y_W), \Delta) \sim \bar{\pi}$.

APPENDIX

Proof of Lemma 1: It is obvious that it suffices to establish the result of the lemma for events of the form $A = A_0 \times \{\mathbf{0}\}$, for $A_0 \subset [-\pi, \pi)$. And by the uniqueness of Carathéodory extension, since the collection of all finite unions of intervals forms an algebra that generates the Borel σ-algebra of $\mathcal{S}$, it further suffices for A_0 to only consider closed intervals, $A_0 = [\theta_1, \theta_2]$; see, e.g., [7], [8] for details. So in the rest of the proof we restrict attention to events A of the form $A = [\theta_1, \theta_2] \times \{\mathbf{0}\}$.

Also note that, from the expressions for the rates given at [1], it is simple to obtain the following bounds on the transition rates $r_\mathcal{A}$, $r_\mathcal{B}$, $r_\mathcal{C}$, $r_\mathcal{D}$, and on $r(\theta)$:

$$\begin{aligned}
\frac{r_0}{2\pi} \exp\left[-\lambda |\mathcal{F}|\right] &\le r_\mathcal{A}(\theta, \theta') \le \frac{r_0}{2\pi}, && (6)\\
r_\mathcal{B}(\theta, \theta', \mathbf{r}'), r_\mathcal{C}(\theta, \theta', \mathbf{r}') &\le \frac{r_0 \lambda}{2\pi}, \\
r_\mathcal{D}(\theta, \theta', s) &\le \frac{M_b \lambda v_0}{\pi}, \\
r(\theta) &\le r_0 + r_0 \lambda |\mathcal{F}| + 2 M_b \lambda v_0. && (7)
\end{aligned}$$

Now, if s is of the form $s = (\theta, \mathbf{0})$ for some $\theta \in [-\pi, \pi)$, then for any $-\pi \le \theta_1 < \theta_2 < \pi$,

$$P\left(S_{i+1} \in [\theta_1, \theta_2] \times \{\mathbf{0}\} | S_i = (\theta, \mathbf{0})\right) = \int_{\theta_1}^{\theta_2} \frac{r_\mathcal{A}(\theta, \theta')}{r(\theta)}\, d\theta',$$

so that, using the lower bound in (6) and the upper bound in (7), we have that, for some fixed constant $\delta_1 > 0$:

$$P\left(S_{i+1} \in [\theta_1, \theta_2] \times \{\mathbf{0}\} | S_i = (\theta, \mathbf{0})\right) \ge \delta_1 (\theta_2 - \theta_1). \tag{8}$$

Then, using the Markov property and applying (8) twice,

$$\begin{aligned}
&P\left(S_{i+2} \in [\theta_1, \theta_2] \times \{\mathbf{0}\} | S_i = (\theta, \mathbf{0})\right) && (9)\\
&\quad\ge\; P\left(S_{i+1} \in \mathcal{S}_B | S_i = (\theta, \mathbf{0})\right) \delta_1 (\theta_2 - \theta_1) \\
&\quad\ge\; 2\pi \delta_1^2 (\theta_2 - \theta_1). && (10)
\end{aligned}$$

Similarly, if s is of the form $s = (\theta, \mathbf{r})$ for some $\theta \in [-\pi, \pi)$ and $\mathbf{r} \in \mathcal{F}$, then by the Markov property,

$$\begin{aligned}
&P\left(S_{i+2} \in [\theta_1, \theta_2] \times \{\mathbf{0}\} | S_i = (\theta, \mathbf{r})\right) \\
&\quad\ge\; P\left(S_{i+1} = (\theta, \mathbf{0}),\ S_{i+2} \in [\theta_1, \theta_2] \times \{\mathbf{0}\} | S_i = (\theta, \mathbf{r})\right) \\
&\quad=\; P_E(\theta, \mathbf{r}) \int_{\theta_1}^{\theta_2} \frac{r_\mathcal{A}(\theta, \theta')}{r(\theta)}\, d\theta' \\
&\quad\ge\; \delta_1 P_E(\theta, \mathbf{r})(\theta_2 - \theta_1) = \delta_1 \exp\{-E(N; \theta, \mathbf{r})\}(\theta_2 - \theta_1) \\
&\quad\ge\; \delta_1 \exp\{-\lambda |\mathcal{F}|\}(\theta_2 - \theta_1). && (11)
\end{aligned}$$

The last inequality holds because, using the definition of $E(N; \theta, \mathbf{r})$, we clearly have $E(N; \theta, \mathbf{r}) \le \lambda |\mathcal{F}|$.

Combining (10) and (11) yields the required result, with $\epsilon = \min\{2\pi \delta_1^2, \delta_1 \exp\{-\lambda |\mathcal{F}|\}\}$. □

Proof of Lemma 2: Since A has positive Lebesgue measure, we can find a rectangle of the form $I = [\theta_1, \theta_2] \times [x_1, x_2] \times [y_1, y_2] \subset \mathcal{S}_W$ with a nonempty interior, such that $\mu'(A \cap I) > 0$. The idea of the main argument here is to show that there is a range of angles (θ_1', θ_2') such that, when the current packet holder travels with a direction in (θ_1', θ_2'), there is a strictly nonzero probability that there are ineligible nodes in $[x_1, x_2] \times [y_1, y_2]$ that can become eligible by changing their direction of travel to a better one within the range $[\theta_1, \theta_2]$.

Since U is continuous, the image $U(I)$ of I is a closed interval $[a, b]$. And since I has a nonempty interior, we must have $a < b$ by Assumption 1. Also, by Assumptions 1 and 4, and noting that $\theta_1 > -\pi$ in order to have $I \subset \mathcal{S}_W$, we must have $b > a > U(-\pi, \mathbf{0})$.

Next, pick some c, d such that $U(-\pi, \mathbf{0}) < c < d < \min\{U(0, \mathbf{0}), a\}$, and let θ_1' and θ_2' be such that $U(\theta_1', \mathbf{0}) = c$ and $U(\theta_2', \mathbf{0}) = d$; such angles are guaranteed to exist by the intermediate value theorem. Also, observe that U is continuous on the compact set $[-\pi, 0] \times [x_1, x_2] \times [y_1, y_2]$, so it is uniformly continuous there, which implies that there is a $\theta_B > -\pi$ with $U(\theta, \mathbf{r}) < c$ for all $\theta \in [-\pi, \theta_B]$ and all $\mathbf{r} \in [x_1, x_2] \times [y_1, y_2]$.

Now take $(\theta', \mathbf{r}') \in I$ and $\theta \in (\theta_1', \theta_2')$ arbitrary. We will bound $r_{\mathcal{C}}(\theta, \theta', \mathbf{r}')$, given, in Section V-B of [1], by

$$r_{\mathcal{C}}(\theta, \theta', \mathbf{r}') = \frac{\lambda r_0}{4\pi^2} \mathbf{1}\left[U(\theta', \mathbf{r}') > U(\theta, \mathbf{0})\right] \times \int_{-\pi}^{\pi} \mathbf{1}[U(\theta'', \mathbf{r}') < U(\theta, \mathbf{0})]\, d\theta'',$$

from below. First note that $U(\theta', \mathbf{r}') > d$ and $U(\theta, \mathbf{0}) < d$, therefore $\mathbf{1}[U(\theta', \mathbf{r}') > U(\theta, \mathbf{0})] = 1$. Also, we have $U(\theta'', \mathbf{r}') < c < U(\theta, \mathbf{0})$ for all $\theta'' \in [-\pi, \theta_B]$. Therefore,

$$r_{\mathcal{C}}(\theta, \theta', \mathbf{r}') \geq \frac{\lambda r_0}{4\pi^2} \int_{-\pi}^{\theta_B} d\theta'' = \frac{\lambda r_0}{4\pi^2}(\theta_B + \pi) > 0. \quad (12)$$

Also recall that $r(\theta)$ is bounded above as in (7).

We are now ready to prove the inequality (3). For any $\theta \in (\theta_1', \theta_2')$, where the interval (θ_1', θ_2') is chosen above,

$$P(S_{i+1} \in A | S_i = (\theta, \mathbf{0})) \geq P(S_{i+1} \in A \cap I | S_i = (\theta, \mathbf{0})) \geq \int_{A \cap I} \frac{r_{\mathcal{C}}(\theta, \theta', \mathbf{r}')}{r(\theta)} d\mu'(\theta', \mathbf{r}') > 0.$$

The last integral is strictly positive because $\mu'(A \cap I)$ is nonzero, $r_{\mathcal{C}}(\theta, \theta', \mathbf{r}')$ is bounded away from zero by (12), and $r(\theta)$ is bounded above by (7). □

Proof of Theorem 1: First we will establish the ψ-irreducibility and aperiodicity [6] of the chain $\{S_i\}$. In fact, we will show that, for any $n \geq 3$ and any state $s \in \mathcal{S}$, the measure $\psi(\cdot)$ is absolutely continuous with respect to the measure $P(S_{i+n} \in \cdot | S_i = s)$. To that end, choose and fix an arbitrary state $s \in \mathcal{S}$ and an arbitrary measurable subset A of $\mathcal{S}$ with $\psi(A) > 0$, so that either $(\mathcal{L}_1 \times \delta_{\mathbf{0}})(A) = \mu(A) > 0$ or $(\mathcal{L}_1 \times \mathcal{L}_2)(A) > 0$ (or both).

In the first case, Lemma 1 implies that $P(S_{i+2} \in A | S_i = s') > 0$ for any s', which, together with the Markov property, implies that $P(S_{i+n} \in A | S_i = s) > 0$ for all $n \geq 2$. In the second case, combining Lemma 1 with Lemma 2 applied to $A \cap \mathcal{S}_W$ and with the Markov property, we obtain that there are $\theta_1' < \theta_2'$ such that,

$$\begin{aligned} & P(S_{i+3} \in A | S_i = s) \\ & \geq \quad P(S_{i+3} \in A, S_{i+2} \in (\theta_1', \theta_2') \times \{\mathbf{0}\} | S_i = s) \\ & \geq \quad \epsilon \int_{\theta_1'}^{\theta_2'} P(S_{i+3} \in A | S_{i+2} = (\theta, \mathbf{0})) d\theta, \end{aligned}$$

where the positivity of the last integral follows again from Lemma 2. Finally, using the Markov property once again, we have that $P(S_{i+n} \in A | S_i = s) > 0$ for all $n \geq 3$, as required.

Now, ψ-irreducibility and aperiodicity, together with the Doeblin bound of Lemma 1, imply [6], [9], that the chain is uniformly ergodic. Specifically, Lemma 1 implies that the state space $\mathcal{S}$ is small, and that the drift condition (V4) of [6] holds with Lyapunov function $V \equiv 1$. Then [6, Theorem 15.0.1] implies that the chain $\{S_i\}$ has a unique invariant (probability) measure π to which the distribution of S_i converges uniformly, as stated in part 1) of the theorem. In particular, the chain $\{S_i\}$ is Harris recurrent, and [6, Theorem 17.0.1] implies that the strong law of large numbers holds for functions $F \in L_1(\pi)$, as stated in part 2) of the theorem. □

Proof of Corollary 1: Since $X_{W,i}$ and $C_i = C(X_{W,i}, Y_{W,i})$ are bounded, and hence π-integrable, functions of $S_i = (\Theta_i, (X_{W,i}, Y_{W_i}))$, the first two results immediately follow from Theorem 1. For the next two, let $\bar{\psi}$ denote the measure $\bar{\psi} = \mathcal{L}_1 \times \delta_{\mathbf{0}} \times [0, \infty) + \mathcal{L}_1 \times \mathcal{L}_2 \times \delta_0$ on $\bar{\mathcal{S}}$. Arguing as in the proof of Theorem 1, it is easy to show that the new chain $\{\bar{S}_i\}$ is $\bar{\psi}$-irreducible and aperiodic, and also uniformly ergodic. Once again, [6, Theorem 17.0.1] implies that the strong law of large numbers holds for $\{\bar{S}_i\}$, and recalling that $X_{B,i} = v_0 \Delta_i \cos \Theta_i$, the last two statements of the corollary will follow as soon as we establish that Δ is π-integrable. Indeed, since, given $\Theta = \theta$, Δ is exponential with rate $r(\theta) \geq r_0 > 0$, we have,

$$E_{\bar{\pi}}(\Delta) = E_{\pi}[E_{\bar{\pi}}(\Delta | \Theta)] = E_{\pi}\left[\frac{1}{r(\Theta)}\right] \leq \frac{1}{r_0} < \infty,$$

completing the proof. □

Acknowledgment

This work has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No. 645220 (Road-, Air- and Water-based Future Internet Experimentation - RAWFIE).

References

[1] R. Cavallari, S. Toumpis, and R. Verdone, "Analysis of hybrid geographic/delay-tolerant routing protocols for wireless mobile networks," in *Proc. IEEE INFOCOM*, Honolulu, HI, Apr. 2018.

[2] N. Bezirgiannidis, F. Tsapeli, S. Diamantopoulos, and V. Tsaoussidis, "Towards flexibility and accuracy in space dtn communications," in *Proceedings of the 8th ACM CHANTS*, 2013, pp. 43–48.

[3] P.-C. Cheng, K. C. Lee, M. Gerla, and J. Härri, "GeoDTN+ Nav: geographic DTN routing with navigator prediction for urban vehicular environments," *Mobile Networks and Applications*, vol. 15, no. 1, pp. 61–82, 2010.

[4] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: Social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, 2011.

[5] R. Cavallari, S. Toumpis, R. Verdone, and I. Kontoyiannis, "Analysis of geographic/delay-tolerant routing in mobile wireless networks," *Preprint, http://arxiv.org/abs/1801.02229*, 2018.

[6] S. Meyn and R. Tweedie, *Markov Chains and Stochastic Stability*, 2nd ed. London: Cambridge University Press, 2009, 1993 edition online: `http://probability.ca/MT/`.

[7] H. Royden, *Real Analysis*. New York: Macmillan, 1988.

[8] D. Williams, *Probability with martingales*. Cambridge University Press, 1991.

[9] I. Kontoyiannis and S. Meyn, "Spectral theory and limit theorems for geometrically ergodic Markov processes," *Ann. Appl. Probab.*, vol. 13, pp. 304–362, February 2003.