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: Draft software extensions for Tasks 1,4		
Confidentiality	CO	**Deliverable type**[1]	R
Deliverable File	D11_UNSURPASSED.pdf	**Date**	31/12/2017
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 Aliaj	AUEB	aliai16@aueb.gr
Georgia Dimaki	AUEB	ginadimaki135@gmail.com
Nikos Fotiou	AUEB	fotiou@aueb.gr
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	31/12/2017	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:
A) We describe Dedalus, a network monitoring platform under development for use with experiments.
B) We describe the software extensions under integration and development for use in Task 1 of UNSURPASSED (Ah Hoc Routing).
C) We describe the software extensions under integration and development for use in Task 4 of UNSURPASSED (Security).
D) We discuss other issues related to the progress of UNSURPASSED, such as meetings, publications, etc.
E) We include, as an appendix, an accepted version of a scientific publication.

Keywords: **UNSURPASSED**

Table of Contents

List of Figures

1. Network Monitoring Platform: Dedalus

Overview

For the efficient use of the experimental USV platform and its associated human resources, and in order to accelerate research, it is crucial that we are able to replicate, as accurately as possible, the envisioned experiments in a controlled environment, so that debugging of the software can take place and some lessons can be learned in the convenience of our lab. To that effect, we have been developing, testing, and using a network monitoring platform called Dedalus, for use in our lab network, and for possible actual use in the actual experimental network. In this section, we present an overview of Dedalus.

The platform permits the setup, execution, and monitoring of a wide variety of experiments related to UNSURPASSED. In more detail, it is an implementation of a self-healing distributed system for gathering real-time statistics from a wireless network and can be attached to that network and used in order to track the status of links, maintain and distribute event logs, as well as directly control the nodes of the network through attached clients, instructing them to create traffic destined for other nodes.

Dedalus can operate in networks of nodes communicating using one or more channels, some of them wired, and their associated network interfaces. When only one channel is available, then both the control traffic as well as the data traffic will go through it. When multiple channels are available, the Dedalus control traffic can go through a control network and the created data traffic through a test network. In our lab setting, the control network can be the WiFi network available in the building where our laboratory is located and the test network can be the wireless ad hoc network built using our various protocol implementations.

Dedalus is based on the majordomo protocol and is designed for resiliency and can operate in unstable networks. In particular, its various entities can be created in any order without any constraints imposed by connectivity. Also, if a connection is lost, message travelling to their destinations will persist until they are delivered or a timeout occurs.

Dedalus has been tested in a set of 5 Raspberry Pis and has already been used in order, for example, to perform modest-scale multihop routing experiments using the Babel ad hoc routing protocol within our laboratory. In the immediate future, we intend to use it to perform experiments involving 10-15 Raspberry Pis spread out throughout our building.

Dedalus Modes

As shown in Fig. 1, Dedalus creates a distributed system comprising of three kinds of entities, Clients, Workers, and Brokers, running at the nodes of the network. Workers create, receive, and forward traffic, and report their status to Brokers. Brokers assign tasks to workers according to instructions received from the Clients. Finally, Clients monitor the conditions of the network and assign tasks to Workers by communicating with the Brokers. The Dedalus application itself can be run in four main modes, one mode for each kind of entity and a mode combining two entities.

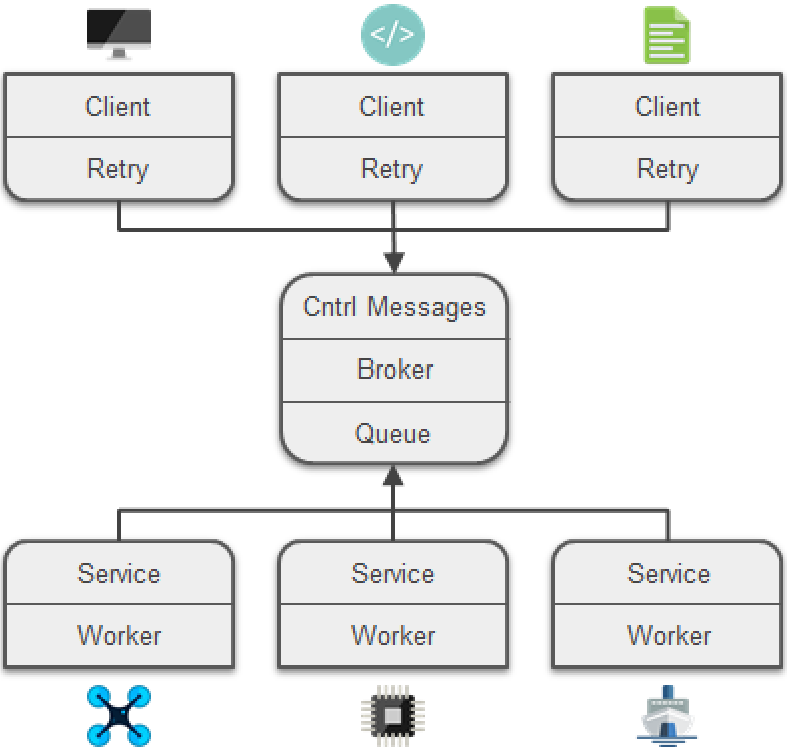


Figure 1 Dedalus architecture: Its three kinds of entities (Clients, Brokers, and Workers) and communication between them)

Client

Clients connect to a Broker instance and log or view information. They run as follows:

```
$ python dedalus client -a 127.0.0.1
```

Clients have only to respect the specific message formats and otherwise are not limited in their implementation details in any way. Therefore, they can be created in most available programming languages and from any operating system. We are also not limited to the design of the client since it can be anything from a desktop application with a complete user interface, to a mobile app, a terminal application or even a simple logger. That said, we have developed a GUI for Clients running on full-fledged computers equipped with a browser that allows human operators to have a clear understanding of the conditions in the network at any time instant. Snapshots of the GUI appear in Figs. 2, 3, and 4.

A typical message coming from a Client and accepted by a Dedalus Broker looks like the one below:

```
['', dedalus_protocol, dedalus_service_a, wid, command_id, args]
```

The arguments are:

- An empty space used by the Broker to distinguish the sender id from the rest of the message.

- The version of the protocol (describing the communication between entities) we want to use.
- The specific worker service we want to operate on.
- The worker id, if it is required.
- Finally, any specific command id and any command arguments, as required.

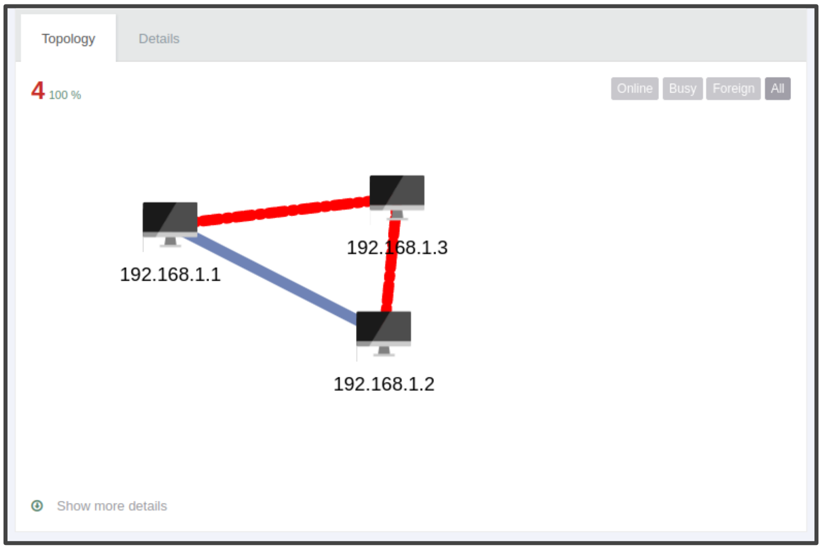


Figure 2 Dedalus Client GUI snapshot 1/3: The client has a readily available view of the connectivity in the network. High-rate links are colored blue, low-rate links are colored red.

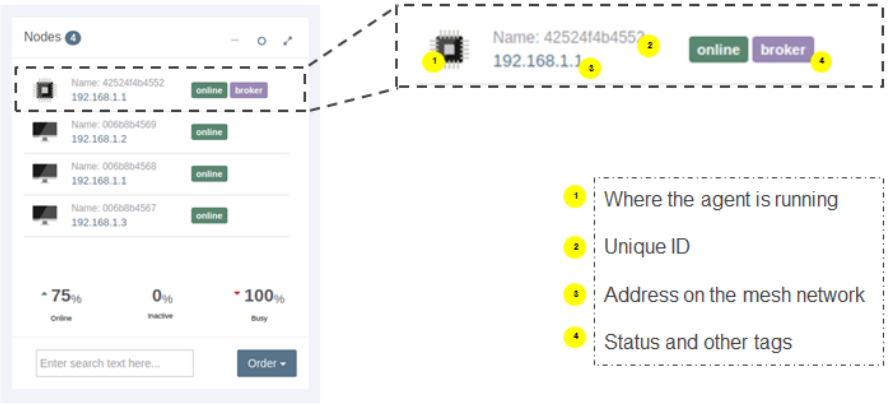
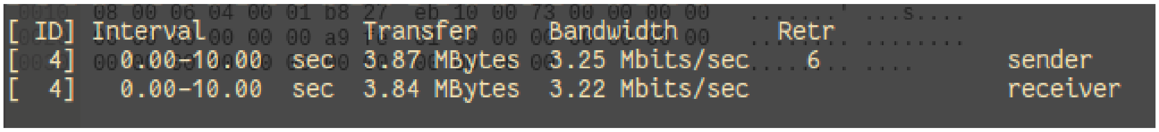


Figure 3 Dedalus Client GUI snapshot 2/3: The client has a readily available list of the various types of entities in the network with basic information on them.



```
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  3.87 MBytes  3.25 Mbits/sec    6             sender
[  4]   0.00-10.00  sec  3.84 MBytes  3.22 Mbits/sec                  receiver
```

Figure 4 Dedalus Client GUI snapshot 3/3: The client monitors link transmission rates.

Worker

Workers connect to a Broker and register themselves for a specific service, making themselves available to serve client requests. They run as follows:

```
$ python dedalus worker -a 127.0.0.1
```

Workers are responsible for performing all of the required operations on each node. As soon as they find a running instance of a Broker, they will connect to it and make available one or more services that they can offer. Once the initial installation phase has concluded, they will be available to answer requests and monitor the node they are running on.

Again, they can be implemented in any of the available languages as long as the rules for communication between different entities are followed.

Workers are exposed to a series of modules able to mainly do one of two things: 1) scan the current hardware and 2) report back usage and network statistics or protocol specific information.

Broker

Brokers accept messages from clients and workers. They run as follows:

```
$ python dedalus broker -a 127.0.0.1
```

In more detail, a Dedalus broker is responsible for passing messages back and forth from all clients and workers connected to him as well as implementing the needed queues for ensuring that all requests eventually will be served.

Brokers also are tasked with basic bookkeeping operations since they have to maintain lists of all active workers as well as test their connectivity and status.

Brokers also have their own service registered along with all other worker services and can operate as an asynchronous backend server.

Worker/Broker

We can run Dedalus so that a joint worker/broker entity is created.

```
$ python dedalus full -a 127.0.0.1 -s
```

This mode of operation is intended for testing and debugging or for when we need to quickly run both a worker and a broker at the same node. That said, we are not limited to only one worker per node, as each new Dedalus entity will be spawned in its own thread.

Building Dedalus

Building Dedalus requires Python and pip installed on the computer.

```
$ git clone git@gitlab.com:ealione/Dedalus.git
$ cd Dedalus

$ pip install -r requirements.txt

$ python setup install
```

2. Software Extensions for Task 1

As discussed in the UNSURPASSED proposal, the main aim of Task 1 is to configure and experiment with ad hoc routing protocols. We have already worked on two of them, Babel and B.A.T.M.A.N. These two protocols have many similarities, thus a comparison of their relative performance would be interesting, as it would highlight the effects of their specific differences on the performance of the network. We have already configured Babel on Raspberry Pis and conducted some experiments with a small number of nodes; work on B.A.T.M.A.N. is pending. In the following, we provide more details.

As specified in the Babel RFC[4], Babel is based on the Bellman-Ford algorithm with additional features aiming at avoiding the creation of loops. A Babel node periodically broadcasts Hello messages to all of its neighbors; when a node receives a Hello message, it replies with an IHU ("I Heard You") message. From the information derived from Hello and IHU messages received from its neighbor B, a node A computes the cost C(A,B) of the link from A to B. In this way, the Bellman-Ford algorithm is performed in every node that can send traffic. Because of its proactive operation, in large, stable networks Babel generates more traffic than protocols that only send updates when the network topology changes.

The B.A.T.M.A.N. (Better Approach To Mobile Ad hoc Networking) protocol originates from the OLSR protocol[5]. Its RFC[6] is in the draft stage, but the protocol is implemented, tested, optimized and included in the latest Linux kernels. In this protocol the in-degree of individual nodes determines the overall perception of the network. For its basic operation, it uses originator messages (OGM). Each OGM is a quadruple consisting of the original sender’s address, the address of the node re-broadcasting the OGM, a TTL (time to live) and a sequence number. B.A.T.M.A.N. nodes broadcast the OGMs periodically, and the recipients discard the OGM with lower sequences numbers that are now obsolete in the B.A.T.M.A.N. procedural technique. The higher sequence numbers of the OGM indicate its freshness with respect to its life-span and helps to determine both route discovery and neighbor selection. B.A.T.M.A.N. provides reduction of signaling or routing overheads, reliability, performance, improvements on throughput, less delay, and lower CPU load. The implementation used (B.A.T.M.A.N. advanced), which is the optimization of the basic B.A.T.M.A.N. protocol, works on the link layer.

For our experimentation with Babel, we have installed babeld, which is available in the Babeld git repository[7], and configured the wireless interface of each Raspberry so as to run in ad hoc

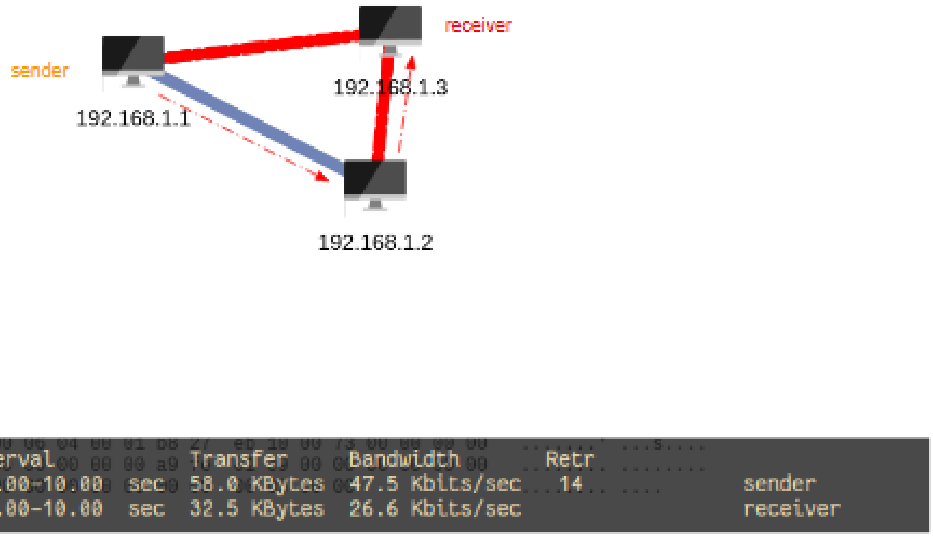
[4] RFC 6126, https://tools.ietf.org/html/rfc6126
[5] RFC 3626, https://tools.ietf.org/html/rfc3626
[6] https://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing-00
[7] https://github.com/jech/babeld

mode and have a static IP. Finally the babeld daemon is configured to automatically start after booting and keep a log of its operation in babaled.log file.

Dedalus, in order to retrieve BABEL specific information, like the current state of the network, makes use of the babeld configuration interface[8]. To this effect, babeld is invoked with a -g flag so it can accept TCP connections from localhost. What the protocol scanning module of Dedalus does is to request a dump of babeld information. Then, a TCP connection is established, the request is sent, and the reply is then parsed so as to provide us with all the information we need. Specifically, we keep track of routing information accompanied by metrics for each route as well as information about neighbor nodes (i.e., nodes in the range of the specific node's WiFi).

Regarding this protocol, we performed our first experiments with three of the Raspberry Pis on a linear topology. In this topology the central node had both other nodes in its range, but the others were not in direct contact with each other. We sent traffic from the one edge node to the other so that the traffic is routed through the central node. One of the tests of the experiment was to disconnect a node and count the time it takes to BABEL to find this out. The convergence was actually slow (> 1min sometimes). Some snapshots of the GUI output while the experiments were running are presented below, in. Figs. 5,6, and 7.

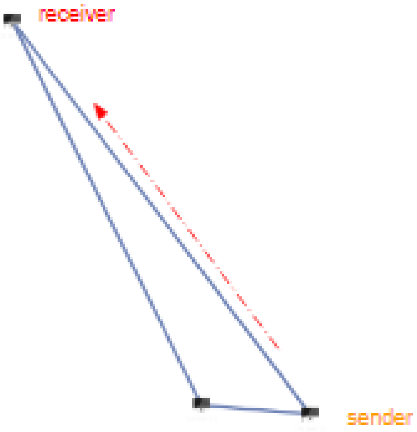
As for B.A.T.M.A.N., we are in the process of installing it on the Raspberries and we are looking forward to creating a scanning module for it in Dedalus. Experiments with this protocol are not conducted yet and will be reported in follow-up deliverables.

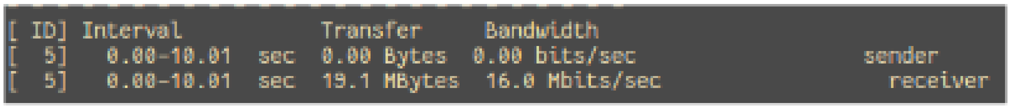


```
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  58.0 KBytes  47.5 Kbits/sec   14             sender
[  4]   0.00-10.00  sec  32.5 KBytes  26.6 Kbits/sec                  receiver
```

Figure 5 Experiment with poor links. Red lines indicate the poor links, blue lines the strong ones.

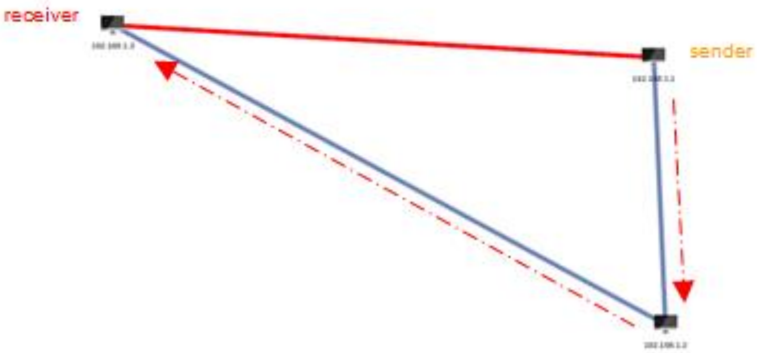
[8] https://www.irif.fr/~jch/software/babel/babeld.html

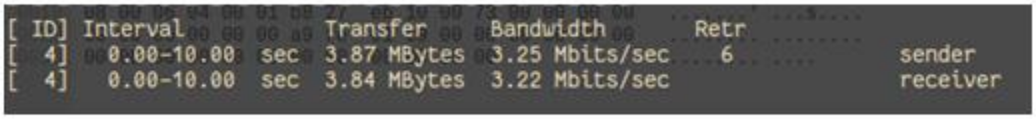




```
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-10.01  sec  0.00 Bytes  0.00 bits/sec                  sender
[  5]   0.00-10.01  sec  19.1 MBytes  16.0 Mbits/sec                  receiver
```

Figure 6 Experiment with well-connected nodes and a resulting high bandwidth





```
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  3.87 MBytes  3.25 Mbits/sec    6             sender
[  4]   0.00-10.00  sec  3.84 MBytes  3.22 Mbits/sec                  receiver
```

Figure 7 Experimenting with Babel in a multihop setting

3. Software Extensions of Task 4

As discuss in the proposal, the UNSURPASSED project will also experiment with identify-based proxy re-encryption, in the context of Task 4.

Identity-based encryption

An Identity Based Encryption (IBE) scheme is a public key encryption scheme in which an arbitrary string can be used as a public key. An IBE scheme is specified by the four algorithms, *Setup*, *Extract*, *Encrypt* and *Decrypt*, summarized as follows:

- Setup: it is executed once by a *Secret Key Generator* (SKG). It takes as input a security parameter and returns a *master-secret key* (SK_{SKG}) and some generator specific *public parameters* (PP). The SK_{SKG} is kept secret by the SKG, whereas the PP are made publicly available.
- Extract: it is executed by a SKG. It takes as input PP, SK_{SKG}, and an arbitrary string ID, and returns a secret key SK_{ID}.
- Encrypt: it is executed by users. It takes as input an arbitrary string ID, a message M, and PP, and returns a ciphertext C_{ID}.
- Decrypt: it is executed by users. It takes as input C_{ID} the corresponding private decryption key SK_{ID}, and returns the message M.

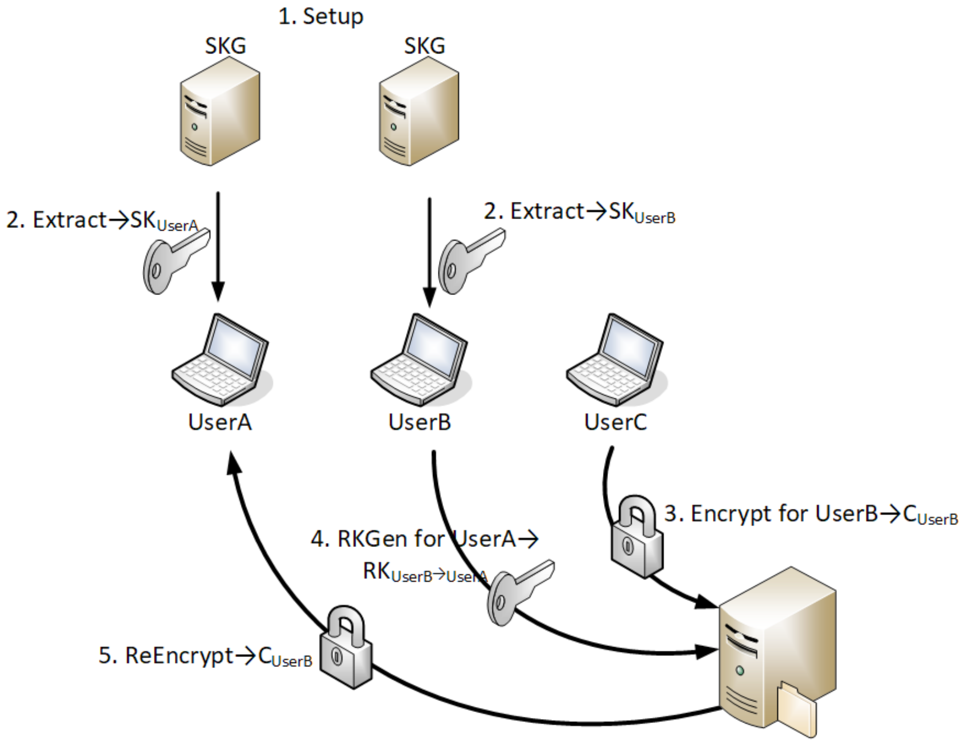


Figure 8 Identity-Based Proxy Re-Encryption (IB-PRE) example

Identity-based proxy re-encryption

An Identity-Based Proxy Re-Encryption (IB-PRE) scheme is a scheme in which a third party is allowed to alter a ciphertext, encrypted using an arbitrary string ID1, in a way that another user that owns a secret key SK_{ID2} can decrypt it. Such a scheme specifies two new algorithms, RKGen and Reencrypt, in addition to the IBE algorithms already discussed:

- RKGen: it is executed by a user that owns a secret key SK_{ID1}. It takes as input PP, the secret key SK_{ID1} and an arbitrary string ID2 and generates a (public) re-encryption key $RK_{ID1 \rightarrow ID2}$.
- ReEncrypt: it is executed by a third party. It takes as input PP, a re-encryption key $RK_{ID1 \rightarrow ID2}$ and a ciphertext C_{ID1} and outputs a new ciphertext C_{ID2}.

The ciphertext generated by the ReEncrypt algorithm can be decrypted only by using SK_{ID2}. The entity that performs the re-encryption learns nothing about the encrypted plaintext or about the secret keys of users ID1 and ID2.

Fig. 8 illustrates an IBE-PRE example. There are three users, namely UserA, UserB, and UserC. UserA and UserB have received the secret key that corresponds to their user identity from two different SKGs (step 2). UserC encrypts a message by executing the IBE Encrypt algorithm with the identity of UserA (that is, “UserA”) as a key and stores the generated ciphertext in a storage entity. Supposedly, UserB wishes to enable UserA to access the ciphertext generated by UserC: she generates an appropriate re-encryption key and sends it to the storage entity. The storage entity is now able to re-encrypt the ciphertext in a way that can be decrypted using the private key of UserA.

Implementation

We have developed a message encryption application using the Green and Ateniese IBE-PRE scheme[9] as implemented by the Charm-Crypto library.[10] The current implementation is composed of 6 python scripts, one for each IB-PRE algorithms. In particular, our implementation is composed of the following scripts

Setup.py: It creates the master secret key and the public parameters

Extract.py: It extracts the private key that corresponds to a user identity. It can be invoked as follows:

```
$ python ./Extract.py <user identity>,
```

where <user identity> is an arbitrary string.

[9] M. Green, G. Ateniese, "Identity-Based Proxy Re-Encryption," in Applied Cryptography and Network Security. Springer Berlin/Heidelberg, 2007

[10] https://github.com/JHUISI/charm

Encrypt.py: It encrypts a message, using a user identity as a key, and stores it in a file. It can be invoked as follows:

```
$ python Encrypt.py <user identity>  <text to encrypt>  <destination file>
```

Decrypt.py: It decrypts an encrypted file and outputs the plaintext message. It can be used as follows:

```
$ python Decrypt.py <user identity> <encrypted file>
```

Re-keygen.py: It creates a re-encryption key, It can be used as follows:

```
$ python Re-keygen.py <source identity> <target identity>
```

Re-encrypt.py: It uses a re-encryption key in order to re-encrypt an already encrypted file. It can be used as follows:

```
$python Re-encrypt.py <source identity> <target identity> <file>
```

4. Other issues

Meetings

1. We participated in the November 24th 2017 Open Call 2 RAWFIE meeting with a talk titled “UNSURPASSED: Unmanned Surface Vehicles as Primary Assets for the Coast Guard” given by Stavros Toumpis. We had extensive discussions with representatives of third-party recipients of Open Call 2 grants.
2. Based on these discussions, an opportunity has arisen for close collaboration with the Democritus team of Drs. Kimon Kontovasilis and Ioannis Manolopoulos working on the ATLAS project, which has also received funding by the RAWFIE Open Call 2. The team visited us on December 12th 2017 and Dr. Manolopoulos gave a talk titled “UxV-bAsed OpporTunistic Networks FaciLitating Connectivity in Remote AreaS: ATLAS”. We are considering the possibility of integrating in our platform the implementation of the DTN routing protocol that will be developed in the ATLAS project. More meetings with the Democritus team will be planned soon and other opportunities for collaboration will be explored.
3. We had a skype meeting with Konstantinos Houmos and Athanasios Korakis of the University of Thessaly, who have extensive experimentation with Raspberry Pis in mobile environments. A talk by Konstantinos Houmos has been tentatively scheduled for February 2018.
4. Weekly meetings are being regularly held between all members of the team.

Personnel

In addition to the team mentioned in the proposal, M.Sc. student Esmerald Aliaj and senior B.Sc. students Georgia Dimaki, Antonios Georgilakis, and Petros Getsopoulos are providing crucial support, as part of their studies in the Department of Informatics. The hiring of more personnel is pending.

Equipment procurement

As discussed, in anticipation of our experiments in the RAWFIE platform, it is crucial that we are able to replicate experiments in a controlled environment, so that debugging of the software can take place in the convenience of our lab. To that effect, as already discussed, experiments with a set of 5 Raspberry Pis have already been performed. In the immediate future, we plan to procure an additional 10 Raspberry Pis (model: Raspberry Pi 3 Model B) along with an assortment of WiFi network adapters, microSD cards etc., in order to perform richer experiments.

Publications

The first publication that includes work motivated by and partly done in the context of UNSURPASSED will be presented in Infocom 2018. The work is titled “Analysis of Hybrid Geographic/Delay-Tolerant Routing Protocols for Wireless Mobile Networks” and is co-authored by S. Toumpis and R. Cavallari and R. Verdone, of the University of Bologna. As we have not performed large scale experiments yet, the work contains only theoretical work on the performance of DTN protocols envisioned for use in USV environments.

Gitlab

We are using Gitlab as a private repository of our work. We can provide RAWFIE access to it upon request. We stress that our work is still under development.

Analysis of Hybrid Geographic/Delay-Tolerant Routing Protocols for Wireless Mobile Networks

Riccardo Cavallari, Stavros Toumpis, and Roberto Verdone

***Abstract*—We study a mobile wireless delay-tolerant network in which nodes travel on an infinite plane along trajectories comprised of linear segments; a single packet is traveling towards a destination located at an infinite distance, employing both wireless transmissions and sojourns on node buffers. In this setting, we calculate analytically the long-term averages of *(i)* the speed with which the packet travels towards its destination and *(ii)* the rate with which the wireless transmission cost is being accumulated with distance traveled. Due to the problem's complexity, we use two intuitive approximations; simulations show that the resulting errors are modest.**

***Index Terms*—Delay-Tolerant Network, Geographic Routing, Mobile Wireless Network, Stochastic Geometry.**

I. Introduction

In this work, we analyze a mobile wireless delay-tolerant network (DTN) model consisting of an infinite number of nodes moving on an infinite plane. Each node moves with a constant velocity magnitude along straight lines, picking, uniformly, new random travel directions at exponential times. Nodes move independently of each other. A single packet must travel to a destination that is located at an infinite distance towards a given direction, opportunistically using two modes of transport, i.e., wireless transmissions and physical transports on the buffers of nodes. Wireless transmissions are instantaneous, but come at a transmission cost that is an arbitrary function of the vector connecting the transmitter and receiver (and not simply the distance covered, therefore in general the cost depends on the transmission's direction). Physical transports, on the other hand, do not have an associated cost, but introduce packet delivery delay. The packet alternates between the two modes of transport using a routing rule belonging to a broad family parametrized by two quantities, i.e., a forwarding region and a potential function. In this setting, we calculate the long-term averages of the speed with which the packet travels to the destination and the rate with which it accumulates transmission cost with distance, using two judicious and intuitive approximations to make the analysis tractable; as shown by simulations, the introduced errors are modest.

The analysis of wireless networks using stochastic geometry tools has recently attracted significant research interest [1], [2], with the large majority of works focusing on networks with *stationary* nodes. In the rest of this section, we review those recent works that have studied networks with *mobile* nodes that are most pertinent to our work.

The authors of [3] considered a network of finite area where mobile nodes move along straight lines, changing their travel direction according to a Poisson process, each node creating packets for an immobile destination whose location it knows. In this setting, nodes employ a geographic routing protocol, called *Constrained Relative Bearing*, under which each packet remains on the buffer of a node provided that node is traveling along a sufficiently good direction towards its destination, otherwise the packet is transmitted to a more suitable node, provided one is available nearby. The scheme is shown to achieve a near-constant per-node throughput with a bounded delivery delay, as the number of nodes increases. With respect to that work, which pursues a network-wide analysis, here we take a more 'local' view, focusing on how a given packet accumulates cost and delay as it travels towards its destination, and avoiding the calculation of metrics up to a multiplicative constant; also, our routing model is much more general.

The authors of [4] have studied a setting related to our own, using the same mobility model and performance metrics. However, with respect to that work, our analysis does not discretize the directions of node travel, can handle anisotropic transmission costs, uses more refined approximations that lead to more accurate results, and, again, handles a much more general routing model; all these differences significantly complicate the analysis.

Many works have specifically focused on one-dimensional mobile wireless DTN models, where node locations are constrained to be on a straight line or torus [5], [6], [7], [8]. Our setting is two-dimensional, hence more challenging.

Recently, the topic of percolation in mobile wireless DTNs has attracted interest. With respect to the classic percolation problem in immobile wireless networks, in mobile wireless networks packets percolate through combinations of both wireless transmissions and physical transports. For example, in [9] the authors study the speed with which information percolates in networks of two or more dimensions for a wide range of mobility models. A related setting, where, notably, only a fixed number of copies of the packet are allowed to propagate, is studied in [10].

The authors of [11] investigate a setting in which a mobile node is moving along a straight line on a plane where stationary base stations (BSs) are placed according to a uniform Poisson point process (PPP); the node is in contact with a BS if the two are closer than a threshold distance. In this setting, the authors study the QoS experienced by the node receiving video through the BSs and the distribution of download times for files being downloaded to the node through the BSs.

Finally, in [12] the authors study a network comprising nodes moving according to independent Brownian motions in $\mathbb{R}^d$; two nodes are in contact whenever they are within some threshold distance from each other. In this setting, the authors calculate three important metrics: the time until a target (mobile or immobile) comes in contact with any of the nodes, the time until the nodes come in contact with all points in a given region, and the time until a target comes in contact with an infinite connected component of the network.

The rest of this work is organized as follows. In Section II we define our network model, and in Section III we define our performance metrics. The analysis is spread over Sections IV, V, and VI. In Section VII we present numerical results and we conclude in Section VIII.

II. Network Model

At time $t = 0$, infinite nodes are placed on the infinite plane, $\mathbb{R}^2$, according to a uniform PPP with **(node) density** $\lambda > 0$. Subsequently, each node travels on $\mathbb{R}^2$, independently of the rest of the nodes, as follows (in the rest of this work directions of travel are specified in terms of the angle they form with the positive x-axis): the node selects a random travel direction $D_1 \in [-\pi, \pi)$, chosen uniformly; the node moves in this direction along a straight line with constant **node speed** (i.e., velocity magnitude) $v_0 > 0$, for a random duration of time E_1 that follows the exponential distribution with mean $1/r_0$, where $r_0 > 0$ is the **(node) turning rate**; the node then picks another random travel direction $D_2 \in [-\pi, \pi)$, also chosen uniformly, and travels in that direction for another random duration E_2, which also follows the exponential distribution with the same mean $1/r_0$, and so on. The random variables (RVs) D_i and E_i, for $i = 1, \ldots,$ are independent of each other. Note that the time instants when the travel direction of a given node changes form a uniform Poisson process with rate r_0. Also note that, according to the Displacement Theorem [1, Theorem 1.10], at any time instant $t > 0$ the locations of all nodes form a uniform PPP with density λ.

Nodes are equipped with transceivers with which they can exchange packets. Transmitting a packet at location $\mathbf{r}$ (with respect to the transmitter) leads to a **(transmission) cost** $C(\mathbf{r})$, which models both the energy cost of transmitting and the need to share the bandwidth with other packets. All packet transmissions are assumed to be instantaneous (in this work, we are interested in quantifying delays that are much larger than transmission times, so this assumption is reasonable).

A tagged packet, created at $t = 0$, must travel to a destination assumed to be at an infinite distance. This assumption is made for mathematical convenience; we expect our results to be applicable in the design of real networks provided packets travel for finite but not small distances. With no loss of generality, we take the destination to be in the direction of the positive x-axis. For this reason, we define a node travel direction θ_1 to be **better** than a travel direction θ_2 if $|\theta_1| < |\theta_2|$. We use the terms **equal**, **best**, **worse**, and **worst**, for travel directions, in related senses, mutatis mutandis.

The packet must travel to the destination using a hybrid geographic/delay-tolerant **routing rule** (RR) that uses combinations of wireless transmissions (this is the geographic part of the RR) and sojourns along the buffers of nodes (this is the delay-tolerant part of the RR). The RR belongs to the family of RRs specified as follows.

First, we associate with each node A a closed, bounded, and convex subset of $\mathbb{R}^2$ called the **Forwarding Region (FR)** $\mathcal{F}(A)$ that contains A in its interior and is moving together with it. $\mathcal{F}(A)$ is described in the polar coordinate system centered at node A as

$$\mathcal{F}(A) = \{\mathbf{r} \triangleq (r, \phi) : -\pi \le \phi < \pi,\ 0 \le r \le b(\phi)\},$$

where the **boundary function** $b(\phi)$ is an arbitrary bounded function of $\phi \in [-\pi, \pi)$. Also, for each location $\mathbf{r} \in \mathcal{F}(A)$, including the location $\mathbf{0}$ of A, and each travel direction $\theta \in [-\pi, \pi)$, we define a continuous **potential** $U(\theta, \mathbf{r})$ which describes how suitable a node located at $\mathbf{r}$ and traveling with direction θ is for keeping the packet; the higher the potential, the more suitable the node is. Different choices of the two functions $b(\phi)$ and $U(\theta, \mathbf{r})$ give rise to different RRs.

Second, we assume that the packet travels on the buffer of a **carrier** A_i until another node A_{i+1}, which we refer to as the **eligible** node, is found that lies either in the interior or the boundary of $\mathcal{F}(A_i)$ and has a potential higher than that of A_i as well as all other nodes within $\mathcal{F}(A_i)$ (all potentials are calculated using the coordinate system centered at A_i); the packet is then immediately transmitted to A_{i+1}. When the packet arrives at A_{i+1}, the rule is applied anew; therefore another node, A_{i+2}, might be immediately eligible (now using the coordinate system of A_{i+1}), in which case the packet is immediately transmitted again at the same instant, otherwise a sojourn on the buffer of node A_{i+1} is initiated, and so on.

Third, $b(\phi)$ and $U(\theta, \mathbf{r})$ satisfy the following:

Assumption 1: No two nodes in the same FR have the same potential. (This assumption is not particularly restrictive, as any RR can be made to satisfy it by very slightly modifying the potential function to resolve such conflicts.)

Assumption 2: If a node A that has no eligible node in its FR changes its travel direction to a better one, then no nodes currently in its FR becomes eligible. (Indeed, there is no reason for a packet to decide that another node is eligible just because its carrier just improved its travel direction.)

Assumption 3: If node A gives the packet to an eligible node B, then all nodes in the intersection $\mathcal{F}(A) \cap \mathcal{F}(B)$ are not eligible to receive the packet after B. (In other words, nodes A and B need to agree that node B is indeed the best choice among all the candidate nodes they have in common.)

III. Performance Metrics

We break the travel of the tagged packet in an infinite sequence of **stages**, indexed with $i = 1, 2, \ldots$. Each stage i comprises the i-th period of time (called **sojourn**) during which the packet travels with a specific travel direction $\Theta_i \in [-\pi, \pi)$ on the buffer of a node A_i as well as all instantaneous transmissions that may follow.

We classify stages in four types, according to how they end:

Type $\mathcal{A}$: A_i changed its travel direction from Θ_i to another direction Θ_i' and no eligible node was immediately available.

Type $\mathcal{B}$: A_i changed its travel direction from Θ_i to another travel direction Θ_i' and an eligible node was immediately available in the interior of $\mathcal{F}(A_i)$.

Type $\mathcal{C}$: A_i did not change its travel direction, but an eligible node became available in the interior of $\mathcal{F}(A_i)$.

Type $\mathcal{D}$: A_i did not change travel direction, but an eligible node became available on the boundary of $\mathcal{F}(A_i)$.

Note that when the carrier picks a new direction, the probability it will equal the previous one is zero, so this case is not considered. Also, as the nodes are Poisson distributed at any time and move independently, the probability that there are *any* nodes on the boundary of $\mathcal{F}(A_i)$ when A_i changes direction is also zero, so this case is also not considered.

In addition to Θ_i, we describe each stage $i = 1, 2, \dots$ using the following RVs: Let T_{i-1} and T_i be the time instants when stage i starts and finishes, $\Delta_i = T_i - T_{i-1}$ its duration, and $N_i \in \mathbb{N}$ the number of transmissions taking place at its end. For the j-th of these transmissions, where $j = 1, \dots, N_i$, let $X_{W,i,j}$ and $Y_{W,i,j}$ be the changes that the transmission induces in the coordinates of the packet along the x-axis and y-axis respectively and $C_{i,j} = C\left((X_{W,i,j}, Y_{W,i,j})\right)$ be the wireless transmission cost. Also, let $X_{W,i} = \sum_{j=1}^{N_i} X_{W,i,j}$ and $C_i = \sum_{j=1}^{N_i} C_{i,j}$. Let $X_{B,i}$ be the change in the x-coordinate of the packet due to its physical transport in the buffer of its carrier during stage i, noting that $X_{B,i} = v_0 \Delta_i \cos\Theta_i$. Finally, let $X_i = X_{B,i} + X_{W,i}$. In the following, we will refer to any change of the x-coordinate of the packet as **progress**.

We describe the performance of the RR in terms of the **(packet) speed** V_p and the **(normalized packet) cost** C_p. The packet speed V_p is defined as

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

Therefore, V_p represents the limit of the average rate (in units of distance over units of time) with which the packet makes progress towards its destination, as the number of stages over which the average rate is taken goes to infinity. The normalized packet cost C_p is defined likewise as

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

Closing the section, we introduce our first approximation.

Second Order Approximation:

1) When a node A receives the packet from a node B, the complete mobility process is reinitialized, except that the position and travel direction of node A are maintained and all nodes that appear (after the reinitialization) in the intersection $\mathcal{F}(A) \cap \mathcal{F}(B)$ and whose potential is greater than the potential $U(\theta, \mathbf{0})$ of A are removed.

2) When a carrier A changes its travel direction from θ to a better *one, θ', the complete mobility process is reinitialized, except that the position and travel direction of node A are maintained and all nodes in $\mathcal{F}(A)$ whose potential is greater than the* new *potential $U(\theta', \mathbf{0})$ of A are removed.*

3) When a carrier A changes its travel direction from θ to a worse *one, θ', the complete mobility process is reinitialized, except that the position and travel direction of node A are maintained and all nodes in $\mathcal{F}(A)$ whose potential is greater than the* old *potential $U(\theta, \mathbf{0})$ of A are removed.*

To motivate the approximation, observe that, once we adopt it, each stage i is completely described by the angle Θ_i and no memory of past stages is preserved. Indeed, at the start of each stage i, and irrespective of how the previous one ended, the mobility process has just restarted excepting the two facts that *(i)* the packet is traveling at the buffer of a node A whose travel direction is Θ_i and *(ii)* $\mathcal{F}(A)$ is devoid of nodes with potential greater than $U(\Theta_i, \mathbf{0})$. Therefore, the approximation makes the discrete-time process $\{\Theta_i,\ i = 1, \dots\}$ Markovian; simple examples exist (omitted due to space constraints) that show that, without it, the process is not Markovian.

We call this approximation 'Second Order' to differentiate it from a simpler approximation, used in [4], under which whenever a node A receives the packet or changes its travel direction the mobility process is reinitialized, keeping node A's position and travel direction, but without removing any nodes, so that new eligible nodes might become available just because of the reinitialization; our approximation ensures that this cannot happen, hence, as simulations verify, it reduces errors notably; however, it significantly complicates the analysis.

IV. Transmissions Analysis

The setting of this section, shown in Fig. 1, is as follows: a node A is traveling with direction $\theta \in [-\pi, \pi)$ and has just received the packet from node B such that A is located in position $\mathbf{r} \in \mathcal{F}(B)$ according to the coordinate system with origin at B. The quantities we calculate in this section will be functions of θ and $\mathbf{r}$. Let $\mathcal{G}(\mathbf{r}) \triangleq \mathcal{F}(A) \cap (\mathcal{F}(B))^c$.

As integrals of the form $\int_{-\pi}^{\pi} \iint_{\mathcal{F}(A)} H(\mathbf{r}, \theta)\, dA d\theta$ will be appearing frequently in what follows, we will be denoting them, with a small abuse of notation, as $\int_{\mathcal{M}} H(\mathbf{r}, \theta)\, dA d\theta$, with $\mathcal{M} \triangleq [-\pi, \pi] \times \mathcal{F}(A)$.

First, let $E(N; \theta, \mathbf{r})$ be the expected number of nodes N in $\mathcal{G}(\mathbf{r})$ whose potential exceeds the potential $U(\theta, \mathbf{0})$ of A:

$$E(N; \theta, \mathbf{r}) = \int_{\mathcal{M}} \frac{\lambda}{2\pi} \mathbf{1}[U(\theta', \mathbf{r}') > U(\theta, \mathbf{0}), \\ \mathbf{r}' \in \mathcal{G}(\mathbf{r})]\, dA' d\theta', \quad (3)$$

where the expression $\mathbf{1}[\mathcal{X}]$ is equal to 1 if the condition $\mathcal{X}$ holds and 0 if it does not.

Next, let $P_E(\theta, \mathbf{r})$ be the probability of the event $\mathcal{E}(\theta, \mathbf{r})$ that $\mathcal{G}(\mathbf{r})$ does not contain an eligible node, and therefore a new stage will commence at the moment node A receives the packet. This event will occur if there are no nodes in $\mathcal{G}(r)$ whose potential is greater than the potential of A. The number N of such nodes follows the Poisson distribution with expected value $E(N; \theta, \mathbf{r})$ given by (3), therefore

$$P_E(\theta, \mathbf{r}) = P(\mathcal{E}(\theta, \mathbf{r})) = \exp\left[-E(N; \theta, \mathbf{r})\right]. \quad (4)$$

Next, let $g(\theta', \mathbf{r}'; \theta, \mathbf{r}) d\theta' dA'$, where $\theta' \in [-\pi, \pi)$, $\mathbf{r}' \in \mathcal{F}(A)$, be the incremental probability that the packet is transmitted, according to the RR used, from node A to an eligible node C lying within a subset of $\mathcal{F}(A)$ of incremental area dA' centered at the point $\mathbf{r}'$, and the travel direction of node C is within the incremental interval $[\theta', \theta' + d\theta']$ (see Fig. 1). We will specify $g(\theta', \mathbf{r}'; \theta, \mathbf{r})$ for all $\mathbf{r}, \mathbf{r}' \in \mathcal{F}(A)$ and all $\theta, \theta' \in [-\pi, \pi)$. To this effect, observe that $g(\theta', \mathbf{r}'; \theta, \mathbf{r}) = 0$ if $U(\theta, \mathbf{0}) \geq U(\theta', \mathbf{r}')$, i.e., node A is at least as suitable as node C for keeping the packet, or if $\mathbf{r}' \notin \mathcal{G}(\mathbf{r})$, i.e., $\mathbf{r}'$ is in the intersection of the FRs $\mathcal{F}(A)$ and $\mathcal{F}(B)$ and so no eligible node may be found there by the Second Order Approximation. If both $U(\theta, \mathbf{0}) < U(\theta', \mathbf{r}')$ and $\mathbf{r}' \in \mathcal{G}(\mathbf{r})$, note that the incremental probability that there will be a node C with travel direction θ' at the location $\mathbf{r}'$ is $\frac{\lambda}{2\pi} d\theta' dA'$. That node will receive the packet if there is no node in $\mathcal{G}(\mathbf{r})$ that is actually better than C. The expected number of such nodes is (cf. with the derivation of (3))

$$\int_{\mathcal{M}} \frac{\lambda}{2\pi} \mathbf{1}[U(\theta'', \mathbf{r}'') > U(\theta', \mathbf{r}'), \mathbf{r}'' \in \mathcal{G}(\mathbf{r})]\, dA'' d\theta'',$$

and their number is Poisson distributed. Combining all cases, we have

$$g(\theta', \mathbf{r}'; \theta, \mathbf{r}) = \frac{\lambda}{2\pi} \mathbf{1}[U(\theta, \mathbf{0}) < U(\theta', \mathbf{r}'),\ \mathbf{r}' \in \mathcal{G}(\mathbf{r})] \times$$
$$\exp\big[-\frac{\lambda}{2\pi} \int_{\mathcal{M}} \mathbf{1}[U(\theta'', \mathbf{r}'') > U(\theta', \mathbf{r}'), \mathbf{r}'' \in \mathcal{G}(\mathbf{r})]\, dA'' d\theta''\big].$$

Staying with the setting of Fig. 1, let X_M be the aggregate progress that the packet will make, due to possibly *multiple* wireless transmissions, from node A onwards, until no eligible node is found and a new stage commences. Let $E(X_M; \theta, \mathbf{r})$ be the expectation of X_M, given the values of θ and $\mathbf{r}$. Note that the packet will either stay at A, in which case $E(X_M; \theta, \mathbf{r}) = 0$, or else it will be transmitted to a new node C, whose location, in the coordinate system centered at A, is $\mathbf{r}' \in \mathcal{F}(A)$ and with a travel direction $\theta' \in [-\pi, \pi)$. In this case, the conditional expectation $E(X_M; \theta, \mathbf{r}, \theta', \mathbf{r}')$ of X_M is

$$E(X_M; \theta, \mathbf{r}, \theta', \mathbf{r}') = r' \cos\phi' + E(X_M; \theta', \mathbf{r}'), \quad (5)$$

where $\mathbf{r}' \triangleq (r', \phi')$ in polar coordinates. Observe that the expected value appearing on the right hand side is $E(X_M; \theta', \mathbf{r}')$ due to the Second Order Approximation: once the packet arrives at C, the memory of its having passed through B is lost, and only θ' and $\mathbf{r}'$ matter in determining the remaining expected progress. Integrating (5) over all possible incremental events, we arrive at the following equation:

$$E(X_M; \theta, \mathbf{r}) = \int_{\mathcal{M}} g(\theta', \mathbf{r}'; \theta, \mathbf{r}) \times \left[r' \cos\phi' + E(X_M; \theta', \mathbf{r}')\right] dA' d\theta', \quad (6)$$

i.e., an inhomogeneous, multidimensional Fredholm integral equation of the second kind. In producing the numerical results of Section VII, we solve it for $E(X_M; \theta, \mathbf{r})$ by discretizing both θ and $\mathbf{r}$ and approximating the integral by a summation, thus converting the equation to a large linear system.

Next, let C_M be the aggregate transmission cost that will be incurred, due to possibly multiple wireless transmissions, from node A onwards, until no eligible node is found and a new stage commences. Let $E(C_M; \theta, \mathbf{r})$ be the expectation of C_M, given the values of θ and $\mathbf{r}$. Working as in the case of $E(X_M; \theta, \mathbf{r})$, we arrive at

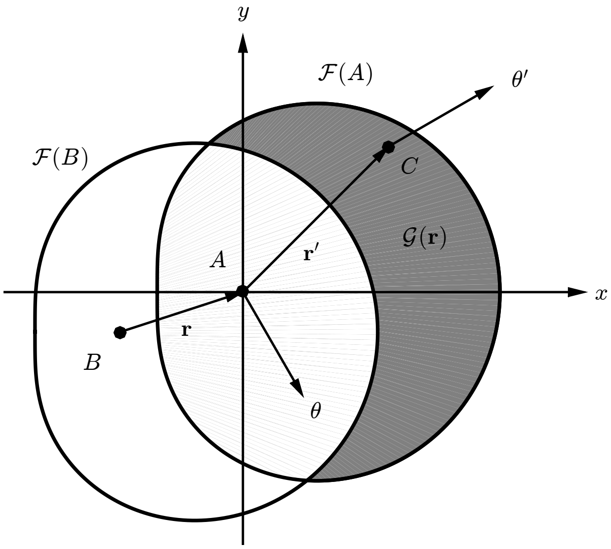


Fig. 1. The setting used in Section IV. Here, there is an eligible node C within $\mathcal{G}(\mathbf{r})$, however this is not always the case.

$$E(C_M; \theta, \mathbf{r}) = \int_{\mathcal{M}} g(\theta', \mathbf{r}'; \theta, \mathbf{r}) \times \left[C(\mathbf{r}') + E(C_M; \theta', \mathbf{r}')\right] dA' d\theta'.$$

This equation has the structure of (6) and, in producing the numerical results, we solve it in the same manner.

Finally, let $h(\chi; \theta, \mathbf{r}) d\chi$, where $\chi \in [-\pi, \pi)$, be the probability of the incremental event $\mathcal{H}(\chi; \theta, \mathbf{r})$ that a packet that was transmitted from node B to node A will eventually (and possibly after more than one hop) start a new stage at a node *other than* A traveling with direction in the range $[\chi, \chi + d\chi]$. $\mathcal{H}(\chi; \theta, \mathbf{r})$ can occur in one of two ways:

1) The packet is transmitted from node A to a node C located in an infinitesimal region of area dA' centered at $\mathbf{r}' \in \mathcal{F}(A)$ and with travel direction in the interval $[\chi, \chi + d\chi]$, which happens with probability $g(\chi, \mathbf{r}'; \theta, \mathbf{r}) dA' d\chi$, and that node finds no eligible node, which happens with probability $P_E(\chi, \mathbf{r}')$.

2) The packet arrives at a node C located in an infinitesimal region of area dA' centered at $\mathbf{r}' \in \mathcal{F}(A)$ with travel direction in the interval $[\theta', \theta' + d\theta']$, where $\theta' \in [-\pi, \pi)$, which happens with probability $g(\theta', \mathbf{r}'; \theta, \mathbf{r}) dA' d\theta'$, and that node finds another eligible node such that eventually the packet arrives at a node with travel direction in the interval $[\chi, \chi + d\chi]$, which happens with probability $h(\chi; \theta', \mathbf{r}') d\chi$. Integrating over all possible incremental events,

$$h(\chi; \theta, \mathbf{r}) = \int_{\mathcal{M}} \Big[\frac{1}{2\pi} g(\chi, \mathbf{r}'; \theta, \mathbf{r}) P_E(\chi, \mathbf{r}') + g(\theta', \mathbf{r}'; \theta, \mathbf{r}) h(\chi; \theta', \mathbf{r}')\Big]\, dA'\, d\theta'. \quad (7)$$

This equation is similar to (6), with the main difference being that there is now an extra parameter, i.e., χ. In producing numerical results, we first discretize χ, and then solve (7) once for each discrete value of χ, using the method applied in (6).

V. Sojourn Analysis

The setting of this section is as follows: at time $t = 0$ the new stage i starts with the packet in the buffer of a node A traveling in the direction $\Theta_i = \theta \in [-\pi, \pi)$; as a new stage has started, the FR $\mathcal{F}(A)$ does not contain an eligible node. The sojourn ends at a random time $\Delta_i > 0$.

We adopt the following approach: we partition the event that the sojourn of stage i ends into four families of disjoint incremental events; each family corresponds to one of the four types of stages defined in Section III. We then use our second approximation, introduced in Section V-C, to calculate the probabilities that each of these incremental events will occur.

A. Four families of incremental events

First, we define the family of incremental events

$$\mathcal{A}(\theta) \triangleq \{\mathcal{A}(\theta, \theta'),\ \theta' \in [-\pi, \pi)\},$$

such that $\mathcal{A}(\theta, \theta')$ is the event that the sojourn ends because, at time Δ_i, node A changes its travel direction from $\Theta_i = \theta$ to another direction in the incremental range $[\theta', \theta' + d\theta']$ and no eligible node is found.

Second, we define the family of incremental events

$$\mathcal{B}(\theta) \triangleq \{\mathcal{B}(\theta, \theta', \mathbf{r}'),\ \theta' \in [-\pi, \pi),\ \mathbf{r}' \in \mathcal{F}(A)\},$$

such that $\mathcal{B}(\theta, \theta', \mathbf{r}')$ is the event that the sojourn ends because, at time Δ_i, node A changes its travel direction from $\Theta_i = \theta$ to some other direction, and an eligible node is immediately found inside $\mathcal{F}(A)$, traveling with direction in the incremental range $[\theta', \theta' + d\theta']$ and located in an incremental region of incremental area dA' centered at $\mathbf{r}'$.

Third, we define the family of incremental events

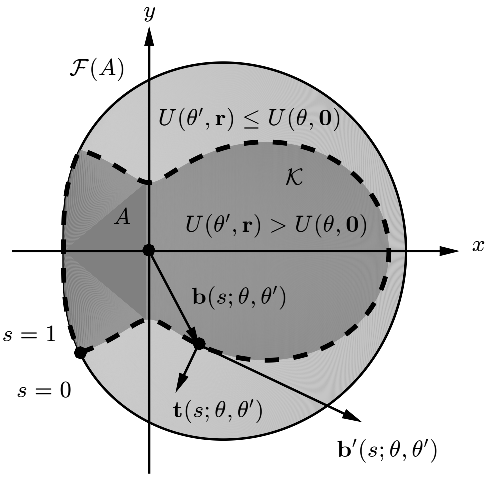
$$\mathcal{C}(\theta) \triangleq \{\mathcal{C}(\theta, \theta', \mathbf{r}'),\ \theta' \in [-\pi, \pi),\ \mathbf{r}' \in \mathcal{F}(A)\},$$

such that $\mathcal{C}(\theta, \theta', \mathbf{r}')$ is the event that the sojourn ends because, at time Δ_i, an eligible node appears, because it changed its direction to a new direction in the incremental range $[\theta', \theta' + d\theta']$, located in a region of incremental area dA' centered at $\mathbf{r}'$ inside $\mathcal{F}(A)$.

Fourth, we define the family of events that corresponds to stages of Type $\mathcal{D}$. First, we need to specify a family of curves: for given values of $\theta, \theta' \in [-\pi, \pi)$, let the **threshold curve**

$$\mathbf{b}(s; \theta, \theta'),\ s \in [0, 1],$$

be the curve separating the part $\mathcal{K}$ of the FR $\mathcal{F}(A)$ of node A (whose travel direction is θ) where $U(\theta', \mathbf{r}) > U(\theta, \mathbf{0})$ with the rest of $\mathbb{R}^2$, i.e., $\mathcal{K}^c$. See Fig. 2. Observe that nodes that hit this curve coming from outside (i.e., $\mathcal{K}^c$) become eligible. Also, let $\mathbf{t}(s; \theta, \theta')$ be a unit vector that is perpendicular to the curve $\mathbf{b}(s; \theta, \theta')$ at the location specified by the parameter s and pointing to the direction of lower potential, and let $\mathbf{b}'(s; \theta, \theta')$ be the derivative of $\mathbf{b}(s; \theta, \theta')$ with respect to the parameter s. Observe that changing the first argument of $\mathbf{b}(s; \theta, \theta')$ from s to $s + ds$ traces an infinitesimal line segment of length $ds|\mathbf{b}'|(s; \theta, \theta')$ that is perpendicular to $\mathbf{t}(s; \theta, \theta')$. We can now define the family of incremental events



y
$\mathcal{F}(A)$
$U(\theta', \mathbf{r}) \le U(\theta, \mathbf{0})$
$\mathcal{K}$
A
$U(\theta', \mathbf{r}) > U(\theta, \mathbf{0})$
x
$\mathbf{b}(s; \theta, \theta')$
$s = 1$
$s = 0$
$\mathbf{t}(s; \theta, \theta')$
$\mathbf{b}'(s; \theta, \theta')$

Fig. 2. Setting used in defining the family $\mathcal{D}(\theta)$.

$$\mathcal{D}(\theta) \triangleq \{\mathcal{D}(\theta, \theta', s),\ \theta' \in [-\pi, \pi),\ s \in [0, 1]\},$$

such that $\mathcal{D}(\theta, \theta', s)$ is the event that the sojourn ends because, at time Δ_i, an eligible node appears, traveling with direction $[\theta', \theta' + d\theta']$, at the incremental stretch of the boundary $[s, s + ds]$ of the curve $\mathbf{b}(s; \theta, \theta')$.

Finally, we define the union of all families

$$\mathcal{U}(\theta) = \mathcal{A}(\theta) \cup \mathcal{B}(\theta) \cup \mathcal{C}(\theta) \cup \mathcal{D}(\theta).$$

Observe that $P\left(\cup_{E \in \mathcal{U}(\theta)} E | \Theta_i = \theta\right) = 1$, i.e., with probability 1 an incremental event will eventually occur that will end the sojourn of stage i. Indeed, changes of direction of the carrier always end a stage, and these occur in exponential times with fixed rate r_0.

B. Incremental and aggregate rates

We define the **incremental rates** $r_{\mathcal{A}}(\theta, \theta')$, $r_{\mathcal{B}}(\theta, \theta', \mathbf{r}')$, $r_{\mathcal{C}}(\theta, \theta', \mathbf{r}')$, and $r_{\mathcal{D}}(\theta, \theta', s)$, to be such that the following equalities hold:

$$r_{\mathcal{A}}(\theta, \theta') d\theta' dt \triangleq P\big[\mathcal{A}(\theta, \theta'),\\ 0 \le \Delta_i \le dt | \Theta_i = \theta\big], \quad \theta,\ \theta' \in [-\pi, \pi),$$

$$r_{\mathcal{B}}(\theta, \theta', \mathbf{r}') d\theta' dA' dt \triangleq P\big[\mathcal{B}(\theta, \theta', \mathbf{r}'),\\ 0 \le \Delta_i \le dt | \Theta_i = \theta\big], \quad \theta,\ \theta' \in [-\pi, \pi),\ \mathbf{r}' \in \mathcal{F}(A),$$

$$r_{\mathcal{C}}(\theta, \theta', \mathbf{r}') d\theta' dA' dt \triangleq P\big[\mathcal{C}(\theta, \theta', \mathbf{r}'),\\ 0 \le \Delta_i \le dt | \Theta_i = \theta\big], \quad \theta,\ \theta' \in [-\pi, \pi),\ \mathbf{r}' \in \mathcal{F}(A),$$

$$\begin{aligned} &r_{\mathcal{D}}(\theta,\theta',s)d\theta' ds dt \triangleq P\big[\mathcal{D}(\theta,\theta',s),\\ &\quad 0 \le \Delta_i \le dt|\Theta_i = \theta\big], \quad \theta,\ \theta' \in [-\pi,\pi),\ s \in [0,1]. \end{aligned}$$

The incremental rates provide the probability that the corresponding incremental events will occur within the incremental time interval $[0, dt]$, right after the sojourn has started and the mobility model has been reinitialized.

Regarding $r_{\mathcal{A}}(\theta,\theta')$, note that the incremental probability $P(\mathcal{A}(\theta,\theta'),\ 0 \le \Delta_i \le dt|\Theta_i = \theta)$ is equal to the probability that node A will change its direction during the interval $[0, dt]$, which is $r_0 dt$, multiplied with the probability that it will pick a direction in the range $[\theta', \theta' + d\theta]$, which is $\frac{d\theta'}{2\pi}$, and finally multiplied with the probability that there are no eligible nodes within $\mathcal{F}(A)$ with a potential at most $U(\theta,\mathbf{0})$ but greater than $U(\theta',\mathbf{0})$, which is (cf. with the derivation of (4))

$$\exp\left[-\int_{\mathcal{M}} \frac{\lambda}{2\pi}\mathbf{1}[U(\theta,\mathbf{0}) \ge U(\theta'',\mathbf{r}'') > U(\theta',\mathbf{0})]\, dA'' d\theta''\right],$$

It follows that

$$\begin{aligned} r_{\mathcal{A}}(\theta,\theta') = \frac{r_0}{2\pi}\exp\Big[-\int_{\mathcal{M}} \frac{\lambda}{2\pi} \\ \times \mathbf{1}[U(\theta,\mathbf{0}) \ge U(\theta'',\mathbf{r}'') > U(\theta',\mathbf{0})]\, dA'' d\theta''\Big]. \end{aligned}$$

Regarding $r_{\mathcal{B}}(\theta,\theta',\mathbf{r}')$, note that if $U(\theta',\mathbf{r}') > U(\theta,\mathbf{0})$ then the incremental probability $P(\mathcal{B}(\theta,\theta',\mathbf{r}'),\ 0 \le \Delta_i \le dt|\Theta_i = \theta)$ is zero, because the condition implies that there was an eligible node before A changed direction. If $U(\theta,\mathbf{0}) \ge U(\theta',\mathbf{r}')$, the incremental probability equals the incremental probability $r_0 dt$ that node A will change its travel direction during the interval $[0, dt]$, multiplied with the probability

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \mathbf{1}\left[U(\theta',\mathbf{r}') > U(\theta'',\mathbf{0})\right]\, d\theta''$$

that its new direction θ'' will lead to a lower potential that $U(\theta',\mathbf{r}')$ (otherwise the packet would have stayed with A), the incremental probability $\frac{\lambda}{2\pi} dA' d\theta'$ that there is a node at the specified location $\mathbf{r}'$ with the specified travel direction θ', and finally with the probability that there is no node in $\mathcal{F}(A)$ that is better than that node, which is (cf. with (4))

$$\begin{aligned} K_1(\theta,\theta',\mathbf{r}') \triangleq \exp\Big[-\int_{\mathcal{M}} \frac{\lambda}{2\pi} \\ \times \mathbf{1}[U(\theta,\mathbf{0}) \ge U(\theta''',\mathbf{r}''') > U(\theta',\mathbf{r}')]\, dA''' d\theta'''\Big]. \end{aligned}$$

It follows that

$$\begin{aligned} r_{\mathcal{B}}(\theta,\theta',\mathbf{r}') = \frac{r_0\lambda}{4\pi^2}\mathbf{1}[U(\theta,\mathbf{0}) \ge U(\theta',\mathbf{r}')] \\ \times \int_{-\pi}^{\pi} \mathbf{1}\left[U(\theta',\mathbf{r}') > U(\theta'',\mathbf{0})\right]\, d\theta'' \times K_1(\theta,\theta',\mathbf{r}'). \end{aligned}$$

Regarding $r_{\mathcal{C}}(\theta,\theta',\mathbf{r}')$, observe that the incremental probability $P(\mathcal{C}(\theta,\theta',\mathbf{r}'),\ 0 \le \Delta_i \le dt|\Theta_i = \theta)$ is zero if $U(\theta',\mathbf{r}') \le U(\theta,\mathbf{0})$. Otherwise, it is equal to the probability that there is a node within the specified incremental area, which is equal to $\frac{\lambda dA'}{2\pi} K_2(\theta,\theta',\mathbf{r}')$ where

$$K_2(\theta,\theta',\mathbf{r}') \triangleq \int_{-\pi}^{\pi} \mathbf{1}[U(\theta'',\mathbf{r}') < U(\theta,\mathbf{0})]\, d\theta'',$$

multiplied with the probability that that node will turn in the specified range of directions $[\theta', \theta' + d\theta]$, which is $\frac{r_0}{2\pi} d\theta' dt$. Therefore,

$$r_{\mathcal{C}}(\theta,\theta',\mathbf{r}') = \frac{\lambda r_0}{4\pi^2}\mathbf{1}\left[U(\theta',\mathbf{r}') > U(\theta,\mathbf{0})\right] K_2(\theta,\theta',\mathbf{r}').$$

Regarding rate $r_{\mathcal{D}}(\theta,\theta',s)$, observe that nodes that move with travel direction θ' appear to node A to be moving with a different relative direction, due to its own movement, i.e., with relative speed[1] $v_0 e^{j\theta'} - v_0 e^{j\theta}$. Also observe that the inner product $(v_0 e^{j\theta'} - v_0 e^{j\theta}) \cdot \mathbf{t}(s;\theta,\theta')$ must be negative in order for the nodes with travel direction θ' to be hitting the boundary from outside $\mathcal{K}$. If this condition holds, the incremental probability $P(\mathcal{D}(\theta,\theta',s),\ 0 \le \Delta_i \le dt|\Theta_i = \theta)$ is equal to the density of nodes $\frac{\lambda d\theta'}{2\pi}$ traveling with direction in the specified interval $[\theta', \theta' + d\theta']$ multiplied with the area of an incremental parallelogram formed by the vectors $-dt(v_0 e^{j\theta'} - v_0 e^{j\theta})$ and $\mathbf{b}'(s;\theta,\theta')ds$. Indeed, the nodes exactly inside this parallelogram traveling with directions in the interval $[\theta', \theta' + d\theta]$ will cross the boundary during the time period $[0, dt]$. Therefore,

$$\begin{aligned} &P(\mathcal{D}(\theta,\theta',s),\ 0 \le \Delta_i \le dt|\Theta_i = \theta) \\ &\quad = \mathbf{1}[(v_0 e^{j\theta'} - v_0 e^{j\theta}) \cdot \mathbf{t}(s;\theta,\theta') < 0]\frac{\lambda d\theta'}{2\pi} \\ &\qquad \times |v_0 e^{j\theta'} - v_0 e^{j\theta}| dt |\mathbf{b}'(s;\theta,\theta')| ds \sin\chi. \end{aligned}$$

where $\chi \in [0,\pi]$ is the angle between the two incremental vectors. By straightforward geometry,

$$(e^{j\theta'} - e^{j\theta}) \cdot (-\mathbf{t}(s;\theta,\theta')) = |e^{j\theta'} - e^{j\theta}| \sin\chi,$$

therefore it follows that, for all $\theta,\ \theta' \in [-\pi,\pi)$ and $s \in [0,1]$,

$$r_{\mathcal{D}}(\theta,\theta',s) = \frac{\lambda v_0}{2\pi}\max\{(e^{j\theta} - e^{j\theta'}) \cdot \mathbf{t}(s;\theta,\theta'), 0\}|\mathbf{b}'(s;\theta,\theta')|.$$

Finally, we also define the following **aggregate rates**:

$$\begin{aligned} r_{\mathcal{A}}(\theta) &\triangleq \int_{-\pi}^{\pi} r_{\mathcal{A}}(\theta,\theta')\, d\theta', \\ r_{\mathcal{B}}(\theta) &\triangleq \int_{\mathcal{M}} r_{\mathcal{B}}(\theta,\theta',\mathbf{r}')\, dA' d\theta', \\ r_{\mathcal{C}}(\theta) &\triangleq \int_{\mathcal{M}} r_{\mathcal{C}}(\theta,\theta',\mathbf{r}')\, dA' d\theta', \\ r_{\mathcal{D}}(\theta) &\triangleq \int_{-\pi}^{\pi}\int_{0}^{1} r_{\mathcal{D}}(\theta,\theta',s)\, ds d\theta', \\ r(\theta) &\triangleq r_{\mathcal{A}}(\theta) + r_{\mathcal{B}}(\theta) + r_{\mathcal{C}}(\theta) + r_{\mathcal{D}}(\theta). \end{aligned}$$

Observe that we must have $r_0 = r_{\mathcal{A}}(\theta) + r_{\mathcal{B}}(\theta)$, as the union of the events belonging to the families $\mathcal{A}(\theta)$ and $\mathcal{B}(\theta)$ is the event that node A changes his direction, which happens with rate r_0. Therefore, we have $r(\theta) = r_0 + r_{\mathcal{C}}(\theta) + r_{\mathcal{D}}(\theta)$.

[1] With a slight abuse of notation, in the following we will denote any vector (x, y) also as the complex number $x + jy$.

C. Time Invariance Approximation and consequences

The incremental events we defined in Section V-A do not have to be independent of the time Δ_i the sojourn ends. This significantly complicates the calculations required for computing the probability that a specific incremental event occurs. We therefore adopt the following approximation:

Time Invariance Approximation: *For each incremental event $E \in \mathcal{U}(\theta)$,*

$$\begin{aligned}P(E, t \leq \Delta_i \leq t+dt | \Delta_i \geq t, \Theta_i = \theta)\\ = P(E,\ 0 \leq \Delta_i \leq dt | \Theta_i = \theta).\end{aligned}$$

Intuitively, under this approximation the rates with which incremental events are occurring are fixed at their values at the instant the sojourn starts and the mobility process has been reinitialized. Therefore, Δ_i must be exponentially distributed; by breaking down the (certain) event that the state ends to all incremental events $E \in \mathcal{U}(\theta)$ and applying this approximation, it is straightforward to show that $E(\Delta_i|\Theta_i = \theta) = [r(\theta)]^{-1}$.

Furthermore, by conditioning on the time at which the stage ends, it is straightforward to show that the probability that an incremental event occurs is a function only of its rate, i.e.:

$$\begin{aligned}P(\mathcal{A}(\theta, \theta')|\Theta_i = \theta) &= \frac{r_{\mathcal{A}}(\theta, \theta')d\theta'}{r(\theta)},\\ P(\mathcal{B}(\theta, \theta', \mathbf{r}')|\Theta_i = \theta) &= \frac{r_{\mathcal{B}}(\theta, \theta', \mathbf{r}')d\theta' dA'}{r(\theta)}, \quad (8)\\ P(\mathcal{C}(\theta, \theta', \mathbf{r}')|\Theta_i = \theta) &= \frac{r_{\mathcal{C}}(\theta, \theta', \mathbf{r}')d\theta' dA'}{r(\theta)}, \quad (9)\\ P(\mathcal{D}(\theta, \theta', s)|\Theta_i = \theta) &= \frac{r_{\mathcal{D}}(\theta, \theta', s)d\theta' ds}{r(\theta)}. \quad (10)\end{aligned}$$

VI. Performance Metrics Expressions

As discussed in Section III, due to the Second Order Approximation, each stage i is completely described by the travel direction Θ_i of the packet during its sojourn. Therefore, $E(\Delta_i|\Theta_i = \theta)$, $E(X_{B,i}|\Theta_i = \theta)$, $E(X_{W,i}|\Theta_i = \theta)$, $E(X_i|\Theta_i = \theta)$, and $E(C_i|\Theta_i = \theta)$ do not depend on i. Thus, it suffices to denote them with $E(\Delta|\theta)$, $E(X_B|\theta)$, $E(X_W|\theta)$, $E(X|\theta)$, and $E(C|\theta)$ respectively. Here, we will calculate the last four of these conditional expectations; the first one, $E(\Delta|\theta)$ has already been calculated in Section V-C:

$$E(\Delta|\theta) = \frac{1}{r(\theta)}.$$

First, note that, having $E(\Delta|\theta)$, the expected progress of the packet due to physical transport is simply

$$E(X_B|\theta) = E(v_0 \Delta \cos\Theta|\theta) = \frac{v_0 \cos\theta}{r(\theta)}.$$

Next, we calculate $E(X_W|\theta)$. Note that we will have wireless transmissions in the case the stage is of Types $\mathcal{B}$, $\mathcal{C}$, or $\mathcal{D}$. In all cases, the progress due to wireless transmissions comprises two parts: the first transmission, and all other transmissions after that. Regarding the expected progress due to all other transmissions after the first one, it is given as the solution of (6). Conditioning on the incremental event with which the stage ends, and using the respective probabilities (8)-(10), we arrive at

$$\begin{aligned}&E(X_W|\theta)\\ &= \frac{1}{r(\theta)} \times \Big[\int_{\mathcal{M}} r_{\mathcal{B}}(\theta, \theta', \mathbf{r}')[x' + E(X_M; \theta', \mathbf{r}')]\, dA' d\theta'\\ &\qquad + \int_{\mathcal{M}} r_{\mathcal{C}}(\theta, \theta', \mathbf{r}')[x' + E(X_M; \theta', \mathbf{r}')]\, dA' d\theta'\\ &\qquad + \int_{-\pi}^{\pi} \int_0^1 r_{\mathcal{D}}(\theta, \theta', s)\\ &\qquad \times [b_x(s; \theta, \theta') + E(X_M; \theta', \mathbf{b}(s; \theta, \theta'))]\, ds d\theta' \Big].\end{aligned}$$

In the above expression, we used the notations $\mathbf{r}' = (x', y')$, and $\mathbf{b}(s; \theta, \theta') = (b_x(s; \theta, \theta'), b_y(s; \theta, \theta'))$. Repeating the same steps, mutatis mutandis, we arrive at similar expression for $E(C|\theta)$ (omitted, due to space constraints).

Finally, noting that $X_i = X_{B,i} + X_{W,i}$, we simply have

$$E(X|\theta) = E(X_B|\theta) + E(X_W|\theta).$$

A. $q(\theta_2|\theta_1)$ and $\Psi(\theta)$

Next, we calculate the **transition rate function** $q(\theta_2|\theta_1)$, $\theta_1, \theta_2 \in [-\pi, \pi)$, defined so that

$$q(\theta_2|\theta_1)d\theta_2 \triangleq P\left[\theta_2 \leq \Theta_{i+1} \leq \theta_2 + d\theta_2 | \Theta_i = \theta_1\right]. \quad (11)$$

To this end, we consider all incremental events that comprise the event $\{\theta_2 \leq \Theta_{i+1} \leq \theta_2 + d\theta_2\}$, given that $\Theta_i = \theta_1$:

Case 1: The carrier changed its direction from θ_1 to a direction in the interval $[\theta_2, \theta_2 + d\theta_2]$ and no eligible node was found (event $\mathcal{A}(\theta_1, \theta_2)$).

Case 2: The carrier changed its direction from θ_1 to some other direction and an eligible node was found located in a region of area dA' centered at $\mathbf{r}'$, traveling with direction in the range $[\theta', \theta' + d\theta']$ (event $\mathcal{B}(\theta_1, \theta', \mathbf{r}')$), and eventually the packet started a new sojourn at another node traveling with direction in the interval $[\theta_2, \theta_2 + d\theta_2]$ (event $\mathcal{H}(\theta_2; \theta', \mathbf{r}')$).

Case 3: The carrier changed its direction from θ_1 to some other direction and an eligible node was found located in a region of area dA' centered at $\mathbf{r}'$, traveling with direction in the range $[\theta_2, \theta_2 + d\theta_2]$ (event $\mathcal{B}(\theta_1, \theta_2, \mathbf{r}')$), and no other eligible node was immediately found (event $\mathcal{E}(\theta_2, \mathbf{r}')$), thus starting a new stage with direction in the interval $[\theta_2, \theta_2 + d\theta_2]$.

Case 4: An eligible node was found located in a region of area dA' centered at $\mathbf{r}'$ inside $\mathcal{F}(A_i)$, traveling with direction in the range $[\theta', \theta' + d\theta']$ (event $\mathcal{C}(\theta_1, \theta', \mathbf{r}')$), and eventually the packet started a new sojourn at another node traveling with direction in the interval $[\theta_2, \theta_2 + d\theta_2]$ (event $\mathcal{H}(\theta_2; \theta', \mathbf{r}')$).

Case 5: An eligible node was found located in a region of area dA' centered at $\mathbf{r}'$ inside $\mathcal{F}(A_i)$, traveling with direction in the range $[\theta_2, \theta_2 + d\theta_2]$ (event $\mathcal{C}(\theta_1, \theta_2, \mathbf{r}')$), and no other eligible node was immediately found (event $\mathcal{E}(\theta_2, \mathbf{r}')$), thus starting a new stage with direction in the interval $[\theta_2, \theta_2 + d\theta_2]$.

Case 6: An eligible node was found at the range of locations $[s, s+ds]$ of the boundary traveling with direction in the interval $[\theta', \theta' + d\theta']$ (event $\mathcal{D}(\theta_1, \theta', s)$) and eventually the packet started a new sojourn at another node traveling with direction in the interval $[\theta_2, \theta_2 + d\theta_2]$ (event $\mathcal{H}(\theta_2; \theta', \mathbf{b}(s, \theta_1, \theta'))$).

Case 7: An eligible node was found at the range of locations $[s, s + ds]$ of the boundary traveling with direction θ_2 (event $\mathcal{D}(\theta_1, \theta_2, s)$) and no eligible node was immediately found (event $\mathcal{E}(\theta_2, \mathbf{b}(s, \theta_1, \theta_2))$), thus starting a new stage with direction in the interval $[\theta_2, \theta_2 + d\theta_2]$.

The probability on the right hand side of (11) can be calculated by adding the probabilities belonging to each of the seven incremental, distinct events, arriving at

$$
\begin{aligned}
q(\theta_2|\theta_1) = \frac{1}{r(\theta_1)} \Bigg(& r_{\mathcal{A}}(\theta_1, \theta_2) \\
& + \int_{\mathcal{M}} \left[r_{\mathcal{B}}(\theta_1, \theta', \mathbf{r}') + r_{\mathcal{C}}(\theta_1, \theta', \mathbf{r}') \right] h(\theta_2; \theta', \mathbf{r}') dA' d\theta' \\
& + \iint_{\mathcal{F}(A)} \left[r_{\mathcal{B}}(\theta_1, \theta_2, \mathbf{r}') + r_{\mathcal{C}}(\theta_1, \theta_2, \mathbf{r}') \right] P_E(\theta_2, \mathbf{r}') dA' \\
& + \int_{-\pi}^{\pi} \int_0^1 r_{\mathcal{D}}(\theta_1, \theta', s) h(\theta_2; \theta', \mathbf{b}(s, \theta_1, \theta'))\, ds d\theta' \\
& + \int_0^1 r_{\mathcal{D}}(\theta_1, \theta_2, s) P_E(\theta_2, \mathbf{b}(s, \theta_1, \theta_2))\, ds \Bigg), \\
& \qquad \theta_1, \theta_2 \in [-\pi, \pi).
\end{aligned}
$$

Finally, let the **(discrete-time) stationary distribution** $\Psi(\theta)$ be such that the incremental percentage of stages for which the angle Θ_i lies within the interval $[\theta, \theta + d\theta]$ is $\Psi(\theta) d\theta$. As, in the long run, the times the Markov chain Θ_i exits the interval $[\theta, \theta + d\theta]$ should be equal to the times it enters this interval, we have the standard formula

$$\Psi(\theta) = \int_{-\pi}^{\pi} q(\theta|\theta') \Psi(\theta')\, d\theta'.$$

Using this, together with the requirement $\int_{-\pi}^{\pi} \Psi(\theta)\, d\theta = 1$, it is straightforward to calculate $\Psi(\theta)$.

B. V_p and C_p

Finally, we provide expressions for the performance metrics defined in Section III. We use an intuitively clear heuristic derivation instead of a formal proof.

Regarding the speed V_p, starting from (1) we write

$$V_p = \frac{\lim\limits_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} X_i}{\lim\limits_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \Delta_i}. \tag{12}$$

With some abuse of notation, we write

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} X_i = \int_{-\pi}^{\pi} \lim_{n\to\infty} \frac{n(\theta)}{n} \cdot \lim_{n\to\infty} \frac{\sum_{i=1}^{n(\theta)} X_i}{n(\theta)}.$$

In this equality, we have broken down the sum of n terms in groups of terms, each group of $n(\theta)$ terms belonging to stages for which the direction belongs to the interval $[\theta, \theta + d\theta]$. As $n \to \infty$, the first fraction inside the integral tends to $\Psi(\theta) d\theta$, whereas the second fraction tends, by the Strong Law of Large Numbers, to the expectation $E(X|\theta)$ of X_i given that $\Theta_i \in [\theta, \theta + d\theta]$. Therefore,

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} X_i = \int_{-\pi}^{\pi} \Psi(\theta) E(X|\theta)\, d\theta.$$

Observe that the above expression is a nested expectation expression for the expected aggregate progress per stage:

$$E(X) \triangleq \int_{-\pi}^{\pi} \Psi(\theta) E(X|\theta)\, d\theta.$$

Likewise, we define

$$E(C) \triangleq \int_{-\pi}^{\pi} \Psi(\theta) E(C|\theta)\, d\theta, \quad E(\Delta) \triangleq \int_{-\pi}^{\pi} \Psi(\theta) E(\Delta|\theta)\, d\theta.$$

Manipulating the denominator of (12), as well as the numerator and denominator of (2), in a similar manner, we arrive at:

$$V_p = \frac{E(X)}{E(\Delta)}, \quad C_p = \frac{E(C)}{E(X)}.$$

VII. Numerical Results

Regarding the setting we use for our numerical study, we consider the following potential function:

$$U(\theta, \mathbf{r}) = -|\theta|.$$

In other words, the packet constantly tries to find nodes with a good travel direction, irrespective of their location relatively to the current holder, provided, of course, they are within the FR. Also, the FR is specified by the boundary function

$$b(\phi) = \frac{a(1 - \epsilon^2)}{1 - \epsilon \cos \phi},$$

where $a > 0$ and $\epsilon \in (0, 1)$. Therefore, the FR is an ellipse whose major axis has length $2a$ and is located along the x-axis, its left focus is located at the origin, and its eccentricity is ϵ. Regarding the transmission cost, we set $C(\mathbf{r}) = |\mathbf{r}|^2$.

In Fig. 3 we study the effects of the shape of the FR on V_p and C_p, by changing the values of the half-axis length a and eccentricity ϵ. We set $\lambda = 1$, $v_0 = 1$ and $r_0 = 1$. Solid black lines denote analytical results (obtained by evaluating numerically the derived formulas), and dotted red lines denote simulation results (obtained by simulations, in MATLAB, of the network model, without resorting to the approximations); observe that there is a good match between the two, justifying the use of our two approximations.

Regarding the effects of the half-axis length a, observe that, as a increases, both packet speed and packet cost increase; this exemplifies a fundamental tradeoff between these two metrics; the increase in the speed is due to the fact that, as a gets larger, firstly, it becomes more probable for a node with a good travel direction to be available at the moment the carrier changes its travel direction from a good direction to a bad one and, secondly, that node is farther ahead, on the average. For the last reason, and also due to the fact that the transmission cost function is quadratic, C_p also increases as a increases. In fact, both the speed and the cost diverge to infinity (rather that approach a finite limit) as a increases.

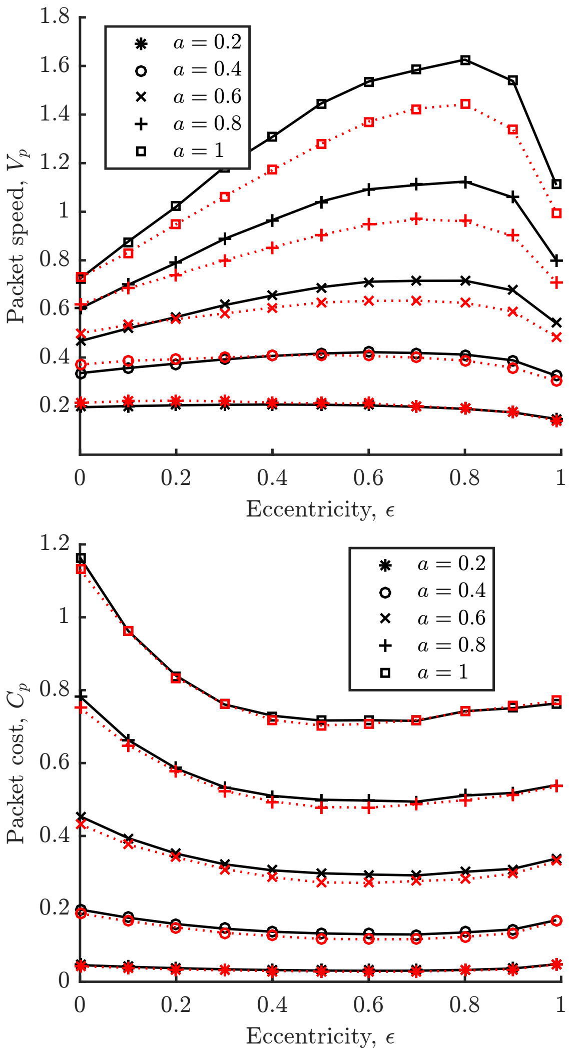


Fig. 3. V_p and C_p versus a and ϵ.

Regarding the effects of the FR eccentricity ϵ, starting from $\epsilon = 0$ and increasing it initially leads to higher speed and lower cost. This is expected, as the value $\epsilon = 0$ corresponds to a circular FR, therefore nodes whose relative position is towards the positive x-axis are not preferred at all with respect to other nodes, when scanning for eligible nodes; this inefficiency is rectified as ϵ initially increases. However, increasing ϵ past the value $\epsilon \simeq 0.6$ actually leads to an increase in the cost. Indeed, if the FR is *too* elliptical, it often happens that the packet is transmitted to nodes that are too far away from the current carrier, albeit with an excellent relative position, although there were other nodes that were much closer to the carrier with a relative position almost as good; as the cost is quadratic, this inadvertently increases the packet cost. Large values of the eccentricity also hurt the speed; this is because when ϵ increases, the area of the FR is reduced, and the packet spends more time traveling towards relatively bad directions on the buffers of nodes.

We note that we also studied the case when a, ϵ and v_0 are fixed, and the parameters λ and r_0 vary. The discrepancies between simulation and analysis are comparable to those of Fig. 3 but results are omitted due to space limitations.

VIII. Conclusions

Our work highlights and quantifies a fundamental tradeoff that exists in mobile wireless DTNs and has been observed elsewhere, e.g., in [3]: the more the packets rely on wireless transmissions, the larger their speed (hence the smaller their delivery delay) but the larger the transmission cost (which implicitly captures the bandwidth needed per packet transmission and, hence, the achievable end-to-end throughput).

Acknowledgement

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) through the National and Kapodistrian University of Athens. Administration support was provided by the Research Centre of the Athens University of Economics and Business.

Travel funding was received from AUEB and RC-AUEB under grant -2818-01.

References

[1] F. Baccelli and B. Blaszczyszyn, *Stochastic Geometry and Wireless Networks, Vols I-II*. Foundations and Trends in Networking, 2009.

[2] M. Haenggi, *Stochastic Geometry for Wireless Networks*. Cambridge University Press, 2012.

[3] P. Jacquet, S. Malik, B. Mans, and A. Silva, "On the throughput-delay tradeoff in georouting networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3230–3242, June 2016.

[4] R. Cavallari, R. Verdone, and S. Toumpis, "Cost/speed analysis of mobile wireless DTNs under random waypoint mobility," in *Proc. Wiopt*, Tempe, AZ, May 2016.

[5] E. Baccelli, P. Jacquet, B. Mans, and G. Rodolakis, "Highway vehicular delay tolerant networks: Information propagation speed properties," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1743–1756, March 2012.

[6] M. Zarei, A. M. Rahmani, and H. Samini, "Connectivity analysis for dynamic movement of vehicular ad hoc networks," *Wireless Networks*, 2016.

[7] E. Baccelli, P. Jacquet, B. Mans, and G. Rodolakis, "Multi-lane vehicle-to-vehicle networks with time-varying radio ranges: Information propagation speed properties," in *Proc. IEEE ISIT*, Istanbul, Turkey, July 2013.

[8] D. Cheliotis, I. Kontoyiannis, M. Loulakis, and S. Toumpis, "Exact speed and transmission cost in a simple one-dimensional wireless delay-tolerant network," in *Proc. IEEE ISIT*, Aachen, Germany, June 2017.

[9] P. Jacquet, B. Mans, and G. Rodolakis, "Information propagation speed in mobile and delay tolerant networks," *IEEE Trans. Inf. Theory*, vol. 56, pp. 5001–5015, Oct. 2010.

[10] Y. Li and W. Wang, "Message dissemination in intermittently connected D2D communication networks," *IEEE Trans. Wireless Commun.*, vol. 13, pp. 3978–3990, July 2014.

[11] P. Madadi, F. Bacelli, and G. de Veciana, "On temporal variations in mobile users SNR with applications to perceived QoS," in *Proc. Wiopt*, Tempe, AZ, May 2016.

[12] Y. Peres, A. Sincalir, P. Sousi, and A. Stauffer, "Mobile geometric graphs: Detection, coverage, and percolation," *Probab. Theory Relat. Fields*, vol. 156, no. 1, pp. 273–305, 2013.