317959

Mobile Opportunistic Traffic Offloading

D5.2 – Evaluation of offloading strategies based on simulations

(public)

SEVENTH FRAMEWORK
PROGRAMME

Grant Agreement No.	317959
Project acronym	*MOTO*
Project title	Mobile Opportunistic Traffic Offloading
Advantage	
Deliverable number	D5.2
Deliverable name	Evaluation of offloading strategies based on simulations
Version	V 1.0
Work package	WP 5 – Experimental Validation
Lead beneficiary	INNO
Authors	Maitane Chaves (INNO), Oscar Lázaro (INNO), Patricia Ortiz (INNO), Iván Prada (INNO), Filippo Rebecchi (TCS), Eva Pirattelli (INTECS), Daniele Azzarelli (INTECS), Andrea Passarella (CNR), Raffaele Bruno (CNR), Antonino Masaracchia (CNR), Giovanni Mainetto (CNR), Alessandro Marchetto (CRF), Leandro D'Orazio (CRF)
Nature	R – Report
Dissemination level	PU – Public
Delivery date	31/08/2015 (M34)

Executive Summary

This document is in charge of describing the results of the different simulations that have been conducted in order to evaluate the performance of the different offloading strategies and protocols implemented in MOTO. Two main types of simulations (pedestrian and vehicular) have been conducted in order to cover the main offloading situations where the MOTO services could have important benefits for both the users and the operators.

The simulation environment is high-level described, identifying the modules that have been added to the last release of the MOTO simulation tool. Moreover, the iTetris enhancements that have been implemented and that have resulted into a new iTetris release are presented: http://www.ict-itetris.eu/

In this sense, in order to validate the viability of the proposed security solution, some security simulations have been conducted. The objective of these simulations is to evaluate the impact of applying the security solution to the offloading algorithms.

Table of Contents

List of Figures

List of Tables

1 OVERVIEW

1.1 Introduction

As addressed in the DoW, the objective of this document is to exploit the new features developed by T5.1 and coordinate all the analysis based on simulations. Different simulations have been conducted based on the scenarios proposed in D2.1. Indeed, two main types of simulations (pedestrian and vehicular) have been conducted in order to cover the main offloading situations where the MOTO services could have important benefits for both the users and the operators.

For each simulation scenario the target, layout and description will be presented. Moreover, the expected results for each of them and the offloading algorithms (Constrained Random Waypoint Model, Linear, Krauss model, etc.) that have been used are described. Finally, the parameters, metrics and results of each simulation scenario are defined.

First, pedestrian simulation scenario based on an indoor environment in which the mobility of users is constrained by the building layout (rooms, corridors, stairs, etc.) is presented. This scenario is used to exemplify use case in which an augmented-reality museum experience is offered to the visitors.

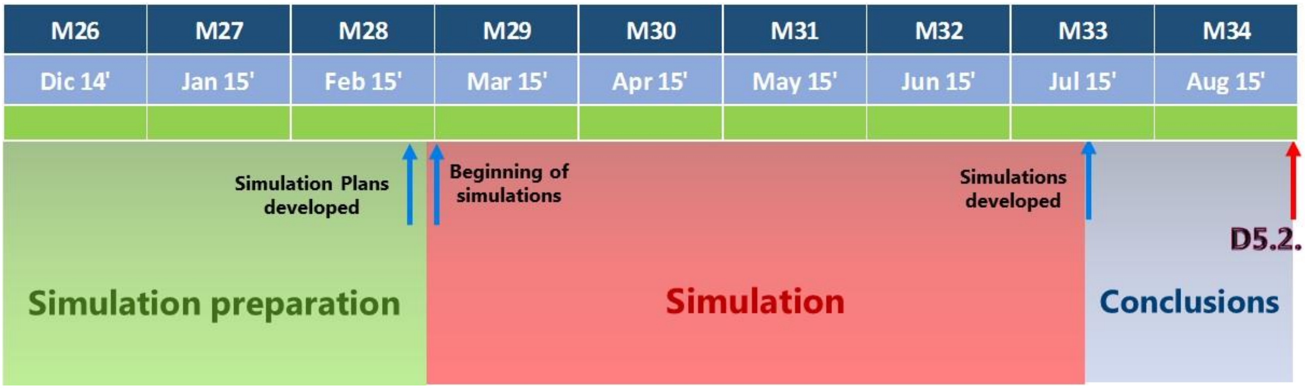
A static simulation scenario considers the case of content distribution in crowded environments such as sport events and concerts. In this scenario, we focus the analysis on the impact of duty-cycling strategies in the neighbour discovery, in order to preserve the battery life of devices.

Second, a vehicular simulation scenario is described that considers a case of geo-relevant content items, being requested by the drivers or passengers (or automatically by the on-board units) of vehicles when entering a certain physical area. The content items are the same for all vehicles, but they can be requested at different points in time.

Third, a simulation of the Map-Based Advanced Driver Assistance Systems for evaluating the behaviour of the proposed mobile opportunistic traffic offloading techniques has been carried out. It takes place in real Italian highways.

Finally, in order to validate the viability of the proposed security solution, some security simulations have been conducted. The objective of these simulations is to evaluate the impact of applying the security solution to the offloading algorithms.

The simulation planning that was scheduled and that has been followed, is the one in Figure 1:



M26	M27	M28	M29	M30	M31	M32	M33	M34
Dic 14'	Jan 15'	Feb 15'	Mar 15'	Apr 15'	May 15'	Jun 15'	Jul 15'	Aug 15'

Figure 1 Simulation planning

1.2 Scope of this Document

This deliverable is organized as follows:

- Section 2 presents a review of the MOTO simulation tool, which is the environment where the MOTO simulations have been conducted. The improvements of the current release are described, as well as the updates of the iTetris simulation tool. This section also presents the main steps to start a simulation.

- Section 3 is divided into three main sub-sections. First, two pedestrian scenarios are presented. Second, the vehicular ones are described. Finally, the security simulations that have been carried out are outlined.
- Section 4 concludes the document.

1.3 Related Documents

This deliverable is related to the following deliverables:

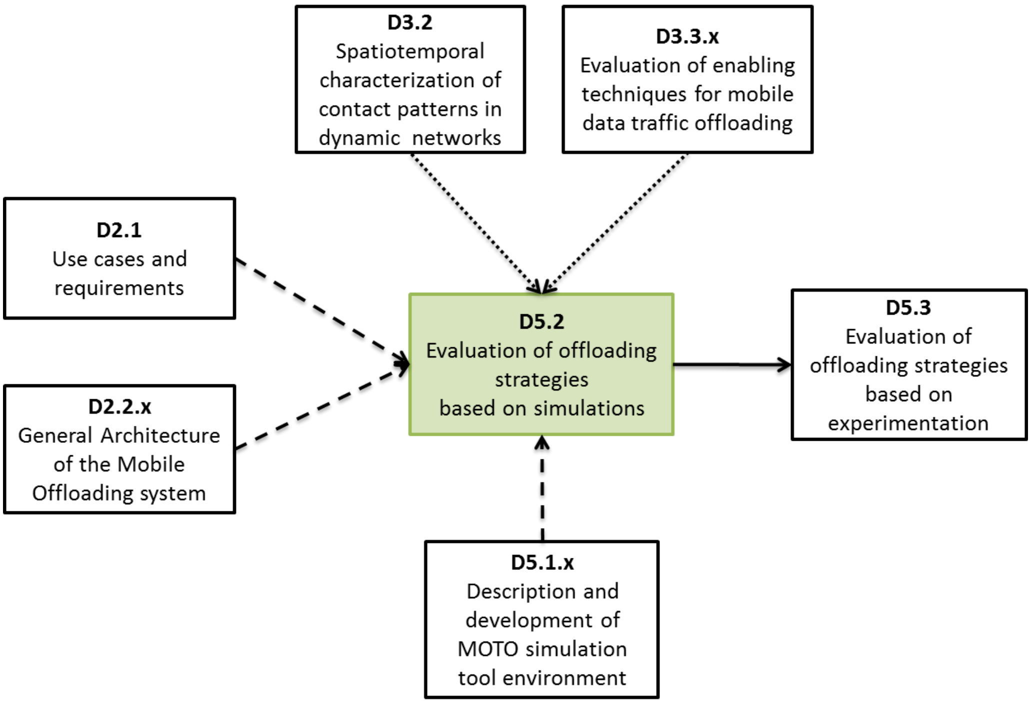

D3.2
Spatiotemporal characterization of contact patterns in dynamic networks
D3.3.x
Evaluation of enabling techniques for mobile data traffic offloading
D2.1
Use cases and requirements
D5.2
Evaluation of offloading strategies based on simulations
D5.3
Evaluation of offloading strategies based on experimentation
D2.2.x
General Architecture of the Mobile Offloading system
D5.1.x
Description and development of MOTO simulation tool environment

Figure 2 Relationship of D5.2 with other deliverables

2 Review of the MOTO simulation tool

In order to evaluate the performance of the different offloading strategies and protocols implemented in MOTO, an appropriate simulation environment is needed. The MOTO simulation platform accomplishes this goal and provides a common simulation platform to be used for the performance evaluation and experimental validation of the proposed algorithms.

The MOTO simulation platform is **based on ns-3**, which is a widely used network simulator for research and education on Internet systems providing a basic environment for running event-driven packet-level simulations, also including packet tracing and collection of statistics. The iTetris platform has been also enhanced in order to fully cover the MOTO requirements for vehicular scenarios.

From the point of view of scalability requirements, the LENA model is aimed to support from several 10 to few 100 eNB and from several 100 to 1000 UE, allowing anyway the creation of interesting interference scenario without having a very large network. Increasing the number of nodes and the simulation time and adding additional features, as D2D communication, the real simulation life grows up exponentially.

About the ns3 development, a modular and flexible architecture has been implemented which provides the creation of new modules to add to ns-3 official release.

In the current release of the **MOTO simulation tool**, the followings modules have been added:

- CDM-Node: it pilots the dissemination procedure; with a periodic evaluation decides if the dissemination status respects what expected and eventually executes the required number of injections.
- UE-Node: it receives from CDM-Node the content through LTE interface and UDP protocol, sends a tracking feedback to CDM-Node by UDP and forwards the content through Wi-Fi interface using epidemic routing algorithm.
- Seeders-Calc: installed on the CDM-Node, offers a customizable interface required to give back the list of UEs designed for the injection procedure.
- Epidemic-Routing: Epidemic Routing spreads contents over the network until the bundle's lifetime is expired by establishing a wireless connection between the Ues that come into contact (neighbours).

About the **iTetris enhancement**, the following activities have been accomplished:

- iTetris has been changed in order to support NS3 v. 3.18 so that all features used in the MOTO simulation tool were granted
- iTetris has been enhanced integrating the LTE support in order to get a tool able to cover vehicular-related scenarios.
- an iTetris sample application has been developed in order to validate the LTE integration.

The new iTetris platform has been uploaded and it is downloadable from the following website: http://www.ict-itetris.eu/. Some user guidelines have been created in order to make it easy to the new users to install the platform. They are also available in this website. In order to obtain both the platform and the guidelines, users should register in the iTetris community, which is exponentially growing since the last iTetris update.

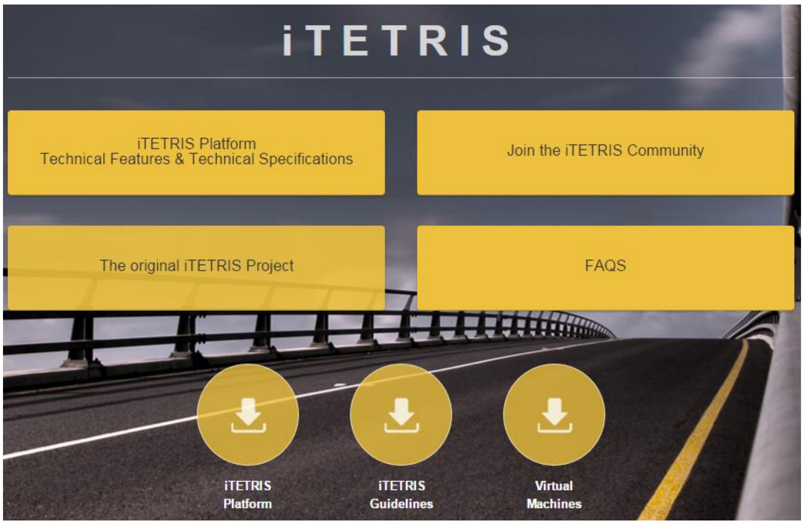

iTETRIS
iTETRIS Platform
Technical Features & Technical Specifications
Join the iTETRIS Community
The original iTETRIS Project
FAQS
iTETRIS Platform
iTETRIS Guidelines
Virtual Machines

Figure 3 iTetris website

2.1 General description and approach

Brief descriptions of the main steps to start a simulation are described in the following summary:

- Configure simulation scenario:
  - Configuration of simulation parameters using the function GeneralConfig()
  - Creation of EPC network
  - Instantiation and configuration of Cdm_Node
  - Creation of n-enbs eNBs
  - Instantiation and configuration of n-ues Ue_Node
  - Mobility configuration for eNBs Ues (ConfigureMobility())
  - Wi-Fi configuration (ConfigureWifi())
  - Installation of epidemic routing on Ues
  - Handover activation
- Set the Stop time for simulation basing on messageTimeLife, *initialPush* and sendPanicZone.
- Start Simulation
- Make available collected metrics (PrintUesMetrics())

The configurable parameters used in the base version of the simulator, to design the simulation scenarios are listed below:

- traceFile: mobility traceFile
- n-ues: number of Ues
- n-enbs: number of eNBs
- messageTimeLife: message lifetime (sec)
- staticNInjects: number of periodic injects
- initialPush: time of first inject (msec)
- enableSendPos: enable Ues forwarding of position message
- frequenceSendPos: frequency of Position forwarding
- deltaT: diffusion State evaluation frequency (msec)
- sendPanicZone: time required for sending messages in Panic Zone (msec)
- dimPacket: bundle dimension (bytes)

- enableEpidemic: enable epidemic routing
- enableTrace: enable trace source connect
- helloIntv: hello messages frequency (msec)
- when-strategy: name of the strategy (initial, linear, slow-linear, fast-linear, square-root)
- who-strategy: name of the strategy (random)
- pushtrack: enable the push and track algorithm
- numMessage: set the number of packets involved in epidemic diffusion

At the end of the simulation, the following files are generated:

1. **CDMLogs.txt**: reports each data printed on standard output, in detail for each Ues:
    a. the modality of reception of the content (LTE injection or Epidemic routing)
    b. number of bytes sent for LTE each interface
    c. number of bytes received for LTE interface
    d. number of bytes sent for WIFI interface
    e. number of bytes received for WIFI interface
    f. throughput for each interface
    g. reception time
    h. delay from first injection
    i. LTE-data acquired through RadioBearerStatsCalculator for each UE (disabled commenting the line lteHelper->EnableTraces)
2. **PieDiffusion.dat**: number of Ues reached by content split up for interface type divided by the total number of reached Ues
3. **NBForInterfaces.dat**: total number of bytes split up for interface type divided by the total number of bytes (sum for all the Ues)
4. **ThrForInterfaces.dat**: total throughput split up for interface type divided by the total throughput (sum for all the Ues)
5. **DelayForInterfaces.dat**: total delay in msec from first inject time split up for interface type
6. **BytesHello.dat**: number of bytes received by WIFI split up for message type (control or data)
7. **GraphResults.ps**: postscript file reporting graphs generated by GnuPlot for point 2, 3, 4, 5

3 Simulation Conduction

In this section, the target, layout and description of each simulation scenario will be presented. Moreover, the expected results for each of them and the offloading algorithms that have been used are described. Finally, the parameters, metrics and results of each simulation scenario are defined. The section has been divided into pedestrian and vehicular simulations.

3.1 Pedestrian simulations

3.1.1 Pedestrian scenario 1

3.1.1.1 Target

In this section, we consider an **indoor environment in which the mobility of users is constrained by the building layout** (rooms, corridors, stairs, etc.). As described in deliverable D2.1, this scenario is used to exemplify a **museum use case** in which an augmented-reality museum experience is offered to the visitors. Specifically, as explained in Section 3.1.1.2, visitors in the museum can download additional multimedia content as they roam through the halls of the museum and get close to the different artworks in each room. The visitors can get this content not only through the cellular network but also by nearby visitors that have that content item in the local caches of their portable devices (e.g., smartphones) using an opportunistic offloading technique.

The main target of these simulations is to **validate the effectiveness of the proposed offloading system also for constrained mobility patterns in indoor conditions**. In addition, we want to investigate the offloading performance as a function of key network parameters, such as the number of available content items, the time the downloaded content items are stored in the local caches (hereafter referred to as the ***sharing timeout***), and the time by which content must be delivered to the requesting users (i.e., the delay tolerance plus panic zone for the content request, hereafter referred to as ***content timeout***).

Therefore, the **performance figure we consider is the offloading ratio**, defined as the fraction of users that have received requested content through the opportunistic network formed by the personal devices of museum visitors, with respect to the total number of users that have requested the content.

3.1.1.2 Layout and description

In this set of simulations, we consider a grid layout that consists of four squared rooms with a side length equal to 20 meters. Then, users move between rooms and within each room using a Constrained Random Waypoint Mobility (CRWM) model. Specifically, in CRWM a user picks a random destination inside the room with probability p_{stay}, or a destination point in one of the other rooms with probability (1-p_{stay}). Then, the user proceeds to this destination point following a straight-line trajectory with constant speed v, and pauses for an exponentially distributed time interval (with mean T_{pause}). As far as the radio propagation model is concerned, we assume a worst-case condition in which users in different rooms cannot directly communicate through the opportunistic channel. Moreover, all users in the same room are in radio visibility of each other. Thus, one-hop opportunistic communications are enough to distribute the content items in the opportunistic network.

We also assume that there is a global set of M content items, but each content item is assigned to a room with an equal probability. Users request content items according to a Poisson process with rate $\boldsymbol{\lambda}$ (i.e. two requests are spaced by an exponentially distributed time interval). It is important to point out that the users can request only content items that are relevant for the room they are visiting. Thus, there is not content dissemination between rooms.

Simulations lasts until all nodes have requested all the content items, and their sharing timeouts are all expired.

3.1.1.3 Expected results

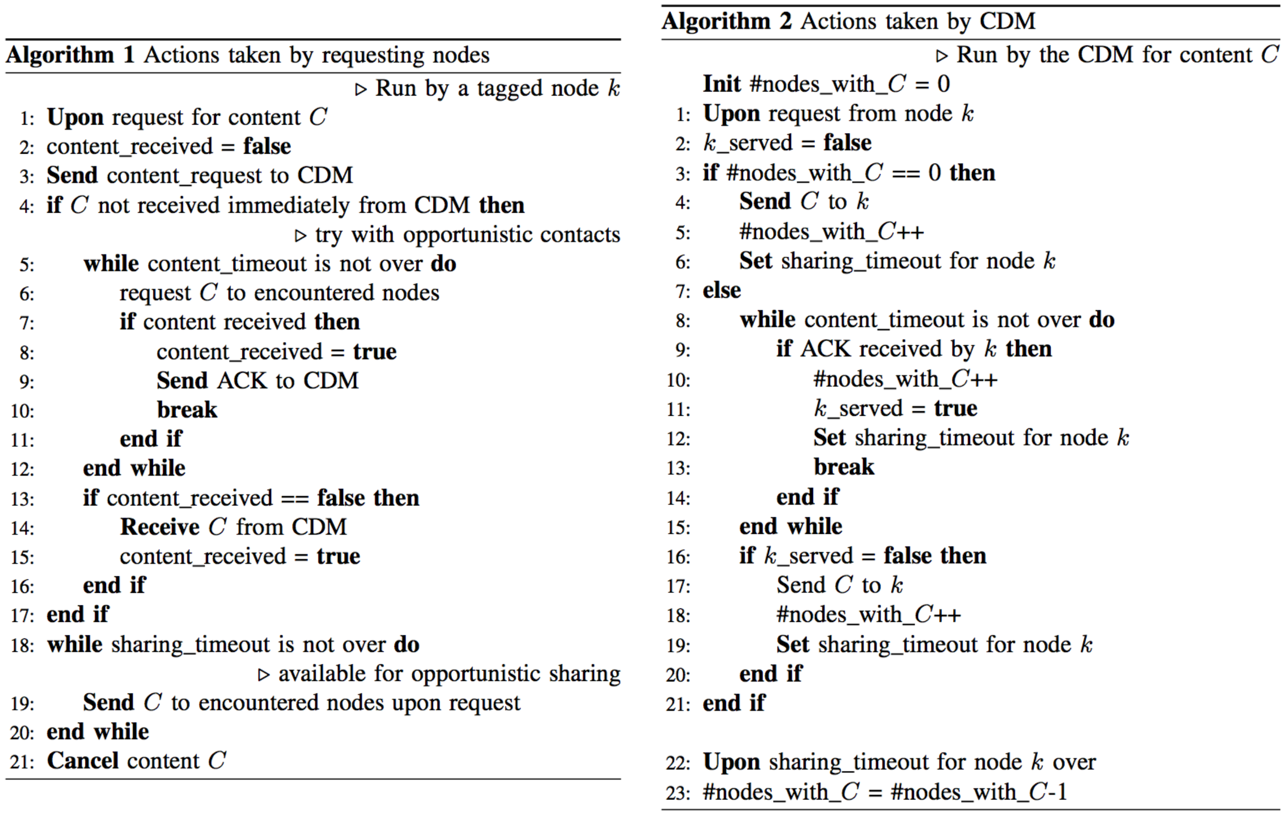
With this simulation, the following high-level results expected are:

- **Dependency on the length of timeout values**: we expect that offloading efficiency would increase with the length of sharing timeouts. On the contrary, the effect of the content timeout is limited because the opportunistic dissemination within each room is fast.

- **Dependency on the number of content items**: we expect that offloading efficiency would decrease with the number of content items, due both to increasing contention of the opportunistic network resources and lower probability that users are simultaneously interested to the same content item.

3.1.1.4 Offloading algorithms involved

The offloading algorithm we have defined is compliant with the general MOTO architecture presented in D2.2.1. We assume the existence of a Central Dissemination Manager (CDM) that can communicate with all nodes through the cellular network and keeps track of the dissemination process. The offloading mechanism is defined by the actions taken by requesting nodes and by the CDM, as described by Algorithms 1 and 2, respectively, shown in Figure 4.



Algorithm 1 Actions taken by requesting nodes

```
                                          ▷ Run by a tagged node $k$
 1: Upon request for content $C$
 2: content_received = false
 3: Send content_request to CDM
 4: if $C$ not received immediately from CDM then
                                     ▷ try with opportunistic contacts
 5:     while content_timeout is not over do
 6:         request $C$ to encountered nodes
 7:         if content received then
 8:             content_received = true
 9:             Send ACK to CDM
10:             break
11:         end if
12:     end while
13:     if content_received == false then
14:         Receive $C$ from CDM
15:         content_received = true
16:     end if
17: end if
18: while sharing_timeout is not over do
                                 ▷ available for opportunistic sharing
19:     Send $C$ to encountered nodes upon request
20: end while
21: Cancel content $C$
```

Algorithm 2 Actions taken by CDM

```
                                    ▷ Run by the CDM for content $C$
    Init #nodes_with_$C$ = 0
 1: Upon request from node $k$
 2: $k$_served = false
 3: if #nodes_with_$C$ == 0 then
 4:     Send $C$ to $k$
 5:     #nodes_with_$C$++
 6:     Set sharing_timeout for node $k$
 7: else
 8:     while content_timeout is not over do
 9:         if ACK received by $k$ then
10:             #nodes_with_$C$++
11:             $k$_served = true
12:             Set sharing_timeout for node $k$
13:             break
14:         end if
15:     end while
16:     if $k$_served = false then
17:         Send $C$ to $k$
18:         #nodes_with_$C$++
19:         Set sharing_timeout for node $k$
20:     end if
21: end if

22: Upon sharing_timeout for node $k$ over
23: #nodes_with_$C$ = #nodes_with_$C$-1
```

Figure 4. Algorithms for offloading in non-synchronized requests.

Let us focus first on the actions taken by requesting nodes (Algorithm 1). When a request is generated at a node, the node sends it to the CDM via the cellular network (line 3). The node is guaranteed to receive the content within a given *content timeout*. During the timeout, the node tries to get the content from encountered nodes (lines 5-12). If the timeout expires, it receives it directly from the CDM (lines 13-16). Upon receiving the content, the node sends an ACK to the CDM (line 9 and, implicitly, line 14). In addition, it keeps the content for a *sharing timeout*, during which it can share the content with other encountered nodes (lines 18-20). After the expiration of the *sharing timeout,* the content is deleted from the local cache. Note that requests and ACKs are supposed to be much shorter than the content size, and thus do not significantly load the cellular network.

Let us now focus on the actions taken by the CDM (Algorithm 2). Thanks to requests and ACKs, the CDM is always aware of the status of content availability in the network. Upon receiving a request, it checks whether some other node is already storing a copy of the content or not. In the latter case (lines 4-6), there is no chance that the user can get the content opportunistically through another node, and the CDM sends the content directly through the cellular network. In the former case (lines 7-21), it waits to receive an ACK during the *content timeout* (lines 8-15), indicating that the node has received the content. If this does not happen, it sends the content directly to the node (lines 16-20). Finally, upon expiration of the *sharing timeout* for a given node the CDM updates the view on the number of nodes with the content (lines 22-23).

3.1.1.5 Simulation parameters

We ran simulations using the **standard MOTO simulation environment** developed in T5.1, for various sets of parameters, as indicated in the below table. We performed at least five simulation runs for each set of parameters, using the independent replication method. Then, we computed the confidence intervals (with 95% confidence level) over the replications for the performance metrics that we describe in Section 3.1.1.6.

Parameter	Variable/fixed	Units	Min	Max
Number of nodes	Fixed	Number	20	20
Number of seeds	Variable (decided by the algorithm)	Number	1	20
Number of eNBs	Fixed	Number	1	1
Type of content		Video/photo/text		
Size of content	Fixed	KBytes	10	10
Number of Messages	Variable	Number	20	60
Message lifetime	Variable	s	10	30
Time to panic zone	Fixed	s	0	0
Mobility patterns of the nodes		**Constrained Random Waypoint Model**		
T_{pause}	Fixed	s	60	60
P_{stay}	Fixed	Number	2/3	2/3
Routing protocol		Direct Transmission		
Speed of nodes	Fixed	m/s	5	5
LTE cell diameter	Fixed	Km	1	1
Transmission range	Fixed	m	30	30
Request rate	Fixed	Req/s	1	1
Sharing timeout	Variable	s	20	60

Table 1. Simulation parameters of the Pedestrian Scenario 1

3.1.1.6 Metrics

The main performance figure we consider is the offloading efficiency, defined as the fraction of content messages that reach the users through opportunistic communications. To get a more precise idea on the dynamics of the offloading process over time, we also computed, on each 5s time window, the average (across simulation replicas) number of copies of content items stored on mobile nodes, and the average number of new content deliveries through the cellular and the opportunistic network, respectively.

Metrics	Units	Min	Max
Offloading efficiency	**%**	0	100
Number of nodes receiving content via LTE	Number	0	20

Number of nodes receiving content via opportunistic	Number	0	20

Table 2. Metrics of the pedestrian scenario 1

3.1.1.7 Simulation results

Figure 5 shows the offloading efficiency for different sharing timeouts and number of content items. It is intuitive to observe that the longer the sharing timeout, the higher the offloading efficiency, as content items remain available in the opportunistic network longer. Furthermore, the more content items there are, the lower is the offloading efficiency. Interestingly the decrease in offloading efficiency is almost linear with the increase in the number of content items. This can be explained by observing that a user can randomly request any of the content items that are assigned to the room where he/she is. Thus, the more the content items and the lower the probability that two users request the same content (although not at the same time). Clearly, the opportunistic offloading technique is effective only when there are multiple requests of the same content. It is also worth pointing out that a content item can be disseminated only in the room it is relevant. In fact, when a user changes room, the content items that are stored in his/her local cache are not disseminated anymore, because in the new room there are not users interested in that content. Thus, the cached content items are simply deleted as the sharing timeouts elapse.

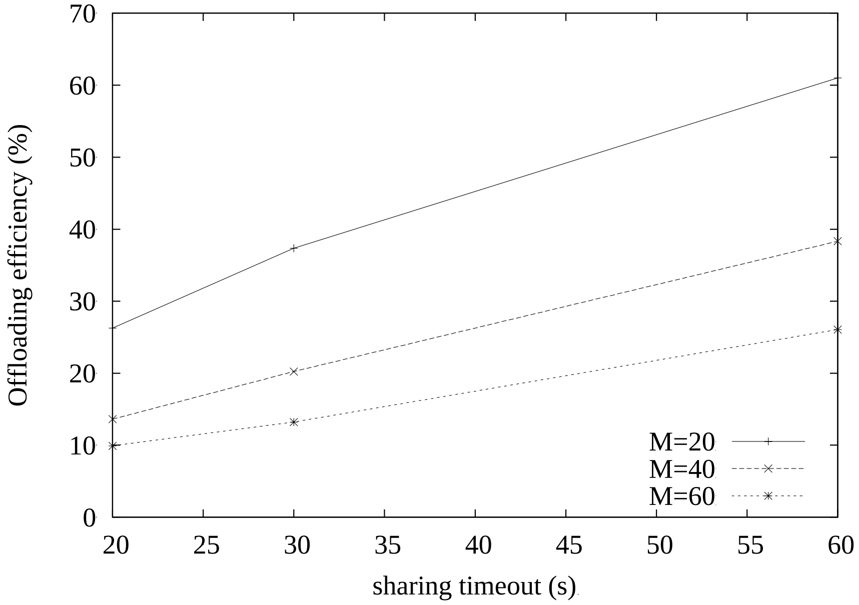


Figure 5. Comparison for different number of content items - req_rate=1 req/s, content timeout 10s

Figure 6 shows the offloading efficiency for different content timeouts. It is interesting to observe that the offloading efficiency is not affected by the content timeout (at least for the considered parameter setting). This can be explained by observing that all users in a room can directly communicate with each other. In this condition, the opportunistic dissemination is typically fast in distributing the content item to the interested users, and the capacity of the opportunistic communications is mainly limited by the channel contention. Thus, increasing the content timeouts does not improve the offloading efficiency because local copies of the content items are rapidly disseminated to interested users.

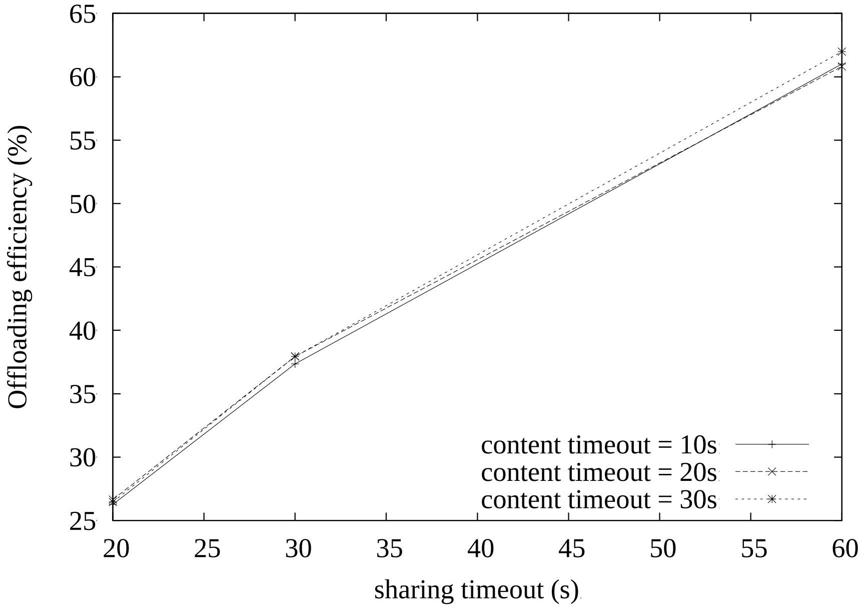


Figure 6. Impact of sharing and content timeout - req_rate=1 req/s, M=20

Finally, from Figure 7 to Figure 10 we show the temporal evolution of content diffusion for short and long sharing timeouts (20 and 60 seconds, respectively) and different numbers of content items (M=20 and M=60). All the showed results refer to a content timeout of 10 seconds. Curves labelled with “LTE” denote the number of users that, up to that time, have received the content via the cellular network. Curves labelled with “Wi-Fi” denote the number of users that, up to that time, have received the content via the opportunistic network. Finally, curved labelled with “Copies in the network” denote the average number of users that, at a given point in time, store a copy of any content item (i.e., how much each content item is replicated in the network). It is important to note that for short sharing timeouts the content items stay available in the opportunistic network for short amounts of time. Thus, the combined effect of short sharing and content timeouts caused content timeouts to frequently elapse before the opportunistic diffusion is able to distribute the cached content item to other interested users. On the other hand, when the sharing timeout increases there are more copies of the content item in the opportunistic network and this boosts the dissemination process. Specifically, as shown in Figure 8 the number of nodes receiving content items via the opportunistic network rapidly increases at the beginning of the simulation and more nodes are served by the opportunistic network than by LTE. However, when nodes start changing rooms (after 180 seconds, on average), the efficiency of the dissemination process decreases and more content items are sent through LTE.

When there are more content items to choose from (Figure 9 and Figure 10), the utility of the opportunistic dissemination further degrades because less nodes are interested in the same content item simultaneously. Thus, even with long sharing timeouts, the number of copies of a specific content item can be rather low.

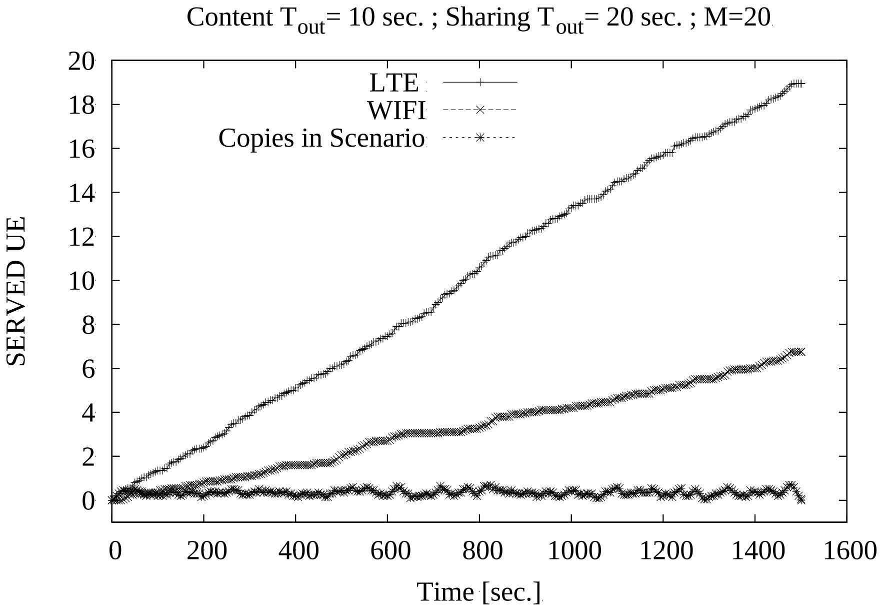


Figure 7. Temporal evolution: short sharing timeouts and M=20.

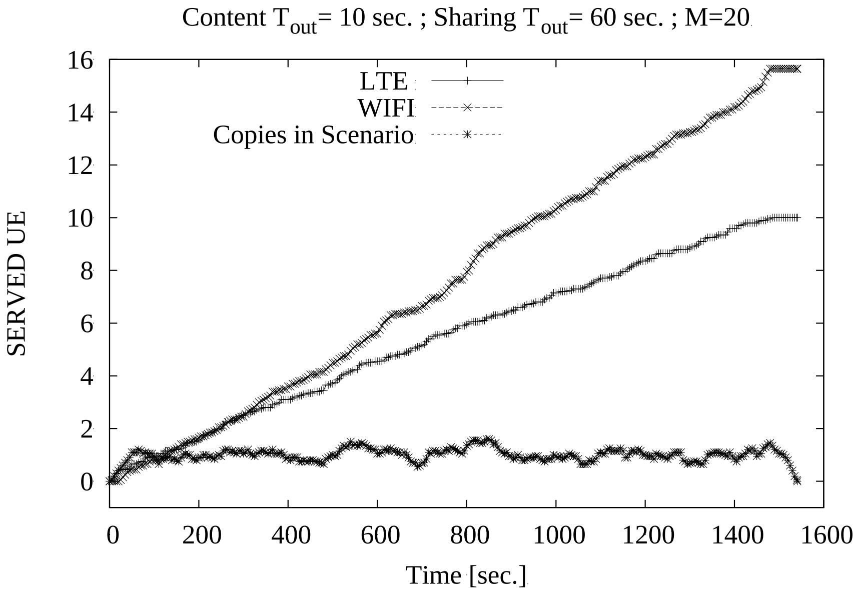


Figure 8. Temporal evolution: long sharing timeouts and M=20.

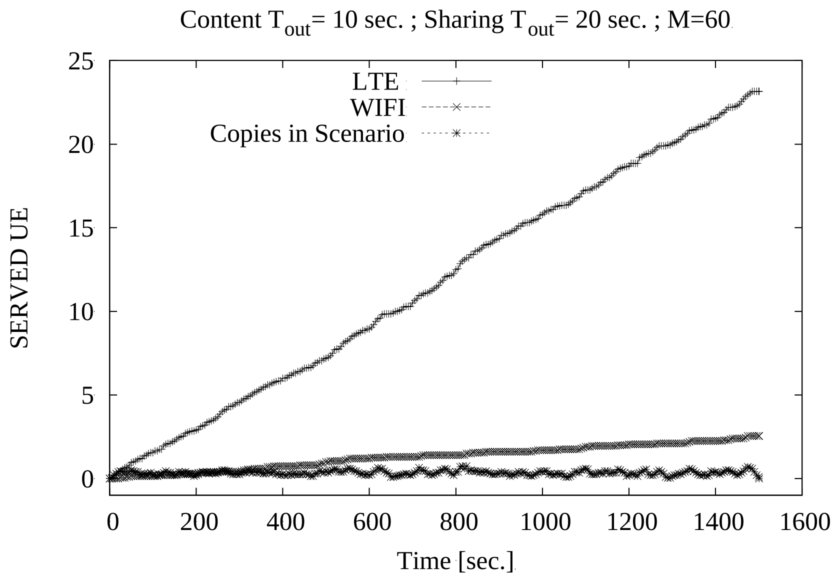


Figure 9. Temporal evolution: short sharing timeouts and M=60

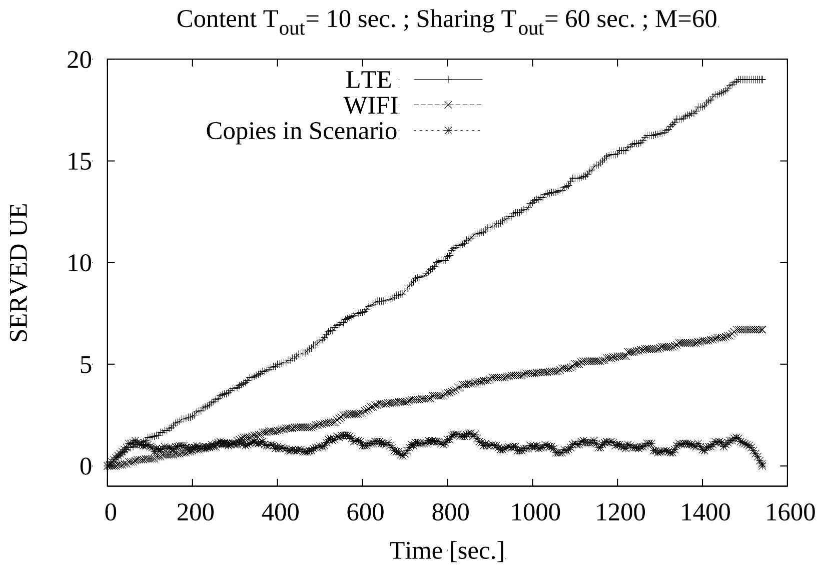


Figure 10. Temporal evolution: long sharing timeouts and M=60.

3.1.1.8 Simulation conclusions

In conclusion, we can say that, as was expected, simulation results validated that **a constrained mobility model** (which is typical in indoor conditions) **reduces the effectiveness of the opportunistic dissemination** because different users can be interested in disjoint sets of content items based on their location. On the positive side, **a long shared timeout is effective in improving the opportunistic diffusion** because more copies stay in the network. Finally, the content timeout may have a negligible impact if the node density is sufficiently high to speed up the opportunistic dissemination process.

3.1.2 **Pedestrian scenario 2**

3.1.2.1 Target

In this section, we consider an **outdoor environment with static mobility**. As described in deliverable D2.1, this scenario is used to exemplify data dissemination in crowded situation to handle peak of traffic demands. As explained in D2.1, Section 3.2.1.4, spectators in a stadium can download picture or video clips taken from other spectators in the same stadium. The spectators can get the content not only through the cellular network but also by nearby visitors that have that content item in the local caches of their portable devices (e.g., smartphones) using an opportunistic offloading technique.

Therefore, the main target of these simulations is to **evaluate the effectiveness of the proposed offloading system for a fixed mobility scenario**, such as the photo in a stadium scenario. However, employing data offloading should satisfy user demand without affecting too much the battery lifetime. Thus, the objective of minimizing traffic volumes in the RAN should not be considered in isolation. Instead, the requirements in terms of battery lifetime of mobile devices participating in the data offloading process should also be taken into consideration.

For this reason, we want to investigate the offloading performance as a function of **duty-cycling** strategies to allow increased battery duration of mobile devices. Duty-cycling is a classic technique, widely employed to minimize the battery consumption in wireless mobile networks without infrastructure, which has already been tested in sensor, delay tolerant, and opportunistic networks.

Therefore, we focus on two performance metrics: **offloading ratio** and **energy consumption**. Offloading ratio is defined as before as the fraction of users who received content through the opportunistic network with respect to the total number of users that have requested the content. The other important figure of interest, energy consumption, is the average energy consumed by the Wi-Fi radio of a user during data dissemination.

3.1.2.2 Layout and description

In this set of simulations, we consider a **static** layout, consisting of a rectangular stand with side lengths equal to 50 x 25 meters. Users are static – we consider that during the sport match, they are seated, they do not move - and they are distributed randomly with uniform probability in the area.

The stand is covered by a single eNB (femto-cell), and all the users in the stand are connected to this eNB. The eNB is located in the middle of the lower side (25, 0).

We also assume that there is only one information flow, which is broadcasted during the game. Each information flow is composed of a global set of M pieces (packets of fixed size of 10 KB). Content is considered received when all the M packets are retrieved by a user. However, users can start disseminating the content before the total completion, as soon as they store any content piece. Simulations lasts until all nodes have retrieved all the content items, and the content timeout is expired.

3.1.2.3 Duty-cycling

While not considering mobility, in this set of simulations we take into account the use of duty-cycling schemas in the neighbour discovery process, making our network truly opportunistic.

In a totally distributed environment, such as those of MOTO, users are in charge of discover neighbour nodes by explicitly probing their vicinity. The continuous probing results in high battery drain, since the radio interface is continuously transmitting *Hello* messages. In addition, probing can be useless because of no neighbouring nodes. By implementing some duty-cycling strategies, we can obviously lower battery depletion, at the cost of a reduced knowledge of neighbourhood, which in turns could result in lower offloading performance.

We compare the performance of the following two configuration parameters with respect to the above performance metrics:

- **Discovery time:** the period of time during which the probing is **active**. For the purpose of this study, while in the discovery phase, the probing happens each 250 ms.
- **Silence time:** the period during which probing is **inactive.**

Discovery and silence time alternates. During a silence time, a node cannot be discovered by its neighbourhood but its wireless interface is still on and it can hear neighbours sending *Hello* messages. It is important to note that nodes in simulation are **not synchronized** with each other and their on-off pattern typically does not overlap. The advantage of duty-cycling time resides also in the fact that during the silence times a node does not pollute the radio medium with its *Hello* messages, offering better chances to other nodes to communicate.

3.1.2.4 Expected results

With this simulation, the following high-level results are expected to be obtained:

- **Dependency on the values of duty-cycling strategies**: we expect that offloading efficiency would decrease with the length of silencing. On the other hand, an aggressive duty cycling should help preserving the battery of users. The interplay between these two conflicting goals should be evaluated.
- **Dependency on the number of content items**: we expect that offloading efficiency would decrease with the number of content items, due both to increasing contention of the opportunistic network resources and lower probability that users are simultaneously interested to the same content item.

3.1.2.5 Offloading algorithms involved

In the following simulations, we adopt the same offloading algorithm we have described in Section 3.1.1.4.

3.1.2.6 Simulation parameters

We ran simulations using the **standard MOTO simulation environment** developed in T5.1, for various sets of parameters, as indicated in the below table. We performed at least five simulation runs for each set of parameters, using the independent replication method. Then, we computed the confidence intervals (with 95% confidence level) over the replications for the performance metrics that we describe in Section 3.1.1.6.

Parameter	Variable/fixed	Units	Min	Max
Number of nodes	Variable	Number	10	20
Number of seeds	Variable (decided by the algorithm)	Number	1	20
Number of eNBs	Fixed	Number	1	1
Type of content		Video/photo/text		
Size of content	Fixed	KBytes	10	10
Number of Messages	Variable	Number	10	1000
Message lifetime	Variable	s	10	30
Time to panic zone	Fixed	s	0	0
Mobility patterns of the nodes		Fixed Random Uniform		
Routing protocol		Direct Transmission		
LTE cell diameter	Fixed	m	25	50
Transmission range	Fixed	m	30	30
Request rate	Fixed	Req/s	1	1
Sharing timeout	Variable	s	10	30
Discovery time	Variable	s	0,25	5
Silence time	Variable	s	0,25	10

Table 3. Simulation parameters of the pedestrian scenario 2

3.1.2.7 Metrics

The main performance figure we consider is the offloading efficiency, defined as the fraction of content messages that reach the users through opportunistic communications. To get a more precise idea on the dynamics of the offloading process over time, we also computed, on each 5s time window, the average (across simulation replicas) number of copies of content items stored on mobile nodes, and the average number of new content deliveries through the cellular and the opportunistic network, respectively.

3.1.2.8 Simulation results

As a reference point, we consider the offloading performance **without any duty-cycling** strategy for 10 users. To reduce the parameter space, in the following simulations we always consider that the node's sharing timeout equals the content timeout. As hinted in Figure 11, the longer the content (sharing) timeout and the higher the offloading efficiency, as the content items have more time to disseminate. However, the number of packets to distribute has the larger impact on efficiency. We note that there is a threshold effect: for a content made of only 10 packets, the offloading is nearly perfect even at 10 s, while for a large content made of thousands of packets offloading capacity is limited (in relative terms). Instead, the impact of the content timeout, is visible for the case with M=100 packets, where larger timeout offers an increased efficiency.

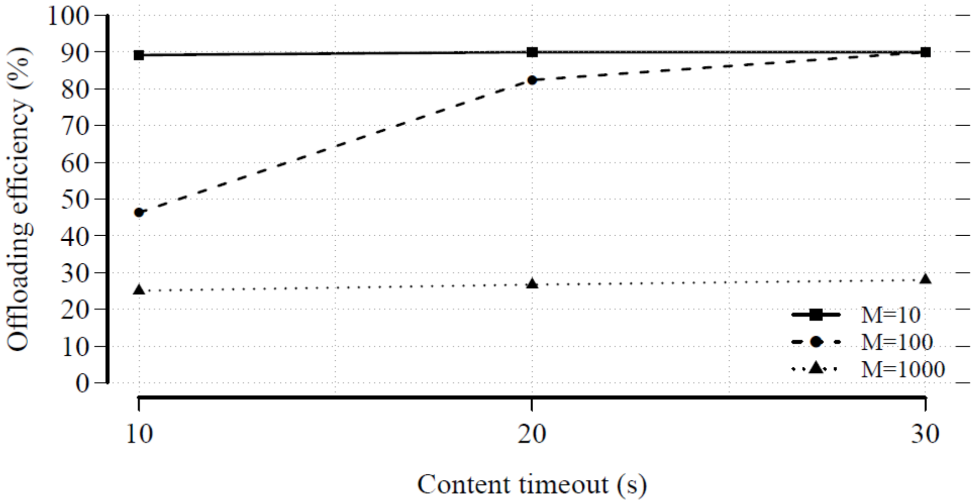


Figure 11: Comparison for different number of content items and timeout without any duty-cycling strategy - req_rate=1 req/s, 10 UEs.

Adding duty-cycling strategies as expected reduces the efficiency of data offloading, allowing decreasing the energy consumption experienced by mobile nodes. To show this effect, we focus on the M=100 (total content size = 1 MB) scenario and we consider a **discovery time** of 250 ms varying the value of the **silence time** parameter, so as to increase the fraction of time without beaconing. As benchmark, we consider a silence time of 250 ms, which represents the usual interval between two successive beacons in case of no duty-cycling. From Figure 12 to Figure 14, we consider the scenario with a respective content timeout 10, 20, and 30 s. In this case, we evaluate the average energy consumption that a user spends during the complete content dissemination process. It is important to note that beaconing starts as soon as users make the request for the content to the CDM, and stops whenever all the M packets are received or when the content timeout expires. Without surprise, in nearly all the simulation, we see that there is a **linear relationship** between the amount of data saved on the cellular infrastructure and the energy consumption.

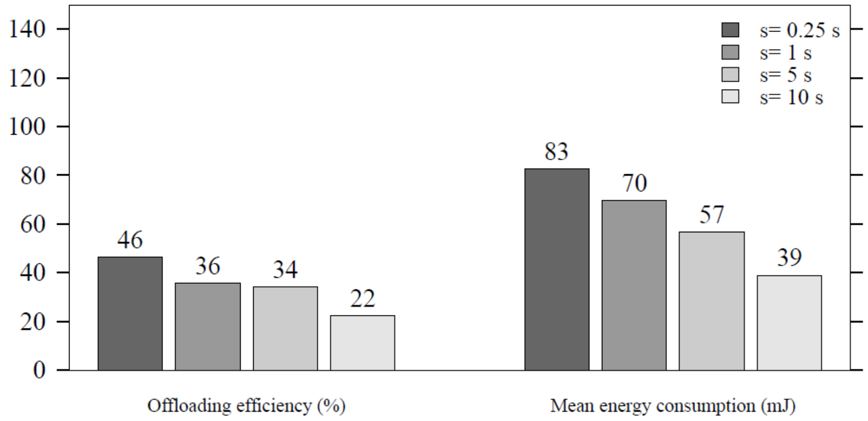


Figure 12: Offloading efficiency and energy consumption under different values of the silence time parameter s. Content timeout=10 s, M=100.

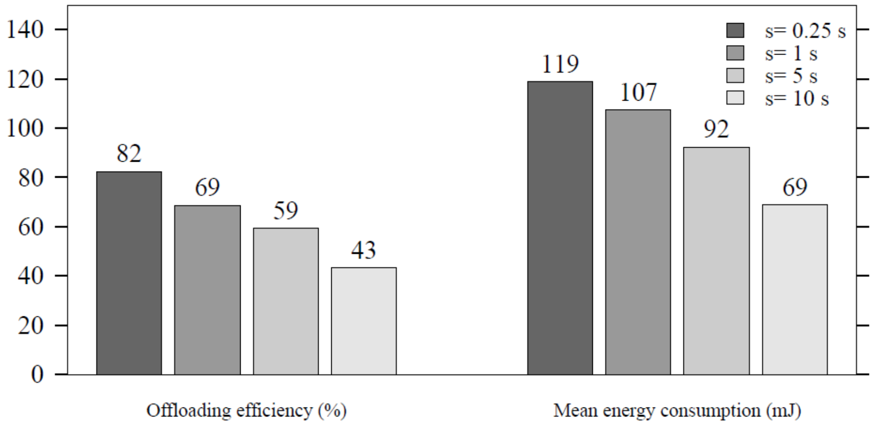


Figure 13: Offloading efficiency and energy consumption under different values of the silence time parameter s. Content timeout=20 s, M=100.

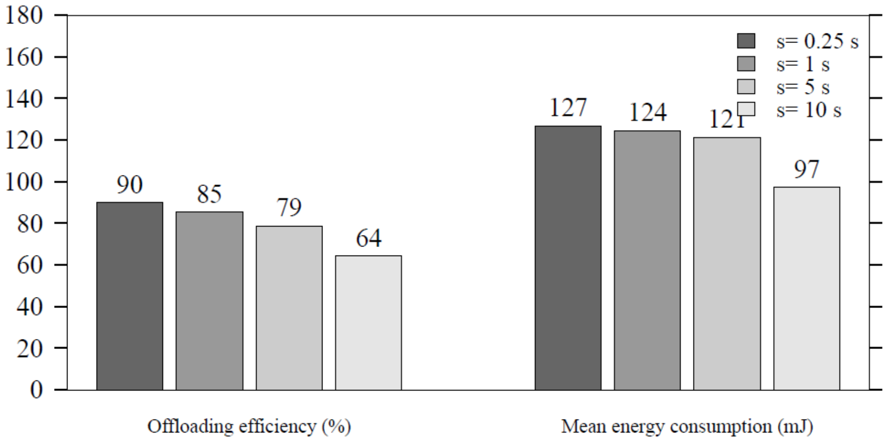


Figure 14: Offloading efficiency and energy consumption under different values of the silence time parameter s. Content timeout=30 s, M=100.

We must consider however, that the figures on the energy consumption include all the transmissions made on the Wi-Fi interface. This is the reason why in Figure 14 the impact of duty-cycling is reduced in terms of consumed energy. In this case, almost the entire energy is spent in the transmission of actual data and not in the discovery of neighbors –the efficiency is much higher in this scenario- reducing inherently energy savings.

Finally, from Figure 15Figure 7to Figure 18 we evaluate the temporal evolution of content dissemination on the different interfaces for different values of content timeouts (10 and 30 seconds, respectively), discovery and silence time (s=250 milliseconds and s= 10 seconds). All the results refer to the case with 10 UEs.

We note that for short content timeouts the packet availability in the opportunistic network is limited, especially in the case with aggressive duty-cycling strategies. Thus, the combined effect of long silence time and short content timeouts cause a lot of panic zone re-injection (the zone between 20 and 35 seconds in Figure 16) lowering the offloading efficiency.

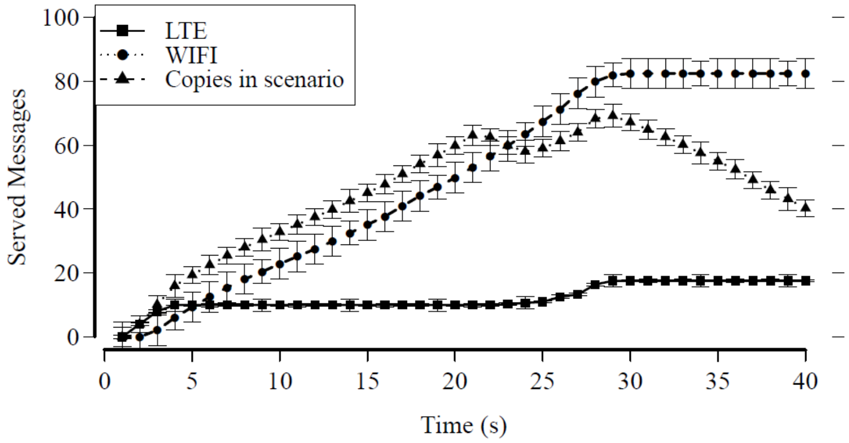


Figure 15: Temporal evolution - content timeouts=10 s, M=100, discovery time=250 ms, silence time=250 ms.

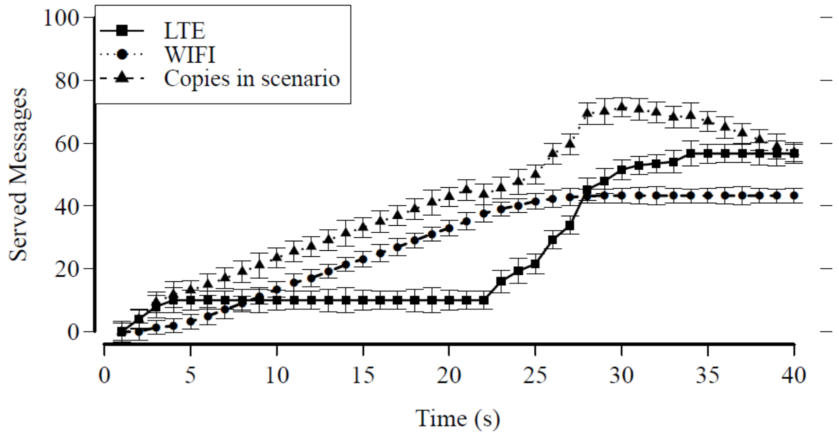


Figure 16: Temporal evolution - content timeouts=10 s, M=100, discovery time=250 ms, silence time=10 s.

On the other hand, when the content timeout increases there are more copies of the packets in the opportunistic network, contributing to boost the dissemination process. In Figure 17, the number of nodes receiving content items via the Wi-Fi network quickly increases at the beginning of dissemination, at the same time as the number of copies stored in the network. In this case, the LTE has the only purpose of kick-start the dissemination.

In this scenario, adding a duty-cycling strategy as in Figure 16 concurs in lowering the share of packets received by opportunistic transmissions. Thus, even with long sharing timeouts, the number of copies of a specific content item can be rather low, requiring the intervention of the CDM by injecting additional copies in panic zone.

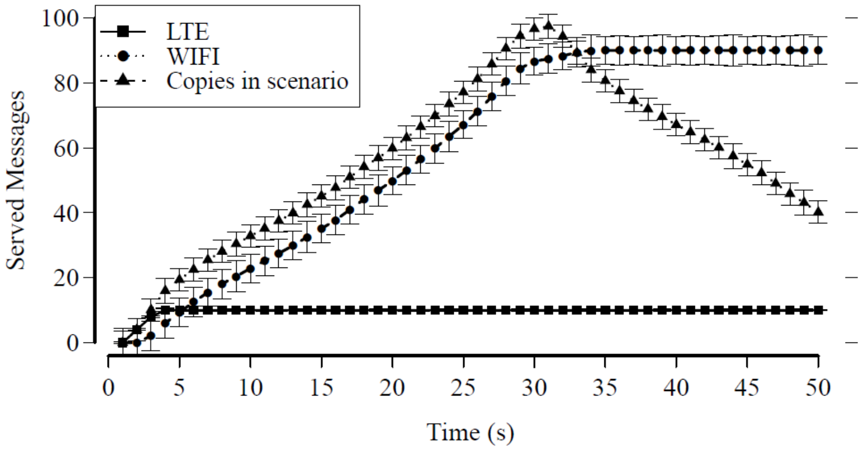


Figure 17: Temporal evolution - content timeouts=30 s, M=100, discovery time=250 ms, silence time=250 ms.

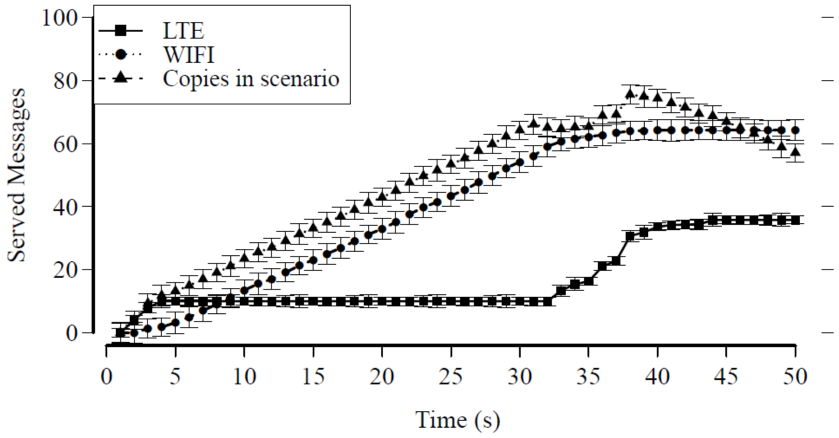


Figure 18: Temporal evolution - content timeouts=30 s, M=100, discovery time=250 ms, silence time=10 s.

3.1.2.9 Simulation conclusion

In this section, we evaluated the impact that **duty-cycling** techniques (a classic strategy to minimize battery consumption in mobile networks) have on the **efficiency of the mobile offloading strategies** proposed in MOTO. What emerges from the analysis is that there is a linear relationship between the energy saved and the reduction of offloading efficiency. Therefore, from the perspective of mobile operators, data offloading should take into account the requirements of mobile nodes. Energy consumption of mobile users becomes a design parameter that should be considered, along with its implication on the efficiency, in the choice of the offloading algorithm. As already pointed out in the previous section, **a long content timeout is effective in improving the opportunistic diffusion at the cost of increased battery usage.**

3.2 Vehicular simulations

3.2.1 Vehicular Scenario 1

3.2.1.1 Target

In this scenario, we consider the case of **geo-relevant content items, being requested by the drivers or passengers** (or automatically by the on-board units) of vehicles when entering a certain physical area. As explained in Section 3.2.1.2, the content items are the same for all vehicles, but they can be requested at different points in time. These content items are therefore temporarily stored at vehicles, which can exchange them opportunistically when encountering each other.

The **main target of the simulation is to test the performance of this offloading system**, as a function of the key network and environment parameters. Specifically we characterize the offloading performance by varying (i) the request generation rate (ii) the number of available content items, (iii) the time during which vehicles share received content, and (iv) the time by which content must be delivered to the requesting users.

Therefore, the performance figure we consider is the offloading ratio, defined as the fraction of users that have received requested content through the opportunistic network formed by vehicles, with respect to the total number of users that have requested the content.

3.2.1.2 Layout and description

In this set of simulations, we extend the settings used in the preliminary evaluation of the tested solution presented in D3.3.2. More precisely, in these simulations we capture cases where a group of vehicles move inside a geographical area covered by a cell, and roam always inside that cell. Vehicles move on a stretch of road crossing the cell, and come back when arriving at the boundary. The resulting traffic is therefore bidirectional. Nodes move with a speed randomly selected (with uniform distribution) in an interval [vmin, vmax], and can exchange content directly while being within a maximum transmission range T_{RX} from each other. We consider N nodes in the simulations, which can all request a set of content items (there are M content items in total). Requests are generated from the beginning of the simulation sequentially, according to a Poisson process with rate $\boldsymbol{\lambda}$ (i.e. two requests are spaced by an exponentially distributed time interval). Simulations last until all nodes have requested all the content items, and their sharing timeouts are all expired.

We ran simulations using the **standard MOTO simulation environment** developed in T5.1, for various sets of parameters, as indicated in Section 3.2.1.4. Specifically, we varied the number of nodes, the request rate, the content timeout, the sharing timeout and the number of content items exchanged. We performed at least five simulation runs for each set of parameters, using the independent replication method. The main performance figure we consider is the offloading efficiency, defined as the fraction of content messages that reach the users through opportunistic communications. For this index, we computed the confidence intervals (with 95% confidence level) over the replications. To get a more precise idea on the dynamics of the offloading process over time, we also computed, on each 5s time window, the average (across simulation replicas) number of copies of content items stored on mobile nodes, and the average number of new content deliveries through the cellular and the opportunistic network, respectively.

3.2.1.3 Expected results

With this simulation, the following high-level results are expected to be obtained:

- Dependency on the length of timeout values: we expect that offloading efficiency would increase with the length of both timeout values (sharing and content timeouts)
- Dependency on the number of content items: we expect that offloading efficiency would decrease with the number of content items, due to increasing contention of the opportunistic network resources
- Dependency on the request rate: we expect that offloading efficiency would increase with request rate, as more copies of the content items should be available in the network

3.2.1.4 Offloading algorithms involved

In the following simulations, we adopt the same offloading algorithm we have described in Section 3.1.1.4.

3.2.1.5 Simulation parameters

To comply with these requirements, the defined parameters for the simulation and their variability will be:

Parameter	Variable/fixed	Units	Min	Max
Number of nodes	Fixed	Number	20	20
Number of seeds	Variable (decided by the algorithm)	Number	1	20
Number of eNBs	Fixed	Number	1	1
Type of content		Video/photo/text		
Size of content	Fixed	KBytes	10	10
Message lifetime	Variable	s	60	120
Time to panic zone	Fixed	s	0	0
Mobility patterns of the nodes		Linear		
Routing protocol		Direct Transmission		
Speed of nodes	Variable	Km/h	80	120
LTE cell diameter	Fixed	Km	4	4
Transmission range	Fixed	m	200	200
Request rate	Variable	Req/s	0.5	1
Sharing timeout	Variable	s	5	120

Table 4. Simulation parameters of the vehicular scenario 1

3.2.1.6 Metrics

The metrics that will be measured within the simulation are the following:

Metrics	Units	Min	Max
Offloading efficiency	**%**	0	100
Number of nodes receiving content via LTE	Number	0	20
Number of nodes receiving content via opportunistic	Number	0	20

Table 5. Metrics of the vehicular scenario 1

3.2.1.7 Simulation results

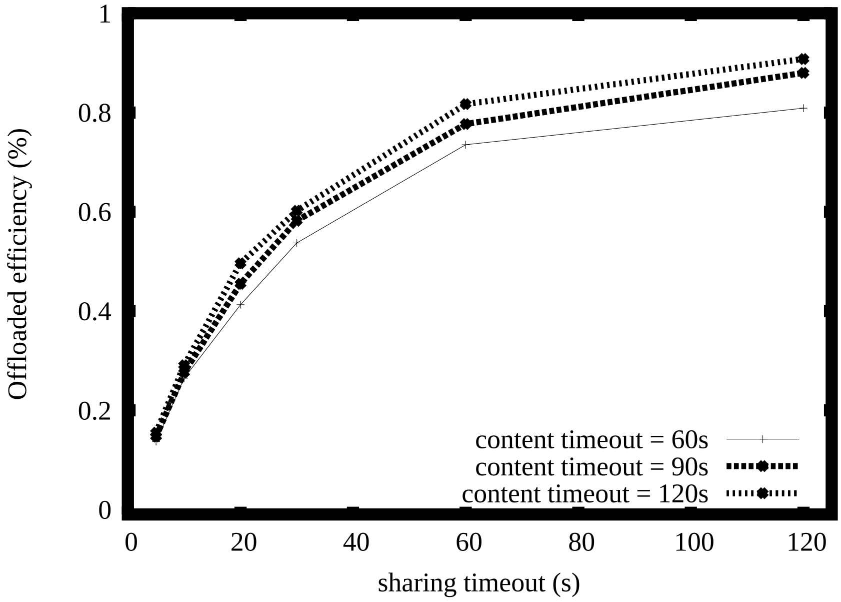


Figure 19. Impact of sharing and content timeout - req_rate=1 req/s, M=10

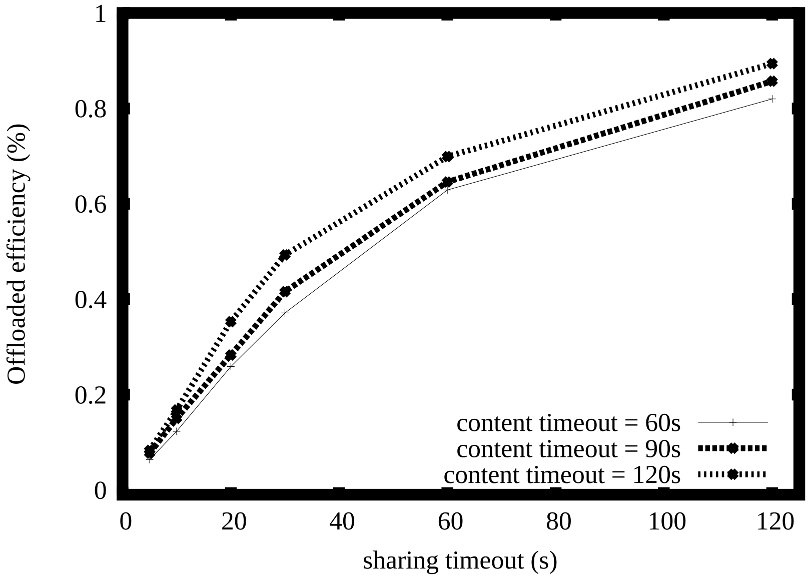


Figure 20. Impact of sharing and content timeout - req_rate=0.5 req/s, M=10

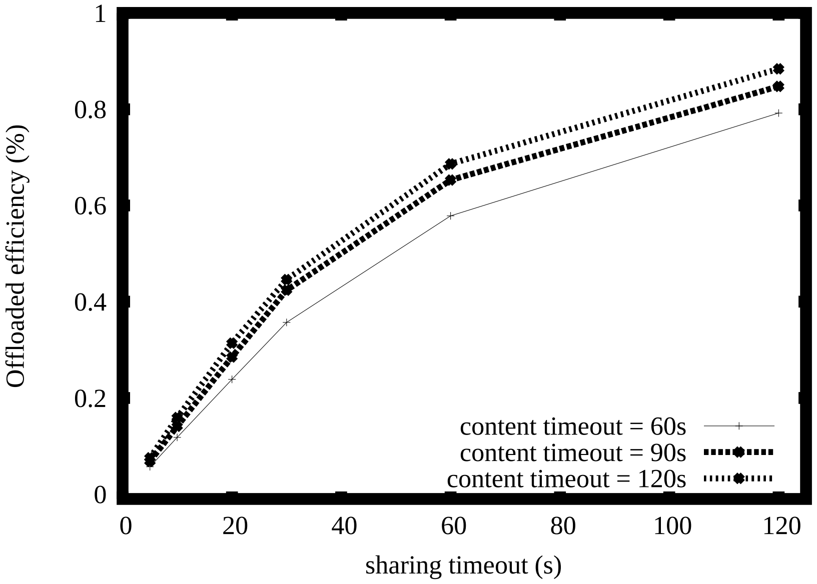


Figure 21. Impact of sharing and content timeout - req_rate=1 req/s, M=20

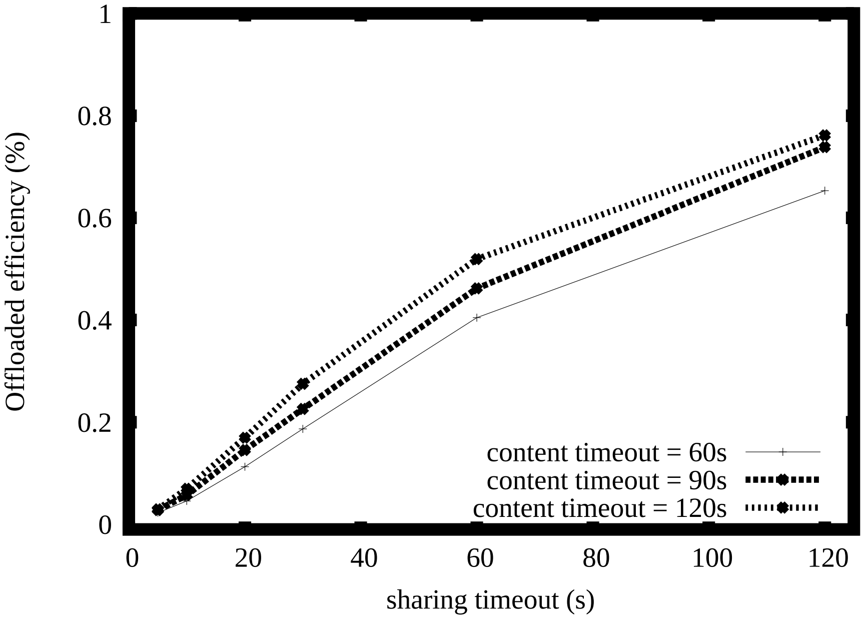


Figure 22. Impact of sharing and content timeout - req_rate=0.5 req/s, M=20

Figure 19 to Figure 22 show the offloading efficiency for different sharing and content timeouts. These results in general confirm our intuition. **The more the content timeout increases, the better**, as content items remain available in the opportunistic network for longer. Similarly, **the longer the sharing timeout the better**, for the very same reason. It is also interesting to note that there is typically a marginal incremental utility in increasing the sharing timeout, as curves tend to flatten out after a certain value. On the other hand, the trend for the content timeout is more variable from this standpoint, and depends on the value of the sharing timeout that is considered.

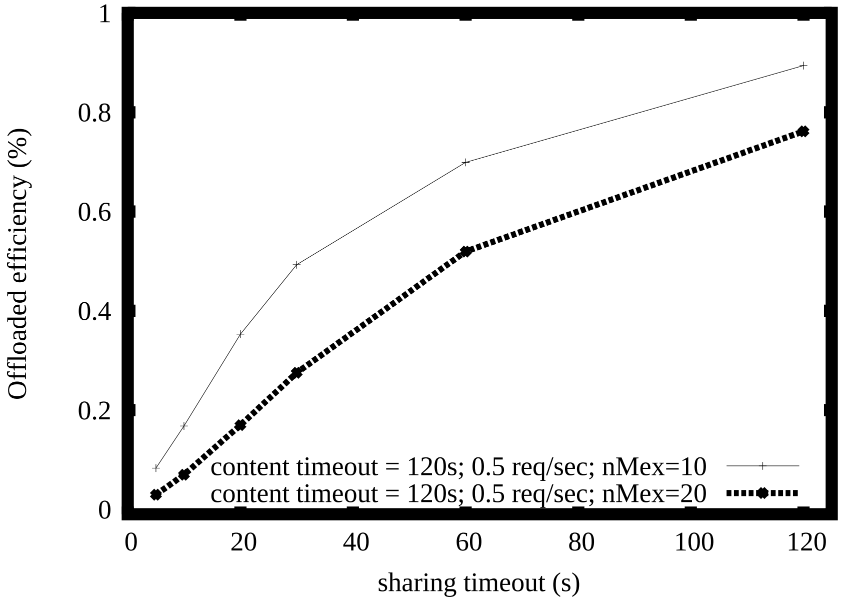


Figure 23. Comparison for different content items - req_rate=0.5 req/s, content timeout 120s

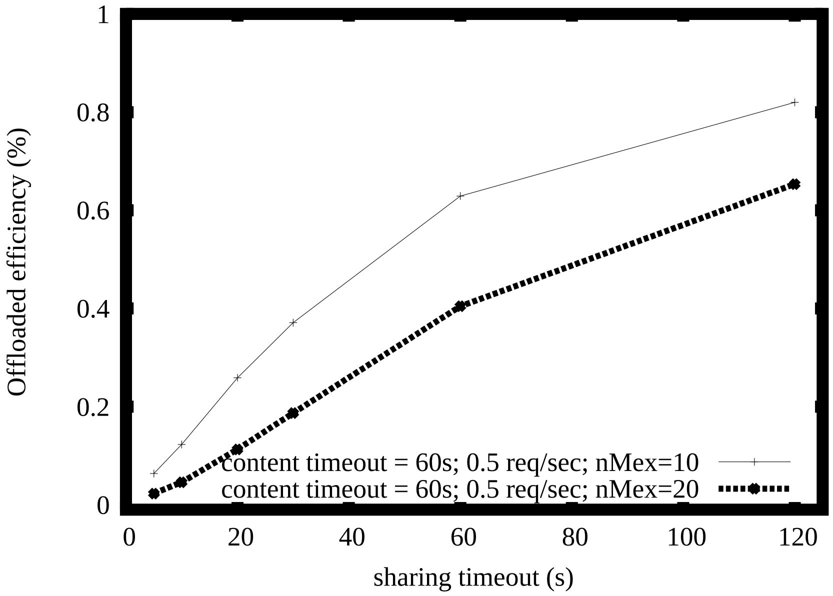


Figure 24. Comparison for different content items - req_rate=0.5 req/s, content timeout 60s

Figure 23 and Figure 24 allows us to appreciate the impact of the number of content items requested by nodes. Specifically, when more content items are exchanged (M=20) they are exchanged less easily in the opportunistic network, due to the contention among multiple contents for possible exchange during contact events. These results, in general, in a decreasing offloading efficiency. As expected, the more time is provided for content items to disseminate the better, i.e. we observe higher efficiencies for longer content and sharing timeouts.

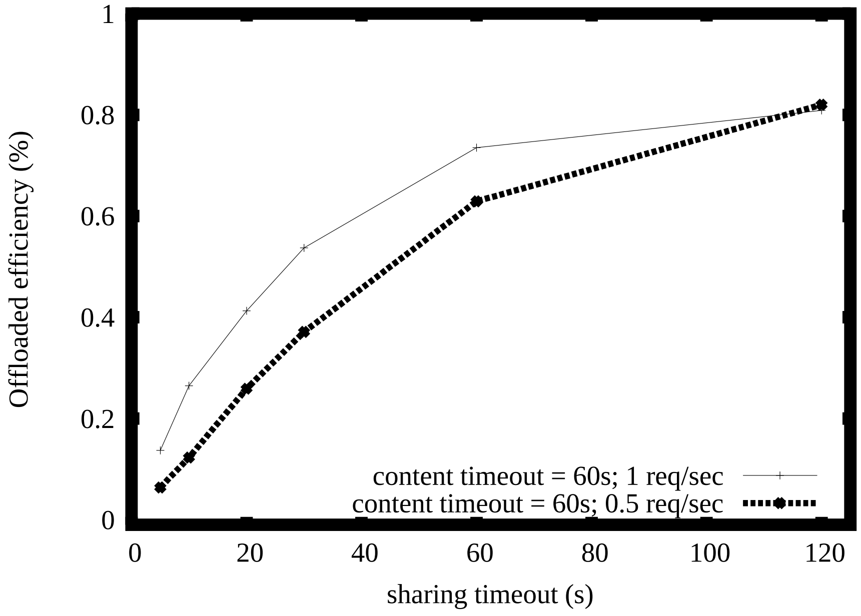


Figure 25. Comparison for different request rates – M=10, content timeout 60s

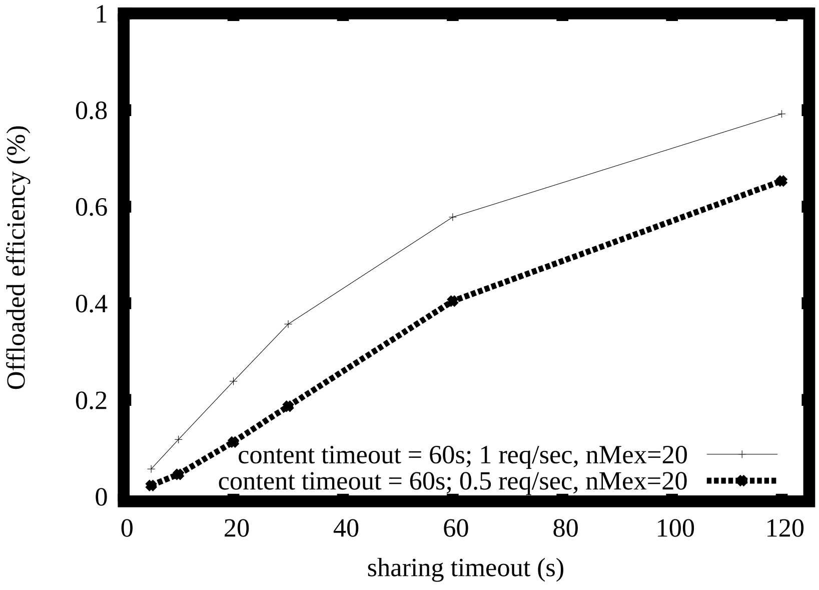


Figure 26. Comparison for different request rates – M=20, content timeout 60s

Figure 25 and Figure 26 show the comparison with respect to the request rate. Specifically, these results show interesting features, which is counterintuitive at a first impression. Figure 25 shows that, unless for the case of sharing timeout equal to 120s, higher request rates result in higher efficiency. This is intuitive, because higher request rates result in requests being more concentrated in time. When nodes share the content items only for very shorter amounts of time, concentrating the requests in time increases the probability of encountering other nodes sharing the content. The behaviour for large sharing timeouts is less intuitive, where higher request rates results in lower offloading efficiency. The reason of this will be clearer when analysing the evolution of dissemination over time. Intuitively, when requests are more concentrated in time, content timeouts for nodes that do not get the content via the opportunistic network are also more concentrated. As we will discuss later, when a timeout expires and content is delivered via the cellular network, this kicks off a fast increase in the dissemination of content via the opportunistic network in the region of the node whose content timeout has expired. When expirations are less concentrated in time (i.e., when request rates are lower), the opportunistic diffusion process has more time to spread content, and therefore the offloading efficiency increases. Note that this behaviour does not appear in the case of M=20, because the dissemination process is slowed down by the higher contention in the opportunistic network.

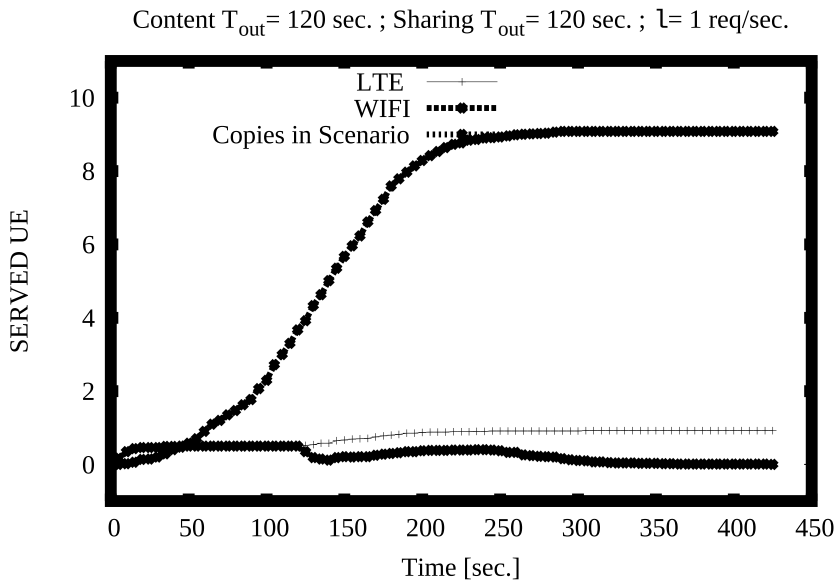


Figure 27. Temporal evolution: long sharing timeouts

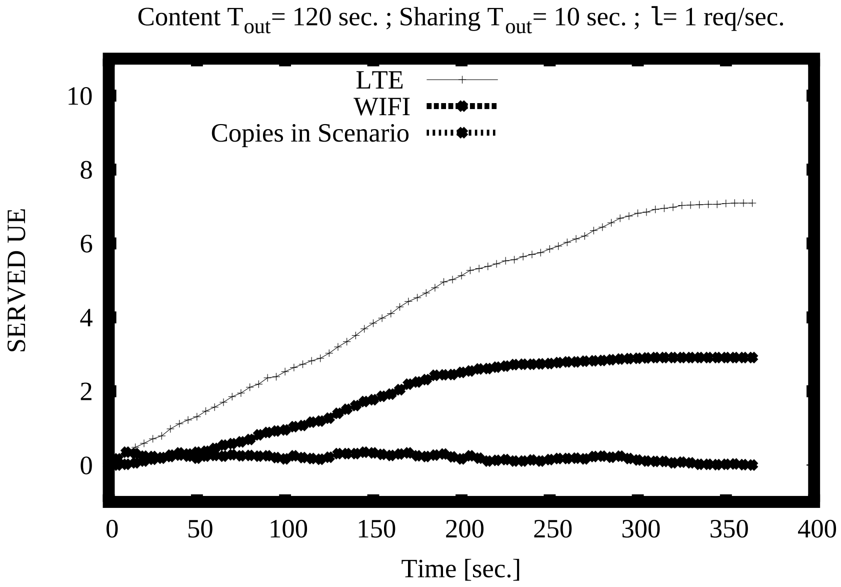


Figure 28. Temporal evolution: short sharing timeouts

Finally, Figure 27 and Figure 28 show the temporal evolution of content diffusion in two relevant cases, i.e. that of long and short sharing timeouts, respectively. In the first case, the first nodes that request some content receive it via LTE. Then, nothing happens (but opportunistic diffusions) until the first content timeouts elapse, i.e. around 120s (during this phase no content timeout has expired, and the CDM knows that there are copies in the opportunistic network due to the long sharing timeout, and therefore no additional transmissions occur over LTE). After that point in time, a few content timeout expire, and content items are served through LTE (the corresponding curve increases). This boosts the opportunistic dissemination process, which results in a marked increase of the number of nodes receiving content items via the opportunistic network. On the other hand, for short sharing timeouts nodes receive content via LTE more uniformly over time. This is because content items stay available in the opportunistic network for short amounts of time. Even during the first 120s (while content timeouts have not expired yet), the CDM knows that some content requests can only be satisfied via the LTE network, because no nodes are storing those items anymore (because their sharing timeout is over).

We have consistently observed this behaviour also for the other sets of parameters.

3.2.1.8 Simulation conclusions

In conclusion, we can say that all our hypotheses are confirmed by simulation results. In addition, we have observed a non-intuitive (but reasonable) dependence between offloading efficiency and request rates for long sharing timeouts: in this case, high request rates result in a concentration of expirations of content timeouts, and a significant number of panic-zone injections, whereas lower request rates give more time to the opportunistic dissemination to diffuse, eventually resulting in higher offloading efficiency. On the other hand, for short sharing timeouts, higher request rates help keeping more copies of the content “alive” in mobile nodes caches, thus resulting in higher offloading efficiency (with respect to the case of lower request rates). The analysis of the evolution of content diffusion over time allows us to track more closely the status of the dissemination, and understand the detailed mechanisms of the offloading algorithm.Overall; these results confirm that **offloading can be very efficient also for non-synchronised content requests**, if the algorithms parameters are tuned appropriately for the considered environment.

3.2.2 Vehicular Scenario 2

3.2.2.1 *Target*

The simulation of the **Map-Based Advanced Driver Assistance Systems** use case scenario is aimed at validating the effectiveness of the proposed offloading architecture in a vehicular environment. In fact, the communication channel available in such scenario is characterized by severe multipath and Doppler conditions, which could affect the performances of the whole system. Moreover, the mobility patterns of the vehicles are also quite different with respect to the ones already considered in the pedestrian scenarios. The simulations executed in this task will be therefore useful for evaluating the behaviour of the proposed mobile opportunistic traffic offloading techniques.

The rest of the chapter is organized as follows. Section 3.2.2.2 provides a description of the process used for preparing the scenario in terms of required input. The expected high-level results are described in section 3.2.2.3. The input and output parameters are listed in sections 3.2.2.4 and 3.2.2.5, respectively. The obtained results are provided in section 3.2.2.6 and the conclusions are drawn in section 3.2.2.7.

3.2.2.2 *Layout and description*

Figure 29 shows the activities (both automatic and user-driven) conducted for preparing all the inputs (e.g., map, vehicle mobility, configuration files, and input parameter values) required for running simulations on the vehicular scenario.

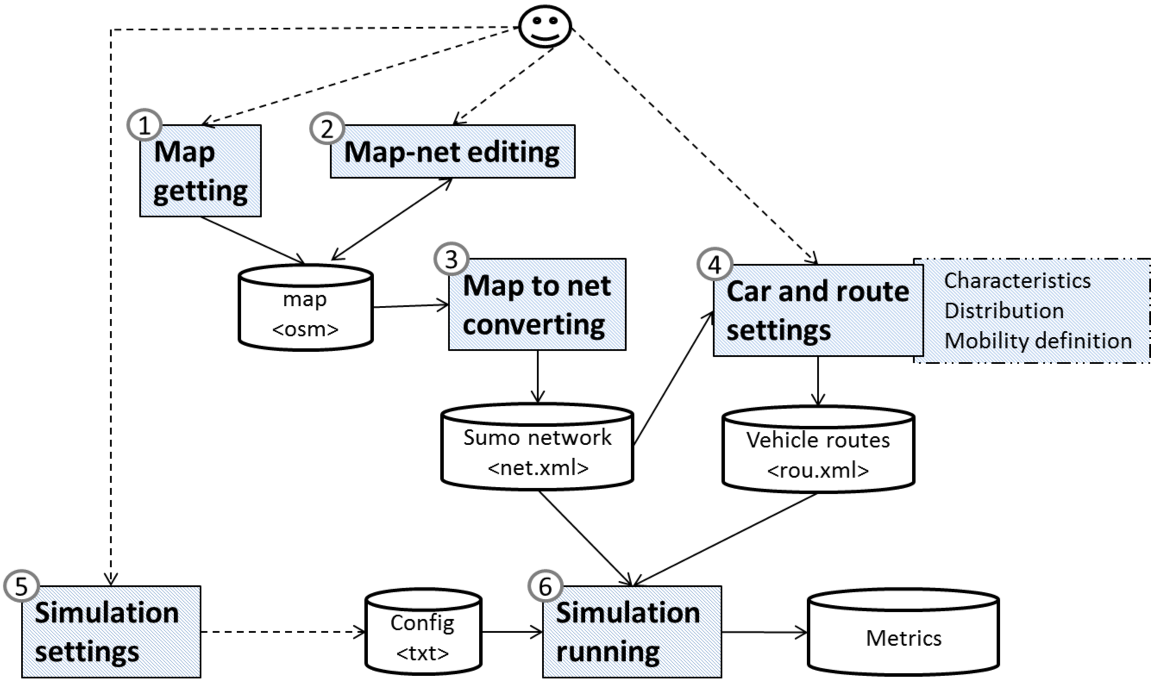
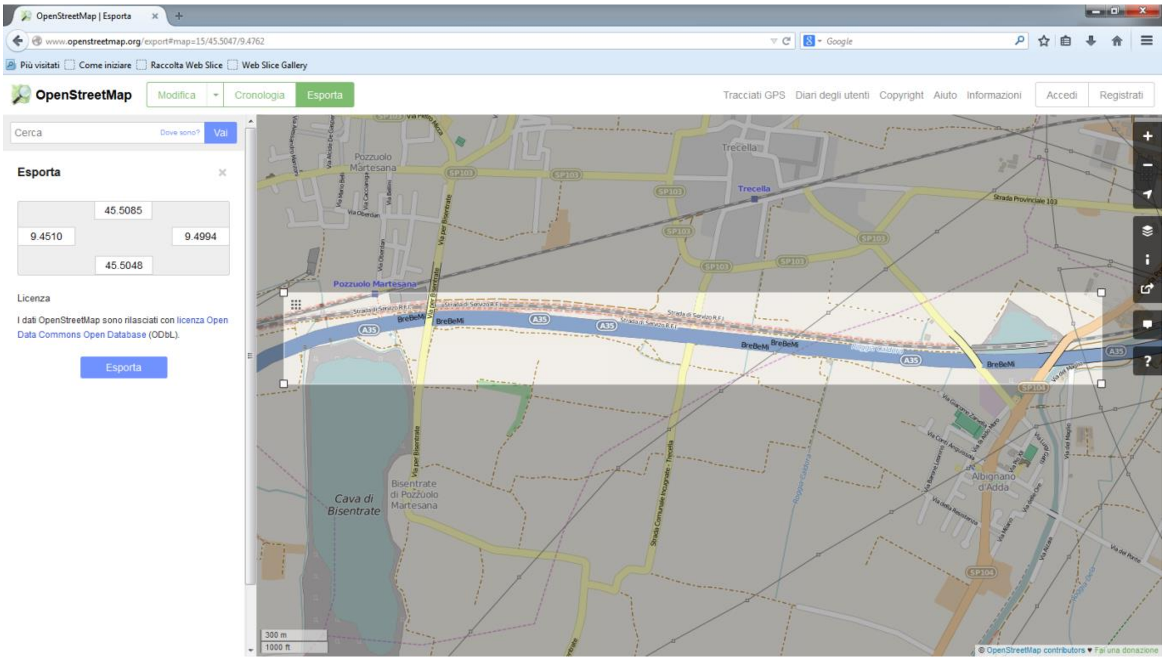


Figure 29: Process adopted

The rest of this section will detail and analyse each conducted activity.

1. ***Map getting***. To get the map of the area of interest in the OpenStreetMap web site (https://www.openstreetmap.org). This step has been manually executed by interacting with the OpenStreetMap web site and using the "Export" facility provided by the web site. By means of this facility, a user can locally download an OSM-based representation of the map of interest. OSM is an XML-based format adopted by OpenStreetMap to represent maps and define their details. Figure 30 presents a screenshot of the OpenStreetMap web site showing a portion of the BreBeMi Italian motorway, also known as "A35" in the Italian motorways official system. In particular, the area selected in the map shown in Figure 30 details the map of interest for the scenario under

simulation, that is an extent of almost 10Km of the BreBeMi (http://www.brebemi.it) motorway. BreBeMi is a very recently built and released motorway that connects Brescia and Milano, two cities in the north of Italy.



Esporta

	45.5085	
9.4510		9.4994
	45.5048	

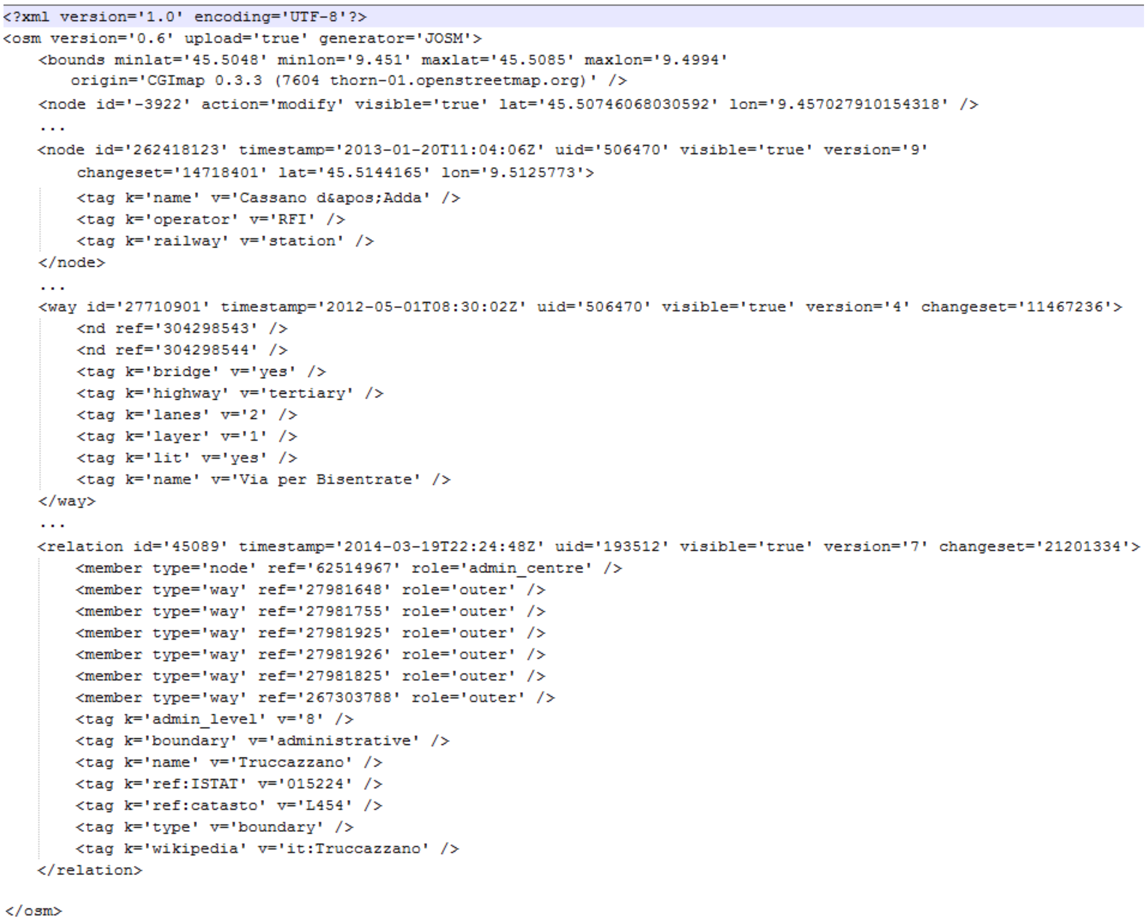
Licenza

I dati OpenStreetMap sono rilasciati con licenza Open Data Commons Open Database (ODbL).

Esporta

Figure 30: The map of the scenario in the OpenStreetMap web site

Figure 31 shows a fragment of the OSM file (http://wiki.openstreetmap.org/wiki/OSM_XML) representing the map downloaded from the OpenStreetMap web site for the scenario under simulation. The sketch of file in Figure 31 contains the typical structure of an OSM file; it is composed of: (1) a set of nodes (tag "node") describing geographical locations expressed into the WGS84 reference system; (2) a set of blocks of ways/streets (tag "way"), i.e., ordered lists of nodes; and (3) a set of relations among nodes (tag "relation"), streets in the maps and other relations (e.g., junctions).

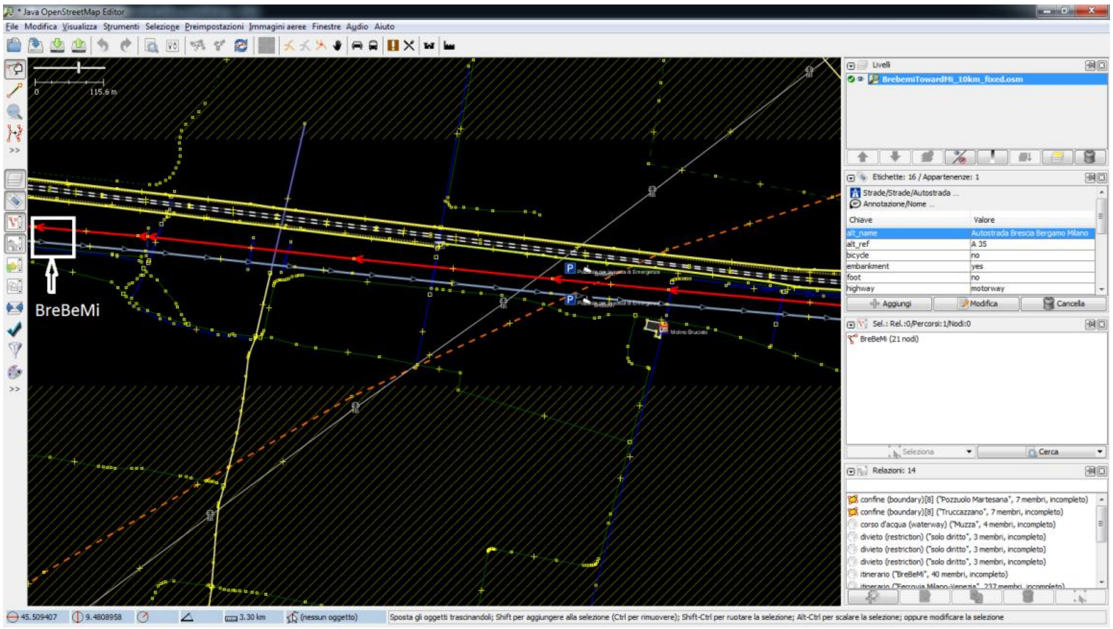


```
<?xml version='1.0' encoding='UTF-8'?>
<osm version='0.6' upload='true' generator='JOSM'>
    <bounds minlat='45.5048' minlon='9.451' maxlat='45.5085' maxlon='9.4994'
       origin='CGImap 0.3.3 (7604 thorn-01.openstreetmap.org)' />
    <node id='-3922' action='modify' visible='true' lat='45.50746068030592' lon='9.457027910154318' />
    ...
    <node id='262418123' timestamp='2013-01-20T11:04:06Z' uid='506470' visible='true' version='9'
        changeset='14718401' lat='45.5144165' lon='9.5125773'>
        <tag k='name' v='Cassano d&apos;Adda' />
        <tag k='operator' v='RFI' />
        <tag k='railway' v='station' />
    </node>
    ...
    <way id='27710901' timestamp='2012-05-01T08:30:02Z' uid='506470' visible='true' version='4' changeset='11467236'>
        <nd ref='304298543' />
        <nd ref='304298544' />
        <tag k='bridge' v='yes' />
        <tag k='highway' v='tertiary' />
        <tag k='lanes' v='2' />
        <tag k='layer' v='1' />
        <tag k='lit' v='yes' />
        <tag k='name' v='Via per Bisentrate' />
    </way>
    ...
    <relation id='45089' timestamp='2014-03-19T22:24:48Z' uid='193512' visible='true' version='7' changeset='21201334'>
        <member type='node' ref='62514967' role='admin_centre' />
        <member type='way' ref='27981648' role='outer' />
        <member type='way' ref='27981755' role='outer' />
        <member type='way' ref='27981925' role='outer' />
        <member type='way' ref='27981926' role='outer' />
        <member type='way' ref='27981825' role='outer' />
        <member type='way' ref='267303788' role='outer' />
        <tag k='admin_level' v='8' />
        <tag k='boundary' v='administrative' />
        <tag k='name' v='Truccazzano' />
        <tag k='ref:ISTAT' v='015224' />
        <tag k='ref:catasto' v='L454' />
        <tag k='type' v='boundary' />
        <tag k='wikipedia' v='it:Truccazzano' />
    </relation>

</osm>
```

Figure 31: Fragment of an OSM-based file

2. ***Map-net editing***. To edit the map downloaded from OpenStreetMap and fix issues (e.g., wrong streets or wrong information contained in the map) and incompleteness (e.g., missing streets) that often persist in existing maps. This step has been manually executed by means of OSM-map editor such as the Java OpenStreetMap Editor - JSOM (http://wiki.openstreetmap.org/wiki/JOSM). JSOM is an editor of OSM files, thus it allows a user to check every detail of a map and change it. In our case, we used JSOM to have the chance of checking the map of the BreBeMi motorway and fixing mistakes and wrong information, before using the map in our simulation runs. Figure 32 shows a screenshot of the JSOM interface where the OSM map of the scenario under simulation has been (pre-) loaded. In the main canvas of the JSOM interface, we can observe a fragment of the motorway map of the scenario. Moreover, boxes in the right-canvas of the tool interface are used to give detailed information (e.g. name, compositions, attributes, closest objects) about the map and the objects it contains that are currently selected.


* Java OpenStreetMap Editor
File Modifica Visualizza Strumenti Selezione Preimpostazioni Immagini aeree Finestre Audio Aiuto
0
115.6 m
BreBeMi
Molino Bruciato
Livelli
BrebemiTowardMi_10km_fixed.osm
Etichette: 16 / Appartenenze: 1
Strade/Strade/Autostrada ...
Annotazione/Nome ...
Aggiungi
Modifica
Cancella
Sel.: Rel.:0/Percorsi:1/Nodi:0
Seleziona
Cerca
Relazioni: 14

Chiave	Valore
alt_name	Autostrada Brescia Bergamo Milano
alt_ref	A 35
bicycle	no
embankment	yes
foot	no
highway	motorway

BreBeMi (21 nodi)

confine (boundary)[8] ("Pozzuolo Martesana", 7 membri, incompleto)
confine (boundary)[8] ("Truccazzano", 7 membri, incompleto)
corso d'acqua (waterway) ("Muzza", 4 membri, incompleto)
divieto (restriction) ("solo diritto", 3 membri, incompleto)
divieto (restriction) ("solo diritto", 3 membri, incompleto)
divieto (restriction) ("solo diritto", 3 membri, incompleto)
itinerario ("BreBeMi", 40 membri, incompleto)
itinerario ("Ferrovia Milano-Venezia", 237 membri, incompleto)

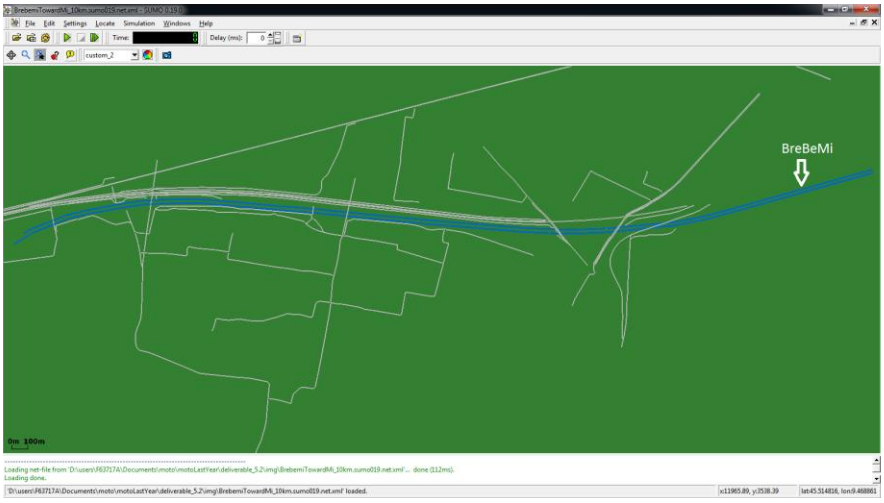
45.509407 | 9.4808958 | 3.30 km | (nessun oggetto)

Sposta gli oggetti trascinandoli; Shift per aggiungere alla selezione (Ctrl per rimuovere); Shift-Ctrl per ruotare la selezione; Alt-Ctrl per scalare la selezione; oppure modificare la selezione

Figure 32: Screenshot of the JSOM OpenStreetMap editor

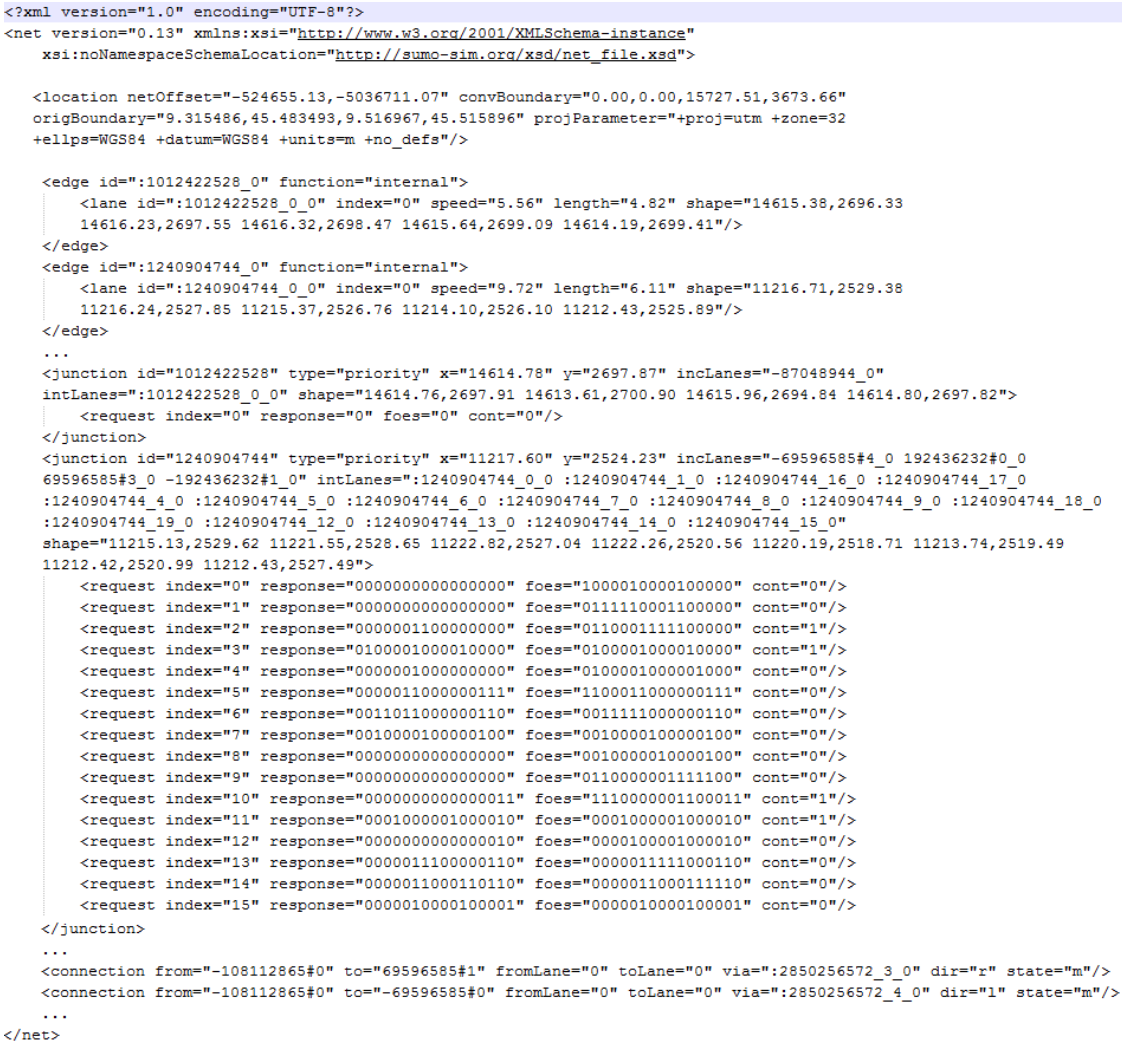
3. ***Map to net converting***. To convert the OSM-based map obtained from the OpenStreetMap web site into an XML format that is automatically readable by the Sumo simulator. This step has been automatically executed by means of *netconvert* (http://sumo.dlr.de/wiki/Networks/Import/OpenStreetMap), a tool utility natively distributed together with the Sumo simulator and that lets us obtain a network representation of the OSM-based map that is usable by both Sumo and MOTO simulation tool. Figure 33 shows a screenshot of the Sumo-gui viewer that shows an overall view of the network recovered from the BreBeMi (OSM-based) map by means of the netconvert utility. The figure shows a set of geometrical objects representing streets and objects of the map that are recognized and usable in simulation runs. In Figure 33, the main street crossing the map represents the BreBeMi motorway, the context of our scenario. Figure 34 shows a fragment of the XML file that represents the network recovered from the BreBeMi map using netconvert and that is visualized by the Sumo-gui in Figure 33. The structure of such a type of file is mainly composed of:
   a. locations (tag "location"), defining the geographical coordinates of the location in the map;
   b. edges (tag "edge"), defining streets and fragment of streets in terms of, e.g., source and destination nodes, lanes, flows direction, maximum allowed speed;
   c. lanes (tag "lane"), characterizing each edge in terms of lanes to be crossed by vehicles;
   d. junctions (tag "junction"), defining points of composition for streets based on edges;
   e. connections (tag "connection"), defining all the connections among map objects (e.g., streets in terms of edges and junctions);
   f. Traffic light logic (tag "tlLogic"), defining the basic logic of traffic light presents in the map.

   While netconvert mainly focuses on objects composing streets and street compositions, another interesting tool utility provided by the Sumo simulator is *polyconvert*, it allows us to extract from an OSM-based map any information about geometrical shapes such as polygons and points of interest (e.g., buildings, parking garages, and shops) contained in the map.


BrebemiTowardMi_10km.sumo019.net.xml - SUMO 0.19.0
File Edit Settings Locate Simulation Windows Help
Time:
Delay (ms): 0
custom_2
BreBeMi
0m 100m
x:11965.89, y:3538.39
lat:45.514816, lon:9.468861

Loading net-file from 'D:\users\F63717A\Documents\moto\motoLastYear\deliverable_5.2\img\BrebemiTowardMi_10km.sumo019.net.xml'... done (112ms).
Loading done.
'D:\users\F63717A\Documents\moto\motoLastYear\deliverable_5.2\img\BrebemiTowardMi_10km.sumo019.net.xml' loaded.

Figure 33: Sumo network of the scenario visualized by the Sumo-gui



```
<?xml version="1.0" encoding="UTF-8"?>
<net version="0.13" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="http://sumo-sim.org/xsd/net_file.xsd">

   <location netOffset="-524655.13,-5036711.07" convBoundary="0.00,0.00,15727.51,3673.66"
   origBoundary="9.315486,45.483493,9.516967,45.515896" projParameter="+proj=utm +zone=32
   +ellps=WGS84 +datum=WGS84 +units=m +no_defs"/>

    <edge id=":1012422528_0" function="internal">
        <lane id=":1012422528_0_0" index="0" speed="5.56" length="4.82" shape="14615.38,2696.33
        14616.23,2697.55 14616.32,2698.47 14615.64,2699.09 14614.19,2699.41"/>
    </edge>
    <edge id=":1240904744_0" function="internal">
        <lane id=":1240904744_0_0" index="0" speed="9.72" length="6.11" shape="11216.71,2529.38
        11216.24,2527.85 11215.37,2526.76 11214.10,2526.10 11212.43,2525.89"/>
    </edge>
    ...
    <junction id="1012422528" type="priority" x="14614.78" y="2697.87" incLanes="-87048944_0"
    intLanes=":1012422528_0_0" shape="14614.76,2697.91 14613.61,2700.90 14615.96,2694.84 14614.80,2697.82">
        <request index="0" response="0" foes="0" cont="0"/>
    </junction>
    <junction id="1240904744" type="priority" x="11217.60" y="2524.23" incLanes="-69596585#4_0 192436232#0_0
    69596585#3_0 -192436232#1_0" intLanes=":1240904744_0_0 :1240904744_1_0 :1240904744_16_0 :1240904744_17_0
    :1240904744_4_0 :1240904744_5_0 :1240904744_6_0 :1240904744_7_0 :1240904744_8_0 :1240904744_9_0 :1240904744_18_0
    :1240904744_19_0 :1240904744_12_0 :1240904744_13_0 :1240904744_14_0 :1240904744_15_0"
    shape="11215.13,2529.62 11221.55,2528.65 11222.82,2527.04 11222.26,2520.56 11220.19,2518.71 11213.74,2519.49
    11212.42,2520.99 11212.43,2527.49">
        <request index="0" response="0000000000000000" foes="1000010000100000" cont="0"/>
        <request index="1" response="0000000000000000" foes="0111110001100000" cont="0"/>
        <request index="2" response="0000001100000000" foes="0110001111100000" cont="1"/>
        <request index="3" response="0100001000010000" foes="0100001000010000" cont="1"/>
        <request index="4" response="0000001000000000" foes="0100001000001000" cont="0"/>
        <request index="5" response="0000011000000111" foes="1100011000000111" cont="0"/>
        <request index="6" response="0011011000000110" foes="0011111000000110" cont="0"/>
        <request index="7" response="0010000100000100" foes="0010000100000100" cont="0"/>
        <request index="8" response="0000000000000000" foes="0010000010000100" cont="0"/>
        <request index="9" response="0000000000000000" foes="0110000001111100" cont="0"/>
        <request index="10" response="0000000000000011" foes="1110000001100011" cont="1"/>
        <request index="11" response="0001000001000010" foes="0001000001000010" cont="1"/>
        <request index="12" response="0000000000000010" foes="0000100001000010" cont="0"/>
        <request index="13" response="0000011100000110" foes="0000011111000110" cont="0"/>
        <request index="14" response="0000011000110110" foes="0000011000111110" cont="0"/>
        <request index="15" response="0000010000100001" foes="0000010000100001" cont="0"/>
    </junction>
    ...
    <connection from="-108112865#0" to="69596585#1" fromLane="0" toLane="0" via=":2850256572_3_0" dir="r" state="m"/>
    <connection from="-108112865#0" to="-69596585#0" fromLane="0" toLane="0" via=":2850256572_4_0" dir="l" state="m"/>
    ...
</net>
```

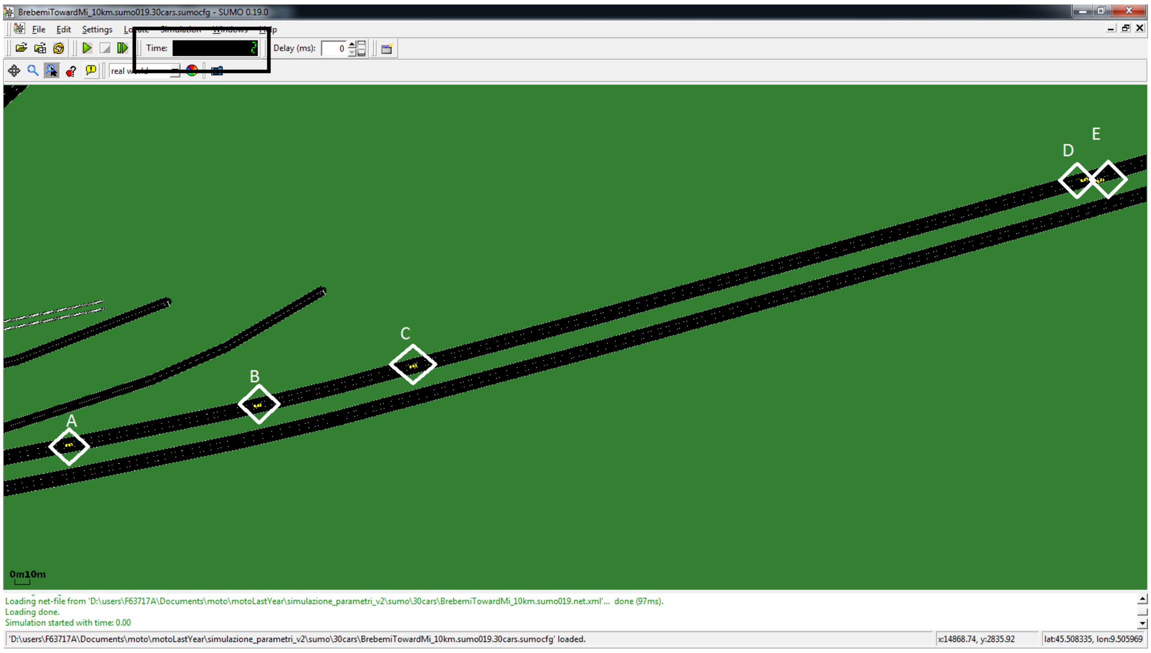
Figure 34: A fragment of an example of Sumo network

4. ***Car and route setting***. To characterize vehicles and their routes in the simulation network. Indeed, given a network representation of an OpenStreetMap map, Sumo provides a way to characterize

each vehicle (defining vehicle type and characteristics) within the simulation scenario (http://sumo.dlr.de/wiki/Definition_of_Vehicles,_Vehicle_Types,_and_Routes) and provides tool utilities, e.g., *randomTrip.py* and *duarouter*, to generate trips and routes for vehicles in a Sumo network (http://sumo.dlr.de/wiki/Demand/Introduction_to_demand_modelling_in_SUMO).

These tool utilities to generate vehicle's trips and routes allowed us to generate flows of vehicles moving in the map, thus simulating the actual movement of vehicles in the area used as context of the scenario simulation. In our case, we generated three initial flows of vehicles (which size ranges in the sets of: 30, 90, and 150 vehicles) crossing the motorway in only one direction (i.e., in the direction from Brescia to Milano). No other vehicles have been added to the scenario to avoid any kind of noise, wrong behaviours and, in particular, to speed-up the execution of the simulation by limiting the presence of non-useful vehicles or objects, that are those vehicles and objects not affecting into the simulation itself. The vehicles composing the three flows of our scenario have been randomly distributed in the full length of the considered roadway by paying attention to preserve a realistic vehicle density (i.e., number of cars per kilometre of street) for a typical Italian motorway scenario, i.e., by avoiding strong congestions as well as totally-free scenarios. In other terms, while the destination point of each vehicle trip/route is represented by the end of the motorway in the considered map, the source point of each trip/route is randomly defined at the beginning of the simulation by considering the full length of the motorway, thus realistically distributing vehicles in the 10 Km of motorway. Figure 35 presents a fragment of a Sumo screenshot showing how 5 vehicles (here named A - E) have been distributed in the motorway network at beginning of the simulation (see in the figure the timestep equal to 2 – out of the considered 200). We can observe that vehicles have different distances among them, for example, between A and B we have around 130 mt, between B and C about 110 mt, between C and D about 430 mt, and between D and E only few mt. Clearly, an increase in the number of vehicles of the scenario (e.g., from 30 to 150), increases also the density of vehicles per kilometre: on average, in our simulation runs, we consider a variable density ranging from 3 to 15 vehicles per kilometre.

Sumo lets us adopt a microscopic mobility modelling for moving vehicles in a network of streets. A microscopic mobility model is a way of flow modelling in which the flow of vehicles is simulated by simulating the movement of every single vehicle on the street, thus assuming that, on the one hand, (i) an overall flow of vehicles can be modelled by modelling the behaviour of each vehicle, while, on the other hand, (ii) the behaviour of each vehicle depends on a set of low-level vehicle characteristics that have to be defined (e.g., vehicle acceleration and deceleration, driver behaviour, vehicle's length, vehicle speed). Sumo implements (http://sumo.dlr.de/wiki/Theory/Traffic_Simulations) different microscopic mobility modelling approaches: e.g., Krauss mobility model, Intelligent driver model, Kerner's three-phase model, and Wiedemann model. However, it is worthy notice that the Krauss mobility model is the Sumo-native one, the most complete and representative mobility model provided by Sumo, other models are supported with a quite incomplete or third party implementation. By means of the adopted mobility model and of characteristics of the built network of streets, Sumo can control movements of vehicles in the network, however, customization of vehicle's characteristics and movements are still possible by changing the set of XML files that contain the information about vehicles and routes statically (e.g., manually before running the simulation) or dynamically (e.g., during the simulation run, by means of telecommunication technology), if needed.



```
Loading net-file from 'D:\users\F63717A\Documents\moto\motoLastYear\simulazione_parametri_v2\sumo\30cars\BrebemiTowardMi_10km.sumo019.net.xml'...  done (97ms).
Loading done.
Simulation started with time: 0.00
'D:\users\F63717A\Documents\moto\motoLastYear\simulazione_parametri_v2\sumo\30cars\BrebemiTowardMi_10km.sumo019.30cars.sumocfg' loaded.    x:14868.74, y:2835.92    lat:45.508335, lon:9.505969
```

Figure 35: Example of car distribution on the Sumo network

5. ***Simulation settings***. To configure the simulation runs to be executed. In the previous activities, we built the map of the scenario and the related Sumo network that are adopted in the simulation, and we defined sets of distribution and mobility of vehicles crossing the motorway network of interest. Hence, the simulation parameters have to be defined before being ready for running any simulation. For example we have to define the following variables:

   a. number of seeds (vehicle receiving the content by the LTE communication technology);

   b. number and position of the eNB node/s (in this simulation scenario we considered only one eNB node positioned closed to the BreBeMi roadway and in the middle of the considered roadway fragment, i.e., 5Km from the beginning of the considered roadway);

   c. type and size of the content to be distributed (geographical map information of different size);

   d. adopted Wi-Fi standard;

   e. propagation models of the Wi-Fi physical channel;

   f. who- and when- strategy of the offloading algorithm.

   See the next Section 3.2.2.4for details about the whole list of parameters to be selected for running the simulation.

6. ***Simulation running***. The simulation has been executed several times by considering several combinations of the parameter values. For running the simulations, we used a quite powerful workstation having a 12-core CPU and 24GB of RAM memory, nevertheless each simulation run required some days since it is often heavy in terms of resources and time consuming in terms of computation time.

3.2.2.3 Expected results

With this simulation, the following high-level results are expected to be obtained:

- **% UEs reached by Wi-Fi**: this metric provides the percentage of user equipment (UEs) reached by Wi-Fi, giving therefore an indication of the effectiveness of the offloading process. In the considered scenario, the IEEE 802.11p Wi-Fi standard is used only for vehicle-to-vehicle (V2V) communication without involving any vehicle-to-infrastructure (V2I) transmission or WAN access. In fact, the higher this metric, the better is the collaboration among users for disseminating the content of interest without passing through the cellular LTE network. This metric is expected to increase with the density of UEs available in a certain area, thanks to the increased collaboration enabled among users. Moreover, its absolute values should increase when the dimension of the offloaded content decrease (i.e., the smaller the content, the better the limited contact time between vehicles can be exploited).

- **Number and percentage of bytes transmitted per interface**: these metrics provide the distribution of transmitted bytes over LTE and Wi-Fi interfaces in terms of total numbers and corresponding percentages, respectively. The number of bytes transmitted should intuitively increase with the dimension of the offloaded content for both interfaces. Moreover, following the considerations done for the previous metric, the LTE involvement in terms of percentage should also increase with the content size, mainly because the effectiveness of the offloading processes is reduced.
- **Number and percentage of bytes transmitted on Wi-Fi for DATA and CONTROL traffic**: these metrics analyse the traffic passing through the Wi-Fi interface in terms of DATA and CONTROL bytes. The absolute number of CONTROL bytes should be independent from the content size. The DATA traffic passing through the Wi-Fi interface should instead increase with the dimension of the content to be disseminated. This behaviour in terms of absolute values is then reflected accordingly into percentage values.
- **Total delay from first inject time per interface**: this metric provides the total delay from first inject time split up for interface time. The value related to the Wi-Fi interface should therefore increase in absolute value when the usage of this channel increases.
- The linear when-strategy should provide better results than the initial one in terms of offloading ratio, particularly when the number of UEs is low.

3.2.2.4 Simulation parameters

This section describes the parameters defined for the simulation and their value variability considered in the simulation runs in order to comply with the simulation requirements. The following four groups of parameters have been identified and defined for the simulation:

- Parameters related to the characterization of the scenario under simulation.
- Parameters related to the adopted offloading algorithm.
- Parameters related to the adopted short-range (Wi-Fi) communication technology.
- Parameters related to the adopted long-range (LTE) communication technology.

In the following paragraphs, we will describe and analyse each type of parameters.

Parameters related to the characterization of the simulated scenario. Table 6 details all the considered parameters that are strongly related to the simulated scenario, which are those parameters that are mainly considered in the simulation for characterizing the context of the scenario under simulation.

In terms of nodes involved in the simulation, we considered:

(i) only one Evolved Node B (eNB) node positioned close to the BreBeMi roadway and in the middle of the considered fragment of motorway (i.e., 5Km from the beginning of the roadway);

(ii) three sets of nodes composed of: 30, 90 and 150 vehicles;

(iii) for each set of vehicles, we consider 10% and 30% of them, as initially (i.e., before the panic zone) seeded by means of the LTE (long-range) communication technology.

As content to be distributed in the scenario, we adopted textually encoded maps with geographical information. We also considered in the simulation runs different size (i.e., 1024, 10240 and 50240 B) of such a content. As already said, the area of interest for the simulation scenario consists on 10Km of an Italian motorway named BreBeMi and we populated it with sets of vehicles randomly distributed by the Sumo tool utilities (see Section 3.2.1.2) over only one direction of the 10Km. The adopted vehicle mobility model is the Krauss model implemented by Sumo, in which each vehicle has a variable speed that ranges from 25 to 180 Km/h when crossing the BreBeMi. As simulation time, a time range from 3.3 to 4.1 minutes is considered in our simulation runs for what concerns the execution of the scenario in Sumo (i.e., without considering the underlying communication technology).

Parameter	Variable/fixed	Units	Min	Max
Number of nodes (vehicles)	Variable	Number	30	150
Number of seeds	Variable	Number	3	45
Number of eNBs	Fixed	Number	1	
Type of content	Fixed	Video, photo, text	Text	
Size of content	Variable	Bytes	1024	50240
Type of the street	Fixed	Motorway, urban, rural, extra-urban and secondary	Motorway	
Street length	Fixed	Km	(about) 10	
Number of lanes	Fixed	Number (typically 2-6)	3	
Road map	Fixed	Simulated, random, manually built, real	Real (OpenStreet - Map)	
Type of node	Fixed	Bus, train, car, motorcycle	Car	
Car speed	Variable	Km/h	25	180
Node mobility model	Fixed	Krauss model, intelligent driver model, Kerner's three-phase model, Wiedemann model	Krauss model	
Initial node distribution	Variable	Random, semi-random, manually	Semi-random	
Node trip/route definition	Fixed	Random, semi-random, manually, map-based, based on real data, Shortest-path bsed, …	Semi-random	
Sumo simulation time	Variable	sec	200	250

Table 6. Simulation parameters of the vehicular scenario 2

Parameter	Variable/fixed	Units	Min	Max
Message lifetime	Fixed	sec	100	
Time to panic zone	Fixed	sec	20	
Frequency of state diffusion	Fixed	sec	20	
Hello message frequency	Fixed	msec	100	
Time of first injection	Fixed	sec	5	

Enable position forwarding	Fixed	Yes / No	No
Who strategy	Fixed	Random	Random
Seed to decide who (randomly) inject	Variable	Number	Positive number
When strategy	Variable	Initial/Linear	Initial and Linear

Table 7: Parameters of the offloading algorithm used in vehicular scenario 2

Parameters related to the adopted offloading algorithm. Table 7 details the parameters related to the adopted offloading algorithm, which are those parameters that are considered in the simulation runs to characterize and evaluate the adopted offloading algorithm in the vehicular scenario. As who-strategy (deciding the strategy to identify the vehicles to be reached by means of the long-range communication technology before the panic zone), we adopted a random strategy driven by a seed that we customized per each run, thus deciding if injecting the same set of vehicles or different sets. The "random" who-strategy randomly selects a subset of nodes among those that have not yet acknowledged reception of the content and sends it to them through the long-range communication technology. These nodes will then act as seeds for dissemination the content to the remaining ones through direct opportunistic communication. As when-strategy (deciding when to inject the content in the vehicles selected by the who-strategy), we adopted in our simulation runs two strategies: initial and linear. The "initial" when-strategy sends the content to all the seeds previously identified by the who-strategy at the beginning of the offloading process. The "linear" when-strategy linearly increases the number of nodes interested by the content injection among the same set of seeds from the beginning of the offloading process up to the panic zone. More details on these who- and when-strategies can be found into the D3.1 project deliverable titled "Initial results on offloading foundations and enablers". The other parameters of the offloading algorithm have been fixed, that is their value has been considered as constant during all the simulation runs. For instance, the time of the first inject has been fixed to 5 sec. to be sure to have enough time to inject the full content on the subset of vehicles selected by the who-strategy to act as seeds for disseminating the content to the other ones through direct opportunistic communication. Similarly, the time to reach the panic zone has been fixed to 20 sec. as well as the frequency of the state diffusion, this time is considered adequate to have all vehicles reached by the distributed content.

Parameter	Variable/fixed	Units	Min	Max
Wi-Fi channel	Fixed	Yans, PhySimWifi, manually-defined, …	Yans	
Wi-Fi standard	Fixed	802.11a, 802.11g, 80211p_CCH, 80211p_SSH	80211p_CCH	
Propagation Delay	Fixed	Random, constant speed	ConstantSpeed	
Propagation Model	Variable	Friis, Three-log distance, Nakagami, Two-ray ground, Jakes	Three-log distance and Nakagami	
TxPowerLevels	Fixed	Number	1	
TxPowerStart	Fixed	dBm	12.5	
TxPowerEnd	Fixed	dBm	12.5	
EnergyDetectionThreshold	Fixed	dbm	-74.5	
CcaMode1Threshold	Fixed	dbm	-77.5	
RxNoiseFigure	Fixed	dB	7	

TxGain	Fixed	dB	1
RxGain	Fixed	dB	1
WifiMac	Fixed	RegularWifiMac, ApWifiMac, StaWifiMac, AdhocWifiMac	AdhocWifiMac
ThreeLogDistancePropagation LossModel :: Distances	Fixed	mt	1, 80, 500
ThreeLogDistancePropagation LossModel :: Exponents	Fixed	Number	1.9, 3.8, 3.8
ThreeLogDistancePropagation LossModel :: ReferenceLoss	Fixed	dB (at 1 mt)	47.86
NakagamiPropagationLoss :: Distances	Fixed	mt	50, 150
NakagamiPropagationLoss :: Exponents	Fixed	Number	1.5, 1
Wifi remote station manager	Fixed	ConstantRateWifiManager, AarfWifiManger, …	ConstantRateWifiManager
Wifi remote station data rate	Fixed	OfdmRate12Mbps, OfdmRate6MbpsBW10MHz, …	OfdmRate6MbpsBW10MHz
Wifi remote station control rate	Fixed	OfdmRate12Mbps, OfdmRate6MbpsBW10MHz, …	OfdmRate6MbpsBW10MHz

Table 8: NS3-based parameters for the 802.11p communication technology

Parameters related to the adopted short-range (Wifi) vehicle-to-vehicle communication technology. Table 8 details the parameters related to the adopted Wifi (short-range) communication technology, that is, those parameters that are considered in the simulation runs to characterize the Wifi communication technology used for the network offloading in the considered vehicular scenario. To simulate the communication in vehicular networks, two basic ingredients are required: (i) communication protocols, and (ii) a communication environment, composed of: a vehicular mobility (see above the parameters related to the characterization of the simulation scenario) and a propagation loss model.

In our simulations, we adopted the ETSI (European Communications Standard Institute) ITS-G5 profile for dedicated short-range communications (DSRC) as it is implemented by NS3. Indeed, NS3 provides the implementation of protocols for communications based on wireless communication at 5.9 GHz and by using the 802.11p standard for the physical layer. In detail, we adopted the Yans Wi-Fi physical channel of NS3, that implements the 802.11p protocol (control channel - CCH and service channel - SCH) by means of: a (1) physical channel layer that implements the 802.11a model standard characterized by several attributes, such as the ones setting the transmission power levels, noise and transmission gain; and a (2) communication channel layer characterized by a propagation loss and delay model.

With the aim of realistically characterizing the simulation of the physical Wi-Fi channel, we considered typical values of its attributes adopted for vehicular-based communication scenarios for setting the: maximum transmission power level (TXPowerStart) and minimal transmission power level (TxPowerEnd), number of transmission power levels (levels between the values of TXPowerStart and TxPowerEnd), as well as the values of reception noise figure (difference between the noise output of the actual receiver and the noise of an ideal receiver having the same characteristics) and of both thresholds for energy detection (energy of a received signal should be higher than this threshold to allow the PHY layer to detect the signal) and for the clear channel assessment model detection (energy of a received signal should be higher than this threshold to allow the PHY layer to declare the CCA BUSY state). We selected, for example, the values listed in Table 8 for TxPowerStart and TxPowerEnd, EnergyDetectionThreshold, CcaMode1Threshold and gain (in both transmission and reception): (i) according to the existing literature; (ii) with the aim of obtaining the typical behaviour of vehicular scenarios, e.g., a short-range (150/200 mt) of communication among vehicles and among vehicles and infrastructure; and (iii) by adopting values that are in the range suggested by the standard. As MAC layer we adopted the NS3 AdhocWifiMac implementation: that is a simple implementation of Wifi MAC that does not perform any kind of specific beacon generation, probing or association. Finally, also the remote station manager and data/control rate have been fixed for all executed simulations considering typical values for vehicular scenarios: 802.11p adopts Orthogonal Frequency Division Multiplexing (OFDM), aiming at preventing inter-symbol interference and inter-carrier interference in multi-path environments.

With the aim of realistically characterizing the simulation of the Wi-Fi communication channel layer, a propagation loss model has to be adopted. Such a model is used to model the performance of a wireless network channel. To this aim, the propagation loss model can compute the signal strength of a wireless transmission at the receiving node/s, thus determining the quality level at which the receiver/s can be reached by a transmission. NS3 provides several implementations of propagation loss models, ranging from: fixed loss models, exponential decay proportional to the distance between node transmitter and receiver/s, models accounting also for ground and fading reflection. We adopted a propagation delay equal to a constant (the speed of light) and we considered two propagation loss models: Three-log distance (deterministic model) and Nakagami (stochastic attenuation model). Three-log distance is a model that assumes exponential path over the distance from sender and receiver as the Log Distance Path Loss (often used in urban scenario simulations), but it additionally applies different factors to the logarithmic path loss for different distance intervals. The Nakagami propagation loss model includes a stochastic fading model in order to account for non-deterministic effects on the communication caused by moving objects. Hence, we ran simulations considering two different combinations of these propagation models: only Three-log distance and Nakagami on top of Three-log distance. The values of the parameters characterizing Three-log distance and Nakagami reported in Table 8 have been chosen because they are recommended in the literature to characterize the use of such propagation loss model in highway-based scenarios for vehicular communications.

Parameter	Variable/fixed	Units	Min	Max
Type of LTE	Fixed	Simple, EPC	EPC	
Dl Bandwidth	Fixed	Resource block (RB)	100 (20MHz)	
Ul Bandwidth	Fixed	RB	100 (20MHz)	
Ue phy :: TxPower	Fixed	dBm	23	
Enb phy :: TxPower	Fixed	dBm	46	
Mac Scheduler	Fixed	FdMtFfMacSheduler, TdMtFfMacScheduler, PfFfMacScheduler,…	PfFfMacScheduler	
Propagation model	Fixed	Friis, Three-log distance, Nakagami, Two-ray ground, Jakes	Friis	
Rrc model	Fixed	Ideal, real	Ideal	
Rrc :: EpsBearerToRlcMapping	Fixed	RlcSmAlways, RlcUmAlways, RlcSmAlways	RlcUmAlways	
Rrc :: SrsPeriodicity	Fixed	msec	160	

Table 9: NS3-based parameters for the LTE communication technology

Parameters related to the adopted long-range (LTE) communication technology. Table 9 details the parameters related to the adopted LTE (long-range) communication technology that is those parameters that are considered in the simulation runs to characterize the LTE communication technology used for the network offloading in the considered vehicular scenario. We adopted the Evolved Packet Core (EPC) architecture of LTE. To configure a realistic radio access network setting, we fixed the frequency division duplex to 20MHz and both the downlink and uplink bandwidth up to 100 resource block (RB). The maximum transmission power for user equipped and evolved nodeB (eNB) nodes have been defined to 23 and 46 dBm respectively. These values of the parameters let us reproduce typical long-range communication radio access network setting when the LTE technology is adopted in vehicular simulation scenarios. We considered the Friis propagation loss model, typically used to model the performance of a wireless network channel by applying a deterministic path loss model that calculates quadratic path loss as it occurs in free space. As a MAC scheduler for the eNB node, we adopted the PfFfMacScheduler implementation of NS3: a proportional fair scheduler that tries to balance between two interests: trying to maximize total wireless network throughput while at the same time allowing all users to receive at least a minimal level of service. Finally, for the eNB node we used the Ideal radio resource control (RRC) protocol provided by NS3 and we fixed the values of two RRC parameters: (i) EpsBearerToRlcMapping equals to RlcUmAlways (segmentation and reassembly of RLC data, RLC header is added, no delivery is guaranteed), with the aim of specifying the unacknowledged mode as the radio link control (RLC) model used for each EPS bearer (i.e., data traffic flow in the LTE network); and (ii) sounding reference signal (SRS) periodicity equals to 160, with the aim of specifying the frequency in which user equipped nodes have to send the SRS signals to the eNB node, that can use it to estimate the channel quality over a wider bandwidth and to frequency selective scheduling over the channel.

3.2.2.5 Metrics

Table 10 summarizes the set of metrics measured in each run of the scenario simulation to characterize the use of the communication technology offloading in the scenario under simulation. In details, the following groups of metrics are computed:

- ***Number and percentage of bytes transmitted per interface*** (i.e., communication technology) detailing the bytes transmitted through each considered communication technology, i.e., long and short range technology.
- ***Number and percentage of bytes transmitted by means of the Wifi communication technology per control and data messages***: detailing the type of traffic transmitted by means of the Wi-Fi for data and for control messages.
- ***Number and percentage of nodes/vehicles reached by Wi-Fi and LTE communication technology***: detailing the communication technology used to reach each vehicle.
- ***Delay from the first inject per interface*** (i.e., communication technology): details the total delay in terms of time required from the first inject per interface (i.e., communication technology).
- ***Content reception time***: detailing the time in which vehicles collect the full distributed content.

Metrics	Units
Number of byte transmitted on Wi-Fi for Data messages	Number of Bytes
Number of byte transmitted on Wifi for Control messages	Number of Bytes
Delay from first injection for LTE	sec
Delay from first injection for Wifi	sec
Percentage of the number byte transmitted for the LTE interface	%

Percentage of the number byte transmitted for the Wifi interface	%
Percentage of nodes reached by means of LTE	%
Percentage of nodes reached by means of Wifi	%
Total number of bytes transmitted by the LTE interface	Number of Bytes
Total number of bytes transmitted by the Wifi interface	Number of Bytes
Content reception time	sec

Table 10: Metrics of the vehicular scenario 2

3.2.2.6 Simulation results

The previously described scenario has been considered for running an intensive simulation campaign with different configurations of parameters, following the range of values presented above.

In total, 72 different simulations have been executed for the target vehicular scenario. It is worth to be noted that each simulation run requires a non-neglected execution time for being completed, ranging from 8 to 46 hours according to the specific used values (with an average value of about 26 hours and a standard deviation of about 14 hours). For this reason, it has not been possible to complete a significant number of repetitions for each set of parameters by the deadline of this document and only a sub-set of them has been identified for getting statistically significant results through the execution of 5 repetitions of each run. These configurations will be identified in the remaining part of this paragraph through the 90% confidence level reported on the corresponding graphs.

The first results have been obtained by considering the linear when-strategy with a total number of UEs equal to 30, 90, and 150. The considered propagation channel includes both attenuation and Nakagami fading model and the size of the offloaded content is equal to 1024 B, 10240 B, and 50240 B. Under these circumstances, the fraction of users, which have been able to obtain the content through the Wi-Fi interface, is reported in Figure 36 for the different considered sizes.

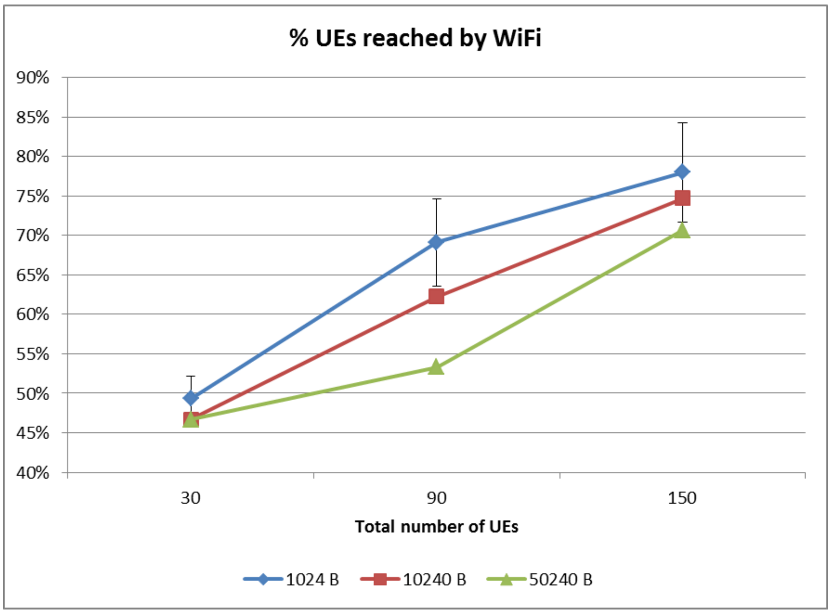
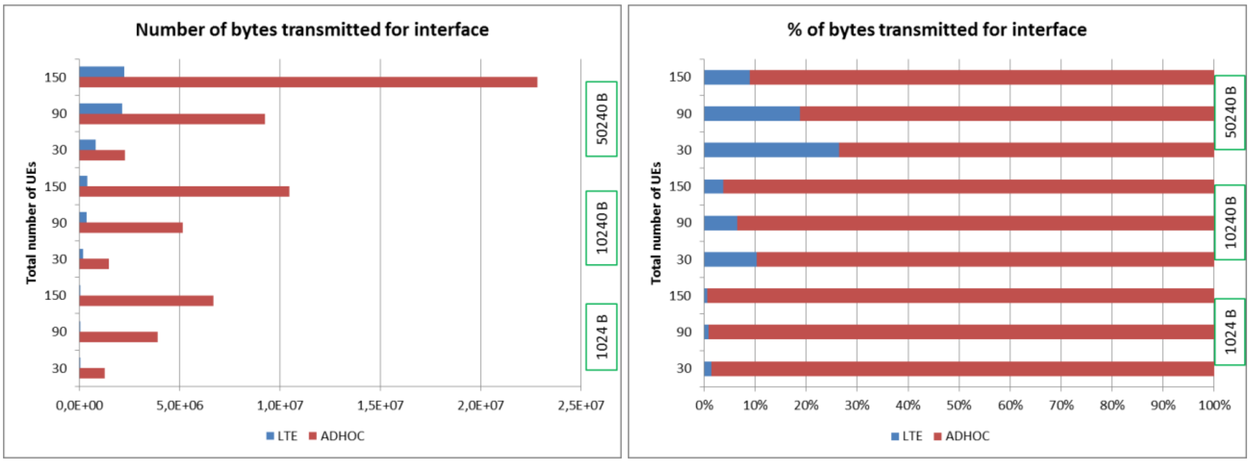


Figure 36: Percentage of UEs reached by Wi-Fi for different content sizes and total number of users

The offloading ratio increases with the total number of UEs mainly due to the improved collaboration among users. In fact, the user density increases accordingly and gives the possibility to the vehicles to better collaborate in order to exchange the content through the Wi-Fi interface instead of using the LTE one. The content size has also a clear impact on the offloading capabilities: the smaller it is and the better will be the collaboration among users through the opportunistic channel. In fact, the limited contact time between vehicles can be better exploited with small contents.

The distribution of transmitted bytes over LTE and Wi-Fi interfaces is depicted in Figure 37 (a) and (b), considering the total number and the corresponding percentages, respectively.

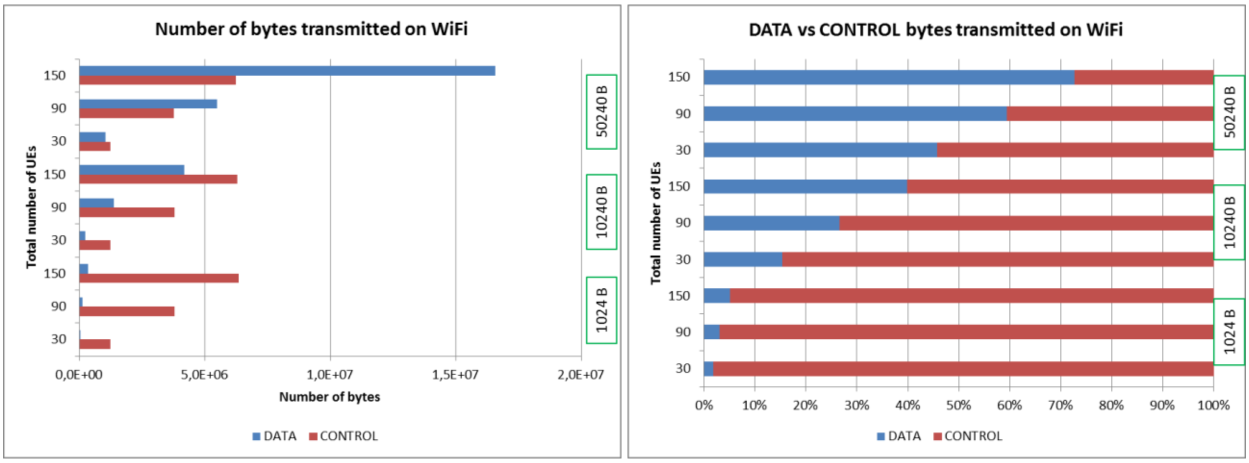


(a) (b)

Figure 37: Number (a) and percentage (b) of bytes transmitted over LTE and Wi-Fi

In general, **the total number of transmitted bytes increases with the content size** as intuitively plausible. Moreover, the LTE involvement has the same behaviour because this interface is more used for disseminating the content once the panic zone has been reached. This trend is more evident when the total number of UEs is low.

The traffic passing through the Wi-Fi interface (depicted in red in the previous Figure) can be divided in DATA and CONTROL bytes according to the number and proportion reported in Figure 38 (a) and (b), respectively.



(a) (b)

Figure 38: Number of bytes (a) and percentage (b) of DATA and CONTROL traffic passing through Wi-Fi

The total number of CONTROL bytes is almost independent from the content size (i.e., their values remain constant in Figure 38 (a)). On the other side, the amount of DATA bytes increases with the content size. This behaviour leads to a sort of less efficient use of the Wi-Fi channel for a small content size. In fact, almost the whole capacity is used for transmitting CONTROL data. When increasing the content size, this behaviour dramatically changes bringing to a more efficient use of the channel in terms of goodput (Figure 38 (b)).

In terms of delivery time, the total delay from first inject time is depicted in Figure 39 split up for interface type. This value increases with the number of users for the Wi-Fi interface.

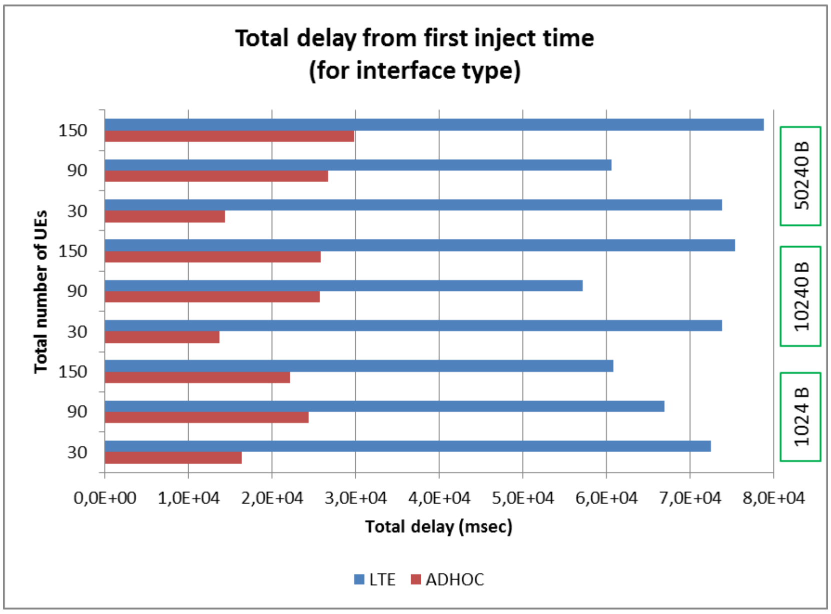
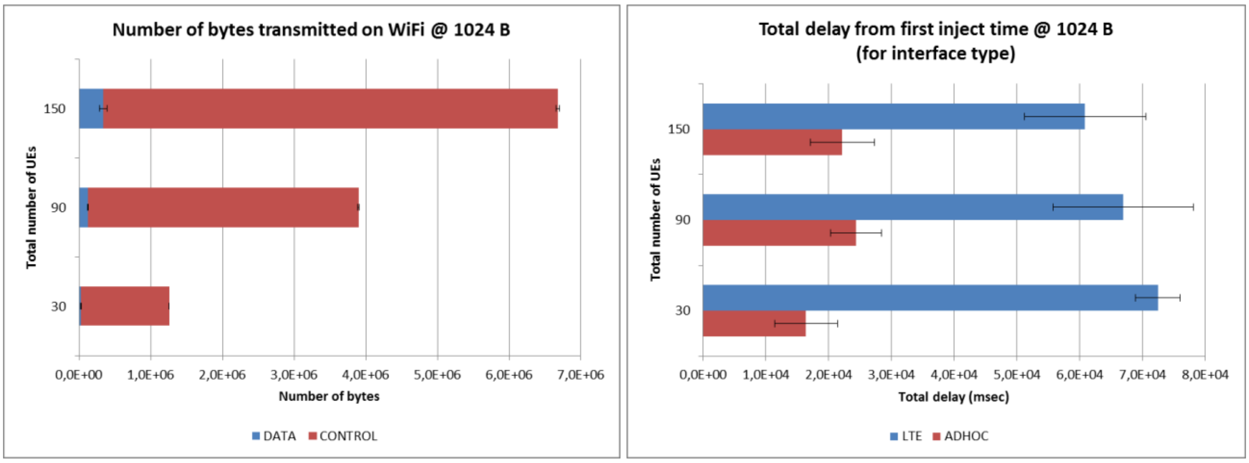


Figure 39: Total delay from first inject time split up for interface type

As stated before, the 90% confidence interval has been obtained only for a subset of the executed simulations, corresponding to all the combinations with a content dimension of 1024 B. Concerning the results presented up to now, the obtained interval is neglected for almost each of the measured metrics. The only two ones that are characterized by a perceptible confidence interval are the number of bytes transmitted on Wi-Fi and the total delay from first inject time, as depicted in Figure 40 (a) and (b), respectively.

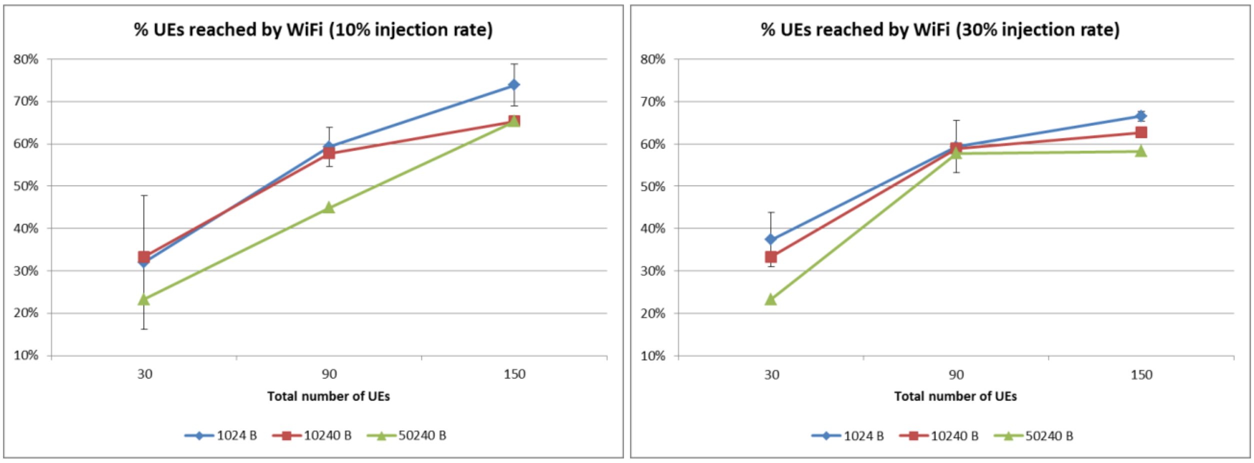


(a) (b)

Figure 40: Number of bytes transmitted on Wi-Fi (a) and total delay from first inject (b) with a packet dimension of 1024 B

A second set of results has been obtained by considering the initial when-strategy with a total number of UEs equal to 30, 90, and 150. The number of UEs which initially receive the content through the LTE data channel is set to 10% or 30% of the total number of users (e.g., 3 users are randomly selected as seeds for a 10% injection rate when the total number of UEs is equal to 30). Again, the considered propagation channel includes both attenuation and Nakagami fading model and the size of the offloaded content is equal to 1024 B, 10240 B, and 50240 B.

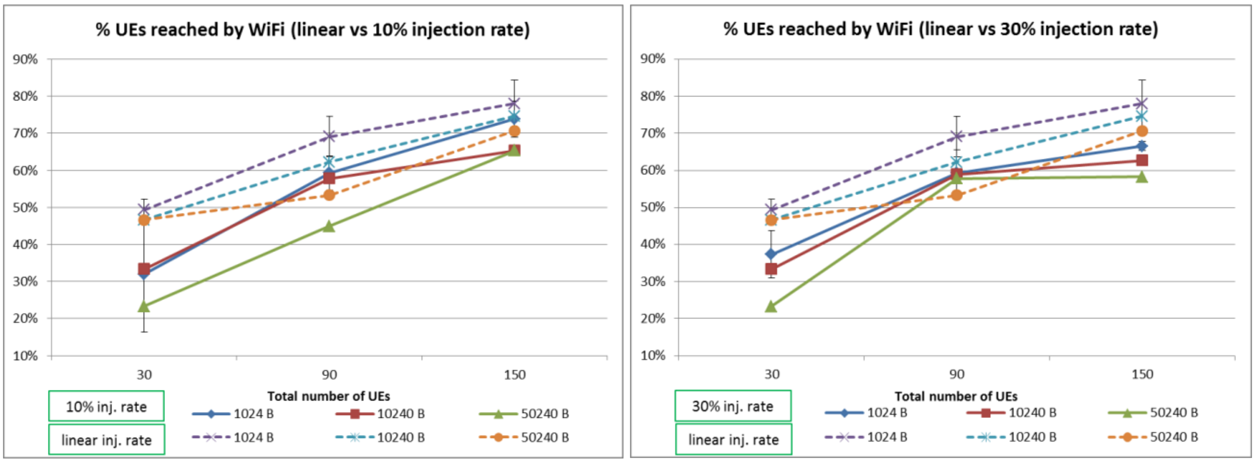
The fraction of users, which have been able to obtain the content through the Wi-Fi interface, is reported in Figure 41 (a) and (b) for the different considered sizes with an initial injection rate of 10% and 30%, respectively.



(a) (b)

Figure 41: Percentage of UEs reached by Wi-Fi for different content sizes and total number of users, with an initial injection rate of 10% (a) and 30% (b)

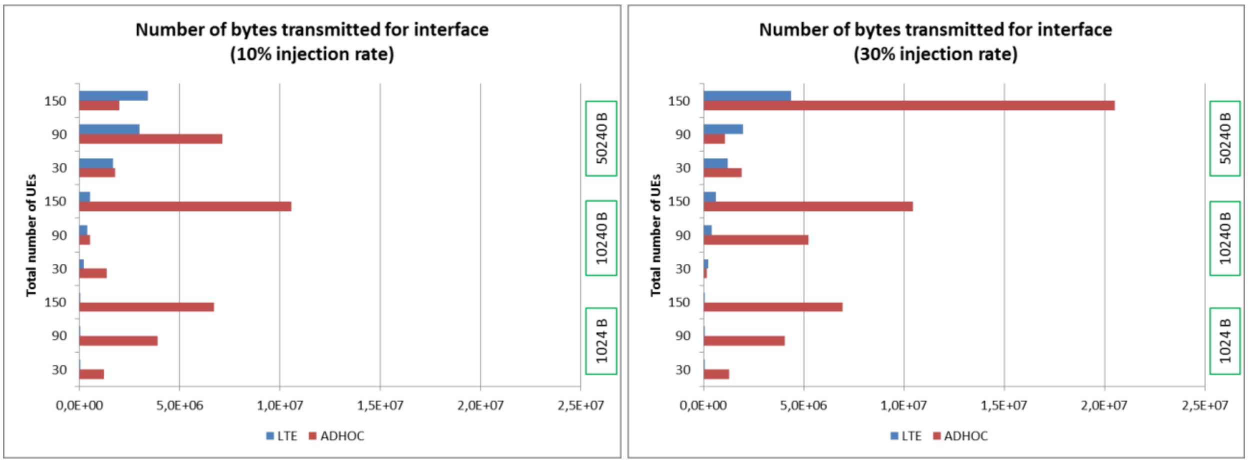
The behaviour observed with the initial when-strategy is similar to the one already described for the linear one. In fact, the offloading ratio increases with the total number of UEs thanks to the corresponding increased number of available initial seeds. The linear strategy provides slightly better results with respect to the initial one for a low number of total UEs and comparable results when this number increases (see Figure 42 (a) and (b)). This is mainly because the number of seeds is fixed and quite limited when few users are considered with the initial when-strategy, whereas it constantly increases with the linear one.



(a) (b)

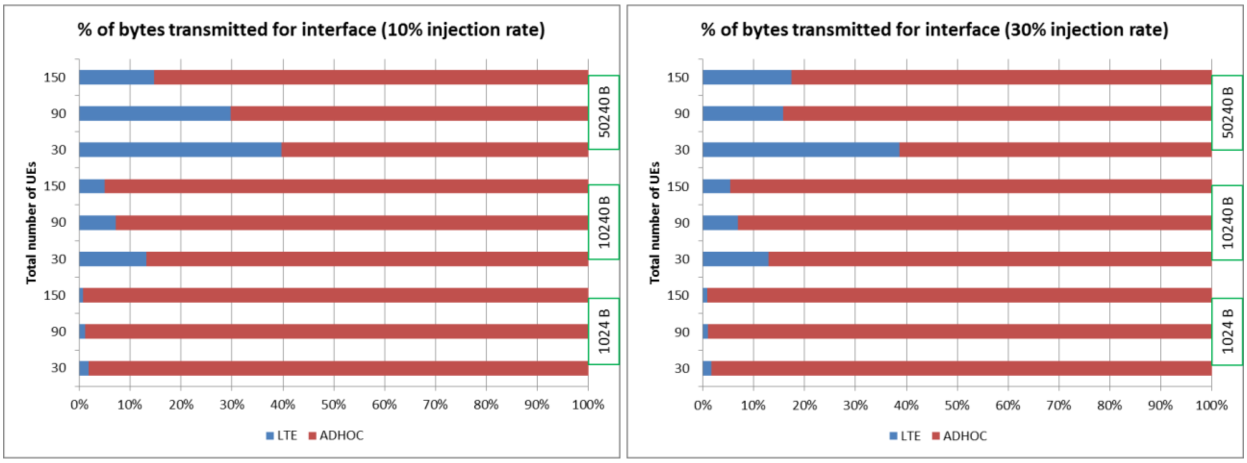
Figure 42: Comparison of the linear and the initial when-strategy in terms of percentage of UEs reached by Wi-Fi, with an initial injection rate of 10% (a) and 30% (b)

The total number of transmitted bytes over LTE and Wi-Fi interfaces is depicted in Figure 43 (a) and (b) for the two considered injection rate, 10% and 30% respectively. The corresponding distribution in terms of percentage is then depicted in Figure 44.



(a) (b)

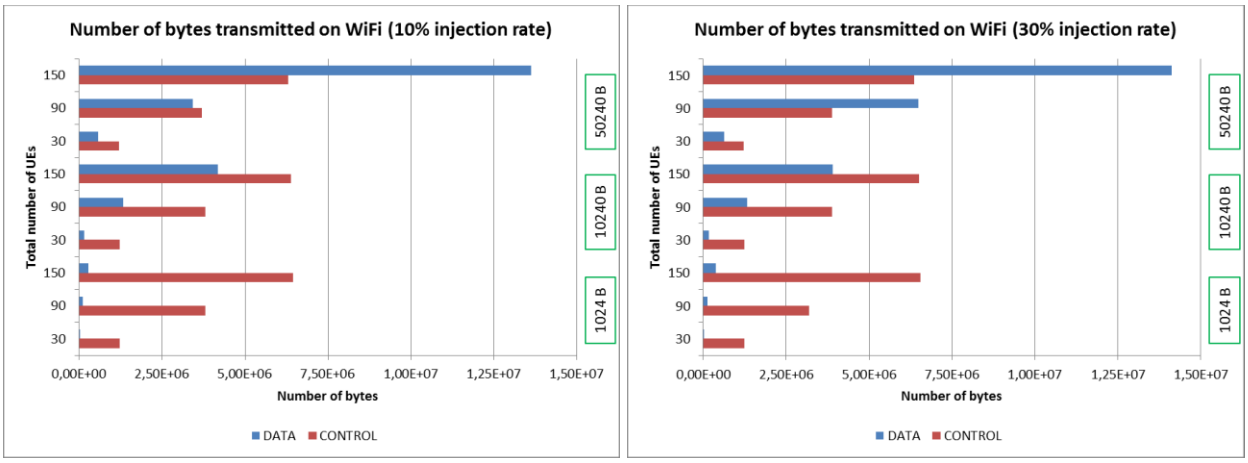
Figure 43: Number of bytes transmitted over LTE and Wi-Fi, with an initial injection rate of 10% (a) and 30% (b)



(a) (b)

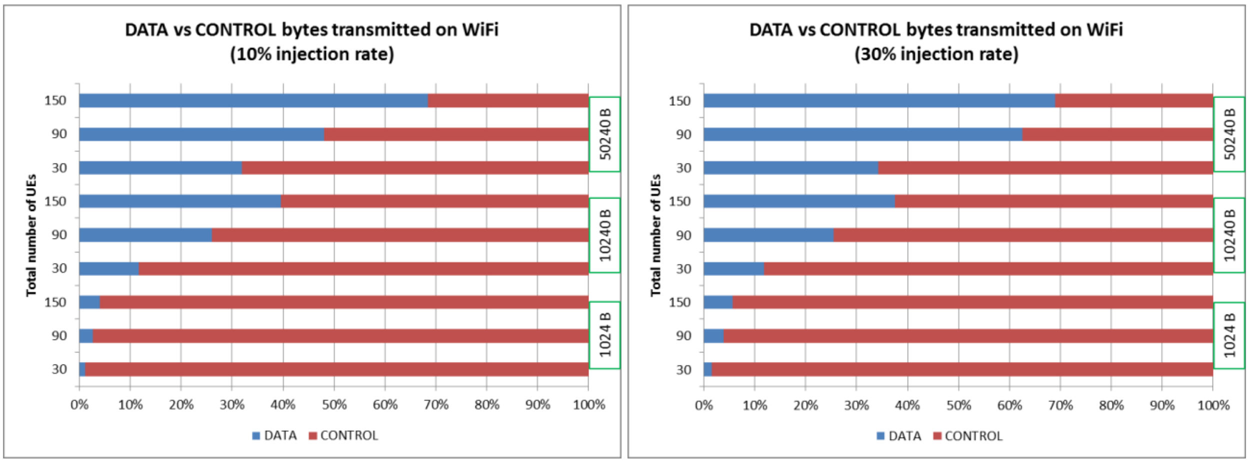
Figure 44: Percentage of bytes transmitted over LTE and Wi-Fi, with an initial injection rate of 10% (a) and 30% (b)

The splitting between DATA and CONTROL bytes for the traffic passing through the Wi-Fi interface is depicted in Figure 45 and Figure 46 in terms of absolute number and proportion, respectively.



(a) (b)

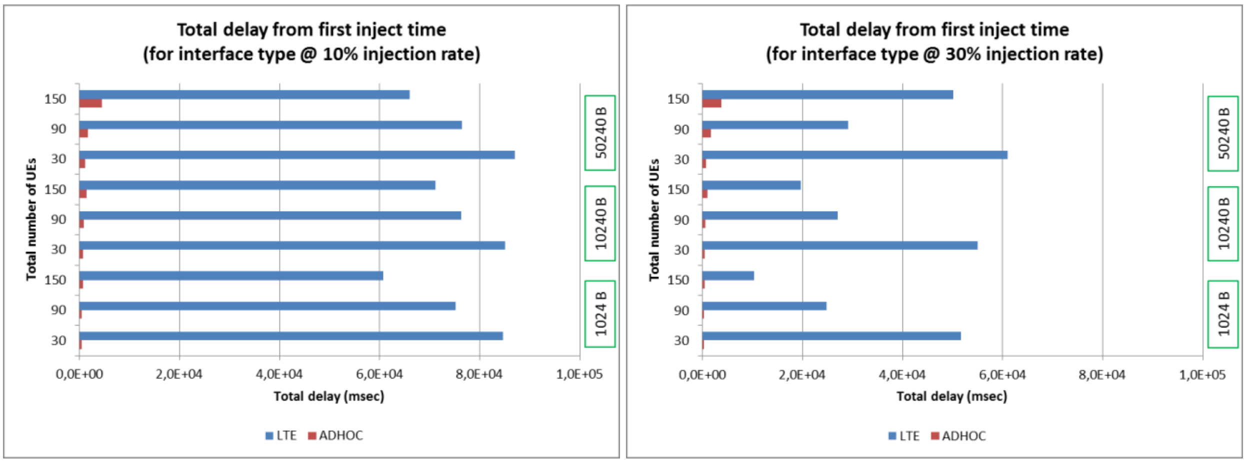
Figure 45: Number of bytes of DATA and CONTROL traffic passing through Wi-Fi, with an initial injection rate of 10% (a) and 30% (b)



(a) (b)

Figure 46: Percentage of DATA and CONTROL traffic passing through Wi-Fi, with an initial injection rate of 10% (a) and 30% (b)

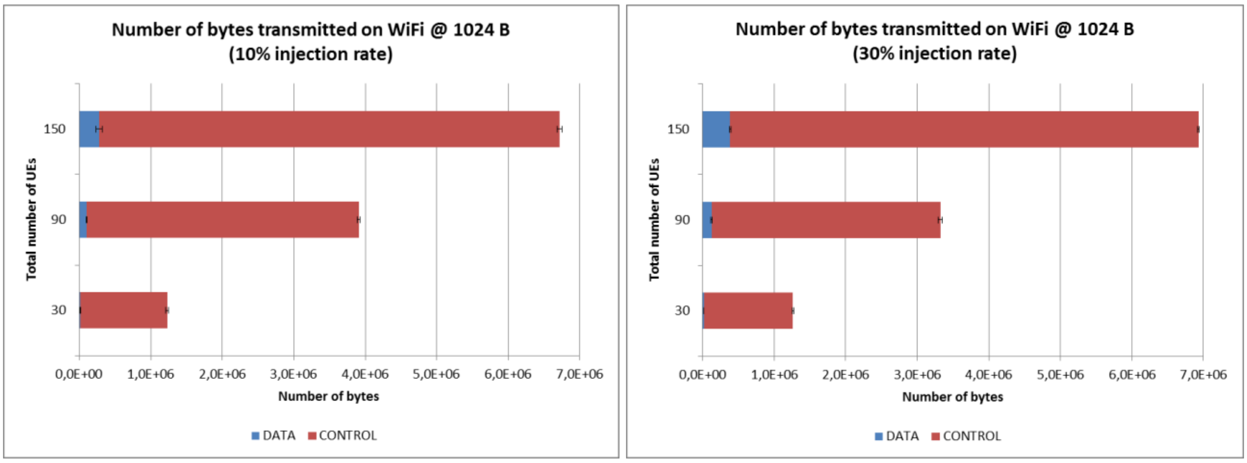
The trend of the metrics presented above for the initial when-strategy is in line with the previously described one related to the linear strategy. In general, it can be observed that the initial strategy is less efficient than the linear one for a small number of UEs (i.e., the 30 users' case); both in terms of efficiency of the Wi-Fi channel and need to use the LTE channel in order to deliver the content to the remaining users. This behaviour can be mitigated by increasing the number of initial seeds selected for injecting the first set of copies of the content to be then offloaded to the other users (i.e., passing from an injection rate of 10% to 30%). The total delay from first inject time is depicted in Figure 47 split up for interface type.



(a) (b)

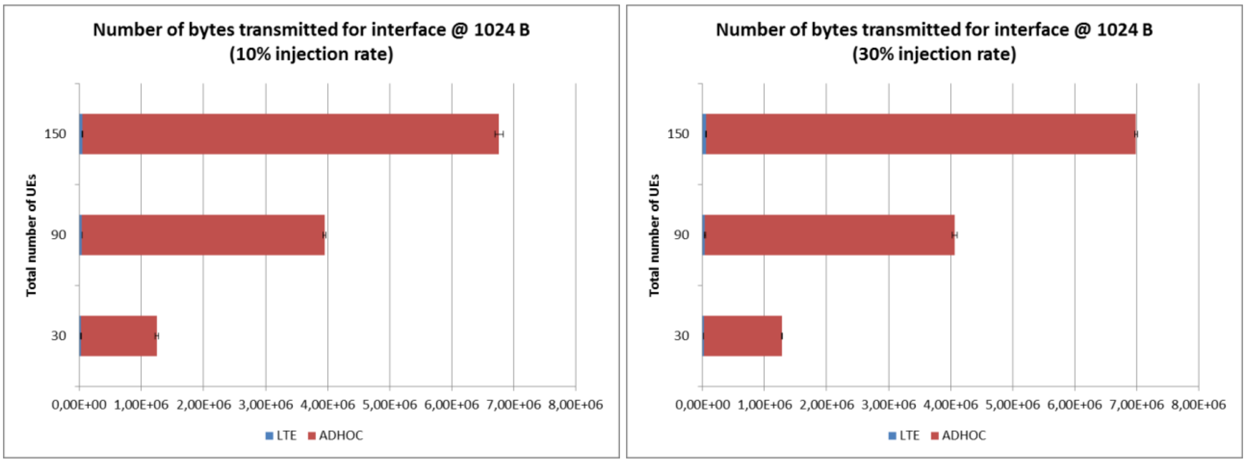
Figure 47: Total delay from first inject time split up for interface type, with an initial injection rate of 10% (a) and 30% (b)

Also for the above presented second set of results, the 90% confidence interval has been obtained only for the simulations that consider a content dimension of 1024 B. This interval is again neglected for almost each of the measured metrics. The ones, which are characterized by a perceptible confidence interval, are the number of bytes transmitted on Wi-Fi (Figure 48), the number of bytes transmitted for interface (Figure 49), and the total delay from first inject time (Figure 50).



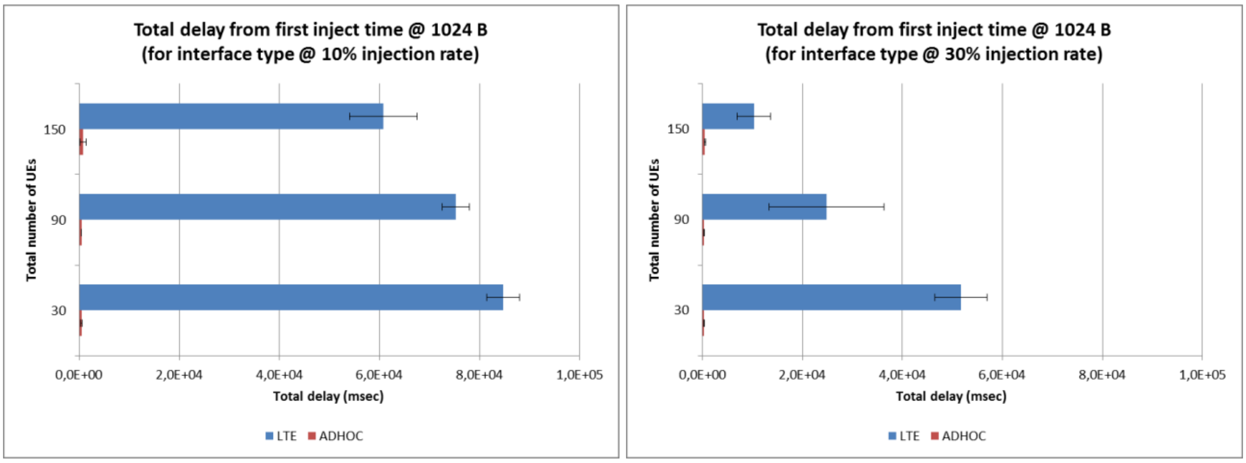
(a) (b)

Figure 48: Number of bytes transmitted on Wi-Fi, with an initial injection rate of 10% (a) and 30% (b)



(a) (b)

Figure 49: Number of bytes transmitted for interface, with an initial injection rate of 10% (a) and 30% (b)



(a) (b)

Figure 50: Total delay from first inject time, with an initial injection rate of 10% (a) and 30% (b)

All the results presented up to now have been obtained by considering a wireless Wi-Fi channel characterized by attenuation and Nakagami multipath fading. An additional set of simulations has been executed in order to understand the level of degradation introduced by the multipath propagation. To do that, the effect of the Nakagami multipath fading has been removed and only the attenuation has been introduced in the considered communication channel by using a log distance path loss propagation model.

This limited set of additional simulations has been executed by considering the initial when-strategy with a 10% injection rate, a content dimension of 1024 B, 10240 B, and 50240 B, and a total number of users equal to 30, 90, and 150.

The fraction of users, which have been able to obtain the content through the Wi-Fi interface, is reported in Figure 51. The offloading ratio is higher when the channel is characterized by attenuation only, especially for a big content dimension. This behaviour is in line with the additional signal degradation introduced by the multipath propagation.

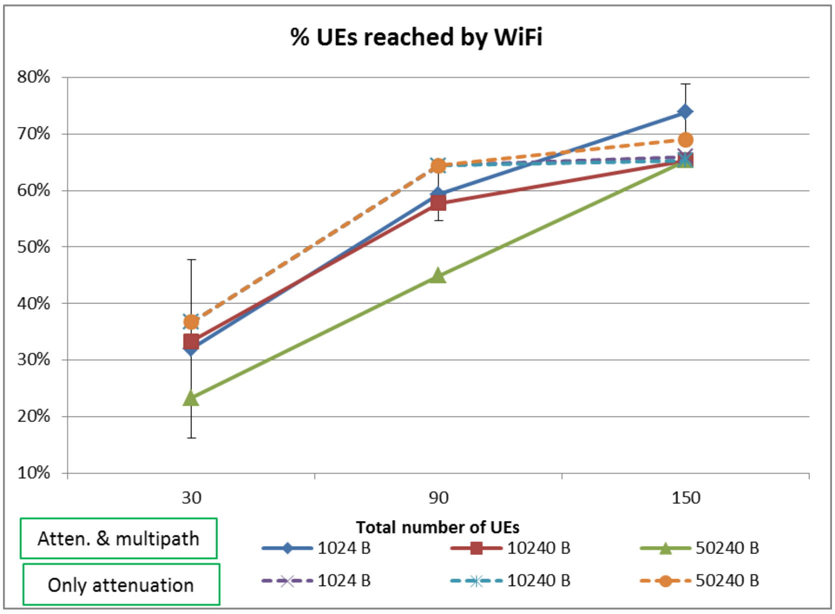
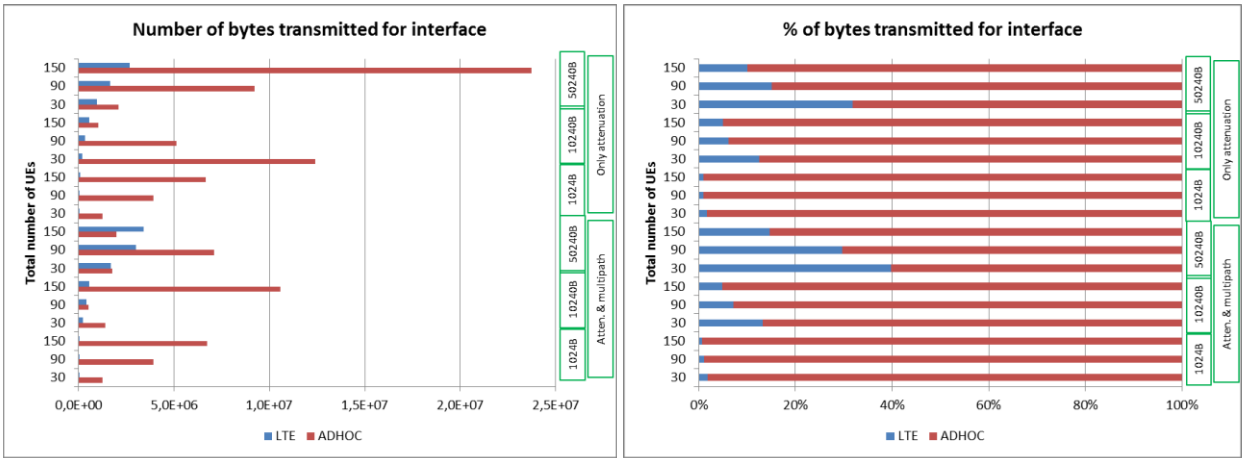


Figure 51: Percentage of UEs reached by Wi-Fi for different communication channels (attenuation with multipath and only attenuation)

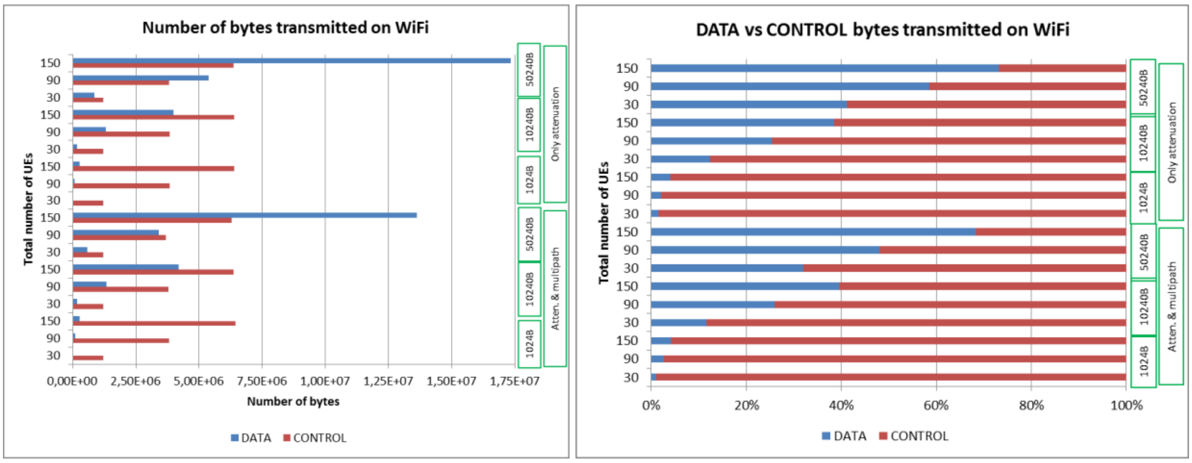
The distribution of transmitted bytes over LTE and Wi-Fi interfaces is depicted in Figure 52 (a) and (b), considering the total number and the corresponding percentages, respectively. The traffic passing through the Wi-Fi interface can be divided in DATA and CONTROL bytes according to the number and proportion reported in Figure 53 (a) and (b), respectively. In general, the channel without multipath propagation is able to offer better performances in terms of usage and goodput.



(a)

(b)

Figure 52: Number (a) and percentage (b) of bytes transmitted over LTE and Wi-Fi for different communication channels (attenuation with multipath and only attenuation)



(a) (b)

Figure 53: Number of bytes (a) and percentage (b) of DATA and CONTROL traffic passing through Wi-Fi for different communication channels (attenuation with multipath and only attenuation)

The total delay from first inject time is depicted in Figure 54 split up for interface type.

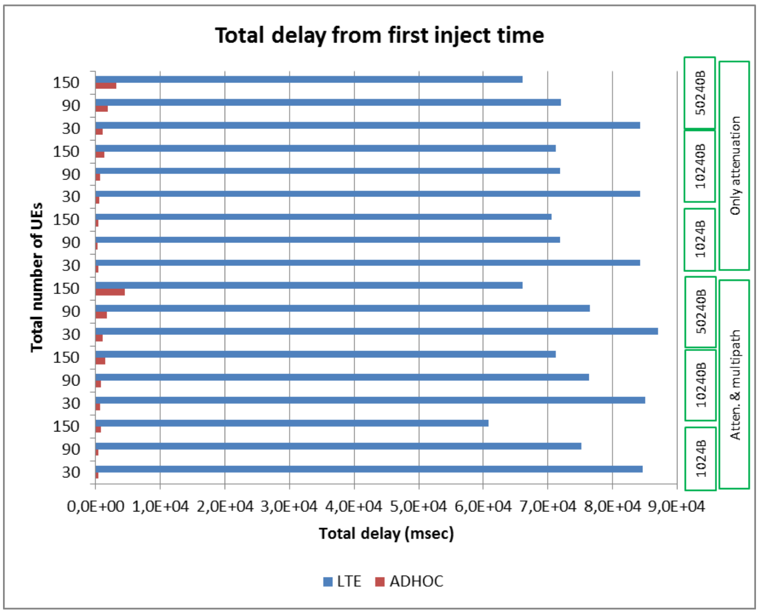


Figure 54: Total delay from first inject time split up for interface type for different communication channels (attenuation with multipath and only attenuation)

3.2.2.7 Simulation conclusions

The effectiveness of the proposed offloading architecture has been proven in a vehicular scenario where cars move towards the same direction on a highway and require the same map content. Several simulations have been executed with different set of parameters in order to obtain a complete overview of the system performance. The scenario has been designed by considering the path of a **real Italian highway**. Moreover, the simulation parameters have been chosen in order to obtain a realistic behaviour in terms of communication capabilities. Under these circumstances, all the previously described expected results have been confirmed by the executed simulations.

The effectiveness of the offloading process (i.e., the percentage of UEs reached by Wi-Fi) **is directly related to the number of users available in a certain area** (i.e., the higher this number, the better is the cooperation among users for disseminating the content of interest). The dimension of the offloaded content also influences such a metric, reducing the efficient usage of the intercontact time when it is increased. Moreover, it also has an impact on the number of bytes transmitted per interface (i.e., the higher the content size, the higher is the number of transmitted bytes and the involvement of the LTE interface). The total amount of CONTROL bytes is not influenced by the content size, but it only depends on the number of users. The DATA traffic is instead strictly dependent from the dimension of the content to be disseminated. The linear who-strategy can be considered more efficient than the initial one when the number of users is low. In other cases, both are quite similar in terms of offloading ratio.

3.3 Security simulation

The security simulation in this task is treated as a special case, in order to validate the viability of the proposed security solution. The objective of these simulations is to evaluate the impact of applying the security solution to the offloading algorithms.

3.3.1.1 Target

The security simulations performed during this task have been devoted to test three main aspects of the proposed security solution for the MOTO offloading approach. In the first place, simulations are intended to test if the delay introduced by the dissemination of pseudonyms and key-pairs within the UEs affect severely the communications. The second set of simulations are devoted to quantify the delay that the authentication messages to be exchanged by UEs previous to the content exchange introduce and evaluate if this delay is assumable under the expected QoE. Finally, the third wave of simulations performed searches to quantify and evaluate how the trust scheme proposed impacts the communications performance. Following a brief summary of the simulations performed under the task is exposed:

- **First wave of simulations**: Monte Carlo simulations are performed to check the efficiency of the distribution of keys and pseudonyms. In such simulations, the time consumed in order to stabilize the network is measured, after a refreshment of keys and pseudonyms under different user densities.
- **Second wave of simulations**: Event simulations are performed to measure the additional delay introduced by connection establishment messages related to security exchanged before content sending.
- **Third wave of simulations**: Event simulations are performed in order to measure the deterioration of the throughput because of the trust mechanisms implementation.

3.3.1.2 Layout and description

The three types of simulations defined in the previous section have been conducted on the same scenario, a square plot area of 100 m^2 (10 m x 10 m). The eNB node has been located in the centre of this square, that is, at the coordinates (X = 5, Y = 5). The remaining nodes (seeds and Ues) are randomly distributed throughout the simulation area. The bandwidth assigned to MOTO platform is the 50% of the bandwidth available. Under this scenario, the three performed simulation waves have been stablished:

1. The first simulation wave is intended to check the time for network stabilization, that is, the time elapsed since MOTO platform launches the refreshing of user’s key pairs and pseudonyms, and the time when all nodes have received their new keys and pseudonyms. The messages that serve to distribute the keys and the pseudonyms are sent under the operators’ infrastructure, this is, LTE. Simulations have been conducted with different key and pseudonyms length so as to evaluate if it is viable to send, in the same set of refreshments, several pairs of keys and pseudonyms, in order to delay as little as possible the rest of the communications

2. The second wave of simulations measures the elapsed time between the instant when MOTO platform sends a content to be distributed, and the time when all intended nodes are in possession of it. This time includes not only the distribution of the content but also the exchange of point-to-point messages related to security (users authentication, authorization, etc.) for UE to UE communication.
3. The third wave of simulations consists of evaluating the delay introduced by the trust mechanisms, taking into account that the trust feedbacks have to be generated, transmitted and processed within the MOTO platform.

3.3.1.3 Expected results

The conducted simulations aim at validating the feasibility of the proposed security approach for the MOTO environment. Of course, some delays / performance impact due to the introduction of security are expected.

Unfortunately, not all the adverse effects of adding security are measurable or capable of being simulated; however, they mostly affect the communications by adding delays, which can be measured. Therefore, the three simulation waves that have been performed take the delay introduced as the main metric to be measured:

1. The first simulation wave is aimed to measure the delay introduced by the pseudonym and key pairs refreshing. Ideally, the time for network stabilization should be the lowest possible, because during this refreshing, nodes will be unable of sending or receiving data, at least those of them which are receiving the pseudonym and keys at each instant. It is foreseen that the total time for network stabilization is highly dependent on the density of nodes. However, it is anticipated that, in the worst scenarios, the refreshing will not introduce a delay that is considered unacceptable for the QoE.
2. The second wave of simulations is aimed to measure the delay of content distribution between UEs under security conditions. Again, it is expected that the inclusion of security, which inevitably increases the number of messages to be exchanged, add delay to the content offloading with respect to an unsecure offloading scheme. However, as in the previous wave, it is reasonable to predict that the delay introduced by security in the offloading is not unacceptable with respect to the compliance of the QoE.
3. Finally, the third wave is aimed to measure the decrease of the throughput of the communications due to the introduction of the trust framework. As such, a delay is predicted to be added considering the time needed by the MOTO platform to process the trust feedback, as well as the addition of trust information in the UEs feedback. However, it is expected that the processing capacity of the MOTO platform is not severely affected by this trust processing, and that the delay is not relevant. Apart from that, we are expecting that the additional information related to trust is minimal and that the overall delay will not be significantly impacted.

3.3.1.4 Simulation parameters

In this section, the parameters that are introduced in the simulation tool in order to perform the stated simulations are listed.

The following table shows the parameters that were set for the first wave of simulations:

Parameter	Variable/fixed	Units	Min	Max
Number of nodes	Variable	Number	10	150
Number of eNBs	Fixed	Number	1	1

Type of content	Fixed	Video/photo/text	text	
Size of content	Variable	bits	320	600
Message lifetime	Variable	s	1	10
Time to panic zone	Variable	s	3	10
Mobility patterns of the nodes	Fixed	Linear, Random…	random	random
Number of periodic injects	Fixed	Number		
Time of first injection	Fixed	S	1	1
Enable position forwarding	Fixed	Yes/No	Yes	
Simulation time	Fixed	s	10	10
SPECIFIC PARAMETERS RELATED TO SECURITY				
Pseudonym strategy	Fixed	Static, variable, variable to different nodes	variable to different nodes	Static

Table 11. Simulation parameters of the security simulations first wave

The following table shows the parameters that were stated for the second and third wave of simulations:

Parameter	Variable/fixed	Units	Min	Max
Number of nodes	Variable	Number	10	150
Number of eNBs	Fixed	Number	1	1
Size of content	Variable	Bytes	40	30 k
Message lifetime	Variable	s	1	10
Time to panic zone	Variable	s	3	10
Mobility patterns of the nodes	Fixed	Linear, Random…	random	random
Number of periodic injects	Fixed	Number	10	10
Time of first injection	Fixed	S	1	1
Enable position forwarding	Fixed	Yes/No	Yes	
Simulation time	Fixed	s	10	10
SPECIFIC PARAMETERS RELATED TO SECURITY				
Pseudonym strategy	Fixed	Static, variable, variable to different nodes	variable to different nodes	static

Table 12.simulation parameters of the security simulations second and third waves

The main difference between the parameters used for the second and third wave of simulations is the inclusion of sent trust information. We are not going to include it, as it will mean to duplicate the same table.

3.3.1.5 Metrics

The main performance metric we consider is the content end-to end-delay, the throughput and the delay introduced by the security implementation.

Metrics	Units
Content End-to-end delay	sec
Throughput	sec
Delay introduced by the pseudonyms authentication	sec

Table 13. Metrics of the security simulations

3.3.1.6 Simulation results

*T*he results of the first type of simulations are shown below:

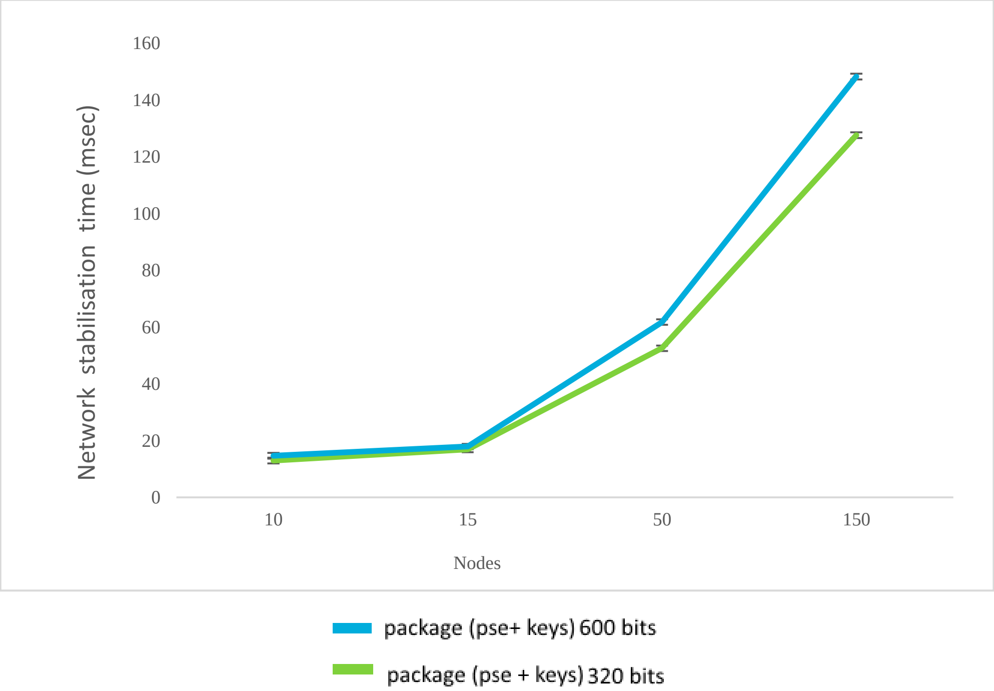


Figure 55 Network stabilization time

During the first wave of simulations, we evaluated the stabilization time of the network for pseudonym and key pairs refreshing. That is, the time elapsed from starting routine of refreshing the nodes pseudonyms and key pairs, until the time when all nodes are capable of starting the ad-hoc communications again.

🟦 package (pse+ keys) 600 bits
🟩 package (pse + keys) 320 bits

This graphical representation, shown in Figure 55, shows that the stabilization time is acceptable. The maximum time interval for the maximum number of nodes simulated (150) to receive the pseudonyms and key pairs, and to be able to perform communication through the Ad-Hoc channel again, is 150 ms. This time is considered acceptable taking into account the delay-tolerant nature of the proposed communication, which is expected to admit delays in the range of tens of seconds.

Apart from that, we have demonstrated that in a same refreshing rush we can send several keys and pseudonyms, allowing reducing the frequency of service interruption due to this security routine.

The results achieved with the second wave of simulations are shown in Figure 56 bellow:

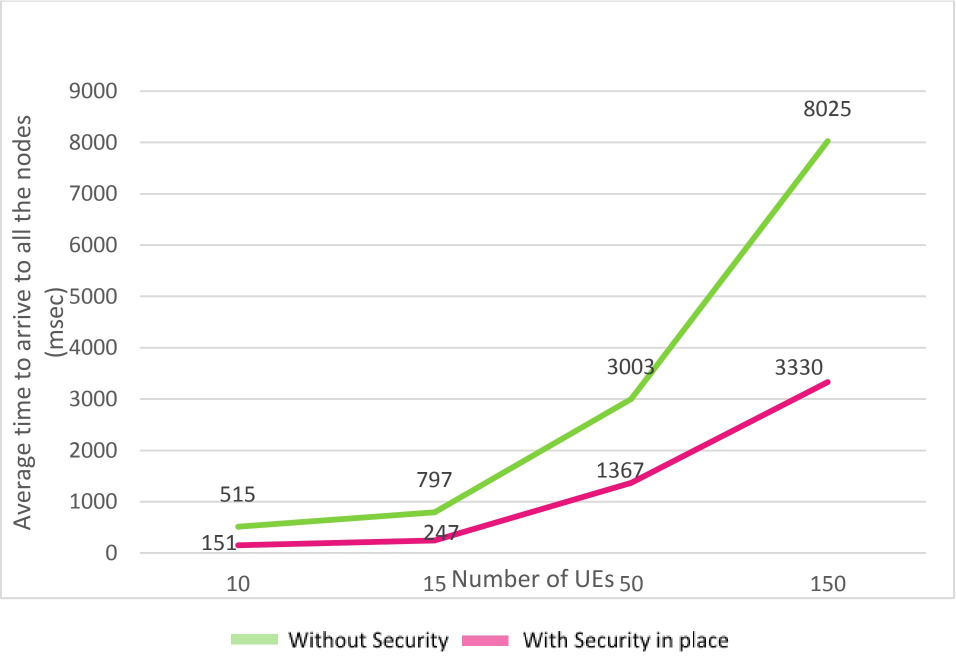


Figure 56 Average time to arrive to all the nodes (security) (msec)

Figure 56 represents a comparison between the average time in which all the nodes receive the content with security and without security in place. This simulation considers that all the nodes have already their corresponding valid keys and pseudonyms. The pink line in Figure 56 shows the corresponding results of a simulation without security in place, that is, without interchanging any security message before the content exchange. On the other hand, values defined by the green line correspond to the results of simulations when security is in place, including the previous interchange of keys and pseudonyms in the ad-hoc communication.

As it can be appreciated, when there are 150 nodes, the time elapsed to reach all the nodes is 3,3s without security and 8s with security. This delay may not be acceptable for services demanding real time communications. However, as within MOTO we are considering delay tolerant communications, this delay can be considered acceptable depending on the constrains stated by the content provider in the SLA.

Finally, the results of the third wave of simulations are shown below in

🟩 Security (pseu) 🟥 Security (pseu + trust)

Figure 57:

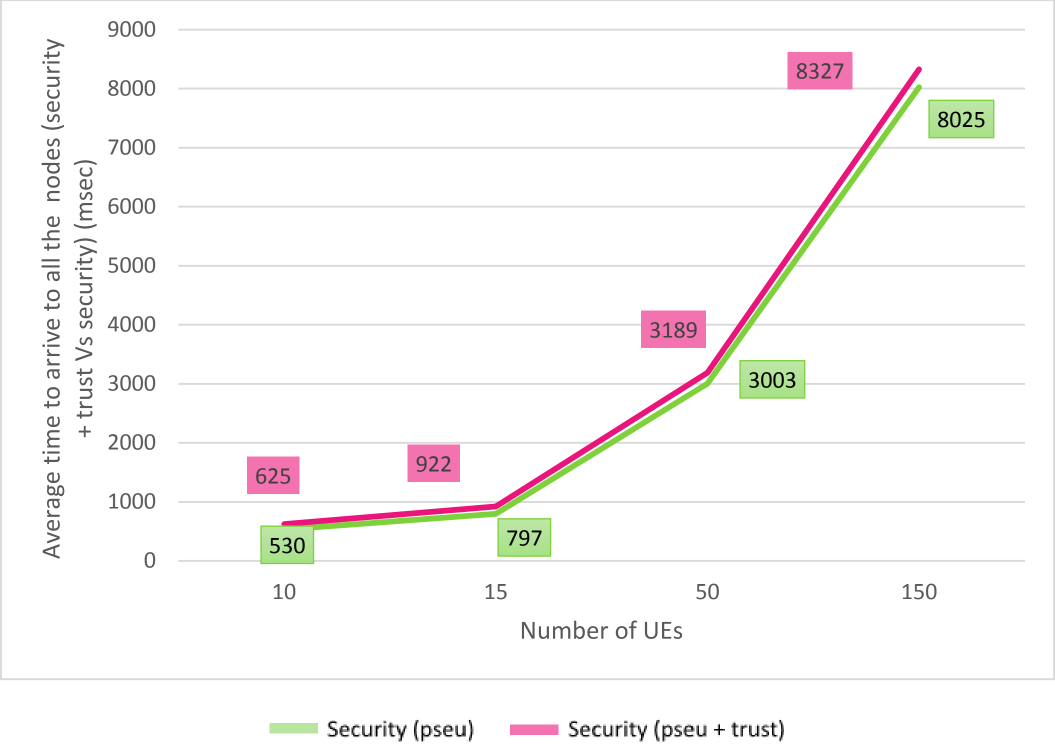


Figure 57 Average time to arrive to all the nodes (security + trust Vs security) (msec)

This third wave of simulations include, not only the security mechanisms already included in the second wave of simulations (ad-hoc authentication and pseudonyms anonymization) but also the trust mechanisms that are needed to allow the functioning of the trust framework proposed. That is, the inclusion of trust information within UEs feedback after a content has been received.

🟩 Security (pseu) 🟥 Security (pseu + trust)

Figure 57 represents a comparison between the average time to receive the content with security (ad-hoc authentication and pseudonyms anonymization) but without the trust (green) and with security and trust all together (pink). It can be seen that the introduction of the trust mechanisms does not involve relevant delays even more taking into account that the benefits of including trust are considered essential.

Finally, to have a global view of the throughput in all the situations presented in this section, and with the variation of the number of nodes, Figure 58 bellow, shows the throughput obtained when the information reaches all the nodes. The different coloured lines represent:

- PINK line: No security in place
- GREEN line: Some security in place (ad-hoc authentication and pseudonyms anonymization) – green
- Yellow line: With all security mechanism in place (ad-hoc authentication, pseudonyms anonymization and trust)

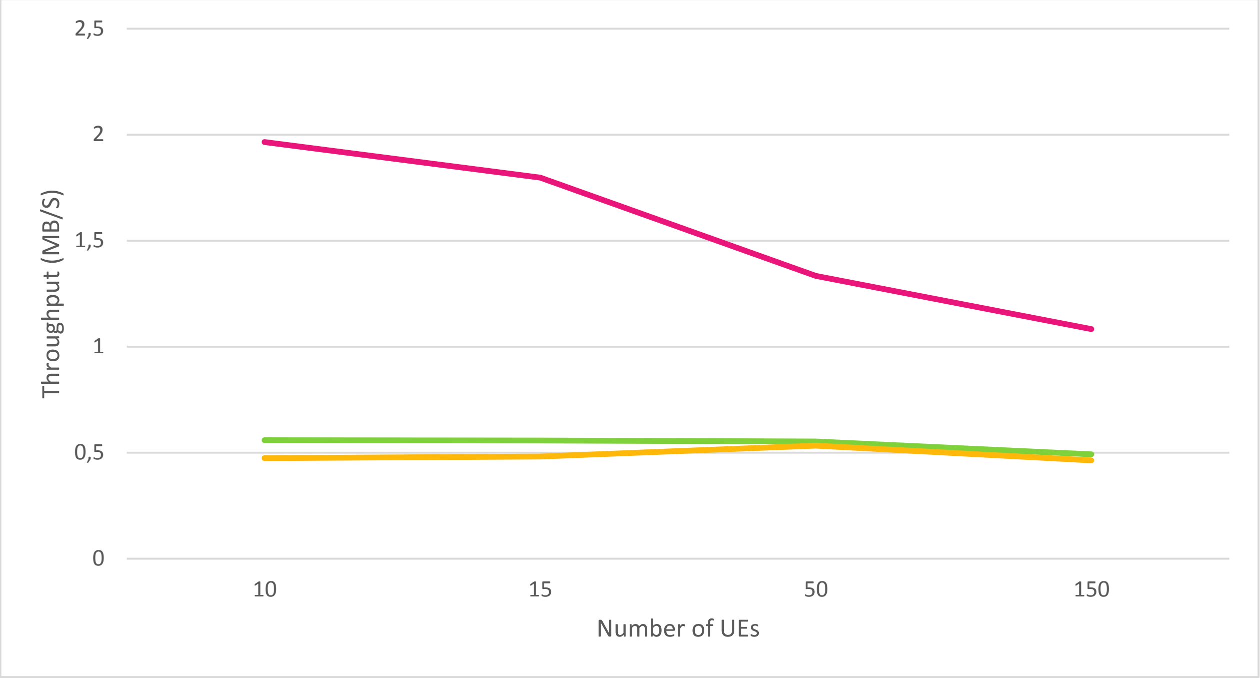


Figure 58. Throughput obtained when the information reaches all the nodes (security vs no security)

3.3.1.7 Conclusions

With these simulations, we have been able to confirm our initial expectations: the performance of the proposed security solution in terms of delay and throughput is acceptable when dealing with delay tolerant services that do not have real time requirements. In the case of real time services, the proposed security approach is acceptable up to 50 nodes, as it has been checked that the delay introduced with 150 nodes is 6 sec higher.

Apart from that, regarding the previous exchange of keys, we have been able to demonstrate that it is possible to send in a same refreshing rush several sets or pseudonyms for different time slots.

4 CONCLUSIONS

This deliverable exposes the simulations carried out in the MOTO project, under task T5.2 to demonstrate the different hypotheses that have conducted the research done. The simulations have been conducted based on the scenarios proposed in D2.1. Indeed, two main types of simulations (pedestrian and vehicular) have been conducted in order to cover the main offloading situations where the MOTO services could have important benefits for both the users and the operators.

These simulations have allowed confirming several hypotheses. In indoor environments, such in a museum, we have been able to observe that a long shared timeout is effective in improving the opportunistic diffusion because more copies stay in the network. Furthermore, the more content items there are, the lower is the offloading efficiency. Moreover, it has been concluded that a constrained mobility model (which is typical in indoor conditions) reduces the effectiveness of the opportunistic dissemination because different users can be interested in disjoint sets of content items based on their location.

Duty-cycling strategies have been evaluated using a fixed-mobility use case. Duty cycling is important with respect to the maximum battery life of users participating in the opportunistic forwarding. An uncontrolled battery drain could have the effect of discouraging users by offering their mobile resources in the offloading process. We observed that there is a linear relation between the loss in offloading efficiency and the energy savings brought by the use of duty-cycling.

By means of the vehicular simulations, we have clearly seen that the offloading efficiency decreases with the number of content items, due to increasing contention of the opportunistic network resources. Equally to the pedestrian scenarios, the more the content and sharing timeout increases, the better, as content items remain available in the opportunistic network for longer. In general, the results obtained confirm that offloading can be very efficient also for non-synchronised content requests, if parameters of the algorithms are tuned appropriately for the considered environment.

The Map-Based Advanced Driver Assistance Systems simulation has allowed us to demonstrate that the effectiveness of the offloading process (i.e., the percentage of UEs reached by Wi-Fi) is directly related to the number of users available in a certain area (i.e., the higher this number, the better is the cooperation among users for disseminating the content of interest). Moreover, we have confirmed that the total amount of CONTROL bytes is not influenced by the content size, but it only depends on the number of users. The DATA traffic is instead strictly dependent from the dimension of the content to be disseminated.

Finally, the security simulations carried out have allowed demonstrating that the performance of the proposed security solution in terms of delay and throughput is acceptable when dealing with delay tolerant services.

DISCLAIMER

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The above referenced consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.