Cooperative download in vehicular environments

Oscar Trullols-Cruces, *Student member, IEEE,* Marco Fiore, *Member, IEEE,* Jose Barcelo-Ordinas, *Member, IEEE*

***Abstract*— We consider a complex (i.e., non-linear) road scenario where users aboard vehicles equipped with communication interfaces are interested in downloading large files from road-side Access Points (APs). We investigate the possibility of exploiting opportunistic encounters among mobile nodes so to augment the transfer rate experienced by vehicular downloaders. To that end, we devise solutions for the selection of carriers and data chunks at the APs, and evaluate them in real-world road topologies, under different AP deployment strategies. Through extensive simulations, we show that carry&forward transfers can significantly increase the download rate of vehicular users in urban/suburban environments, and that such a result holds throughout diverse mobility scenarios, AP placements and network loads.**

***Index Terms*— Vehicular networks, cooperative downloading, delay tolerant networking, carry&forward transmission**

I. Introduction

Vehicles traveling within cities and along highways are regarded as most probable candidates for a complete integration into mobile networks of the next generation. Vehicle-to-infrastructure and vehicle-to-vehicle communication could indeed foster a number of new applications of notable interest and critical importance, ranging from danger warning to traffic congestion avoidance. It is however easy to foresee that the availability of on-board communication capabilities will also determine a significant increase in the number of mobile users regularly employing business and infotainment applications during their displacements. As a matter of fact, equipping vehicles with WiMAX/LTE and/or WiFi capabilities would represent a clear invitation for passengers on cars or buses to behave exactly as home-based network users. The phenomenon would thus affect not only lightweight services such as web browsing or e-mailing, but also resource-intensive ones such as streaming or file sharing.

In this paper, we focus on one of the latter tasks, namely the download of large-sized files from the Internet. More precisely, we consider a urban scenario, where users aboard cars can exploit roadside Access Points (APs) to access the servers that host the desired contents. We consider that the coverage provided by the roadside APs is intermittent: this is often the case, since, in presence of large urban, suburban and rural areas, a pervasive deployment of APs dedicated to vehicular access is often impractical, for economic and technical reasons. We also assume that not all on-board users download large files all the time: indeed, one can expect a behavior similar to that observed in wired networks, where the portion of queries for large contents is small [1]. As a result, only a minor percentage of APs is simultaneously involved in direct data transfers to downloader cars in their respective coverage area, and the majority of APs is instead idle.

O. Trullols-Cruces and J. M. Barcelo-Ordinas are with the Universitat Politècnica de Catalunya, Barcelona, Spain.

M. Fiore is with INSA Lyon and INRIA, Lyon, France.

An earlier version of this work appeared at IEEE MASS'09,

This work was partially supported by the EuroNF NoE and by Spanish grants TIN2010-21378-C02-01 and 2009-SGR-1167,

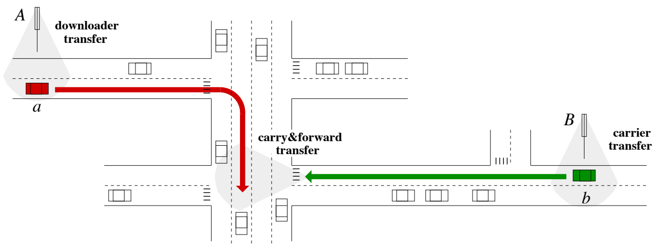


Fig. 1. Vehicle a downloads part of some content from AP A. The idle AP B delegates another portion of the same content to a vehicle b. When b encounters a, vehicle-to-vehicle communication is employed to transfer to a the data carried by b

Within such a context, we study how opportunistic vehicle-to-vehicle communication can complement the infrastructure-based connectivity, so to speed up the download process. We exploit the APs inactivity periods to transmit, to cars within range of idle APs, pieces of the data being currently downloaded by other vehicles. Cars that obtain information chunks this way can then transport the data in a *carry&forward* fashion [2], and deliver it to the destination vehicle, exploiting opportunistic contacts with it, as in Fig. 1. We remark that the concept of cooperative download in vehicular networks has been already proposed for highway environments: however, unlike what happens over unidimensional highways, urban/suburban road topologies present multiple route choices that make it hard to predict if vehicles will meet; moreover, the presence of traffic lights, stop and yield signs renders cars contact timings very variable. These key aspects make highway-tailored solutions impracticable in complex non-linear road scenarios, for which we are, to the best of our knowledge, the first to identify challenges and propose solutions.

After a discussion of the literature, in Sec. II, we outline the major challenges of vehicular cooperative download in urban environments and devise original solutions to them, in Sec. III. In Sec. IV we present the scenarios considered for our performance evaluation, whose results are then discussed in Sec. V. Conclusions are drawn in Sec. VI.

II. Related work

The cooperative download of contents from users aboard vehicles has been first studied in [3], that introduced SPAWN, a protocol for the retrieval and sharing of contents vehicular environments. SPAWN is designed for unidirectional traffic over a highway, and is built on the assumption that all on-road vehicles are active downloaders of a same content. Instead, we target urban environments where users may be interested in different contents. Similar considerations hold for the works in [4] and [5].

In [6], the highway scenario is replaced by a circular bus route within a campus, which however implies again easily predictable vehicular contacts: indeed, the focus of the work is on the prefetching and multi-hop transfer of data at each individual AP, while carry&forward communications are not taken into consideration. Conversely, [7] and [8] examine urban environments. In [7], the authors study the upload of small-sized contents from vehicles to roadside gateways, rather than the large downloads we target. The work in [8] considers instead data transfers to vehicular users in grid-like road topologies, but the focus is on the problem of optimizing direct communications between cars and infrastructure, without taking into account cooperation among mobile users. Recently, the performance bounds of vehicular cooperative download in urban scenarios have been studied in [9]: there, however, the authors assume perfect knowledge of the car traffic and outline a centralized optimal solution, rather than the distributed practical techniques we envisage in this work.

As far as opportunistic data exchanges are concerned, the potential of such a networking paradigm in vehicular environments was first shown in [10], further explored in [11], [12], and exploited in [13], [14] among the others. However, most of these works focus on routing delay-tolerant information in vehicular networks, while none copes with the problem of cooperative download. Also, techniques for Medium Access Control [15] and network coding [16] that have been proposed for cooperative vehicular download are orthogonal to the problems we address, and could complement the solutions outlined in this paper.

Finally, since we study the impact of the infrastructure deployment on the cooperative download, our work also relates to the topic of AP placement in vehicular networks. In [17], the authors studied the impact of random AP deployments on data routing in urban road topologies: we will prove that such an approach is inefficient when targeting cooperative download. More complex solutions for the deployment of APs over road topologies have been proposed in [18], to favor delay-tolerant data exchange among vehicles, and in [19], for information dissemination purposes. However, the diverse goals in these works lead to in different approaches and results with respect to ours. More recently, the problem of AP placement to provide Internet access to vehicles has been addressed in [20], [21] and [22]. In all these works, however, the aim is to maximize vehicle-to-infrastructure coverage or contacts, and no cooperation among cars is considered.

III. Cooperative download

Let us first point out which are the major challenges in the realization of a vehicular cooperative download system within complex urban road environments. With reference to the transfer model proposed in Sec. I, we identify two main problems:

- *the selection of the carrier(s)*: contacts between cars in urban/suburban environments are not easily predictable. Idle APs cannot randomly or inaccurately select vehicles to carry data chunks, or the latter risks to be never delivered to their destinations. Choosing the right carrier(s) for the right downloader vehicle is a key issue in the scenarios we target;
- *the scheduling of the data chunks*: determining which parts of the content should be assigned to one or multiple carriers, and choosing in particular the level of redundancy in this assignment, plays a major role in reducing the probability that destination vehicles never receive portions of their files.

In the following, we first discuss the selection of carriers at the APs, proposing to leverage historical information on large-scale traffic flows to drive data transfer decisions. Then, we outline several solutions to the chunk scheduling problem, that are characterized by different levels of redundancy.

A. Carriers selection

The first problem we address is that of the selection of data chunk carriers at APs that are idle, i.e., that are currently not transferring data directly to vehicular downloaders. As previously discussed, these APs can opt to employ their spare airtime to delegate, to mobile users within range, portions of files being downloaded. Taking such a decision means to answer to two questions: (i) *which, among the vehicles in range of an idle AP, should be picked as carriers, if any?* and (ii) *which of the downloaders should these carriers transport data for?*

The key to the answers is to know in advance whether (and possibly when) one or more cars currently within coverage of an AP will meet a specific downloader vehicle, so to perform the selection that maximizes the download rate. Also, by choosing carriers depending on their future contacts, the destination of the data becomes constrained to the elected carriers, and the second question above is inherently solved along with the first one. However, assuming that the roadside infrastructure has perfect knowledge of the future route of each user is unrealistic, other than raising privacy issues. At the same time, the movement of individual vehicles over urban road topologies cannot be easily predicted as in unidimensional highways. We then adopt a probabilistic approach, by leveraging the fact that large-scale urban vehicular flows tend to follow common movement patterns [23], [24], [25]. More precisely, the solution we propose leverages *contacts maps*, that are built by exploiting historical data on contacts between car flows, and then used to estimate the meeting probability between downloaders and candidate data carriers.

Contacts map

We denote as p^k_{Aa} the k-th *production phase* of vehicle a with respect to AP A, i.e., the k-th of the disjoint time intervals during which vehicle a can steadily download data from A [26]. From a specific AP perspective, we tag production phases as *local* if they involve that particular AP: as an example, p^h_{Bb} is a local production phase for AP B, $\forall\, b, h$. On the other hand, we label as f^m_{ab} the m-th *forward phase* of vehicle b with respect to vehicle a, i.e., the m-th of the disjoint time intervals during which vehicle b can steadily forward data to vehicle a. Note that production and forward phases do not necessarily correspond to actual data transfers, but just to contacts which could be exploited for data transfers. We also use $t(\cdot)$ to indicate the time at which a production or forward phase starts, and $\Delta t(\cdot)$ to tag its duration. For production phases only, $\alpha(\cdot)$ denotes the general direction of movement[1] of the vehicle at the beginning of the production phase, and $v(\cdot)$ its speed at that same time. The notation is summarized in Fig. 2.

Structure. A contacts map is a data structure that provides an AP with information on the probability of contact between a vehicle involved in a local production phase and another vehicle. With reference to the example in Fig. 2, the contacts map at AP B allows B to know the probability of contact between the local vehicle b and the generic vehicle a. In particular, AP B knows that b started a local production phase p^h_{Bb} at time $t(p^h_{Bb})$, while moving with direction $\alpha(p^h_{Bb})$ and speed $v(p^h_{Bb})$; also, let us

[1]The general direction is obtained as the angle of movement between the location where the vehicle started its trip, and its current location. This represents a more reliable information than the instantaneous direction, and it is not harder to obtain from a GPS receiver than the latter.

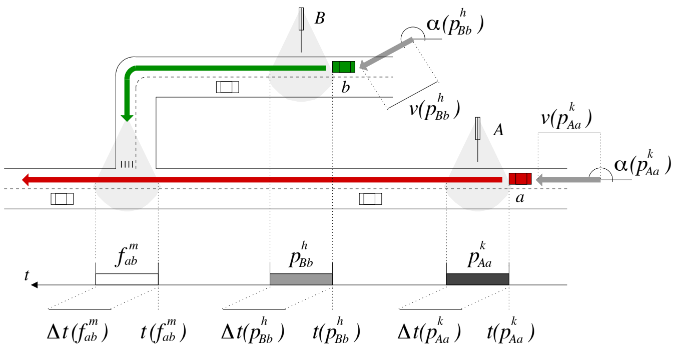


Fig. 2. Notation for contacts map structure and construction

assume B has been informed that a started a production phase p^k_{Aa} with AP A at time $t(p^k_{Aa})$, while moving with direction $\alpha(p^k_{Aa})$ and speed $v(p^k_{Aa})$. Then, the contacts map at B allows to associate the couple of production phases $\left(p^h_{Bb}, p^k_{Aa}\right)$ to historical data on the encounters between vehicles that have previously generated production phases at the two APs B and A with timings and mobility similar to those of b and a. We stress that such historical data refers to any two vehicles with movement patterns akin to those of b and a, and not to b and a only: thus, the data concerns vehicular flows rather than individual couples of cars.

More formally, a contacts map is a set of one-to-one associations between *keys*, that encode the significant characteristics of two production phases, and *values*, that store the contacts properties for all couples of production phases that share such characteristics. The *key* for two generic production phases p^h_{Bb} and p^k_{Aa} is a vector $\Bbbk(p^h_{Bb}, p^k_{Aa})$ of the form

$$\left[A, \lfloor\frac{t(p^h_{Bb}) - t(p^k_{Aa})}{\delta t}\rfloor, \lfloor\frac{\alpha(p^h_{Bb})}{\delta\alpha}\rfloor, \lfloor\frac{\alpha(p^k_{Aa})}{\delta\alpha}\rfloor, \lfloor\frac{v(p^h_{Bb}) - v(p^k_{Aa})}{\delta v}\rfloor\right],$$

where $\delta\alpha$, δt and δv are the units (in degrees, seconds, and meters/second, respectively) used to discretize angles, times and speeds. A couple of production phases is thus characterized by the identity of the AP involved in the second production phase, the time elapsed between the start of the two production phases, the direction of the two vehicles at the beginning of the respective production phases, and the difference between their speeds at that same time. We remark that the identity of the other AP is not necessary: as detailed next, the first production phase is always a local one. A *value* is instead a vector of four fields:

1) n_{opps}, the number of contact opportunities, i.e., the number of times that the AP observed a couple of production phases with characteristics as from the associated key;
2) n_{cons}, the number of actual contacts, i.e., the number of times that vehicles from the aforementioned couples of production phases actually generated a forward phase;
3) t_{del}, the average time elapsed between the start of the last production phase and the start of the forward phase, if any of the latter has ever occurred;
4) t_{dur}, the average duration of the forward phase, if any has ever occurred.

It is to be noticed that each AP builds its own contacts map, in which it stores only values associated to keys where the first production phase, as already said, is a local one. As an example, an AP B will only store values for keys of the type $\Bbbk(p^h_{Bb}, p^k_{Aa})$, $\forall h, b, k, A, a$. The rationale is that local production phases represent the vehicle-to-infrastructure contacts that an AP can exploit for carriers selection, and are thus the only an AP is interested to record data for.

Construction. The steps for the construction of the contacts map at an AP are best described by means of an example, so we consider once more the situation depicted in Fig. 2. When the production phase p^k_{Aa} starts, the AP A logs the time $t(p^k_{Aa})$, the relative vehicle identifier a, its general direction $\alpha(p^k_{Aa})$, and its current speed $v(p^k_{Aa})$. This information is shared, via the wired backbone, with other APs in the same area, and updated, when the production phase ends, with the information on the duration $\Delta t(p^k_{Aa})$. This way, when the production phase p^k_{Aa} terminates, AP B has memory of the event, including all related details. Similarly, at the beginning of p^h_{Bb}, B records and shares with other APs identical information on the production phase, which is then updated at the end of p^h_{Bb}.

Regarding the exchange of data among the APs, we emphasize that a generic AP does not need to be informed about the vehicle-to-infrastructure contacts occurring at all other APs in the network. Indeed, if two APs are too far apart, they can avoid sharing production phase data, as the excessive distance makes contacts too hard to predict and leads to unbounded carry&forward transfer delays. Thus, in order to limit the traffic on the wired backbone and guarantee system scalability, the reciprocal exchange of information about production phases can be constrained to APs within limited geographical distance[2].

Proceeding in our example, at the end of p^h_{Bb}, AP B, as every other AP does at the end of its own local production phases, checks whether p^h_{Bb} can be considered as an opportunity for cooperative download with respect to other production phases it is aware of, i.e., if another production phase is *compatible* with p^h_{Bb}. We will discuss production phases compatibility later in this section; for the moment, let us assume that p^k_{Aa} is compatible with p^h_{Bb}. Then, B looks in its local contacts map for the value associated to key $\Bbbk(p^h_{Bb}, p^k_{Aa})$. If an entry is not found, it is created; in both cases, the n_{opps} field in the entry is incremented.

Let us now assume that, later on, vehicle b meets vehicle a, generating the forward phases f^m_{ab} and f^m_{ba}. We focus on the first one, as it is that of interest in our example. Both vehicles record the forward phase start time $t(f^m_{ab})$, as well as the other vehicle identifier. Upon loss of contact, a and b also log the forward phase duration $\Delta t(f^m_{ab})$. These same cars upload these information, together with similar data on all other forward phases they have experienced, to the next AP they encounter, which will again share them with the APs in the area.

When AP B is notified of the forward phase f^m_{ab}, it tries to understand if f^m_{ab} can be related to any of the opportunities it has previously recorded. Thus, B scans its database for local production phases compatible with f^m_{ab}; once more, we will discuss the compatibility between production and forward phases next. Assuming that p^h_{Bb} satisfies the compatibility constraint, B then looks for production phases of vehicle a, with any AP, that are compatible with p^h_{Bb}. B finds again p^k_{Aa}, and thus finally relates f^m_{ab} to the couple of production phases (p^h_{Bb}, p^k_{Aa}). At this point, B retrieves the value associated to the key $\Bbbk(p^h_{Bb}, p^k_{Aa})$, and updates the n_{cons}, t_{del} and t_{dur} fields. The first is incremented by one, to record that the opportunity previously stored actually generated a forward phase. The second and the third elements are

[2] A thorough study of the management of control messages over the backbone of the infrastructured network is out of the scope of this paper. In our tests we imposed a maximum distance of 10 km among data-sharing APs, so to bound the delay between production and forward phases at approximately 15 minutes, given an average vehicular speed of 20 km/h in urban areas.

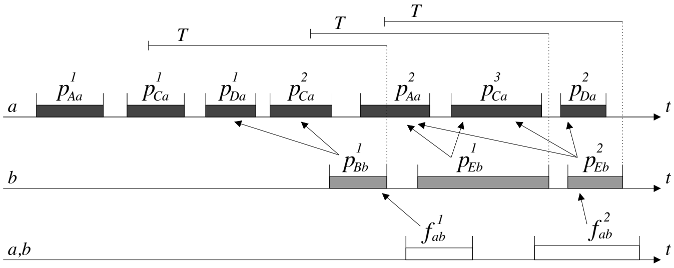


Fig. 3. Example of phases compatibility.

updated using samples $t(f^m_{ab}) - t(p^h_{Bb})$ and $\Delta t(f^m_{ab})$, respectively. The way these last updates are performed depends on the desired level of detail on the vehicle-to-vehicle contact: in our case, we opted for keeping track of the mean of the samples.

Phases compatibility. Phases compatibility rules determine when two production phases generate an opportunity for cooperative download, as well as when a forward phase can represent a contact for a local production phase. These rules formally relate phases in a way that avoids inconsistencies in the resulting contacts maps, an event otherwise common, especially when considering that phases often overlap in time and/or refer to a same AP (i.e., it can be that A and B in all previous discussions are indeed the same AP). We first introduce the set of rules for the compatibility of a local production phase with respect to a forward phase. A local production phase p^h_{Bb} is said to be compatible with a forward phase f^m_{ab} if the following conditions are verified:

1) the forward phase has ended after the end of the local production phase, or
$$t(f^m_{ab}) + \Delta t(f^m_{ab}) > t(p^h_{Bb}) + \Delta t(p^h_{Bb}),$$
as b must receive the data from B before it can forward them to a. Note that the position of the subscripts already implies that, for the phases to be compatible, the same vehicle b must realize the production phase with B and be the potential carrier in the forward phase;
2) the forward phase is the first involving a and b and satisfying rule 1) above to have terminated after the end of the local production phase, or
$$\nexists n \mid t(f^n_{ab}) + \Delta t(f^n_{ab}) > t(p^h_{Bb}) + \Delta t(p^h_{Bb}),\quad t(f^n_{ab}) < t(f^m_{ab}),$$
which guarantees that at most one forward phase is associated to each production phase;
3) the local production phase at B is the last, involving b and satisfying rule 1) above, that started before the forward phase end, or
$$\nexists n \mid t(f^m_{ab}) + \Delta t(f^m_{ab}) > t(p^n_{Bb}) + \Delta t(p^n_{Bb}),\quad t(p^n_{Bb}) > t(p^h_{Bb}),$$
which guarantees that at most one production phase is associated to each forward phase.

Then, we introduce the rules that define the compatibility between two production phases. A production phase p^k_{Aa} is said to be compatible with a local production phase p^h_{Bb} if the following conditions are verified:

4) the first production phase has ended before the end of the local production phase, or
$$t(p^h_{Bb}) + \Delta t(p^h_{Bb}) > t(p^k_{Aa}) + \Delta t(p^k_{Aa}),$$
which accounts for the fact that an AP can only destine carry&forward data to production phases it is aware of;
5) the first production phase has ended at most a time T before the end of the local production phase, or
$$t(p^h_{Bb}) + \Delta t(p^h_{Bb}) - t(p^k_{Aa}) - \Delta t(p^k_{Aa}) \leq T,$$
that avoids considering obsolete production phases;
6) the first production phase is the last, involving a and A as well as satisfying rules 1) and 2) above, to have ended before the end of the local production phase
$$\nexists n \mid 0 < t(p^h_{Bb}) + \Delta t(p^h_{Bb}) - t(p^n_{Aa}) - \Delta t(p^n_{Aa}) \leq T,\quad t(p^n_{Aa}) > t(p^k_{Aa}),$$
which guarantees that at most one production phase involving same vehicle/AP couple is associated to each local production phase.

Fig. 3 provides some examples of phases compatibility as from the rules listed above. There, the first timeline depicts the time intervals during which a vehicle a experiences production phases with APs A, C and D, while the second timeline reports the sequence of production phases of car b with APs B and E. Finally, the last timeline shows when forward phases of b to a occur.

Arrows from the second to the first timeline indicate which production phases of a are compatible with those of b. As an example, p^1_{Aa} is not compatible with p^1_{Bb} because it occurred too early in time (i.e., it ended more than a time T before the end of p^1_{Bb}, see rule 5 above), while p^2_{Aa} is not compatible with p^1_{Bb} because it is not yet concluded when p^1_{Bb} ends (rule 4). Similarly, p^1_{Ca} is not compatible with p^1_{Bb} since a more recent production phase between a and C, i.e., p^2_{Ca}, is compatible with p^1_{Bb} (rule 6). Conversely, p^2_{Ca} and p^1_{Da} satisfy all the compatibility conditions, and are thus compatible with p^1_{Bb}.

Forward phase compatibilities are shown as arrows from the third to the second timeline. We can notice that f^2_{ab} is compatible with p^2_{Eb} but not with p^1_{Eb}, since p^1_{Eb} is not the last production phase involving b and E to have ended before the end of f^2_{ab} (rule 3 above), while p^2_{Eb} is. Moreover, f^2_{ab} is not compatible with p^1_{Bb}, because another forward phase of b to a, i.e., f^1_{ab}, already concluded after the end of p^1_{Bb} (rule 2).

Carriers selection algorithms

Contacts maps can be exploited by APs to select local cars as data carriers in the cooperative download process, by retrieving their contact probability estimates with respect to downloader vehicles. Firstly, it is necessary that APs know which cars in their surroundings are interested in some content. Thus, every time a downloader vehicle starts a production phase, the fact that it is requesting data, as well as the nature of the desired content, is attached to the usual information on the production phase that the local AP shares with other APs. This way, each AP can track downloaders through their production phases history.

Thanks to such knowledge, an AP that has active local production phases can compute the *delivery potential* $\mathbb{p}_a$ resulting from electing one (or some, or all) of the local vehicles as carrier(s) for data destined to a specific downloader vehicle a. The delivery potential is obtained as the sum of the individual contact probabilities $\mathbb{p}_b$, derived from assigning data for the downloader

```
01 set $\mathbb{p}$ equal to $\mathbb{P}_{min}$
02 for each downloader vehicle $a$ do
03   if $a$ is in range of $B$ do
04     if $a$ is closer to $B$ than previous direct downloaders do
05       select $a$ as destination for direct transfer
06       select no vehicles as carriers for carry&forward transfer
07       set $\mathbb{p}$ equal to $\infty$
08     done
09   else
10     for each production phase $p^k_{Aa}$ of $a$ do
11       until all on-going local production phases are not processed do
12         update delivery potential $\mathbb{p}^k_{Aa}$
13         update carriers list $\bar{\mathbb{c}}^k_{Aa}$
14       done
15       if $\mathbb{p}^k_{Aa}$ is the highest potential computed for $a$ do
16         set $\mathbb{p}_a$ equal to $\mathbb{p}^k_{Aa}$
17         set $\bar{\mathbb{c}}_a$ equal to $\bar{\mathbb{c}}^k_{Aa}$
18       done
19     done
20     if $\mathbb{p}_a$ is strictly higher than $\mathbb{p}$ do
21       select $a$ as destination for carry&forward transfer
22       select vehicles in $\bar{\mathbb{c}}_a$ as carriers for carry&forward transfer
23       set $\mathbb{p}$ equal to $\mathbb{p}_a$
24     done
25   done
26 done
```

Fig. 4. Pseudocode for carriers selection at AP B

```
get next on-going local production phase $p^h_{Bb}$
set $\mathbb{p}_b$ equal to a random value $\in (0, 1]$
add $\mathbb{p}_b$ to delivery potential $\mathbb{p}^k_{Aa}$
add $b$ to carriers list $\bar{\mathbb{c}}^k_{Aa}$
mark local production phase $p^h_{Bb}$ as processed
```

Fig. 5. Blind pseudocode for $\mathbb{p}^k_{Aa}$, $\bar{\mathbb{c}}^k_{Aa}$ update

a to each elected local carrier b[3]. The process is repeated for each known downloader car, and, in the end, the downloader vehicle associated with the highest delivery potential $\mathbb{p}$ is chosen as the target of a carry&forward transfer through local carriers that contributed to $\mathbb{p}$. Note that $\mathbb{p}$ is a potential and not a probability: indeed, $\mathbb{p}$ can be higher than one to counter uncertainties in probability estimates.

The framework for carriers selection run at a generic AP B is shown as pseudocode in Fig. 4. There, priority is always given to direct data transfers to downloader cars, and fairness among them is provided by always picking the vehicle that is the closest to the AP. The parameter $\mathbb{P}_{min}$ controls the minimum delivery potential required to attempt cooperative download through local carriers. The value of such parameter (line 01 in Fig. 4), together with the way the delivery potential $\mathbb{p}^k_{Aa}$ associated to the downloader production phase p^k_{Aa} and its relative carriers list $\bar{\mathbb{c}}^k_{Aa}$ are updated (lines 12 and 13 in Fig. 4), distinguish the following carriers selection algorithms.

The **Blind** carriers selection algorithm aims at fully exploiting the airtime available at APs, by delivering data to all available local carriers whenever possible. This algorithm does not make use of the contacts map, but randomly chooses a downloader car as the destination of the data: we thus employ it as a benchmark for the other schemes. The pseudocode for potential and carriers list updating is outlined in Fig. 5, while $\mathbb{P}_{min}$ is set to 0, so that cooperative download is always attempted when at least one local carrier is present.

The **p-Driven** carriers selection algorithm is a probability-driven version of the Blind algorithm above. It again tries to exploit as much as possible the APs wireless resources, but this time cooperative download destinations are selected according to the delivery potential obtained from the contacts map.

[3]Thanks to the broadcast nature of the wireless channel, a single transmission is sufficient to transfer the same data to all elected local carriers.

```
get next on-going local production phase $p^h_{Bb}$
get key $\mathbb{k}(p^h_{Bb}, p^k_{Aa})$
if a contacts map entry for such key exists do
  get relative value $\{n_{opps}, n_{cons}, t_{del}, t_{dur}\}$
  set $\mathbb{p}_b$ equal to $\frac{n_{cons}}{n_{opps}}$
  add $\mathbb{p}_b$ to delivery potential $\mathbb{p}^k_{Aa}$
  add $b$ to carriers list $\bar{\mathbb{c}}^k_{Aa}$
done
mark local production phase $p^h_{Bb}$ as processed
```

Fig. 6. p-Driven pseudocode for $\mathbb{p}^k_{Aa}$, $\bar{\mathbb{c}}^k_{Aa}$ update

```
get next on-going local production phase $p^h_{Bb}$
get key $\mathbb{k}(p^h_{Bb}, p^k_{Aa})$
if a contacts map entry for such key exists do
  get relative value $\{n_{opps}, n_{cons}, t_{del}, t_{dur}\}$
  set $\mathbb{p}_b$ equal to $\frac{n_{cons}}{n_{opps}}$
  if $\mathbb{p}_b$ is equal to or greater than $\mathbb{P}_{ind}$ do
    add $\mathbb{p}_b$ to delivery potential $\mathbb{p}^k_{Aa}$
    add $b$ to carriers list $\bar{\mathbb{c}}^k_{Aa}$
  done
done
mark local production phase $p^h_{Bb}$ as processed
```

Fig. 7. p-Constrained pseudocode for $\mathbb{p}^k_{Aa}$, $\bar{\mathbb{c}}^k_{Aa}$ update

As a matter of fact, carry&forward data is consigned by each AP to all available local vehicles, and destined to the downloader vehicle which maximizes the sum of its contact probabilities with all the local carriers, as detailed in the pseudocode of Fig. 6. We stress that non-compatible production phases generate keys that are not present in the contacts map, and are thus not considered for cooperative download. As the p-Driven algorithm is designed to exploit carry&forward whenever there is a minimal chance of delivery, $\mathbb{P}_{min}$ is set to 0: this allows cooperative download even in presence of very small delivery potentials. Exploiting contacts maps, the p-Driven scheme is however expected to be more precise than the Blind one in the selection of carriers.

The **p-Constrained** carriers selection algorithm builds on top of the p-Driven scheme, adding constraints on probabilities, as from the pseudocode in Fig. 7. In particular, local vehicles with individual contact probability $\mathbb{p}_b$ lower than $\mathbb{P}_{ind} > 0$ are not considered for data carrying, and $\mathbb{P}_{min}$ is set to a value higher than 0, so that downloader vehicles with delivery potential $\mathbb{p}_a$ lower than $\mathbb{P}_{min}$ are discarded. Thanks to the lower bounds on individual probability and delivery potential, the p-Constrained algorithm is expected to further increase the delivery precision and reduce the load at APs with respect to the p-Driven scheme. However, quality could come at cost of quantity, as the thresholds may hinder potentially successful cooperation among vehicles.

The **(p,t)-Constrained** carriers selection algorithm adds time constraints to the probability bounds of the p-Constrained scheme. It introduces a distributed database $\bar{\mathbb{t}}_a$, maintained for each active downloader vehicle a by APs in a same area, controlling what portions of a's airtime are assigned to which specific carriers[4]. As shown in the pseudocode in Fig. 8, the (p,t)-Constrained algorithm processes local vehicles b in decreasing order of contact probability with the downloader car a, skipping those with probability lower than $\mathbb{P}_{ind}$ (lines 02 to 16 in Fig. 8). Every time the unprocessed local vehicle with maximum contact probability $\mathbb{p}_{max}$ is processed, the algorithm exploits information on the average time to contact (t_{del}) and contact duration (t_{dur}) to predict the time interval during which the local vehicle will meet

[4]We recognize that maintaining such database can pose synchronization and consistency issues, whose management is out of the scope of this paper. We however note that we do not require frequent updates or high accuracy in $\bar{\mathbb{t}}_a$, since the update periodicity is in the order of seconds and errors in the database are overshadowed by inaccuracy in the contact estimation.

```
01 set $\mathbb{p}_{max}$ equal to $\mathbb{P}_{ind}$
02 for each on-going local production phase $p^h_{Bb}$ do
03   if $p^h_{Bb}$ is marked as processed do
04     continue
05   done
06   get key $\mathbb{k}(p^h_{Bb}, p^k_{Aa})$
07   if a contacts map entry for such key exists do
08     get relative value $\{n_{opps}, n_{cons}, t_{del}, t_{dur}\}$
09     set $\mathbb{p}_b$ equal to $\frac{n_{cons}}{n_{opps}}$
10     if $\mathbb{p}_b$ is equal to or greater than $\mathbb{p}_{max}$ do
11       set $p$ equal to production phase $p^h_{Bb}$
12       set $v$ equal to production phase $p^h_{Bb}$ vehicle $b$
13       set $\mathbb{p}_{max}$ equal to $\mathbb{p}_b$
14     done
15   done
16 done
17 if $\mathbb{p}_{max}$ is equal to $\mathbb{P}_{ind}$ do
18   mark all unmarked local production phases as processed
19 else
20   get key $\mathbb{k}(p, p^k_{Aa})$
21   get relative value $\{n_{opps}, n_{cons}, t_{del}, t_{dur}\}$
22   for each time step $t \in \left[\lfloor \frac{t(p)+t_{del}}{\mathbb{T}} \rfloor, \lfloor \frac{t(p)+t_{del}+t_{dur}}{\mathbb{T}} \rfloor\right]$ do
23     if $\mathbb{p}_{max}$ is lower than or equal to $\mathbb{P}_{min} - \bar{\mathbb{t}}_a(t)$ do
24       add $\mathbb{p}_{max}$ to delivery potential $\mathbb{p}^k_{Aa}$
25       add $v$ to carriers list $\bar{\mathbb{c}}^k_{Aa}$
26       set $\bar{\mathbb{t}}_a(t)$ equal to $min\{\bar{\mathbb{t}}_a(t) + \mathbb{p}_{max}, \mathbb{P}_{min}\}$
27       continue
28     done
29   done
30   mark local production phase $p$ as processed
31   if $\mathbb{p}^k_{Aa}$ is equal to or greater than $\mathbb{P}_{min}$ do
32     mark all unmarked local production phases as processed
33   done
34 done
```

Fig. 8. (p,t)-Constrained pseudocode for $\mathbb{p}^k_{Aa}$, $\bar{\mathbb{c}}^k_{Aa}$ update

the downloader car a. Then, it discretizes time with step $\mathbb{T}$, and tries to fit the estimated contact probability $\mathbb{p}_{max}$ in one of the time steps within the aforementioned time interval (lines 20 to 30 in Fig. 8).

The process is terminated when either the required delivery potential $\mathbb{P}_{max}$ has been reached (lines 31 to 33 in Fig. 8), or no more local vehicles are available (lines 17 to 19 in Fig. 8). In the second case, the delivery potential constraint is not fulfilled, as $\mathbb{P}_{min}$ is higher than zero, and thus no carriers can be selected for the current production phase p^k_{Aa} (see line 20 in Fig. 4). The (p,t)-Constrained algorithm therefore employs information about contact times to improve the delivery precision, reducing the data carriers involved in the cooperative download.

B. Chunk scheduling

Upon selection of a destination for the carry&forward transfer, jointly with the associated local carriers, an AP must decide on which portion of the data the downloader is interested in is to be transferred to the carriers. To that end, we assume that each content is divided into *chunks*, i.e., small portions of data that can be transferred as a single block from the AP to the carriers, and then from the latter to the destination. Since a same chunk can be transferred by one or multiple APs to one or more carriers, the chunk scheduling problem yields a tradeoff between the reliability (i.e., the probability that a downloader will receive at least one copy of a chunk) and the redundancy (i.e., how many copies of a same chunk are carried around the road topology) of the data transfer. Next, we introduce three chunk scheduling schemes that embody growing levels of redundancy, and that are thus intended to provide increasing communication reliability.

The **Global** chunk scheduling assumes that APs maintain per-vehicle distributed chunk databases, similar to the time databases $\bar{\mathbb{t}}_a$ introduced before[5]. These databases store information on which chunks have already been scheduled for either direct or carry&forward delivery to each downloader.

The Global scheme, whose flow diagram is depicted in Fig. 9(a), completely distributes the chunk scheduling among APs, since it forces an AP to pick a new, unscheduled chunk every time it performs a direct or carry&forward transfer. In other words, each chunk is scheduled for transfer just once in the entire network. We stress that, even then, multiple carriers can be given the same chunk, as carriers selection algorithms can (and usually do) identify more than one vehicle for a single carry&forward transfer.

The **Hybrid** chunk scheduling, in Fig. 9(b), allows overlapping between carry&forward transfers scheduled by different APs. Indeed, in case of a data transfer to carriers, an AP picks the first chunk that *it* has not yet scheduled, ignoring the carry&forward scheduling at the other APs. Conversely, non-overlapping scheduling is still enforced for direct chunk transfers: every time it has to deliver some portion of a content to a downloader, an AP always selects a new chunk, not yet scheduled by any other AP in the region. The Hybrid scheme is thus implicitly more redundant that the Global one, as different APs independently delegate carriers for a same data chunk. Also, note that the Hybrid scheme does not need the aforementioned per-vehicle chunk databases. As a matter of fact, it commends that overlapping is avoided only for direct transfers, that however occur during contacts between APs and the downloader vehicle: as a consequence, the chunk scheduling history can be easily maintained at vehicles, and communicated to the current AP at the beginning of the local production phase.

The **Local** chunk scheduling is similar to the Hybrid scheme, since different APs can schedule the same chunks when delegating data to carriers. However, as shown in Fig. 9(c), it also allows overlapping between direct and carry&forward transfers. An AP can thus directly transfer to a downloader within range chunks that were already scheduled, but through a carry&forward delivery. Namely, an AP can employ a direct transfer to a downloader car to fill gaps in its chunk list. The Local scheduling is thus the most redundant among the schemes we propose, trading some cooperative download potential for increased reliability in data delivery.

IV. Evaluation scenarios

In order to evaluate the cooperative download mechanisms outlined in the previous sections, we consider several large-scale vehicular traffic scenarios, that are representative of real-world road topologies. We also take into account different deployments of APs, that, as we will show, have a major impact on the download performance.

A. Vehicular mobility

We selected real-world road topologies from the area of Zurich, Switzerland, to assess the performance of the cooperative download solutions presented in the previous sections. This choice was mainly driven by the availability of large-scale microscopic-level traces of the vehicular mobility in the region, from the CS Department of ETH Zurich [27]. The simulation techniques and mobility models employed to generate the traces allow to reproduce vehicular movements over very large road topologies,

[5]The same observations on database maintenance apply here as well.

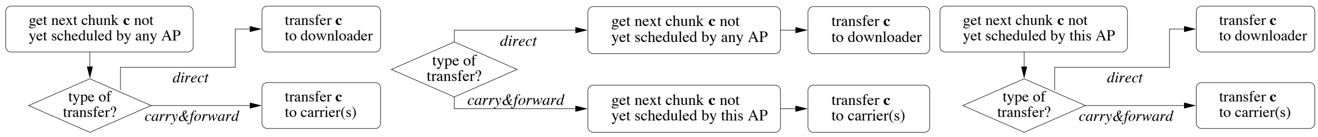
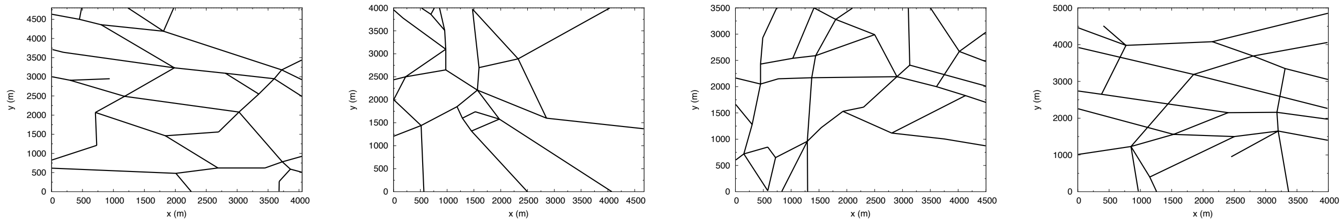


Fig. 9. Flow diagrams of the (a) Global, (b) Hybrid, and (c) Local chunk scheduling algorithms



(a) West Zurich (b) Central Zurich (c) North Zurich (d) Schlieren

Fig. 10. The road topologies considered in our study, representing urban (b), suburban (a,c), and rural (d) areas in the canton of Zurich, Switzerland

yet with a good degree of precision [28]. More precisely, the traces replicate macroscopic patterns of real-world vehicular traffic flows, made up of thousand of cars, as well as microscopic behaviors of individual drivers in urban environments, such as pauses at intersections that depend on roads capacity and congestion. This macro- and micro-mobility realism is important in our study, since, on the one hand, we exploit large-scale properties of urban vehicular mobility in designing the cooperative download system, and, on the other, realistic small-scale mobility is required to reproduce vehicle-to-vehicle and vehicle-to-AP networking interactions. We stress that the mobility traces we employed only reproduce important traffic arteries in each scenario, while they do not consider movements over minor streets. This, however, does not impact our study, since the traffic over minor roads is too sporadic to deserve the deployment of dedicated APs, and does not provide significant opportunities for collaboration among vehicles.

We focused on four scenarios, representing urban, suburban, and rural areas within and nearby the city of Zurich. All the areas considered cover surfaces between 15 and 20 km^2, and frame several tens of kilometers of major roads, whose layouts are shown in Fig. 10. In particular, the Central Zurich scenario reproduces the downtown of Zurich, and it is thus characterized by dense traffic uniformly distributed over the road layout. The West and North Zurich scenarios are representative of suburban areas, where the traffic congestion is less evident than in the city center, but still present over a few major freeways that attract most of the vehicular mobility. Finally, the Schlieren scenario portrays the street topology around a town not far from Zurich, characterized by a sparse presence of vehicles over most roads in the area.

B. AP deployment

The placement of APs over the urban road topology has a major influence on the cooperative download architecture. In order to capture such an effect, we extend our analysis by considering diverse AP deployments over the different road topologies presented above. The goal of all the deployment strategies is to position, along a road topology, a given number N of APs; in our performance evaluation, we will discuss the impact of the value of N as well.

Under the **Random** AP positioning scheme, each point of the road topology has the same probability of being selected for the deployment of an AP. The resulting placement may be considered representative of a completely unplanned infrastructure [29], [30], and it is used in our performance evaluation as a baseline for the other deployment techniques. We emphasize that the results we will present for the Random positioning scheme were obtained by generating different deployments at each simulation run, so to avoid biases due to more or less favorable random AP distributions.

The **Density-based** AP deployment technique aims at maximizing the probability of direct data transfers from APs to downloader vehicles. To that end, this techniques places the APs at those crossroads where the traffic is denser. The rationale behind such choice is that the volume of direct downloads is proportional to the number of APs that a downloader vehicle encounters during its movement through the road topology. Since the identity of downloaders cannot be known in advance, the best option is to deploy APs at those locations that a generic vehicle will most probably visit along its route, i.e., the most congested intersections.

The **Cross volume-based** AP placement is designed to favor carry&forward transfers, by increasing the potential for collaboration among vehicles. This technique exploits the predictability of large-scale urban vehicular traffic flows, which are known to follow common mobility patterns over a road topology [23], [24], [25]. By studying such traffic dynamics, it is possible to determine the way vehicular flows spread over the streets layout and employ this information to guide the AP placement. In the remainder of this section, we introduce the concept of *cross volume* and employ it to determine the relative AP deployment strategy.

Let us imagine that the road topology is represented by a graph where vertices are mapped to intersections and edges to streets connecting them, as in Fig. 11. The graph is undirected, and an edge exists even if the corresponding road is one-way. Focusing on a particular edge i of the graph, we can track all traffic leaving such edge[6], in both directions, and draw a map of how vehicular flows (measured in vehicles/s) from i unfold over the road topology. We refer to these flows as the vehicular flows *generated* at i. As an example, in Fig. 11, the dark grey arrows depict flows generated at edge i. Different flows have different size, in vehicles/s, represented by their associated number (values in the example are only illustrative).

Let us now consider a generic edge $k \neq i$, and isolate the

[6]Note that we do not make any assumption on the origin of the traffic, that could thus be constituted of vehicles that started their trip from an intermediate point of the road, or that had previously arrived from a different road.

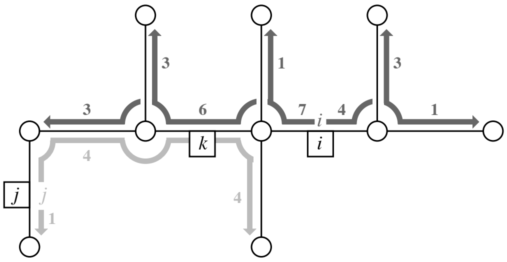


Fig. 11. Sample vehicular flows over a road topology graph. Flows generated at edge i are dark grey, while those generated at j are light grey. Assuming a travel time $= 1$ at all edges, the partial cross volume h_{ij}^{k} is equal to $min\{6, 4\} + min\{0, 0\} = 4$, while the crossing volume h_{ij} is $4 + 3 = 7$

flows passing in strict sequence through i and k. In this case, we distinguish the two directions of movement over k, and define two *traversing flows* from i to k:

- $\overrightarrow{f_{ik}}$, the vehicular flow generated at i and subsequently traversing k in the $\rightarrow$ direction;
- $\overleftarrow{f_{ik}}$, the vehicular flow generated at i and subsequently traversing k in the $\leftarrow$ direction.

We do not impose any rule in defining the two directions at k, which, e.g., could be based on vertices numbering or geographical coordinates. Our only concern is that, for each edge, the two directions are unambiguously identified. Also, the direction at i is not specified, and a traversing flow could have visited its generating edge in any direction (including both of them, if flows generated at i in opposite directions then merge at k in the same direction).

Traversing flows at an edge k can be translated to traversing volumes (measured in vehicles), by evaluating the average time vehicles spend to travel over the entire road segment corresponding to edge k. Also in this case, we can distinguish the two directions of movement, and define the two travel times $\overrightarrow{t_k}$ and $\overleftarrow{t_k}$, in the $\rightarrow$ and in the $\leftarrow$ directions, respectively. Considering again traversing flows from i, the two corresponding *traversing volumes* are

$$\overrightarrow{v_{ik}} = \overrightarrow{f_{ik}} \cdot \overrightarrow{t_k} \quad , \quad \overleftarrow{v_{ik}} = \overleftarrow{f_{ik}} \cdot \overleftarrow{t_k}$$

and represent the average number of vehicles that, having already visited i during their trip, travel over k in the $\rightarrow$ and $\leftarrow$ direction, respectively.

Let us now introduce a second group of flows, generated at an edge $j \neq i$, depicted in light grey in Fig. 11. The same consideration we made for flows generated at i are valid, and, picked an edge $k \neq j$, we can compute the traversing volumes from j to k, $\overrightarrow{v_{jk}}$ and $\overleftarrow{v_{jk}}$. By considering both sets of flows at once, we can define the *partial cross volume* of i and j at k, as

$$h_{ij}^{k} = \begin{cases} min\left\{\overrightarrow{v_{ik}}, \overleftarrow{v_{jk}}\right\} + min\left\{\overleftarrow{v_{ik}}, \overrightarrow{v_{jk}}\right\}, & \text{if } k \neq i, k \neq j \\ 0, & \text{otherwise.} \end{cases}$$

The partial cross volume h_{ij}^{k} corresponds to the amount of traffic from i and j that merges at edge k. We notice that h_{ij}^{k} only couples flows that travel on opposite directions over k, hence the name of partial *cross* volume. The rationale is the following: imagine one car that has visited edge i in its trip, and now enters edge k in the $\rightarrow$ direction: such a car thus belongs to the $\overrightarrow{f_{ik}}$ flow. Considering vehicles that come from edge j and now travel over k, our car can generate two types of contacts:

- with the $\overleftarrow{v_{jk}}$ vehicles that travel in the opposite direction. These contacts are certain, since u-turn are not allowed on road segments connecting two adjacent intersections;
- with the $\overrightarrow{v_{jk}}$ vehicles that travel in the same direction. However, contacts are not certain in this case: the relative speed is close to zero[7] and contacts mostly depend on the position of the $\overrightarrow{v_{jk}}$ vehicles over k, when our car enters the road segment. Indeed, even if it enters edge k while some of the $\overrightarrow{v_{jk}}$ vehicles are nearby, and thus generates contacts with them, our car will only meet that very small fraction of the overall $\overrightarrow{v_{jk}}$ vehicles.

Since there are $\overrightarrow{v_{ik}}$ cars such as the one considered above, we couple $\overrightarrow{v_{ik}}$ with $\overleftarrow{v_{jk}}$, as these volumes correspond to certain contacts, while we do not couple $\overrightarrow{v_{ik}}$ with $\overrightarrow{v_{jk}}$, as these volumes have an unpredictable (and, in most cases, negligible) contribution in terms of contacts. Such a coupling is performed by taking the minimum between the two facing traffic volumes, which is that imposing a more strict constraint on the number of encounters.

Finally, the concept of cross volume can be unbound from intermediate edges and related to couples of roads only. If I is the set of edges in the road topology graph, we define the *cross volume* of i and j as

$$h_{ij} = \begin{cases} \sum_{k \in I} h_{ij}^{k}, & \text{if } i \neq j \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

which implies that $h_{ij} = h_{ji} \geq 0, \forall i, j \in I$. The cross volume h_{ij} provides a measure of the potential for contact, and thus cooperation, over the entire road network, among vehicles leaving edges i and j. We can exploit such a measure to formalize the problem of the AP deployment. Let us assume that there are $|I| > N$ roads in the topology: the problem becomes that of picking N out of the $|I|$ edges of the graph for AP deployment. We associate to each edge i a binary decision variable x_i:

$$x_i = \begin{cases} 1, & \text{if an AP is deployed on the road mapped to } i \\ 0, & \text{otherwise,} \end{cases}$$

and refer to their vector as $x = \left\{x_1, \ldots, x_{|I|}\right\}$. Since opportunities for cooperation between vehicles are proportional to the crossing volume between each couple of edges, the APs should be positioned so to maximize the sum of crossing volumes between each pair of APs over the whole road topology. This leads to the formulation of the following mixed-integer quadratic programming (MIQP) problem:

$$\max_{x} \quad f(x) = \frac{1}{2} x' H x \tag{2}$$

$$s.t. \quad x_i \in \{0, 1\}, \quad \forall i \in I \tag{3}$$

$$\sum_{i \in I} x_i \leq N. \tag{4}$$

Here, that in Eq. 2 is the objective function to be maximized, with $H = \left\{h_{ij}\right\}$ being a $|I| \times |I|$ matrix in $\mathbb{R}$, filled with the crossing volumes computed for each couple of edges as in Eq. 1. Also, Eq. 3 states that x_i is a binary variable, $\forall i \in I$, whereas Eq. 4 bounds the overall number of APs to be deployed to N.

By solving the optimization problem, we obtain an AP deployment that, as originally stated, augments the opportunities for carry&forward transfers in the download process. We note that this formulation solves the AP deployment problem from a large-scale viewpoint, i.e., it allows to determine the roads where APs

[7]In the urban, suburban and rural scenarios we consider, the low speed limits and the reduced number of lanes hinder overtakings. This is proved by the fact that, in the scenarios in Sec. IV-A, we observed, on average, less than one overtaking per vehicle and per trip.

should be positioned. However, it does not specifies the exact location of each AP over the selected roads. In Sec. V-B, we will show that such small-scale deployment has a negligible impact on the performance of the system.

V. Performance evaluation

We conducted an extensive simulation campaign aimed at evaluating multiple aspects of cooperative download in non-highway vehicular networks. The computational complexity of the simulations, that reproduce the movement and network traffic of several thousands of vehicles at a time, prevented the use of a traditional network simulator, such as *ns-2*. Instead, we developed a dedicated simulator [31], which employs the mobility extracted from the Zurich traces, models a random access channel contention, and implements all the cooperative download techniques previously presented, but replaces the traditional packet-level simulation with a more scalable chunk-level one, avoiding the detailed reproduction of the entire network stack at each node.

In all our tests, a "best case" performance reference is provided by an **Oracle** carriers selection algorithm. The Oracle scheme assumes that APs have a perfect knowledge of the future trajectories of all vehicles, in terms of both routes and timings. This information is exploited during production phases at APs to foresee contacts between local and downloader vehicles, and thus to pick carriers that are certain to later meet their target downloader vehicle. The Oracle scheme exploits a per-downloader database, identical to that employed in the (p,t)-Constrained algorithm, to avoid that multiple carry&forward transfers are scheduled at a same time for the same downloader vehicle. Also, since vehicular contacts are known in advance, any redundancy in the scheduling of chunks is pointless: thus, only one carrier can be selected for the transfer of each carry&forward chunk, and we always couple the Oracle algorithm with the non-redundant Global chunk scheduling. Note that the Oracle carriers selection algorithm is not optimal in that it does not take into account that multiple carry&forward transfers can be scheduled at the same time for different downloaders that are within transmission range of each other. In such situation, since only one of the interfering downloaders can receive its chunks, part of the data cannot be delivered to the second downloader.

The main metrics we are interested in evaluating are:

- the *download rate*, i.e., the average file transfer speed experienced by downloader vehicles traveling through the scenario. Such rate is the aggregate of a *direct* rate, due to direct data downloads from APs, and a *cooperative* rate, due to carry&forward transfers. According to our simulation settings, listed next, the maximum download rate achievable by a vehicular user is 5 Mbps, which corresponds to the case of a car continuously receiving data during its whole trip through the simulation scenario;
- the *undelivered chunk ratio*, i.e., the average ratio of chunks that are not delivered to a downloader vehicle, computed over all those scheduled for that vehicle.

The system parameters were set for all simulations to $\delta t = 5s$, $\delta\alpha = 45°$, $\delta v = 5\frac{m}{s}$, $T = 500s$, $\mathbb{T} = 1s$, $\mathbb{P}_{min} = 2.5$, $\mathbb{P}_{ind} = 0.5$. If not stated otherwise, an average of ten downloader cars is present at the same time over the road topology, and a simple disc model is considered for signal propagation, so to fulfill the low complexity constraints imposed by the size of the simulations. The net application-level data transmission rate over the wireless channel, during both production and forward phases, is assumed equal to 5 Mbps for a transmission range of 100 m. The latter values are consistent with the outcome of real-world experiments on car-to-infrastructure [32] and car-to-car data transfers [33]. For each simulation we performed one run to train the framework (i.e., gather the data necessary to deploy the APs and to build the contacts map at each APs), and ten runs to collect statistics. Each run simulated around three hours of urban traffic, encompassing various vehicular density conditions. For all results we measured 99% confidence intervals, reported as error bars in the plots.

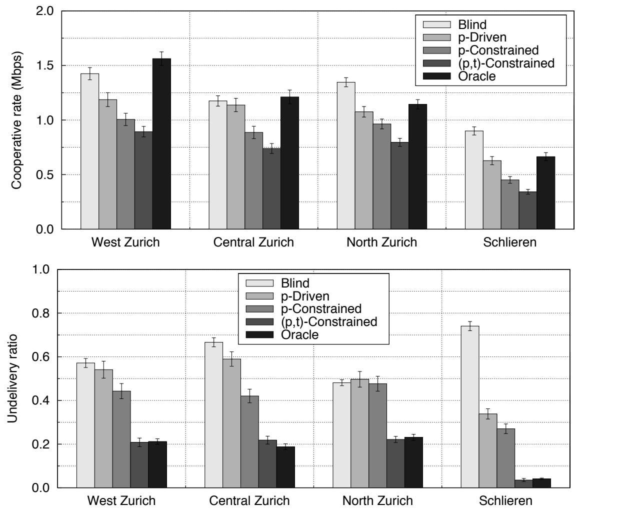


Fig. 12. Average cooperative download rate (top) and undelivery ratio (bottom) for different carriers selection schemes, in the four road topologies

A. Carriers selection and chunk scheduling

We first analyze the different carriers selection algorithms and chunk scheduling techniques, detailed in Sec. III-A and Sec. III-B. Since we are interested in a comparative evaluation of the all these schemes, we select a particular AP deployment scenario (6 APs positioned according to the Cross volume-based strategy), and focus on the carry&forward download performance (as direct downloads are not influenced by they way we select carriers or chunks). We will consider different AP deployments, and study their impact on direct download rates in Sec. V-B.

The average cooperative download rate, obtained from carry&forward transfers, and the mean undelivery ratio are depicted in Fig. 12. There, we report the results obtained under each road scenario by the different carrier selection scheme, when coupled with a Global chunk scheduling. We can notice how the Blind, p-Driven, p-Constrained, and (p,t)-Constrained algorithms yield, in this order and throughout all road scenarios, decreasing cooperative rates, as well as reduced undelivery ratios. Indeed, increasing the precision of carry&forward transfers also implies missing opportunities for vehicle-to-vehicle data exchanges.

The exact balance in the tradeoff between the cooperative download rate and the delivery precision varies with the road scenario. In suburban areas (West and North Zurich), a few major freeways attract most of the traffic in the region. On the one hand, such concentration results in a large number of vehicle-to-vehicle contacts, and thus in higher cooperative rates with respect to other scenarios. On the other hand, it favors inaccurate carriers selection algorithms, such as the Blind one, over more precise schemes, such as the p-Driven and p-Constrained ones: as a matter of fact, even randomly selected cars have a high probability of traveling

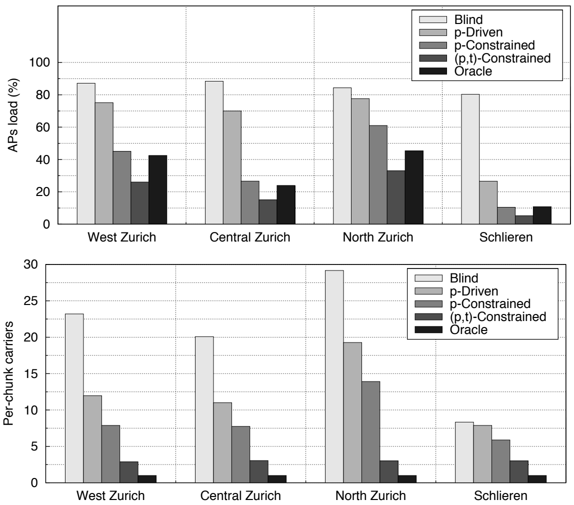


Fig. 13. Average AP load (top) and number of carriers ferrying a same chunk (bottom) for different carriers selection schemes, in the four road topologies

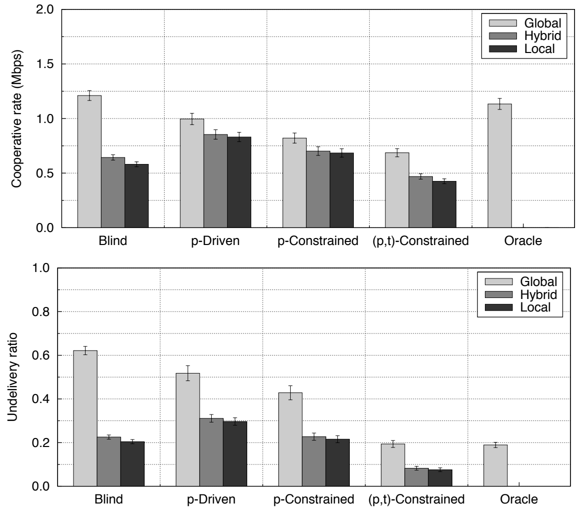
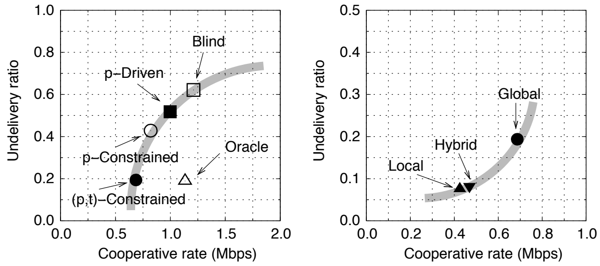


Fig. 14. Average cooperative download rate (top) and undelivery ratio (bottom) for different chunk scheduling algorithms, under the diverse carriers selection schemes. Results are averaged over all road topologies



(a) Carriers selection (b) Chunk scheduling

Fig. 15. Cooperative download rate / undelivery ratio working points for different carriers selection schemes (left, with Global chunk scheduling) and chunk scheduling algorithms (right, with (p,t)-Constrained carriers selection). Results are averaged over all road topologies

on the same road, and thus to meet each other. In the rural Schlieren scenario, the sparsity of traffic leads to reduced car-to-car contacts and thus lower cooperative rates. Also, the higher heterogeneity in the movement of vehicles, due to the absence of traffic-gathering roadways, makes the Blind scheme extremely imprecise in delivering chunks, with respect to contact map-based ones. Finally, the urban Central Zurich scenario presents traffic densities that are similar to those observable in the suburban areas, but dynamics that are closer to those of the rural case: such scenario thus yield high cooperative rates but undelivery ratios that significantly vary under the diverse algorithms.

However, we can notice that, no matter the road scenario considered, the (p,t)-Constrained carriers selection algorithm significantly outperforms all the other solutions in terms of undelivery ratio. Indeed, the precision achieved by the (p,t)-Constrained algorithm in the carry&forward chunk delivery is comparable to that of the Oracle algorithm. Although this latter scheme attains higher cooperative rates, we can state that the overall performance of the (p,t)-Constrained algorithm is not too far from that obtained through a perfect knowledge of future contacts among vehicles.

In Fig. 13, we also report, for the same combinations of carriers selection algorithms and road scenarios, the average load measured at the APs, i.e., the percentage of airtime used by an AP to transfer data to carriers, and the mean number of carriers ferrying a same chunk to a target downloader. From the plots it is clear how more precise algorithms result in a lower AP load and a smaller number of carriers per chunk. In other words, a higher precision in the carry&forward delivery of chunks yields a lower charge on the infrastructure and a reduced demand of resources by cooperating vehicles.

The different chunk scheduling schemes are then compared, in combination with every carriers selection algorithm, in Fig. 14. For the sake of brevity, the results are aggregated over all four road topologies: indeed, the same behaviors we previously discussed for the diverse mobility scenarios were observed also in this case. As a general comment, the increased redundancy introduced by the Hybrid and Local chunk scheduling leads, as one could expect, to lower cooperative rates but increased delivery precision. On a per-carriers selection algorithm basis, however, differences can be spotted. In particular, the Blind scheme suffers a dramatic 50% reduction in the cooperative rate when redundancy is increased in the chunk scheduling: indeed, scheduling multiple times the same chunks impairs the major strength of the Blind algorithm, i.e., the sheer number of unique chunks sent out for randomly selected downloaders. Conversely, when coupled with more redundant chunk schedulings, the algorithms based on contact maps enjoy a significant reduction in the undelivery ratio (from 35% to 65%, depending on the algorithm) at some smaller cost (15% to 30%) in terms of cooperative rate.

To conclude our analysis on carriers selection and chunk scheduling, we summarize the results in Fig. 15, showing the working points of each technique in the download rate/undelivery ratio space. The results, aggregated over all road topologies, evidence the tradeoff between the download volume and the reliability of the cooperative process. In particular, the non-linear distribution of the working points, evidenced by the grey curves in the plot, seem to indicate the (p,t)-Constrained carriers selection with Global chunk scheduling as the combination that better adapts to the different scenarios.

B. *Impact of the AP deployment*

The way the infrastructure is deployed can have a significant impact on the performance of the cooperative download framework. Thus, in this section, we evaluate how the strategy employed for the AP placement and the number of fixed stations influence the rates and undelivery ratios experienced by the downloader vehicles. In the light of the results in the previous section, we consider in the following a (p,t)-Constrained carriers

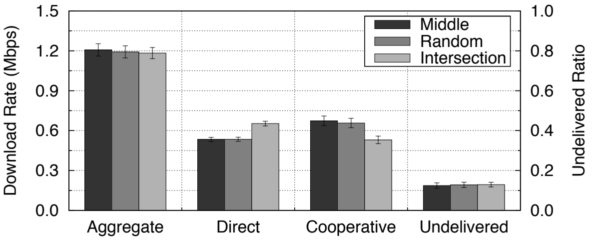


Fig. 16. Download rates and undelivered chunks ratios for different road-level placement policies, averaged over all road topologies

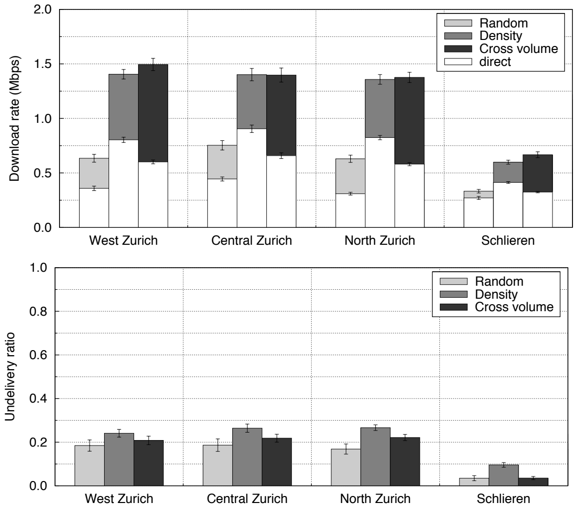


Fig. 17. Average download rates (top) and undelivery ratio (bottom) for different AP deployment strategies, in the four road topologies. Results refer to the (p,t)-Constrained carriers selection scheme

selection with Global scheduling as our default configuration.

We initially demonstrate the negligible impact of the small-scale deployment of APs under the Cross volume-based strategy outlined in Sec. IV-B. To that end, we compare three policies for the placement of APs over the roads resulting from the optimization problem. The *middle* policy places an AP equidistant from the intersections that end the selected road segment, the *intersection* policy deploys an AP at the most crowded of such intersections, while the *random* policy picks a random location over the selected road segment. Fig. 16 shows that the relevance of road-level deployment is minimal, as the three schemes achieve almost identical download rate and undelivered chunk ratio. The only notable difference is in that the intersection strategy favors direct downloads and penalizes cooperative ones: indeed, crossroads are characterized by high densities of slow vehicles, and placing APs there favors AP-to-vehicle transfers. At the same time, however, it deprives vehicles of transfer opportunities, since intersections also represent network clustering points where car-to-car contacts occur frequently [34], thus reducing the cooperative download rate. We thus consider APs to be deployed at the intermediate point of road segments, as this appears to bring a slight advantage over the other policies in terms of aggregate rate.

The different AP deployment strategies discussed in Sec. IV-B are compared in Fig. 17, in presence of 6 APs deployed in each road scenario. It is clear that a Random AP deployment results in the worst performance, as both the direct rate, i.e., the portion of the total download rate due to chunks directly retrieved from APs, and the cooperative rate, resulting instead from carry&forward transfers, are lower than in the other strategies. The reduced aggregate rate does not even bring an advantage in terms of undelivery ratio, which is comparable to that obtained under

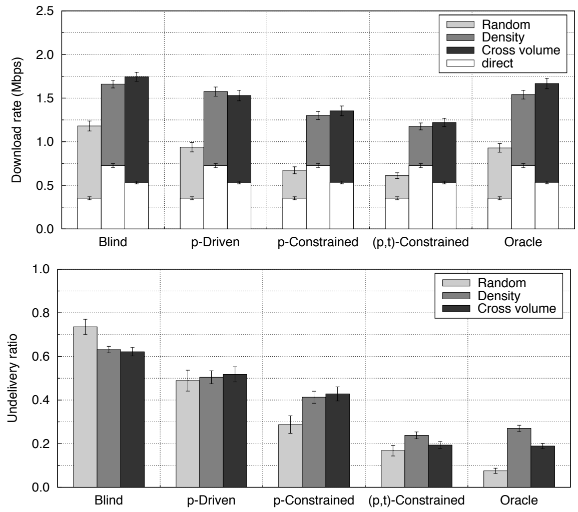


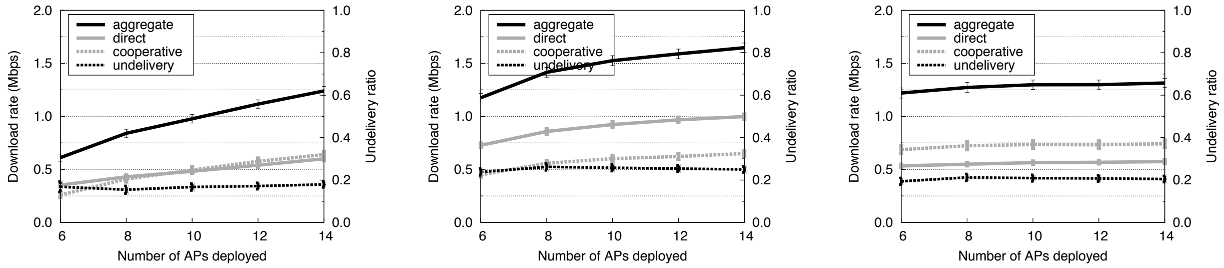
Fig. 18. Average download rates (top) and undelivery ratio (bottom) for different AP deployment strategies, under the diverse carriers selection algorithms. Results are averaged over all road topologies

other deployments. Conversely, the Density-based and the Cross volume-based AP placement lead to similar aggregate rates: however, in the former the contribution of direct transfers is significantly larger than in the latter, that instead mainly leverages the cooperation among vehicles. Such result is consistent with the objectives of the two strategies, that try to maximize, respectively, vehicle-to-infrastructure and vehicle-to-vehicle contacts. Interestingly, the Density-based strategy yields slightly higher undelivery ratios with respect to the other deployments: as already discussed, deploying APs at intersections renders many opportunistic contacts among vehicles unusable for planned data exchanges, an effect here exacerbated by the high densities of the junctions selected by the Density-based deployment.

Focusing on the proportion between direct and cooperative rates, we can observe that, depending on the road topology and deployment scenario, the carry&forward contribution typically varies between 35% and 60% of the total download rate, implying a remarkable 50% to 120% speedup in the download. As far as the road topologies are concerned, the same considerations made in the previous sections hold for the overall rates and undelivery ratios. Moreover, different scenarios do not appear to induce significant differences in the relative performance of each deployment, nor in terms of the proportion between direct and cooperative rates.

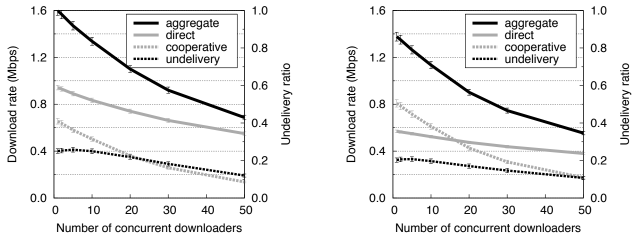
One may wonder how different AP placements affect carriers selection algorithms other than the (p,t)-Constrained one. In Fig. 18 we can observe that the strategy adopted in the deployment of the infrastructure has a very similar influence on all the algorithms. The cooperative rates achieved by the diverse schemes are consistent with those presented in the previous section, and thus have an even higher impact on the aggregate rate, with respect to the case of the (p,t)-Constrained algorithm studied above. Such an improvement comes, however, at a high cost in terms of undelivered chunks, with the exception of the Oracle scheme, which is clearly favored by its preemptive knowledge of future contacts.

Not only the position, but also the number of the APs can impact the performance of the vehicular cooperative download. In Fig. 19, we vary the number of APs deployed in each scenario, and show the average rates and undelivery ratio attained under the three AP placement strategies. Under a Random deployment,



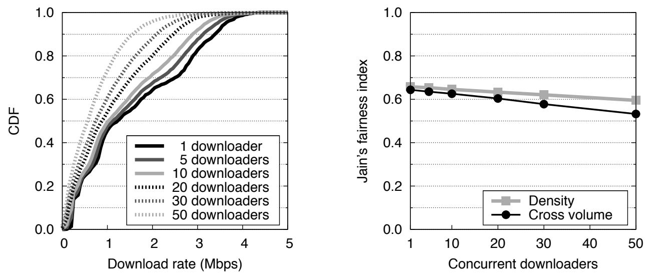
(a) Random (b) Density (c) Cross volume

Fig. 19. Average download rates and undelivery ratio for a varying number of APs, under the diverse deployment strategies. Results refer to the (p,t)-Constrained carriers selection scheme and are averaged over all road topologies



(a) Density (b) Cross volume

Fig. 20. Average download rates and undelivery ratio for a varying number of concurrent downloaders and 10 APs. Results refer to the (p,t)-Constrained carriers selection scheme and are averaged over all road topologies



(a) Download rate PDF (b) Jain's fairness index

Fig. 21. PDF of the aggregate download rate for a varying number of concurrent downloaders (left, for the Density-based deployment) and associated fairness (right). Results refer to the (p,t)-Constrained carriers selection scheme in presence of 10 APs, and are averaged over all road topologies

both direct and cooperative rates significantly increase as the infrastructure becomes more pervasive. Additional APs imply a higher number of direct transfer opportunities for the downloaders, which explains the improved direct rate. Similarly, denser APs are deployed closer to each other, making forward phases (i.e., contacts among vehicles) closer in time to production phases (i.e., contacts between vehicles and APs) and thus easier to predict.

The same behavior can be observed for the Density-based strategy, which, however, shows a slower rates growth for high numbers of APs. This is explained by the fact that, while in the Random deployment each new AP has a similar impact on the download process, in the Density-based case additional APs are located at intersections characterized by decreasing vehicular densities. Therefore, the rate gain brought by the presence of extra APs tends to be lower and lower. In the case of APs positioned according the Cross volume-based policy, the rates increase is significantly lower than in the other cases. As a matter of fact, the optimization problem behind this placement strategy struggles to find new locations that guarantee an increase in the crossing volumes, and thus picks positions that introduce very small gain in the download process. That is, the same reasoning made for the Density-based deployment holds, exacerbated, in the Cross volume-based case.

Finally, it is interesting to note that, under all deployments, as the number of APs grows the ratio between direct and cooperative rates is either unchanged or slightly shifted in favor of the latter. Moreover, the undelivery ratio remains constant. Thus, we can conclude that the positive impact of carry&forward transfers on the download process persists as the number of APs deployed on the road topology varies.

C. Scalability in the number of downloaders

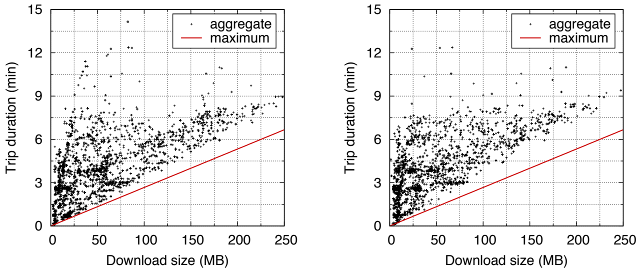
We evaluate the scalability of the cooperative download framework by increasing the number of downloaders concurrently traveling over each scenario, up to 50. This last value corresponds to a stressed network, where the vehicles actively downloading large-sized contents from the Internet jointly cover around one third of the whole road surface with their transmission ranges. Such a condition creates a significative contention for the channel, since it is very probable for two downloaders to interfere with each other, and thus impairs both direct and cooperative download.

This notwithstanding, in Fig. 20 we can observe that the system scales quite well, as each of 50 simultaneous users experiences an average aggregate rate that is only one half of the rate enjoyed by a lone downloader. We can also notice that cooperative transfers are affected more significantly than direct ones: since direct transfers always have a higher priority over carry&forward ones, the increasing presence of direct downloaders reduces the airtime that APs can dedicate to carriers, thus limiting the exploitation of the latter paradigm. The undelivery chunk ratio is instead positively affected by the downloaders' density: by increasing the spectrum of targets, it is easier for the APs to find one downloader that will encounter the local carriers with high probability.

D. Per-downloader performance analysis

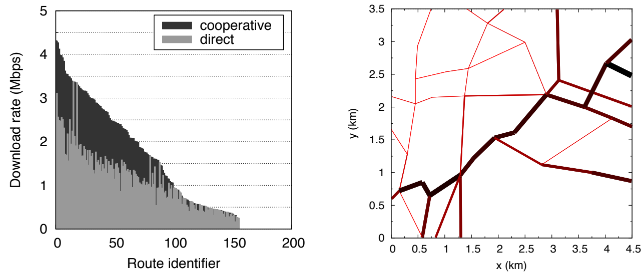
One legitimate question at this point would be if these average figures are representative of the experience of every downloader. By looking at Fig. 21(a), the answer seems to be no. Indeed, the cumulative distribution of the aggregate download rate shows a significative unfairness among downloaders: as an example, in presence of a Density-based AP deployment[8] and 10 concurrent downloaders, the least fortunate 30% of the downloaders gets a goodput of at most 700 Kbps, while the top 30% can retrieve the desired content at a rate that is at least four times higher. To better capture such unfairness, in Fig. 21(b) we portray the Jain's fairness index associated to the distributions of Fig. 21(a).

[8]Similar results were obtained under the Cross volume-based deployment.

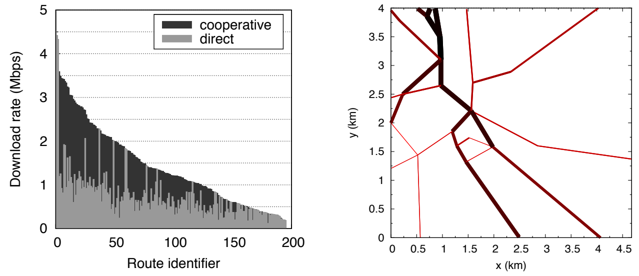


(a) Density (b) Cross volume

Fig. 22. Scatterplot of the downloaded size versus the trip duration. Results refer to the (p,t)-Constrained carriers selection scheme in presence of 10 APs, and are averaged over all road topologies



(a) Density, North Zurich, rates (b) Density, North Zurich, map



(c) Cross, Central Zurich, rates (d) Cross, Central Zurich, map

Fig. 23. Extract of per-route analysis results, referring to the Density-based AP deployment in North Zurich (top) and to the Cross volume-based AP deployment in Central Zurich (bottom): in both cases, we report the average direct and cooperative rates over the possible vehicular trajectories, that are ordered by decreasing aggregate rate (left), and the download rates over the road topology, where darker and thicker lines represent road segments with higher aggregate rates. All results refer to the (p,t)-Constrained carriers selection scheme in presence of 10 APs

We can notice that the index ranges between 0.5 and 0.7, i.e., low values that confirm the scarce equity with which the overall download rate is distributed among downloaders. The number of concurrent downloaders appears instead to have a minor impact on the fairness, as it only induces a slight reduction of the index.

In order to understand the reason of the diverse download experience of the different users, we first observe if there is a correlation between the amount of downloaded data and the duration of the trip of a vehicle, intended as the interval between the instants at which the car enters and exits the road scenario. The relative scatterplots are depicted in Fig. 22, where we also report the line representing the maximum downloadable file size per trip length, computed as the 5 Mbps data rate times the trip duration. Interestingly, we can observe that download sizes close to the maximum are attained by vehicles with short as well as long trips. Moreover, downloaders engaged in medium-to-long trips can have very different download experiences, resulting in both very high and very low download sizes. Our conclusion is that the duration of the trip is not the cause behind the unfairness observed in the download performance of different users.

We then increase the level of detail of our analysis, and consider not just the duration of a trip, but its exact trajectory. More precisely, we record all the possible routes (i.e., ordered sequences of roads) traveled by downloaders, and measure the average download rates experienced by users on each route. Fig. 23(a) and 23(c) show, for two sample scenarios[9], the direct and cooperative rate attained by vehicles traveling along the different routes, that are ordered over the x axis, by decreasing aggregate download rate. It appears now clear that the unfairness among downloaders is the result of an unfairness among trajectories, as some routes allow average rates of 3 Mbps or more, whereas others yield much poorer (e.g., 500 Kbps or less) performance. It is interesting to note that, no matter which AP deployment strategy and road scenario are considered, the routes that guarantee the highest rates are often those where carry&forward transfers are exploited the most. On the other hand, cooperation is completely absent on those trajectories that allow very low throughput. Fig. 23(b) and 23(d) complement the previous results, showing which routes provide higher download rates: the difference between lighter, thinner low-rate trajectories and darker, thicker high-rate ones is even more evident. We can therefore state that there are routes that are more keen to take advantage from a cooperative download framework, and others that are less so. Moreover, we also found a moderate positive correlation between the carry&forward download rate and the vehicular density that characterize a same road segment, as shown in Tab. I: this suggests that it is over routes where the traffic is denser that cooperation among vehicles brings the highest advantage, thanks to a wider range of opportunistic contacts that can be leveraged for data transfers.

[9]Similar results, omitted for the sake of brevity, were obtained for all others combinations of AP deployment strategies and road topologies.

Scenario	Density	Cross volume
West Zurich	0.74	0.75
Central Zurich	0.57	0.64
North Zurich	0.59	0.63
Schlieren	0.42	0.60

TABLE I

CORRELATION BETWEEN THE COOPERATIVE RATE AND VEHICULAR DENSITY CHARACTERIZING A SAME ROAD, IN DIFFERENT SCENARIOS

Our conclusion is that, if use of the carry&forward paradigm is judiciously limited to selected trafficked routes, cooperation among vehicles can provide significative and consistent increments in the download rates experienced by users traveling along such trajectories. The results in Fig. 23 indicate that, in such situations, one could expect typical gains in the order of 50% to 120% with respect to the case where only direct transfers from APs are considered.

VI. CONCLUSIONS AND OPEN PROBLEMS

We presented a complete study of cooperative download in urban vehicular environments. We identified and proposed solutions to the problems of carriers selection and chunk scheduling, and extensively evaluated them. The main contribution of this work lies in the demonstration that vehicular cooperative download in urban environments can bring significant download rate improvements to users traveling on trafficked roads in particular.

Since our study is, to the best of our knowledge, the first of its kind, a number of research directions remain to be explored before cooperative downloading systems can be deployed in urban areas. First, a more thorough exploration of the chunk scheduling problem is needed, so to identify potential optimal working points in the reliability/redundancy tradeoff. Second, an analysis of the management of control messages related to the cooperative download over the access and backbone networks is required, so to unveil the scalability limits of the signalization phase and identify solutions to overcome them. Third, more efficient AP deployment strategies could be designed, leveraging our discovery that routes exhibit different levels of fitness to the cooperative download process.

REFERENCES

[1] F. Aidouni, M. Latapy, C. Magnien, *"Ten weeks in the life of an eDonkey server"*, Hot-P2P'09, Rome, Italy, May 2009.

[2] K. Fall, *"A delay-tolerant network architecture for challenged Internets"*, ACM Sigcomm'03, Karlsruhe, Germany, August 2003.

[3] A. Nandan, S. Das, G. Pau, M. Gerla, M.Y. Sanadidi, *"Co-operative downloading in vehicular ad-hoc wireless networks"*, WONS'05, St.Moritz, Switzerland, January 2005.

[4] M. Sardari, F. Hendessi, F. Fekri, *"Infocast: A New Paradigm for Collaborative Content Distribution from Roadside Units to Vehicular Networks"*, IEEE SECON'09, Rome, Italy, June 2009.

[5] O. Trullols-Cruces, J. Morillo, J. Barcelo-Ordinas, J. Garcia-Vidal, *"A Cooperative Vehicular Network Framework"*, IEEE ICC'09, Dresden, Germany, June 2009.

[6] B.B. Chen, M.C. Chan, *"MobTorrent: A Framework for Mobile Internet Access from Vehicles"*, IEEE INFOCOM'09, Rio de Janeiro, Brasil, April 2009.

[7] J. Zhao, G. Cao, *"VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks"*, IEEE INFOCOM'06, Barcelona, Spain, April 2006.

[8] S. Yoon, H.Q. Ngo, C. Qiao, *"On 'Shooting' a Moving Vehicle with Data Flows"*, IEEE MOVE'07, Alaska, USA, May 2007.

[9] F. Malandrino, C. Casetti, C.-F. Chiasserini, M. Fiore, *"Content Downloading in Vehicular Networks: What Really Matters"*, IEEE INFOCOM'11, Shangai, China, April 2011.

[10] Z. Chen, H. Kung, D. Vlah, *"Ad hoc relay wireless networks over moving vehicles on highways"*, ACM MobiHoc'01, Long Beach, USA, October 2001.

[11] J. Burgess, B. Gallagher, D. Jensen, B. Levine, *"MaxProp: routing for vehicle-based disruption-tolerant networks"*, IEEE INFOCOM'06, Barcelona, Spain, April 2006.

[12] H.-Y. Huang, P.-E. Luo, M. Li, D. Li, X. Li, W. Shu, M.-Y. Wu, *"Performance evaluation of SUVnet with real-time traffic data"*, IEEE Transactions on Vehicular Technology, vol. 56, no. 6, November 2007.

[13] H. Wu, R. Fujimoto, R. Guensler, M. Hunter, *"MDDV: a mobility-centric data dissemination algorithm for vehicular networks"*, ACM VANET'04, Philadelphia, USA, October 2004.

[14] A. Skordylis, N. Trigoni, *"Delay-bounded routing in vehicular ad hoc networks"*, ACM MobiHoc'08, Hong Kong, China, May 2008.

[15] J. Zhang, Q. Zhang, W. Jia, *"A novel MAC protocol for cooperative downloading in vehicular networks"*, IEEE GLOBECOM'07, Washington, USA, December 2007.

[16] S. Ahmed, S.S. Kanhere, *"VANETCODE: network coding to enhance cooperative downloading in vehicular ad hoc networks"*, ACM IWCMC'06, Vancouver, Canada, July 2006.

[17] G. Marfia, G. Pau, E. Giordano, E. De Sena, M. Gerla, *"Evaluating vehicle network strategies for downtown Portland: opportunistic infrastructure and importance of realistic mobility models"*, ACM MoBiOpp'07, San Juan, Puerto Rico, June 2007.

[18] Y. Ding, C. Wang, L. Xiao, *"A Static-Node Assisted Adaptive Routing Protocol in Vehicular Networks"*, ACM VANET'07, Montreal, Canada, September 2007.

[19] C. Lochert, B. Scheuermann, C. Wewetzer, A. Luebke, M. Mauve, *"Data aggregation and roadside unit placement for a vanet traffic information system"*, ACM VANET'08, S.Francisco, USA, September 2008.

[20] Z. Zheng, P. Sinha, S. Kumar, *"Alpha Coverage: Bounding the Interconnection Gap for Vehicular Internet Access"*, IEEE INFOCOM 2009, Rio de Janeiro, Brasil, April 2009.

[21] Z. Zheng, Z. Lu, P. Sinha, S. Kumar, *"Maximizing the Contact Opportunity for Vehicular Internet Access"*, IEEE INFOCOM 2010, San Diego, USA, March 2010.

[22] O. Trullols, M. Fiore, C. Casetti, C.-F. Chiasserini, J.M. Barcelo-Ordinas, *"Planning roadside infrastructure for information dissemination in intelligent transportation systems"*, Computer Communications, vol. 33, January 2010.

[23] A. Fleisher, *"On prediction and urban traffic"*, Papers in Regional Science, vol.7, no. 1, December 1961.

[24] K. Ashok, M.E. Ben-Akiva, *"Estimation and prediction of time-dependent origin-destination flows with a stochastic mapping of path flows and link flows"*, Transportation Science, vol. 36, no. 2, May 2002.

[25] H. Yin, S.C. Wong, J. Xu, C.K. Wong, *"Urban traffic flow prediction using a fuzzy-neural approach"*, Transportation Research C, vol. 10, no. 2, April 2002.

[26] J. Ott, D. Kutscher, *"Drive-thru Internet: IEEE 802.11b for automobile users"*, IEEE Infocom'04, Hong Kong, China, March 2004.

[27] ETHZ traces, *http://www.lst.inf.ethz.ch/research/ad-hoc/car-traces*.

[28] N. Cetin, A. Burri, K. Nagel, *"A large-scale multi-agent traffic microsimulation based on queue model"*, STRC'03, Ascona, Switzerland, March 2003.

[29] J. Bicket, D. Aguayo, S. Biswas, R. Morris, *"Architecture and Evaluation of an Unplanned 802.11b Mesh Network"*, ACM MobiCom'05, Koln, Germany, August 2005.

[30] V. Bychkovsky, B. Hull, A.K. Miu, H. Balakrishnan, S. Madden, *"A Measurement Study of Vehicular Internet Access Using In Situ Wi-Fi Networks"*, ACM MobiCom'06, Los Angeles, USA, September 2006.

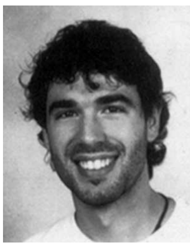
[31] VANET chunk scheduling simulator, *http://personals.ac.upc.edu/trullols*

[32] R. Gass, J. Scott, C. Diot, *"Measurements of in-motion 802.11 networking"*, IEEE WMCSA/HotMobile'06, Washington, USA, April 2006.

[33] F. Hui, P. Mohapatra, *"Experimental Characterization of Multi-hop Communications in Vehicular Ad Hoc Networks"*, ACM VANET'05, Koln, Germany, September 2005.

[34] M. Fiore, J. Härri, *"The networking shape of vehicular mobility"*, ACM MobiHoc'08, Hong Kong, China, May 2008.

Oscar Trullols-Cruces received both B.Sc. and M.Sc degrees in computer engineering from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in July 2008. He joined the Computer Networking Research Group in July 2007, where he is pursuing his PhD. He was visiting Royal Institute of Technology (KTH), Stockholm, Sweden, during spring semester in 2009. His research interests include vehicular networks, delay tolerant networks and mobility modeling. He is a member of the IEEE.



Marco Fiore is an Assistant Professor at INSA Lyon, and an INRIA researcher within the SWING team hosted by the CITI Lab. He received his PhD from Politecnico di Torino, in 2008, and M.Sc degrees from University of Illinois and Politecnico di Torino, in 2003 and 2004, respectively. He has been a visiting researcher at Rice University, in 2006 and 2007, and Universitat Politècnica de Catalunya, in 2008. His research interests are in the field of mobile networking, with focuses on vehicular networks, mobility modeling and analysis, opportunistic communication, and security. He is a member of the IEEE.

Jose M. Barcelo-Ordinas is an Associate Professor at Universitat Politècnica de Catalunya (UPC). He received his PhD and M.Sc at UPC (1998 and 1991). He joined the CompNet research group in 1993. He has participated in several European projects such as EXPLOIT, BAF, EXPERT, NETPERF, MOEBIUS, WIDENS projects, and EuroNGI, EuroNFI and EuroNF (VII FP) Networks of Excellence (NoE). His currently research areas are Delay Tolerant Networks, VANETs and Wireless Sensor Networks. He is a member of the IEEE.