# HAL
## open science

# Offloading Massive Data onto Passenger Vehicles: Topology Simplification and Traffic Assignment

Benjamin Baron, Prométhée Spathis, Hervé Rivano, Marcelo Dias de Amorim

## ▶ To cite this version:

HAL Id: hal-01247713

https://hal.sorbonne-universite.fr/hal-01247713

Submitted on 22 Dec 2015

# Offloading Massive Data onto Passenger Vehicles: Topology Simplification and Traffic Assignment

Benjamin Baron, Prométhée Spathis, Hervé Rivano, and Marcelo Dias de Amorim

*Abstract*—Offloading is a promising technique for alleviating the ever-growing traffic load from infrastructure-based networks such as the Internet. Offloading consists in using alternative methods of transmission as a cost-effective solution for network operators to extend their transport capacity. In this paper, we advocate the use of conventional vehicles equipped with storage devices as data carriers whilst being driven for daily routine journeys. The road network can be turned into a large-capacity transmission system to offload bulk transfers of delay-tolerant data from the Internet. The challenges we address include how to assign data to flows of vehicles and while coping with the complexity of the road network. We propose an embedding algorithm that computes an offloading overlay where each logical link spans over multiple stretches of road from the underlying road infrastructure. We then formulate the data transfer assignment problem as a novel linear programming model we solve to determine the optimal logical paths matching the performance requirements of a data transfer. We evaluate our road traffic allocation scheme using actual road traffic counts in France. The numerical results show that 20% of vehicles in circulation in France equipped with only one Terabyte of storage can offload Petabyte transfers in a week.

*Index Terms*—Offloading, Data Vehicular Networks, Delay-Tolerant Networks, Road Infrastructure.

## I. INTRODUCTION

With the ever-growing amount of data generated on the Internet [1], [2], there is an increasing demand for bandwidth-intensive background services, such as maintenance and pro-visioning activities between remote data centers (e.g., virtual machine migration and offline backups) [3]. In this context, *traffic offloading* is gaining more and more interest, as it represents a cost-effective solution to *extend* network capacity. Offloading involves exploiting alternative transmission media and data delivery models [4]. The strategy we consider in this paper is to turn conventional vehicles equipped with storage devices into data carriers with the aim of converting the road infrastructure into a large capacity transmission system [5].

We propose to exploit the delay-tolerance of background traffic to offload bulk data transfers from an infrastructure-based network such as the Internet over the road network. We target data transfers in the context of applications with a delay tolerance of several days (e.g., distribution of large

Benjamin Baron, Prométhée Spathis, and Marcelo Dias de Amorim are with the LIP6/CNRS Computer Science laboratory, Université Pierre et Marie Curie, Paris, France. Emails: {bbaron,spathis,amorim}@npa.lip6.fr. Hervé Rivano is with Inria, Université de Lyon, INSA-Lyon/CITI, Villeurbanne, France. Email: herve.rivano@inria.fr.
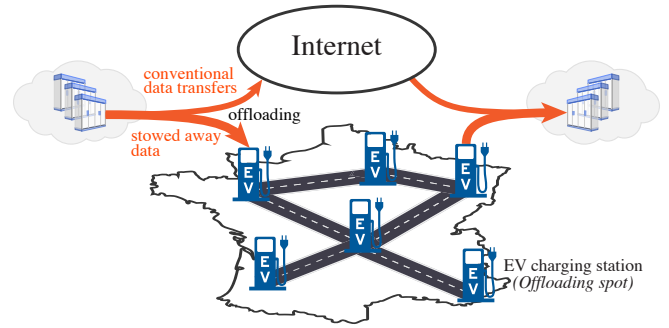
Fig. 1: Conceptual representation of the vehicular traffic offloading. Part of conventional traffic transferred on the Internet is stowed away onto the electric vehicles travelling the roads connecting the source to the final destination. The transfer of offloaded data results from the electric vehicles' journeys interspersed with stops at the charging stations.

scientific datasets or data resulting from maintenance and provisioning activities) [6]. Our offloading service takes op-portunistic advantage of the increasing number of journeys involving vehicles [7], as well as the time vehicles spent parked. According to recent reports, a vehicle spends 80% of its time parked at home, 17% at a destination, and the rest of its time driving [8]. Data is stowed away seamlessly on board of the vehicles as a way to enhance the end-to-end throughput observed by, for instance, a content provider who needs to transfer bulk delay-tolerant data.

We present our offloading service in the context of a network of charging stations for electric vehicles (EVs). Data is transferred from or loaded off the EVs, without the drivers being aware, while charging their batteries as they usually do. This context provides support for the modeling and analysis of our service in practical settings, including the driving range of EVs (e.g., 200 miles for the Tesla Model S) and the charging time of the batteries (e.g., 20 minutes when charging at a Tesla supercharger) [9]. Our service fits with the recent emerging trends in the EV market, as car manufacturers and charging stations operators are seeking additional revenues beyond sales from added services offered while EVs are being charged [10]. The offloading service is depicted in Figure 1.

Related proposals suggest offloading data onto other trans-mission media such as planes [11], trains [12], postal ser-vices [13], or even birds [14]. The underlying objective is to provide cost-effective solutions for transferring data without relying solely on the Internet. Nevertheless, they fail to achieve massive offloading and can be used only in very specific scenarios. As we will show later in this paper, our solution is more ambitious and can transfer large amounts of data
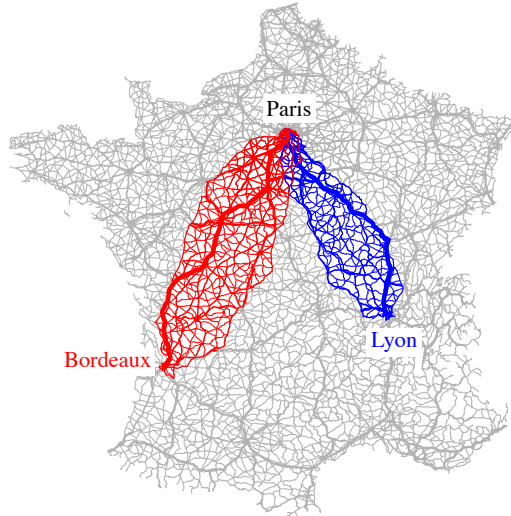
Fig. 2: Main roads connecting Paris to Lyon and Paris to Bordeaux.

reaching a Petabyte per week in the scenario we evaluate. Also, a parcel delivery service (e.g., FedEx trucks transporting Amazon Import/Export Snowball[1]) would be an alternative, or even complementary, solution for the offloading problem we tackle; it requires, though, a dedicated fleet of vehicles traveling pre-planned routes per data transfer. Our service draws on the daily routine journeys of conventional vehicles to opportunistically transfer bulk transfers of delay-tolerant traffic.

The objective of this paper is to assess the concept of vehicular traffic offloading. The main step toward the feasibility of this concept can be stated as follows: *given a request to offload a data transfer from an infrastructure-based network, how to select the road network path and the flows of vehicles matching the data transfer requirements.* Our offloading service receives requests to offload bulk transfers of delay-tolerant data between two remote locations. Data offloads of a transfer are stowed away on board of vehicles routinely traveling the road paths connecting those two locations. The resulting flows of loaded vehicles should match the bandwidth and delay requirements of the data transfer. To realize this service, we need first to cope with the scale of the road infrastructure. The high degree of complexity of the road networks topology and the large number of daily routine journeys [7] makes the vehicle allocation problem computationally intractable. We illustrate the need for a scalable allocation mechanism in Figure 2, where we depict the many possible trips that can be allocated to offload transfers initiated from Paris and destined to two distant cities in France, namely Bordeaux and Lyon. A second challenge is how to efficiently allocate the data transfers to flows of vehicles. We need to design an efficient allocation process that maximizes the cost benefit of offloading traffic on the road network.

In this paper, we address these challenges by formulating and solving the road resource allocation problem as a

mathematical optimization model. We first propose a mapping algorithm that creates an offloading overlay network on top of the road infrastructure. Nodes in the overlay network correspond to charging stations and are connected through logical links, which correspond to road paths, connecting multiple stretches of road in the underlying road infrastructure. Each logical link is characterized by a set of attributes such as delay, capacity, and data loss. The offloading overlay network mitigates the complexity of the substrate network and makes dynamic allocation applicable. Second, we formulate the data transfer assignment problem as a novel linear programming model that determines the optimal logical paths, consisting of a collection of logical nodes connected together by logical links in the offloading overlay. We provide numerical results using actual road traffic counts in France. We devise a concrete deployment plan of charging stations for electric vehicles travelling the French roads. The search space is further reduced by considering the physical paths satisfying the delay requirements of an offloading request.

In summary, our main contributions in this paper are:

- **Space reduction.** We design a mapping algorithm to mitigate the complexity of the road infrastructure. The output of our algorithm is an offloading overlay that gives a comprehensive representation of the underlying resources.
- **Data transfers.** We formulate the data transfer assignment problem as a novel linear programing (LP) model. We solve the LP model to determine the optimal logical path matching the performance requirements of a data transfer.
- **Feasibility study.** We evaluate our road traffic allocation procedure on the French road network using actual road traffic counts. The results show that our service can transfer large amounts of data, with potential transfers in the order of one Petabyte during a one-week time window.

The rest of this paper is organized as follows. In Section II, we present a motivating scenario we will use throughout this paper. In Section III, we introduce the offloading service model and the notations we use in our model. We present the mapping algorithm that creates the offloading overlay network in Section IV and formulate the assignment problem as a linear programing model in Section V. In Section VI, we evaluate the model using actual road traffic counts for the French road infrastructure. Section VII provides a discussion of our results and of open issues. We then review related work in Section VIII, and conclude the paper with a summary and give a outlook of future work in Section IX.

## II. VEHICULAR OFFLOADING SERVICE

In this section, we present the reference scenario we will use throughout this paper. Recall that the objective is to increase the capacity of the Internet when it comes to transferring large amounts of delay-tolerant data, while avoiding costly infrastructure upgrades. We then describe the overall operation to enable efficient transfer of the offloaded traffic over the road infrastructure, as well as the actions taken at the charging stations.

---

[1]https://aws.amazon.com/about-aws/whats-new/2015/10/aws-import-export-snowball/

## A. Reference scenario, assumptions, and roles

The scenario we consider includes electric vehicles (EVs) and their owners, the offloading service provider that can be the EV manufacturers themselves, the charging station operator, and a content provider that operates multiple data centers.

The electric vehicles are equipped with one or more data storage devices, such as magnetic disks or other non-volatile solid-state storage devices. We assume that the content of the storage devices is not accessible by the drivers and the data encrypted when piggybacked onto the vehicle. The term "electric vehicle" refers to both passenger and commercial vehicles propelled by one or more electric motors powered by a rechargeable on-board battery. Commercial vehicles may be part of a fleet owned or leased by a business or a governmental agency. We assume that vehicles also embed one or more communication network interfaces and a positioning system (GPS). The positioning system includes a navigation system that generates routes and guidance between a geographic location and a destination. We refer to the charging stations as *offloading spots* where the data is seamlessly transferred on or unloaded from the on-board memory of the vehicles using short-range radio while they charge their battery.

The offloading service provider, if different from the EV manufacturers, proposes a "get paid to drive" program to the car owners. The service provider installs the memory devices and the car owners receives a monthly fee or a discount on the cost of charging their vehicle in exchange of driving their normal routine. The discount rate is negotiated with the charging station operator (e.g., ChargePoint who operates a world-wide network of EV charging stations offering cloud-network services[2]) and is calculated based on the driving pattern including coverage and mileage. If the EV manufacturers take on the role of service provider, vehicles are equipped as standard with on-board memory storage and service is provided without involving or compensating the vehicles' drivers. The service provider charges the content provider for the amount of data to offload on the road infrastructure and shares the revenues with the charging station operator.

The offloading spots are featured with storage capabilities where data is transloaded from the infrastructure network and warehoused as chunks until transferred on a charging vehicle. The content provider uses a border dray transfer system if the data originates from a distant repository, typically a wide-area data network such as the Internet connecting the content provider to the edge offloading spots. The service provider monitors the status of the offloading spots which include the amount of free memory and the destination of the data chunks waiting to be transshipped. The service provider keeps record of the status of the offloading spots in a specific database. The service provider also queries the vehicles' positioning system to determine their current geographic destination. The historical locations are stored in a geographic location database managed by the service provider to help the offloading spots predict the remaining itinerary of the passing vehicles.
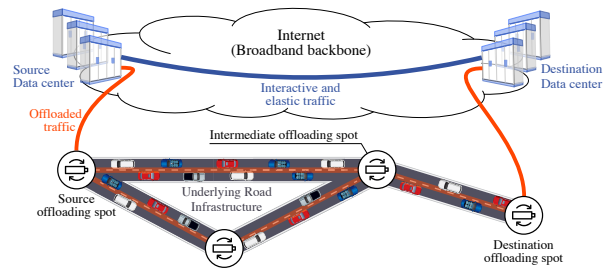
Fig. 3: Architecture related to the reference scenario.

## B. Offloading system operation

Figure 3 gives a schematic overview of the operation to offload a large amount of delay-tolerant background data over the road infrastructure between two remote data centers. We target data transfers lasting several days to a few weeks that result from provisioning or maintenance activities required for virtual machine migrations or offline backup between data centers. The data offloaded from the infrastructure network follows the same path as the flow of vehicles traveling the stretches of road connecting a sequence of offloading spots. Multiple consecutive offloading spots may be involved if the data needs to be shipped across a large body of country before reaching geographically long distant destinations.

To avoid the need of relying on vehicles solely traveling all the way from the source to the destination of the data transfers, offloading spots act as data exchange relay points. Vehicles in contact with an offloading spot unload their cargo if heading to a different direction with the destination of the data transfer. The data is stored until transferred on a subsequent vehicle heading toward the intended destination. The data is therefore "hitchhiked" hop-by-hop through the network of offloading spots before reaching its final destination. Transfers of offloaded data result from the electric vehicles' routine journeys interspersed with stops at the charging stations as they usually do to charge their batteries.

The service provider receives the requests to offload data transfers from the content provider. Each request specifies the delay and bandwidth requirements for the corresponding data transfer. The service provider keeps track of the status of the offloading spots which include statistics about the charging vehicles. The offloading spot operator also reports information about the data chunks locally discharged for later shipment. The service provider, which collects the information gathered from the offloading spots, has an up-to-date view of the offloading infrastructure.

Upon receiving a request to offload a data transfer, the service provider computes the road network path that can accommodate the data transfer requirements and how much data to allocate to each flow of vehicles. The road network path consists of a sequence of offloading spots, each configured with the list of actions to perform on each vehicle traveling in the direction of the next-hop offloading spot. Each action defines the behavior that the offloading spot operator needs to perform with the data belonging to the offloaded transfer. The corresponding data can either be already stored at the offloading spot or carried by the charging vehicle. Common actions include loading data chunks on or off the vehicles

while charging their battery. The service provider defines these actions based on the information the offloading spot operator reports on the flows of vehicles passing through the offloading spots. For each vehicle stopping by an offloading spot, the offloading spot operator matches the destination of the vehicle against the data transfers currently offloaded on the road network and performs the actions as dictated by the service provider.

For each offloading request, the service provider determines the road network path by solving the data transfer allocation problem we formulate as a multi-commodity flow allocation model in Section V. An existing allocation plan may be dynamically modified in case the vehicles change direction unexpectedly or to account for new data transfers as they arrive. The dynamic allocation method we describe is applied to flows of vehicles. Such a flow refers to the group of electric vehicles traveling in the same direction between two adjacent offloading spots. We do not consider the complete journey of each vehicle traveling the underlying road network. The scheduling of the data transfers (i.e., the selection of the data to be piggybacked), and the security and privacy issues are out of scope of this article. In section VII, we will discuss potential future directions for addressing these issues.

## III. OFFLOADING SERVICE MODEL AND NOTATIONS

In this section, we describe the offloading service model presented in the previous section. We first introduce an abstract representation of the road infrastructure we refer to as an offloading overlay and then describe the data transfer assignment problem.

### A. Offloading overlay

**Road network.** We represent the road network by a directed graph $G^R = (N^R, L^R)$, where $N^R$ and $L^R$ denote the set of physical nodes and links, respectively. The set of nodes $N^R = N^J \cup N^S$ consists of two subsets: the set of road junctions ($N^J$) and the set of charging stations ($N^S$). A road junction refers to a location where vehicles can change their direction of travel. A link in the road network corresponds to a stretch of road connecting two adjacent junctions or a junction and a charging station. We consider the stretches of road and the traffic flowing in both directions homogeneous as they share the same profile in terms of capacity and free-flow speed. The properties of the physical links are detailed in Section IV.

**Offloading overlay.** The offloading overlay provides an abstract view of the resources that will be allocated to the offloaded data transfer. More specifically, the overlay mitigates the combinatorial explosion of the number of road paths in the road network. We represent the offloading overlay by a directed graph $G^O = (N^O, L^O)$, where $N^O$ and $L^O$ denotes the set of logical nodes and links, respectively. A logical node of $N^O$ is an *offloading spot*, which corresponds to a location where vehicles can park close and long enough to transfer data. In our scenario that includes electric vehicles, the offloading spots correspond to charging stations. A logical
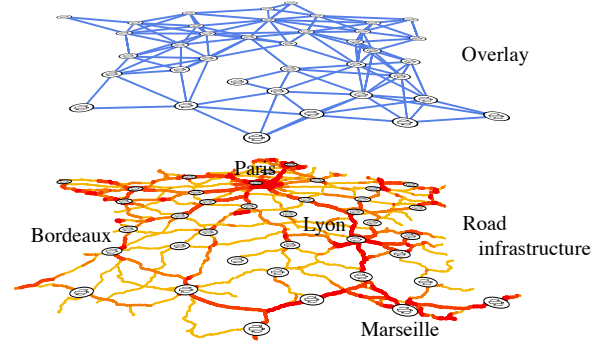


Fig. 4: The offloading overlay resulting from a concrete deployment plan of charging stations we devised for France (the darker and bolder the stretches of road, the higher the vehicle density).

link of $L^O$ represents the road path (i.e., sequence of stretches of road) connecting two adjacent offloading spots in the road network. Note that multiple road paths may connect two adjacent offloading spots [15], [16]. In Figure 4, we show an example of realization of an offloading overlay on top of the French road network. In Section IV, we characterize a logical link $(i, j) \in L^O$ by the weighted travel time $t(i, j)$, aggregated capacity $c(i, j)$ and data leakage $\mathcal{L}(i, j)$. The data leakage refers to the loss rate on logical link $(i, j) \in L^O$ due to vehicles unexpectedly changing direction, vehicles hijacking, accidents, or breaks down. We assume that offloading spots are not constrained by the amount of transfers they can serve and have the adequate storage capacity, so that the overall service is stable. We evaluate the benefits of the offloading overlay over the raw road network in Section VI-C.

### B. Data transfer assignment problem

**Data transfer requests.** We denote by $\mathcal{R}^{st}$ the request to offload a data transfer from a source offloading spot $s \in N^O$ to a target offloading spot $t \in N^O$, within a maximal delay $\mathcal{T}^{st}$ (or delay tolerance), and with a leakage tolerance $\mathcal{L}^{st}$. We denote by $\mathcal{R}$ the matrix of all requests indexed by the pairs of source and destination offloading spots of the corresponding data transfers. The leakage tolerance $0 < \mathcal{L}^{st} \leqslant 1$ refers to the loss ratio tolerated during the transfer, provided that the data is replicated at the source according to redundancy techniques (e.g., Reed-Solomon or RAID) [17], combined to the use of error correcting codes [18]. The leakage tolerance is consistent with the data encoding mechanism. Finally, we denote by $\mathcal{B}^{st}$ the total amount of data (including redundant data) transferred between $s$ and $t$.

**Replication models.** We assume that the sequence and the properties of the links comprising the path followed by a data transfer remain unchanged for the time of the observation. We consider that offloading spots undertake two different behaviors when incoming vehicles unload their cargo data:

- **Local replication (*rep*).** This model assumes that all offloading spots replicate data on multiple outgoing vehicles. For a given request $\mathcal{R}^{st}$ and for each adjacent offloading spot $j$, offloading spot $i$ replicates data $\rho^{st}(i, j)$ times on logical link $(i, j) \in L^O$ (i.e., the same data is

loaded onto $\rho^{st}(i, j)$ vehicles) characterized by leakage $\mathcal{L}(i, j)$ to satisfy the leakage tolerance $\mathcal{L}^{st}$ of the request.

- **Source replication (*nrep*).** The second model assumes that none of the offloading spots replicate data on the outgoing vehicles, except at the source offloading spot. It takes into account the combination of leakages on the logical links of the logical path, denoted by $\mathcal{L}_p(i, j)$ (we assume the source can characterize in advance the entire paths to be allocated using traffic forecasting techniques for instance [15], [16]). For path $p$ and request $\mathcal{R}^{st}$, the source offloading spot $s$ replicates the data $\rho^{st}(p)$ times and satisfies the leakage tolerance $\mathcal{L}^{st}$ of the request.

**Cost model.** We aim at measuring the cost benefit of our offloading service over an infrastructure-based solution when delivering the same amount of data. We consider a data transfer request $\mathcal{R}^{st}$ characterized by the amount of data to transfer $\mathcal{B}^{st}$ and the transfer duration $\mathcal{T}^{st}$ we target. Our model describes the gross margin defined as the difference between the price charged by an Internet Service Provider (ISP) for transferring $\mathcal{B}^{st}$ within $\mathcal{T}^{st}$, and the operational costs induced by achieving the transfer with our offloading service. Below, we define the Internet-based price and the operational costs.

- **Internet-based cost.** $\gamma^{st}$ is the unit cost to send a bit of data from $s$ to $t$ over the legacy Internet (e.g., a leased line between $s$ and $t$). As an example, $\gamma^{st}$ may vary in function of the distance between $s$ and $t$. Without loss of generality, we assume that $\gamma^{st}$ is linear with the volume of data transferred. $\gamma^{st}$ may also be seen as the savings one can achieve by using our offloading service instead of an Internet-based solution.

- **Operational costs.** $\alpha_i$ is the operational cost needed for the maintenance of the storage facility at offloading spot $i$, as well as the financial compensations given to the drivers of the vehicles involved in the offloading service. We assume $\alpha_i$ is also linear with the volume of data exchanged at offloading spot $i$. The value of $\alpha_i$ depends on the replication model we consider. $\alpha_i^{\mathrm{rep}}$ corresponds to the local replication model and $\alpha_i^{\mathrm{nrep}}$ corresponds to the source replication model. The local replication model will require the data to be stored for longer periods of time. Each copy is held until it can be transferred on a vehicle matching the destination of the data. Hence, the need for extra storage facilities will induce higher operational costs for the local replication model $\alpha_i^{\mathrm{rep}}$, compared to the source replication model $\alpha_i^{\mathrm{nrep}}$.

Note that we do not explicitly consider the capital expenditure costs (CAPEX) to deploy the infrastructure required by the offloading service. The CAPEX includes the storage facilities at the offloading spots and the storage devices installed on the vehicles. Instead, we consider the infrastructure already deployed and exploited by the offloading spot operator (e.g., the charging station operator). Hence, the CAPEX costs are amortized and accounted for in the operational costs we consider.

In the following, we state our data delivery problem as a linear optimization problem. We aim at maximizing the overall maximal gross margin that can be achieved for the entire set of

TABLE I: Table of notations for the data transfer assignment problem

| Variable | Meaning |
|---|---|
| $\mathcal{R}^{st}$ | Transfer request between source $s$ and destination $t$ |
| $\mathcal{T}^{st}$ | Delay tolerance of request $\mathcal{R}^{st}$ |
| $\mathcal{L}^{st}$ | Leakage tolerance of request $\mathcal{R}^{st}$ |
| $\mathcal{B}^{st}$ | Amount of data transferred for request $\mathcal{R}^{st}$ |
| $\gamma^{st}$ | Internet-based cost for request $\mathcal{R}^{st}$ |
| $\alpha_i$ | Operational cost at offloading spot $i$ |
| $t_i$ | Waiting time at offloading spot $i$ |
| $f(p)$ | Flow on logical path $p$ |
| $t(p)$ | Travel time of logical path $p$ |
| $\rho^{st}(p)$ | Replication factor for request $\mathcal{R}^{st}$ on path $p$ |
| $t(i, j)$ | Weighted travel time of logical link $(i, j) \in L^O$ |
| $c(i, j)$ | Capacity of logical link $(i, j) \in L^O$ |
| $\mathcal{L}(i, j)$ | Leakage of logical link $(i, j) \in L^O$ |
| $\mathcal{L}_p(i, j)$ | Multiplied leakage of logical path $p$ at link $(i, j) \in L^O$ |
| $\rho^{st}(i, j)$ | Replication factor for request $\mathcal{R}^{st}$ on link $(i, j) \in L^O$ |

data transfer requests $\mathcal{R}^{st}$ between each (source, destination) couple $s, t$. The objective is constrained by the delay tolerance $\mathcal{T}^{st}$ and the capacity $c(i, j)$ of the logical links.

For the sake of readability, we summarize in Table I the variables we use to formalize our problem.

**Summary.** The problem we introduced in the previous section is twofold: (1) to propose a mapping algorithm to create the offloading overlay network on top of the road infrastructure and (2) to model the data transfer assignment problem on the offloading overlay with a linear programming model that aims at maximizing the maximal gross margin of our service. In the following, we consider a snapshot of the system, i.e., a state of a system characterized by given parameters valid for the duration of the data transfers (request demands, road traffic flows, travel times, leakages).

## IV. MAPPING OF THE OFFLOADING OVERLAY

As mentioned in Section III-A, a logical link $(i, j) \in L^O$ is characterized by the weighted travel time $t(i, j)$, aggregated capacity $c(i, j)$, and data leakage $\mathcal{L}(i, j)$. In this section, we detail each attribute of a logical link.

In the following, we consider O-D flows between two stations, i.e., flows that go from station $i \in N^S$ to station $j \in N^S$. Let $r$ be the number of "reasonable" paths between $i$ and $j$, i.e., paths that do not backtrack [19].[3] We note that the value $r$ depends of the route choice and traffic assignment strategies we consider (e.g., All-or-Nothing, Wardrop's equilibrium, or Dial's stochastic assignments) [15], [16].

### A. Travel time of logical links

For stretch of road $(a, b) \in L^R$, let $v_{ab}$ be its nominal volume of vehicles (vehicles per unit of time), $c_{ab}$ be its capacity (vehicles per unit of time), and $\tau_{ab}(0)$ be its corresponding travel time at free-flow speed (i.e., when $v_{ab} = 0$). The travel time of $(a, b)$ is given by the BPR (Bureau of Public Roads) function defined as [20]:

$$\tau_{ab}(v_{ab}) = \tau_{ab}(0)\left[1 + \alpha\left(\frac{v_{ab}}{c_{ab}}\right)^{\beta}\right], \qquad (1)$$

[3]A path that does not backtrack is a path that takes a traveler away from origin and closer to destination [19].

where $\alpha$ and $\beta$ are BPR parameters that depend on the road profile ($\alpha = 0.15$ minutes and $\beta = 4.0$ are typically used) [21].

We can deduce from Eq. (1) the travel time of physical path $p$, denoted $\tau_p$, which is the sum of all travel times of the stretches of road the path consists of (we do not consider any turning delays at junctions):

$$\tau_p = \sum_{(a,\,b)\in p} \tau_{ab}(v_{ab}). \tag{2}$$

From Equation (2), we deduce an expression of the average travel time $t(i,\,j)$ experienced on the $r$ physical paths between nodes $i$ and $j$, weighted by the road traffic flow $v_p$ on each path $p$:

$$t(i,\,j) = \frac{\sum_{p\in\mathcal{P}^{ij}} \tau_p\, v_p}{r \sum_{p\in\mathcal{P}^{ij}} v_p}, \tag{3}$$

where $\mathcal{P}^{ij}$ is the set of all simple physical paths between $i$ and $j$ (i.e., with no cycles in the path).

### B. Capacity of logical links

The capacity $c(i,\,j)$ of the overlay link $(i,\,j) \in L^O$ depends on the sum of the traffic flows $v_p$ of the simple paths between offloading spots $i$ and $j$ (i.e., the number of vehicles per unit of time going from $i$ to $j$ on path $p$). The capacity $c(i,\,j)$ of the overlay link also depends on the market penetration ratio $\mathcal{M}$ of the vehicles participating in the offloading service and the storage size $\mathcal{S}$ on each vehicle. We assume that all vehicles are equipped with a storage device of the same size $\mathcal{S}$.

$$c(i,\,j) = \mathcal{M} \times \mathcal{S} \sum_{p\in\mathcal{P}^{ij}} v_p. \tag{4}$$

### C. Leakage of logical links

The leakage $\mathcal{L}(i,\,j)$ (comprised between 0 and 1) of logical link $(i,\,j) \in L^O$ represents the proportion of data that is lost between offloading spots $i$ and $j$. The leakage increases as more vehicles carrying data prematurely exit the road (e.g., the vehicles may exit the highway before reaching the offloading spot or an accident may have occurred). The leakage depends on characteristics that are inherent to the physical paths mapped in the offloading overlay. Additionally, the leakage accounts for the errors inherent to the scheduling and forwarding at the offloading spots. In our evaluations in Section VI, we evaluate the impact of the leakage on the performance of the offloading service.

## V. OFFLOADING SERVICE REVENUE MAXIMIZATION

In Section III, we described the offloading service model. Here, we present the two revenue maximization models which differ depending on the data replication models used for the data transfers. In the following, $p$ denotes a logical simple path (i.e., without any cycles) defined as a set of offloading spots connected by logical links. $\mathcal{P}^{st}$ denotes the set of all logical simple paths from $s$ to $t$.

We define the travel time $t(p)$ experienced on logical path $p \in \mathcal{P}$ as the sum of the travel times $t(i,\,j)$ on the logical links of $p$ and the waiting time $t_i$ at each intermediate offloading spot on the path:

$$t(p) = \sum_{(i,\,j)\in p} \big(t(i,\,j) + t_i\big). \tag{5}$$

We denote by $f(p)$ the resulting flow (throughput) on logical path $p \in \mathcal{P}^{st}$ for a given request $\mathcal{R}^{st}$. For the same request, the transfer of an amount of data $\mathcal{B}^{st}$ is constrained by the delay tolerance $\mathcal{T}^{st}$ and does not depend on the data replication model. Hence, the delay constraint is the same for both models. The delay to transfer $\mathcal{B}^{st}$ between $s$ and $t$ is the sum of the duration to transfer this quantity over all logical paths between $s$ and $t$ and the average travel time experienced on these paths, weighted by the flow on each path. The delay constraint is expressed as:

$$\frac{\sum_{p\in\mathcal{P}^{st}} f(p)t(p)}{\sum_{p\in\mathcal{P}^{st}} f(p)} + \frac{\mathcal{B}^{st}}{\sum_{p\in\mathcal{P}^{st}} f(p)} \leqslant \mathcal{T}^{st}. \tag{6}$$

Eq. (6) is not a linear expression, which can lead to some difficulties to solve the linear optimization problem. But it can be rewritten as the following linear expression:

$$\sum_{p\in\mathcal{P}^{st}} f(p)\big(\mathcal{T}^{st} - t(p)\big) \geqslant \mathcal{B}^{st}. \tag{7}$$

### A. The "local replication" model

Recall that, in this model, every offloading spots store and replicate data. Consider a given request $\mathcal{R}^{st}$ with leakage tolerance $\mathcal{L}^{st}$ and $p \in \mathcal{P}^{st}$ a path between $s$ and $t$. If $\rho^{st}(i,\,j)f(p)$ data is transmitted on overlay link $(i,\,j) \in p$, $f(p)$ data is received at destination offloading spot $j$. Here, the replication factor $\rho^{st}(i,\,j)$ is calculated by each offloading spot $i$ that sends data to remote offloading spot $j$ on logical link $(i,\,j) \in L^O$ as a function of $\mathcal{L}^{st}$ and $\mathcal{L}(i,\,j)$ as follows:

$$\mathcal{L}(i,\,j)^{\rho^{st}(i,\,j)} \leqslant \mathcal{L}^{st} \implies \rho^{st}(i,\,j) \geqslant \frac{\log(\mathcal{L}^{st})}{\log(\mathcal{L}(i,\,j))}. \tag{8}$$

The flow on logical link $(i,\,j) \in L^O$ is expressed as the sum of all flows sent by offloading spot $i$ to offloading spot $j$ on logical link $(i,\,j)$ for data transfer request $\mathcal{R}^{st}$. The flow is constrained by the capacity of the logical link $c(i,\,j)$:

$$\sum_{s,\,t\in\mathcal{R}} \sum_{\substack{p\in\mathcal{P}^{st}\\ p\ni(i,\,j)}} \rho^{st}(i,\,j)\, f(p) \leqslant c(i,\,j). \tag{9}$$

Recall that, the maximal gross margin is expressed as the difference between the cost of an Internet-based service (using the cost factor $\gamma^{st}$) and the operational costs of our offloading service (costs $\alpha_i^{\text{rep}}$ at each offloading spot $i$). Our objective is to maximize the maximal gross margin, expressed as:

$$\sum_{s,\,t\in\mathcal{R}} \mathcal{B}^{st}\gamma^{st} - \sum_{i\in N^O} \alpha_i^{\text{rep}} \sum_{p\ni i} f(p). \tag{10}$$

One can note that the contribution to the value of Eq. (10) of each $f(p)$ is negative. Besides, the only lower bonding

constraint on $f(p)$ is Eq. (7), which can be interpreted as a requirement on the total flow between each pair $(s, t)$. Therefore, maximizing the maximal gross margin tends to minimize $\sum f(p)$, and Eq. (7) is tight at optimality. It is thus possible to replace $\mathcal{B}^{st}$ in Eq. (10) as follows.

$$\sum_{s,\, t \in \mathcal{R}} \sum_{p \in \mathcal{P}^{st}} f(p) \big[ \gamma^{st} \big( \mathcal{T}^{st} - t(p) \big) - \alpha^{\text{rep}}(p) \big], \qquad (11)$$

where $\alpha^{\text{rep}}(p) = \sum_{i \in p} \alpha_i^{\text{rep}}$.

In Eq. (11), if $\gamma^{st} \big( \mathcal{T}^{st} - t(p) \big) - \alpha^{\text{rep}}(p)$ is negative, then $f(p)$ is null (objective function decreases otherwise). Hence, from Eq. (11), we have:

$$t(p) + \frac{\alpha^{\text{rep}}(p)}{\gamma^{st}} \geqslant \mathcal{T}^{st} \implies f(p) = 0. \qquad (12)$$

Since Eq. (12) is a weight on the logical paths (resulting from the weights of the logical links) that depends only on the offloading overlay, we can narrow the search space of the paths that do not satisfy Eq. (12). Finally, the formulation can be reduced to the maximal gross margin maximization objective subject to the capacity constraint on each logical link $(i, j) \in L^O$:

Maximize $\displaystyle \sum_{s,\, t \in \mathcal{R}} \sum_{p \in \mathcal{P}^{st}} f(p) \psi_{st}^{\text{rep}}(p)$

Subject to

$$\sum_{s,\, t \in \mathcal{R}} \sum_{\substack{p \in \mathcal{P}^{st} \\ p \ni (i,j)}} \rho^{st}(i,j) f(p) \leqslant c(i,j) \qquad \forall (i,j) \in L^O$$

where $\psi_{st}^{\text{rep}}(p) = \gamma^{st} \big( \mathcal{T}^{st} - t(p) \big) - \alpha^{\text{rep}}(p)$ can be considered as a weight on logical path $p \in \mathcal{P}^{st}$.

*B. The "source replication" model*

In this model, data is replicated at the source offloading spot to satisfy the leakage tolerance $\mathcal{L}^{st}$ of the request $\mathcal{R}^{st}$. We assume again that the source can characterize all logical simple paths between the source $s$ and the destination $t$ so it can compute the resulting path leakage $\mathcal{L}_p(i, j)$ for every logical link $(i, j)$ in logical path $p \in \mathcal{P}^{st}$.

We denote by $\mathcal{L}_p(i, j)$ the multiplied leakage experienced at logical link $(i, j) \in L^O$ on logical path $p$. If $p = (i_0, \ldots, i_n), i_k \in N^O$ is a logical simple path between $i_0$ and $i_n$, then for all $0 \leqslant k < n$, the leakage $\mathcal{L}_p(i_k, i_{k+1})$ at logical link $(i_k, i_{k+1}) \in L^O$ is expressed as:

$$\mathcal{L}_p(i_k, i_{k+1}) = 1 - \prod_{j=1}^{k} \big( 1 - \mathcal{L}(i_{j-1}, i_j) \big), \qquad (13)$$

where $0 \leqslant k < n$. Since the source $s$ knows the attributes of the logical path $p$, it may deduce the replication factor $\rho^{st}(p)$ introduced in Section III-B. For request $\mathcal{R}^{st}$ transmitted on logical path $p = (s, i_1, \ldots, i_{n-1}, t) \in \mathcal{P}^{st}$, the replication factor $\rho^{st}(p)$ is calculated by the source $s$ as follows:

$$\mathcal{L}_p(i_{n-1}, t)^{\rho^{st}(p)} \leqslant \mathcal{L}^{st} \implies \rho^{st}(p) \geqslant \frac{\log(\mathcal{L}^{st})}{\log(\mathcal{L}_p(i_{n-1}, t))}. \qquad (14)$$

For each logical link $(i, j) \in L^O$, the capacity constraint limits to the logical link capacity $c(i, j)$ the amount of assigned flows $f(p)$ for each path $p$ that go through the logical link $(i, j)$ with the proportion of leftover flow at the logical link $1 - \mathcal{L}_p(i, j)$ and replication factor $\rho^{st}(p)$ for request $\mathcal{R}^{st}$:

$$\sum_{s,\, t \in \mathcal{R}} \sum_{\substack{p \in \mathcal{P}^{st} \\ p \ni (i,j)}} \rho^{st}(p) \big( 1 - \mathcal{L}_p(i,j) \big) f(p) \leqslant c(i,j). \qquad (15)$$

As in the local replication model, the maximal gross margin is expressed as the difference of the cost of an Internet-based service (with the cost factor $\gamma^{st}$) and the operational cost of the offloading service we propose $\alpha^{\text{nrep}} \ll \alpha^{\text{rep}}$. In this model, we also aim at maximizing the maximal gross margin:

$$\sum_{s,\, t \in \mathcal{R}} \left[ \mathcal{B}^{st} \gamma^{st} - \sum_{i \in N^O} \alpha_i^{\text{nrep}} \sum_{j \in N(i)} \sum_{\substack{p \in \mathcal{P}^{st} \\ p \ni (i,j)}} \rho^{st}(p) \big( 1 - \mathcal{L}_p(i,j) \big) f(p) \right]. \qquad (16)$$

If Eq. (7) is optimal, then it turns into an equality since it is the only equation that constrains $\mathcal{B}^{st}$, given that $\gamma^{st} > 0$. Therefore, we can rewrite Eq. (16) as follows:

$$\sum_{s,\, t \in \mathcal{R}} \sum_{p \in \mathcal{P}^{st}} f(p) \left[ \gamma^{st} \big( \mathcal{T}^{st} - t(p) \big) - \sum_{\substack{p \in \mathcal{P}^{st} \\ p \ni (i,j)}} \alpha_i^{\text{nrep}} \rho^{st}(p) \big( 1 - \mathcal{L}_p(i,j) \big) \right]. \qquad (17)$$

In Eq. (17), we note that if the factor of $f(p)$ is negative, then, since $f(p) \geqslant 0$, the objective function decreases, making it non-optimal. Thus, if the factor of $f(p)$ is negative, the flows $f(p)$ must be null. From the former Eq. (17), we have:

$$t(p) + \frac{\rho^{st}(p)}{\gamma^{st}} \sum_{\substack{p \in \mathcal{P}^{st} \\ p \ni (i,j)}} \alpha_i^{\text{nrep}} \big( 1 - \mathcal{L}_p(i,j) \big) \geqslant \mathcal{T}^{st} \implies f(p) = 0. \qquad (18)$$

Similar to Eq. (12), Eq. (18) is a weight on the logical paths (although, not resulting from the weights of the logical links) that only depend on the offloading overlay; we can narrow the search space of the paths that do not satisfy Eq. (18).

Finally, the formulation can be reduced to the total revenue maximization objective subject to the capacity constraint on each logical link $(i, j) \in L^O$:

Maximize $\displaystyle \sum_{s,\, t \in \mathcal{R}} \sum_{p \in \mathcal{P}^{st}} f(p) \psi_{st}^{\text{nrep}}(p)$

Subject to

$$\sum_{s,\, t \in \mathcal{R}} \sum_{\substack{p \in \mathcal{P}^{st} \\ p \ni (i,j)}} \rho^{st}(p) \big( 1 - \mathcal{L}_p(i,j) \big) f(p) \leqslant c(i,j) \quad \forall (i,j) \in L^O,$$

where

$$\psi_{st}^{\text{nrep}}(p) = \gamma^{st} \big( \mathcal{T}^{st} - t(p) \big) - \sum_{\substack{p \in \mathcal{P}^{st} \\ p \ni (i,j)}} \alpha^{\text{rep}}(p) \rho^{st}(p) \big( 1 - \mathcal{L}_p(i,j) \big).$$

The amount of data transferred $\mathcal{B}^{st}$ within the delay tolerance $\mathcal{T}^{st}$ imposed by request $\mathcal{R}^{st}$ is deduced from Eq. (7):

$$\mathcal{B}^{st} = \sum_{p \in \mathcal{P}^{st}} f(p)\big(\mathcal{T}^{st} - t(p)\big). \qquad (19)$$

We can finally obtain from Eq. (19) the average throughput of request $\mathcal{R}^{st}$ by dividing the amount of data $\mathcal{B}^{st}$ by the duration of the transfer $\mathcal{T}^{st}$.

# VI. EVALUATION

We use real traffic counts measured on France's road network to evaluate the capacity enhancement expected from vehicular offloading. To be consistent with our motivating scenario, and without loss of generality, we assume that electric vehicles have an autonomy of 300 km. The on-board data storage units have a capacity $\mathcal{S}$. We note $\mathcal{M}$ the proportion of vehicles that participate in the offloading process.

## A. Dataset

In our evaluation, we use a dataset collected in 2011 featuring the AADT (Annual Average Daily Traffic) of the major roads in France covering a combined distance of 20,000 km [22]. The AADT is the total volume of traffic passing a stretch of road in both direction for one year, divided by the number of days in the year. The traffic volumes are collected using strategically located automatic traffic recorders (ATR). The thickness of the road segments depicted earlier in Fig. 4 reflect the traffic counts given by the AADTs. The graph consists of 3,310 edges covering over 20,000 km of roads.

It is important to underline that the AADT is a fundamental statistic used in traffic engineering and transportation planning. The use of the AADT helps reduce the effects of seasonal bias and missing data mainly due to equipment failure, construction schedules, and installation dates that plague continuous traffic monitoring [23]. Traffic volumes for each road segment can be directly obtained by multiplying AADT by the duration in days of a transfer. As stated previously, we consider delay tolerance levels of several days to a few weeks, which avoids the influence of daily variations of the road traffic on our evaluations.

## B. Planning of the network of charging stations

We consider a network of stations equivalent to the one Tesla is currently rolling out in North America. This network of stations helps electric vehicles face the problem of limited autonomy and achieve long distance travels [9]. Recall that the offloading process will take place in these stations in a transparent manner to the driver. Since there is no such a network in France for the time being, we plan a simple yet realistic network of stations over the road infrastructure of France. To plan such a network of stations, we consider a facility-allocation problem that minimizes the number of facilities to allocate, a problem we adapted from the maximal covering location problem [24]. The problem takes demand points and candidate locations as inputs:



(a) Allocation of stations over the French road infrastructure.

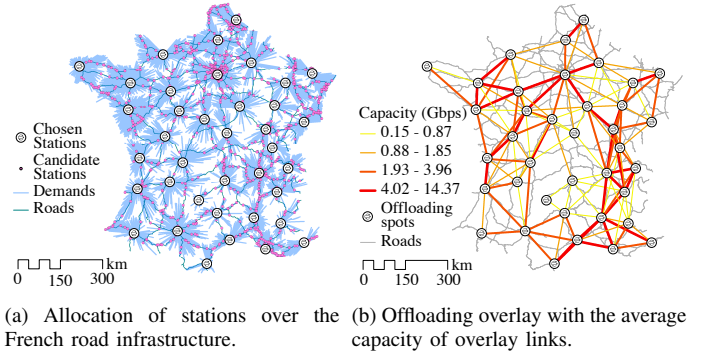(b) Offloading overlay with the average capacity of overlay links.

Fig. 5: Facility-allocation result and offloading overlay. The big dots are the chosen stations.

- The demand points are the 9,555 cities of France with a population greater than 1,000.
- The candidate locations are the 1,024 gas stations of Total, a major oil company in France.

The facility-allocation algorithm selects the stations such that maximum demand points are allocated to the stations within a range of 150 km, while minimizing the number of chosen stations. We assume that car ownership is uniform throughout the territory — we can then weight the cities by their population. The chosen stations are allocated at most 150 km away from each other. This is enough for a vehicle with an autonomy of 300 km to reach the next closest station or return to the same station without depleting the battery. Finally, we assume that the stations have a capacity that suits the demand such that the waiting time is uniform across all stations and avoid any waiting time. The waiting time is then restricted to the service time, which corresponds to the duration of the battery charge.

The resulting allocation outputs 38 stations scattered on the French road infrastructure, as shown in Fig. 5a. We note that stations are mainly allocated near major cities, as the demand from urban areas is higher than the demand from rural areas.

## C. Mapping of the offloading overlay

With the network of stations planned, we can now create the offloading overlay as defined in Section III-A. To compute the resulting overlay depicted in Fig. 5b, we consider the All-or-Nothing traffic assignment strategy that assigns all the traffic between a source and a destination to the shortest path, here defined as the path with the lowest travel time [15], [16]. The traffic flow on this path is set to the minimum AADT of the stretches of road composing it. Then, we use the results of the most recent French travel household survey made in 2008 (ENTD), which distinguishes two kinds of travels [25]: local travels are shorter than 80 km from home, while long-distance travels are greater than 80 km from home.

According to the same survey, long-distance travels account for about, in average, 2.8% of the total amount of travels recorded. This kind of travel is of interest for our offloading service since it corresponds to the case where vehicles are likely to stop at more than two offloading spots to charge their depleted batteries while on a trip. We assume the proportion of long-distance travels is uniform on all the stretches of road. We
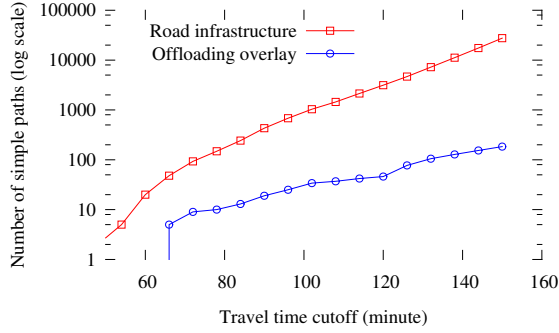
Fig. 6: Total number of simple paths in the road infrastructure and logical simple paths in the offloading overlay as a function of the travel time cutoff ($y$-axis is in a logarithmic scale).

consider a market penetration ratio $\mathcal{M}$ of 20% (which is the case for Renault in France [26]) and a per-node storage unit $\mathcal{S}$ of one TB (assumed to be the same for each electric vehicle). The market penetration ratio corresponds to the proportion of vehicles equipped with storage capabilities and able to participate in our offloading service.

Since the AADT is measured on both directions of a stretch of road, we divide the measured values by two. To compute the throughput of the logical link, we multiply the resulting flow by $\mathcal{S}$. We use the BPR function (Eq. (1)) to calculate the travel time on each stretch of road with: $\alpha = 0.15$ minutes, $\beta = 4.0$, the practical capacity of each stretch of road given by the dataset, and the AADT of the stretch of road for the assigned volume. We use then Eq. (3) to deduce the travel time of the logical links using the shortest physical road paths.

Fig. 6 shows the benefits of using the offloading overlay compared to the road infrastructure. We note that the number of simple paths grows exponentially with the travel time cutoff and is much larger on the road infrastructure. Also, the difference in the number of paths grows exponentially. This exponential growth is the main complexity factor to consider when solving the linear programming models we introduced in Section V.

### D. Data transfer assignment problem

We interface our offloading overlay with CPLEX using the linear optimization models we presented in Section V. We consider a scenario with the following three different requests (distances are Euclidean):

- Request R1: from Paris to Lyon (384 km).
- Request R2: from Paris to Bordeaux (492 km).
- Request R3: from Paris to Marseille (646 km).

It is important to note that requests R1 and R3 will compete for the flows since they share some common subpaths, as we can see in Fig. 4. This leads to fairness issues, as competing requests will be favored over other requests with the gross-margin maximization objective, depending on how well they perform. In the following, we discuss how to ensure fairness among the competing flows.

We use a breadth-first search algorithm to generate all the simple paths for each request. The cutoffs of the simple paths are given by Eqs. (12) and (18) for the local replication *rep* and source replication *nrep* models, respectively. However, the

generation of the simple overlay paths between a source and a destination is exponential, as seen in Fig. 6. To solve this issue, we reduce our search space by applying a default cutoff of 12 hours on the travel time of the simple logical paths we generate for our experiments. A 12-hour cutoff is sufficient for each generated path to cover all end-to-end trips within France, and small enough to avoid unnecessary path computation.

We express the profit margin $\gamma^{st}$ as an exponential function of the distance (in km) between $s$ and $t$:

$$\gamma^{st} = \left[\mathrm{dist}(s,\,t)\right]^{\beta}, \text{ where } \beta \in \mathbb{R}. \qquad (20)$$

In our analyses, we investigate the impact of the following parameters: the profit margin exponent $\beta$, the delay tolerance $\mathcal{T}^{st}$, the leakage tolerance $\mathcal{L}^{st}$, and the link leakage $\mathcal{L}(i,\,j)$ for logical link $(i,\,j) \in L^{O}$. By default, for all requests, we set leakage tolerance to $10^{-2}$, link leakage to 0.05, and delay tolerance to 96 hours (4 days), such that there is enough time for the allocated paths and flows to stabilize (i.e., in steady state). The operational costs $\alpha_i^{\mathrm{rep}}$ and $\alpha_i^{\mathrm{nrep}}$ are weighted by the demands allocated to offloading spot $i$, resulting from the facility-allocation problem. For offloading spot $i$, we set $\alpha_i^{\mathrm{nrep}}$ as the normalized weight of the demands allocated to the offloading spot and $\alpha_i^{\mathrm{rep}} = 1,000 \times \alpha_i^{\mathrm{nrep}}$. Finally, for each offloading spot $i$, we consider a waiting time $t_i = 20$ minutes, which corresponds to the duration of a charge that provides a 300-km range to the vehicles (without any queuing time) [9].

For the sake of clarity, we chose not to show any results on the maximal gross margin. As our objective is to obtain fair flow allocation, we tune the profit margin exponent $\beta$ accordingly. The arbitrary choice of $\beta$ clearly impacts the resulting maximal gross margin that can be achieved. That said, different $\beta$ values lead to different settings for the parameters we consider (delay tolerance, leakage tolerance, and link leakage).

We tackle the transfer assignment problem from a macroscopic point of view. In this way, we do not have to consider the scheduling problem and the forwarding decision, which both require an analysis on a per-vehicle basis. The results presented in this section are thus achieved under ideal condition and show an upper-bound of the performance of the service we propose. However, to provide more realistic results, we take account of the scheduling and forwarding errors by means of the logical link leakage.

In Fig. 7, we first note that the *rep* model (with solid lines) always achieves better throughputs than the *nrep* model (with dashed lines). Indeed, the *nrep* model takes into account all link leakages in the paths it considers and replicates the data accordingly, sending more data than the *rep* model. We also note that our offloading service can achieve transfers with a throughput above 10 Gbps and an aggregated data transfer in the petabyte range with the *rep* model.

Fig. 7a represents the evolution of the throughput of the three requests as a function of $\beta$, given the delay tolerance, the leakage tolerance, and the link leakage. We note that the value of $\beta$ is decisive concerning the allocation of the flow to favor: either short distance (R1) or long distance (R2 and R3). This behavior happens in both models, although the breaking points (represented by a circle in the graph) are not the same for the

(a) Throughput as a function of profit margin exponent $\beta$

(b) Throughput as a function of delay tolerance $\mathcal{T}^{st}$

(c) Throughput as a function of leakage tolerance $\mathcal{L}^{st}$.

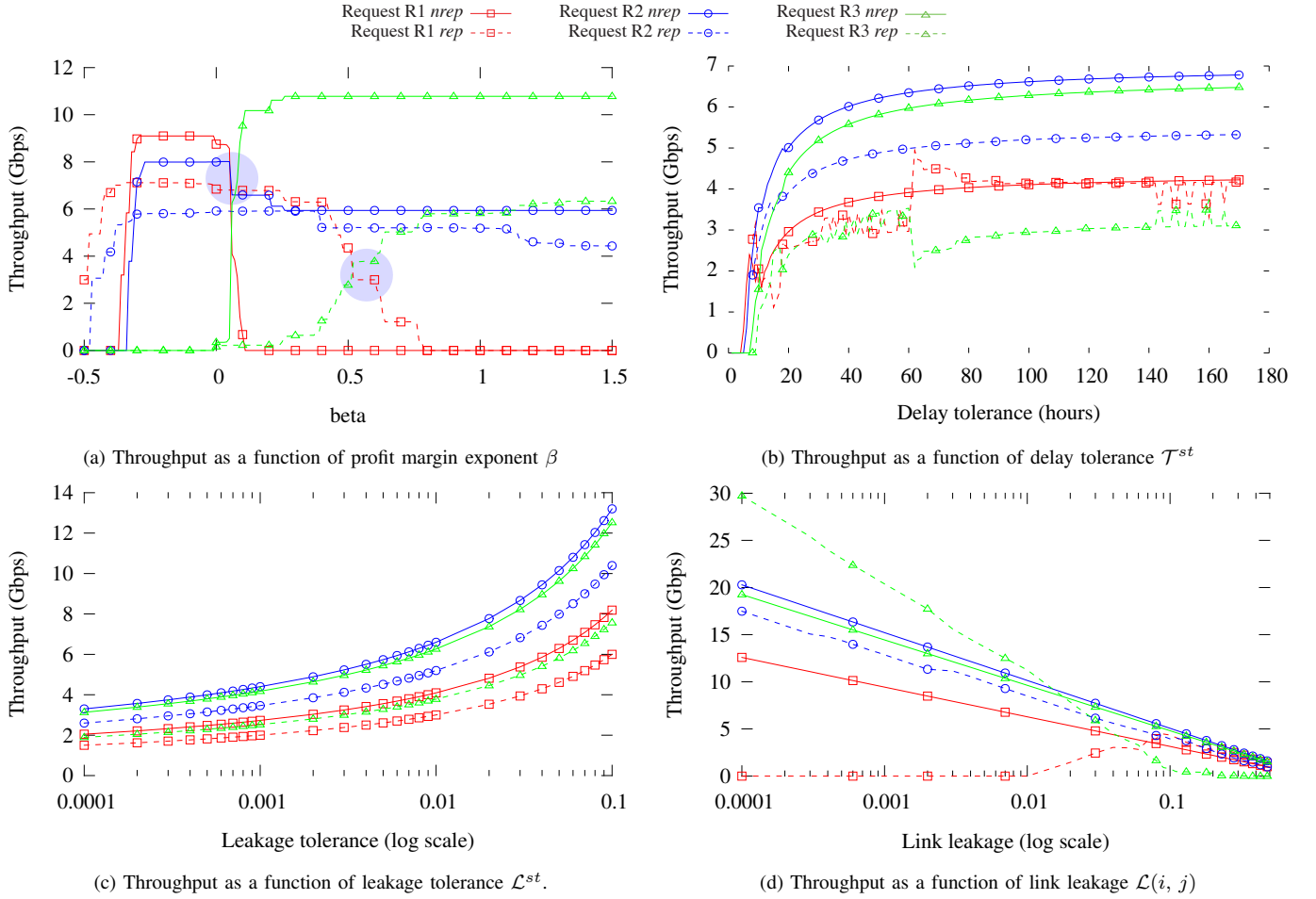(d) Throughput as a function of link leakage $\mathcal{L}(i, j)$

Fig. 7: Evaluation results with, by default, delay tolerance $\mathcal{T}^{st} = 96$ hours, leakage tolerance $\mathcal{L}^{st} = 0.01$, link leakage $\mathcal{L}(i, j) = 0.05$ for all requests $\mathcal{R}^{st}$. The values of the profit margins $\beta^{\text{rep}}$ and $\beta^{\text{nrep}}$ are chosen such that the standard deviation of the throughput of all flows is minimized, to ensure fairness among competing data transfer requests.

two models: the break happens at $\beta = 0.06$ for the *rep* model, and $\beta = 0.6$ for the *nrep* model. Beyond these points, long-distance requests are favored compared with short-distance ones. Indeed, $\gamma^{st}$ is a factor in the maximization objective (defined in Section V) and the higher the $\gamma^{st}$, the bigger the total revenue we aim to maximize. Since $\beta$ is the exponent of the Euclidean distance between $s$ and $t$, we have:

- If $\beta < 0$, $\gamma^{st}$ decreases when the distance increases, favoring short-distance requests.
- If $\beta = 0$, $\gamma^{st} = 1$, favoring low travel time requests.
- If $\beta > 0$, $\gamma^{st}$ increases with the distance, favoring long-distance requests.

Although the impact of $\beta$ on the allocation of the flows between requests R1 and R3 is strong, the impact is weaker with request R2, as this latter does not share many subpaths with R1 and R3. Since we want a fair flow allocation in our offloading service, we choose $\beta^{\text{rep}}$ and $\beta^{\text{nrep}}$ such that the standard deviation of the throughput of all flows is minimized.

It is important to keep in mind, however, that this allocation has an impact on the total revenue because $\gamma^{st}$ is the factor that generates the revenue. Since $\beta^{\text{rep}} < \beta^{\text{nrep}}$, the *nrep* model has a revenue that is much larger than the *rep* model.

Fig. 7b shows the throughput as a function of the delay tolerance $\mathcal{T}^{st}$ (i.e., the duration of the transfer). We notice

that the throughput stabilizes as the duration of the transfer increases. For the *rep* model (solid lines), requests with longer distances are favored (request R2 and R3) over requests with shorter distances (request R1) since the total revenue is increased when favoring requests with longer distances, as seen on Fig. 7a. For the *nrep* model (dashed lines), the allocation of requests R1 and R3 oscillates, favoring one or the other. Indeed, both requests have flows allocated on shared subpaths, depending on the maximization of the total revenue. The oscillation results from the variation of the value of $\beta^{\text{nrep}}$, which depends on the minimization of the standard deviation of the throughput of the allocated flows.

Finally, Figs. 7c and 7d show the throughput as a function of, respectively, the leakage tolerance $\mathcal{L}^{st}$ and link leakage $\mathcal{L}(i, j)$. The throughput increases when the leakage tolerance increases or the link leakage decreases. The *nrep* model almost achieves the same performance, if not better, than the *rep* model when there is a low link leakage ($\mathcal{L}(i, j) < 0.05$). The link leakage also has an effect on the allocation of the flows for the *nrep* model when the flows share subpaths, contrary to the *rep* model: shorter distances (requests R1) are favored with a high link leakage ($\mathcal{L}(i, j) > 0.05$) and longer distances (request R2 and R3) are favored with a low link leakage ($\mathcal{L}(i, j) < 0.05$). Indeed, the objective function of the *nrep*

model depends on the multiplied link leakage $\mathcal{L}_p(i, j)$.

## VII. DISCUSSION ON ASSUMPTIONS AND SPECIFIC DESIGN CHOICES

In this section, we discuss in more details some of our modeling assumptions: estimation of the traffic flow, scheduling at the offloading spots, security of our offloading service, and data exchange at the offloading spots (from data centers to offloading spots and from offloading spots to vehicles).

### A. On the choice of traffic counts

The purpose of our performance evaluation is to assess the capacity enhancement brought to the Internet by the road network. Our evaluation aims at determining the amount of data that can be transferred on the road within a given time period. A typical transfer lasts from a few days to a week, as we can offload up to one Petabyte. The traffic counts we use are provided by the AADT (Annual Average Daily Traffic), which are yearly averages. As so, AADTs average out the effects of seasonal and diurnal variations or missing data due to flawed monitoring. To determine the traffic volume of a road segment over the duration of a transfer, we multiply the corresponding AADT by the transfer duration measured in days. The use of the AADTs prevents our evaluation results from being affected by the diurnal, seasonal, or flawed bias.

To transpose our work in a practical setting or commercial use, thinner-grained traffic count averages should be used to account for transfers lasting few hours or several weeks. An established practice for inferring traffic count averages for different time periods consists in using temporal allocation factors applied to the annual AADTs. An example of temporal allocation factors can be found in "The Traffic Monitoring Guide" where they are referred to as *group factors*. These factors are provided by the US Federal Highway Administration (FHWA) to help State Department of Transportation plan their local Highway Performance Monitoring Systems (HPMS) [23], [27]. The guide describes how to calculate the factor groups (i.e., temporal allocation factors) from the traffic data collected. The procedure depends on the location and the characteristics of the road segment, as well as the time period of interest. The traffic assignment models presented in this paper remain pertinent and can be used in combination with the temporal allocation factors.

### B. Scheduling at the offloading spots

The scheduling refers to the decision whether to load or unload the data from electric vehicles when they arrive at an offloading spot to charge their depleted batteries. In our paper, we consider the flows of vehicles from a *macroscopic* point of view, which abstracts the scheduling problem. Hence, we consider *vehicle flows* (i.e., vehicles going in the same direction) between two adjacent offloading spots within the vehicles' range. The offloading spots know the direction of each flow of vehicles and know, for the vehicles that are charging their batteries, to which flow they belongs.

However, in a *microscopic* point of view, the offloading spots have to make a forwarding decision for each vehicle that

is charging their battery: the offloading spots have to decide whether to unload the data from their on-board storage device and/or load the ready-to-ship data on the vehicle. To make this decision, the offloading spots have to predict the next station the vehicles will stop at. Thus, we advocate the use of a central controller (managed by the service provider) to monitor the status of the offloading spots. This includes the available parking stalls, the free space in the offloading spots' local storage facilities, and the available ready-to-ship data at the offloading spots. In addition, to predict the next direction of the vehicles, the controller may have access to GPS information the vehicles may or may not share (e.g., planned itinerary, next direction, and final destination). The controller also has access to databases, including *traffic forecasting databases* to monitor and update the state of the logical links between the offloading spots, and *customer databases* to predict which station the vehicles are more likely to stop at next, based on the customer history. A confidence estimation evaluates the accuracy of the prediction and is accounted for in the logical link leakage.

### C. Security of the service

The owners of the electric vehicles may try to access the data payload they are carrying. Therefore, the data has to be encrypted so that only the content provider is able to read it. For instance, a public-key infrastructure can be considered (e.g., RSA, ECC). The data is encrypted with the public key of the remote entity (the destination of the data) by the originating entity (the source of the data) and will be in turn decrypted by the remote entity using their private key. Also, we can use certificates to authenticate the originating entity of the data by the remote entity.

Also, we consider a multiple path allocation of the data both on the road infrastructure and on the offloading overlay, which offers path diversity. Therefore, the data that is part of the same flow (or even the same data when replicated) will not take the same path (either road path or logical path). This provides robustness for data transfers, as well as increased security since the data belonging to the same flow will be scattered over multiple paths, making it difficult to attack the whole data transfer by preventing single vantage point attacks.

### D. Data transmission

In the paper, we assumed all data transfer requests to be known in advance and allocated all at once. However, the arrival of data transfer requests may also be modeled as a stochastic process (e.g., following a Poisson process). In this case, upon receiving a new transfer request, the value of the parameters used to process the previous current transfers are updated. Hence, all data transfer requests need to be reallocated on the offloading overlay with the new settings (i.e., the remaining delay tolerance, the remaining amount of data to transfer, and the given leakage tolerance).

We assume data is loaded onto or unloaded from a vehicle using wireless connection that supports high throughputs (e.g., MIMO 802.11ac) [28]. These technologies would allow us to transfer the entire storage device (1 TB) within the battery charging time (20 minutes in our case).

## E. Scalability and fairness

In this subsection, we add more precision on two key aspects of the offloading service we propose.

**Scalability.** The scalability of the offloading service relies on the processing efficiency of the linear programming problems *rep* and *nrep* detailed in Sections V-A and V-B, respectively. These models are adapted from the fractional multi-commodity flow problem, which can be solved using polynomial-time algorithms (e.g., the ellipsoid method or interior point methods) [29]. The complexity of these algorithms can be as low as $O(n^{3.5}L^2)$, where $n$ is the dimension of the problem (the number of variables) and $L$ the number of bits in the input [30].

**Fairness.** The offloading service must also provide sufficient resources for all data transfer requests. Hence, the flow allocation problems must result in a fair flow allocation, while satisfying the data transfer request requirements. Max-min fairness or proportional fairness provide ways to ensure fair flow allocation [31], [32]. Implementation of such techniques is left for future work. However, in this paper, we fine-tune the $\beta$ parameter such that the standard deviation between the allocated flows is minimized.

## F. Comparison with dedicated vehicles

The offloading service we propose relies on the opportunistic mobility of common vehicles. Alternatively, our service may also rely on a dedicated fleet of *ferry vehicles* to transport the data from its source directly to its destination. This problem refers to the *vehicle routing problem* in transportation research [33]. However, in our case, we can relax hypothesis of this problem. First, the vehicles transporting the data do not have to come back to the source of the data. Second, we consider big enough data traffic to load at least the entire storage of a dedicated vehicle, which prevents the vehicle to visit multiple data destinations. These two relaxations greatly simplify the problem. In this case, the offloading cost only depends on $\sigma^{st}$, the cost of transporting one unit of data between the source location $s$ and the destination location $t$ of the data. The total offloading cost is expressed as the difference between the offloading gain and the offloading cost:

$$\sum_{s,t\in\mathcal{R}} \mathcal{B}^{st}\gamma^{st} - \sum_{s,t\in\mathcal{R}} \mathcal{B}^{st}\sigma^{st} = \sum_{s,t\in\mathcal{R}} \mathcal{B}^{st}\left(\gamma^{st} - \sigma^{st}\right).$$

This solution is viable only if $\gamma^{st} > \sigma^{st}$. However, the cost here is proportional to $\sigma^{st}$, which can be expressed as the product of the distance from $s$ to $t$ and the throughput of the data transfer. Conversely, in our offloading system, the cost increases linearly with the load at the offloading spots. Asymptotically, our solution generates more profits. Moreover, our system allows spreading the data of a single transfer over multiple independent vehicles, which increases the robustness of the transfer.

## VIII. RELATED WORK

### A. Vehicular Delay-Tolerant Networking (DTN)

Our proposal somehow falls under the umbrella of delay-tolerant networking as it leverages the properties of delay-

tolerant data [34]. Since there is no end-to-end connectivity, traditional MANET routing protocols cannot be deployed in such networks. Therefore, intermediate nodes use store-carry-and-forward scheduling to transmit data to neighbor nodes whenever there is a contact.

Many propositions make use of delay-tolerant networks to offer opportunistic connectivity to vehicles. Ott *et al.* proposed a drive-thru Internet solution [35], which provides on-the-go connectivity to vehicles (e.g., for on-board entertainment systems) via access points deployed along the roadside. Other proposals also aim at providing vehicular connectivity, such as Cabernet [36] and CarTel [37] designed to deliver data to vehicles when driving near access points. UMass DieselNet leverages opportunistic bus-to-bus contacts to provide end-to-end connectivity. Measurement campaigns helped to collect a dataset used to characterize delay-tolerant networks [38]. In particular, the analysis of the DieselNet dataset led to the proposal of routing protocols as MaxProp [39].

Zhao *et al.* proposed to enhance the performance of delay-tolerant networks by using *throwboxes*, small inexpensive battery-powered devices equipped with storage and wireless interfaces [40]. When placed in strategic points, throwboxes increase the contact opportunities. In contrast to throwboxes, which are fixed nodes, other approaches exploit the mobility of special nodes to increase the throughput of the system. Shah *et al.* proposed the MULE (*Mobile Ubiquitous LAN Extensions*) architecture to provide wide-area connectivity to sparse sensor networks by exploiting the random mobility of humans, animals, and vehicles to opportunistically collect data from the sensors and deliver it to collection points [41]. Instead of considering random node mobility, Zhao *et al.* exploit the non-randomness in the movement of nodes to deliver data [42]. Special mobile nodes, called *message ferries*, carry data between disconnected nodes and act as relays. In this scheme, message ferries can either move according to specific predefined routes, allowing regular nodes to meet up with ferry nodes, or ferries can adjust their trajectory to get closer to regular nodes and exchange data to be delivered. Message ferrying was proposed to solve the problem of Internet connectivity in remote areas, such as DakNet [43], which uses public buses to offer a connectivity to remote villages in developing countries, and KioskNet [44], which intend to deploy a viable low-cost architecture in India.

### B. Bulk data transfers

Examples of large data transfers over the Internet range from exchanges of large scientific datasets to distribution of high-resolution movies.

Grid computing technologies allow scientists to generate, compute, and store data that is distributed across multiple sites. Processing the data implies transferring large volumes of data over wide-area networks. For instance, the Large Hadron Collider at CERN can generate data in the Terabyte-scale that needs to be processed and multi-casted to remote locations for analysis [45]. To this end, file transfer mechanisms have been developed to schedule, monitor, and manage the data transfers over wide area networks so as to meet specific requirements, as

fault tolerance or transfer concurrency. GridFTP is a protocol that achieves secure and reliable data transfers up to few Terabytes per day [46]. Today's large scale data centers are facing these same issues, as the same data has to be replicated in multiple locations to be the closest to the user. Hence, the wide-area network that connects the data centers (inter-data center network) is a critical infrastructure, that must have an excellent efficiency. However, Hong *et al.* found that most inter-data center networks have a poor efficiency and proposed SWAN, a SDN-based scheme to efficiently allocate WAN resources in a congestion-free manner [3].

NetStitcher [47] takes into account unused bandwidth in inter-data center networks (during low link utilization periods) by using multi-path and store-and-forward scheduling at intermediate nodes for bulk transfers. This allows NetStitcher to provide a delay guarantee for the bulk transfer and a good return on investments for dedicated links between data centers.

Although bulk data transfers are at the basis of some popular services such as high-definition multimedia content delivery, many ISPs are using scheduling, traffic shaping, and queue management techniques to limit the rate of bandwidth-intensive applications. An example of service affected by the ISPs policies is Netflix, who is responsible for about 29% of North American fixed Internet access bandwidth utilization [48]. Initially launched as a DVD rent-by-mail company, Netflix began streaming movies online in 2007 in an attempt to avoid the postal costs for delivering DVDs by mail. To address ISPs' rate-limiting policies and the costs of serving content, recommendation algorithms are expected to be used in combination with peer-to-peer networking by big data service providers. The idea behind the use of these algorithms is to determine subscribers' particular interests so as to create geographically logical clusters of subscribers who share the same interests. Instead of pointing subscribers to providers' central servers, P2P network of subscribers' boxes will allow content to be served from other boxes in the same cluster.

Most of the current cloud storage providers offer services that allow their customers to send hard drives to the nearest storage facility. Examples include Amazon Web Services Import/Export[4] or Microsoft Azure Import/Export Service[5]. These services help customers transfer large amounts of data by addressing common challenges including high network costs, long transfer times, and security concerns. Amazon also offers its customers the AWS Import/Export Snowball[6] for rent, an appliance that can transport up to 50 terabytes of data.

### C. Data offloading

With the explosion of the Internet volumes over the past few years, the idea of offloading data from the Internet have become more and more significant. The idea to transfer data by physically moving removable storage media is not new

and is commonly referred as SneakerNet [4]. RFC 1149 (and related later improvements) specifies an experimental method to transmit IP datagrams using avian carriers [14]. Although adapted in some very specific scenarios, this solution becomes unfeasible in large-scale networks due to the resulting high delay and low throughput. The joint use the Internet and postal services (such as FedEx or UPS) for data transfers was first proposed by Wang *et al.* [13] and evaluated by Cho and Gupta [49]. As in the previous example, their proposal relies on a complex scheduling and faces high shipping costs to fit to the specific cases of data transfers that tolerate large delays (in magnitude of days). Proposed in the context of new generation mobile networks such as 4G LTE, mobile traffic offloading make use of complementary network communication technologies (i.e., Wi-Fi) to deliver mobile data traffic [50], [51], [52]. A more recent approach relies on DTN communications to migrate cellular data traffic [53]. Non-realtime information only need to be delivered to a small fraction of users who will in turn resume the dissemination once connected with other users.

In a companion poster, we motivated the use of vehicular offloading by considering a single stretch of road [5]. The results showed that the proposed scheme can be highly efficient in terms of transfer latency even for amounts of data in the petabyte range. Keränen and Ott proposed to transmit data between airports using scheduled flight connections [11]. The authors show that the scheme achieves throughputs in the magnitude of a regular TCP connection by loading the equivalent of three DVDs in the passengers' mobile phones. Zarafshan-Araki *et al.* proposed TrainNet [12] to transport delay-tolerant data on trains, which act as a mechanical backhaul. In TrainNet, trains and stations feature massive data storage and the trains deliver data between stations, similarly to the offloading service we propose. The authors focus on the scheduling, as the train have limited storage available and a limited amount of data can be exchanged at stations. Contrary to our work, planes and trains both travel on predefined routes according to a predefined timetable.

## IX. CONCLUSION

We proposed a bulk-data transfer service to offload delay-tolerant Internet traffic onto the road infrastructure network connecting geographically distant offloading spots. Our service takes advantage of the stops drivers make at charging/swapping stations to opportunistically load data on vehicles or deliver it. To this end, we designed an offloading overlay to mitigate the complexity of the road infrastructure and make the data assignment problem simpler. Regarding this problem, we proposed two linear programming optimization models under different assumptions on the replication of data at the stations. We evaluated our system using real traffic numbers available for the French road infrastructure. Our offloading service allows data transfers in the Petabyte range per week with a market share of 20% and only one terabyte of storage per vehicle. These results confirm the offloading potential of our service, which can help operators handle big data.

---

[4]http://aws.amazon.com/importexport

[5]http://azure.microsoft.com/en-us/documentation/articles/storage-import-export-service/

[6]https://aws.amazon.com/about-aws/whats-new/2015/10/aws-import-export-snowball/

This work opens up many research directions. In our models, we consider replicating data at the source or at intermediary offloading spots. A possible enhancement would be to replicate the data only partially at some offloading spots. Though a request has a single source, the data to be transferred may be expected at multiple destinations. These requests may be handled as multicast transfers. As previously pointed out, our proposal involves some security and privacy concerns, and addressing them is an interesting research problem as well.

## REFERENCES

[1] Cisco Visual Networking Index, "The zettabyte era–trends and analysis," *Cisco white paper*, 2013.
[2] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," *IDC iView: IDC Analyze the Future*, 2012.
[3] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer, "Achieving high utilization with software-driven wan," in *ACM Sigcomm*, Hong Kong, China, Aug. 2013.
[4] D. Patterson, "A conversation with jim gray," *ACM Queue*, 2003.
[5] R. A. Gorcitz, Y. Jarma, P. Spathis, M. D. de Amorim, R. Wakikawa, J. Whitbeck, V. Conan, and S. Fdida, "Vehicular carriers for big data transfers (poster)," in *IEEE Vehicular Networking Conference*, Seoul, South Korea, Nov. 2012.
[6] N. Laoutaris, G. Smaragdakis, P. Rodriguez, and R. Sundaram, "Delay tolerant bulk data transfers on the internet," in *ACM SIGMETRICS Performance Evaluation Review*. ACM, 2009.
[7] European Commission, "Statistical pocketbook 2013," http://ec.europa.eu/transport/facts-fundings/statistics/pocketbook-2013_en.htm, [Online; accessed 10-December-2013].
[8] G. Pasaoglu, D. Fiorello, A. Martino, G. Scarcella, A. Alemanno, C. Zubaryeva, and C. Thiel, *Driving and Parking Patterns of European Car Drivers: A Mobility Survey*. Publications Office, 2012.
[9] Tesla Motors, Inc, "Supercharger," http://www.teslamotors.com/supercharger, [Online; accessed 10-December-2013].
[10] I. R. Limited, "Global charging equipment for ev market 2014-2018," August 2014.
[11] A. Keränen and J. Ott, "DTN over aerial carriers," in *ACM Mobicom workshop on challenged networks (CHANTS)*, Beijing, China, Sep. 2009.
[12] M. Zarafshan-Araki and K.-W. Chin, "Trainnet: A transport system for delivering non real-time data," *Computer Communications*, 2010.
[13] R. Y. Wang, S. Sobti, N. Garg, E. Ziskind, J. Lai, and A. Krishnamurthy, "Turning the postal system into a generic digital communication mechanism," *ACM Sigcomm*, 2004.
[14] D. Waitzman, "A standard for the transmission of ip datagrams on avian carriers," Internet Requests for Comments, RFC Editor, RFC 1149, April 1990. [Online]. Available: http://tools.ietf.org/html/rfc1149
[15] J. de Dios Ortúzar and L. Willumsen, *Modelling transport*. Wiley Chichester, 2011, vol. 7.
[16] Y. Sheffi, *Urban transportation networks: equilibrium analysis with mathematical programming methods*. Prentice Hall, 1985.
[17] P. Chen, E. Lee, G. Gibson, R. Katz, and D. Patterson, "RAID: High-performance, reliable secondary storage," *ACM Computing Surveys*, Jun. 1994.
[18] F. MacWilliams and N. Sloane, *The Theory of Error-correcting Codes: Part 2*. Elsevier, 1977, vol. 16.
[19] R. B. Dial, "A probabilistic multipath traffic assignment model which obviates path enumeration," *Transportation Research/UK/*, vol. 5, 1971.
[20] United States. Bureau of Public Roads, *Traffic assignment manual for application with a large, high speed computer*, 1964.
[21] National Research Council (U.S.). Transportation Research Board, *Highway Capacity Manual*, 2000.
[22] DGITM/DIT – SETRA – IGN, "Recensement de la circulation sur le réseau routier national en 2011," http://tinyurl.com/otfbewv.
[23] T. Wright, P. S. Hu, J. Young, and A. Lu, "Variability in traffic monitoring data."
[24] R. Church and C. R. VELLE, "The maximal covering location problem," *Papers in regional science*, 1974.
[25] MEDDE, "Mobilité – déplacement," http://tinyurl.com/otfbewv.
[26] CCFA, "Analysis and statistics 2012," http://www.ccfa.fr/IMG/pdf/ccfa_gb_ra12b_bat.pdf, [Online; accessed 10-December-2013].
[27] T. M. Guide, "Us department of transportation," *Federal Highway Administration*, September 2013.
[28] R. Van Nee, "Breaking the gigabit-per-second barrier with 802.11 ac," *IEEE Wireless Communications*, 2011.
[29] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, "Network flows: theory, algorithms, and applications," 1993.
[30] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *ACM symposium on Theory of computing*, 1984.
[31] M. J. Neely, E. Modiano, and C.-P. Li, "Fairness and optimal stochastic control for heterogeneous networks," *Networking, IEEE/ACM Transactions on*, 2008.
[32] M. Pióro and D. Medhi, *Routing, flow, and capacity design in communication and computer networks*. Elsevier, 2004.
[33] P. Toth and D. Vigo, *The vehicle routing problem*. Society for Industrial and Applied Mathematics, 2001.
[34] K. Fall, "A delay-tolerant network architecture for challenged internets," in *ACM Sigcomm*, Karlsruhe, Germany, Aug. 2003.
[35] J. Ott and D. Kutscher, "A disconnection-tolerant transport for drive-thru internet environments," in *IEEE INFOCOM*, 2005.
[36] J. Eriksson, H. Balakrishnan, and S. Madden, "Cabernet: vehicular content delivery using wifi," in *ACM MobiCom*, 2008.
[37] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden, "A measurement study of vehicular internet access using in situ wi-fi networks," in *ACM MobiCom*, 2006.
[38] X. Zhang, J. Kurose, B. N. Levine, D. Towsley, and H. Zhang, "Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing," in *ACM MobiCom*, 2007.
[39] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks." in *IEEE INFOCOM*, 2006.
[40] W. Zhao, Y. Chen, M. Ammar, M. Corner, B. Levine, and E. Zegura, "Capacity enhancement using throwboxes in dtns," in *IEEE MASS*, 2006.
[41] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Networks*, 2003.
[42] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *ACM MobiHoc*, 2004.
[43] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: Rethinking connectivity in developing nations," *Computer*, 2004.
[44] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav, "Low-cost communication for rural internet kiosks using mechanical backhaul," in *ACM MobiCom*, 2006.
[45] G. Khanna, U. Catalyurek, T. Kurc, R. Kettimuthu, P. Sadayappan, I. Foster, and J. Saltz, "Using overlays for efficient data transfer over shared wide-area networks," in *IEEE SC*, 2008.
[46] W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, L. Liming, and S. Tuecke, "Gridftp: Protocol extensions to ftp for the grid," *Global Grid ForumGFD-RP*, 2003.
[47] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter bulk transfers with NetStitcher," in *ACM Sigcomm*, Toronto, Canada, Aug. 2011.
[48] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang, "Unreeling netflix: Understanding and improving multi-cdn movie delivery," in *IEEE INFOCOM*, 2012.
[49] B. Cho and I. Gupta, "Budget-constrained bulk data transfer via internet and shipping networks," in *ACM ICAC*, Karlsruhe, Germany, Jun. 2011.
[50] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting mobile 3G using WiFi," in *ACM MobiSys*, San Francisco, CA, USA, Jun. 2010.
[51] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: how much can WiFi deliver?" *IEEE/ACM Transactions on Networking*, 2013.
[52] P. Raveneau, R. Stanica, M. Fiore, S. Uppoor, M. Cunche, H. Rivano, and Z. Smoreda, "Urban-scale Cellular Offloading through Wi-Fi Access Points: a Measurement-based Case Study," in *RTSI*, Turin, Italy, 2015.
[53] Y. Li, G. Su, P. Hui, D. Jin, L. Su, and L. Zeng, "Multiple mobile data offloading through delay tolerant networks," *(preprint) IEEE Transactions on Mobile Computing*, 2013.