Software-Defined Vehicular Backhaul

Benjamin Baron, Prométhée Spathis, Hervé Rivano, Marcelo Dias de Amorim, Yannis Viniotis, Joseph Clarke

Software-Defined Vehicular Backhaul

Benjamin Baron[⋆], Prométhée Spathis[⋆], Hervé Rivano[‡], Marcelo Dias de Amorim[⋆], Yannis Viniotis[†,◇], and Joseph Clarke[◇]

[⋆]CNRS/LIP6 and Sorbonne Universités, UPMC Univ Paris 06 [‡]Inria, U. de Lyon, INSA-Lyon CITI [†]NC State University [◇]Cisco

***Abstract*—The network of roads and highways is a promising candidate to help network operators offload their infrastructure and cope with the ever-growing amount of data exchanged on the Internet. By piggybacking data onto vehicles, roads can be turned into a large-capacity transmission system when considering the increasing number of journeys involving vehicles. The data to be transferred is opportunistically loaded on or off the vehicles at specific locations referred to as offloading spots. Two of the main challenges of such a system are how to assign the road paths matching the data transfer requirements and how much data to allocate to each flow of vehicles. We propose a centralized SDN-like architecture consisting of a central controller acting as a service broker and the offloading spots as SDN agents. The controller computes the road paths that accommodate the data transfer requirements and installs the corresponding forwarding states at each offloading spot along those paths. We describe our SDN-controlled offloading system and evaluate its performance using road traffic counts from France. Our numerical results show that the controller can achieve efficient and fair allocation of multiple data transfers between major cities of France. Each transfer successfully delivers over 10 PB of data within a week when considering that 10% of vehicles on the road are equipped with 1TB of storage.**

I. Introduction

We consider a large-scale offloading system that takes opportunistic advantage of the mobility of vehicles to offload traffic from infrastructure networks, such as the Internet. In a previous work [1], we proposed to equip common vehicles with storage devices, as they can be turned into data carriers while making their routine daily journeys. The flow of vehicles traveling the roads act as a mechanical backhaul connecting specific locations referred to as *offloading spots*. Offloading spots behave as data exchange relays where vehicles may be parked nearby and long enough to transfer the data.

In contrast to human-carried devices commonly found in DTNs [2], [3], vehicles present the advantage of covering large geographical areas at higher speeds with very low, if any, power resource constraints. Previous works [3], [4], [5], [6] have confirmed the feasibility of using the mobility of vehicles for routing in intermittently connected networks. Data MULEs [3] or message ferries [4] acting as middle nodes take on the responsibility of delivering messages to overcome network partitions. The message delivery ratio is improved by controlling the middle nodes in their movements. To increase contact opportunities between message ferries, stationary wireless devices called throwboxes [5], are placed strategically and act as intermediary exchange relay nodes. In Daknet [6], bus-based ferrying provides basic Internet connectivity to rural villages in developing nations. The focus

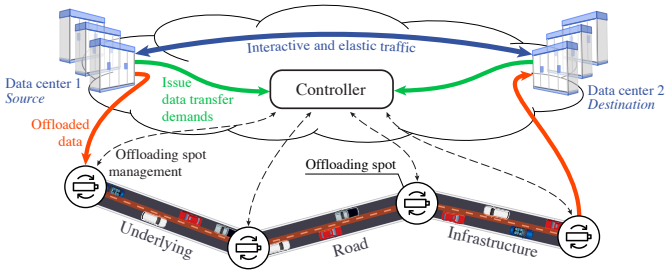


Fig. 1: SDN-controlled vehicular backhaul. The controller receives the request to offload a bulk transfer of delay-tolerant data between two data centers. The controller configures the road network data plane by changing the forwarding behavior of the offloading spots. The flows of vehicles traveling between those offloading spots are allocated to carry data towards the destination.

of most of this research has been on routing in networks operating in challenging environments, as vehicles carrying data can enhance the coverage of these networks to remote locations with poor or non-existing connections. In our work, we leverage on the increasing number of common vehicles driven and miles traveled [7] to offload large chunks of data from an infrastructure-based network such as the Internet. Our offloading system deployed throughout France's road network can transport up to 120 exabytes in a single day if each vehicle in circulation carry only 1 TB of data.

In this paper, we present an SDN-based approach for flexible and scalable configuration of the network of offloading spots to enable efficient transfers of data carried by road vehicles. We present the benefits of the SDN paradigm in the context of a vehicular backhaul in Section II. A central remote controller is in charge of mapping data transfers onto a sequence of offloading spots. Each offloading spot acts as a forwarding entity that forwards the offloaded traffic according to its flow table. This table holds a set of flow entries, each of which contains the list of actions to be performed on the vehicles matching the flow entry. Typical actions include loading data on or off the vehicles passing by the offloading spots. The controller defines these actions based on the information each offloading spot reports on the flow of vehicles passing through the offloading spot. To realize this architecture, we address the two following challenges.

The first challenge we face is how the controller computes the road paths and configures the road network data plane that can accommodate the data transfer requirements. Fig. 1 shows a scenario where a large amount of delay-tolerant background data needs to be transferred between two remote data centers. Such data transfers may result from provisioning or mainte-

nance activities required for offline virtual machine migration or backups between data centers. Data offloaded from the infrastructure network follows the path which consists of the flow of vehicles traveling the stretches of road connecting the offloading spots, and acts as dictated by the controller. This helps avoid the costs of leasing a dedicated line [8].

The second challenge we face is how much data to allocate to each flow of vehicles traveling the road paths. To maintain efficient utilization, we need to design scalable allocation mechanisms, given the high degree of complexity of the road networks topology and the large number of daily routine journeys involving vehicles [7]. To address this challenge, the controller solves the data transfer allocation problem as a multi-commodity flow allocation model to determine the network path consisting of a sequence of offloading spots. The controller then configures the data plane by changing the behavior of each offloading spot. The controller may also dynamically modify their existing behavior in case the vehicles change direction unexpectedly or to account for new data transfers as they arrive.

In summary, the contributions of this paper are:

- **SDN-controlled vehicular backhaul.** We present an SDN-based approach that enables efficient control of the road infrastructure to offload bulk delay-tolerant traffic from an infrastructure network.
- **Dynamic vehicle allocation.** We design a scalable allocation mechanism that centrally computes the road paths matching the performance requirements of a data transfer.
- **Real-traffic-based evaluation.** We evaluate our approach for multiple reliable data transfers assigned on the French road network using actual road traffic counts.

SDN provides a logical centralization that enables efficient configuration of the road infrastructure to offload bulk traffic transfers. Our results show that transfers of 10 PB each can be offloaded and transferred on the roads in no less than a week on distances of several hundreds of kilometers.

II. Vehicular offloading backhaul

In the following, we first give an overview of our vehicle-based scheme for offloading bulk delay-tolerant traffic from an infrastructure network. We then describe an SDN-based architecture for flexible and scalable configuration of the road network to enable efficient transfers of the offloaded traffic.

A. Offloading scheme overview

Vehicles are equipped with one or more removable memory storage devices such as magnetic disks or other non-volatile solid-state storage devices. The term vehicle refers to both passenger and commercial vehicles. In the latter case, it may be part of a fleet of vehicles owned or leased by a business or a governmental agency. Vehicles also embed network communication interfaces. The flow of vehicles so equipped acts as a mechanical backhaul connecting specific locations referred to as offloading spots. The term offloading spot refers to specific locations acting as intermediate exchange relay points where vehicles may be parked nearby and long enough to transfer the data. For example, the offloading spots can be located in highway rest areas, shopping center parking lots, or at drivers' homes or at the office. The offloading spots exchange the data to be transferred with the vehicles using short-range radios.

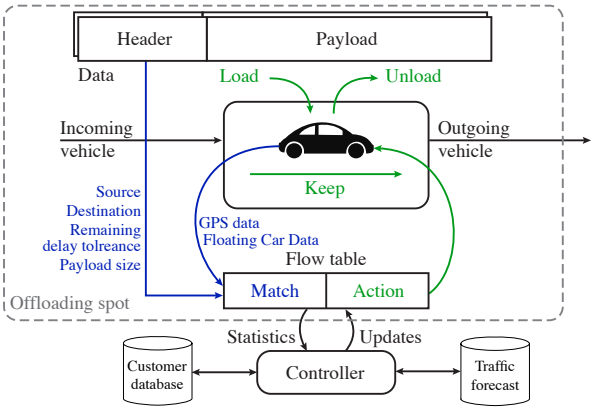


Fig. 2: Forwarding at an offloading spot. The controller installs the forwarding states, including the list of actions to perform upon the arrival of a vehicle. The offloading spot acts as a data exchange relay where cargo data can be dropped off for future pick-up or transferred on an empty vehicle.

Every time a vehicle comes into contact with an offloading spot, the direction of the vehicle is matched against the destination of the data. A sequence of consecutive offloading spots may be involved if the data needs to be shipped across a large area of country before reaching geographically distant destinations. The offloading spots enable the transshipment of data as they can be dropped off for future pick-up by subsequent passing vehicles. The data is therefore carried hop-by-hop through the network of offloading spots toward the intended destination. Given the increasing number of vehicles driven and miles traveled, large chunks of data can be offloaded from an infrastructure network such as the Internet. Vehicles can also provide communication service by overcoming partitions in sparse networks.

In Section III, we investigate a use case involving electric vehicles that are expected to charge or replace their depleted batteries at charging stations. These stations can serve as offloading spots as the data is loaded on or off the vehicles while they charge or replace their batteries.

B. SDN-driven vehicular offloading

SDN-driven data transfers. SDN is a networking paradigm that enables faster deployment of new services, while supporting key features such as network virtualization [9]. SDN separates the data plane from the control plane and facilitates provisioning and configuration of network connections. Operators can change the network behavior in a centralized fashion, while avoiding any dependency on proprietary protocols and configuring independently multiple highly hardware specialized devices. In this work, we leverage on the advantages of the logical centralization provided by SDN to enable efficient control of the road infrastructure to offload bulk delay-tolerant traffic from an infrastructure network.

Central controller and offloading spots. The SDN-based architecture we propose consists of two components: a central

controller and the offloading spots acting as forwarding entities. The offloading spots maintain a flow table populated and manipulated by the controller. The controller receives the requests to offload data transfers on the road network. Each request specifies the delay and bandwidth requirements for the corresponding data transfer. The controller keeps track of the status of the offloading spots, which includes statistics about the passing vehicles. Each offloading spot also reports information about the data locally transshipped. The controller, which collects the information gathered from the offloading spots, has thus an up-to-date view of the offloading infrastructure.

Offloading spot flow tables. The controller computes the road network paths that can accommodate the data transfer requirements and how much data to allocate to each flow of vehicles. Each road network path consists of a sequence of offloading spots to which the controller connects to install a new flow table entry. A flow table consists of flow entries, each corresponding to a set of data transfers. A flow entry indicates the next-hop offloading spot computed for the data transfers corresponding to this entry. As depicted in Fig. 2, a flow table entry also contains a list of actions to perform on each vehicle traveling in the direction of the next offloading spot specified by the flow entry. Each action defines the forwarding behavior for the data belonging to the same transfer.

Forwarding process. The offloading spots make the forwarding decision by looking up the destination of a vehicle in their flow tables, as shown in Fig. 3. The destination of a passing vehicle can be made available through a positioning system such as the vehicle navigation system that generates routes and guidance towards a destination. The vehicles' historical locations are also stored in a geographic location database managed at the controller to help the offloading spots predict the remaining itinerary of the passing vehicles [10].

Upon the arrival of a vehicle, an offloading spot matches the destination of the vehicle against the flow table. If no matching flow entry is found, the vehicle unloads its cargo data, if any, at the offloading spot. If a flow table entry is found, the offloading spot belongs to the road network path computed for a set of data transfers. The matching flow table entry defines the actions to perform on the data belonging to those transfers. The data can either be already stored at the offloading spot or carried on the passing vehicle. If no such data can be found, no actions are performed and the vehicle continues its journey.

If the vehicle already carries data that belongs to one of the transfers represented by the matching flow entry, the data is left on the vehicle that continues its journey. Otherwise, if the cargo data is not consistent with the flow entry, this data is transhipped at the offloading spot for future pick-ups. In case of an empty vehicle, the offloading spot checks whether data matching the vehicle direction was locally transshipped from a previous vehicle or transloaded from the infrastructure network. If such data is locally ready to be shipped, the data is transferred onto the passing vehicle. Otherwise, the vehicle continues its journey without a transfer.

Data leakage. The offloading spots buffer the data transferred on the vehicles for later recovery in case a vehicle unexpectedly changes direction or if the vehicle runs into an accident or breaks down. Such events result in losses we will take into account by introducing a parameter named the link leakage (see Section III).

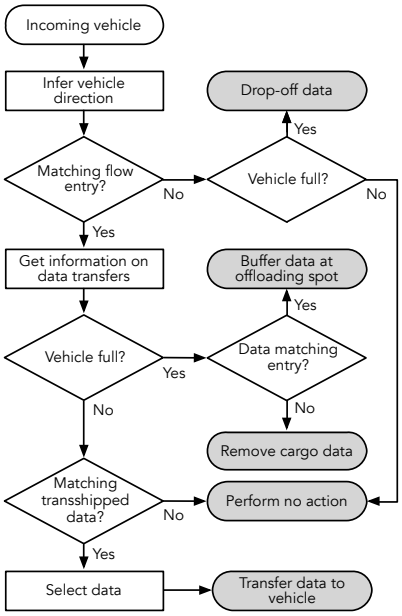


Fig. 3: Forwarding process at the offloading spot.

III. Dynamic vehicle allocation

While the offloading architecture presented in Section II is conceptually straightforward, we must address two challenges to realize its design. First, we need a scalable architecture to cope with the high degree of complexity of the road network's topology and the large number of daily routine journeys involving vehicles [7]. Second, we need an efficient allocation process that maximizes the road traffic utilization, while matching the performance requirements of the offloaded data transfers. Updates to allocation decisions are required for maintaining high utilization in face of changes in the road traffic and for processing new data transfers.

We present a dynamic allocation method to serve the demand for vehicles to carry data offloaded from an infrastructure network. This method integrates the computation of the data road paths, the selection of the data to be carried, and the amount of data allocated to each path. The allocation uses as inputs the data transfers characterized by the amount of data to transfer and the *data shelf time*, which refers to the time data may be stored or in-transit before becoming unfit for use. We also use as inputs the routes available between the transfers' source and destination, the vehicle traveling time, and, for each stretch of road, the vehicle traffic volume (i.e., vehicles per time unit). The controller uses the information on the road infrastructure to compute a logical map of the underlying road. This map represents an offloading overlay that mitigates

the complexity of the substrate network and makes dynamic allocation applicable. The controller computes the data transfer assignment and configures the offloading spots' data plane. The forwarding states are updated to take account of changes in the road traffic and for new data transfers.

A. Offloading spots

Without the loss of generality, we assume the vehicles are powered by on-board electrical motor and that they travel long distances. Electric vehicles offer the opportunity of capitalizing on the frequent need of battery recharging (e.g., 200 miles for the Tesla Model S), as well as the slow process of charging (e.g., 20 minutes when charging at a Tesla supercharger) [11]. The vehicles charge their depleted batteries at charging stations acting as offloading spots (e.g., the network of superchargers that Tesla is currently rolling out in Europe and North America). The data transloaded from the Internet is loaded on or off the vehicles during the charging time of their batteries.

B. Offloading overlay

We use the mapping algorithm that creates an offloading overlay on top of the road infrastructure [1]. Nodes in the offloading overlay correspond to the offloading spots in the substrate network and are connected through logical links, which correspond to road paths consisting of consecutive stretches of road in the underlying road infrastructure. A logical link (i, j) connecting two offloading spots i and j is characterized by the following attributes: the average travel time $t(i, j)$ needed to reach offloading spot j for a vehicle leaving offloading spot i, the capacity $c(i, j)$ representing the amount of cargo data carried per unit of time on (i, j), and the data leakage $l(i, j)$, which refers to the loss rate due to vehicles unexpectedly changing direction, hijacked vehicles, accidents, or break downs.

C. Reliable data transfers

To mitigate the effect of leakage along the logical path taken by a data transfer, the controller requires data to be replicated before the transfer begins. We propose to replicate data according to redundancy techniques such as RAID [12]. The vehicles' storage devices are configured in a RAID arrangement. More specifically, we use RAID level 6, which protects arrays of storage devices against up to two device failures. The data originating from the same source is divided into N arrays of $n \geqslant 4$ storage devices including two redundant storage devices, $n-2$ being available for use. We assume the data leakage equivalent to the failure probability of storage devices, as both are independent and identically distributed. We can express the data linkage experienced by a data transferred on link (i, j) protected by RAID 6 redundancy, denoted by $l^{st}_{\text{red}}(i, j)$, in terms of the data leakage $l(i, j)$ without data redundancy as follows:

$$l^{st}_{\text{red}}(i, j) = 1 - \sum_{k=0}^{2} \binom{n}{k} l(i, j)^k \big(1 - l(i, j)\big)^{n-k}, \quad (1)$$

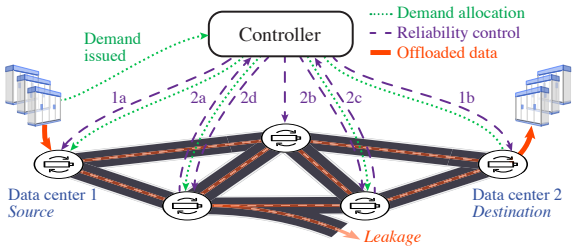


Fig. 4: A data legacy network allows the controller to access and change the forwarding behavior of the offloading spots between two data centers (dotted arrows). Controller-to-offloading spot communication is also responsible for reliable data delivery by replicating data (dashed arrows 1a-1b) and retransmitting unacknowledged data (dashed arrows 2a-2d).

where n is the number of storage devices arranged in RAID 6.

Replications increase the amount of data sent by a factor w^{st}_{red}. We denote by $\mathcal{B}^{st}$ the amount of data to transfer between s and t and by $\mathcal{S}$ the storage capacity of the vehicles. For a data transfer involving n storage devices arranged in N arrays, assuming $\mathcal{B}^{st} \bmod \mathcal{S}(n-2) = 0$, $w^{st}_{\text{red}} = {}^{n}/_{n-2}$.

Besides RAID redundancy techniques, we also propose to rely on an ARQ mechanism to achieve reliable data transfers over the road network. We use SR-ARQ (*Selective-Repeat ARQ*) [13] to retransmit the data lost as a consequence of the link leakage. Consider a data transfer passing through a logical link (i, j). Once the data transferred to a vehicle at offloading spot i, the controller notifies offloading spot j when to expect the arrival of the vehicle (arrows 2a and 2b in Fig. 4). If the vehicle fails to arrive on time, offloading spot j sends a NAK (*negative acknowledgment*) back to offloading spot i through the controller (arrows 2c and 2d in Fig. 4). The data is retransmitted either carried by subsequent vehicles or via the infrastructure network after a number of failed attempts. Communication between the controller and the offloading spots incurs low bandwidth overhead and may be handled via a legacy data network. To analyze the performance of this retransmission method, we assume the buffer capacity of the offloading spots unlimited and data losses mutually independent. We assume that the feedback is noiseless, as no NAK losses occur. SR-ARQ introduces an overhead denoted by $w^{st}_{\text{ret}}(i, j)$ equal to $R^{st}(i, j)$, the average number of retransmissions needed to successfully send data over a logical link (i, j). We express $R^{st}(i, j)$ in terms of $l^{st}_{\text{red}}(i, j)$ the link leakage of (i, j), as follows:

$$R^{st}(i, j) = w^{st}_{\text{ret}}(i, j) = \frac{1}{1 - l^{st}_{\text{red}}(i, j)}. \quad (2)$$

Retransmissions combined with data redundancy help ensure reliable data transfers over the road network. They both incur a total overhead O^{st} given by:

$$O^{st} = w^{st}_{\text{red}} \times w^{st}_{\text{ret}}(i, j) \times f(p),\ p \ni (i, j), \quad (3)$$

where $f(p)$ refers to the rate achieved by the flow of vehicles allocated for a data transfer and entering logical path p. The number of storage devices per array, denoted by n, is defined such that the total overhead is minimized.

D. Data transfer allocation

We formulate the data transfer allocation problem as a linear programming (LP) model that determines the logical paths matching the performance requirements of a data transfer. The controller solves the LP for each data transfer request and allocates cargo data to the vehicles traveling the corresponding logical paths. In the following, we denote by $\mathcal{P}^{st}$ the set of all possible logical paths between s and t, a pair of source and destination. Each logical path $p \in \mathcal{P}^{st}$ consists of a sequence of logical links connecting pairs of offloading spots in the offloading overlay.

The travel time $t(p)$ experienced by a data transfer allocated to the logical path p is the sum of travel times $t(i,\, j)$ of each logical link along p and the waiting time t_i at each offloading spot i along the path. The travel time $t(p)$ also depends on the average number of retransmissions needed on each logical link to successfully transmit data hop-by-hop:

$$t(p) = \sum_{(i,\, j) \in p} \big[R^{st}(i,\, j) t(i,\, j) + t_i\big]. \tag{4}$$

The time needed to move $\mathcal{B}^{st}$ data between s and t is given by:

$$\frac{\sum_{p \in \mathcal{P}^{st}} f(p) t(p)}{\sum_{p \in \mathcal{P}^{st}} f(p)} + \frac{\mathcal{B}^{st}}{\sum_{p \in \mathcal{P}^{st}} f(p)} \leqslant \mathcal{T}^{st}, \tag{5}$$

where the first term is the average travel time of the flow of vehicles traveling on each logical path p, weighted by $f(p)$. The second term is the loading time needed to transfer the $\mathcal{B}^{st}$ data onto the vehicles. We note that the loading time is accounted only for the first offloading spot. Finally, the transfer time is constrained by $\mathcal{T}^{st}$, the shelf time of the data to transfer.

Eq. (5) can be re-written in a linear form as follows:

$$\sum_{p \in \mathcal{P}^{st}} f(p)\big(\mathcal{T}^{st} - t(p)\big) \geqslant \mathcal{B}^{st}. \tag{6}$$

The rate $f(p)$ achieved by the flow of vehicles traveling along the logical path p is constrained by the capacity of the logical links forming p. The amount of carried traffic also includes the overhead incurred by SR-ARQ retransmissions and RAID level 6 data redundancy. We account for this overhead by using w^{st}_{ret} and w^{st}_{red} respectively as follows:

$$\sum_{s,\, t} w^{st}_{\text{red}} \sum_{\substack{p \in \mathcal{P}^{st} \\ p \ni (i,\, j)}} w^{st}_{\text{ret}}(i,\, j) f(p) \leqslant c(i,\, j), \tag{7}$$

where $c(i,\, j)$ is the capacity of the logical link $(i,\, j)$.

The objective of the data transfer allocation problem is to maximize the overall throughput achieved by the flows of vehicles assigned to each data transfer, subject to the constraints of road capacity and data shelf time. To compute the road paths that match the data shelf time, while maintaining efficient road utilization, we solve the following linear program:

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

Subject to

$$\sum_{s,\, t} w^{st}_{\text{red}} \sum_{\substack{p \in \mathcal{P}^{st} \\ p \ni (i,\, j)}} w^{st}_{\text{ret}}(i,\, j) f(p) \leqslant c(i,\, j) \qquad \forall (i,\, j)$$

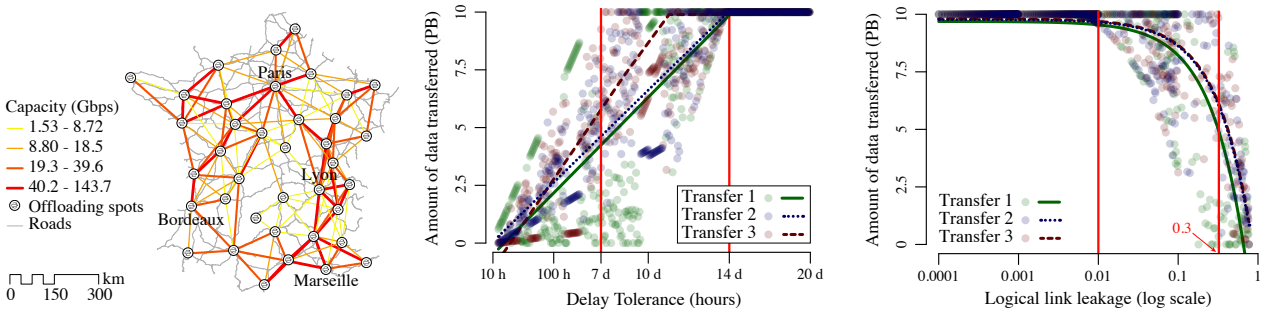
$$\sum_{p \in \mathcal{P}^{st}} f(p)\big(\mathcal{T}^{st} - t(p)\big) \geqslant \mathcal{B}^{st} \qquad \forall (s,\, t)$$

IV. Results

Evaluation setup. To evaluate our dynamic vehicle allocation method, we construct a real offloading network infrastructure at the scale of an area the size of France. We devise a concrete deployment plan of battery charging stations, as shown in Fig. 5, covering the entire French territory. This plan extends the driving range of electric vehicles, while minimizing the number of charging stations. The charging stations are located 150 km apart and their placement is determined by solving a variation of the covering location problem [14]. We connect the neighboring charging stations via a set of disjoint alternative routes selected in the road map of France by running the algorithms presented in [15]. The selected routes share up to 80% of the shortest route, while their length is not less than 80% of the shortest route's distance. To estimate the traffic volume of these routes, we use the C-logit traffic assignment model [16]. This model works by assigning a weight to the routes connecting the pairs of offloading spots located in a radius of 300 km (i.e., the driving range of the electric vehicles). We then use the entropy maximization model proposed by Zuylen and Willumsen [17] to infer the origin-destination traffic matrix for the network of charging stations. We feed this model with the actual traffic counts provided by the AADT (*Annual Average Daily Traffic*)[1] of the major roads in France covering a combined distance of 20,000 km [18]. Finally, we expect our service to be gradually adopted. We consider a conservative market penetration ratio of 10%, which represents the share of vehicles equipped with on-board storage devices. Each vehicle has a data cargo capacity of 1 TB.

Data transfer demand allocation. We evaluate the LP allocation of three data transfers of 10 PB each on top of the offloading network constructed as described above. The three transfers are shown in Fig. 5: from (1) Paris to Lyon with solid green line and green dots, (2) Paris to Bordeaux with dotted blue line and blue dots, and (3) Paris to Marseille with dashed red line and red dots. We consider that all the links in the offloading overlay have the same link leakage of 30%. Fig. 5b shows the amount of data transferred as a function of the delay tolerance applied to all transfers. The plotted data results are fitted using a segmented regression model. We observe that the LP allocation provides fairness across transfers 1 and 2, as the two cover roads share similar traffic density. We can see that transfer 3 carries noticeably more data than transfer 2, as the roads to reach Marseille offer higher capacities (see Fig. 5a) in the case of transfer 3. The vehicle flows allocated for all

[1]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.



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

(b) Amount of data transferred as a function of the delay tolerance.

(c) Amount of data transferred as a function of the link leakage.

Fig. 5: Evaluation of our offloading service.

three transfers successfully deliver the 10 PB of data within a maximum of 14 days. We can see that some transfers are completed in no more than 7 days. We can extend those results by stressing our offloading system to its limit. If all vehicles in circulation on the roads of France carry 1 TB of data, the total amount of en-route data adds up to 120 exabytes over a single day. Fig. 5c shows the amount of data that can be transferred in a limit of 7 days as a function of the link leakage in the overlay network. The scatter plots are fitted using a log regression model. We can see that the LP allocation provides fairness across all transfers. All transfers are successfully completed for a link leakage less than 0.01 and each transfer delivers a total of 10 PB of data.

V. Conclusion and Perspectives

In this article, we have identified the benefits of the SDN paradigm in the context of a vehicular backhaul network consisting of offloading spots acting as intermediary data exchange relays. Vehicles traveling the roads connecting the offloading spots take on the responsibility of delivering large amount of delay-tolerant data to remote destinations. To overcome the high degree of complexity of the road networks topology, we proposed an SDN-like architecture consisting of a central controller in charge of mapping the data transfers onto a sequence of offloading spots. The logical centralization provided by SDN enables efficient control of the road infrastructure to offload bulk delay-tolerant traffic from an infrastructure network. To maximize the amount of data transferred onto the vehicles, the controller solves the data transfer allocation problem as a multi-commodity flow allocation model that determines the road paths matching the performance requirements of the offloaded data transfers. SDN allows flexible and scalable configuration of the offloading spots' data plane. The controller modifies the forwarding behavior of the offloading spots by installing specific actions defined for matching flows of vehicles. We evaluate our approach for multiple reliable data transfers assigned on the French road network using actual road traffic counts. With 10% of vehicles on the road equipped with 1 TB of storage, our results show that 10 PB of data can be offloaded in a single transfer covering several hundreds of kilometers, while delivered in no less than a week.

As future work, we plan to extend our architecture by transferring the forwarding capabilities of the offloading spots to the vehicles, as data can be exchanged without requiring stationary data exchange relays. We also intend to equip vehicles with sensing and processing capabilities, as they can be turned into mobile sensors in the context of Smart Cities and the Internet of Everything.

References

[1] B. Baron, P. Spathis, H. Rivano, M. D. De Amorim, *et al.*, "Vehicles as big data carriers: Road map space reduction and efficient data assignment," in *VTC2014-Fall*, 2014.

[2] K. Fall, "A delay-tolerant network architecture for challenged internets," in *ACM Sigcomm*, (Karlsruhe, Germany), Aug. 2003.

[3] 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.

[4] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *ACM MobiHoc*, 2004.

[5] W. Zhao, Y. Chen, M. Ammar, M. Corner, B. Levine, and E. Zegura, "Capacity enhancement using throwboxes in dtns," in *IEEE MASS*, 2006.

[6] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: Rethinking connectivity in developing nations," *Computer*, 2004.

[7] J. Krumm, "How people use their vehicles: Statistics from the 2009 national household travel survey," tech. rep., SAE Technical Paper, 2012.

[8] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *ACM SIGCOMM*, 2008.

[9] O. N. Fundation, "Software-defined networking: The new norm for networks," *ONF White Paper*, 2012.

[10] J. Krumm and E. Horvitz, "Predestination: Inferring destinations from partial trajectories," in *UbiComp*, Springer, 2006.

[11] Tesla Motors, Inc, "Supercharger." http://www.teslamotors.com/supercharger.

[12] P. Chen, E. Lee, and G. Gibson, "RAID: High-Performance, Reliable Secondary Storage," *ACM Computing*, 1994.

[13] S. Lin and D. J. Costello, *Error control coding*. Prentice-hall Englewood Cliffs, 2004.

[14] C. Toregas, R. Swain, C. ReVelle, and L. Bergman, "The location of emergency service facilities," *Operations Research*, 1971.

[15] I. Abraham, D. Delling, A. V. Goldberg, and R. F. Werneck, "Alternative routes in road networks," *Journal of Experimental Algorithmics (JEA)*, 2013.

[16] E. Cascetta and A. Nuzzolo, "A modified logit route choice model overcoming path overlapping problems: specification and some calibration results for interurban networks,"

[17] H. J. Van Zuylen and L. G. Willumsen, "The most likely trip matrix estimated from traffic counts," *Transportation Research Part B: Methodological*, 1980.

[18] DGITM/DIT – SETRA – IGN, "Recensement de la circulation sur le réseau routier national en 2011." http://tinyurl.com/otfbewv.