DROiD: Adapting to Individual Mobility Pays Off in Mobile Data Offloading

Filippo Rebecchi, Marcelo Dias de Amorim, Vania Conan

DROiD: Adapting to Individual Mobility Pays Off in Mobile Data Offloading

Filippo Rebecchi[◇,†], Marcelo Dias de Amorim[◇,⋆], and Vania Conan[†]
[◇]UPMC Sorbonne Universités [⋆] CNRS [†]Thales Communications & Security
{filippo.rebecchi, marcelo.amorim}@lip6.fr vania.conan@thalesgroup.com

Abstract*—Cellular operators count on the potentials of offloading techniques to relieve their overloaded data channels. Beyond standard access point-based offloading strategies, a promising alternative is to exploit opportunistic direct communication links between mobile devices. Nevertheless, achieving efficient device-to-device offloading is challenging, as communication opportunities are, by nature, dependent on individual mobility patterns. We propose, design, and evaluate DROiD (Derivative Re-injection to Offload Data), an original method to finely control the distribution of popular contents throughout a mobile network. The idea is to use the infrastructure resources as seldom as possible. To this end, DROiD injects copies through the infrastructure only when needed: (i) at the beginning, in order to trigger the dissemination, (ii) if the evolution of the opportunistic dissemination is below some expected pace, and (iii) when the delivery delay is about to expire, in order to guarantee 100% diffusion. Our strategy is particularly effective in highly dynamic scenarios, where sudden creation and dissolution of clusters of mobile nodes prevent contents to diffuse properly. We assess the performance of DROiD by simulating a traffic information service on a realistic large-scale vehicular dataset composed of more than 10,000 nodes. DROiD substantially outperforms other offloading strategies, saving more than 50% of the infrastructure traffic even in the case of tight delivery delay constraints. DROiD allows terminal-to-terminal offloading of data with very short maximum reception delay, in the order of minutes, which is a realistic bound for cellular user acceptance.

***Index Terms*—Mobile data offloading; hybrid networks; delay-tolerant networks.**

I. Introduction

We propose DROiD, a feedback based offloading scheme to efficiently distribute popular contents to a multitude of mobile users. DROiD helps cellular operators to relieve their infrastructure network by exploiting direct communications between users. The recent boom in the smart mobile devices market calls for efficient offloading strategies, as the global mobile traffic is expected to increase significantly (18-fold between 2011 and 2018 as reported by Cisco [1]). Cellular providers are already under heavy pressure, attempting to accommodate such an amount of traffic on their networks. As a consequence, they must intervene with major investments to scale their access networks. Nevertheless, expenses to buy more licensed band or to build more base stations are very high; unfortunately, the increase in network capacity brought by these methods will hardly keep up with traffic growth. Recent studies disclosed an alternative solution when many co-located users are interested in the same contents [2], [3], [4].

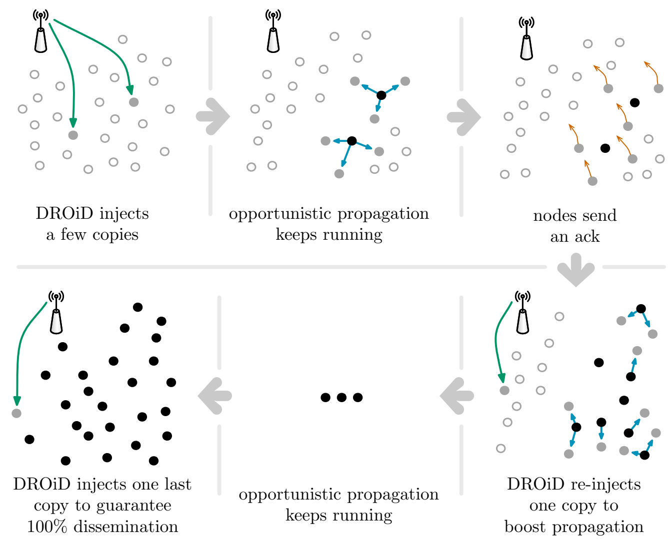


Fig. 1: DROiD functioning: the infrastructure channel initially injects copies to kick start the dissemination process. Content is diffused through the opportunistic communications. Upon content reception, users acknowledge the offloading agent using the feedback channel. The infrastructure channel may re-inject copies to boost the diffusion. 100% delivery ratio is reached through fall-back re-injections.

The idea is to benefit from node mobility and delay tolerance of a number of content types to help the infrastructure to shift a portion of the traffic from the *primary* (cellular) channel to an *alternative* (terminal-to-terminal) channel. Carriers may offer incentives and pricing discounts to motivate mobile subscribers to offer their battery and storage resources to this end [5].

The core mechanism behind DROiD aims at alleviating the load on the operator's infrastructure by reducing redundant traffic. DROiD adapts to the heterogeneous individual mobility pattern of nodes and to the *current* evolution of the dissemination process. This heterogeneity is at the base of a stepwise evolution in the content dissemination, alternating flat zones (plateaux) to steep periods of diffusion, as we will explain in Section II. DROiD adopts the re-injection decision analyzing the outlook of the diffusion rather than comparing the infection level to an objective function. In this way, DROiD detects plateaux in the content diffusion evolution, and, if needed, *adaptively re-injects additional copies in the system to finely control the pace at which the contents are disseminated*. To this extent, a persistent feedback channel, connecting mobile users with the offloading coordinator, becomes instrumental, allowing at each instant the coordinator to track the content dissemination status and to anticipate the correct re-injection decision. We show a high level functioning schema of DROiD in Fig. 1. Thanks to its adaptive re-injection strategy, DROiD leads to much better performance than existing strategies that are bounded to an objective function that remains fixed during the entire offloading process [6].

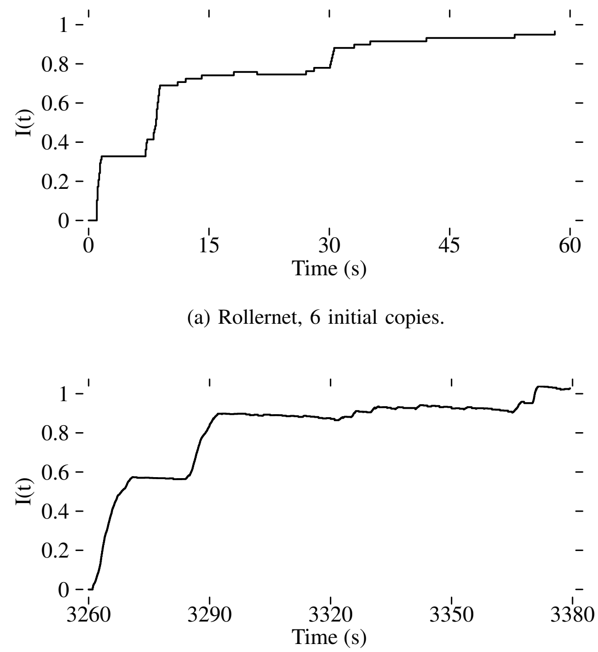
We confirm the proper functioning of DROiD through simulation, where we mimic a location-based service; a content, supposed of general interest, must be distributed to a multitude of users within a given maximum reception delay (in order to guarantee a minimal QoS on a per-content basis). We employ a realistic large-scale vehicular trace derived from multiple fine-grained traffic measurements in the city of Bologna (about 10,000 vehicles). We then compare DROiD's performance under tight delays with other offloading solutions proposed in literature, and with an oracle, taken as a benchmark. Simulation results display that DROiD substantially outperforms objective function-based offloading strategies for any considered delay tolerance value, reducing by more than half the infrastructure load. In addition, we show that DROiD performs very close to the oracle. As a summary, the contributions of this paper are threefold:

- We turn the attention to the heterogeneity of contact patterns in opportunistic networks. This heterogeneity is at the origin of the dynamic creation and dissolution of clusters. We harness this property to explain why epidemic diffusion presents a stepwise behavior.
- We propose DROiD, an offloading system that, thanks to its derivative-based re-injection strategy, better adapts to the contact patterns between nodes to offer enhanced offloading efficiency.
- We compare DROiD with other objective function-based strategies and show that DROiD outperforms them even under tight delivery delay constraints.

The remainder of this paper is organized as follows. In Section II, we motivate our work showing a typical stepwise behavior of epidemic diffusion in opportunistic networks. In Section III, we introduce DROiD and motivate the choice of a derivative-based feedback strategy. In Section IV, we describe the scenarios and the dataset we used to evaluate our proposal. Several results are presented in Section V. We postpone related work to Section VI and conclude the paper in Section VII.

II. Motivating Context: Stepwise epidemic diffusion

At the heart of DROiD is the idea of allowing the diffusion process to adapt to the idiosyncrasies of individual mobility patterns. To see what happens, let us take two examples using very different datasets: the small-scale Rollernet dataset, composed of only 62 nodes [7], and the Bologna dataset, composed of more than $10,000$ nodes. We plot in Fig. 2 the evolution of the content diffusion in the network. We start the diffusion by injecting a small number of initial copies (6 and 10 respectively) to random nodes at t_0, and let the epidemic diffusion of the message progress with subsequent direct



(a) Rollernet, 6 initial copies.

(b) Bologna, 10 initial copies

Fig. 2: Epidemic diffusion of the content. The diffusion behavior alternates steep zones and flat zones that are the result of changing encounter probability among mobile nodes.

contacts. A node that has received the message is said *infected*, while a node that has not yet received the content is *sane*. The instantaneous infection ratio $I(t) \in [0, 1]$ follows a stepwise pattern, alternating plateaux (flat areas) to periods of heavy infection (steep areas) before reaching complete diffusion. We may find a similar dissemination evolution pattern for very different datasets such as the ones considered. This is a typical example of the way any given diffusion process progresses due to the randomness of contact patterns in opportunistic networks. In particular, the plateaux correspond to periods during which the dissemination does not make any progress, because no sane nodes come into range of the already infected nodes.

Let us now dig into the relationship between mobility patterns and progress of the epidemic diffusion. The first obvious point is that this phenomenon is intrinsically related to the heterogeneity of contact patterns, i.e., the fact that two different nodes do not meet on average the same number of other nodes. If the contact process of nodes is Poisson homogeneous and stationary, each pair of nodes meets with intensity λ. Assuming that each contact means an opportunity to transfer the content, neglecting the contact duration, the resulting epidemic diffusion follows the logistics equation [8] - the curve does not exhibit any plateau, which is in contrast to our observation. Since for homogeneous contact process, each pair of nodes meet with the same probability, the longer the time that a copy has to propagate and the greatest benefit that copy brings. This would lead us to wrongly believe that the best offloading strategy is to inject the right amount of copies at t_0, forget about the infection evolution, and let the opportunistic dissemination do the work for us.

To capture the heterogeneity of patterns, we adopt a Marked Poisson Process model of node contacts [9], [10]. In this model, the meeting times of any two nodes (i, j) follow a Poisson Process with rate $\lambda_{ij} = \lambda p_{ij}$. The inter-contact times T_{ij} are thus independent exponentials with parameter λ_{ij}, and the matrix $C = (p_{ij})$ captures the patterns of interactions between nodes. In the homogeneous case, C is the identity matrix, i.e., all nodes can see each other with the same probability. At any given time instant of the dissemination process, a set S of nodes is infected. We are interested in the random plateau duration T_S^p during which the dissemination does not progress. This corresponds to the random time during which this set of nodes does not meet any other nodes. Looking at the set of links between nodes in S and its complement, one can see that $T_S^p = \inf_{i \in S, j \notin S} T_{ij}$. By Poisson calculus, and noting the cut value $\partial S = \sum_{i \in S, j \notin S} p_{ij}$, T_S^p is an exponential random variable with parameter $\lambda \partial S$ [11]. The expected plateauing duration, once set S has been reached, is thus $1/\lambda \partial S$.

This simple argument shows that T_S^p is directly related to the structural properties of the contact matrix C, providing a natural connection between the community structure of the contact graph and the progression (or lack of progression) of the opportunistic dissemination process. Nodes in a community have high conductance (they are well knit to one another), and the ratio of the weight of inter-cluster edges to the total weight of all edges is low [12]. Applying these ideas to C (which represents the probability of two nodes to meet) means that a community S of users will spread the message quickly within the cluster (high conductance), but will reach a plateau once the nodes in the group all have the message, because the weight of inter-cluster edges and thus its cut value ∂S is low.

In practice, we observe strong dynamic clustering for the considered datasets. As a working example, we confirm this statement through numerical analysis of the Rollernet dataset. Since the matrix C varies in time, we compute it for $t \in [0, 60]$ seconds. As we know the set of nodes that have been initially infected in the realization, and how diffusion progresses, we employ the analytical model described above to calculate the average duration T_S^p for the first plateau in Fig. 2a, starting at $t = 1.7$ seconds. At that time, S contains 18 infected nodes, 12 of which have been infected opportunistically. The numerical analysis gives us an estimated exponential parameter $\lambda \partial S = 0.21$, an expected plateau duration $T_S^p = 4.78$ seconds. The observed plateau duration of the realization lasts 5.4 seconds, confirming the legitimacy of our intuitions. This observation provides the motivation of our further investigation of adaptive offloading strategies that are able to chase the individual mobility of nodes, re-injecting copies when the diffusion evolution runs into a plateau.

III. Shrinking the cellular load: A Derivative-based Solution

We design DROiD following the observations discussed in Section II. We will first present the general architecture of our system before providing the design details.

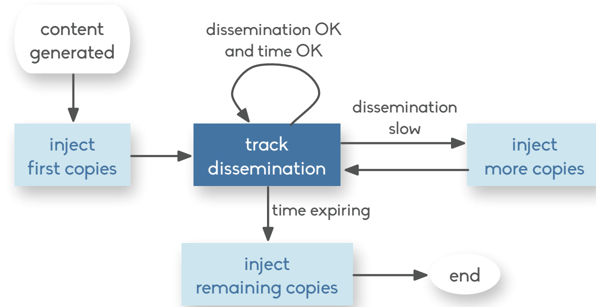


Fig. 3: Flowchart of the high-level operation of DROiD.

A. High-level operation

The proposed system employs a *Pub-Sub* paradigm, with users sending `subscribe` messages upon entering the simulation area, and `unsubscribe` messages upon leaving it, in order to acknowledge their interest to the *offloading coordinator*. A subset of subscribers initially receives content through the cellular channel and propagates it opportunistically using the ad hoc interface. Whenever a node receives content from a neighbor, it acknowledges the reception to the coordinator through the infrastructure network, forming a feedback loop in the system. This simple mechanism allows DROiD to monitor in real time the evolution of the content dissemination process, and possibly to account for data usage. The offloading coordinator continually estimates the infection ratio and may decide to re-inject additional copies of the content in order to boost the diffusion. Since acknowledgments sent by mobile nodes on the infrastructure channel are relatively lightweight (compared to the size of the disseminated content), the proposed system is expected to guarantee considerable reduction of the infrastructure load.

Opportunistic communications depend heavily on the particular mobility of nodes, and only probabilistic guarantees of successful content delivery and reception times can be given. To solve this issue, when we approach the maximum delivery delay D (i.e., the validity of content), and the time left is equal to the time required to send the message through the infrastructure, denoted as P, the offloading coordinator enters a *panic zone* and pushes the content to all uninfected nodes through the infrastructure, guaranteeing full dissemination. Note that the feedback loop guarantees a fall-back method to overcome various issues that may appear in the network, such as node failures or mobile users behaving selfishly – the occurrence of these events could heavily impact the opportunistic diffusion [13]. The high-level operation of DROiD is illustrated in Fig. 3.

B. Derivative-based re-injection strategy

The core of DROiD resides on the intelligence of the offloading coordination agent. This latter is in charge of deciding *when* to re-inject additional copies depending on the evolution of the opportunistic dissemination process. Every re-injection decision is expected to bring benefit to the system, but it depends on the re-injection time and the target node (to which copies will be sent through the infrastructure). In fact, there is a difficult trade-off to consider. On the one hand, if too many copies are injected in the beginning (in general, earlier injections have more time to diffuse), the system may be overestimated (as we do not know in advance how nodes will encounter). On the other hand, if the system injects too few copies in the beginning and waits for the panic zone to compensate for lags, many opportunistic encounters might be wasted because of the lack of enough copies in the network. Re-injection is beneficial when the subsequent opportunistic transmissions save additional infrastructure pushes. Of course, the benefit can be null if the offloading coordination agent selects a node that would have received the message later from another node. Finding the good trade-off is difficult, as the offloading agent is essentially blind and the only information available is the list of currently subscribed users and the list of those who already received the content (inferred from acknowledgments).

DROiD achieves high offloading efficiency by making the re-injection decision dependent not only on the actual dissemination level, but also on the trend of the infection ratio. DROiD anticipates and avoids the insurgence of long-lasting plateaux in the content diffusion through its reactive strategy. This is not the case in other strategies such as Push-and-Track, in which the offloading agent makes the re-injection decision according to the distance between the instantaneous infection ratio and a fixed *a priori* target objective function [6]. Re-injection decisions do not take into account the general evolution of the infection, but only its instantaneous values. Static strategies react too late when the infection ratio is above the objective function but still not evolving, or overreact when the infection evolves well but its instantaneous value still lies under the objective function. Late or too brutal re-injections result in a waste of messages pushed through the infrastructure. Another limitation of existing static strategies is that they do not propose a single solution, but instead a multitude of objective functions; the problem is that different objective functions behave differently depending on the content lifetime and network status.

DROiD keeps in memory a short snippet of past infection ratio values. Each content has an associated tracker that stores the evolution of the infection ratio for a temporal sliding window of size W (i.e., at time t the values that will be considered are the ones between $[t-W,t]$). W is a design parameter that must be a multiple of the time step used for the evaluation of the infection ratio, τ, and smaller than D (recall, the validity of the content). The size of the sliding window trades off how far in time DROiD looks back and dictates the reactivity to sudden changes in the infection ratio. In our experiments, we found that reasonable values of W fall in the range $[2\tau, 10\tau]$.

As illustrated in Fig. 4, at evaluation time step t, the offloading coordinator performs a forward difference quotient

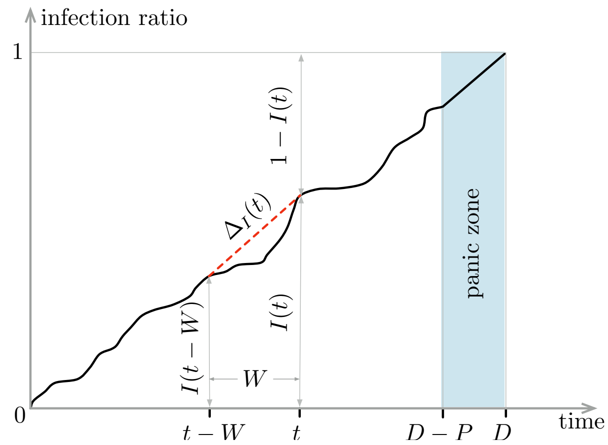


Fig. 4: Discrete time slope detection performed by DROiD. For clarity we consider the content creation time $t_0 = 0$.

on the instantaneous infection ratio $I(t)$ that approximates to a discrete derivative:

$$\Delta_I(t) = \begin{cases} \frac{I(t)-I(t-W)}{W}, & t-t_0 \geq W, \\ \frac{I(t)}{t-t_0}, & t-t_0 < W. \end{cases} \tag{1}$$

Note that $I(t)$ is not monotonically increasing, since nodes may exit the simulation area at any time. $\Delta_I(\cdot)$ approximates the slope of the infection ratio and is one of the parameters that influence the re-injection decision. DROiD re-injects additional copies of the content whenever the discrete derivative $\Delta_I(\cdot)$ is below a Δ_{lim} threshold computed on line as the ratio between the fraction of sane nodes and the time remaining before the panic zone. This because a steeper slope is needed when time gets closer to panic zone or the infection ratio lags (different from when we are at the beginning of the infection process). Formally speaking, we have:

$$\Delta_{\text{lim}}(t) = \frac{1-I(t)}{(D-P)-(t-t_0)}. \tag{2}$$

As a final step, the injection rate $r_{\text{inj}}(t)$ is computed as a piecewise function, depending on the ratio of the current $\Delta_I(t)$ value and the Δ_{lim} threshold:

$$r_{\text{inj}}(t) = \begin{cases} c, & \Delta_I(t) \leq 0, \\ c\left[1 - \frac{\Delta_I(t)}{\Delta_{\text{lim}}(t)}\right], & 0 < \Delta_I(t) \leq \Delta_{\text{lim}}(t), \\ 0, & \Delta_I(t) > \Delta_{\text{lim}}(t). \end{cases} \tag{3}$$

Where $c \in [0,1]$ is a clipping value used to limit the overall amount of re-injected copies in the case of negative values of Δ_I.[1] Finally, $r_{\text{inj}}(t)$, which represents the percentage of uninfected nodes that need to be targeted, is multiplied to the

[1]Clipping value c represents the maximum fraction of content that can be pushed through the infrastructure at evaluation time. Negative values of $\Delta_I(\cdot)$ may happen in the case of infected nodes leaving the system. We address this issue in Section V.

number of uninfected nodes to find the number $\mathcal{R}(t)$ of copies to re-inject at t:

$$\mathcal{R}(t) = \lceil (1 - I(t)) \times |N(t)| \times r_{\text{inj}}(t) \rceil. \tag{4}$$

Where $|N(t)|$ is the instantaneous number of nodes subscribed to the content update.

IV. Evaluation setup, scenarios, and dataset

We evaluate DROiD considering the problem of distributing popular content to a multitude of mobile nodes. We assume that nodes are equipped with two wireless interfaces (e.g., most smartphones or infotainment systems), so that they are able to communicate through two interfaces simultaneously. Possible combinations involve 3G and 4G to communicate with the cellular infrastructure and Bluetooth or Wi-Fi ad hoc to communicate with neighboring devices.

A. Experimental Dataset

To evaluate our strategy, we use a large-scale vehicular mobility trace representing the city of Bologna (Italy), and consisting of $10,333$ nodes. This dataset, initially exploited to evaluate cooperative road traffic management strategies within the FP7 iTetris project [14], covers 20.6 km^2 comprising 191 km of roads. The dataset is drawn from real traffic measurements acquired by 636 induction loops deployed citywide and inferred into a micro-mobility model through the SUMO simulator [15]. The dataset captures real city traffic conditions, with speed of vehicles varying from 0 to around 50 km/h depending on road congestion. From the mobility trace, we derive a contact trace that features contacts between nodes when the distance between them is below a given threshold (we consider in our analysis a range of 100 meters). The final contact trace has duration of about one hour; in average, $3,500$ nodes are present at the same time (because some nodes leave while others join during the observation period). Differently from other datasets available, we have a clear high turnover rate and no apparent social links between nodes. The distribution of contact duration is exponential. Most contacts are very short [6], confirming the highly dynamic nature of the dataset. Only few contacts last for more than a few minutes.

B. Scenario variations

Without loss of generality, we consider location-based traffic information service, where a centralized server issues a new content update every t_P seconds. DROiD must guarantee the delivery of each of the updates to all nodes interested in the content within a maximum delay D. Contents are issued periodically, with the previous one expiring when a new one is created (so $t_P = D$, and a single content is active in the system at a time). Possible contents of interest include popular geo-relevant data, such as localized traffic and roadwork alerts, generalized public utility information or geographic advertising; nevertheless, the proposed system also supports the efficient distribution of software updates. The choice of which content to offload is made in advance, depending on its delay-tolerant characteristics. The content may be delivered directly through the infrastructure network or retrieved from a neighboring node in an opportunistic fashion. Despite this work considers all users as interested in the content, the combined use of the *Pub-Sub* paradigm and `ack` messages makes the system easily extensible in the case of multiple contents and non uniform nodes' interest.

Since nodes may enter and leave the target area during the lifetime of the content (and this impacts the results, as we will see later), we consider two different scenarios in our analyses.

Scenario 1: Partial population. In this scenario, target recipients are nodes that remain in the target area during the whole update content lifetime (i.e., during the period $[t_0, t_0 + D]$). Initial recipients of the traffic update are the users already subscribed to the update service (e.g., mobile nodes within the interest area). Users subscribing during the distribution period are served on a best-effort way (they may receive the update but have to wait for the next update to be sure to be served). Nodes that leave the target area are no further concerned with the content and lose their status of potential recipient. Nevertheless, latecomers and nodes leaving before $t_0 + D$ can still participate in the opportunistic dissemination, concurring to increase the delivery capacity of the system. Formally speaking, we denote with $N(t)$ the set of subscribed users at time $t \in [t_0, t_0 + D]$, $S = N(t_0)$ the set of potential target recipients of the message (i.e., the subscribed nodes when the update is issued), and $K = N(t_0 + D) \cap S$ the subset of potential target nodes that remained until $t_0 + D$ (the nodes that *must* receive the update).

Scenario 2: Full population. In this scenario, we target the distribution of updates to *any* node that are part of the network for any period of time within $[t_0, t_0 + D]$. We define the set of all target nodes for the content as $K = \bigcup_t N(t), \forall t \in [t_0, t_0 + D]$.

C. Simulation setup

No network simulators among those publicly available today perform well in scenarios with several thousand nodes at the same time [16], [17]. Therefore, we built a streamlined event based simulator heavily inspired by the ONE simulator [16]. In our implementation, we consider a simple contact-based ad hoc MAC model, where a node may transmit only to a single neighbor at a time. Transmission times are deterministic since we do not take into account complex phenomena that occur in the wireless channel such as fading and shadowing (we are not really interested here on the exact physical evolution of communications taking place during the offloading process). Communications consist of two different classes of messages (content and control). All transfers, including `ack` messages, may fail due to nodes moving out of each other's transmission range or exiting the simulation area. In addition, it is possible that the same message be concurrently received through the two interfaces. In that case, we consider the one that is processed first. The ad hoc routing protocol employed by nodes to disseminate the content is the epidemic forwarding.

Parameters in simulation are set to mimic the functioning of communication technologies currently available to consumers. In each simulation run, the downlink bit-rate for the infrastructure network is set to 100 KB/s, while uplink is fixed at 10 KB/s. These values are in line with the average bit-rate experienced by users of a typical HSPA network. The bit-rate for the ad hoc link is set to 1 MB/s, also in line with the advertised bit-rate of the IEEE 802.11p standard. The size of each content update is set at 100 KB. The size of the acknowledgment messages is 256 bytes, as it carries very little information (content and node identifiers). For the other parameters, we use $\tau = 1$ seconds, $c = 0.05$, and $W = 5\tau$. The panic time duration P is fixed at 1 second.

V. Results

A. Comparison with existing strategies

We investigate how our system performs under tight delivery constraints, when the maximum delivery delay D lies in the range $[30, 180]$ seconds. This contrasts with what is done in most approaches in the literature that consider long time scales for content reception (up to some hours). Instead, we are interested in very short maximum reception delays, in the order of minutes, as otherwise users would not realistically accept to trade-off reception delays for cellular capacity. State-of-the-art solutions, benefiting from more relaxed reception constraints, can take advantage of a sort of stochastic regularity in contact patterns of users [2], [3], [4]. Centralized optimization frameworks based on Monte Carlo sampling [2] or *temporal reachability graphs* [18] require the complete contact graphs among users, and are known to have high computational complexity. Therefore, they are unable to evaluate the offloading strategy on large-scale datasets in real-time. Indeed, if the mobility patterns of subscribers change, the selected strategy might not be optimal anymore. In addition, none of the proposed strategies deal with nodes entering or leaving the system. These considerations make it difficult to compare existing approaches with DROiD, which targets the microscopic mobility of users and the unpredictable contact dynamics on small time intervals.

B. Reference strategies and evaluation

We use two reference strategies for evaluation purposes: “infrastructure only” (Infra) and “connected component oracle” (Oracle). In the Infra strategy, there is no offloading at all, and the infrastructure represents the only means of distributing content. In the Oracle strategy, the coordinator has a real-time picture of the ad hoc connectivity of the entire network (unrealistic assumption, but useful to provide an upper bound on performance). The coordinator pushes the content to a random node within each existing connected component. Singleton nodes are targeted as well. The underlying idea is to push only one copy per connected component in order to get close to the minimal number of infrastructure copies. Oracle achieves near-optimal performance because of its perfect knowledge of the system connectivity; however, it does not account for transmission times and future movements of nodes.

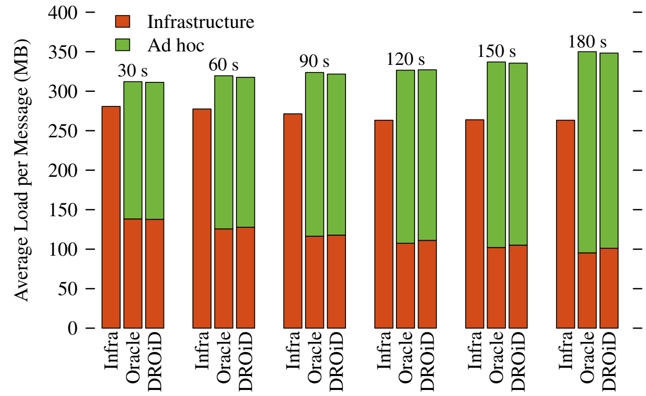


Fig. 5: Partial population scenario: infrastructure vs. ad hoc load per message sent using the Infra, the Oracle, and the DROiD strategies. Different maximum reception delays for messages are considered.

In addition to these two baseline cases, we compare DROiD with Push-and-Track, which represents nowadays the offloading alternative that offers 100%-delivery ratio guarantees with tight delivery times. Since it offers primarily a methodology rather than a specific offloading strategy, it is difficult to state a priori which target objective function gives the best results [6]. To be as fair as possible, we compare DROiD with the objective function that gives, for each scenario, the best results, namely the *linear* and the *slow start* objective functions.

All the results presented in this section are averages over 10 simulation runs. We focus primarily on the aggregate load that flows through the infrastructure and across the ad hoc links. Load measurements take also into account `ack` messages as well as failed and aborted transfers. `Ack`, `subscribe`, and `unsubscribe` messages constitute the infrastructure overhead. The offloading efficiency metric depends on the amount of traffic flowing on the infrastructure link when we use the offloading process, denoted with L, and the traffic on the infrastructure in the absence of any offloading strategy (i.e., Infra strategy), denoted with L_{ref}. Formally, the offloading efficiency is computed as $1 - L/L_{ref}$.

C. Partial population scenario

DROiD performs very well in terms of reduced infrastructure load, by delivering the majority of traffic through device-to-device communications even in the case of tight delays. Fig. 5 displays the average amount of traffic per message that flows through the infrastructure and ad hoc interfaces. In this picture, we compare DROiD to reference strategies only, to show how DROiD approaches Oracle. An interesting phenomenon appears: while the sum of the infrastructure and ad hoc load for both the oracle and DROiD increases with the message lifetime, the reference load for the Infra strategy slightly decreases. This particular effect depends on the scenario under consideration, as we target nodes that remain in the system for the entire lifetime of the content. Intuitively, an increase in the maximum reception delay makes nodes more likely to exit the simulation area before the deadline,

TABLE I: Partial population scenario: infrastructure overhead (%) for different strategies and reception delays.

	30s	60s	90s	120s	150s	180s
Oracle	0.43	0.48	0.51	0.55	0.59	0.64
DROiD	0.43	0.47	0.51	0.54	0.58	0.62
Static Linear	0.42	0.45	0.46	0.48	0.51	0.54
Static Slow Start	0.40	0.45	0.47	0.48	0.49	0.53

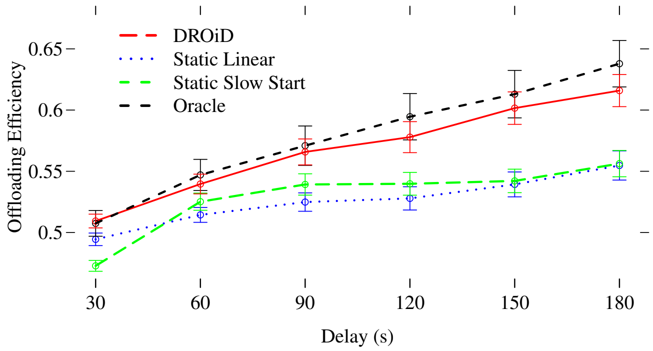


Fig. 6: Partial population scenario: average offloading efficiency. Different maximum reception delays for messages are considered. 95% confidence intervals are plotted.

reducing the number of reference infrastructure messages. Ad hoc load increases, instead, because nodes that are not direct targets still participate in the opportunistic dissemination. This ad hoc overuse is not particularly worrisome, since direct transmissions have no monetary costs associated; nevertheless this may result in congestion in dense networks. On the other hand, the overhead caused by `ack` messages on the cellular channel is kept at minimum level. Thanks to the small size of the `ack` messages, the feedback mechanism is never responsible for more than 0.64% of the infrastructure load (values in Table I). As expected, there is a linear relationship between the overhead and the number of contents received through ad hoc links (and the resulting `ack` messages).

In terms of saved infrastructure load, DROiD pays only a small penalty compared to the reference oracle. For instance, in the 30-seconds scenario DROiD performs nearly as well as Oracle. When the delay tolerance increases, the performance of Oracle improves, as the opportunistic dissemination has more time to propagate the message to the entire network. Oracle is able to choose the most favorable nodes at t_0, resulting, in average, in fewer re-injections during panic time.

DROiD always obtains better performance than static strategies. In Fig. 6, DROiD always outperforms the static linear and slow-start strategies in terms of offloading efficiency. Note that the results would have been even better if we had picked another objective function. The gap between DROiD and static strategies increases when the tolerance to the delay increases, suggesting a better adaptation to the diffusion evolution. This curve shows also a well-known phenomenon: an increase in the reception delay corresponds to an increase in the

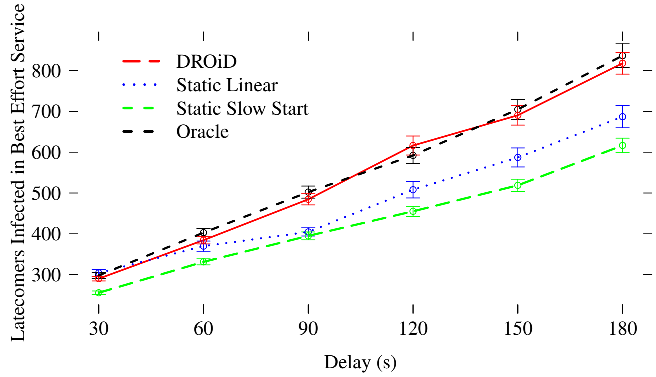


Fig. 7: Partial population scenario: average amount of latecomer users served in best-effort service. 95% confidence intervals are plotted.

offloading efficiency. If we translate the offloading efficiency into aggregate infrastructure traffic savings, for a one-hour-long simulation with 30-seconds reception delay, DROiD saves around 420 MB if compared to the linear strategy and 1.2 GB if compared to the slow start strategy (numbers not shown in the figure). These values, although not really sound in absolute terms, give an idea on the potential savings that the network operator would be able to obtain when it has to handle multiple parallel requests. Fig. 6 also confirms that objective function-based strategies strongly depend on content lifetime and network status, with a relative performance that varies with the delay. DROiD, on the other hand, consistently offers better performance. Even if we can tolerate only a small maximum reception delay as short as of 30 seconds, DROiD offloads more than 50% of the infrastructure traffic.

We also confirm the advantages of DROiD by analyzing the number of infected latecomer nodes. Recall that latecomers are served in a best-effort fashion. They also participate in the ad hoc diffusion of the content, increasing the overall network delivery capacity. Fig. 7 presents the average number of latecomer nodes infected in a best effort service. Also in this case, DROiD obtains much better dissemination ratios than the two fixed objective functions, especially with longer delays. It is worth noting that DROiD can even lead to better levels of infection of latecomers than Oracle when the delay tolerance is 120 seconds. The number of latecomers hints at the dynamics of the considered dataset.

D. *Full population scenario*

DROiD performs even better in the *full population* scenario. Recall that, in this scenario, all nodes entering the target area are expected to receive the content, regardless of their dwell time in the system. Therefore, unlike in previous scenario, the load in the Infra strategy increases with the message lifetime. Simulation results, plotted in Fig. 8, show that DROiD uses roughly the same infrastructure load of the oracle to guarantee 100%-delivery ratio. Sudden variations in the infection ratio, due to nodes that dynamically enter and leave, are well handled by the feedback mechanism of DROiD. While the load in the Infra strategy increases linearly (as a longer

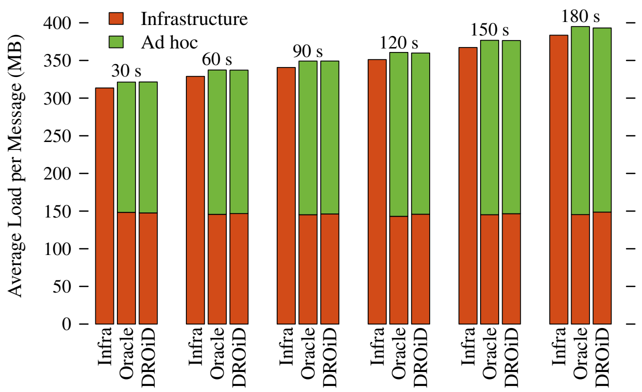


Fig. 8: Full population scenario: infrastructure vs. ad hoc load per message sent using the Infra, the Oracle, and the DROiD strategies. Different maximum reception delays for messages are considered.

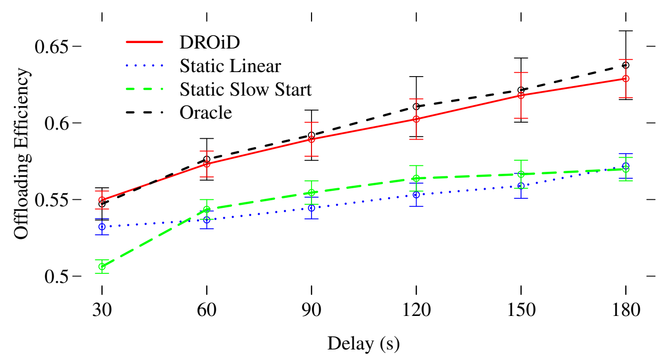


Fig. 9: Full population scenario: offloading efficiency. Different maximum reception delays for messages are considered. 95% confidence intervals are plotted.

TABLE II: Full population scenario: infrastructure overhead (%) for different strategies and reception delays.

	30s	60s	90s	120s	150s	180s
Oracle	0.43	0.48	0.50	0.54	0.57	0.64
DROiD	0.43	0.48	0.51	0.54	0.57	0.61
Static Linear	0.42	0.44	0.47	0.50	0.52	0.56
Static Slow Start	0.40	0.45	0.47	0.50	0.52	0.55

content lifetime implies a major number of nodes entering the system), the infrastructure loads for Oracle and DROiD remain always nearly the same, translating in an increased offloading efficiency. The ad hoc transmissions overuse is significantly less pronounced in this scenario, and is dominated by failed and aborted transfers due to nodes moving out of each other's transmission range, or messages concurrently received on both interfaces. Table II compares the infrastructure overhead due to control messages. Similarly to the previous case, infrastructure overhead results negligible, accounting for at most 0.64% of total cellular traffic.

Compared to static strategies, DROiD always leads to better results, as shown in Fig. 9. In this scenario, DROiD saves between 55% and 63% of traffic for different message delays. In terms of aggregate savings, for a one-hour-long simulation and 30-seconds tolerance to delay, DROiD saves around 360-MB compared to the linear strategy, and 1.44-GB compared to the slow start strategy. These numbers are again very motivating if we consider the total amount of cellular traffic that an operator could save in the case of a real deployment.

Although DROiD and Oracle show more or less the same trend in the offloading efficiency curve, this result is achieved through two completely different strategies. On the one hand, Oracle, exploits its perfect knowledge of the connectivity status in the network, pushing the content to specific high potential nodes. On the other hand, DROiD has a much less complete, and slightly out of sync, view of the system, and employs its advanced derivative-based re-injection algorithm to guess when additional copies of the content are required. In addition, Oracle presents always larger confidence interval than DROiD (and static strategies in general). This is related to the mobility and turnover of nodes: the resulting connectivity changes in time influencing the Oracle prediction performance. A mobility and connectivity agnostic framework such as ours is less sensitive to this issue.

VI. Related Work

Recent mobile data explosion fostered the interest in alternative methods to relieve the load on the wireless infrastructure. Mobile data offloading is seen as a simple and inexpensive method to improve the capacity of mobile networks. The mobility of users and the delay tolerance of some contents has been the subject of a number of papers in the literature.

Balasubramanian et al. propose Wiffler, a system to augment the capacity of a cellular network by exploiting opportunistic associations with Wi-Fi APs [19]. Yetim et al. consider the decision of waiting for Wi-Fi hotspot encounters rather than using cellular connectivity as a linear programming scheduling problem [20]. Lee et al. present a quantitative study on the benefit of data offloading through APs [21]. Dimatteo et al. propose MADNet, an architecture that integrates cellular, Wi-Fi APs, and mobile-to-mobile communications to offload the cellular network [22]. Trestian et al. propose to upgrade the network capacity only at a selected number of locations, called Drop Zones, in the movement patterns of a large number of users [23].

A more recent approach exploits device-to-device communications, mobility of end-nodes, and the popularity of certain contents to offload data through opportunistic communications. It is noteworthy to point out that also 3GPP is focusing on device-to-device communication technology as a viable offloading solution [24]. Han et al. identify the opportunity to offload data exploiting the social ties between users, proposing a subset selection mechanism based on the history of contacts [2]. Similarly, Li et al. mathematically formulate the problem of DTN-based traffic offloading of multiple contents. Under the assumption of Poisson contact rates between nodes, they study the optimal subset selection as a problem of utility function maximization under multiple constraints [4]. Barbera et al. analyze contacts between end-nodes in order to select a subset of seed VIP users that are socially important in terms of centrality for the network. The main idea is to turn these few central VIP nodes into data forwarders between other nodes and the Internet [3]. Ioannidis et al. assume that the cellular infrastructure has a fixed bandwidth that should be allocated between different end-users. Mobile users share opportunistically any stored content with other users, in order to improve the overall network capacity [25]. Unlike these optimization frameworks, our system continues in the footsteps of Push-and-Track [6], not requiring any training period, or knowledge of the mobility patterns of users. We developed a simpler reactive strategy able to make up for this lack of knowledge through the use of the cellular channel to control content dissemination.

VII. Conclusion

In this paper, we first described the stepwise behavior of the epidemic diffusion in opportunistic network, demonstrating that it depends on the dynamic clustering of nodes. We also offered an analytical explanation of this behavior. To obtain efficient offloading in such a context, we proposed and evaluated DROiD, an offloading strategy that adapts to the varying opportunistic dissemination evolution to improve the distribution of popular contents throughout a mobile network. By leveraging on opportunistic communications between mobile nodes, DROiD relieves the congestion of the infrastructure network. The system tracks the evolution of content diffusion through user-sent acknowledgments and, thanks to its smart re-injection algorithm, guarantees better offloading performance than other offloading strategies. DROiD's enhanced delivery system takes into account not only the actual infection value, but also its trend. DROiD perceives when the evolution of the content diffusion stagnates, and reacts in advance with respect to traditional strategies that considers only the actual infection rate. We confirmed through simulations that the proposed strategy consistently does better than existing offloading systems, performing very close to an oracle that has the real-time picture of the ad hoc connectivity of the entire network.

Future work is manifold. First, we want to push the characterization of the epidemic diffusion further, especially in real scenarios. We also intend to investigate an analytical model that predicts the impact of intermittent connectivity on the dynamic formation and dissolution of clusters. Finally, as an ongoing work, we are defining all the protocols involved in DROiD's process in order to experiment it in a real scenario.

Acknowledgment

Filippo Rebecchi and Marcelo Dias de Amorim carried out part of the work at LINCS (http://www.lincs.fr). This work is partially supported by the European Commission in the framework of the FP7 Mobile Opportunistic Traffic Offloading (MOTO) under grant agreement number 317959.

References

[1] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update (2012 – 2017)," 2013.

[2] B. Han, P. Hui, V. S. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan, "Mobile data offloading through opportunistic communications and social participation," *IEEE Transactions on Mobile Computing*, vol. 11, no. 5, pp. 821–834, May 2012.

[3] M. V. Barbera, J. Stefa, A. C. Viana, M. D. de Amorim, and M. Boc, "VIP delegation: Enabling vips to offload data in wireless social mobile networks," in *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, Barcelona, Spain, Mar. 2011.

[4] Y. Li, G. Su, P. Hui, D. Jin, L. Su, and L. Zeng, "Multiple mobile data offloading through delay tolerant networks," in *ACM CHANTS*, Las Vegas, NV, USA, Sep. 2011.

[5] X. Zhuo, W. Gao, G. Cao, and Y. Dai, "Win-Coupon: An incentive framework for 3G traffic offloading," in *IEEE International Conference on Network Protocols (ICNP)*, Vancouver, Canada, Oct. 2011.

[6] J. Whitbeck, Y. Lopez, J. Leguay, V. Conan, and M. D. de Amorim, "Push-and-track: Saving infrastructure bandwidth through opportunistic forwarding," *Pervasive and Mobile Computing*, vol. 8, no. 5, pp. 682–697, Oct. 2012.

[7] P.-U. Tournoux, J. Leguay, F. Benbadis, J. Whitbeck, V. Conan, and M. D. de Amorim, "Density-aware routing in highly dynamic DTNs: The RollerNet case," *IEEE Transactions on Mobile Computing*, vol. 10, no. 12, pp. 1755–1768, Dec. 2011.

[8] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," *Computer Networks*, vol. 51, no. 10, pp. 2867–2891, Jul. 2007.

[9] I. Carreras, D. Miorandi, and I. Chlamtac, "A framework for opportunistic forwarding in disconnected networks," in *ICST International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)*, San Jose, CA, USA, Jul. 2006.

[10] A. Picu, T. Spyropoulos, and T. Hossmann, "An analysis of the information spreading delay in heterogeneous mobility DTNs," in *IEEE WoWMoM*, San Francisco, CA, USA, Jun. 2012.

[11] P. Bremaud, *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. Springer, 1999.

[12] G. W. Flake, R. E. Tarjan, and K. Tsioutsiouliklis, "Graph clustering and minimum cut trees," *Internet Mathematics*, vol. 1, no. 4, pp. 385–408, 2004.

[13] M. Karaliopoulos, "Assessing the vulnerability of DTN data relaying schemes to node selfishness," *IEEE Communications Letters*, vol. 13, no. 12, pp. 923–925, Dec. 2009.

[14] "FP7 European project iTETRIS: An integrated wireless and traffic platform for real-time road traffic management solutions," http://www.ict-itetris.eu.

[15] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO – Simulation of Urban MObility: An overview," in *International Conference on Advances in System Simulation*, Barcelona, Spain, Oct. 2011.

[16] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *ICST International Conference on Simulation Tools and Techniques (SIMUTools)*, Rome, Italy, Mar. 2011.

[17] NS-3, "Network simulator," http://www.nsnam.org.

[18] J. Whitbeck, M. D. de Amorim, V. Conan, and J.-L. Guillaume, "Temporal reachability graphs," in *ACM Mobicom*, Istanbul, Turkey, Aug. 2012, pp. 377–388.

[19] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting mobile 3G using WiFi," in *ACM Mobisys*, San Francisco, CA, USA, Jun. 2010.

[20] O. B. Yetim and M. Martonosi, "Adaptive usage of cellular and WiFi bandwidth: An optimal scheduling formulation," in *ACM CHANTS*, Istanbul, Turkey, Aug. 2012.

[21] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: How much can WiFi deliver?" *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 536–550, Apr. 2013.

[22] S. Dimatteo, P. Hui, B. Han, and V. O. K. Li, "Cellular traffic offloading through WiFi networks," in *IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, Valencia, Spain, Oct. 2011.

[23] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci, "Taming the mobile data deluge with drop zones," *IEEE/ACM Transactions on Networking*, vol. 20, no. 4, pp. 1010–1023, Aug. 2012.

[24] 3GPP, "3GPP TSG SA: Feasibility study for proximity services (ProSe) (release 12)," 2012, http://www.3gpp.org/ftp/specs/html-INFO/22803.htm.

[25] S. Ioannidis, A. Chaintreau, and L. Massoulié, "Optimal and scalable distribution of content updates over a mobile social network," in *IEEE Infocom*, Rio de Janeiro, Brazil, Apr. 2009.