Offloading through Opportunistic Networks with Dynamic Content Requests

Raffaele Bruno, Antonino Masaracchia, and Andrea Passarella
IIT-CNR
Via G. Moruzzi 1, 56124 Pisa, Italy
{first.last}@iit.cnr.it

***Abstract*—Offloading is gaining momentum as a technique to overcome the cellular capacity crunch due to the surge of mobile data traffic demand. Multiple offloading techniques are currently under investigation, from modifications inside the cellular network architecture, to integration of multiple wireless broadband infrastructures, to exploiting direct communications between mobile devices. In this paper we focus on the latter type of offloading, and specifically on offloading through opportunistic networks. As opposed to most of the literature looking at this type of offloading, in this paper we consider the case where requests for content are *non-synchronised*, i.e. users request content at random points in time. We support this scenario through a very simple offloading scheme, whereby no epidemic dissemination occurs in the opportunistic network. Thus our scheme is minimally invasive for users' mobile devices, as it uses only minimally their resources. Then, we provide an analysis on the efficiency of our offloading mechanism (in terms of percentage of offloaded traffic) in representative vehicular settings, where content needs to be delivered to (subsets of the) users in specific geographical areas. Depending on various parameters, we show that a simple and resource-savvy offloading scheme can nevertheless offload a very large fraction of the traffic (up to more than 90%, and always more than 20%). We also highlight configurations where such a technique is less effective, and therefore a more aggressive use of mobile nodes resources would be needed.**

I. Introduction

In the last few years, we have observed a drastic surge of data traffic demand from mobile personal devices (smartphones and tablets) over cellular networks [1]. This has already generated famous collapses of 3G networks in the recent past, (e.g. [2]), showing that standard cellular technologies may not be enough to cope with this data demand. Even though significant improvement in cellular bandwidth provisioning are expected through LTE-Advanced systems, the overall situation is not expected to change significantly [3]. Besides personal mobile devices, the diffusion of M2M and IoT devices is expected to increase at an exponential pace (the share of M2M devices is predicted to increase 5x by 2018 [1]), which is likely to generate a corresponding increase in the demand for mobile traffic (11-fold increase by 2018 [1]).

Offloading part of the traffic from the cellular to another, complementary, network, is currently considered one of the most promising approaches to cope with this problem, with offloaded traffic being foreseen to account for at least 50% of the overall traffic in the coming years [1]. Among the various forms of offloading that are currently investigated (described in more detail in Section II), in this paper we consider offloading through opportunistic networks. Opportunistic networks [4] exploit physical proximity between mobile nodes to enable direct communication between them. They typically exploit ad hoc enabling technologies like WiFi-direct or Bluetooth, and support dissemination of messages through multi-hop space-time paths, i.e., multi-hop paths that develop both over space - as in conventional ad hoc multi-hop networks - and over time - by exploiting contact opportunities between nodes that become available over time due to their mobility. In offloading schemes based on opportunistic networks (e.g., [5], [6], [7]), content is initially seeded on a subset of the mobile nodes, and then it spreads through the opportunistic network to reach all interested users. Cellular bandwidth is thus used only to seed the network, possibly to add additional seeds over time (as in [6]), and to send content directly to interested users upon expiration of the deadline for its delivery. These schemes permit to significantly offload the cellular network, while at the same time guaranteeing bounded delays in content delivery.

Most of the literature on opportunistic-based offloading investigates the scenario where a specific piece of content is generated, and the set of users to whom it has to be delivered is known already at that time and does not change subsequently, i.e. requests are *synchronised*. While significant, this scenario only partially captures relevant use cases. In particular, it does not cover cases where content demand is *dynamic*, i.e. users' requests for the same piece of content can arrive at different time instants. In the latter scenario offloading can still be applied: upon a request, content can reach the requesting user either through the opportunistic network, exploiting an ongoing dissemination process, or through the cellular network, in case the opportunistic dissemination does not reach the user in time. Offloading may even be more needed in case of dynamic requests, as synchronised requests could in principle be served also through multicast transmissions (although [8] shows that offloading is beneficial also when multicast is applied).

In this paper we start investigating dynamic content requests, with a particular focus on vehicular scenarios. We deliberately use a very simple offloading scheme, described in Section III, whereby resources provided by mobile nodes are minimally used. Nodes interested in a content store it for a limited amount of time after receiving it. New requests from other users are satisfied either when the requesting user encounters another user storing a copy of the content, or through the cellular network upon expiration of the delivery deadline.

As opposed to most of the literature looking at offloading through opportunistic networks, in our scheme we do not use any epidemic dissemination mechanism. On the one hand, this allows us to test a minimally invasive offloading scheme from the mobile users' perspective. As additional resources spent by mobile devices are sometimes considered a possible roadblock for offloading, our results show the offloading efficiency when this additional burden is extremely low. On the other hand, this simple scheme allows us to stress the efficiency of offloading in a particularly unfavourable configuration, thus providing a worst-case analysis, all other conditions being equal.

We focus on two complementary scenarios. In the first one, users move in a given physical area, and *all* request a piece of content, though at different points in time. This scenario is representative of users moving inside a limited area, and accessing very popular content, though not particularly time critical (i.e., content that does not generate a surge of requests immediately when it is generated). In the second scenario, users enter and exit (after a short amount of time) a given geographical area, and request content after a random amount of time after they entered the area. This complementary scenario is thus representative of users traversing a geographical area, as opposed to roaming there. Finally, in this scenario we also consider the case where content is requested only with a certain probability, i.e., when content has different levels of popularity.

We analyse the offloading efficiency in these scenarios, defined as the fraction of nodes receiving content through the opportunistic network. We characterise efficiency as a function of key parameters such as the number of users, the deadline of content requests, the time after which users drop the content after having received it, the popularity of the content. As we show in Section IV, even with an unfavourable opportunistic dissemination scheme, we find that offloading can be very efficient, as it is possible to offload up to more than 90% of the traffic. In other configurations, we find that the considered offloading scheme is less efficient, resulting in an offloading of only about 20%. In such cases, however, there is ample room for improvement, by further leveraging opportunistic networking resources, e.g., through more aggressive content replication schemes.

II. Related Work

Offloading can take several forms. In some cases, traffic is offloaded by using modifications inside the cellular architecture (e.g. LIPA/SIPTO [9] or small cells [10]), or other wireless access infrastructures, primarily WiFi [11], [12].

In this paper we consider offloading that exploits direct communications between mobile devices. Also in this case there are several approaches. In the 3GPP area, the device-to-device (D2D) [13] architectural modification to LTE has been defined, that devotes part of the cellular resources to direct communication between devices under strict control of a common eNB. Instead, we focus on using opportunistic networks together with cellular networks, as previously proposed, e.g. in [5], [6], [7], [8]. In this case, offloading exploits technologies (such as WiFi direct or Bluetooth) that do not interfere with cellular transmissions, and therefore no coordination is required with the eNB. In addition, mobile devices run self-organising networking algorithm to disseminate offloaded content without strict control of the eNBs or any other central controller.

The most common scenario where opportunistic offloading is used is content dissemination to a set of interested users. In most cases, it is assumed that the set of users interested in receiving a piece of content is known when the content is generated (or, alternatively, the content is implicitly requested by all interested users immediately when it is generated) and do not change over time. In addition, content is "seeded" through the cellular network on a subset of interested users, and then a dissemination process starts in the opportunistic network in order to reach the rest of the users [5]. Typically, epidemic dissemination is assumed [14]. In addition, some other papers (e.g. [6], [7]) consider that content must be delivered to users within a given deadline. To meet this deadline, content can be sent through the cellular network to additional seeds during the dissemination process, and is finally sent to users that are still missing it when the deadline is about to expire ("panic zone"). To know which users have received the content, a lightweight control channel is implemented through the cellular network, whereby users send an ACK to a central controller that tracks the status of the dissemination process, and determines when to seed additional copies of the content, and when to directly deliver content to the users in the panic zone.

With respect to this body of work, this paper differs in two main aspects. On the one hand, we release the assumption that users interested in a content request it simultaneously. In our scenarios content requests occur over time dynamically. On the other hand, we do not assume epidemic dissemination of content, but consider that content is exchanged in the opportunistic network only between users that have requested it, when they encounter directly. Therefore, our scenario covers more general cases with respect to strictly synchronised requests, and, in addition, provides a worst-case analysis of the potential of offloading, as we use the least possible aggressive form of dissemination in the opportunistic network.

To the best of our knowledge, the only other paper where content requests are not synchronised is [15]. That paper assumes that users become interested in the content after a random amount of time after its generation, and the goal of the proposed system is to maximise the probability that the user have already the content by then. This is very different from our scheme, which works reactively, *after* users generate requests.

Finally, offloading has been also proposed specifically in vehicular environments. In this case offloading schemes often assume the presence of RoadSide Units (RSU) [16] to support the dissemination process (e.g., by pre-fetching popular contents), which we do not assume here, to obtain a solution requiring no additional infrastructure development. Last but not least, offloading is proposed also for aggregating and uploading traffic generated by cars, e.g., in the context of Floating

Car Data (FCD) [17]. This is clearly a different application and offloading scenario with respect to that considered in this paper.

III. Offloading mechanisms

As anticipated in Section I we deliberately consider a simple scheme that uses very little resources of mobile nodes to support the offloading process. In general, we support scenarios where content is requested by users at random points in time. Similarly to [6], we assume the existence of a Central Dissemination Manager (CDM), that can communicate with all nodes through the cellular network and keeps track of the dissemination process. Without loss of generality[1], in the following we focus on the dissemination of a single piece of content to the set of interested users. The offloading mechanism is defined by the actions taken by requesting nodes and by the CDM, as described by Algorithms 1 and 2, respectively.

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

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

[1]Strictly, this is the case when congestion on the opportunistic network is low, and therefore the effect of multiple contents offloaded at the same time can be neglected. This is typically assumed in the literature on offloading through opportunistic networks.

[2]Note that the CDM implementation could be further simplified by allowing the nodes that select a content to send a message over the cellular network to inform the CDM. In this way, the CDM does not need to maintain separate timers for each of the nodes that have received the content. It is also reasonable to assume that such confirmation message would be a negligile overhead for the cellular network.

Algorithm 1 Actions taken by requesting nodes

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

Algorithm 2 Actions taken by CDM

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

22: Upon sharing_timeout for node k over
23: #nodes_with_C = #nodes_with_C-1
```

With respect to offloading mechanisms proposed for opportunistic networks (e.g., [5], [6]) our algorithms present several differences. First, there is no proactive seeding of the network. This is because requests arrive at the CDM dynamically, and there is no knowledge of which nodes will generate a request, and when. Therefore, we adopted a reactive policy, i.e. we wait for requests without doing any proactive seeding. Second, we want to use minimally mobile node resources in the opportunistic network. This is to make the offloading mechanism less intrusive as possible, as the additional mobile devices' resource usage brought about by offloading is often considered a possible severe drawback. Therefore, we do not use epidemic dissemination in the opportunistic network. For the same reasons, we assume that users drop content some time after receiving it. Still, our algorithms guarantee bounded delay, and impose similar overhead on the CDM as in previous proposals [6]. Clearly, Algorithms 1 and 2 can be easily modified to exploit additional resources of mobile devices (e.g., using more aggressive forms of dissemination or doing initial proactive seeding), if needed.

IV. Performance evaluation

A. Scenarios and performance indices

We test the performance of the proposed offloading schemes in two different vehicular scenarios, hereafter denoted as Scenario A and B.

In Scenario A we capture cases where a group of vehicles move inside a geographical area covered by a cell, and roam always inside that cell. Vehicles move on a stretch of road crossing the cell, and come back when arriving at the boundary. The resulting traffic is therefore bidirectional. Nodes move with a speed randomly selected (with uniform distribution) in an interval $[v_{min}, v_{max}]$, and can exchange content directly while being within a maximum transmission range T_{RX} from each other. We consider N nodes in the simulations, which all request the content. Requests are generated from the beginning of the simulation sequentially, according to a Poisson process with rate λ (i.e. two requests are spaced by an exponentially distributed time interval). Simulations lasts until all nodes have requested the content, and their *sharing timeouts* are all expired. In other words, we start from a condition where no nodes have any copy of the content, and we analyse the behaviour of the system until no copy of the content is available after all nodes have received it. While assuming vehicles go back and forth on a given road segment is a simplification, the scenario is still representative of movement patterns confined in a geographical area served by a cellular network, where a given content is very popular and thus requested by all users (though at different points in time). More in general, the scenario is representative of movement patterns whereby vehicles roam in such a geographical area, can move in opposite directions and can communicate with each other when being close enough, irrespective whether such movements occur on the same street or on different, nearby streets.

In Scenario B we capture cases where nodes are not necessarily staying in the same area, but there is a constant flux of vehicles entering and exiting the area. Again, we assume that vehicles move on a road and we focus on a road segment covered by a cell (we select speeds as in Scenario A). Traffic is again bidirectional, and we keep the number of nodes constant, and assume that a new vehicle enters the area when another one has left. When entering the area, vehicles become interested in the content with a given probability p. If they are interested, they generate a request after a time interval uniformly distributed between the time when they enter and the time when they reach the centre of the cell. Taking the same terminology of [6], we define a panic zone as the area of the cell Δ meters before the boundary. The *content timeout* is set so that the CDM sends the content directly when vehicles enter the panic zone. Finally, vehicles keep the content while being inside the cell. At the beginning of simulations, nodes are distributed randomly (with a uniform distribution) in the cell, are interested in content with probability p, and generate a request at a point in time uniformly distributed between the simulation start time and when they are midway towards the border of the cell. Simulations stop after 100 requests have been generated (50 in the case of low popularity content, without noticeable loss of statistical significance of the results), and the corresponding users have been all served. With this scenario we explore different cases with respect to Scenario A. After an initial transient phase, we are able to show a steady-state behaviour of offloading, in cases where vehicles enter and exit an area with a given flux and density. In other words, we can show how much offloading is efficient in making a given content "survive" in a geographical area, by only exploiting replicas available on vehicles of interested users passing through that area. This is an application of the basic floating content idea [18] to the case of vehicular networking environment in presence of offloading. In addition, only a fraction of the nodes can be interested into the content, i.e. the content can have different levels of popularity.

We ran simulations, using the NS3 with the LENA module for LTE[3], for various sets of parameters, as indicated in Table I. Specifically, we varied the number of nodes in both Scenarios, the request rate, the *content timeout* and the *sharing timeout* in Scenario A, and the content popularity in Scenario B. Request rate and popularity obtain similar effects in the two scenarios, as they modify the average number of nodes interested (and receiving) content at any point in time, and thus the density of nodes with a content replica. We performed at least 5 simulation runs for each set of parameters, using the independent replication method [19]. The main performance figure we consider is the offloading efficiency, defined as the fraction of content messages that reach the users through opportunisitc communications. For this index we computed the confidence intervals (with 95% confidence level) over the replications. To get a more precise idea on the dynamics of the offloading process over time, we also computed, on each 5s

[3]http://networks.cttc.es/mobile-networks/software-tools/lena/

time window, the average (across simulation replicas) number of copies of content stored on mobile nodes, and the average number of new content deliveries through the cellular and the opportunistic network, respectively.

TABLE I
SIMULATION PARAMETERS

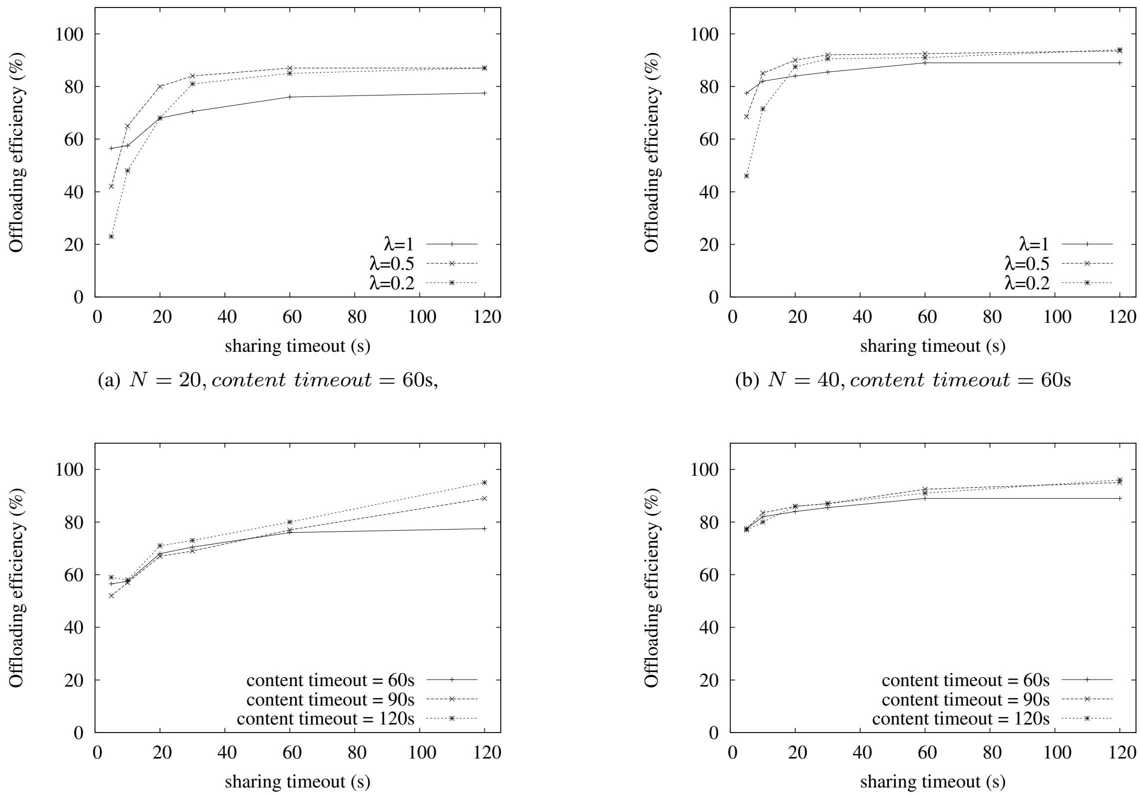
	Scenario A	Scenario B
speed (Km/h)	[80,120]	[90,110]
cell diameter (Km)	4	1
N (nodes)	20, 40	20, 40
T_{RX} (m)	200	50
p	1	0.5, 0.75, 1
λ (req/s)	1, 0.5, 0.2	–
content timeout (s)	60, 90, 120	–
sharing timeout (s)	5,10,20,30,60,120	–
Δ (m)	–	50

B. Analysis of scenario A

We start by analysing the system performance in Scenario A. To this end, Figure 1 shows the offloading efficiency obtained in a wide set of different network configurations, in which we vary the node density, the content request rate, as well as the *content timeouts* and the *sharing timeouts*. Several general observations can be drawn from the shown results. First, the offloading efficiency increases with the node density. The main reason is that the higher the node density, the higher the contact rate between the mobile devices. Thus, there are more opportunities for opportunistic dissemination between interested users. As far as the impact of the request rate (λ) we observe two regimes. When the *sharing timeout* is low, higher request rates result in higher offloading. This is intuitive, because higher request rates results in requests being more concentrated in time. When nodes share the content only for very short amounts of time (see for example the case of 5s), concentrating the requests in time increases the probability of encountering other nodes sharing the content. Less intuitive is the behaviour for large sharing timeouts, where higher request rates results in *lower* offloading efficiency. The reason of this will be more clear when analysing the evolution of dissemination over time (Figure 2). Intuitively, when requests are more concentrated in time, *content timeouts* for nodes that do not get the content via the opportunistic network are also more concentrated. As we will discuss later, when a timeout expires and content is delivered via the cellular network, this kicks off a fast increase in the dissemination of content via the opportunistic network in the region of the node whose *content timeout* has expired. When expirations are less concentrated in time (i.e., when request rates are lower), the opportunistic diffusion process has more time to spread content, and therefore the offloading efficiency increases.

A second interesting observation is related to the impact of the *sharing timeout* on the offloading efficiency. Our results indicate that if the content is sufficiently persistent in the network (e.g., *sharing timeout* $\geq$ *content timeout*) then the impact of the *sharing timeout* on the offloading efficiency is negligible. On the other hand, if the content is volatile, i.e., it is cached in the local memory of interested users only for few seconds, then the number of copies of that content in the environment may be too small to allow an efficient opportunistic dissemination. For instance, with 20 mobile devices and a content request rate of 0.2 req/s the offloading efficiency can be as low as 20% (this degradation of the offloading efficiency is less remarkable in denser networks). Interestingly, if the content request rate is high (i.e., $\lambda = 1$ req/s) then even a *sharing timeout* $= 5$s can still provide an offloading efficiency up to 60% in a cell with 20 mobile devices. A last observation is related to the effect of the *content timeout*. As shown in Figure 1c and Figure 1d an increase in the *content timeout* results into an increase of the offloading efficiency. This is more noticeable for large *sharing timeouts*, i.e. when content stays available on nodes for opportunistic dissemination longer. This is basically a joint effect of the fact that (i) content is available longer in the opportunistic networks (longer *sharing timeouts*) and (ii) interested nodes wait longer for requesting it via the cellular network (longer *content timeouts*).

To get a deeper understanding of the offloading dynamics, plots in Figure 2 show the temporal evolution of (i) the total number of mobile devices that have received the content via the cellular and the opportunistic network, respectively, and (ii) the number of copies of the content available in the network (i.e., the number of nodes that are storing and sharing a copy of the content at that time). Note that plots are typically shown until just after the time when the last requesting node has received the content. The system evolution after that time is not particularly interesting: nodes progressively drop the content when their *sharing timeouts* expire. We show plots for extreme values of the considered parameters. Specifically, in Figures 2a and 2b we focus on two extreme values of the *sharing timeout* parameter, for the case of 40 nodes and 1 req/s (*content timeout* is always 60s). As expected, the main difference is the number of copies of the content available in the network, which is much higher in Figure 2b, resulting in a higher offloading efficiency. It is very interesting to observe the behaviour of the system after 60s, i.e. when the *content timeouts* for the first nodes generating requests expire. For larger *sharing timeouts* (larger than the *content timeout*), before that time, only 1 node (the first one requesting the content) can receive the content via the cellular network, as it is clear from Figure 2b. This is due to the behaviour of the CDM explained in Section III, that sends immediately a content to the first requesting node (as no other node stores the content yet), and then waits the *content timeouts* (i.e. 60s) for the next requests before taking any action. In other words, it is impossible to have more than one delivery via the cellular network in the first 60s, due to the CDM algorithm. When the *sharing timeout* is short, more copies of the content can be sent via the cellular network also before the first *content timeout* expires. This happens whenever a new request is generated and all nodes that have previously received the content have already dropped it (due to expiration of the *sharing timeout*). In both cases, after 60s from the start of the simulation, *content timeouts* start expiring, and



(a) $N = 20, content\ timeout = 60$s,

(b) $N = 40, content\ timeout = 60$s

(c) $N = 20, \lambda = 1$,

(d) $N = 40, \lambda = 1s$

Fig. 1. Scenario A: offloading efficiency for varying request rates, $content\ timeouts$ and the $sharing\ timeouts$.

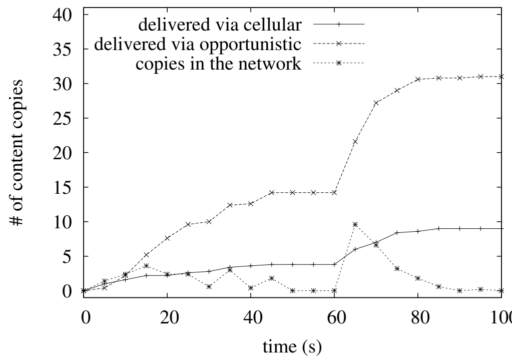
new copies of the content are sent via the cellular network. This generates a burst of dissemination in the opportunistic network, that is noticed by the steep increase of the curve related to opportunistic deliveries around that time. Note, in particular, that after 60s in both cases the rate of increase of the delivery via the opportunistic network is higher than the rate of increase of cellular deliveries. This means that each delivery via the cellular network is significantly amplified by deliveries in the opportunistic network.

Figures 2c and 2d show the evolution over time in the least favourable conditions for offloading, i.e. for low request rates ($\lambda = 0.2$) and very short *sharing timeout* (5s). Curves confirm the behaviour described before. In particular at low densities ($N = 20$) the *sharing timeout* is not long enough to sustain significant dissemination over the opportunistic network. The situation improves for denser networks ($N = 40$), but still the *sharing timeout* makes nodes drop content too fast with respect to the rate of arriving requests (anyway, the offloading efficiency is still between 20% and 40% even in these cases).

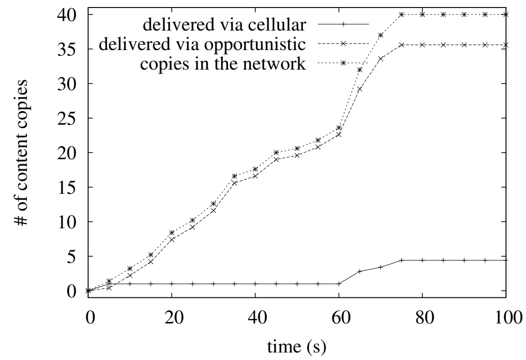
C. Analysis of scenario B

Figure 3 shows the offloading efficiency in Scenario B for the two considered densities of nodes and the different content popularities (p). Results basically confirm previous observations. This is nevertheless important, as Scenario B is more representative of a "steady state" behaviour of the offloading system, as nodes constantly enter and exit the cell at a given rate, and continuously generate requests (with a given probability). Again, denser networks ($N = 40$) achieve higher offloading efficiency. The effect of the popularity parameter is similar to that of the request rate in Scenario A: the higher the popularity, the higher the number of nodes sharing content, the higher the offloading efficiency. It is interesting to note, however, that, due to the mobility of the nodes, they stay within the cell only for about 30s in total, and, on average, stay in the cell for about 22s after having generated a request. This is the "useful time window" during which they can receive content via opportunistic dissemination. Even though this time window is rather short, offloading is very efficient, even at quite low popularities ($p = 0.2$).

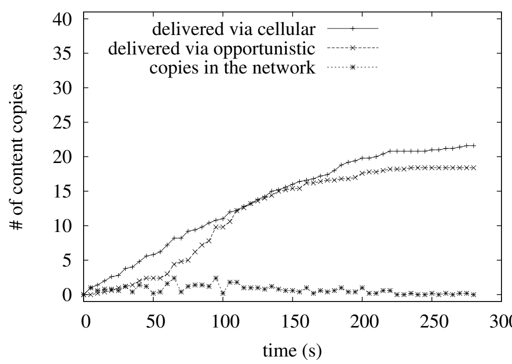
Finally, Figures 4a and 4b show the evolution over time for $N = 40$ nodes at the extreme popularity values. Besides confirming the general behaviour observed also in Scenario A, it is interesting to note that at high popularity the opportunistic dissemination alone is sufficient to keep enough copies of the content in the cell so that requesting nodes can find at least one before exiting. This is shown by the fact that the curve of delivery via the cellular network flattens out after an initial "seeding" interval. Instead, in case of less popular contents, there are cases where nodes do not encounter other nodes sharing a copy of the content before getting out of the



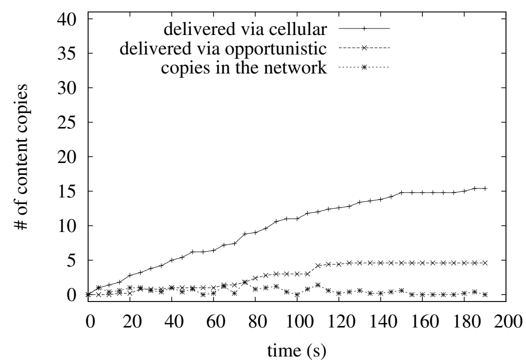
(a) $N = 40, \lambda = 1$ req/s, $sharing\ timeout = 5$s,



(b) $N = 40, \lambda = 1$ req/s, $sharing\ timeout = 120$s



(c) $N = 40, \lambda = 0.2$ req/s, $sharing\ timeout = 5$s,



(d) $N = 20, \lambda = 0.2$ req/s, $sharing\ timeout = 5$s

Fig. 2. Scenario A: temporal evolution of the number of content copies and served content requests in different network scenarios.

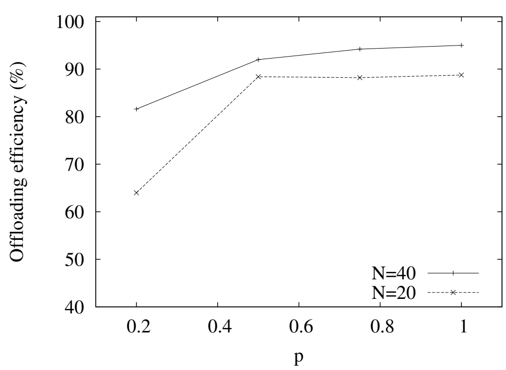
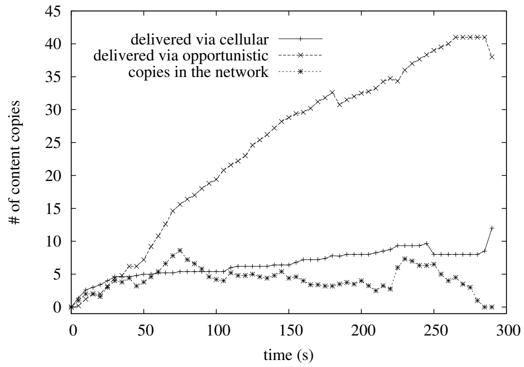


Fig. 3. Scenario B: offloading efficiency for different content popularities.

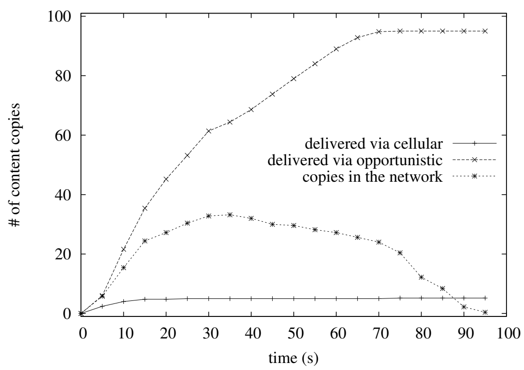
cell, and therefore the CDM needs to serve them through the cellular network. Fluctuations in the number of copies stored in the network are mainly due to statistical fluctuations in the contacts and requests events. In addition, the curves drop towards the end of the simulation when only few requests need to be satisfied and no new requests are generated (remember that simulations stop when a maximum number of requests is reached).

V. Conclusions

In this paper we have started to study the performance of offloading through opportunistic networks, in cases where content request are generated dynamically, and are not all synchronised at the moment when content becomes available. This general scenario is still to be satisfactorily addressed in the literature, and represents a large number of more specific scenarios. Interestingly, in such cases no support from cellular multicast mechanisms can be used, therefore offloading is even more critical. We have defined offloading mechanisms that guarantee bounded delays in content delivery, but, differently from existing literature, use as little as possible resources of mobile users' devices. This is also a critical point, as additional consumption of mobile devices' resources (storage, battery, etc.) is a drawback of offloading with opportunistic networks, that could limit its practical applicability. By considering minimal use of mobile devices' resources, we show that offloading can still be able to drastically reduce the traffic over the cellular network, also in a configuration that is unfavourable for its efficiency. Specifically, we tested the performance of our offloading schemes in vehicular environments, considering different densities of nodes, different popularity of content, and different parameters of the offloading protocols. Our results show that offloading can be very efficient also when using very limited resources of mobile devices, achieving



(a) low popularity ($p = 0.2$)



(b) high popularity ($p = 1$)

Fig. 4. Scenario B: temporal evolution of the number of content copies and served content requests in a network with $N = 40$ users.

offloading ratios up to more than 90%. Moreover, our results also highlight configurations of the protocols and parameters of the investigated scenarios where offloading is less efficient, and therefore would benefit from more aggressive policies, using additional resources of mobile devices. It is worth noticing, however, that offloading efficiency never drops below 20% in the considered cases, even though they may be quite challenging for the considered offloading mechanisms.

Acknowledgment

This work was partly funded by the EC under the EINS (FP7- FIRE 288021), MOTO (FP7 317959), and EIT ICT Labs MOSES (Business Plan 2014) projects.

References

[1] Cisco, "Cisco visual networking index: Global mobile data traffic forecast, update, 2013–2018," Cisco, Tech. Rep., http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html, 2013.

[2] J. Wortham, "Customers Angered as iPhones Overload AT&T," *NY Times*, 2009, http://www.nytimes.com/2009/09/03/technology/companies/03att.html?_r=0.

[3] "Growing data demands are trouble for verizon, lte capacity nearing limits." [Online]. Available: http://www.talkandroid.com/97125-growing-data-demands-are-trouble-for-verizon-lte-capacity-nearing-limits/

[4] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *Communications Magazine, IEEE*, vol. 44, no. 11, pp. 134–141, November 2006.

[5] M. V. Barbera, A. C. Viana, M. D. de Amorim, and J. Stefa, "Data offloading in social mobile networks through {VIP} delegation," *Ad Hoc Net.*, vol. 19, no. 0, pp. 92 – 110, 2014.

[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 Mob. Comp.*, vol. 8, no. 5, pp. 682 – 697, 2012.

[7] L. Valerio, R. Bruno, and A. Passarella, "Adaptive data offloading in opportunistic networks through an actor-critic learning method," in *ACM CHANTS*, 2014.

[8] F. Rebecchi, M. D. de Amorim, and V. Conan, "Flooding data in a cell: Is cellular multicast better than device-to-device communications?" in *ACM CHANTS*, 2014.

[9] K. Samdanis, T. Taleb, and S. Schmid, "Traffic offload enhancements for eutran," *Comm. Surveys Tutorials, IEEE*, vol. 14, no. 3, pp. 884–896, Third 2012.

[10] S. Singh and J. G. Andrews, "Joint resource partitioning and offloading in heterogeneous cellular networks," *IEEE TWC*, vol. 13, no. 2, pp. 888–901, February 2014.

[11] F. Mehmeti and T. Spyropoulos, "Performance analysis of on-the-spot mobile data offloading," in *IEEE GLOBECOM*, Dec 2013.

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

[13] F. Malandrino, C. E. Casetti, C. F. Chiasserini, and Z. Limani, "Fast resource scheduling in hetnets with d2d support," in *IEEE INFOCOM 2014*.

[14] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, Tech. Rep. CS-200006, 2006.

[15] X. Wang, M. Chen, Z. Han, D. Wu, and T. Kwon, "Toss: Traffic offloading by social network service-based opportunistic sharing in mobile social networks," in *IEEE INFOCOM*, 2014.

[16] F. Malandrino, C. Casetti, C. Chiasserini, and M. Fiore, "Content download in vehicular networks in presence of noisy mobility prediction," *IEEE TMC*, vol. 13, no. 5, pp. 1007–1021, May 2014.

[17] R. Stanica, M. Fiore, and F. Malandrino, "Offloading floating car data," in *IEEE WoWMoM*, June 2013, pp. 1–9.

[18] J. Ott, E. Hyytiä, P. Lassila, J. Kangasharju, and S. Santra, "Floating content for probabilistic information sharing," *Pervasive and Mobile Computing*, vol. 7, no. 6, pp. 671 – 689, 2011.

[19] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*. McGraw-Hill, 2000.