The Price of Fog: a Data-Driven Study on Caching Architectures in Vehicular Networks

The Price of Fog: a Data-Driven Study on Caching Architectures in Vehicular Networks

Francesco Malandrino, Carla Chiasserini
Politecnico di Torino, Italy

Scott Kirkpatrick
The Hebrew University of Jerusalem, Israel

ABSTRACT

Vehicular users are expected to consume large amounts of data, for both entertainment and navigation purposes. This will put a strain on cellular networks, which will be able to cope with such a load only if proper caching is in place; this in turn begs the question of which caching architecture is the best-suited to deal with vehicular content consumption. In this paper, we leverage a large-scale, crowd-sourced trace to (i) characterize the vehicular traffic demand, in terms of overall magnitude and content breakup; (ii) assess how different caching approaches perform against such a real-world load; (iii) study the effect of recommendation systems and local content items. We define a *price-of-fog* metric, expressing the additional caching capacity to deploy when moving from traditional, centralized caching architectures to a "fog computing" approach, where caches are closer to the network edge. We find that for location-specific items, such as the ones that vehicular users are most likely to request, such a price almost disappears. Vehicular networks thus make a strong case for the adoption of mobile-edge caching, as we are able to reap the benefit thereof – including a reduction in the distance travelled by data, within the core network – with little or none of the associated disadvantages.

1. INTRODUCTION

Back in 2010, the traffic demand of newly-introduced iPhones briefly disrupted some cellular networks [1]. It is uncertain whether such disruptions are likely to happen again; however, there is no doubt that *if* they do happen, vehicular users will be among the main culprits.

The reason for this trend is multifold. First, vehicles carry people, and people carry multiple, data-hungry mobile devices. Second, vehicles themselves are increasingly often equipped with entertainment devices, which only add to the problem. Third, vehicles themselves download navigation data, e.g., map updates: while this is a minor component of the overall traffic today, it is expected to increase by orders of magnitude with the introduction of self-driving vehicles, which will need much more detailed and more up-to-date map information.

To make things worse, virtually *all* such data demand will be served by cellular networks. Indeed, most offloading solutions target pedestrian users, because their position changes relatively slowly over time and because they are more likely to be covered by such networks as Wi-Fi.

Caching is a primary way in which cellular network operators plan to react to this demand surge. One of the most popular solutions is to move caches as close as possible to the users, in the context of an approach known as *fog computing* (a term created by Cisco [2]). It is expected that doing so will increase the cache hit ratio, while reducing its latency and the traffic within the cellular core network. On the negative side, it will require deploying multiple, smaller caches. Additional help is expected from recommendation systems, whose effect is to shape the demand concentrating it around the most popular content items. Intuitively, having fewer, popular items to serve will improve caching performance.

In this context, our paper targets three main questions.

Vehicular demand. What is the data demand generated by today's vehicular users? Which apps and services represent the most significant contributions thereto?

Caching architectures. Given a target hit ratio, what is the relationship between the caching architecture and the size of the caches we need to deploy? How does moving the caches from core-level switches to individual base stations impacts the total cache size, as well as the distance data must travel within the core network, and the load thereof? What changes if a recommendation system is in place?

Location-specific content. Content items consumed by future vehicular networks are expected to strongly depend on the location – augmented maps for self-driving vehicles being the most obvious example. How does the emergence of this kind of content impact caching?

We answer these questions using a set of real-world, large-scale measurement data, coming from users of the WeFi app [3]. Due to its crowd-sourced nature, our dataset includes data for: (i) multiple apps, including video (e.g., YouTube) and maps; (ii) multiple types of users, from pedestrian to vehicular ones; (iii) multiple network technologies, including 3G, LTE, and Wi-Fi; (iv) multiple network operators.

We describe our dataset, as well as the additional processing we need to perform in order to overcome its limitation, in Sec. 2. Then, in Sec. 3 we explain how we model caching and caching architectures in our vehicular scenario. Sec. 4 summarizes our numerical results and the insights we obtain from them. Finally, Sec. 6 concludes the paper and sketches future work directions.

2. INPUT DATA

We describe the WeFi dataset we have access to in Sec. 2.1. Then, in Sec. 2.2 we detail the processing steps we need, in order to extract further information that is not directly included therein. Finally, Sec. 2.3 explains how we complement the available information using other datasets and well-known information.

Table 1: The Los Angeles dataset

Metric	Value
Time of collection	Oct. 2015
Total traffic	35 TByte
Number of records	81 million
Unique users	64,386
Unique cell IDs	47,928
Mobile operators (number of cells)	AT&T (16,992) Sprint (2,764) T-Mobile (24,290) Verizon (3,882)

2.1 The WeFi dataset

Our data comes from the users of an app called WeFi [3]. The WeFi app provides its users with information on the safest and fastest Wi-Fi access points available at the user's location. At the same time (and with their consent), it collects information about the user's location, connectivity and activity. Wefi reports over seven million downloads of the app globally, and over three billion daily records. In this work, we use a dataset relative to the city of Los Angeles – a vehicle-dominated environment. Its main features are summarized in Tab. 1.

Each record contains the following information:

- day, hour (a coarse-grained timestamp);
- anonymized user identifier and GPS position;
- network operator, cell ID, cell technology and local area (LAC) the user is connected to (if any);
- Wi-Fi network (SSID) and access point (BSSID) the user is connected to (if any);
- active app and amount of downloaded/uploaded data.

If the location of the user or the networks she is connected to change within a one-hour period, multiple records are generated. Similarly, one record is generated for each app that is active during the same period. The fact that location changes trigger the creation of multiple records allows us to assess whether, and how much, each user moves during each one-hour period. As we will see in Sec. 2.2, this is instrumental in distinguishing between static and vehicular users. Combining this knowledge with network technology information allows us to ascertain which types of traffic cellular networks ought to worry about.

Fig. 2 shows the cell deployments of the four main operators present in our trace. We can see that all operators cover the whole geographical area we consider, but using radically different strategies. T-Mobile and, to a lesser extent, AT&T, deploy a large number of cells, each covering a comparatively small area. Sprint and, especially, Verizon, follow the opposite approach: their networks are composed of relatively few cells, each covering a fairly large area.

This fundamental difference reflects on the topologies of each operator's core networks, and potentially on the effectiveness of different caching architectures. It is worth to stress that using a real-world, crowd-sourced trace such as ours, we are able to properly account for these factors, which are typically neglected by more abstract models.

2.2 Further data processing steps

From the WeFi dataset we easily identify several types of users and the content that they consume.

Table 2: Content categories

Category	Description
YouTube	All class names pertaining to YouTube
OnDemand	On-demand video services such as Netflix, Time Warner, and ShowTime
RealTime	Real-time streaming, e.g., Periscope and DirectTV
Players	Player apps such as VLC and HTC Video
Weather	Most notably Weather.com
Maps	Most notably Google Maps
News	Including CNN and NBC
Sports	NFL, Fox Sports and the like

User type. The WeFi app can be installed on a variety of mobile devices. The users carrying them can be static (e.g., sitting in a cafè), pedestrian (e.g., walking or jogging), or vehicular. We discriminate among these cases by looking at the *distance* covered by each user during each one-hour period. Fig. 1(a) shows the distribution thereof: we have almost 40% of static users, which do not move at all, a large number of pedestrian users covering moderate distance, and some users covering larger ones.

In order to be conservative, we label as vehicular those users that cover a distance exceeding 5 km in any one-hour period[1]. As a sanity check, we plot in Fig. 1(b) the fraction of vehicular users as a function of time. We can observe the familiar morning and afternoon peak times, when the fraction of vehicular users increases.

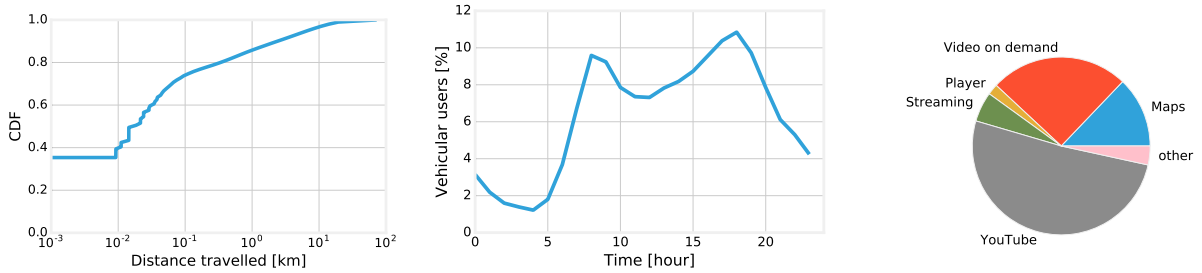
Content type. As recalled in Sec. 2, records contain an `app` field, containing the class name of the active application, e.g., `COM.GOOGLE.ANDROID.APPS.YOUTUBE.KIDS`. However, we cannot use this information directly, for two main reasons. First and foremost, different class names may correspond to the same app, e.g., both `COM.GOOGLE.ANDROID.APPS.YOUTUBE.KIDS` and `COM.GOOGLE.ANDROID.YOUTUBE` correspond to YouTube. Furthermore, we are not only interested in individual apps, rather in the *category* they belong to, as summarized in Tab. 2.

It is important to point out that different content categories lend themselves to caching to radically different extents. Caches are virtually useless for real-time streaming content (while LTE broadcasting [4] represents a more promising alternative). On-demand video content can be successfully cached, especially if popular. Sport and news content is even easier to cache, as there is a limited number if items that is likely to be requested (e.g., the highlights of yesterday's games). Finally, weather and map content is highly local, as users are very likely to need information about their current location.

The relative importance of the aforementioned categories is summarized in Fig. 1(c). YouTube and other on-demand content dominate the vehicular traffic, while real-time streaming represents much of the rest. This is good news from the caching viewpoint, as much of the vehicular traffic is represented by content that can be successfully cached.

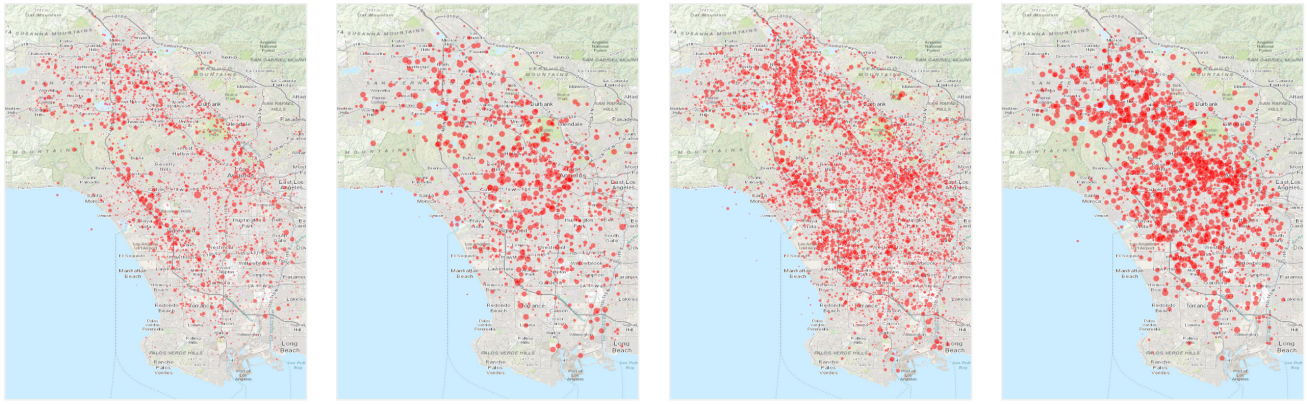
Finally, it is important to stress that over 70% of vehicular traffic in our dataset is served by cellular networks, compared to 11% of the global demand. This further confirms the importance of making cellular networks able to withstand an increase in the vehicular traffic, for which fewer offloading options are available.

[1]Notice that the same user can be vehicular in some time periods and static in others.



(a) (b) (c)

Figure 1: Distribution of the distance covered by users in the dataset (a); fraction of vehicular users as a function of time (b); most popular app categories among vehicular users (c).



(a) (b) (c) (d)

Figure 2: Deployment for AT&T (a), Sprint (b), T-Mobile (c), Verizon (d). Each dot represents a cell, and the size of dots is proportional to the coverage area thereof, as estimated from the location of users reporting the same cell ID.

2.3 Network topology and content demand

There are two types of information that are altogether missing in our WeFi dataset: network topology (both access and core), and content demand. In the following, we explain how we reconstruct this information using other existing datasets and/or common knowledge.

Network topology. In order to study the effectiveness of different caching architectures, we need information about how base stations are connected to each other. Sadly, such information is not only absent from the WeFi dataset, but virtually impossible to obtain for any network. Indeed, this is highly sensitive information for network operators. We estimate the position of base stations from the users' locations, as follows:

1. from each record, we extract the ID of the cell the user is connected to and her latitude/longitude coordinates;
2. the convex hull of these locations corresponds to the cell coverage area (notice that such areas can and do overlap);
3. we assume base stations sit at the baricenter of each convex hull.

As for the core network, we assume, as in [5], a tree topology where:

- base stations are grouped into *rings* of ten;
- rings are connected to aggregation-layer *pods*;
- pods are connected to *core*-level switches.

Finally, we assume completely separate network topologies for each operator.

Per-content item demand. Our dataset tells us how many users use, for example, YouTube, and how much data they consume. However, it contains no information about *which* of the countless YouTube videos they are watching, which is crucial to study the effectiveness of caching schemes. We cope with this limitation through different approaches, depending on the content category:

- RealTime, Players: each request refers to a different content ID, modeling the fact that no caching is possible;
- News, Sports: with probability 0.9, the content item is selected from 50 popular ones, otherwise, a new content ID is generated;
- Meteo, Maps: with probability 0.9 the item is selected from 10 location-specific ones, otherwise, a new content ID is generated;
- YouTube, OnDemand: the content ID is extracted from the YouTube measurement [6], with a probability that is proportional to the number of each video's views.

The above assignment policy reproduces the qualitative differences between content categories, and therefore the different ways each lends itself to caching. Also, note that content items belonging to different applications are always considered to be different.

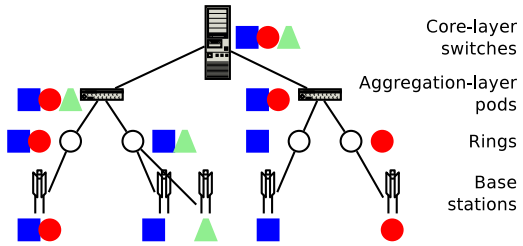


Figure 3: In this simplified network architecture, base stations are connected to aggregation pods and then to a core switch. Shapes correspond to cache-worthy content items; the total cache capacity is 6 if caches are deployed at the base stations or at the rings; decreases to 5 if caches are moved to aggregation pods, and to 3 if they are located at the core switch.

3. CACHING ARCHITECTURES

Our purpose is to evaluate not caching *policies*, i.e., how to choose the content to cache, rather cache *architectures*, i.e., at which level of the network topology caches should be deployed. Four options are possible:

- individual *base stations*: each base station has its own cache, bringing the fog-computing vision to its extreme;
- base station *rings*: caches are shared among the base stations (typically around ten) connected by the same ring, reducing the number of caches to deploy;
- *aggregation-layer pods*: they typically serve hundreds of base stations within a fairly wide area; this represents a more centralized caching architecture;
- *core-layer switches*: the most centralized caching architecture.

Given the user demand information, we consider a *target* hit ratio, and seek to determine the cache capacity needed to obtain such a ratio under different architectures. More precisely, we proceed as follows:

1. we keep track of the popularity (i.e., number of requests) of each content item within each cell;
2. we sort the item/cell pairs by decreasing popularity;
3. we mark as *cache-worthy* enough pairs to guarantee the target hit ratio, starting from the most popular ones;
4. we identify the location at which cache-worthy content items should be stored;
5. we add at most one copy of the cache-worthy content item at said location;
6. we evaluate the total cache size needed.

The network node at which content copies are stored (as per item 4 above) depends on the current caching architecture: if caches are deployed at base stations, then it is the base station itself; otherwise, it is the core network entity (ring, aggregation pod, core-layer switch) serving that base station.

Fig. 3 exemplifies the relationship between caching architecture and cache size. The closer caches are to base stations and end-users, the more likely we are to cache multiple copies of the same content item (at different locations), thus increasing the total cache size. On the other hand, caches that are closer to end-users tend to be smaller, which can result in significant cost reduction.

3.1 Performance metrics

Price-of-fog. We can formally define the price-of-fog metric as the ratio of the cache size to deploy under a given architecture to the cache size to deploy at the core switches. In the example case of Fig. 3, the price-of-fog is $\frac{5}{3} \approx 1.67$ when placing caches at aggregation pods, and $\frac{6}{3} = 2$ when placing them at base stations or at the rings. Clearly, content popularity distribution and content locality have a major impact on the price-of-fog.

Suppose that exactly the same set of content items were deemed cache-worthy at all base stations – perhaps as a consequence of an effective recommendation system. In the network of Fig. 3, the price-of-fog would raise as high as 4 – and much higher in real networks, where core nodes have more descendants. At the other extreme, if the set of cache-worthy content items at every base station were disjoint, the price-of-fog would drop to 1, the lowest possible value. Indeed, one of the main contributions of our paper is to assess to which of these extreme cases current *and* future vehicular networks are closer.

Distance travelled by data. Fog computing essentially means moving data closer to the users, thus reducing the load on the core network. We quantify this effect by measuring the physical distance between the network node at which content items are cached (e.g., aggregation-layer pods or core-layer switches) and the base station serving it.

3.2 Recommendation systems and local content

We study two factors that can alter the content demand and the distribution thereof: recommendation systems and the presence of location-specific content. The latter is expected to become a dominant factor in the near future, especially for vehicular applications.

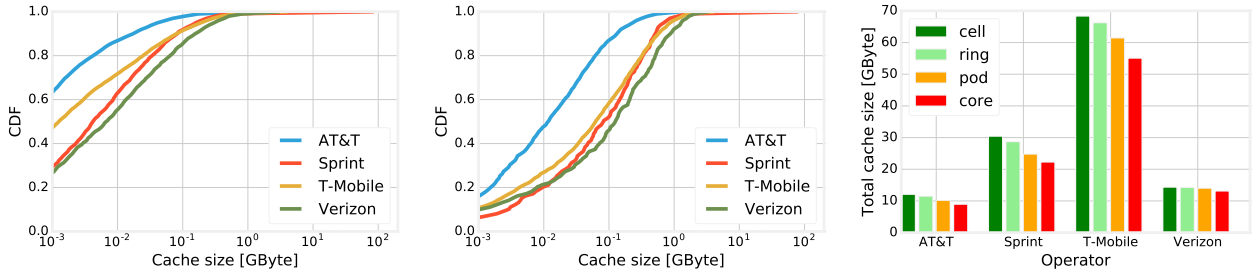
- Recommendation systems have the high-level effect of concentrating the demand towards the most popular items. To model this, we first track the top 5% most popular content items for each app; then, for each request, we switch the requested content to one of those popular ones with a probability p. The higher p, the stronger the bias towards popular content.
- In the case of location-specific content, we create 5 new content items specific to each cell; then, for each request, we switch the requested content to one of those local ones with a probability q. The higher q, the stronger the correlation between user location and content demand.

4. NUMERICAL RESULTS

A first aspect we are interested into is cache size. For each cache architecture, we are interested in (i) the distribution of cache sizes, and (ii) the total size thereof.

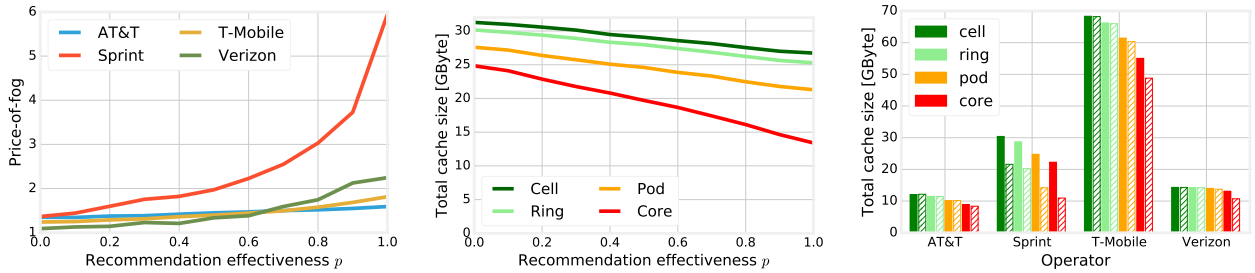
Comparing the distributions in Fig. 4(a) and Fig. 4(b) we can easily see that the closer caches are to end users, the smaller their size becomes – consistently with what one might intuitively expect. Interestingly, there are major differences between operators: as shown in Fig. 2, Verizon has fewer cells with larger coverage areas, therefore, it tends to deploy larger caches. T-Mobile, on the other hand, has many smaller cells, and therefore smaller caches.

Moving to the total cache size, shown in Fig. 4(c), highlights that both the total cache size *and* how it changes



(a) (b) (c)

Figure 4: Distribution of the cache size when they are deployed at base stations (a) and rings (b); total cache size for different architectures (c).



(a) (b) (c)

Figure 5: Recommendation system: price-of-fog (a); average cache size as a function of p (b); per-operator breakdown when $p = 0$ (solid bars) and $p = 0.5$ (bars with pattern) (c).

across caching architectures strongly depends on the operator and its network. T-Mobile, with their numerous small cells, has to deploy the most caches, followed by Verizon with their few bigger ones. The other operators follow intermediate approaches, and have smaller total cache sizes.

As for the price-of-fog metric defined in Sec. 3.1, it is actually quite modest, ranging between 1.15 for Verizon and 1.25 for AT&T. In other words, even considering the *current* demand of *current* networks, mobile operators (and their users) can reap the benefits of fog at the cost of a moderate increase in the total cache capacity they need to deploy.

Recommendation system. We now assume that there is a recommendation system in place, as described in Sec. 3.2, and study the effect of the p-value modeling its effectiveness. Somehow surprisingly, the price-of-fog depicted in Fig. 5(a) *increases* as p grows; in other words, an effective recommendation system makes the fog computing approach more onerous in terms of required caching capacity.

Recall, however, that the price-of-fog is a ratio between two size values. As we can see from Fig. 5(b), cache capacity *decreases* as p grows, for *all* caching architectures. However, the size of core-level caches decreases faster, hence the growing price-of-fog.

It is also interesting to point out that, as we can see from Fig. 5(c), both the decrease in cache size and the price-of-fog strongly depend on the operator and its network topology. As an example, T-Mobile reaps significant benefits when caches are deployed at the core level and negligible ones otherwise, while Verizon experiences a decrease in cache size under all architectures. This is again due to the differences in network deployments, shown in Fig. 2. Cells covering very small areas, such as in the case of T-Mobile, are unlikely to be a good location to place a cache.

Location-specific content. Let us now see the effect of location-specific content; recall that, as mentioned in Sec. 3.2, the q-value expresses how strong the correlation between location and content demand is. Comparing Fig. 6(a) to Fig. 5(a) above we can clearly see that the price-of-fog is (i) much lower, and (ii) virtually constant for all values of q. At a high level, this tells us that if demand and location are strongly correlated, then embracing a fog computing-style caching approach comes at virtually no penalty.

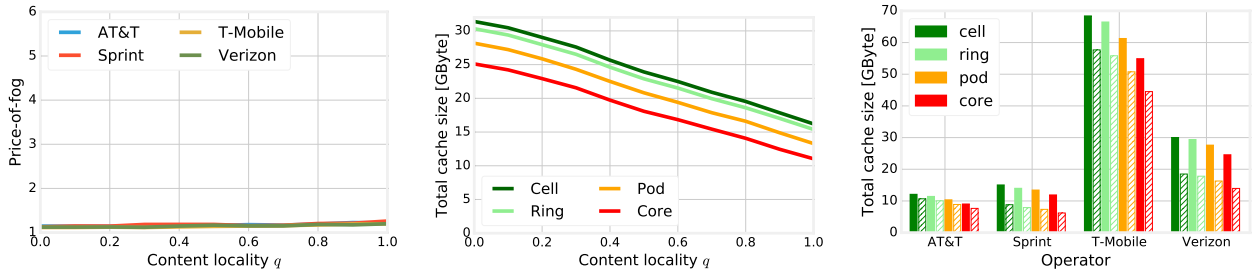
Consistently, Fig. 6(b) shows that cache sizes steadily decrease as q grows, for all caching architectures. Also notice, from Fig. 6(c), that the effect has roughly the same magnitude for all operators.

Last, Fig. 7 explores how caching architectures, recommendation systems and content locality influence the average distance travelled by data, as defined in Sec. 3.1. We can clearly see the benefit of fog computing, as to more decentralized architectures invariably correspond shorter distances. Furthermore, such a benefit strongly depends on the operator – and their deployment, as laid out in Fig. 2 –, and changes little if a recommendation system is in place or content is location-specific.

The reason for the latter is that we keep the target hit ratio fixed, and deploy the minimum amount of cache necessary to achieve it. In other words, we exploit recommendation systems and content locality to reduce the cache size (i.e., the price of the fog) rather than to enhance the benefits thereof (i.e., data travelling shorter distances).

5. RELATED WORK

Our paper falls in the general area of caching for mobile (specifically, cellular) networks. The most significant recent



(a) (b) (c)

Figure 6: Location-specific content: price-of-fog (a); average cache size as a function of q (b); per-operator breakdown when $q = 0$ (solid bars) and $q = 0.5$ (bars with pattern) (c).

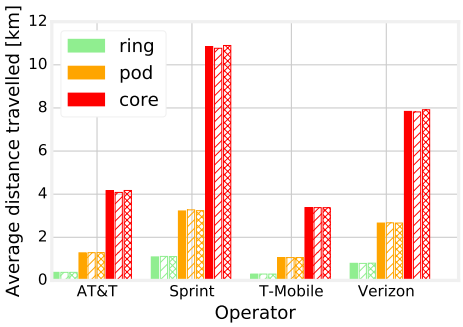


Figure 7: Distance travelled by data for different operators and cache architectures. Solid bars correspond to the default scenario, lines pattern to local content with $p = 0.5$, grid pattern to recommendation system with $p = 0.5$.

trend in this field is *fog computing*, also called *mobile edge computing*. Compared to traditional cloud computing, the emphasis is to move processing and caching capabilities as close to the access networks (and users) is possible, so as to (i) reduce the load on the core network, and (ii) provide more customized service.

A first body of works deal with the fundamental problem of *where* to locate the cached content items, given some degree of knowledge about user demand. For example, the authors of [7] exploit concepts from information-centric and content-centric networking to maximize the cache hit ratio, while [8] leverages mobility information for the same purpose. Other works [9] take a more holistic approach, moving both caches and virtual machines around the network as the load changes.

An especially relevant application of caching is video streaming. As an example, [10] accounts for layered video coding techniques, and addresses the problem of placing the right layers at the right cache. Interestingly, it also models the cases when multiple mobile operators cooperate to reduce each other's load. Other works [11, 12] aim at *foreseeing* the content demand, in order to proactively populate caches [11] or to serve users [12].

6. CONCLUSION AND CURRENT WORK

Traffic demand from vehicular users is set to rapidly grow in the next years, and cellular networks will bear most of the burden. In this context, we compared the most popular caching architectures from the viewpoint of the total cache size operators need to deploy to reach a target hit ratio.

Leveraging a real-world, large-scale, crowd-sourced dataset coming from the WeFi app, we found that fog computing approaches pair remarkably well with highly localized content, such as navigation information for future self-driving vehicles. On the other hand, more centralized caching approaches perform better along with traditional recommendation systems, that make globally-popular content more popular.

We are currently extending our work by including caching policies, e.g., least-recently-used, into the picture. This would allow us to more realistically model the interaction between caching policies and caching architectures.

7. REFERENCES

[1] B. S. Arnaud, "iPhone slowing down the Internet – Desperate need for 5G R&E networks," 2010.
[2] Cisco, "Transform Data into Action at the Network Edge," 2015.
[3] "Wefi," http://www.wefi.com.
[4] C. Borgiattino, C. Casetti, C.-F. Chiasserini, and F. Malandrino, "Efficient area formation for LTE broadcasting," in *IEEE SECON*, 2015.
[5] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, "Softcell: Scalable and flexible cellular core network architecture," in *ACM CoNEXT*, 2013.
[6] X. Cheng, C. Dale, and J. Liu, "Statistics and social network of YouTube videos," in *IEEE IWQoS*, 2008.
[7] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5g systems," *IEEE Comm. Mag.*, 2014.
[8] A. Mahmood, C. Casetti, C. Chiasserini, P. Giaccone, and J. Härri, "Mobility-aware edge caching for connected cars," *IEEE WONS*, 2016.
[9] F. Sardis, G. Mapp, J. Loo, and M. Aiash, "Dynamic edge-caching for mobile users: Minimising inter-as traffic by moving cloud services and vms," in *IEEE WAINA*, 2014.
[10] K. Poularakis, G. Iosifidis, A. Argyriou, I. Koutsopoulos, and L. Tassiulas, "Caching and operator cooperation policies for layered video content delivery," in *IEEE INFOCOM*, 2016.
[11] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Comm. Mag.*, 2014.
[12] F. Malandrino, M. Kurant, A. Markopoulou, C. Westphal, and U. C. Kozat, "Proactive seeding for information cascades in cellular networks," in *IEEE INFOCOM*, 2012.