A Novel Paradigm for Context-aware Content Pre-fetching in Mobile Networks

Pavan Kamaraju*†, Pietro Lungaro†, Zary Segall†
*Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, USA
†Mobile Service Lab, Royal Institute of Technology (KTH), Kista, Sweden
Email: pavan4@umbc.edu, pietro@kth.se, segall@kth.se

***Abstract*—The current content provision methods and associated pricing and business models are challenged by the traffic requirements anticipated for future "data intensive" services. In order to deliver substantially higher peak rates operators will need to deploy a much denser infrastructure and/or acquire more spectrum, thus significantly increasing their CAPEX and OPEX and reducing revenues. To improve the utilization of available network resources this paper presents ActiveCast, a disruptive content delivery paradigm that supports opportunistic content pre-fetching by introducing semantic and context awareness in the currently "agnostic" networking paradigm. The experimental investigations presented in the paper focus on mobile video provision and a content provider, integrated with Facebook and YouTube, has been developed and used to identify socially relevant content for a set of test users. Part of the studies presented in the paper aim at experimentally understanding the structure of the energy costs associated with pre-fetching and on defining a delivery strategy that allows controlling the amount of energy invested. A comparison between a centralized implementation, in which pre-fetching is coordinated by the mobile operators, and an Over-The-Top (OTT) implementation of ActiveCast are also presented. The results show that complementing the context information available at individual user terminals with traffic information, shared by mobile operators through the ActiveCast API, can substantially reduce the energy costs of content delivery, as compared with "on demand" video streaming. Additionally, opportunistically exploiting connections with WiFi APs can amplify the gains already achievable by pre-fetching on wide area networks.**

I. Introduction

Recent years have seen an unprecedented growth in the adoption of mobile data services. This can be attributed to both the enormous and rapid technological development seen on the user equipment side, i.e. the post iPhone growth of offering for smartphones, and the pricing strategy, based on flat-rate subscription, adopted by most of mobile operators to incentivize the usage of mobile Internet. The success of mobile data is also reflected in a drastic decrease of traffic shares for voice services, which no longer dominate the mobile usage. However, while voice and SMS are significantly more profitable than mobile Internet services, the current trends clearly indicate that operators are losing increasing shares of their revenues due to users' adoption of substitute services, which are provided through mobile apps and therefore included in their mobile Internet subscriptions. Moreover, about 50% of the current traffic on user devices is constituted by video streaming services and this is expected to grow up to 70% of the total traffic by 2016 [1]. The success of video content depends on the widespread adoption of innovative services capitalizing on user-generated content, i.e. YouTube, on the new avenues for "social" sharing of content, e.g. Facebook and Twitter, and on the adoption of "on demand" content distribution by the major content providers, e.g. newscasting, TV programs and films. On the other hand, since the video resolution on users' devices is rapidly increasing, the average size of the videos to be served to match such resolution is becoming substantially larger.

In order to deliver increased user experience mobile operators have been investing significantly in upgrading their network infrastructure, with the goal of increasing the peak downlink rates to users' devices. This has been translated in deploying additional Base Stations (BSs) to achieve a denser infrastructure, in bidding for additional spectrum, in adopting new and more efficient radio technologies, e.g. LTE. One critical problem the operators face is that, with increased file sizes (e.g. HD videos) and service requirements, the traffic currently served by the network tends to be very "bursty", thus, a "classical" network dimensioning approach, which targets "peak hour" loads, is likely to lead to over-provisioning of resources, with most of the cells experiencing a significant excess of capacity during large portions of the day. Following nowadays trends, future data intensive services are likely to pose an even greater threat to the current service provision paradigm and associated business model. In fact, one of the keys to the explosion of mobile Internet is flat rate pricing, which on the other hand decouples the revenue intakes of operators from the amount of traffic circulating in their network. An important observation is that the current content provision and business models do not scale well with increased future traffic requirements, since to serve the expectedly higher datarate requirements of future services, operator will need to either deploy more BSs or acquire more spectrum, which in turn will significantly increase CAPEX and OPEX and reduce revenues. Current proposals are perceived as rather myopic, since they are essentially aiming either at changing the current pricing strategy, e.g. introducing service prices proportional to the transferred bits, or limiting the access to some service, e.g. data intensive services or services which can substitute high margin services like voice or SMS.

A. Context-aware Content Delivery

An alternative approach would consist of transforming the currently agnostic content delivery paradigm (not aware of both content and service semantics and users', networks' and terminals' contexts) into semantic- and context-aware. In this respect, this paper proposes a novel content delivery paradigm called ActiveCast, designed to support a range of embodiments of opportunistic content pre-fetching over mobile networks.

Content pre-fetching has the potential to deliver high levels of Quality of Experience (QoE) at low network deployment costs, since it significantly improves the utilization of the available radio infrastructure (e.g see [2], [3], [4]). By decoupling the times

978-1-4673-5939-9/13/$31.00 ©2013 IEEE

in which content is delivered to the terminals from the ones in which the content is accessed by the end users, varying network conditions and different radio interfaces can be opportunistically exploited for optimizing various aspects associated with the content delivery. For example, (parts of) content likely to be accessed by the end users in the near future can be delivered to users terminals while these are within coverage of cells exposed to low traffic loads, or while the terminals are within range of high speed access points. Some of the current proposals (e.g. [5] and [6]) are precisely targeting systems for content pre-fetching which are capable of performing opportunistic content delivery on various available networks.

An important requirement for achieving the aforementioned gains is the capability of correctly predicting which content is more likely to be requested and accessed by the end users. If future user demand can be accurately anticipated, content can be opportunistically delivered at datarates that are likely to be higher than with the "on-demand" provision, thus lowering the average energy costs associated with content retrieval and delivering "instantaneous gratification" to the end users, e.g. no interruptions during a video playback. Moreover, by exploiting locations and times in which the networks experience excess of resources, this approach can potentially reduce the traffic loads to be served during peak hours. However, if the content prediction is incorrect, content pre-fetching might lead to a significant increase in energy expenditures at both network and terminal sides. Given the possibility of estimating future content requests, another aspect which has great importance for effectively performing content pre-fetching is the possibility of gathering network and terminal context information and use it to perform opportunistic content retrieval decisions. Schulman et al., used signal strength as a context parameter to design a pre-fetching system and have shown energy savings up to 60% [7]. This means that a "context manager" is required to monitor the network performances that are achieved while performing content delivery operations, to monitor terminal context, e.g. battery levels and/or screen resolution, to optimize the retrieved video quality and the overall energy costs, to gather information about alternative networks (e.g. WiFi APs or LTE BSs) that are available to serve the content while meeting different service requirements and performance profiles.

II. Problem statement

Since content prediction is one of the key aspects of content pre-fetching, in this paper we propose, implement and experimentally analyze a radically novel approach, in which content prediction is "outsourced" to the content providers. In recent literature [5] content pre-fetching has been typically considered performed by an entity, located either at the terminal side or in a server location, which monitors and records user traffic and uses the collected information to predict future user requests. While this approach is likely to create significant privacy concerns and therefore limited user adoption, it puts also significantly increased requirements, in terms of data gathering and processing, on the content delivery system. However, individual content providers are already using models of users' content requests (e.g. collaborative filtering) and have already available significant amount of information, concerning both individual users and overall user population, which could be effectively used for pre-fetching purposes. In the approach considered in our work, the content providers can submit "pointers" to data objects likely to be accessed by specific users using a novel ActiveCast API. This has been implemented, for the Android platform, to perform real life testing of the proposed context-aware content delivery schemes. The experiments illustrated in this paper are designed to bring an understanding of the potential benefits of this technology for users, content providers and mobile operators, with particular focus on video services.

One of the goals of this paper is to quantify the potential energy savings that can be achieved, at the terminal side, by adopting the proposed ActiveCast API for supporting content pre-fetching. The experimental studies have been comparing the current video streaming approach with different context-aware video pre-fetching strategies. Two alternative embodiments of content pre-fetching are considered: one in which pre-fetching is performed Over-The-Top (OTT), i.e. without the active involvement of a mobile operator, and another one in which information from both operator and mobile terminals is used by the context manager to activate the pre-fetching of specific video items. While the performances of the OTT case are experimentally evaluated using a context manager application developed for Android phones, the performances of the latter embodiment are evaluated through emulation, using data only gathered through terminal measurements, since the authors of the paper do not have access to information available at the operator side. The main purpose of this comparison is to understand the potential performance losses, e.g. increased energy costs due to context gathering operations at the terminal side (probes), that could be introduced by the distributed OTT embodiment.

III. Energy Model

To estimate the energy costs associated with the activations of the wireless interfaces in the terminals, while performing either streaming or pre-fetching content delivery, the PowerTutor [8] application was used. PowerTutor implements a usage-based power model in which a given wireless interface is associated with a fixed cost per unit of time. The cost varies with the specific mode of activation for the selected interface. For example, the 3G radio interface is characterized by different energy cost coefficients, depending on whether the interface is on the IDLE, FACH or DCH mode. Similarly the WiFi interface has a low and a high power mode. By keeping track of the duration in time of the various modes, the energy model adopted by PowerTutor can provide an accurate estimate of the energy costs for completing the delivery of a specific data object.

The model is further completed by the inclusion of the "tail time" component to the energy budget (see [9] for more information). The tail time corresponds to the time spent in one of the higher power states, while no data is transferred to a user terminal, before its demotion to the idle state. The duration of the tail time depends on specific operator implementation.

In order to simplify the model, we approximated the different energy cost coefficients for the active modes (e.g. FACH and DCH) with a single coefficient c_a, extracted by fitting the experimental energy data, and representing the average cost per second of active data reception. This was done to create a mapping between the currently achievable datarate and the expected energy costs for completing the download of a specific data object at that rate, which is a critical part of our proposed pre-fetching strategy. Considering both the active transfer and tail time phases, Eqn. 1 summarizes the energy cost $E(B)$,

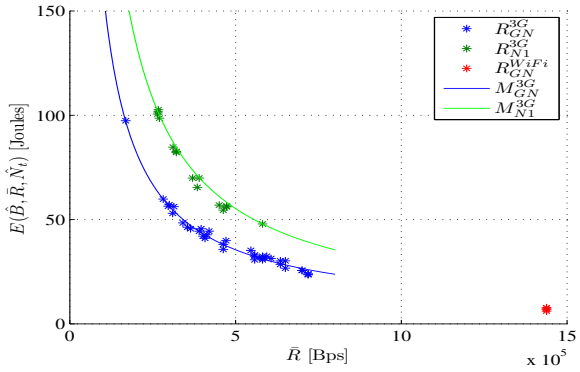

Fig. 1. Different realizations of energy costs $E(\hat{B}, \bar{R}, \hat{N}_t)$ for different phones and radio interfaces, all obtained for $\hat{B} = 27$MBytes and $\hat{N}_t = 1$. The realization points R_{GN}^{3G} and R_{GN}^{WiFi} correspond respectively to the Galaxy Nexus phone on 3G and WiFi interfaces while the points identified by R_{N1}^{3G} are obtained with the Nexus One on 3G. The curves obtained with the linear model of Eqn. 1 are also shown for the Nexus One (M_{N1}^{3G}) and the Galaxy Nexus (M_{GN}^{3G}) both on 3G. Note the accordance between the energy costs associated with the proposed model and the ones quantified with PowerTutor for the various realizations.

required for retrieving a data object of size B Bytes:

$$E(B, \bar{R}, N_t) = c_a * T_B + c_t * N_t(B) = \frac{c_a}{\bar{R}} * B + c_t * N_t(B), \quad (1)$$

where T_B represents the total time required to download the data object of size B, c_t the energy cost associated with the duration of a tail time, $N_t(B)$, the number of times in which the download has encountered a tail time during the retrieval of the object and $\bar{R}$ is the average datarate experienced during the active data transfer phases. In Fig. 1, different realization of energy estimations with PowerTutor and the corresponding $E(B)$ fitting curves are shown for different terminals and different wireless interfaces, for a data object of size $B = 27$MBytes. During each realization of data download both the average datarate and the number of tail time occurrences have been recorded, together with the total energy costs associated with the content delivery. Note that the energy profiles vary with different terminal types (e.g. the realizations R_{GN}^{3G} for the Galaxy Nexus and the R_{N1}^{3G} for the Nexus One phones) and different radio interfaces (e.g the realizations R_{GN}^{WiFi} for the Galaxy Nexus on WiFi). In practice, the specific energy profiles associated with the various radio interfaces available in a given terminal can be learned in time by a simple software agent recording and profiling the various energy costs for content delivery. Considering the realizations shown in Fig. 1, the cost coefficients for the Nexus One on 3G have been estimated to be $c_a = 0.9656$ Watts and $c_t = 2.5169$ Joules, while for the Galaxy Nexus on the same interface are $c_a = 0.5739$ Watts and $c_t = 4.16$ Joules. Due to very moderate traffic variations on the WiFi AP used for the experiments, all realizations concerning the WiFi interface are concentrated around 1.4 MBps. In general, due to significantly higher datarates achievable on WiFi, content pre-fetching using local networks has the potential of reducing content download energy costs of some order of magnitude as compared to both pre-fetching and "on demand" content delivery on 3G.

IV. Content delivery API

The purpose of the proposed content delivery API is to create an effective interface for sharing information, among the different content delivery stakeholders, to support and to optimize the content pre-fetching operations. In particular, the stakeholders considered in our approach include *content providers*, *end users* and their *terminals* and *mobile operators*. The exchange of information is also supported and facilitated by the presence of an *ActiveCast server* and an *ActiveCast software application* running on the end users' terminals.

A. Content identification

In the proposed model, various content providers are in communication, through the ActiveCast API, with the ActiveCast server. The providers are in charge of both identifying the data objects that are more likely to be of interest for specific users and associating with them an access probability value (e.g. mapped into a priority number between 1 and 5), which could be based on historic consumption behavior specific for the users, and/or based on the overall population (or a subset of) for that content provider. An additional piece of information, which might be relevant in some implementations, is represented by an "expiration date", indicating the latest time in which the content should reach the intended user terminals. The complete information concerning a specific data object, together with its URL and the ID of the intended recipient, can be communicated to the ActiveCast server using the ActiveCast API primitives, which effectively replace the *post* and *get* functionalities currently available at the content provider side. Among various functionalities already implemented, content providers can also track the status of their submitted content pre-fetching jobs, e.g. receiving notifications when the submitted content reaches the intended user terminals or when the users access it. For some data types, content providers can also submit an encrypted version of the content, which is then decrypted upon user access to the content stored in the terminal cache. In the current design, the transfer of decryption keys and billing operations can be performed directly between users and content providers, without direct involvement of the ActiveCast system.

B. Mapping access probability into pre-fetching context

Once the information concerning a new content pre-fetching job reaches the ActiveCast server, it is forwarded to the terminal of the intended recipient, which adds the new pre-fetching request to its queue of pending data objects. The ActiveCast terminal application contains also software modules for monitoring terminal and network contexts. This software has the capability of monitoring and recording a series of service variables, including cell ID, availability of WiFi access, signal quality with different mobile networks, battery level, user activity and to estimate currently achievable datarates via probing. Historical information concerning datarates achieved in specific locations and times of the day, together with information concerning the time of user access to specific services, durations of the sessions and amount of data consumed are also potentially available to the terminal context manager. One of the critical decisions to be performed by the ActiveCast terminal application is the mapping of the access probability, associated with a content item, into a target datarate and a possible association with a specific network interface to perform content retrieval.

An example of a mapping relationship between these quantities is described in the following equation:

$$\hat{R}(I_i) = R| \left\{ E(B, R, N_t) / E(B, R_{av}, N_t) \leq p_a(I_i) \right\}, \quad (2)$$

where $p_a(I_i)$ is the access probability associated to the item I_i and R_{av} is the average datarate achievable when items are consumed "on demand". The latter quantity can also be computed specifically for given item types (e.g. videos) and/or for content providers (e.g. YouTube videos or CNN videos). Eqn. 2 is here introduced to limit the maximum energy cost to be invested for pre-fetching a specific data item with a given access probability. For example, in order to break-even from an energetic perspective, if an item has 20% probability of being consumed by the end user the system should aim at pre-fetching it with an average datarate leading to 1/5 (or less) of the energy cost associated to its "on demand" consumption.

Depending on the specific implementation, the access probabilities, submitted by the content providers to describe the importance of their items for a given user, can be modified in the user terminal by an entity called "storage manager". This monitors the user's access to the content stored in the terminal cache and keeps tracks of the real access probabilities to different data objects submitted by the various content providers.

C. Context monitoring

In the considered approach, mobile operators can also be interfaced through the API to the ActiveCast servers and exchange updated information concerning traffic loads at specific BSs. This information can be used to support pre-fetching operations by complementing the network context information which is available in individual user terminals. In particular in the OTT case, i.e. without operator support, user terminals can only gather updated information on achievable datarates by initiating content pre-fetching while simultaneously monitoring the datarate performances within given time windows of a few seconds. This implies that in order to estimate if the currently achievable datarate matches the one required to meet the energy budget allocated for completing the pre-fetching of a given item (gathering network context information), user terminals need to perform a partial content download, which in many cases can introduce additional energy costs. If the target average datarate is not met, the content pre-fetching operations need to be interrupted, leading to energy inefficiencies introduced by both lower activation datarates and additional tail times. To compensate for the lower activation datarates, the ActiveCast terminal app can increase the target datarate for that item, proportionally to the amount of data still remaining to be downloaded. In general, the OTT approach might lead to suboptimal pre-fetching decisions. On the other hand, having access to updated traffic information estimates from mobile operators can be very valuable to reduce the amount of traffic and energy spent for gathering updated network context information.

V. Implementation

The content delivery system is implemented for Android Operating System on the mobile terminal and the server side runs with the help ruby on rails. We used a HP ProLiant DL380 G7 rack server to host the ActiveCast webapp. We used a rooted Nexus One and a stock Galaxy Nexus running Android 2.3.7 and 4.1.1 respectively. The overall architecture is shown in Fig.2, where three planes are highlighted. The user plane represents the terminal side of the content delivery system. There, the "Connection Analyzer" is an entity designed for gathering network and terminal context information and for managing the content retrieval operations. These include maintaining the queue of pending pre-fetching jobs and performing decisions on whether to start, pause or stop the downloads of specific data objects. The "storage manager", instead, is in charge of maintaining the memory cache in which the various pre-fetched items are stored. Among other tasks, it keeps track of the user's accesses to the various items stored in the cache and implements a cache policy to remove data objects when the cache is full.

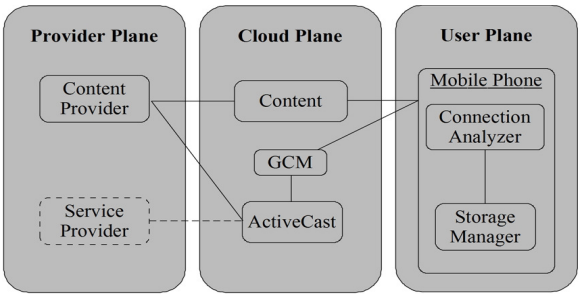

Fig. 2. ActiveCast system architecture, API and apps. The various system components are grouped into "Provider Plane", "Cloud Plane" and "User Plane".

The cloud plane includes the server in which the content is stored (outside the ActiveCast system) and also the ActiveCast cloud server. The communication between the ActiveCast system and the terminal application is performed using Google Cloud Messaging (GCM). Finally, the provider plane represents the domain of the various content providers and the service providers (e.g. mobile operators and local access providers). Since currently no content provider has access to the ActiveCast API, for testing purposes we implemented our own content provider, "Social Video", a third party application integrated with Facebook and designed to identify YouTube videos populating the "feed" of the Facebook accounts that have registered with the ActiveCast system and subscribed to content pre-fetching services. The ActiveCast server polls at periodic intervals the registered accounts to identify any new videos populating their feeds. Whenever a new videos is detected for a specific user, a new content pre-fetching entry is created (with all information described in Section IV-A) and transmitted to the mobile terminal corresponding to that specific Facebook account. Later on, whenever the user tries to access the video, the mobile app serves the content from the local cache if the video has already been entirely pre-fetched.

During the actual content download operations, the implemented OTT strategy compares the currently achievable datarate R with the target datarate $\hat{R}$, computed to maintain a specific energy budget as described in Section IV-B. The average datarate is monitored every 5 seconds during the active transfer of content. Whenever it falls below the $\hat{R}$ threshold the data transfer is paused and resumed only after τ seconds. This quantity is defined as the "wake-up timer" and throughout the experiments described in this paper this timer was set to a constant value equal to thirty seconds. In general, the value of τ can vary according to an adaptive scheme where the wake up time increases if the previous n-attempts to transfer failed to meet the target.

VI. Evaluation

A. Video provision performances

This section describes the comparison between the performances of pre-fetching and current streaming solutions, for the

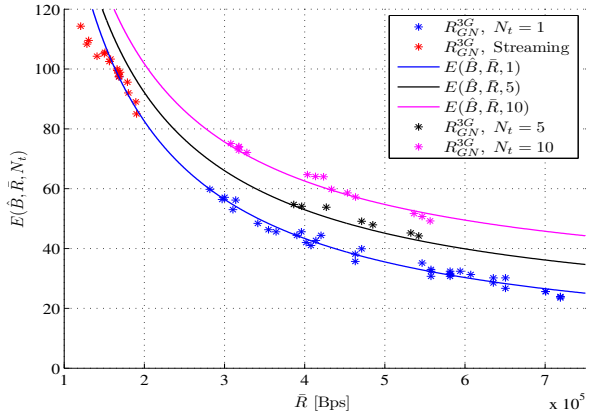

Fig. 3. Total energy costs, as function of $\bar{R}$, for delivering a data object of size $\hat{B} = 27$MBytes. In this figure a subset of experimental realizations (R^{3G}_{GN}), organized in function of specific numbers of experienced tail time occurrences (i.e. $N_t = 1, 5, 10$), is shown together with the corresponding energy curves obtained with the model of Eqn. 1 and a number of realizations for the streaming content provision via YouTube.

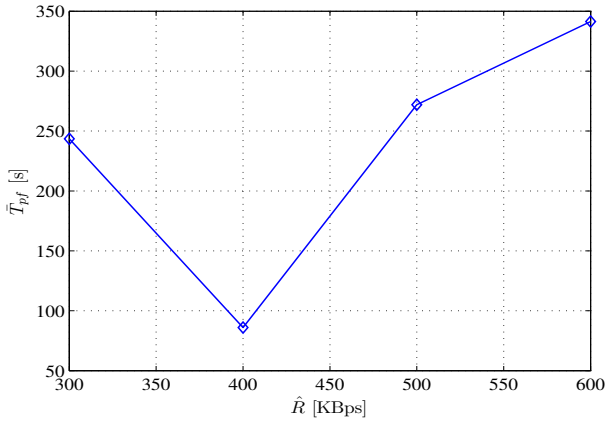

Fig. 4. Average time to complete the content pre-fetching T_{pf} as function of different pre-fetching target datarates $\hat{R}$, for an item of size $\hat{B} = 27$MBytes.

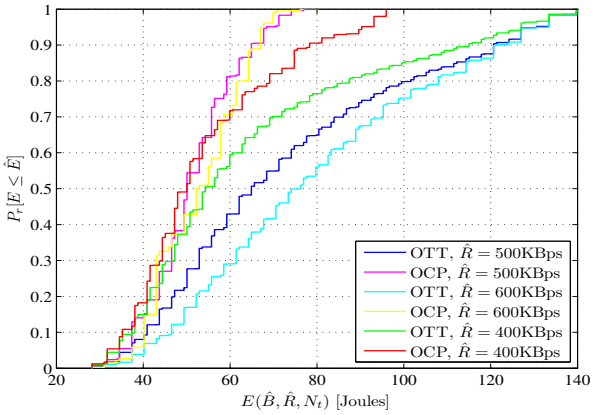

Fig. 5. Cumulative distribution function for the energy costs of pre-fetching a video object of size $\hat{B} = 27$MBytes ($E(\hat{B}, \hat{R}, N_t)$), for both the OTT and the OCP implementations and for different datarate thresholds $\hat{R}$.

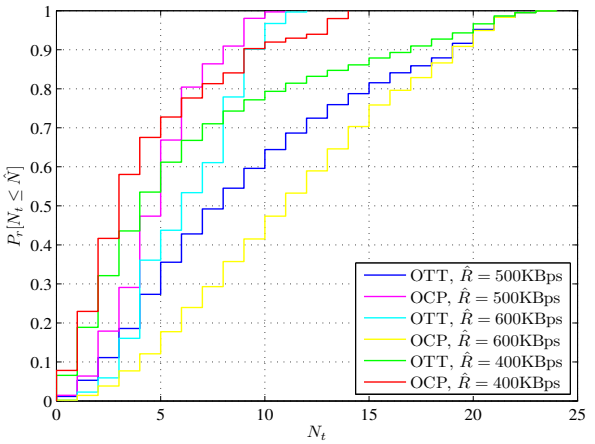

Fig. 6. Cumulative distribution function the number of tail time occurrences N_t when pre-fetching a video object of size $\hat{B} = 27$MBytes ($E(\hat{B}, \hat{R}, N_t)$), for both the OTT and the OCP implementations and for different thresholds $\hat{R}$.

provision of YouTube videos. The results concerning the pre-fetching energy costs are shown in Fig. 3, for different values of $\bar{R}$ achieved during the content download. Note that the actual energy realizations R^{3G}_{GN} are grouped based on the number of tail time occurrences, with focus on those having $N_t = 1$, $N_t = 5$ and $N_t = 10$. The various realizations are obtained considering a set of values for the target datarate $\hat{R}$, ranging between 300KBps and 800KBps, and are recorded during several days at different times (thus exposed to different average traffic loads in the serving cell). In the same Fig. we have also plotted the energy values of the realizations corresponding to standard YouTube video streaming content provision. The results show that the current streaming solutions are extremely inefficient from an energetic perspective, leading to increased energy costs by a factor between 1.33 and about 5 times, as compared to video pre-fetching performed on the same 3G interface. Apart from opportunistically exploiting higher datarates for downloading portions of a video, the gains of content pre-fetching also depend on the specific policy adopted by YouTube (but also by other providers) for minimizing their dimensioning costs: after an initial burst in which a given percentage of the video is moved in the streaming buffer at the highest achievable speed, the bandwidth allocated by the YouTube server rapidly converge to the bitrate of the video.

Apart from the quantification of its potential energy savings it is also relevant to understand the amount of time T_{pf}, required in average by content pre-fetching to completely pre-load a video of size $\hat{B} = 27$MBytes in the user terminal. This result is shown in Fig. 4. There it is shown that it might take between slightly more than a minute and about six minutes, depending on the targeted datarate $\hat{R}$. The minimum value is obtained for $\hat{R} = 400$KBps, which closely corresponds to the average datarate $\bar{R}$ achieved across all realizations. In particular, for values of $\hat{R}$ lower than $\bar{R}$, longer pre-fetching durations depend on the fact that in most of the cases the pre-fetching operations are performed at slower speeds. For values of $\hat{R}$ greater than $\bar{R}$, instead, the longer durations of the pre-fetching phases depend on the difficulty of achieving the wanted target rates, which in turns increases the number of tail time occurrences (with an associated waiting time of 30 seconds for each occurrence).

B. Limits of the OTT approach

To evaluate the potential impact of network-context gathering through probing (see Section IV-C) on the performances of the proposed OTT content prefetching scheme, we performed a simulated investigation comparing our proposed OTT scheme with an ideal case, called Operator Controlled Pre-fetching (OCP). In the OCP scheme the content pre-fetching operations are coordinated by a mobile operator, which is considered both capable of ideally estimating the datarate achievable by a mobile terminal and aware of the target datarate required for downloading a pending data object. Whenever the achievable datarate exceeds the target $\hat{R}$, the mobile operator triggers the download of the content by notifying the terminal app. After the content pre-fetching is initiated, it is the terminal app that keeps track of the average datarate in time intervals of 5 seconds. As soon as $\bar{R}$ falls below $\hat{R}$ the terminal pauses the download, until a new activation opportunity is notified by the mobile operator.

The comparison between the performances of our proposed OTT scheme and the ideal OCP has been performed by simulation, considering as input real datarate measurements, recorded by our testing devices in a continuous download of data for 4 hours using the Melange application [2].

The energy cost for completing the pre-fetching of a video of size $\hat{B} = 27$MBytes is shown in Fig. 5 for both the OTT and OCP cases and different values of $\hat{R}$. The results show that the OTT scheme incurs moderate losses as compared to the ideal OCP. In particular, for the median user the OTT scheme has an increase in energy budget that varies between 1.08 and 1.44, depending on the target datarate. This also reflects in an increased number of tails experienced by the OTT scheme, as shown in Fig. 6, where it is shown that the median user incurs between 1.33 and 1.83 more tails than with the ideal OCP.

C. Optimal Tail Time

The tail time is a design parameter with a major impact on the energy consumption of mobile data services. At the same time, determining an optimal value for the tail time is not trivial since it needs to account for complex user behavior [10] and specific services. In some cases larger values of tail time can be beneficial for certain types of traffic while in some other cases [11] mobile terminals can waste up to 30% of their energy in tail times. These contrasting objectives are clearly illustrated in [9], where the authors showed that increasing the tail time saved 30% of the energy for a particular app, while wasting 41% for another app, thus leading to a net loss of 11%. Current proposals are covering a broad spectrum of solutions, with on the one hand methods like fast dormancy, designed to shorten the tail time, but difficult to implement and therefore potentially leading to limited gains. On the other extreme, other researches are aiming at reusing the tail time for pushing content [12]. In agreement with [13] we believe that the tail duration should be adaptive, optimizing its duration in function of the specific services that are provided. In particular, while performing content pre-fetching, the tail time should be minimized, provided that the mobile operators can be made aware, e.g through the ActiveCast API, when they are serving pre-fetching traffic.

VII. Conclusions

The key system components of ActiveCast, a novel context-aware paradigm for content delivery in mobile networks, have been presented and experimentally evaluated in respect of video provision services. By adopting an opportunistic content pre-fetching scheme, which exploits information on content access probability to select appropriate datarate requirements for content pre-fetching, the ActiveCast system has been shown to deliver predictable energy costs for content pre-fetching and that these are significantly lower than the energy costs currently achievable with video streaming content provision.

In order to perform the experimental evaluations presented in this paper, the entire components of the ActiveCast system, together with an API for content providers and mobile operators as well as an Android mobile application have been developed and included into a "living laboratory" testbed, allowing to perform the actual performance evaluation of novel services in real networks, with real users and mobile devices.

By tracking achievable data rates in different times and locations we have also gathered relevant evidence to support the definition of novel and more flexible SLAs with content providers, providing strict guarantees on the upper bounds of the delivery delays while allowing to optimize the energy budget invested by both the users' devices and the wide area networks to complete the download of content in the user terminals. The evaluation of different SLAs for content pre-fetching and the validation of the associated business models constitute one of the key directions of our current research. At the same time, we are also currently pursuing opening the ActiveCast API to a set of selected content providers and increase the size of our testbed user base by opening the ActiveCast app on the Android market and the AppStore.

References

[1] Cisco visual networking index: Global mobile data traffic forecast update. [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf

[2] P. Lungaro *et al.*, "An experimental framework to investigate context-aware schemes for content delivery," in *Vehicular Technology Conference 2011*.

[3] P. Lungaro *et al.*, "Context-aware rrm for opportunistic content delivery in cellular networks," in *Proc. of the 2010 Third International Conference on Communication Theory, Reliability, and Quality of Service, CTRQ '10*.

[4] P. Lungaro *et al.*, "Application-centric content delivery schemes for future wireless networks," in *IEEE 22nd International Symposium on Personal Indoor and Mobile Radio Communications, PIMRC '11*.

[5] B. D. Higgins *et al.*, "Informed mobile prefetching," in *Proc. of the 10th international conference on Mobile systems, applications, and services, MobiSys '12*.

[6] B. Higgins *et al.*, "Intentional networking: opportunistic exploitation of mobile network diversity," in *Proc. of the sixteenth annual international conference on Mobile computing and networking, MobiCom '10*.

[7] A. Schulman *et al.*, "Bartendr: a practical approach to energy-aware cellular data scheduling," in *Proc. of the sixteenth annual international conference on Mobile computing and networking, MobiCom '10*.

[8] L. Zhang *et al.*, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *Proc. of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis, CODES/ISSS '10*.

[9] A call for more energy efficient apps. [Online]. Available: http://www.research.att.com/articles/featured_stories/2011_03/201102_Energy_efficient?fbid=R48eKchNsAg

[10] H. Falaki *et al.*, "Diversity in smartphone usage," in *Proc. of the 8th international conference on Mobile systems, applications, and services, MobiSys '10*.

[11] J. Huang *et al.*, "A close examination of performance and power characteristics of 4g lte networks," in *Proc. of the 10th international conference on Mobile systems, applications, and services, MobiSys '12*.

[12] H. A. Lagar-Cavilla *et al.*, "Traffic backfilling: subsidizing lunch for delay-tolerant applications in umts networks," in *Proc. of the 3rd ACM SOSP Workshop on Networking, Systems, and Applications on Mobile Handhelds, Mobiheld '11*.

[13] F. Qian *et al.*, "Characterizing radio resource allocation for 3g networks," in *Proc. of the 10th ACM SIGCOMM conference on Internet measurement, IMC'10*.