Demo: Media Download Optimization through Prefetching and Resource Allocation in Mobile Networks

Christian Koch[1], Nicola Bui[23], Julius Rückert[1], Guido Fioravantti[23], Foivos Michelinakis[23], Stefan Wilk[1], Joerg Widmer[2], David Hausheer[1]
[1]Technische Universität Darmstadt, [2]IMDEA Networks Institute, [3]Universidad Carlos III de Madrid
{nicola.bui|foivos.michelinakis|guido.fioravantti|joerg.widmer}@imdea.org,
{ckoch|rueckert|hausheer}@ps.tu-darmstadt.de, swilk@cs.tu-darmstadt.de

ABSTRACT

Mobile network operators are expected to face significant traffic increase in the upcoming years. One alternative method is to intelligently move transmissions to times of network underutilization, either on 3G/4G or by offloading to WiFi. Video content, predicted by Cisco to constitute 69% of mobile traffic, offers the greatest potential for offloading. To this end, the demonstrated app strives to relieve the mobile network in a two ways. First, long-term prefetching of promising videos based on posts from the user's Online Social Network feed is performed. The knowledge about which video is likely being requested in the near future offers the opportunity to schedule the transmission according to its probability of being watched. Second, the approach is complemented with short-term prefetching, which is used whenever a content could not be downloaded by long-term prefetching. In this case, resources are optimized so as to maximize the communication efficiency while preserving the quality of service. The demonstrated app considers the smartphone's observed cellular network history to optimize the mobile throughput. A customized video player implements both the long-term and short-term prefetching. It reduces both the load on mobile networks, decreases playback pausing events and hereby achieves a high QoE. Thus, the player addresses both the operators' and the users' needs.

Categories and Subject Descriptors

H.5.1 [**INFORMATION INTERFACES AND PRESENTATION**]: Multimedia Information Systems—*Video*

General Terms

Experimentation, Performance

Keywords

Video Streaming, Prefetching, Social-aware, Network-aware

MMsys '15, Mar 18-20 2015, Portland, OR, USA
ACM 978-1-4503-3351-1/15/03.
http://dx.doi.org/10.1145/2713168.2713187

1. INTRODUCTION

Mobile Internet access is an indispensable part of today's life. Many services, e.g. Facebook and Twitter are mainly accessed through mobile networks. The worldwide mobile traffic has increased by 81% in 2013 [3]. The monthly data volume is expected to increase tenfold, whereas the capacity is expected to only increase twofold till 2018. This raises new challenges for mobile network operators and endangers a stable service. Femto cells and the allocation of additional frequency bands for mobile networks alone are not sufficient to overcome this issue [8]. Efficient content distribution strategies, e.g. prefetching, can reduce this gap.

This demonstration shows that efficient content distribution strategies as developed in the eCOUSIN research project [1] can help both users and mobile operators. The demo consists of three main contributions: First, mobile network offloading by enabling smartphones to prefetch videos form Online Social Networks (OSNs) over WiFi is demonstrated. Second, the impact of resource allocation optimization is shown by showing how a video stream could be transmitted with a more efficient resource usage while maintaining the users' QoE. Third, we visualize the achieved gains in network performance with controlled experiments. An Android app called *Mobile Social Prefetcher* capable of prefetching videos over WiFi is demonstrated. Additionally, if prefetching is not suitable, streaming is optimized to ensure a high QoE. In particular, when the smartphone user starts the playback of a remotely stored video, the app opens a customized video player, which allows both local playback of prefetched content and QoE-aware streaming. This player - instead of, e.g. the YouTube app or the browser - displays the video. If the video is not prefetched, it will be streamed by dynamically adapting the transfer rate depending on the playback buffer and the network conditions. The system demonstrates that by prefetching content on-demand over WiFi, the costs can be significantly reduced including energy and mobile data plan expenses. Short-term prediction helps optimizing the bandwidth usage of the mobile connection (e.g., to smooth traffic peaks and to reduce load on congested cells). Long-term prediction helps to actively load contents in advance. For this case, it has been shown, e.g., in [9], that OSNs are predictive for single user video consumption. In the following, Section 2 describes the application architecture, while Section 3 gives details about the demonstration.

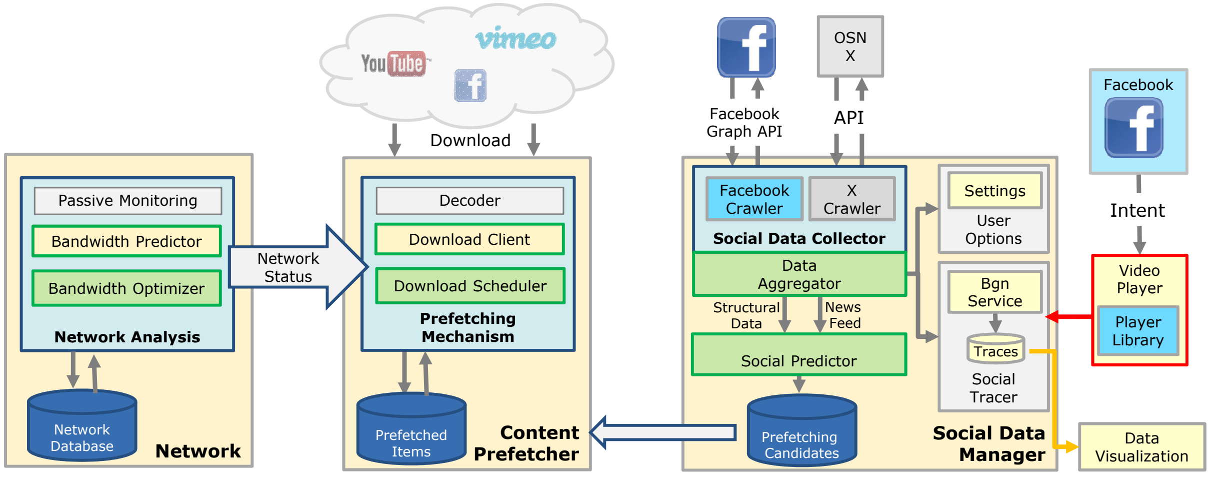

Figure 1: Architecture of the Mobile Social Prefetcher

2. SYSTEM ARCHITECTURE

Fig. 1 depicts the architecture of the *Mobile Social Prefetcher* app. The main components include the *Social Data Manager*, the *Content Prefetcher*, the *Network* component and a customized video player. All components run on the user's mobile device.

2.1 Social Data Manager

The *Social Data Manager* continuously monitors the user's OSNs. For every OSN, a special crawling module is used. This key module of the demonstrated software accesses the user's OSN-feed to request information about video posts. Additionally, the ongoing interactions of the user with his friends and his content-specific interactions are monitored.

The *Social Data Collector* hands the information acquired by these modules to the *Data Aggregator Module.* Meta data such as timestamps and dates are stored in a uniform manner in a local database by the *Social Tracer*. To determine which items should be prefetched first the *Social Predictor* stores the candidates with a priority in the local *Prefetching Candidates* database, which is used by the *Content Prefetcher*. Prioritizing different videos is a topic of ongoing research. For the demo, the priority is derived by a simple metric based on the video's popularity. To this end, the number of Facebook likes for a given video is used. Currently, the application supports Facebook, but it will be extended for further OSNs in the future.

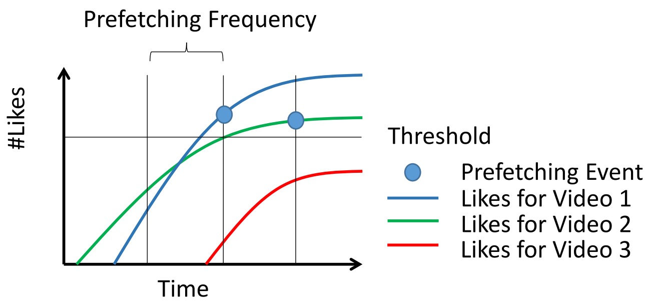

Figure 2: Videos are prefetched in order of their like count if a certain threshold is exceeded.

2.2 Content Prefetcher

The *Content Prefetcher* identifies relevant videos for a user and downloads them whenever Wifi connectivity is available. This is done at a particular interval (prefetching frequency) which has been set to 15 minutes for the presented demo. Relevant videos include popular videos that have recently been posted on the user's Facebook feed. For the demo, only videos posted within the last 24 hours are considered. A video is considered popular if the number of Facebook likes for the corresponding video post exceeds a certain threshold. The like count also determines the order in which videos are prefetched. To illustrate this, Fig. 2 provides an example with three video posts on the user's Facebook feed and the corresponding like count evolution over time. As the like count exceeds the threshold, the respective videos are downloaded by the Content Prefetcher during the next prefetching interval. The maximum number and frequency at which videos that are downloaded are design parameters which will be investigated in more detail in future work. The *Decoder* translates the OSN URL to the URL of the corresponding mp4 file and passes it to the *Download Client*. The *Download Client* performs the content download, takes care of connection interruptions, and selects a resolution appropriate for the smartphone's display if multiple are available. The *Download Scheduler* allows download scheduling for the prefetching candidates. This enables postponing the download until certain conditions are met, e.g. WiFi is available.

2.3 Bandwidth Optimization

To be able to feed the *Bandwidth Optimization* module with predictions and their confidence, a *bandwidth prediction* module is used. This module considers statistical information about user mobility and the available bandwidth in a given mobile network cell. In order to combine all the statistical information, different predictors will be jointly used. In particular, we use different solutions for the short (e.g., tens of seconds, a few minutes) and the medium-long (e.g., tens of minutes, hours) term predictions.

The short term prediction is achieved by means of a simple AutoRegressive-Moving Average (ARMA) filter [5]. The algorithm for applying the filter and its coefficients have been previously tuned according to user's past information.

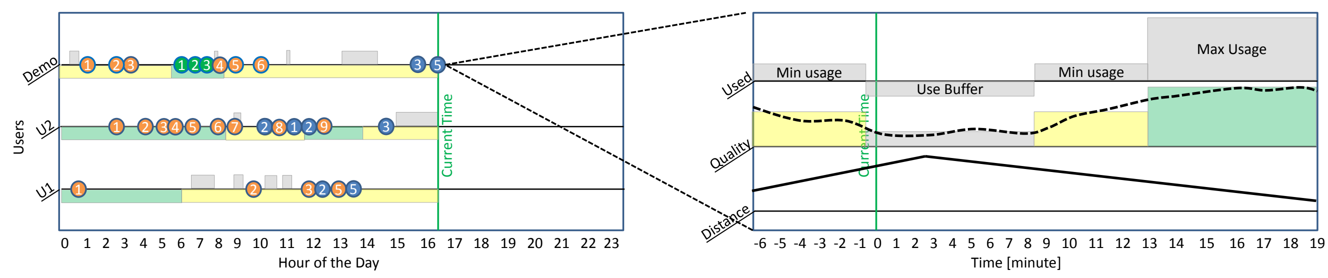

Figure 3: Left side: circles represent video events on the time line of a given users. These events are color coded so that orange stands for publish time, green for prefetch time and play for watch time. The area behind the dots is coloured yellow or green if the mobile has mobile or WiFi access respectively. Right side: Video 5 is streamed from remote as it has not been prefetched before. The figure illustrates the bandwidth usage policies depending on the measured and forecast signal quality.

Depending on the user's movement speed, the prediction validity varies between a few ten of seconds (fast movements) up to some minutes (quasi-static scenarios). During the validity time, the filter predicts the future mobile throughput.

The medium-long term prediction is performed by statistical models [4]. These models account for the degradation on the accuracy of both the user position and the network cell congestion while the prediction is made in the future.

By combining the two prediction techniques, this component decides when it is best to prefetch a content and, if the content has to be streamed, what is the best way to allocate resources in order not optimize the cost.

3. DEMONSTRATION

The demonstration involves an Android smartphone and two WiFi hotspots. One hotspot simulates a cellular network during the demo (SSID: *Cellular*). A second hotspot is considered to be an ordinary WiFi hotspot (SSID: *WiFi*). One part of the demo is shown on a smartphone on which the app is installed. The app's behaviour on the smartphone is shown by connecting it to one of the two hotspots. As soon as the user starts a video playback, our video player, which is integrated in the app, can be selected and chosen for playback. The demonstration will focus on two scenarios, in which background processes are visualized on a laptop. The app is the basis for an international study focussing on Germany, Spain, and France. During the demonstration, the collected data will be visualized in a privacy-preserving manner. Fig. 3 shows such an example of a typical usage diagram. The dots on the left indicate video posts on the corresponding user's Facebook feed. For example, the first orange circle of the demonstration user, marked with 1, indicates a video on his Facebook feed. As soon as WiFi gets available, the app begins prefetching the videos, in this case Video 1, 2, and 3. Due to legal restrictions only during the demonstration content prefetching is enabled. The experimental version [2], that everyone can retrieve disables prefetching. Prefetched videos are indicated by a green circle. If a user watches a video, it is indicated by a blue circle, e.g., Circle 5. In case of this video prefetching prediction did not list it as a candidate. It has not been prefetched. Thus, it has to be streamed. The buffer filling strategy and the network connectivity prediction is shown on the right side. During the demonstration, this network conditions and the predictions are simulated using real traces. The reason for this is that the prediction model is based on the ARMA model, which is based on regular movement and connectivity patterns of the app user, as described earlier. The app is the basis for an international study which focuses on Germany, Spain, and France. During the demonstration, the collected data will be shown in a privacy-preserving manner. Fig. 3 shows an example view of what we plan to demonstrate. Here the data for a subset of all participating users and the demo device will be visualized live. New posts occurring on the users feed are indicated by a dot on the user's time-line. The vertical green line indicates the current time, which moves during the demo. Prefetched videos are indicated by highlighting the post-circles green, which applies for the demo device only. Furthermore, above the user time-line it is indicated if the screen of the user was enabled and for how long. This information offers meaningful data, e.g., when and for how long users use their smartphone and enables the development of individual prediction models. Specific models are planned as further research described in [7]. Underneath the user time-line, the network interface by which the user is connected to the Internet, is shown. Here three different colors indicate the connection type. WiFi networks are indicated by green, 3G as grey, and LTE as yellow. WiFi connections observed are important, since they indicate offloading possibilities. This is an important observation since prefetching performed over WiFi is about 10-times more energy efficient and often faster than over cellular networks [6].

3.1 Scenario: Prefetching

The first demonstrated scenario is the *long-term prefetching scenario*. Prefetching will only be performed if the smartphone is connected to a *WiFi* network. The video content, on which the app relies, is retrieved from a feed of a Facebook profile created for the demonstration. This profile is assigned to a couple of channels which post videos. Additionally, the audience is offered to create new posts on this Facebook feed. To illustrate the app's behaviour in a repetitive manner, a *discard* function is implemented in the app, which will delete all prefetched videos. Furthermore, it enables our second scenario, where a video is streamed in a network-friendly manner. In case the app is connected to *WiFi* the prefetching of the latest videos becomes visible on the smartphone immediately. Additionally, for the demonstration, a live view on the content becoming available at the feed is visualized with orange circles, see Figure 3.

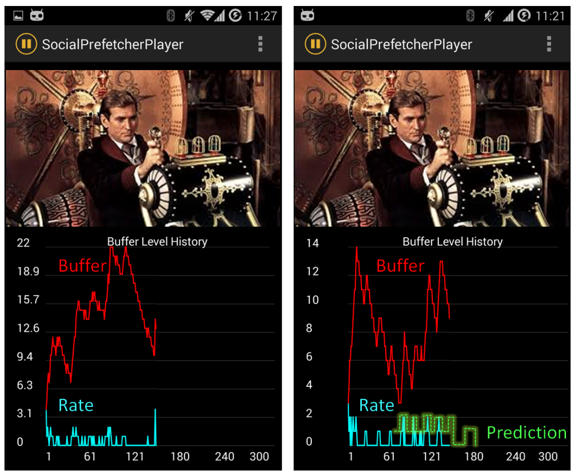

Figure 4: Left side: Normal player internal. Right side: Prediction-based optimized player.

3.2 Scenario: Streaming

The second scenario to demonstrate is the *short-term prefetching scenario*. If the connection is established to a cellular network, the *Mobile Social Prefetcher* will perform a local playback if the content was prefetched during a previous WiFi connection. All playback functionality is implemented in a custom video player. In case the video is not prefetched, the *Bandwidth Optimizer* module optimizes the video streaming. To show this during the demo, the bandwidth of the simulated cellular network will be shaped to illustrate the app's operation under realistic conditions, when the connectivity changes over time. Our custom video player shows at the top of the screen the actual video, and, at the bottom, the internals of the application related to the buffer status, the used bandwidth according to the *Bandwidth Optimizer* module and the predicted bandwidth availability.

The video playback is smoothed without pausing or stalling, while the bandwidth will be only used when the (emulated) capacity will be high. In order to show the difference with respect to the normal operation of the Android's default player we will run the same demonstration excluding the *Bandwidth Optimizer* module. See in Figure 4 two screenshots of the application UI: on the left, the buffer status and the bandwidth usage of a normal player are shown in red and cyan respectively, while on the right the same information is visualized for our improved application as well as the predicted bandwidth availability (green).

Compared to the normal player, our optimized application is able to refrain from using the bandwidth if it is low and it is not needed for the desired quality of experience. In addition, the prediction visualization illustrates how effectively the application made used of the periods in which the prediciton of the achievable data rate has been higher. Finally, the buffer status of the optimized player is consistently emptier compared to a normal player, thanks to prediction. In fact, knowing that a higher data rate is available in the future allows the optimized player to avoid buffering too much data, when this is not necessary.

4. SUMMARY

The demonstrated *Mobile Social Prefetcher* app has the potential to enhance the users' QoE and at the same time to cope with varying network conditions. Relying on the videos posted on the user's personal social network feed, the app prefetches videos if WiFi is available. The playback of prefetched videos will be performed locally when requested by the user. If a video could not be prefetched because it was posted quite recently or no WiFi is available, a network-friendly streaming is performed. Here, the video player buffer is filled depending on the current network conditions and, therefore, reduces stalling events even under bad network conditions. This is demonstrated in a comparison of our app against the native Android video player under realistic network conditions.

Since we use a video player which can be handled as any other video player, our approach minimizes the effort for the user. Concluding, this demonstration offers an interactive experience using Facebook and video playbacks. The video post publishing, prefetching, playback, and streaming are shown on a live view during the demo. Additionally, data collected from other users using the app in a user study are also shown live.

Acknowledgments

This work has been supported in parts by the EU (FP7/#317846, SmartenIT and FP7/#318398, eCOUSIN) and the DFG as part of the CRC 1053 MAKI. The authors would like to thank Björn Scheurich, Mohammad Hosseinian, and Timo Thräm for their contribution of the mobile social prefetcher implementation.

5. REFERENCES

[1] EU FP7 eCOUSIN. http://www.ict-ecousin.eu/.
[2] Social Monitor App. http://tinyurl.com/socialmonitor.
[3] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014.
[4] N. Bui et al. A Model for Throughput Prediction for Mobile Users. In *European Wireless*, 2014.
[5] N. Bui and J. Widmer. Modelling Throughput Prediction Errors as Gaussian Random Walks. In *1st KuVS Workshop on Anticipatory Networks*, 2014.
[6] J. Huang et al. A Close Examination of Performance and Power Characteristics of 4G LTE Networks. In *ACM MobiSys*, pages 225–238, 2012.
[7] C. Koch and D. Hausheer. Optimizing mobile prefetching by leveraging usage patterns and social information. In *IEEE ICNP*, 2014.
[8] A. Solheim. Microwave Backhaul Radios Meet The Evolving Traffic Challenge. Mobile Dev & Design, February 2013.
[9] Y. Zhao et al. O2SM: Enabling Efficient Offline Access to Online Social Media and Social Networks. In *Middleware*, volume 8275, pages 445–465. Springer LNCS, 2013.