Multi-Source Mobile Video Streaming with Proactive Caching and D2D Communication

Vasilios A. Siris and Dimitrios Dimopoulos
Mobile Multimedia Laboratory, Department of Informatics
School of Information Sciences & Technology
Athens University of Economics and Business, Greece
Email: {vsiris, dimdimopoulos}@aueb.gr

***Abstract*—We present testbed experiments of multi-source mobile video streaming that jointly exploits mobility and throughput prediction to prefetch video data in caches located at hotspots that the mobile will encounter and device-to-device (D2D) communication to opportunistically obtain parts of a video from neighboring mobile devices. A contribution of this paper includes a procedure to efficiently utilize multirate Wi-Fi links, which can be legacy Wi-Fi hotspot links and device-to-device links using Wi-Fi direct.**

I. INTRODUCTION

A major trend in mobile networks is the exponential increase of powerful mobile devices, such as smartphones and tablets, with multiple heterogeneous wireless interfaces that include 3G/4G/LTE and Wi-Fi. The proliferation of such devices has resulted in a skyrocketing growth of mobile traffic, which in 2014 grew 69%, becoming nearly 30-times the global Internet traffic in 2000, and is expected to grow 10-fold from 2014 until 2019[1]. Moreover, mobile video was 55% of the total traffic by the end of 2014 and is expected to increase 13-fold from 2014 to 2019, becoming 72% of the total mobile traffic. Efficient support for video streaming in future mobile environments, in terms of both network resource utilization and energy consumption, will require the integration of heterogeneous wireless technologies with complementary characteristics; this includes cellular networks that support wide-area coverage and Wi-Fi hotspots that support high throughput and energy efficient data transfer. The industry has already verified the significance of mobile data offloading to exploit fixed broadband and Wi-Fi technology: globally, 46% of total mobile data traffic was offloaded onto Wi-Fi networks or femtocells in 2014[1].

The contribution of this paper is to jointly evaluate prefetching and device-to-device communication for improving the performance of multi-source mobile video streaming. Moreover, we present and evaluate a new procedure for efficiently utilizing multiple Wi-Fi links with different transmission rates, which can include legacy Wi-Fi hotspot links and Wi-Fi direct links between mobile devices. Prior work has verified that mobility and throughput prediction is possible; this paper is not concerned with developing a system for such prediction, but rather focuses on an actual implementation of mechanisms that exploit such prediction. The work in this paper is different from our previous work in [17], [15] that considers mobile data offloading for delay tolerant traffic, which requires transferring a file within a time threshold, and delay sensitive traffic, which requires minimizing the file transfer time; unlike these traffic types, video streaming requires a continuous transfer of video data to avoid degradation of a user's QoE (Quality of Experience), thus necessitates a totally different prefetching procedure and evaluation. Compared to [2], in this paper we jointly consider the performance gains achieved with prefetching and device-to-device communication, and we present a procedure for efficiently utilizing multirate Wi-Fi links.

The rest of the paper is structured as follows: In Section II we present related work. In Section III we present the design of the multi-source mobile video streaming client and in Section IV we present the mechanisms for prefetching and efficient utilization of multirate Wi-Fi links. In Section V we present experiments that illustrate the performance gains in terms of increased mobile data offloading and improved QoE.

II. RELATED WORK

Prior work has demonstrated bandwidth can be predicted for both cellular networks [20] and Wi-Fi [11]. Bandwidth prediction for improving video streaming is investigated in [21], [5], and for client-side pre-buffering to improve video streaming in [14]. The work in [21], [5], [14] focuses on cellular networks, whereas we consider integrated cellular and Wi-Fi networks. Moreover, our goal is not to develop a new system for mobility and bandwidth prediction, but to exploit such prediction to prefetch data in order to improve mobile video streaming.

Multi-source video streaming for improving robustness in mobile ad hoc networks is investigated in [13], which focuses on video and channel coding, and [9]. The work in [3] investigates joint routing and rate allocation for multi-source video streaming in wireless mesh networks. Load balancing over multiple radio interfaces is investigated in [4], which focuses on client-side scheduling. [18] investigates load balancing

This research has been co-financed by the European Union (European Social Fund-ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF)-Research Funding Program: Aristeia II/I-CAN.

[1]Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014-2019, Feb. 5, 2015

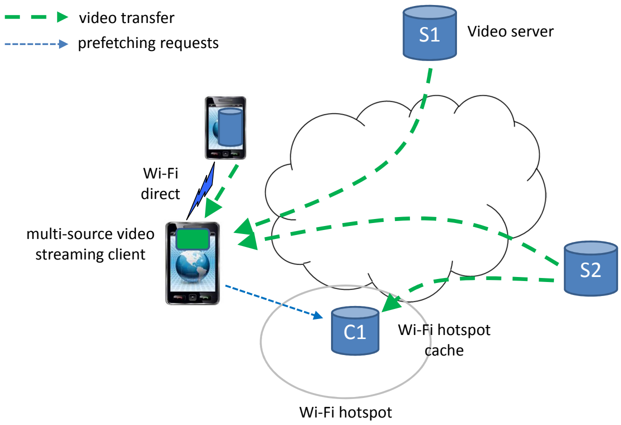


Fig. 1. The multi-source mobile video streaming client can obtain video data from multiple sources: remote video servers, local hotspot caches that prefetch video data, and neighboring devices.

by probabilistically splitting a video flow across multiple radio interfaces based on video transmission patterns. The adaptation of P2P techniques for multi-source video streaming to Android clients is investigated in [12]. The work in [8] investigates cooperation between mobile devices to exploit device-to-device communication for video streaming.

The feasibility of using prediction for prefetching is investigated in [1], which however does not propose or evaluate specific prefetching algorithms. Prefetching for improving video file delivery in cellular femtocell networks is investigated in [7], and to reduce the peak load of mobile networks by offloading traffic to Wi-Fi hotspots in [10]. Our work differs from the above by presenting an testbed implementation of multi-source mobile video streaming that jointly exploits prefetching and device-to-device communication, and a procedure to efficiently utilize multirate Wi-Fi links.

III. Multi-source mobile video streaming client design

The multi-source mobile video streaming client can utilize both cellular and Wi-Fi interfaces, and request different parts (chunks) of a video from different sources. These sources can be remote video servers, local caches in Wi-Fi hotspots that the mobile will encounter, or neighboring devices, Figure 1.

In addition to load balancing and fault tolerance mechanisms which are described in [2], the client implements the following two procedures:

- Video prefetching for enhanced offloading: the client exploits mobility and throughput prediction to request from local caches in hotspots that it will encounter to prefetch parts of the video, so that they are immediately available when the mobile connects to these hotspots.
- Efficient utilization of multirate Wi-Fi links: Scheduling of video chunk transfer over multiple Wi-Fi links, which can include Wi-Fi hotspot and Wi-Fi direct links, is determined based on their transmission rates and the ADSL backhaul throughput.

The prefetching procedure has already been presented in our previous work [2]; the contribution of this paper is to jointly evaluate prefetching with device-to-device communication using Wi-Fi direct.

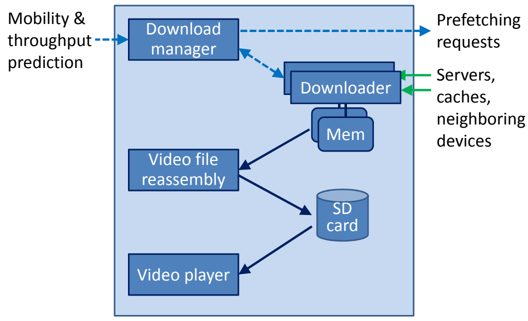


Fig. 2. Multi-source mobile video streaming client design.

The high-level design of the mobile video streaming client is in Figure 2. The main components are the download manager and the downloaders. The download manager uses mobility and throughput prediction information to instruct caches, located in Wi-Fi hotspots that the mobile will encounter, to prefetch video chunks. The download manager also controls and synchronizes the downloaders, and is responsible for discovering neighboring devices with which the mobile can connect to using Wi-Fi direct. Each downloader transfers video chunks from a different source. TCP-based video streaming, by breaking the video into multiple chunks, is used in the MPEG-DASH standard. We did not use the MPEG-DASH video standard because at the time of our implementation there was no stable MPEG-DASH player for Android. Nevertheless, the design of the multi-source mobile video streaming client and the procedures implementing the aforementioned functionality are independent of the specific transfer protocol.

To download video from multiple servers, the mobile client needs to know the IP addresses of these video servers, which can be included in the mobility prediction information or in metadata files such as MPEG-DASH's Media Presentation Description (MPD). Alternatively, knowledge of the video servers' IP addresses is not necessary in Information-Centric Network (ICN) architectures, where users request content based on the name for the content [19].

A device's mobility, in terms of different connectivity options and download throughput for cellular, Wi-Fi, and xDSL backhaul links, is emulated based on an experiment scenario description contained in an XML (Extended Markup Language) file. The scenario description file specifies the mobile's connectivity, e.g. Wi-Fi or cellular, for different segments and the IP addresses of the video servers and caches (in the case of prefetching); the connectivity segments are specified by their starting time. For each segment we can define a throughput value, which is enforced using the wondershaper network traffic shaping tool. The scenario description can also specify that the client searches for neighboring devices it can connect to using Wi-Fi direct. The above mobility (in terms of time-varying connectivity type) and throughput emulation provides the necessary flexibility to perform experiments with a range of parameters and assess the performance of the system in scenarios with a different number, location, and throughput of Wi-Fi hotspots. Finally, our testbed implementation can support scenarios where both the mobile and WiFi interfaces are used simultaneously; this is achieved with the tethering feature of Android devices, which allows both the mobile and Wi-Fi interfaces to be active at the same time.

Next we discuss some experiences with the implementation of Wi-Fi direct connectivity in the multi-source mobile video streaming client on Android mobile devices. A first observation is that to establish a Wi-Fi direct connection between two devices, it is necessary that the user explicitly accepts the Wi-Fi direct connection request; this is necessary both when the Wi-Fi direct connection is performed using Android's software and when an application performs the connection using the corresponding system calls. A second observation is that an Android device can simultaneously have a legacy Wi-Fi hotspot connection and a Wi-Fi direct connection to another device. Indeed, these two connections can operate on the same or on different channels. However, operating the two connections on different channels would require that the device performs channel switching; this can result in significant throughput degradation, even if one of the Wi-Fi links does not carry traffic. Finally, both Android's device discovery, which is performed by a device to discover neighboring devices to which it can connect using Wi-Fi direct, and service discovery, which is used to discover service offered by neighboring devices, can consume a significant amount of resources. For this reason, device and service discovery should be used in the beginning of a service connection and then stopped to avoid throughput degradation.

Unlike the multi-source mobile video streaming client which implements prefetching and video chunk scheduling mechanisms, the video server simply accepts requests for video chunks, which are stored locally in separate files. Finally, caches located in Wi-Fi hotspots are involved in video transfer only when prefetching is used.

IV. Mechanisms

In this section we describe in more detail the two mechanisms implemented in the multi-source mobile video streaming client: video data prefetching and efficient utilization of multirate Wi-Fi links.

A. Prefetching video data for enhanced offloading

Mobility prediction provides knowledge of how many Wi-Fi hotspots a mobile will encounter, when they will be encountered, and for how long the node will be in each hotspot's range. In addition to this mobility information, we assume that information on the estimated throughput in the Wi-Fi hotspots and the cellular network is also available; for the former, the information includes both the throughput for transferring data from a remote location, e.g., through an ADSL backhaul, and the throughput for transferring data from a local cache.

The procedure to exploit mobility and throughput prediction for prefetching is shown in Algorithm 1, which is implemented in the download manager of the multi-source mobile video streaming client. The algorithm extends the one investigated using trace-driven simulation in [16], by exploiting knowledge of the video buffer playout rate to reduce the throughput which it downloads video data over the mobile network. The procedure defines the mobile's actions when it exits a Wi-Fi hotspot, hence has only mobile access (Line 9), and when it enters a Wi-Fi hotspot (Line 14). Mobility and throughput prediction allows the mobile to determine when it will encounter the next Wi-Fi hotspot that has higher throughput than the cellular network's throughput. From the time to reach the next hotspot and the average video buffer playout rate, the mobile can estimate the position that the video stream is expected to reach (CurrentPosition + Offset) when it arrives at the next Wi-Fi hotspot (Line 10). It then sends a request to the cache in the next hotspot it will encounter to start caching video data from that position (Line 11). The video buffer playout rate is also used to estimate the throughput at which it should download video data while in the mobile network (Line 12).

Algorithm 1 Using mobility and throughput prediction to prefetch video data

1: **Variables:**
2: R_{playout}: average video buffer playout rate
3: $T_{\text{next Wi-Fi}}$: average time until node enters range of next Wi-Fi
4: *CurrentPosition*: current position of video stream
5: *Offset*: estimated offset of video stream when node enters next Wi-Fi hotspot
6: B: amount of video data in buffer
7: *RateMobile*: rate at which video is downloaded from mobile network
8: **Algorithm:**
9: **if** node exits Wi-Fi hotspot **then**
10: $\quad$ $\textit{Offset} \leftarrow R_{\text{playout}} \cdot T_{\text{next Wi-Fi}}$
11: $\quad$ Start caching video stream in next Wi-Fi starting from $\textit{CurrentPosition} + \textit{Offset}$
12: $\quad$ $\textit{RateMobile} \leftarrow R_{\text{playout}} - \frac{B}{T_{\text{next WiFi}}}$
13: $\quad$ Download video data from mobile network with rate *RateMobile*
14: **else if** node enters Wi-Fi hotspot **then**
15: $\quad$ Transfer video data that has not been received up to *Offset* from original location
16: $\quad$ Transfer video data from local cache
17: $\quad$ Use remaining time in Wi-Fi hotspot to transfer video data from original location
18: **end if**

When the node enters a Wi-Fi hotspot, it might be missing some portion of the video stream up to the offset from which data was cached in the hotspot; this can occur if, due to time variations, the node reaches the Wi-Fi hotspot earlier than the time it had initially estimated. In this case, the missing data needs to be transferred from the video's original remote location (Line 15), through the hotspot's backhaul link. Also, the amount of data cached in the Wi-Fi hotspot can be smaller than the amount the node could download within the time it is in the hotspot's range. In this case, the node uses its remaining time in the Wi-Fi hotspot to transfer data, as above, from the video's original location (Line 17).

B. Efficient utilization of multirate Wi-Fi links

Figure 3 shows a mobile with two Wi-Fi links, which can be a Wi-Fi link to a hotspot and a Wi-Fi direct link. We assume that the two links operate on the same channel, to avoid throughput degradation due to channel switching when the links operate on different channels. The mobile device can download video data over the Wi-Fi hotspot link, through the ADSL[2] backhaul, from a remote server and from the neighboring mobile device over the Wi-Fi direct link. If the Wi-Fi direct link has a higher transmission rate than the Wi-Fi hotspot link, then the mobile achieves a higher throughput if it uses only the Wi-Fi direct link.

On the other hand, if the Wi-Fi hotspot link has a higher transmission rate than the Wi-Fi direct link, then whether only the Wi-Fi hotspot or both the Wi-Fi hotspot and Wi-Fi direct links should be used to maximize the download throughput depends on the ADSL throughput. Specifically, if the ADSL throughput is higher than the throughput of the Wi-Fi hotspot link, then the aggregate throughput is maximized if only the Wi-Fi hotspot link is used to transfer video data from the remote server; in this case the Wi-Fi direct link is not used. However, if the ADSL backhaul is the bottleneck, i.e. the ADSL throughput is smaller than the throughput of the Wi-Fi hotspot, then the aggregate throughput is maximized if both the Wi-Fi hotspot and Wi-Fi direct links are used. The percentage of data transferred over the two links that maximizes the aggregate download throughput depends on the Wi-Fi hotspot and Wi-Fi direct transmission rates, and the ADSL throughput. To find this percentage we use the modelling framework presented in [6]. Assume there are two Wi-Fi interfaces operating on the same channel, with rates R_a and R_b, that transmit packets with the same size p. The throughput X achieved by each transmitter can be approximated by

$$X = \frac{p}{T(p, R_a) + T(p, R_b)} ,$$

where $T(p, R)$ is the time for transmitting a packet of size p over a Wi-Fi link with rate R; this time includes the IEEE 802.11 physical and MAC overhead. The above equation captures the influence of rate diversity of Wi-Fi transmitters operating on the same channel. For the scenario in Figure 3, if $R_{\text{hotspot}} > R_{\text{direct}}$, then the aggregate throughput is maximized if the ratio θ^* of video data transferred over the Wi-Fi direct link over the data transferred over the Wi-Fi hotspot satisfies

$$X_{\text{adsl}} = \frac{p}{T(p, R_{\text{hotspot}}) + \theta^* \cdot T(p, R_{\text{direct}})} . \tag{1}$$

The last equation holds for the following reason: Because $R_{\text{hotspot}} > R_{\text{direct}}$, we will have $T(p, R_{\text{hotspot}}) < T(p, R_{\text{direct}})$. Hence, for $\theta > \theta^*$, the Wi-Fi link becomes a bottleneck and the ADSL backhaul is under-utilized. On the other hand, for $\theta < \theta^*$ the ADSL backhaul becomes the bottleneck and the Wi-Fi network is under-utilized.

To apply equation (1) in a practical setting we consider the approximation[3] $p/T(p, R_{\text{hotspot}}) \approx X_{\text{hotspot}}$, where X_{hotspot} is

[2]Note that, although in our discussion and experiments we consider an ADSL backhaul, the key conclusions of this paper hold for other variants of xDSL, since the throughput of the newer 802.11n and 802.11ac protocols is typically higher than the throughput of xDSL links.

[3]The 802.11 physical and MAC layer overhead can be analytically computed [6]. In our scenario, since video data is transferred over TCP, acknowledgment packet in addition to data packets are transmitted over the Wi-Fi link, hence both would need to be considered in a detailed analytical expression. The proposed approximation using the achieved throughput approximately captures the influence of both data and acknowledgement packets.

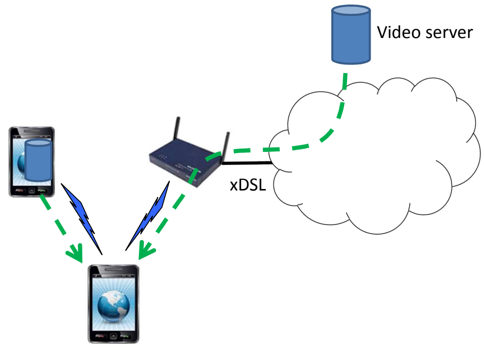


Fig. 3. Video transfer over Wi-Fi hotspot and Wi-Fi direct links.

the throughput achieved when only the Wi-Fi hotspot link is used. Similarly, we consider $p/T(p, R_{\text{direct}}) \approx X_{\text{direct}}$, where X_{direct} is the throughput achieved when only the Wi-Fi direct link is used. Considering the above, (1) gives the following approximation for θ^*:

$$\theta^* = \frac{\frac{1}{X_{\text{adsl}}} - \frac{1}{X_{\text{hotspot}}}}{\frac{1}{X_{\text{direct}}}} . \tag{2}$$

The multi-source streaming client downloads video chunks in rounds. In the beginning of each round the number of chunks that will be downloaded from different sources can be computed based on various policies. For example, load balancing across multiple source and multipe paths is implemented if the number of chunks downloaded from each source is proportional to the measured throughput from the source [2]. To implement the proposed procedure for efficient utilization of multirate Wi-Fi links, (2) is used to estimate the number of chunks that will be downloaded from each source, in each round.

V. Experiments

In this section we present experiments showing the gains for multi-source video streaming when jointly using prefetching and Wi-Fi direct communication, and when the procedure for efficient utilization of multirate Wi-Fi links is used.

The video used in the experiments was a 596 second clip from Big Buck Bunny, encoded at 1280x720 and with an average rate of approximately 1.65 Mbps. The video was segmented into 1224 chunks, each with size approximately 97 KBytes. In the experiments the multi-source mobile video streaming client was running on a Galaxy S3 smartphone with Android 4.4.2. The video server in the experiments with Wi-Fi direct was running on a Galaxy Ace 4 with Android 4.4.4. The video and cache servers were running on two virtual machines with Ubuntu 14.04, in a workstation with VirtualBox 4.3.18.

A. *Prefetching and Wi-Fi direct*

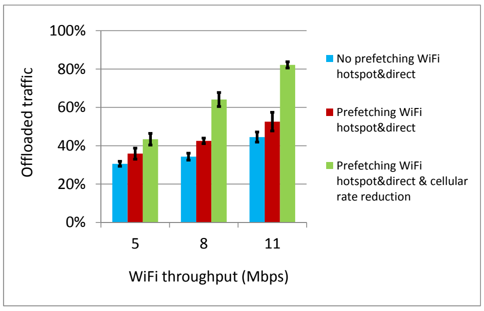
The first set of experiments involves 6 Wi-Fi hotspots and/or Wi-Fi direct devices with the requested video file, which the mobile encounters at time 0, 100, 200, 300, 400, and 500 seconds. Specifically, in the experiments with Wi-Fi direct, there are two Wi-Fi direct encounters and 4 Wi-Fi hotspots: the mobile encounters a neighboring device through which it downloads parts of the video at time 100 and 300 seconds, while at time 0, 200, 400, and 500 seconds the mobile encounters a Wi-Fi hotspot. On the other hand, in the experiments with only Wi-Fi hotspots, hotspots are encountered at times 0, 100, 200, 300, 400, and 500 seconds. The mobile is able to download video data in each hotspot or from the neighboring device with Wi-Fi direct for a duration of 20 seconds; note that prefetching cannot be performed in the first hotspot, since the experiments begin when the mobile is in the first hotspot. Each Wi-Fi hotspot has an emulated ADSL backhaul with throughput 3 Mbps.

We assume that there is some randomness in the time a hotspot or Wi-Fi direct device is encountered and in the throughput of each segment. We consider a time and throughput variability equal to 2% and 5%, respectively. A 5% variability for throughput 1 Mbps means that the actual throughput is randomly selected from the interval $[950, 1050]$ Mbps, hence there is a mismatch between the predicted and actual throughput. The results we present show the average of five runs and the corresponding 95% confidence interval.

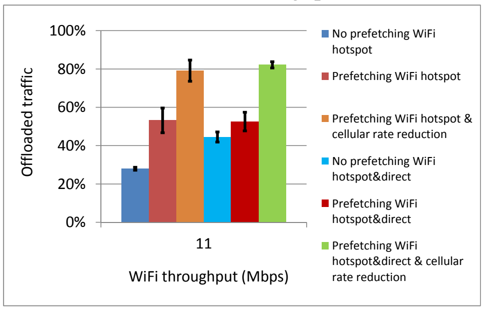
1) Percentage of offloaded traffic: Figure 4(a) shows the percentage of video traffic offloaded to Wi-Fi for different values of the Wi-Fi throughput; note that the Wi-Fi throughput value is the throughput for both the Wi-Fi hotspot and Wi-Fi direct link. Results are shown for three schemes: 1) no prefetching (i.e., when the mobile enters a hotspot the video is downloaded from a remote server using the maximum ADSL bauckhaul throughput), 2) prefetching and downloading of video data over the mobile network at the maximum rate, and 3) prefetching and downloading video data over the mobile network at a smaller rate using Algorithm 1. Observe that the percentage of offloaded traffic with prefetching increases when the Wi-Fi throughput increases, verifying that prefetching but also Wi-Fi direct can utilize the higher Wi-Fi throughput. Without prefetching the percentage of offloading increases slower due to Wi-Fi direct only, since for the path through the Wi-Fi hotspot the ADSL backhaul is the bottleneck. Also, a higher percentage of offloading is achieved with prefetching and reduction of the mobile throughput using Algorithm 1.

Figure 4(b) considers Wi-Fi throughput 11 Mbps and compares the case with 6 Wi-Fi hotspots (i.e., no Wi-Fi direct) and the case with 2 Wi-Fi direct links and 4 Wi-Fi hotspots. Observe that the offloading percentage for the prefetching and prefetching with cellular rate reduction are the same for both cases. When there is no prefetching, the performance in the scenario which includes Wi-Fi direct is higher; this is because when there is no prefetching the Wi-Fi hotspot is limited by the ADSL backhaul, which is not the case when video is downloaded from a neighboring device with Wi-Fi direct.

2) Video QoE: Next we investigate the improved QoE that can be achieved with prefetching and Wi-Fi direct, in terms of fewer video frame pauses. Figure 5(a) shows that the gains in terms of fewer pauses is higher when the mobile throughput is smaller; this is expected since more frame pauses occur when the mobile throughput is smaller, which is when the higher throughput of Wi-Fi can be utilized with prefetching



(a) Wi-Fi throughput



(b) $R_{\text{Wi-Fi}} = 11$ Mbps

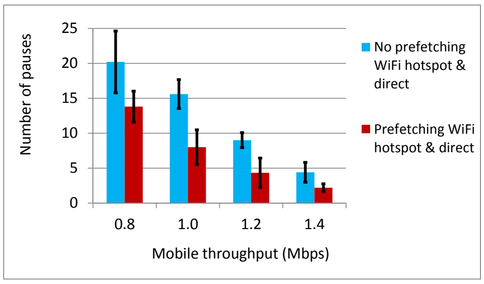
Fig. 4. Mobile video offloading. $R_{\text{mobile}} = 2$ Mbps, $R_{\text{adsl}} = 3$ Mbps.

to download more video data and avoid frame pauses; on the other hand, when prefetching is not used, while in a hotspot the video downloading rate is constrained by the ADSL throughput. Note that in the scenarios of this subsection traffic is downloaded over the mobile network using the maximum throughput, hence we do not differentiate between the two prefetching schemes considered previously.

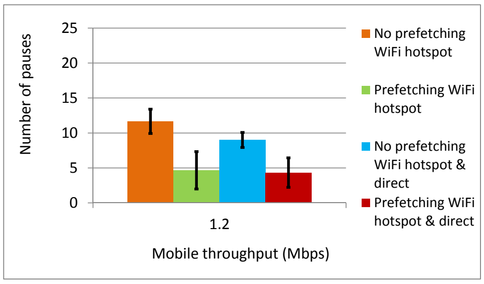
Figure 4(b) considers mobile throughput 1.2 Mbps and compares the case with 6 Wi-Fi hotspots (i.e., no Wi-Fi direct) and the case with 2 Wi-Fi direct connections and 4 Wi-Fi hotspots. The performance when there is prefetching is the same for the two cases. On the other hand, the number of pauses without prefetching is higher in the case of 6 Wi-Fi hotspots, compared to the case where there are 2 Wi-Fi direct encounters and 4 Wi-Fi hotspots; this is because the download throughput for Wi-Fi hotspots is constrained by the ADSL throughput, which is not the case for Wi-Fi direct transfers.

B. Efficient utilization of multirate Wi-Fi links

Next we evaluate the procedure presented in Section IV-B to efficiently utilize two Wi-Fi links with different transmission rates. To perform controlled experiments we consider a topology where two Raspberry Pi (RPi) computers are connected to a smartphone, using the latter's tethering function. The RPi's are used in order to control 802.11's transmission rate, since such control is not possible with an Android device. This scenario emulates the topology shown in Figure 3: one RPi emulates the connection of a Wi-Fi direct link from a smartphone with the requested video to the smartphone running the multi-source mobile video streaming client; we set the transmission rate of this RPi to 1 Mbps. The other RPi emulates the connection of the smartphone to the Wi-Fi



(a) Mobile throughput



(b) $R_{\text{mobile}} = 1.2$ Mbps

Fig. 5. Mobile data QoE. $R_{\text{Wi-Fi}} = 8$ Mbps, $R_{\text{adsl}} = 3$ Mbps.

hotspot; we set the transmission rate of this second RPi to 11 Mbps. We consider different ADSL throughput values.

Figure 6 shows the video QoE when the throughput across the two links is the following: 1) based on the multirate sharing estimation described in Section IV-B, 2) using solely the (emulated) Wi-Fi hotspot link, and 3) using the same throughput on both the (emulated) Wi-Fi hotspot and Wi-Fi direct links. The results show that the proposed multirate sharing procedure achieves significantly better QoE, in terms of the fewer video pauses: the reduction is more than 45% for ADSL throughput 1 Mbps, more than 60% for 1.2 Mbps, and more than 75% for 1.4 Mbps. Moreover, as expected, the number of pauses are fewer as the ADSL throughput increases.

VI. Conclusions and Future Work

We have presented a testbed evaluation of multi-source mobile video streaming that jointly exploits enhanced offloading by prefetching video data in local hotspot caches and device-to-device communication. Future work includes extending our framework for efficient utilization of multirate Wi-Fi links to consider scenarios with 3G/4G connectivity and exploiting adaptive video streaming.

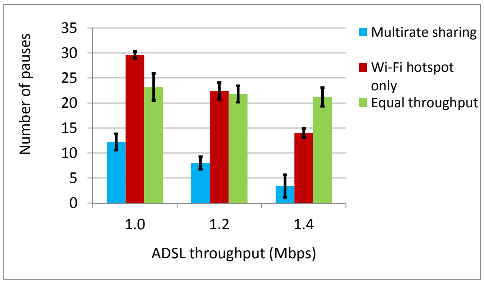


Fig. 6. QoE for video transfer over multirate Wi-Fi links.

References

[1] P. Deshpande, A. Kashyap, C. Sung, and S. Das. Predictive Methods for Improved Vehicular WiFi Access. In *Proc. of ACM MobiSys*, 2009.

[2] D. Dimopoulos, C. Boursinos, and V. A. Siris. Multi-Source Mobile Video Streaming: Load Balancing, Fault Tolerance, and Offloading with Prefetching. In *9th Int'l Conference on Testbeds and /research Infrastructures for the Development of Networks and Communities (Tridentcom)*, 2014.

[3] Y. Ding, Y. Yang, and L. Xiao. Multi-Path Routing and Rate Allocation for Multi-Source Video On-Demand Streaming in Wireless Mesh Networks. In *Proc. of IEEE INFOCOM*, 2011.

[4] K. Evensen, D. Kaspar, C. Griwodz, P. Halvorsen, A. F. Hansen, and P. Engelstad. Improving the Performance of Quality-Adaptive Video Streaming over Multiple Heterogeneous Access Networks. In *Proc. of ACM Multimedia Systems*, 2011.

[5] K. Evensen, A. Petlund, H. Riiser, P. Vigmostad, D. Kaspar, C. Griwodz, and P. Halvorsen. Mobile Video Streaming Using Location-Based Network Prediction and Transparent Handover. In *Proc. of ACM NOSDAV*, 2011.

[6] X. Fafoutis and V. A. Siris. Handover incentives for self-interested wlans with overlapping coverage. *IEEE Transactions on Mobile Computing*, 11(12):2033–2046, 2012.

[7] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire. FemtoCaching: Wireless Video Content Delivery through Distributed Caching Helpers. In *Proc. of IEEE Infocom*, 2012.

[8] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, and A. Markopoulou. MicroCast: cooperative video streaming on smartphones. In *Proc. of ACM MobiSys*, 2012.

[9] S.-B. Lee, A. Smeaton, and G.-M. Muntean. Quality-oriented multiple-source multimedia delivery over heterogeneous wireless networks. *IEEE Transactions on Broadcasting*, 57(2):216–230, 2011.

[10] F. Malandrino, M. Kurant, A. Markopoulou, C. Westphal, and U. C. Kozat. Proactive Seeding for Information Cascades in Cellular Networks. In *Proc. of IEEE Infocom*, 2012.

[11] A. J. Nicholson and B. D. Noble. BreadCrumbs: Forecasting Mobile Connectivity. In *Proc. of ACM Mobicom*, 2008.

[12] U. R. K. P. M. Eittenberger, M. Herbst. RapidStream: P2P Streaming on Android. In *Proc. of 19th IEEE Intl Packet Video Workshop*, 2012.

[13] T. Schierl, K. Ganger, C. Hellge, T. Wiedand, and T. Stockhammer. SVC-based Miltisource Streaming for Robust Video Transmission in Mobile Ad hoc Networks. *IEEE Wireless Communications*, pages 96–103, October 2006.

[14] V. Singh, J. Ott, and I. Curcio. Predictive Buffering for Streaming Video in 3G Networks. In *Proc. of IEEE WoWMoM*, 2012.

[15] V. A. Siris and M. Anagnostopoulou. Performance and Energy Efficiency of Mobile Data Offloading with Mobility Prediction and Prefetching. In *Proc. of IEEE Workshop on Convergence among Heterogeneous Wireless Systems in Future Internet (CONWIRE), co-located with IEEE WoWMoM*, 2013.

[16] V. A. Siris, M. Anagnostopoulou, and D. Dimopoulos. Improving Mobile Video Streaming with Mobility Prediction and Prefetching in Integrated Cellular-WiFi Networks. In *10th Int'l Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous)*, 2013.

[17] V. A. Siris and D. Kalyvas. Enhancing Mobile Data Offloading with Mobility Prediction and Prefetching. In *Proc. of ACM MOBICOM MobiArch Workshop*, 2012.

[18] W. Song and W. Zhuang. Performance analysis of probabilistic multipath transmission of video streaming traffic over multi-radio wireless devices. *IEEE Transactions on Wireless Communications*, 11(4):1554–1564, April 2012.

[19] G. Xylomenos, X. Vasilakos, C. Tsilopoulos, V. A. Siris, and G. C. Polyzos. Caching and Mobility Support in a Publish-Subscribe Internet Architecture. *IEEE Comm. Mag.*, 50(7):128–136, 2012.

[20] J. Yao, S. S. Kahnere, and M. Hassan. An Empirical Study of Bandwidth Predictability in Mobile Computing. In *Proc. of ACM WinTech*, 2008.

[21] J. Yao, S. S. Kahnere, and M. Hassan. Quality Improvement of Mobile Video Using Geo-intelligent Rate Adaptation. In *Proc. of IEEE WCNC*, 2010.