Eittenberger, Philipp M.; Hamatschek, Michael; Großmann, Marcel; Krieger, Udo R.

Monitoring Mobile Video Delivery to Android Devices

Date of secondary publication: 15.04.2026

Accepted Manuscript (Postprint), Conferenceobject

Persistent identifier: urn:nbn:de:bvb:473-irb-114732x

Primary publication

Eittenberger, Philipp M.; Hamatschek, Michael; Großmann, Marcel; Krieger, Udo R. (2013): Monitoring Mobile Video Delivery to Android Devices, in: Carsten Griwodz (Ed.), MMSys ’13: Proceedings of the 4th ACM Multimedia Systems Conference, New York, NY, USA: ACM, pp. 119–124, doi: 10.1145/2483977.2483992.

Monitoring Mobile Video Delivery to Android Devices

Philipp M. Eittenberger Michael Hamatschek Marcel Großmann Udo R. Krieger

Faculty of Information Systems and Applied Computer Science
Otto-Friedrich University
Bamberg, Germany
{firstname.lastname}@uni-bamberg.de

ABSTRACT

The proliferation of smart devices for mobile networks is a major traffic generator nowadays. These devices provide the ability to receive media content in nearly every situation. Despite that video streaming in high quality is getting more and more popular in mobile scenarios, the performance and bottlenecks of mobile applications over wireless networks, especially, during the transmission of media streams, are poorly understood yet. In order to tackle this new challenge, we present an Android based framework to capture the relevant wireless network behavior, geo-coordinates and packet traces for popular streaming applications on Android certified devices. A dataset has been obtained by measurement trials, which have been performed in a 3G network for both HTTP and peer-to-peer video streaming applications. The trials comprise also an additional WiFi measurement for comparison purposes. The presented dataset enables future research to determine the quality of service and network characteristics of different streaming methodologies, which are affected by the typical conditions encountered in wireless networks, like hand-over effects, signal fading, connection losses etc. We hope that both, the presented dataset and the framework, may prove to be useful for the traffic measurement and the multimedia research communities.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms

Measurement, Performance

Keywords

Android, Mobile Video, Traffic Measurement

1. INTRODUCTION

Nowadays, people use smart phones and tablets to access media services in mobile scenarios. Due to the mobility of these devices, two major problems arise. On the one hand, there is the battery drainage of mobile devices and on the other hand, there is the need for continuous connectivity and sufficient bandwidth. The power consumption is influenced by high resolution displays of new mobile devices and the demand to watch videos in high quality. Moreover, high quality videos cause a higher amount of necessary downlink capacity, which is also increasing the power consumption of the device. To provide deeper insights into these new challenges, we have performed measurements with different media players and different dissemination approaches. Since Android is a popular open source operating system for mobile devices and due to its steadily increasing market share, it is in the sole focus of our work. However, the gained insights might also be found valid for other smart device eco systems like iOS. The measurements themselves are performed with an in-house developed Android based framework, called *KTR-MobileObserver*. This framework provides the functionality to log information w.r.t. the mobile network, current geo-location coordinates and battery statistics. Additionally, packet traces have been captured by *tcpdump* to scrutinize the behavior of current streaming applications in wireless and cellular networks.

Two different applications have been investigated in our measurement campaign: *YouTube*, which uses HTTP streaming, and *SopCast*, which is a popular representative of a peer-to-peer (P2P) live streaming application. The selection of the used streaming protocols is based on the popularity and availability of Android apps supporting these protocols. In particular, the choice of the investigated applications is motivated as follows. HTTP Streaming is selected for its widespread use in todays Internet streaming services and YouTube is the world's largest online video content delivery network (CDN). SopCast is a representative of a P2P live streaming application and has, at least in the "wirelined" Internet, a large user community.

Current research investigates many aspects of the mobile video challenge. Among many others there is, for instance, the work of Huang *et al.* [7], which developed *3GTest*, a cross platform measurement tool for mobile devices to investigate the energy consumption of these devices during media streaming. Liu *et al.* [8], [9] examined the power consumption of wireless network interfaces (WNIC) during the reception of video streams. They observed that battery consumption is related to the streaming quality, in particular

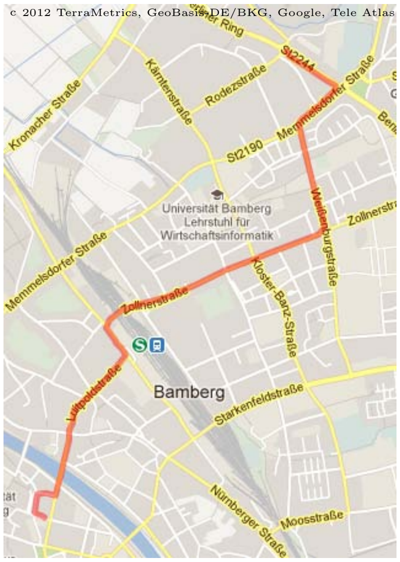


Figure 1: Track of the measurements

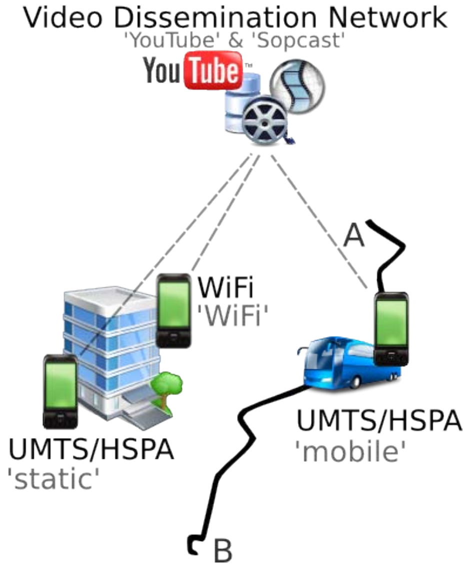


Figure 2: Measurement setup

to the power save mode of the WNIC, which is affected by different streaming protocols and different dissemination architectures, i.e., Client-Server, Client-Proxy-Server or P2P. Additionally, they have taken the battery consumption of the CPU and the display into account and analyzed how much data is transmitted by the investigated applications. Eittenberger, Schneider and Krieger [5] have developed a framework to perform mobile measurements, e.g., of mobile video streaming. This framework was developed for laptops connected to wireless networks and includes the usage of a dedicated GPS tracking device.

This paper shall enable further performance evaluation and traffic analysis of popular video streaming approaches on mobile devices in wireless networks. The measurement on the mobile device provides a genuine insight into the behavior of the used applications and the corresponding protocols. We believe that the analysis of the presented data set can provide new insights on these applications in the wireless, mobile scenario. However, even the preliminary analysis has already revealed some interesting findings which are as follows: The YouTube application uses a progressive download approach to provide streaming content for clients connected via a cellular network and segment based streaming for WiFi connected devices. The interaction of mobile devices in P2P networks is reduced to the functionality of a leecher, i.e., mobile devices are not seeding content to other peers. Further evaluation of the available data set can provide a deeper insight into the streaming process, e.g., how do handovers affect mobile media streaming performance and how well adapt these streaming approaches w.r.t. the new requirements of user mobility in cellular networks. Furthermore, by the inclusion of the complete packet traces, the data set provides a valuable contribution for the teletraffic modeling community, since it allows a detailed analysis of the packet reception process at the network edge. In the following, a brief overview of the investigated applications of this measurement campaign is provided.

1.1 HTTP Streaming

HTTP streaming delivers video content with TCP as transport protocol from video server(s) to the requesting client. Ma *et al.* [10] describe three HTTP transmission paradigms, which will be shortly outlined: First, a *straight download* mechanism is considered, where in return to one `GET`-request the video stream is transmitted progressively. Second, in a *segment download*, pieces of the video are sent as a response to subsequent `GET`-requests from the client. In the third case the transmission of the video content is paced, but only the server handles the rate adaption without the necessity of additional requests from the client. Additionally, to decrease the playback latency at the client side, most of these paradigms use a buffering mechanism after the request for a video or a video segment. HTTP streaming is nowadays the de facto standard to stream video content due to the omnipresence of HTTP, the large deployed delivery infrastructure and its firewall and NAT traversal capabilities. In our study we investigate HTTP streaming on mobile devices via the native YouTube app for Android. Finnamore *et al.* [6] did already analyze the video download mechanisms of the mobile YouTube website, but they did not investigate the mobile YouTube app.

1.2 Peer-to-Peer Streaming

P2P streaming networks are overlay data dissemination networks. Each participant, called *peer*, can be server and client at the same time. Peers are downloading segmented streaming content and are uploading this content again to supply the content to other peers. Therefore, peers are sending requests for media chunks to their neighboring peers. Those networks have proved to be scalable for file-sharing and media streaming. In our examination we will investigate SopCast [1], one of the most prominent P2P networks to broadcast and watch live video streaming, especially live sport events. SopCast provides a native Android app, which allows the device to participate in the dissemination network. For this purpose it uses a proprietary P2P streaming protocol that transmits the video content via UDP.

Type	**Mobile 1.1**	**Static 1.1**	**WiFi 1.1**	**Mobile 1.2**	**Static 1.2**	**WiFi 1.2**
Start Time	11:25:45	11:25:29	11:25:14	11:40:54	11:41:27	11:40:59
Capture Time	12:13 min.	12:42 min.	12:50 min.	6:39 min.	10:26 min.	10:54 min.
Bytes	24,308,867	18,761,349	29,960,350	41,904,302	45,040,066	129,177,352
Bytes In	23,674,008	18,474,903	29,313,700	41,053,388	44,526,854	126,712,387
Bytes Out	634,859	286,446	646,650	850,644	513,212	2,464,965
MBit per Sec.	0.265	0.197	0.311	0.838	0.575	1.578
MBit per Sec. In	0.258	0.194	0.304	0.822	0.569	1.560
MBit per Sec. Out	0.007	0.003	0.007	0.017	0.007	0.030
Packets	25,662	18,701	28,798	41,165	40,395	120,484
Packets In	16,349	13,611	19,892	28,728	31,451	84,936
Packets Out	9,313	5,090	8,906	12,437	8,944	35,548
Bytes per packet	947.271	1,003.227	1,040.362	1,017,953	1,114.991	1,072.154
Bytes per packet In	1,448.040	1,357.351	1,473.643	1,429.037	1,415.753	1,491.857
Bytes per packet Out	68.169	56.276	72.608	68.396	57.381	69,342

Table 1: Measurement runs 1.1 and 1.2 (YouTube)

Type	**Mobile 2.1**	**Static 2.1**	**WiFi 2.1**	**Mobile 2.2**	**Static 2.2**	**WiFi 2.2**
Start Time	12:10:56	12:09:30	12:09:37	12:24:01	12:24:10	12:23:48
Capture Time	09:27 min.	11:00 min.	10:46 min.	10:39 min.	11:59 min.	12:22 min.
Bytes	69,412,994	72,275,921	134,778,852	67,498,025	64,267,640	272,686,220
Bytes In	66,435,656	67,697,495	94,958,322	64,667,106	60,173,774	109,024,213
Bytes Out	2,977,338	4,578,426	39,820,530	2,830,919	4,093,866	163,662,007
MBit per Sec.	0.978	0.876	1.667	0.845	0.715	2.939
MBit per Sec. In	0.936	0.820	1.175	0.809	0.669	1.175
MBit per Sec. Out	0.042	0.055	0.493	0.040	0.046	1.764
Packets	85,475	113,597	212,094	86,082	107,360	405,961
Packets In	56,314	59,809	108,313	55,856	55,177	198,538
Packets Out	9,313	53,788	103,781	30,226	52,183	207,423
Bytes per packet	812.085	636.248	635.468	784.113	598.618	671.705
Bytes per packet In	1,179.736	1,131,895	876,703	1,157.747	1,090.559	549.135
Bytes per packet Out	102.100	85.120	383.698	93.658	78.452	789.025

Table 2: Measurement runs 2.1 and 2.2 (SopCast)

2. METHODOLOGY

All mobile measurements have been carried out in the 3G UMTS/HSPA network of T-Mobile, Germany. The measurements were performed with three different settings. First, a *mobile* measurement was performed during a bus ride between two final stops of an inner city route in Bamberg (see Figure 1). The track has a length of 3.6 kilometers and the maximum speed was 50km/h. Additionally, two stationary measurements have been performed in a university office. These local measurements consist of a 3G connected *static* measurement and a *WiFi* measurement, which was conducted in a 802.11n WiFi network. The measurement setup is coarsely depicted in Figure 2.

Each measurement consists of the traces of three devices and each includes two measurement runs. In the mobile measurements the first run started at location 'A' and stopped at location 'B' (see Figure 2) and the second was performed in the reverse direction. The mobile run is realized in an urban environment to obtain realistic mobility aspects. To keep the environmental influences stable, especially to avoid interference by movement, the static measurement was performed to enable a comparison with the mobile measurement. The WiFi measurement is used to examine quality-of-service aspects and to reveal differences of the used applications in relation to the used access network. Two measurements have been performed for each application to compensate for abnormalities, like, e.g., a temporal lack of network coverage.

The experiments have been carried out with three off-the-shelf Android tablets and their built-in network interfaces. For the mobile measurement a Huawei S7-301u MediaPad running Android 4.0 was used. The static measurement was performed with a Samsung Galaxy Tab GT-P1000 running Android 2.1. The third device for the WiFi measurement was a Google Nexus 7 running Android 4.1. The device specific hardware, like RAM and CPU, does not matter regarding the evaluation of applications' streaming behavior.

To capture device and application specific information, our measurement app KTR-MobileObserver was used to obtain network information of the wireless networks, current geolocation coordinates and further network traffic information. For every measurement interval (in this case 10 seconds) KTR-MobileObserver logs different parameters of the device, e.g., geocoordinates, cell id, location area code, signal strength, network type, battery statistics etc. An exemplary excerpt of a KTR-MobileObserver log file is shown in Table 3. The functionality of KTR-MobileObserver provides the ability to capture data for one specified application running on the mobile device. Thus, only the behavior and data of the specific application is observed during the particular measurement. The captured information is stored as a CSV file into a user-defined folder on the device. Additionally, the network traffic of the application has been captured by tcpdump. The captured traces are used for the evaluation of the application's streaming behavior. Furthermore, by relating the captured packet traces and KTR-MobileObserver's logs, influences on streaming quality caused by network effects, like handovers, signal fading etc., can be evaluated. To yield a deeper insight on the battery consumption of mobile applications, the implementation of KTR-MobileObserver could be extended with further functionality. Therefore, the inclusion of PowerTutor [11] into our framework was investigated. However, we found that PowerTutor is providing only marginal support for different hardware vendors respectively for various hardware platforms. The best way to obtain a fine grained battery consumption of a particular application would be given by using the Android API. However, the API methods to access the battery statistics are only cryptically documented yet and therefore, they are not usable at the time of the study. KTR-MobileObserver will be released to the public in addition to the captured packet traces[1].

---

[1] Both can be downloaded at `http://www.ktr.uni-bamberg.de/project/mobileTV.html`.

Date	Latitude	Longitude	# of Updates	Operator ID	Operator Name	GSM RSS	Cell ID	LAC ID	Network Type
2012-11-09 12:10:49	no signal	no signal	1	26201	Telekom.de	−73	2712901	17559	UMTS
2012-11-09 12:10:59	49.89327453	10.89270941	2	26201	Telekom.de	−75	2712901	17559	UMTS
2012-11-09 12:11:09	49.8932118	10.89294955	4	26201	Telekom.de	−67	2712901	17559	UMTS
2012-11-09 12:11:19	49.89392365	10.89313584	6	26201	Telekom.de	−71	2712901	17559	UMTS
2012-11-09 12:11:29	49.89479142	10.89367843	8	26201	Telekom.de	−77	2712901	17559	UMTS
2012-11-09 12:11:39	49.89547501	10.89390839	9	26201	Telekom.de	−79	2712901	17559	UMTS
2012-11-09 12:11:49	49.89595606	10.89401208	10	26201	Telekom.de	−81	2712901	17559	UMTS
2012-11-09 12:11:59	49.89601317	10.89399404	10	26201	Telekom.de	−81	2712901	17559	UMTS
2012-11-09 12:12:09	49.89600424	10.89399	10	26201	Telekom.de	−81	2712901	17559	UMTS
2012-11-09 12:12:19	49.895995	10.89398961	12	26201	Telekom.de	−81	2712901	17559	UMTS
2012-11-09 12:12:29	49.89610917	10.89400788	13	26201	Telekom.de	−83	2712901	17559	UMTS
2012-11-09 12:12:39	49.89666433	10.89408863	13	26201	Telekom.de	−83	2712901	17559	UMTS
...									

Table 3: Excerpt of an exemplary log file of KTR-Mobile Observer: 3G Mobile Sopcast - Measurement 2.1

3. TRACE FORMATS

The presented data set comprises three different compressed folders, one for each network type, containing the captured pcap files and the log files obtained by KTR-Mobile-Observer[2] for each measurement run. The full pcap packet traces have been included to enable further analysis by other network analysis tools, like Wireshark or Atheris [4] which rely on the pcap format as input data. The inclusion of the full pcap files enables also additional traffic classification analysis that relies on deep packet inspection. To avoid side effects in the evaluation of the packet traces all "cross traffic" caused by other applications was removed. To filter the "cross traffic" for the YouTube measurement traces, all packets sent or received by IPs outside of YouTube's AS are filtered. SopCast's port number on the client devices was 4001, therefore, all packets not sent or received at port 4001 have been discarded too.

4. TRAFFIC MEASUREMENT AND ANALYSIS

4.1 YouTube Measurements

In this section we discuss the YouTube measurements to provide insights into mobile HTTP streaming. The key statistics of the YouTube measurements are given in Table 1. The average duration of each measurement run is approximately 10 minutes with one exception, mobile measurement 1.2 was affected by a connection loss in the city center of Bamberg. It is noticeable that the 3G connected devices receive a similar amount of packets during all measurements. The corresponding throughput statistics are visualized in Figure 3. In the graphs the throughput over time is visualized. The upper graph shows the outbound data-rate sent and the lower one all inbound data retrieved by the client in KBit/s. In the graph of the mobile measurement handovers in connection with a switch of the network type are visualized (light blue for EDGE and no color for UMTS). In both HTTP measurement trials no video artifacts or playback interruptions have been observed.

4.1.1 Measurement 1 & 2

Measurement 1.1 was performed with a low quality YouTube video [2] without the ability to adapt the streaming quality. The first run shows an higher amount of sent and received packets (cf. Table 1) and a higher throughput of the mobile and WiFi measurements in comparison to the corresponding static measurement. This observation can be explained by an automatic adaption of video resolution, which is adjusted to the display resolution of the particular device. The display resolution of the device used for the static measurements is 1024 x 600 pixel versus a resolution of 1280 x 800 pixel of the mobile and WiFi device. Another possible reason could be given by the different Android versions running on the test devices and the associated YouTube application. To exclude effects of the network type (e.g. UMTS or HSPA), the captured log files of KTR-MobileObserver were also inspected. The comparison of the throughput statistics shows similar patterns in the mobile and WiFi traffic.

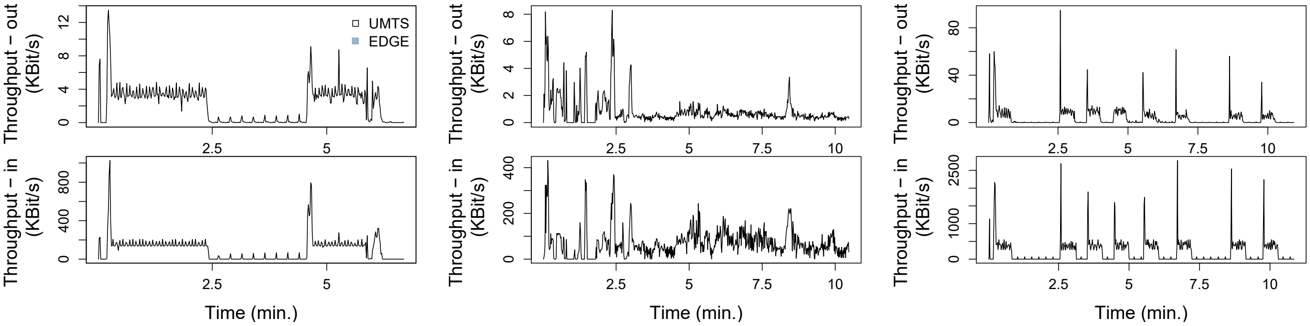
To investigate the behavior of the devices in relation to a higher quality of service, the second measurement (1.2) was performed with a YouTube video in HD resolution [3]. Clients in the 3G network are automatically throttled to a low quality resolution in comparison to the WiFi client. By this mechanism it is possible to save download capacity and to reduce the battery consumption of the mobile devices. The duration of the mobile measurement run is shorter than the static one, but under consideration of the video content buffering, the measurement period can be neglected in this case. The mobile and WiFi measurements are showing similar packet transmission patterns during the video content transmission in high quality, although the WiFi connected device is receiving a higher number of packets. This observation leads to the conclusion that the YouTube application adapts the streaming quality according to the network connection. The steep throughput increase at the beginning of the streaming process due to a pre-loading phase of the video content is common to all measurements. The partial download mechanism, which Finamore *et al.* [6] observed with the mobile YouTube website, could not be confirmed for the Android YouTube app. In fact, in most measurement runs only one `GET`-request for the video data is exchanged. Furthermore, the packet inspection of measurement run 1.1 and 1.2 shows that low quality video content is progressively downloaded and high quality media is streamed via a segment-based approach, where the segments are distributed among a few video servers. This statement is supported by a comparison of the visualized throughput rates (cf. Figure 3) and an inspection of the intensity rates of exchanged packets.

4.2 SopCast Measurements

To provide an overview of the SopCast measurement runs, the network statistics are shown in Table 2 and the corresponding throughput of inbound and outbound packets are visualized in Figure 4. The metrics of the SopCast measurements are similar for the mobile and static measure-

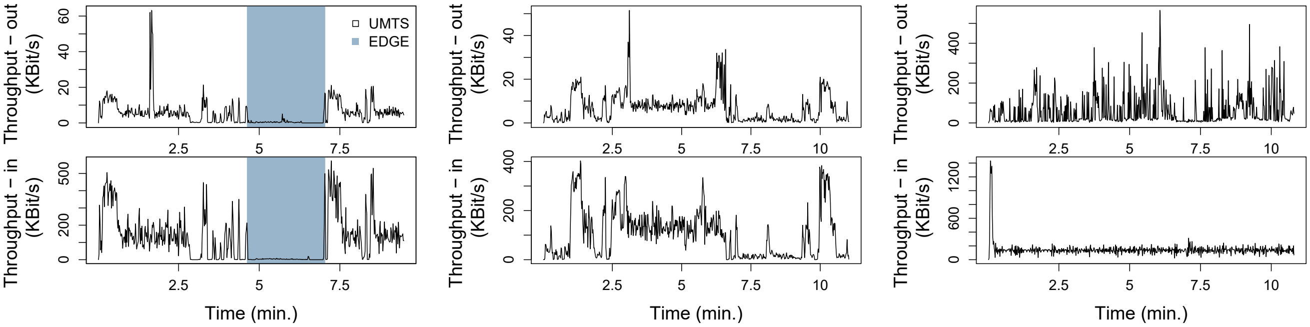
---

[2] It has to be noted that w.r.t. the KTR-MobileObserver log files, the Huawei MediaPad reports always UMTS as network type, even when it most probably uses the HSPA network.



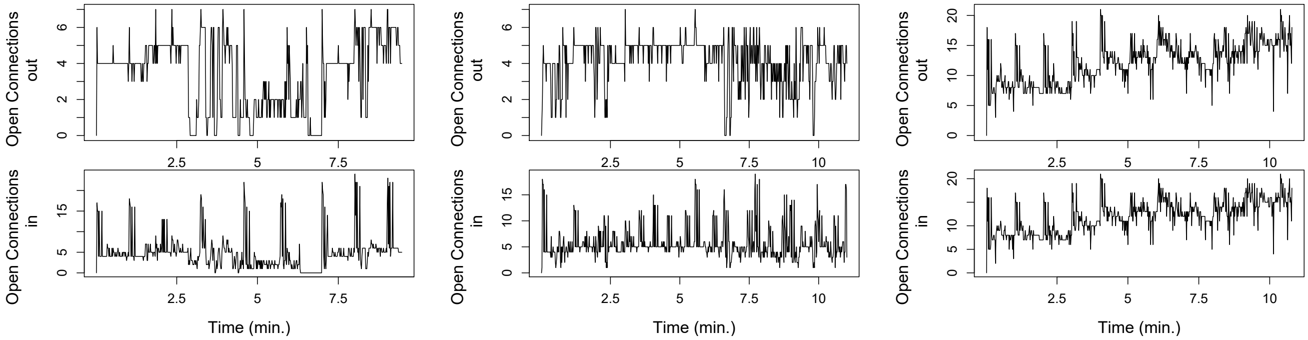
(a) mobile (b) static (c) WiFi

Figure 3: YouTube measurement 1.2



(a) mobile (b) static (c) WiFi

Figure 4: SopCast measurement 2.1



(a) Mobile (b) Static (c) WiFi

Figure 5: Open connections for SopCast measurement 2.1

ments. Thus, we do not see indications for a quality adaption to the particular hardware of the device. However, the type of the used network is affecting the behavior of the SopCast application. In the 3G measurements the devices (mobile/static) were affected by a breakdown of the connection between the mobile device and the P2P network (compare with Figure 4a). During the video playback in the 3G measurements buffering delays, video artifacts and stalling effects have been observed. User experience is seriously suffering by such a quality degradation. Both WiFi measurements show a buffer phase at the start of the packet stream and afterwards, a constant streaming rate of approximately 140 KBit/s (compare with Figure 4c). The plot of the mobile measurement is also annotated by handovers in combination with switching between network types (i.e. vertical handover). When the device is connected over EDGE, the influence is clearly visible in a steep throughput degradation.

4.2.1 Measurement 1 & 2

The mobile and static measurements show the same noticeable problems. At first, both are affected by interruptions of the P2P network connection. Furthermore, mobile peers only download content and do not seed content to other peers. The analysis of the overall packet length of sent packets shows that nearly all outbound packets are signaling messages, e.g., to request video data or to maintain the P2P overlay. The examination of the WiFi measurement yields another interesting result. WiFi peers are really participating in the P2P data dissemination, i.e., they provide video content for other peers. The observations of measurement 2.1 w.r.t the peers connected via a cellular network can be confirmed by measurement run 2.2. However, in comparison to measurement 2.1, the WiFi peer has a larger upload volume and supplies a larger amount of distinct peers with media content.

4.2.2 Open Connections & Upload Ratio

Figure 5 illustrates the average number of open in- and outbound connections to other peers during the SopCast measurements. In the mobile and static measurements the average number of outbound connections to other peers is about 3-4 peers and video data are received from approx. 5 peers. This stands in contrast to an average number of 10-15 open connections of a WiFi peer. The obtained traces for devices using SopCast over the 3G network (mobile and static) do not differ significantly. In both measurements only a few packets are seeded to other peers in the P2P network by the mobile peers. In general, we found that peers operating in a 3G network did only contribute very little upstream capacity to the P2P network in all our measurements. The same observation can be also confirmed by comparing the total number of outbound packets. This behavior could be intentional to reduce the data usage for 3G subscribers. However, it also shows that by themselves and possibly by design, mobile devices do not establish a fair load sharing in a P2P network. Since modern HSPA networks do not induce such a drastic uplink and downlink asymmetry, this circumstance needs further investigations. However, if this observation holds true, this could grow to a severe problem for current P2P networks by the presumably growing number of mobile peers in the future. Contrary to this observation, devices connected by WiFi participate fairly during the P2P streaming process.

5. CONCLUSION

In this study we presented an Android based measurement app and several packet traces of video streaming applications for Android devices in cellular and wireless networks. The preliminary inspection of the packet traces and log files of KTR-MobileObserver implies different conclusions: First, the observation of YouTube's mobile application shows that the media content is adapted to the display resolution of the remote device and the current Internet connection. An additional observation is that media streaming to 3G connected devices is mainly done with a progressive download pattern. In contrast, WiFi connected devices use a segment-based approach that supports several video servers. Furthermore, the observation of the media streaming via SopCast points out a possible future problem of P2P networks. Mobile clients do not participate fairly in the P2P data dissemination, since video content is only received from other peers and it is not seeded. Further usage of the provided data set might encompass statistical modeling of the teletraffic characteristics. For example, a characterization of the encountered distribution of the video data flows' packet inter-arrival times might be helpful for the provisioning of network capacity and the resource allocation. Moreover, the provided traces might also prove useful for application developers. They can use the data set to extract metrics of interest (e.g. jitter, achievable throughput, latency etc.) to emulate the network behavior encountered in cellular and wireless networks. Furthermore, a more detailed analysis of the captured traces and log files might hopefully help to provide deeper insights on the influence of cellular networks and mobility aspects on the performance of mobile video streaming.

6. ACKNOWLEDGMENTS

The authors are grateful to Dr. Ingo Dahm of the Deutsche Telekom AG for the provided access to the T-Mobile 3G network.

7. REFERENCES

[1] Sopcast. `www.sopcast.com`.

[2] YouTube Video: Defcon 18 Pwned By the owner. `https://www.youtube.com/watch?v=U4oB28ksiIo`.

[3] YouTube Video: Sintel HD. `https://www.youtube.com/watch?v=HomAZcKm3Jo`.

[4] P. Eittenberger and U. Krieger. Atheris: A first step towards a unified peer-to-peer traffic measurement framework. PDP '11, 2011.

[5] P. Eittenberger, K. Schneider, and U. Krieger. Location-aware traffic analysis of a peer-to-peer streaming application in a hspa network. PDP '13, to appear.

[6] A. Finamore, M. Mellia, M. M. Munafò, R. Torres, and S. G. Rao. Youtube everywhere: impact of device and infrastructure synergies on user experience. IMC '11, pages 345–360, 2011.

[7] J. Huang, Q. Xu, B. Tiwana, Z. M. Mao, M. Zhang, and P. Bahl. Anatomizing application performance differences on smartphones. MobiSys '10, pages 165–178, 2010.

[8] Y. Liu, L. Guo, F. Li, and S. Chen. An empirical evaluation of battery power consumption for streaming data transmission to mobile devices. MM '11, pages 473–482, 2011.

[9] Y. Liu, F. Li, L. Guo, and S. Chen. A measurement study of resource utilization in internet mobile streaming. NOSSDAV '11, pages 33–38. ACM, 2011.

[10] K. Ma, R. Bartos, S. Bhatia, and R. Nair. Mobile video delivery with http. *IEEE Communications Magazine*, 49(4):166 –175, 2011.

[11] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. CODES/ISSS '10, pages 105–114, 2010.