Eittenberger, Philipp M.; Schneider, Klaus; Krieger, Udo R.

Location-Aware Traffic Analysis of a Peer-to-Peer Streaming Application in a HSPA Network

Date of secondary publication: 28.04.2026

Accepted Manuscript (Postprint), Conferenceobject

Persistent identifier: urn:nbn:de:bvb:473-irb-114862x

Primary publication

Eittenberger, Philipp M.; Schneider, Klaus; Krieger, Udo R. (2013): Location-Aware Traffic Analysis of a Peer-to-Peer Streaming Application in a HSPA Network, in: Peter Kilpatrick, Peter Milligan, and Rainer Stotzka (Ed.), Proceedings of the 21st Euromicro International Conference on Parallel, Distributed, and Network-based Processing : 27 February-1 March 2013, Belfast, UK, Piscataway: IEEE, pp. 444–448, doi: 10.1109/PDP.2013.72.

Location-Aware Traffic Analysis of a Peer-to-Peer Streaming Application in a HSPA Network

Philipp M. Eittenberger, Klaus Schneider, Udo R. Krieger
Faculty of Information Systems and Applied Computer Science
Otto-Friedrich University,
Bamberg, Germany
Email: philipp.eittenberger@ieee.org

***Abstract*—The amount of mobile video streaming traffic is rapidly growing and a potential candidate to reduce the load on the content delivery infrastructure is given by the peer-to-peer (P2P) paradigm. One might argue about the adequacy of P2P in this mobile context, yet, it has already become reality. Popular P2P streaming networks like SopCast or TVU networks have recently deployed their apps for mobile devices. In this paper the current state of a location-aware measurement framework is presented, which was mainly developed to evaluate P2P systems in cellular networks. We use our generic framework to present the first measurement study that investigates the behavior of the popular P2P video streaming application SopCast when it is operating in a UMTS High Speed Packet Access (HSPA) network and compare the results with reference Ethernet measurements. The aim of this study is to discover the impact of cellular networks and user mobility on the performance of such P2P applications. In particular we could observe that as consequence some of the drawbacks are characterized by poor connectivity among peers and negligible upload contributions.**

I. Introduction

The traffic amount of mobile video streaming is rapidly growing with higher video resolutions on the one hand and a raising amount of mobile devices like notebooks, smart-phones and Tablet PCs on the other. Recently, client applications of popular P2P video streaming services have been published for mobile devices (e.g. TVU-Player for Android & iOS [3] and SopCast for Android [2]). These smart phone client applications (apps) connect to the "normal" P2P networks, i.e. there are no dedicated networks just for such devices. Therefore, they exchange data with other mobile clients, but due to the current supremacy of wired devices in P2P networks they get most of the data from wired peers. Despite the currently provider induced traffic caps on mobile networks, which can be right now quite a show stopper for P2P applications, due to the ongoing advent of ubiquitous computing, users will become more and more accustomed to use their favourite app where ever they are and when ever they want.

Doubtlessly, the traffic of mobile video is going to increase and P2P could still be used to distributed the traffic load more evenly in the network. Yet, P2P video streaming, especially live streaming, introduces a few additional requirements to achieve acceptable video quality and user experience. On the one hand, a minimum average bandwidth is required and on the other, and in a very special way with regard to live streaming, the demand for low delay and hence, low latency emerges. While stationary Ethernet links can provide low-latency and high-bandwidth connections, this may be a problem for mobile applications that are operating in cellular networks with their varying network coverage and link quality.

The migration of P2P applications to a mobile environment imposes new challenges, since mobile devices are mainly battery powered and have, in general, less computing power compared to stationary PCs. In addition, P2P applications running in a mobile environment encounter different network dynamics compared to a "wired" scenario. Although next generation mobile networks strive to provide broadband-like downstream capacity, the intermittent connectivity, handovers and the fluctuating link quality negatively affect the transmission performance of P2P applications.

There are other consequences of cellular networks as well that need to be addressed by future adoptions of the P2P technology: For instance, in terms of energy consumption the uplink traffic is more expensive for the terminal than the downlink traffic. High packet loss and high latencies are prevalent in cellular networks. The heterogeneity of networks and the variety of access technologies, like 2G GPRS networks, EDGE (2.5G), UMTS (3G) or LTE (4G) next generation networks, impose a new requirement on P2P as well. To adapt the P2P data dissemination to cellular networks and user mobility, the protocol of a P2P application should be as energy efficient as possible, avoid too much signalling traffic and, in general, keep the communication overhead at a minimum. Furthermore, user mobility imposes new fluctuating conditions on P2P applications, like fading signals and disruptions occurring during handovers. Thus, their protocols must be able to respond to the intermittent connectivity dynamically. We conclude that important future development objectives of mobile P2P applications are the sustainable usage of network resources, i.e. to be as energy and resource efficient as possible, and the adaptation of the communication protocols to the intermittent connectivity.

To cope with the presented changes we introduce a generic framework to assess common network metrics and to relate them to geo-coordinates. The presented location-aware traffic measurement and analysis proves to be especially useful for the evaluation of mobile applications in cellular networks. In the presented measurement campaign we compare common performance metrics of the popular P2P video streaming

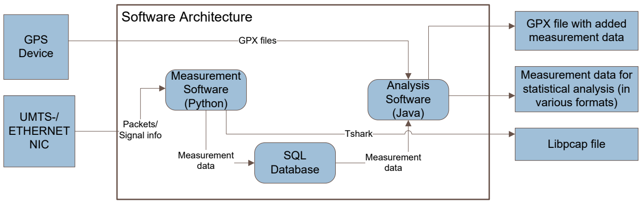

Figure 1. Software Architecture

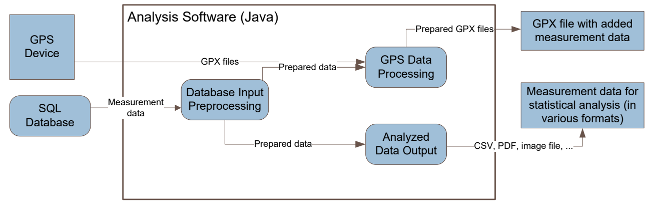

Figure 2. Analysis Software

application SopCast in different scenarios, i.e. with a stationary UMTS HSPA connection, with a mobile UMTS HSPA connection and, in contrast, as a reference with an Ethernet connection.

II. System Architecture

The measurement framework, which is depicted in Figure 1, can be coarsely divided into three parts: The traffic measurement software, which was written in Python, the analysis software implemented in Java and the *SQL*-Database, which provides the connection between the other two parts. The measurement software captures packets from the network interface via the library *libpcap* and requests signal information from the UMTS device. It gathers data by various methods (which will be further elaborated on in Section II-A) and stores them into the SQL-Database. Furthermore, it dumps the packet trace to disk using the libpcap format. The analysis software processes the data from the database, calculates and prepares the performance metrics in human-readable output. In addition, to discover the impact of user mobility we implemented geolocation via GPS. This allows the investigation of other influential factors like travelling speed, network coverage etc. The framework, in conjunction with a GPS device, is thereby able to correlate the calculated metrics with the geo-coordinates, which enables visualization using satellite or street map data.

A. Data Collection

One item that proved to be problematic was the determination of the UMTS signal strength. When using WiMax or WiFi, every packet has a special IEEE 802.11 header, which contains information about the signal strength given in decibel. Such header information is not provided by libpcap when measuring UMTS. The packet capture was only able to provide information of the network layer and above. To compensate it, our framework pulls the signal information directly per AT command from the UMTS device. Yet, this method is relatively inaccurate and returns only integer numbers in the range from 1 to 32. Even more problematic, the conversion into decibel depends on the UMTS hardware and differs with each hardware manufacturer. Another problem we encountered that is specific to UMTS is the poor indoor signal reception. In our lab it was not possible to establish a stable connection, so all the stationary UMTS measurements had to be executed outside.

B. Data Analysis

This section describes the analysis part of the software, Figure 2 depicts its architecture.

The component *Database Input* reads the measurement data from the database and calculates certain performance metrics like average hop count, percentage of successful ping attempts, number of connected hosts per interval etc. Those metrics can either be written into output files for further statistical analysis or merged with the GPS data, which is the task of the *GPS Data Processing* component. The GPS Data Processing merges the prepared data with the given GPX files. Therefore, it compares the timestamps in the GPX file with the timestamps respectively the measurement interval of the data. Subsequently, it writes the measurement information with an accuracy of one interval length to the corresponding entry of the GPX file and stores this file to disk. The software GPSbabel represents one possibility to convert the proprietary format of the GPS device into GPX files, which are based on the open XML standard. To visualize the processed GPX files, further tools, like e.g. gpsvisualizer.com, are applicable. Moreover, the GPX files can be visualized by OpenStreetMap, so that the resulting graphs can be distributed under a free license.

III. Traffic Measurement and Analysis

The next section presents first measurement results and a short preliminary analysis, which have been gathered by the help of the framework. We measured the performance of the SopCast P2P video streaming software due to its wide popularity. In all measurement runs we have always recorded the same SopCast channel CCTV 1 to get a roughly comparable behavior of the P2P network. We used a standard PC with a Gigabit network interface device running Ubuntu 10.04 for the *Ethernet* measurements to obtain comparative reference measurements. This machine was connected to the Internet via a LAN connection. The *static* and *mobile UMTS* measurements were executed on a laptop running Ubuntu 11.10 with a Huawei E352s-5 UMTS stick. This UMTS stick provides a maximum bandwidth of 21.6 Mbit/s for HSDPA, respectively 5.76 Mbit/s for HSUPA. All UMTS measurements were conducted in the HSPA network of T-Mobile. We ensured that the mobile service provider did not cap the bandwidth of our connection during the measurements.

The detailed hardware information of the measurement hosts, like CPU, RAM etc., do not matter, as our software is not performance critical and the bottleneck is the capacity of

Connection Type	I: Ethernet	I: Mob. UMTS	II: Ethernet	II: Stat. UMTS	III: Ethernet	III: Mob. UMTS	IV: Mob. UMTS
Start Time	Mar 28 17:06	Mar 28 17:06	Mar 30 12:28	Mar 30 12:28	Mar 30 15:32	Mar 30 15:32	Mar 31 08:57
Capture Time	20.0 Min.	20.0 Min.	30.0 Min.	30.0 Min.	15.0 Min.	15.0 Min.	120.0 Min.
Suc. Pings (in %)	49.6	31.1	58.2	82.4	61.9	85.9	75.8
Avg. RTT (in ms)	453.762	540.159	404.079	431.794	386.671	358.730	391.813
σ RTT (in ms)	249.841	392.98	86.478	152.973	46.732	245.332	431.674
Bytes per Sec.	412,194.708	41,735.276	230,626.246	149,981.743	207,980.984	138,414.721	118,045.363
Bytes per Sec. In	155,147.482	36,444.826	139,241.656	142,241.898	142,575.469	133,213.619	113,455.849
Bytes per Sec. Out	257,047.226	5,290.450	91,384.589	7,739.844	65,405.516	5,201.102	4,589.514
Bytes per Pkt. In	423.835	950.487	786.960	1,185.849	834.480	1,213.030	1,210.935
Bytes per Pkt. Out	639.004	135.386	513.249	89.660	385.285	71.522	68.147
Avg. Hosts per Int.	34.000	5.958	6.656	3.039	11.578	2.522	2.464
Avg. Incoming Con.	5.332	3.177	5.725	2.584	9.573	1.915	2.056
σ Incoming Con.	1.511	2.446	1.065	0.910	1.292	0.507	0.833
Avg. Outgoing Con.	9.596	6.939	6.554	3.622	10.268	3.505	2.980
σ Outgoing Con.	4.308	5.344	2.130	1.641	1.798	2.021	1.591
Unique Hosts	214	185	61	56	42	33	78
Avg. Hop Count	18.168	15.773	18.179	19.208	17.444	18.905	15.211

Table I
MEASUREMENT RESULTS

the network interface. Regarding the map coloring, the value in the legend indicates the minimum value for the related color. Hence, if the legend shows a value of 80,000 outgoing bytes per interval for the color green, it means that at least 80,000 bytes per interval were transmitted during every green part in the graph. We have conducted several measurement campaigns, but due to the page limitation we will present 4 exemplary measurement runs that are elaborated in the following:

a) Measurement I: The first measurement compares a Ethernet connection with a mobile UMTS connection on a route from Bamberg to the nearby village Hirschaid and back. The most important observation is that the traffic volume of the mobile peer is one order of magnitude smaller compared to the Ethernet based peer (see Table I). Furthermore, in this measurement run the mobile peer did not receive enough data, hence, the video playback stopped halfway between Bamberg and Hirschaid due to poor UMTS network coverage.

b) Measurement II: In the second measurement we compared an Ethernet connection with a stationary UMTS connection. We noticed, that while the amount of incoming bytes stayed nearly the same, the number of outgoing bytes highly differed. With around 7 kilobytes per second nearly no video data at all were uploaded to other peers (cf. Table I).

c) Measurement III: The third measurement shows the route from Bamberg to another nearby village Bischberg, which lasted 15 minutes. One can notice that the measurement of the signal strength drops at some point on the highway to 0 (see Figure 4). One explanation for the bad connection quality at half way on the highway could be due to bad network coverage. However, as Figure 3 shows, the measurement host received enough data most of the time to display the video properly. We found that the number of successful ping attempts or especially the resulting RTT values provide a much better indication of the connection quality in contrast to the signal strength (cf. Table I: *I: Mob. UMTS* and *III: Mob. UMTS*).

d) Measurement IV: The duration of the fourth measurement was considerably longer with a run time of 2 hours and a distance of approx. 230 km from Bamberg to Munich. The signal strength was not measured in this measurement run due to the already elaborated reason. We noticed that the connection quality between Bamberg and Hirschaid suffers at the same location as in the first measurement, presumably due to poor network coverage (cf. Figure 5). Yet, in the northern parts of Bavaria along the highway to Munich, T-Mobile's UMTS network provides reasonable coverage, and thus, the video playback was sufficiently stable except for a few minor distortions due to tunnels, bridges etc. A further interesting fact is the minimal influence of the travelling speed on the downlink throughput. As one can observe in Figure 6, speeds up to 150 km/h do not heavily affect the download performance.

Table I includes additional meta data and some metrics that provide a very high level overview of the captured data traces. It is nevertheless obvious that in all UMTS experiments a drastically smaller amount of data volume was uploaded by the measurement host to other peers in comparison with the reference Ethernet measurements (cf. Table I : *Bytes per Sec. Out*). This effect can also be observed when the average size of the outgoing packets is considered: In all UMTS scenarios this metric is always significantly smaller, indicating that mostly signalling traffic has been sent, which consists mainly of connection establishment, overlay maintenance and chunk request messages. Signalling traffic consists of small packets, up to at most 300 bytes (cf. e.g. [4]), whereas video traffic is transported in larger frames in the range between approx. 1000 to 1514 bytes. Since signalling messages are not directly involved in the video playback, they can be regarded as overhead that has to be paid to leverage the video data dissemination via P2P. In all Ethernet measurements the average size of the outbound messages is comparatively bigger. It indicates that video data have been provided to other peers. Interestingly, almost the same total number of unique peers has been contacted in the according measurement traces. Though, the number of connected peers per measurement interval is much smaller in all UMTS traces. In addition, there is no

Figure 3. Measurement III - Incoming Bytes per Interval

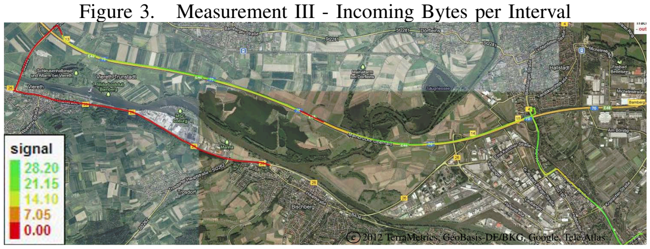

Figure 4. Measurement III - Signal Strength

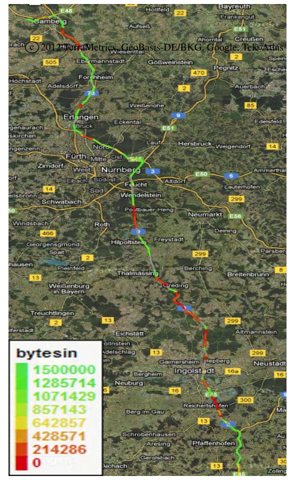

Figure 5. Measurement IV - Incoming Bytes per Interval

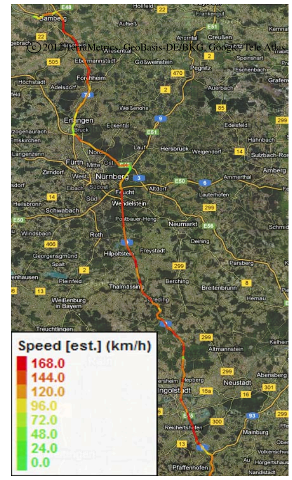

Figure 6. Measurement IV - Traveling Speed

large deviation between the average RTTs and the average hop counts with regard to the reference Ethernet measurements and the related UMTS measurement runs. Figure 7 illustrates the influence of handovers and network coverage on the inbound and outbound traffic intensity and the signalling overhead rate for *Measurement IV*. All metrics have been computed in 10 second intervals. It is clearly visible that the ratio of signalling messages in relation to video payload, i.e., the already described overhead is clearly rising (up to 400 % in a 10 seconds interval) when the throughput is decreasing. The SopCast application is not receiving enough video data and hence, sends more chunk request and keep-alive messages to avoid a playback distortion (cf. [6]). Since the throughput reduction is mainly caused by connection outages, it might be advisable to investigate the protocol performance more thoroughly and to develop potential adaptation techniques in future work. Figure 8 illustrate the mean RTT values with the standard deviation σ (std. dev.) for each measurement interval wrt. *Measurement II*. We use 2 seconds as time-out value for the replies of the RTT measurements. All answers that have been received later are considered as lost. Yet, it has to be noted that RTT values of more than 2000 milliseconds are not that rare in cellular networks due to the influence of handovers. It is quite surprising that the mean of the RTT values in the UMTS network is not much higher compared to the Ethernet measurements (cf. Table I : *Avg. RTT (in ms)* for the mean RTTs of the whole traces). However, the variability of the RTTs increases drastically. This network characteristic has an important side effect for the implementation of mobile P2P streaming applications, since the thresholds for time-out values have to be adjusted accordingly.

In the case of handovers, the state of each connection needs to be transferred from one base station to the next one. It is therefore vital for the operation of a cellular network that a P2P application uses its resources in a sustainable manner. Therefore, another important metric for cellular networks is given by the number of concurrently *open connections*, as each open connection uses the scarce resources of the underlying network infrastructure. We define a *connection* as the directed communication between two IP addresses. Figures 9 - 10 illustrate this metric for the measurement *I*. In all cases SopCast uses more outgoing than incoming connections, i.e. the number of peers, SopCast sends data to, is greater in comparison to the number of peers that send data to this peer. In addition, there are indications that there is a kind of network awareness incorporated into SopCast. As shown it uses in all captured traces less connections in total when it operates in the UMTS network. The captured data provide certainly more interesting findings and a thorough statistical analysis will reveal a much deeper insight. But we leave this item for future work. However, to enable an independent analysis, the captured packet traces are made available on-line for public use[1].

IV. Related Work

Despite the fact that there exists a large amount of scientific traffic measurement and analysis studies investigating P2P video streaming, there is relatively little work regarding the measurement of P2P streaming applications in cellular networks. One of the first studies is presented by Zhang *et al.* [7]. They introduce a Symbian based P2P video streamer and evaluate the energy consumption of the proposed system on mobile devices. Furthermore, Diaz *et al.* [1] conducted a measurement study with a focus on mobile-to-mobile P2P video streaming on a Symbian based application over cellular networks. One of their conclusions is that mobile-to-mobile P2P streaming is not feasible without the deployment of high speed radio access technologies like HSPA. In our study we could show that regardless of the scenario (mobile-to-mobile or mobile-to-wired) even in a HSPA network P2P video streaming needs definitively further protocol adaptations

[1] http://www.ktr.uni-bamberg.de/project/traces/mobileP2P.html

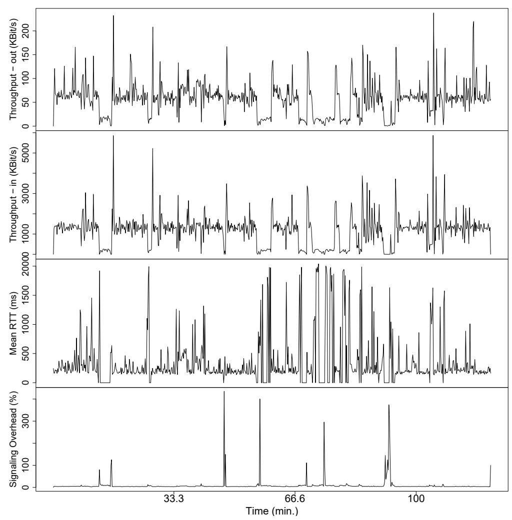

Figure 7. Measurement IV: Handover and Network Coverage Effects

to ensure satisfying quality of experience. More recently, Liu *et al.* [4] and [5] investigate the data dissemination patterns and power consumption of P2P video streaming on mobile devices. However, in their studies they are not investigating the influence of user mobility on the performance of the video data dissemination, and in contrast to our work, the measurements have been conducted in WiFi networks. In contrast to our work, all related studies investigate prototypical P2P streaming applications. To the best of our knowledge, we present the first traffic measurement study of a successfully deployed, widely used P2P streaming application operating in a cellular network.

V. Conclusion

In this study we presented a location-aware traffic analysis framework and we were able to yield the following insights: Despite the fact that the overall benefit of mobile P2P streaming has not been adequately addressed by current research, it has been already deployed and it works quite successfully. We could further point out some of the effects of cellular networks on the operations of P2P networks. The intermittent connectivity and the high variability of the RTT measurements are effects of the fluctuating conditions in cellular networks and impose a new challenge on the developers of P2P protocols. In addition, the sustainable usage of the resources of the cellular network has been partly addressed by Sopcast by using less open connections compared to the operation in Ethernet. Next, the link quality does not significantly depend on the travelling speed. Our measurements indicate that travelling speeds up to 150 km/h have only a minor impact on the connection quality. We conclude that the HSPA network coverage has a much larger importance. At last, SopCast's mobile peers do not upload a significant amount of data volume. We found that peers operating in a HSPA network did only contribute very little upstream capacity to the P2P network, even in the stationary HSPA measurements. Since modern HSPA networks

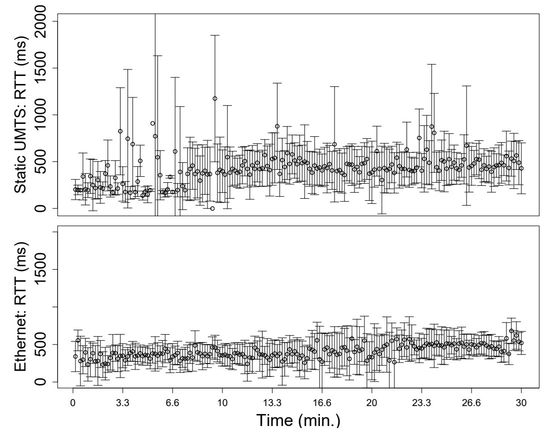

Figure 8. Measurement II: Mean RTT with C.I.

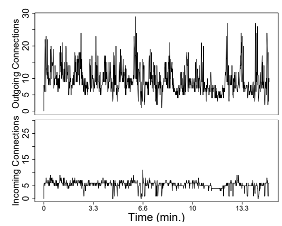

Figure 9. Measurement I: Open Connections Ethernet

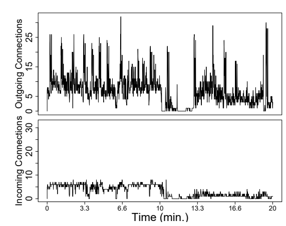

Figure 10. Measurement I: Open Connections UMTS

do not induce such a drastic uplink and downlink asymmetry, this circumstance needs further investigations. If this observation holds true, this could generate a problem for current P2P networks by the presumably growing number of mobile peers in the future. To complete the work presented in this paper, further statistical analysis of the measured traffic metrics is needed. However, our work raises the interesting question what will happen to such P2P networks like SopCast when the user population shifts to a higher level of mobility due to the steadily growing number of mobile devices and apps.

Acknowledgments

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

References

[1] A. Diaz, P. Merino, L. Panizo, and A. M. Recio. Experimental analysis of peer-to-peer streaming in cellular networks. AINA '07, pages 784–791, 2007.

[2] http://www.sopcast.org/download/android.html. Sopcast's android player.

[3] http://www.tvunetworks.com/. Tvuplayer.

[4] 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.

[5] Y. Liu, F. Li, L. Guo, Y. Guo, and S. Chen. Bluestreaming: towards power-efficient internet p2p streaming to mobile devices. MM '11, pages 193–202, 2011.

[6] S. Tang, Y. Lu, J. M. Hernández, F. Kuipers, and P. Mieghem. Topology dynamics in a p2ptv network. NETWORKING '09, pages 326–337, 2009.

[7] J. Zhang, J. Niu, R. He, J. Hu, and S. Limin. P2p-leveraged mobile live streaming. AINAW '07, pages 195–200, 2007.