Mobile Video Streaming Using Location-Based Network Prediction and Transparent Handover

Kristian Evensen[1], Andreas Petlund[1,2], Haakon Riiser[3], Paul Vigmostad[3], Dominik Kaspar[1], Carsten Griwodz[1,2], Pål Halvorsen[1,2]

[1]Simula Research Laboratory, Norway
[2]Department of Informatics, University of Oslo, Norway
[3]Netview Technology AS, Norway

ABSTRACT

A well known challenge with mobile video streaming is fluctuating bandwidth. As the client devices move in and out of network coverage areas, the users may experience varying signal strengths, competition for the available resources and periods of network outage. These conditions have a significant effect on video quality.

In this paper, we present a video streaming solution for roaming clients that is able to compensate for the effects of oscillating bandwidth through bandwidth prediction and video quality scheduling. We combine our existing adaptive segmented HTTP streaming system with 1) an application layer framework for creating transparent multi-link applications, and 2) a location-based QoS information system containing GPS coordinates and accompanying bandwidth measurements, populated through crowd-sourcing. Additionally, we use real-time traffic information to improve the prediction by, for example, estimating the length of a commute route. To evaluate our prototype, we performed real-world experiments using a popular tram route in Oslo, Norway. The client connected to multiple networks, and the results show that our solution increases the perceived video quality significantly. Also, we used simulations to evaluate the potential of aggregating bandwidth along the route.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems; H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems

General Terms

Performance, Reliability

Keywords

Performance, Multilink, Streaming, Location-based, Mobile devices, Roaming, Bandwidth prediction

1. INTRODUCTION

A large portion of the traffic on the Internet today is streamed multimedia content from services such as YouTube, Fancast, Hulu, Joost, Spotify and Last.fm, or from one of the many providers of live video and audio feeds like BBC, TV2 and NRK. To adapt to varying resource availability, media adaptation is frequently used, and in the case of video delivery, adaptive HTTP streaming (e.g., [13, 15, 22]) is currently the dominating technology. In the case of streaming to mobile clients, a well known problem is fluctuating bandwidth [18]. The current HTTP solutions enable the video quality to adapt and follow the bandwidth fluctuations in the network. However, challenges still include handling of network outages, moving into and out of networks and determining which video quality to use. The video quality should preferably not change too often, nor should it change by too many levels in a single jump as it negatively affects the user's perceived quality [14, 23].

In this paper, we address the fluctuating bandwidth challenge in mobile streaming scenarios. The solution is based on our bandwidth prediction service [17] for commuters, which uses a lookup service to predict 3G (HSDPA) bandwidth availability based on the geographical location of a device. We extend this by combining it with a technique for transparent network roaming and connection handover, without requiring operating system (OS) modifications or introducing new network protocols. Multi-homed devices can jump between available networks, without changing applications running on top, and, for example, concurrently use several interfaces to achieve bandwidth aggregation [5].

The network and bandwidth lookup service relies on a database of monitored networks. The database is populated through crowd-sourcing and contains bandwidth measurements made at different geographical locations along different commuter routes. Streaming clients query the lookup service to predict and plan which networks to use when, and the amount of data that can be downloaded or streamed at different locations. Such a solution enables buffering to fill gaps that result from periods of no connectivity or handover delays, smooth out quality-reducing bandwidth fluctuations, and increasing the video quality if higher quality networks are available along the path.

Our HTTP-based streaming video system [8] was used to evaluate the performance of the solution described in this paper. The system divides videos into two second segments, and then encodes these segments at different bitrates. We present experimental results from a tram commute route in Oslo (Norway), where the multi-link framework seamlessly switched between WLAN and 3G networks, Oslo's real-time traffic information system (Trafikanten) provided hints about tram arrivals and the lookup service provided resource availability information along the path to plan for the best possible video quality. The experiments demonstrate significant improvements in perceived video quality.

The rest of the paper is structured as follows. Section 2 introduces examples of related work, while section 3 describes the location-based transparent handover solution. The results from our experiments are presented and discussed in section 4, and we conclude our work in section 5.

2. RELATED WORK

System support for mobile video streaming has for some time been a hot research topic, and many video services are available for mobile devices. A remaining challenge is to adapt streaming to the unpredictable behavior of wireless networks like General Packet Radio Service (GPRS) and High-Speed Downlink Packet Access (HSDPA). With mobile receivers in such networks, fluctuating network bandwidths strongly influence the video streaming service performance [4, 18], raising a need for bitrate adaptation to cope with temporary connection loss, appearance of new networks, high error rates, insufficient channel capacity, etc.

In the area of *video streaming to mobile devices*, there are several existing systems, and, as in the wired network scenario, adaptive HTTP streaming solutions are frequently used. For example, Apple's HTTP Live Streaming [15] is used with Quicktime and the Darwin Streaming server to stream to the iPhone, and Windows-based phones can use Microsoft's Smooth Streaming [22], i.e., both monitor the download speed of video segments and dynamically adapt to resource availability changes by switching between video segments coded in different qualities and bitrates. To deal with fluctuations and connection loss, most systems today (HTTP streaming included) perform pre-buffering.

All major operating systems (OS) support *multi-homing*, i.e., a client can be connected to multiple networks simultaneously. However, the default behavior is to only use one link at a time. Different ways of enabling multiple links have been suggested for every layer of the network stack, but all of the approaches are tuned to a specific application, scenario, require modifications to either end system or are based on invalid assumptions. For example, [11] makes significant changes to both the client and server, while [6] is based on assumptions that are not valid in real world networks (it requires changes in all WLAN-access points). An example of an application-specific multi-link solution is presented in [5]. We used multiple links simultaneously to improve an adaptive HTTP-based streaming solution.

Roaming clients has been a popular research topic for many years. In addition to the IETF-standardized Mobile IP [9,16], different solutions have been proposed. For example, the work done in [2] concludes in a semi-transparent handover solutions. No changes have to be made to the actual applications, however, the solution requires knowledge of the applications running on top (the port numbers). Also, it relies on active probing to determine if links are available and user interaction to switch between links. Another example is Wiffler [1], where the main idea is to delay transfers in order to save money. Applications that send data specify their delay threshold, and the solution makes use of a proxy, switches between 3G and WLAN and tries to predict WLAN-availability. However, unlike our work, the WLAN-availability is calculated on the fly and network switching only works for upstream traffic, among others.

Lookup services for QoS information and prediction have been suggested before. For example, Horsmanheimo et al. [7] investigated the usefulness of location-aided planning and adaptive coverage systems for network planning and resource management of mobile networks, and Sun et al. [19] presented an idea of using network resource awareness on the application level. Moreover, Wac et al. [20] described a general QoS-prediction service containing real-world performance data, and the feasibility of using such predictions based on geographical location, time and historical data was proven for a mobile health monitoring application [21]. Liva et al. [12] predicted network outage gaps using a wave-guide theory model and calculated the signal behavior, and, recently, a geo-predictive media delivery system [3] has been presented that predicts network outages and pre-buffers video.

In summary, a lot of work describing useful components in order to build our target solution exists – either as envisioned ideas, simulations or different application scenarios. However, to the best of our knowledge, no other system integrates everything into one solution and presents a running prototype with real-world experiments, i.e., they do not experience all the possible real-world complications [10, 17]. Our solution predicts the availability of several networks and their bandwidths at the different locations along the path, and then it plans and adapts video quality to available bandwidth for a mobile device.

3. SYSTEM ARCHITECTURE

This section describes our individual components and how they are integrated to meet the challenges of mobile video streaming. First, we introduce the core concepts of our multi-link framework, called MULTI. Then, we describe the lookup service, before showing how we use these components for bitrate prediction and video quality scheduling.

3.1 MULTI

To the best of our knowledge, generic, transparent ways of enabling multiple links simultaneously does not exist. To allow for the development of transparent multi-link solutions, we have created our own framework, MULTI. MULTI monitors the network subsystem of a client device, configures the interfaces and routing subsystem, and enables multiple links simultaneously. The applications that implement MULTI have access to a continuously updated list of available network interfaces. In this section, we first introduce the framework, before describing how we used it to implement transparent connection handover.

3.1.1 Enabling multiple links

How MULTI was used to implement transparent roaming is summarized in figure 1. MULTI consists of several modules and a globally reachable proxy is its central element that allows for the creation of transparent multi-link solutions. Without it, the remote application (for example a web-server) would have to be changed in order to support multiple connections. This is often not possible or desirable.

When MULTI detects a change in link state on a multi-homed client (for example using Linux' RTNetlink library), the link module either requests an IP address using DHCP or reads it from a pre-loaded configuration. Once the required

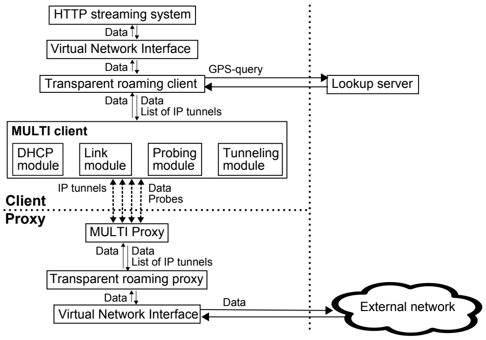


Figure 1: Roaming client architecture

information has been received or found, the link module configures the network interface (binds the IP address) and the routing tables. The routing tables must be updated, otherwise, the OS kernel does not know how to route packets to and from the different networks.

After the link has successfully been assigned an IP address, the link module notifies the probing module. The probing module is responsible for maintaining MULTI's multilink overlay network between the client and the proxy. When notified of a change in link state, an IP tunnel is either added to or removed from the overlay network. Using IP tunnels allows MULTI to support most network and transport layer protocols. We use UDP as the tunneling transport protocol, and the probing module performs NAT hole punching by sending probes to the proxy with a configurable interval. The proxy replies to these probes, forcing the NAT to keep a state for each "connection". Finally, when a tunnel is added or has been removed, the application implementing MULTI is notified. The application is also informed of the different MTU sizes, allowing developers to, for example, discard packets that do not fit inside one or all of the tunnels.

The proxy-part of MULTI consists only of the probing module. Each tunnel is given a unique ID at the client, and each client a unique ID by the proxy. For each new tunnel, the proxy stores the "connection" information, and a list of all available tunnels is exposed to the proxy application. The probe packets contain also a list of all available tunnels. The probing module uses this information, in combination with timers, to determine when a tunnel has been removed and a link is no longer available.

3.1.2 Roaming and transparent handover

In order to support transparent connection handover, the transparent roaming client and proxy (the applications implementing MULTI) create virtual interfaces. Desired routes are configured to go through the virtual network, and applications cannot distinguish a virtual from an actual network interface. A virtual interface, however, is controlled by a user space application. All data sent to this interface is received by the application, and it is up to this application to send the data to the network.

To perform transparent handover, the roaming client at given intervals, or when changes in link state occurs, queries the lookup service described in the next section, using its GPS-coordinates. This service returns the capacity of the networks in the area, and the client selects the available link (tunnel) with the most capacity to use for upstream traffic. The ID of the selected tunnel is relayed to the roaming proxy, and the handover is completed when the proxy updates which tunnel it uses for downstream traffic. A handover is also performed if a tunnel is removed. Because the "normal" applications on the multi-homed client are bound to the virtual interface, they will not be affected by the handover and will proceed as normal. Thus, they do not need to be modified to support handover.

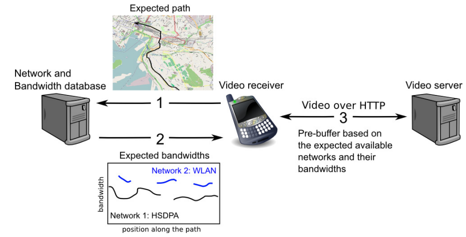


Figure 2: Streaming architecture

The roaming proxy uses source NAT to enable the client requesting data outside of the overlay network. Source NAT rewrites the source IP of packets to be that of the proxy. The remote server believes it is connected to the proxy, and all packets destined for the client is routed through the proxy.

3.2 GPS-based lookup service

In [17], we introduced a *GPS-based bandwidth lookup service*. For the solution presented in this paper, we extended the lookup service with support for storing information about different alternative networks (shown in figure 2). The following steps are performed:

- The client requests network and bandwidth availability information along its intended route. The route is known in advance, for example a commute route.
- The lookup service returns the positions of the available networks and a sequence of bandwidth samples for each point listed in the path description.
- During a streaming session, the receiver combines the measured, observed bandwidth with stored data in the database of the lookup service. Knowing network availability and bandwidth oscillations that have been measured by other clients before, the streaming system can calculate the estimated number of bytes that it can download during the remaining time of the trip. This technique is explained in detail in [17].

Using this approach, it is possible to predict the amount of data that can be downloaded in the future to smooth the video quality over the whole session.

3.2.1 Network and bandwidth logging

To provide the datasets of observed real-world network availability and their bandwidths to the streaming devices, we rely on crowd-sourcing. The idea is that the users themselves populate the database with their perceived performance at a given location [20]. Supporting this step is not restricted to this particular service – a client can contribute to the improvement of the database whenever it performs bulk data transfer. During each video streaming session, the roaming client monitors streaming performance, and this information is reported back to the lookup service server and stored in the lookup database.

3.2.2 Lookup database

The data points representing the expected throughput at different geographical locations are, as described above, collected by the users of the video service, and the network information is stored in a database with standardized geographic information system (GIS) extensions for handling location-based calculations. The database used in our first prototype is PostgreSQL using the PostGIS extensions [1].

All the information about the network and the performance from one measurement at a given time is stored as a single record in the database. This record includes network id, time, GPS coordinates and observed performance metrics like bandwidth, round-trip time and packet loss rate. Applications can then, for example, send the query

```
SELECT network_id, AVG(bandwidth)
FROM table_observed_performance
WHERE query_gps = gps AND time < 10-days-old
GROUPBY network_id
```

which returns the predicted average bandwidth for all available networks at a given GPS location based on measurements from the last 10 days.

The above information is sufficient if a user moves arbitrarily around. However, users often follows a given path, e.g., when commuting (our scenario), which can be used to perform long-term bandwidth availability prediction. Our database therefore defines a table for known paths, such as well-known commute routes in Oslo, returning a list of GPS coordinates and the respective time spent at given locations (for example within the vicinity of a GPS-coordinate or at a station). Thus, using the query above for every location, the media downloader can calculate the predicted amount of data that can be downloaded. It gains the ability to fill gaps in the stream that are caused by expected network outages, and it can schedule for a more stable video quality.

Information about the network provider of a given network ID is kept in separate table, and can be used to look up relevant data such as pricing. Although such a policy is not implemented in our prototype, the user can then for example choose the cheapest network at a given position.

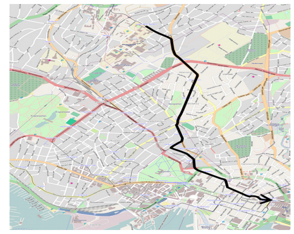
3.3 Video streaming and quality scheduling

In our commute scenario, the location-based network and bandwidth lookup service predicts the available networks and calculates their historically observed bandwidths along the path. Based on this, the segmented adaptive HTTP streaming system can make a target schedule for the video quality of each segment, targeting continuous playout (even during network outages) and a stable perceived quality (even during heavy bandwidth oscillations). Connection handover and link selection is currently performed transparently by the roaming client and proxy.

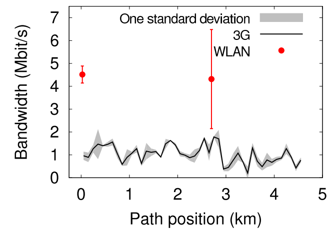
Our streaming system provides two video quality schedulers (described in detail in [17]) a *reactive* and a *predictive* algorithm. The reactive algorithm is basically intended for sessions where predictions are unavailable, and the quality levels that are chosen are decided by current bandwidth and buffer fullness. E.g., when the buffer reaches a certain size for a given level, the system is allowed to increase the quality. Similarly, when draining the buffer, the selected quality is reduced if the buffer shrinks below a threshold. Here, the thresholds are defined to avoid quality oscillations [17].

The predictive algorithm takes the bandwidths predicted by the lookup service as input and calculates the highest quality level that can be used for the rest of the trip without getting a buffer underrun anywhere (according to bandwidth predictions). Then, to adapt to deviations between observed

[1]http://postgis.refractions.net/



(a) Tram commute path



(b) Networks and bandwidths

Figure 3: Map and observed resource availability.

and predicted bandwidths, the schedule is recalculated occasionally to verify that the predicted resource availability matches the observed availability, and the selected quality adjusted if necessary. Additionally, to prevent too optimistic predictions, we use the reactive algorithm as a ceiling for the chosen quality. Also, if the transparent roaming client is forced to chose another network than the one with the highest stored bandwidth (which is used to create the segment quality schedule), a certain period of instability will occur until the HTTP streamer recalculates its schedule.

4. EXPERIMENTS AND RESULTS

To test our proposed solution, combining segmented adaptive HTTP streaming, multi-link and location-based QoS-information lookup services, we have performed real-world experiments on a tram commute route with different available networks in Oslo. Here, we present our results as a proof of concept.

4.1 Test scenario

Our real-world experiments was performed on a tram commute route between the University of Oslo (Blindern) and Oslo downtown, as shown in figure 3(a). Several networks with varying network availability depending on location are available along this route. Our client and server ran Linux 2.6.35, and the web server used was Apache 2. In our tests, the client was able to connect to the *Eduroam* WLAN [2], available at the University tram station and outside the Oslo University College, and the Telenor's [3] 3G network for the rest of the route. The predicted available bandwidths as a function of the distance from the start of the route are shown in figure 3(b). We observe that the WLAN has high bandwidth (compared to 3G), but it has a very limited range (tram stations only), whereas the 3G network was available everywhere along the path. To predict the duration of WLAN availabilities during our trip, we used the online traffic information systems Trafikanten to estimate the time until the arrival of the next tram.

For content, we used European soccer matches. These videos are encoded in 2-second segments where each segment is available in six quality levels for adaptation. The different quality levels range in average rates, from the lowest at 250 kbps to the highest at 3000 kbps.

4.2 Results

We performed three different sets of experiments: 1) 3G only, 2) switching between WLAN and 3G and 3) aggregating the WLAN and 3G bandwidths. For all tests, the performance of both video quality schedulers was evaluated. The

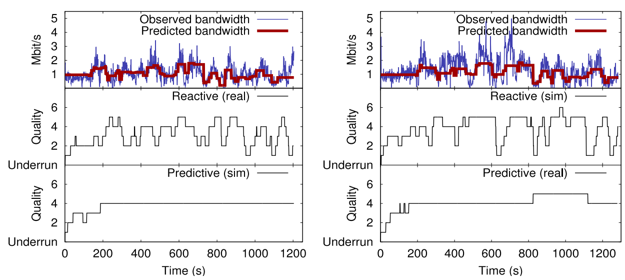
[2]http://www.eduroam.org

[3]http://www.telenor.no

first two sets of experiments were performed on the tram. Because any two runs will experience different conditions (for example, available bandwidth differ and the total travel time varies due to traffic), results using the other scheduler was simulated and plotted based on the gathered bandwidth traces. Thus, the real-world performance of the reactive scheduler was compared with simulated performance of the predictive scheduler, and opposite. This was done to get directly comparable results. Our bandwidth aggregation results were gathered using simulations.

4.2.1 3G only

3G was used as our base case and an average bandwidth of about 1000 kbps was observed. The 3G-network was available for the entire trip and it provided stable performance at all times of day. Thus, our prediction algorithm was able to improve the video quality significantly compared to the reactive scheduler, as shown in figure 4. The reactive scheduler, which resembles existing HTTP streaming solutions, followed the available bandwidth and gave a more unstable video quality than the predictive scheduler. With respect to the achieved quality, the video quality rarely exceeded level 4 at 750 kbps (level 5 requires about 1500 kbps).



(a) Real-world: reactive (b) Real-world: predictive

Figure 4: Only 3G

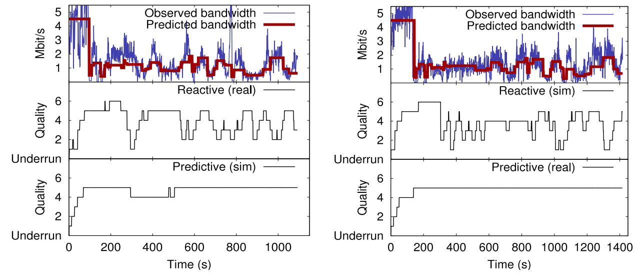
4.2.2 Switching networks (WLAN and 3G)

Figure 5 shows the results when the client switched between networks. The Eduroam WLAN always outperformed the 3G network, and was chosen whenever available. Both schedulers benefited from the increased capacity of the WLAN. The video quality was significantly higher than with 3G only (figure 4). With the predictive scheduler, the media player was allowed to stream at quality level 5 (1500 kbps) for most of the trip, compared to level 4 (750 kbps) when only 3G was used. The reason is that the higher bandwidth of the WLAN enabled the client to receive more data. Thus, it was able to work up a bigger buffer and could request segments in a higher quality. Also, the predictive scheduler achieved a much more stable video quality than the reactive scheduler.

As we described earlier, the handover was handled transparently, and with respect to handover performance, we have plotted the throughput for the streaming sessions from figure 5 in figure 6. From the plots, we can observe that the handover time is minimal and that the client receives data without any significant idle periods.

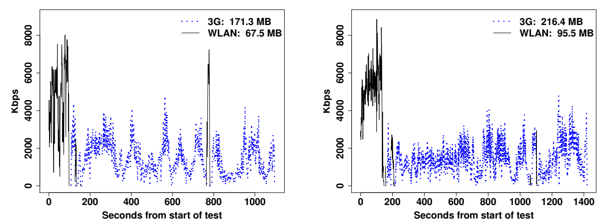
4.2.3 Aggregating networks (WLAN + 3G)

To evaluate the performance of aggregating WLAN and 3G, we simulated several streaming sessions. The simulated bandwidth of WLAN and 3G was based on traces from our real-world experiments. Because WLAN was only available



(a) Real-world: reactive (b) Real-world: predictive

Figure 5: Switching between WLAN and 3G

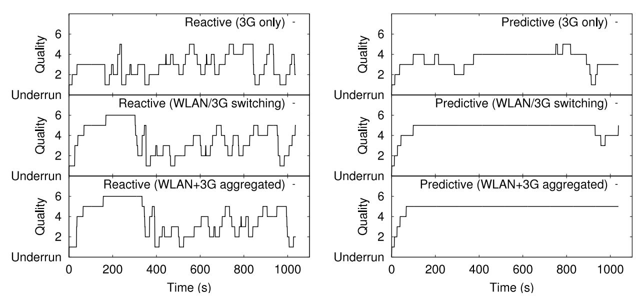


(a) Handover: reactive (b) Handover: predictive

Figure 6: Achieved throughput and handovers

during the first minute or so of the route, the available bandwidth was most of the time equal to that of 3G.

The results from one representative set of simulations is shown in figure 7. As expected, the performance improved when more bandwidth was available. For example, when the client could aggregate WLAN and 3G bandwidths, the predictive scheduler was able to avoid a dip in quality towards the end. Also, the additional bandwidth allowed a faster growth in video quality.



(a) Reactive (b) Predictive

Figure 7: Perceived video quality: 3G only vs. WLAN/3G switching vs. WLAN+3G aggregated

4.3 Discussions

The solution and prototype implementation presented in this paper demonstrate the potential of combining adaptive HTTP-based video streaming with a predictive video quality scheduler, a location-based resource lookup service, transparent handover and bandwidth aggregation. However, there are still many issues that must be addressed.

The performance of networks varies depending on external factors such as weather conditions, network outages, time-of-day and day-of-week (e.g., rush hour differences), inaccurate position measurements, and so forth. The prototype used in this paper does not take any such factors into account. This requires long-term observation studies that we plan perform in future work.

Also, the current evaluations are all based on the assumption that users commute along a fixed path. However, this is not always the case. For example, the user might be in the passenger seat of a car. We are currently investigating several alternative scenarios, looking at path prediction, per-hop (between stations) bandwidth prediction, and so forth.

In the network switching scenario (section 4.2.2), there are several possible policies of how to chose which network to use. We chose to prioritize WLAN due to its higher bandwidth, which resulted in the highest achievable video quality. Another policy might prioritize based on power consumption or based on network pricing (some users may have very expensive 3G-data plans). For example, several providers [4] offer subscribers free access to WiFi-hotspots. By using these when they are available, a user can save both money (after an initial amount of data has been received, the user is charged per MB) and bandwidth (many providers reduce the bandwidth over 3G when a transfer cap is reached).

5. CONCLUSION

In this paper, we have presented a video streaming solution aimed at roaming clients that is able to reduce the effect of fluctuating bandwidth. Our solution enables the use of multiple networks, if available, and performs seamless handover if the client is within range of a network with a higher priority. Network selection is based on lookups in a QoS-database filled with GPS coordinates and corresponding bandwidth measurements. The database is populated by crowd-sourcing, and enables the video streaming client to predict bandwidth and schedule video quality.

The performance of a prototype was evaluated on a popular commute route. The results from the real-world experiments show that our proposed solution can increase perceived video quality. Also, we performed several simulations to determine the potential of bandwidth aggregation, with positive results. However, the are many challenges left to solve, and ongoing work includes the evaluation of various policies for changing and prioritizing networks, various factors such as time of day and bandwidth variation, more advanced techniques for path prediction and implementing bandwidth aggregation in our prototype.

6. REFERENCES

[1] Balasubramanian, A., Mahajan, R., and Venkataramani, A. Augmenting mobile 3g using wifi. In *Proc. of ACM MobiSys* (2010), pp. 209–222.

[2] Caro, G. A. D., Giordano, S., Kulig, M., and Vanini, S. Mobility across heterogeneous networks.

[3] Curcio, I. D. D., Vadakital, V. K. M., and Hannuksela, M. M. Geo-predictive real-time media delivery in mobile environment. In *Proc. of MoViD - ACM MM workshops* (Oct. 2010), pp. 3–8.

[4] Diaz-Zayas, A., Merino, P., Panizo, L., and Recio, A. M. Evaluating video streaming over GPRS/UMTS networks: A practical case. In *Proc. of IEEE VTC Spring* (Apr. 2007), pp. 624–628.

[5] Evensen, K. R., Kaspar, D., Griwodz, C., Halvorsen, P., Hansen, A. F., and Engelstad, P. E. Improving the Performance of Quality-Adaptive Video Streaming over Multiple Heterogeneous Access Networks. In *Proc. of ACM MMsys* (2011), pp. 57–69.

[6] Guo, F., and Chiueh, T.-C. Device-transparent network-layer handoff for micro-mobility. In *Modeling, Analysis & Simulation of Computer and Telecommunication Systems* (2009), pp. 1–10.

[7] Horsmanheimo, S., Jormakka, H., and Lähteenmäki, J. Location-aided planning in mobile network-trial results. *Wireless Personal Communications 30* (2004), 207–216.

[8] Johansen, D., Johansen, H., Aarflot, T., Hurley, J., Kvalnes, A., Gurrin, C., Zav, S., Olstad, B., Aaberg, E., Endestad, T., Riiser, H., Griwodz, C., and Halvorsen, P. DAVVI: A prototype for the next generation multimedia entertainment platform. In *Proc. of ACM MM* (2009), pp. 989–990.

[9] Johnson, D., Perkins, C., and Arkko, J. Mobility Support in IPv6. RFC 3775 (Proposed Standard), June 2004.

[10] Kaspar, D., Evensen, K., Hansen, A. F., Engelstad, P., Halvorsen, P., and Griwodz, C. An Analysis of the Heterogeneity and IP Packet Reordering over Multiple Wireless Networks. In *Proc. of IEEE ISCC* (2009).

[11] Kim, K.-H., and Shin, K. G. PRISM: Improving the Performance of Inverse-Multiplexed TCP in Wireless Networks. *IEEE Transactions on Mobile Computing* (2007).

[12] Liva, G., Diaz, N. R., Scalise, S., Matuz, B., Niebla, C. P., Ryu, J.-G., Shin, M.-S., and Lee, H.-J. Gap filler architectures for seamless DVB-S2/RCS provision in the railway environment. In *Proc. of IEEE VTC Spring* (May 2008), pp. 2996–3000.

[13] Move Networks. Internet television: Challenges and opportunities. Tech. rep., Move Networks, Inc., Nov 2008.

[14] Ni, P., Eichhorn, A., Griwodz, C., and Halvorsen, P. Fine-grained scalable streaming from coarse-grained videos. In *Proc. ACM NOSSDAV* (2009), pp. 103–108.

[15] Pantos, R., Batson, J., Biderman, D., May, B., and Tseng, A. HTTP live streaming. http://tools.ietf.org /html/draft-pantos-http-live-streaming-04, 2010.

[16] Perkins, C. IP Mobility Support for IPv4. RFC 3344 (Proposed Standard), Aug. 2002. Obsoleted by RFC 5944, updated by RFC 4721.

[17] Riiser, H., Endestad, T., Vigmostad, P., Griwodz, C., and Halvorsen, P. Video streaming using a location-based bandwidth-lookup service for bitrate planning (accepted for publication). *ACM Transactions on Multimedia Computing, Communications and Applications* (2011).

[18] Riiser, H., Halvorsen, P., Griwodz, C., and Hestnes, B. Performance measurements and evaluation of video streaming in HSDPA networks with 16QAM modulation. In *Proc. of IEEE ICME* (June 2008), pp. 489–492.

[19] Sun, J.-Z., Sauvola, J., and Riekki, J. Application of connectivity information for context interpretation and derivation. In *Proc. of ConTEL* (2005), pp. 303–310.

[20] Wac, K., van Halteren, A., and Konstantas, D. Qos-predictions service: Infrastructural support for proactive qos- and context-aware mobile services (position paper). Springer Lecture Notes in Computer Science (vol. 4278). 2006, pp. 1924–1933.

[21] Wac, K., van Halteren, A., and Konstantas, D. QoS-predictions service: Infrastructural support for proactive QoS- and context-aware mobile services (position paper). In *Proc. of OTM Workshops*. 2006, pp. 1924–1933.

[22] Zambelli, A. Smooth streaming technical overview. http://learn.iis.net/page.aspx/626/smooth-streaming-technical-overview/, 2009.

[23] Zink, M., Künzel, O., Schmitt, J., and Steinmetz, R. Subjective impression of variations in layer encoded videos. In *Proc. of IWQoS* (2003), pp. 137–154.

[4] AT&T in the US and Telia in several European countries (through their Homerun-service) are two examples.