Exposé

Modern Media Streaming and its Transport in Mobile Networks

Florian Metzger
1046544

für
A 786 880
Dr.-Studium der technischen Wissenschaften (Dissertationsgebiet: Informatik)
Universität Wien

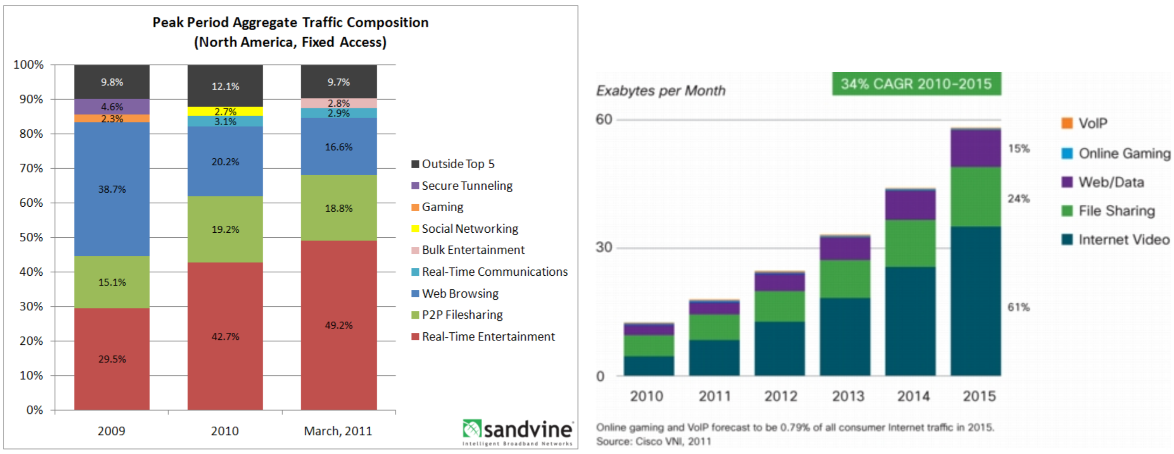
November 2011

Contents

1 Introduction

The Internet began its life in the 1960s after the invention of packet switching [1]. Throughout the years an amalgamation of networks, protocols, architectural evolutions, constant and changing principles, and applications lead to the Internet in its current form.

Today, services based on the World Wide Web dominate the Internet's landscape. Especially noteworthy is the dominance of video streaming services in the traffic mixes of today and in future predictions (cf. Figures 1a and 1b). This development was supported through the availability of cheap and small digital camcorders and especially the now sufficient access bandwidth to support high quality video streams. A similar development for cellular networks is now ongoing. Through the advent of affordable high performance smartphones and access technologies like UMTS and LTE many are now using their phones as the primary device for interacting with the Internet, including video streaming services such as YouTube, Netflix, or Hulu.



(a) Netvine, North American peak traffic (Source: [2]).

(b) Cisco, global consumer Internet traffic prediction (Source: [3]).

Figure 1: Traffic measurements and predictions.

This can put pressure on the overly complex cellular network structures. The radio transmissions have only access to a limited radio frequency spectrum that, moreover, has to be shared with any other phone user in the same cell. But there is also deemed to be significant pressure on the traffic management mechanisms of the mobile core networks backing up and aggregating the numerous radio cells of an operator. Little is known of the exact make-up of these networks as they are closely guarded secrets of the operator.

The popularity of these video services can to a degree be attributed to their full integration into the WWW. While previous offerings used completely different techniques which required the use of additional software, today one can enjoy videos just by directing ones browser to a website. Furthermore, other services related to video streaming are now also beginning to migrate to Web platforms. An example that is becoming increasingly popular at the moment are cloud gaming offerings which uses mechanisms very similar to real time video streaming but has stringent temporal constraints.

The topic of the thesis will be the research of the inner workings of these new streaming services and mechanisms and especially its impact on the aforementioned cellular core network structures. This means conducting performance evaluations of the streaming mechanisms on itself and in the context of mobile networks as well as performing systematic assessments and classifications of the mechanisms.

The result will be an understanding of the quantitative attributes related to these new forms of streaming. Furthermore, the thesis should provide tools and methods that help decide all participants of media streaming and mobile network operators which protocols and methods to choose and which are best suited for specific applications.

The tackled protocols, systems, and mechanisms are described in Section 2. Section 3 details the methods that are and are planned to be used for the research. The final section gives a rough estimation on the thesis' schedule.

2 Research Areas and Related Research

Video streaming touches many aspects of computer communication network research. They can be categorized in aspects touching on the one hand the service itself and on the other hand the transport and underlying network. This chapter splits the topic roughly on a top-down layer approach. At first, modern video streaming applications and their mechanisms are presented. Afterwards, the streaming transport is discussed with a concluding section on the influences of wired and especially mobile network architectures.

2.1 Services in the Internet

Most of today's services base itself on the Web and HTTP as "transport protocol". It is even proposed to use a slightly modified version of HTTP as the basic end-to-end protocol for a future iteration of the Internet[4] as it already fulfils many of the demands proposed for the future of the Internet.

2.1.1 Classical Push-Based RTP Streaming Approach

The established standards for video streaming use combinations of RTP, RTSP, and RTCP [5, 6]. They are the classic approach to video streaming according to literature (cf., e.g., [7, p. 589ff], and [8, p. 426ff]), RTP is a dedicated streaming protocol suite, that offers out-of-band control using TCP with separate UDP-based content transport channels. However, the requirement of several open UDP sockets does not work well in environments using middleboxes, .e.g, firewall or network address translation (NAT) nodes, because of the difficulty to forward incoming UDP packets to the destined host. Furthermore, it mandated additional software to be installed at the client.

RTP streaming is a push-based design. This means, that the server is in control of every aspect of the streaming. When the client requests the start of the playback over the control channel, the server starts pushing down the data over one RTP path. The server application completely controls the transmission speed and the video quality. Therefore, performance and quality metrics have to be exchanged between the two communicating nodes to allow for informed choices on the server side.

RTP also offered intricate multicasting mechanisms, i.e., the ability to simultaneously deliver the same stream to multiple nodes using specially configured routers. These mechanisms were however never fully adopted by the majority of Internet users. On the one hand the required infrastructure was only available in some provider networks, never at the Internet's full scale. On the other hand, the rise of community pages like YouTube has shown, that the interest does not lie in watching the same content at the same time, but rather in high individualism. Therefore, multicast is less relevant for today's streaming. If there is a media event that is streamed live, one can always fall back to using the relatively new structures of Content Delivery Networks (CDN) to be able to serve large groups of users while still conserving bandwidth on the Internet backbones.

There are also other proprietary and standardized streaming systems which better fulfil the requirements of specific fields of applications. Multimedia Broadcast Multicast Services (MBMS)[9, 10] is a specification defined by the 3GPP group for multicasting multimedia traffic specific to the architecture in mobile networks. But similar to RTP the number of implementations and their acceptance are negligible.

2.1.2 Novel Pull-Based HTTP Streaming Approaches

HTTP streaming pursuits a pull-based approach. The client establishes TCP connections to send HTTP requests for video files stored on the Web server, which are then sent to the client. During the sequential downloading process the client can at any time start playing the file even before it is completely downloaded, resulting in a so-called pseudo-streaming behavior. With this principle the client makes its own decisions regarding the playback process. It has intrinsic knowledge on the current and estimated future connection quality. This leads to a shift of control logic from the server to the client. The former can now be implemented very lightweight, allowing, e.g., for a more distributed content placement in the Internet, especially when using.

By using HTTP the ideal platform for the client is either a plugin living inside a Web browser, the method chosen by Flash, or the browser itself, that in most cases has built-in video playback capabilities.

The capabilities and shortcomings of these novel mechanisms are not yet fully researched making it one of the prime foci of the thesis. Of special interest are:

- *Playback control and media data buffering.* Using the reliable TCP as transport protocol means that there will be no packet loss visible to the application layer. The video file will always reach the client in the same form as stored on the server. If packets are lost, the result will additional delay and decreased throughput through the occurring retransmissions. If video data does not reach the client in time before its playback buffer is depleted, this will results in video stalling and therefore a perceptible loss of quality. This situation can be alleviated or even avoided by carefully planning the playback process and the buffering behavior.

- *Application layer flow control.* HTTP file downloading is a very simple process. The server has very little influence on this and just forwards requested stored data into a

TCP socket. This is not an optimal behavior for video streaming as it does not allow for any throttling or rate management. The buffer space in mobile or embedded devices can be very limited, it would be useful to limit the amount of received data while still ensuring sufficiently available future playback data in the buffer. Services have begun to implement application layer flow control mechanisms for their video streaming application, e.g. YouTube [11]. Figure 2 shows a comparison of several possible rate management modes. The first sequence diagram shows the unaltered transfer mode observable in regular HTTP file transmissions. The second and third diagrams show possible ways of implementing application layer flow control with either evenly spread out transmissions or sending in blocks. The last diagram displays a mode using multiple requests for one video which can be facilitated for rate adaptations discussed in the next item.

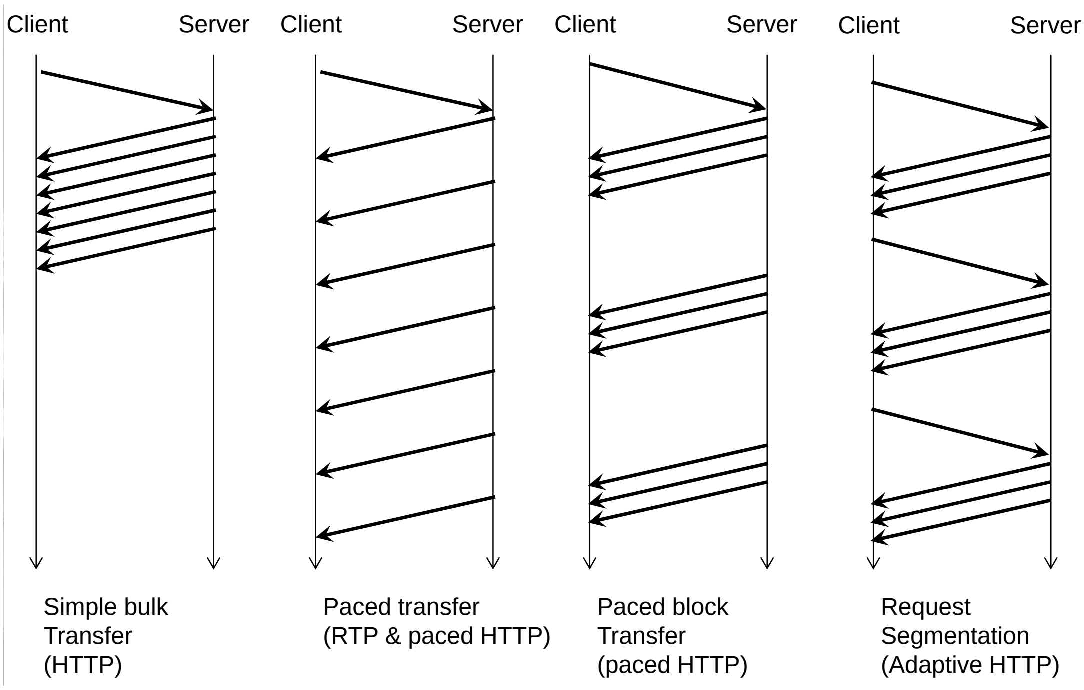


Figure 2: Comparison of several possible streaming transfer modes (Source: [12]).

- *Video quality and rate adaptation.* To be able to support varying video bitrate levels similar to RTP some adaptations to HTTP streaming are needed. At first, the video has to be encoded multiple times to several output bitrates. Also, the video container needs to be able to support switching the stream at will with everything required for the playback in place. Additionally, an independently available index file needs to correlate the video files with their contents. Alternatively, the video can be segmented into short pieces of several seconds outlined by a separate index file. Then the client can choose the stream variant that best fits its current condition, which depends on parameters including the display size or varying network QoS. [12] gives an overview over some possibilities. Several proprietary protocols, some of them in the process of standardization, tackle this, including HTTP Live

Streaming [13] and Microsoft Smooth Streaming [14]. They are either based on file segmentation or HTTP RANGE requests. Initial experiments in [15] evaluate the viability of this kind of approach. We plan to extend this by modelling the fundamental building blocks the mechanisms have in common and after that giving tools to find and evaluate combinations best suited to specific network conditions. This includes evaluations of the optimal length of segments as well as tuning the aforementioned buffering models to be able to cope with at least two new degrees of freedom, i.e. being able to load multiple segments at once using more than one connection as well as the ability to switch to another video encoding level in case of changing connection capacity or buffer pressure conditions.

The focus of the thesis will be to explore the possibility space of the aforementioned mechanisms and give the ability to make informed choices on the viability of them or whole protocols for specific use cases.

2.1.3 Further Services

There are other applications related to video streaming and are showing similar transmission characteristics. Examples include remote desktop services or cloud gaming. Cloud gaming facilitates the same core principles as video streaming. However, it adds strong bidirectional real time requirements as user interaction needs to be immediately reflected in the streamed images. One of the concerns of the thesis could be an investigation of streaming models for cloud gaming in the context of mechanisms used . Initial work has already been done, e.g., in [16, 17, 18, 19].

2.2 Media Transport

It can be questioned whether HTTP and TCP are the right choices for transporting media streams. While there are noteworthy benefits from the features, the ease of use, and the pervasiveness, it was also necessary to resort to application layer flow control to circumvent the lack of direct influence on them.

Through TCP's reliability the whole video file will always be transferred during the streaming process. As mentioned, adverse network conditions can cause TCP's mechanisms to increase latency, jitter as well as reduce the throughput. Protocols that offers congestion control but no reliable delivery and video codecs that are robust to packet loss might be more desirable for streaming. DCCP [20] is an example for such a compromise and might prove beneficial for the streaming process.

HTTP is a state-less request-response protocol. The synchronous behavior of the request-response mechanism does not allow for server events to be sent in a timely manner to the client. This increases the difficulty of implementing some extended features. Examples are server-side load balancing, or real time or live streaming. WebSocket [21, 22] is a protocol running atop of HTTP offering connection multiplexing and asynchronous as well as full duplex communication. It could be used to implement a more flexible HTTP video streaming offering or unlocking further use cases. Similar approaches should be included and evaluated in the research for this thesis.

2.3 Influences of Layer 3 and Beneath

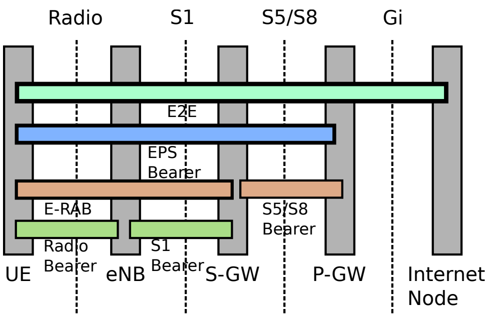
Wired Internet access has a very narrow choice of protocols on the ISO/OSI Layers 1 and 2. The typical use-case consists of a Local Area Network using Ethernet which is then tunneled through or translated into one of several access technologies, e.g., DSL, DOCSIS, or PON. Applications often make assumptions that rely on the presence of these protocols and their specific characteristics.

However, Internet access today is similarly often achieved using mobile cellular networks. The latest standardized iteration of these is LTE and the accompanying EPS core network infrastructure [23]. This is the first evolution of standards that completely removes the classical circuit switched domain making room for more radio frequency bandwidth to be used with the all-IP services achieving shared transmission capacities – comparable to today's 802.11n WiFi – albeit on much larger cell sizes of 1 to 30 kilometres. The EPS network (cf. Fig. 3b) acts as an intermediary between the radio access stations and the Internet enabling strong traffic control mechanisms as well as mobility anchored at the Serving Gateway (S-GW). Traffic is routed through the core by using tunneling over the S-GW and P-GW based on the traffic bearer concept defined either in the GTP or the PMIPv6 protocols. For every mobile device connected to the network there is one default and up to ten dedicated bearers carrying traffic filtered by pre-set QoS parameters. Control is enforced through a logically separate network control plane, that is also used to setup and tear down these bearers. Figure 3c displays the disparity between the Internet's protocol stack and that of an LTE network encapsulating all user traffic in additional protocol layers by the tunneling process.

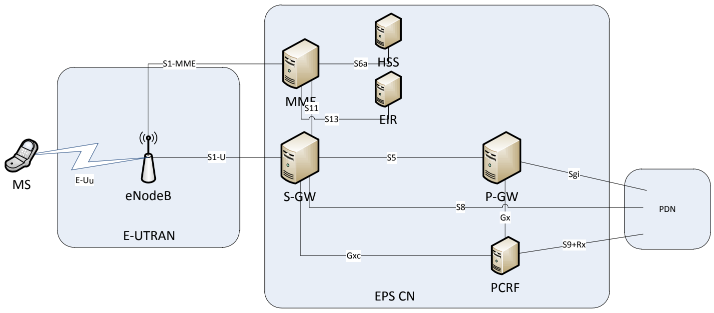
Research work is ongoing how to best work with this complex network setup. It is expected, that with the rise of mobile access the core network comes under heavy traffic pressure with negative affects on the QoS of best-effort traffic. Endeavors are required to study the loaded network's behavior. Also very little work has gone into exploring the control plane characteristics of these networks, including their performance. A novel approach could also be, to make mobile device applications, e.g. video streaming players, aware of the core networks capabilities and allow the request of tunnels tailored to their specific QoS requirements resulting in a possible increase of perceived quality.

The protocols used for the radio transmissions behave very differently when compared to Ethernet and assumptions made by higher layers may not hold any more. This can apply to, e.g., reliability, frame sizing and fragmenting, and latency amounting to undesired effects on higher-layer traffic. For example, loss in GSM and UMTS networks is often caught transparently on layer 2 and a retransmission is conducted. However, in the time the retransmission takes the transport layer may have already run into a time-out and re-requested the missing segment on its own, resulting in additional delay and a waste of bandwidth. This is especially detrimental for time-critical applications like video streaming, possibly resulting in buffer underruns and degraded quality. Transport and application layer mechanisms need to be able to understand this and cope with the effects. E.g., TCP retransmissions and congestion control could be adjusted in the course of understanding this.

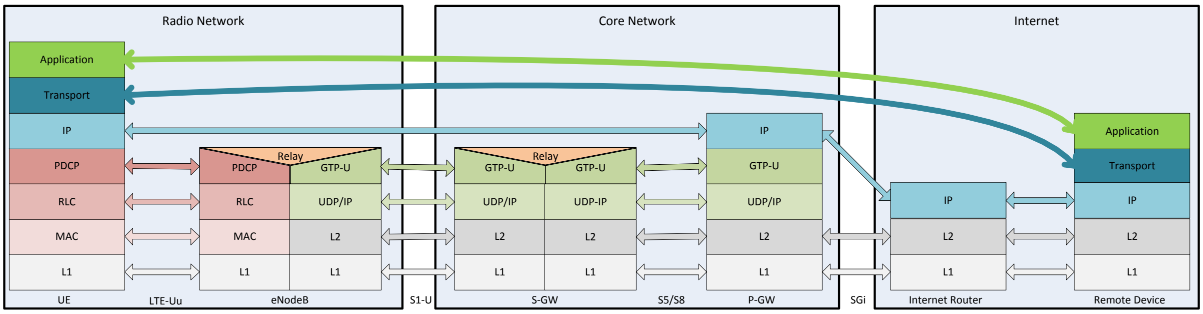
Furthermore, traffic could be avoided during cell handover occurrences. This would require cross-layer cooperation and an awareness of the application when an handover is supposed to occur. The application then could schedule its traffic accordingly. Traffic falling into a handover is subject to especially high latency and loss because the mobile network acts as a mobility anchor which needs to internally reroute incoming traffic to the mobile device's new position. HTTP traffic is especially suited to this scheduling behavior because of its statelessness and consistence of small objects that can be requested and transferred independently.



(a) LTE tunneling concepts.



(b) Minimal working subset of a LTE and EPS architecture.



(c) LTE protocol stack.

Figure 3: LTE concepts and architecture.

2.4 Manners of Control Information Exchange

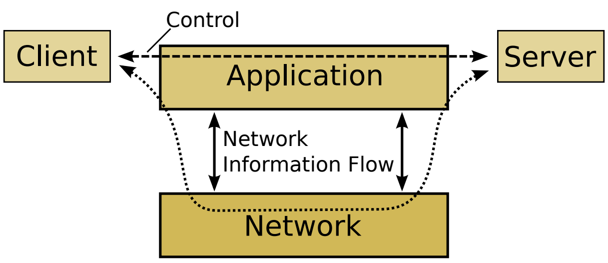


Figure 4: Theoretical information exchange paths between streaming partners.

In every network architecture there is a control information flow between its participating layers, protocols and elements (cf. also Fig. 4). Control information may need to be exchanged to coordinate and negotiate the modes of communication between two endpoints and can be helpful in many ways. For example can a network explicitly expose network's quality of service data to applications or these application can make reservation requests to the network. Communication can also work implicitly, demonstrated in the distributed TCP congestion control enabled by the lack of resource isolation. This shows two major ways for the information flow to be implemented: Either through directly and explicitly exchanging information or by one participant implicitly drawing conclusions on another participants resources and behavior.

These different approaches can also be observed in video streaming. The explicit control structure of protocol suites like IMS[24] and MBMS weaves application and network layer tightly together. This theoretically allows for an improved streaming performance at the cost of universally applicable behavior. Today's IP-based HTTP streaming forms the other side of the pole having only an implicit network information flow by guessing the current network conditions.

For the future there are open questions on how to best incorporate this information flow into streaming applications. Of special interest is the kind of information an application should rely on, and which information the application really requires. Moreover, it is to be decided what and how information should flow in the network. To support the decision making, models and ways to evaluate and compare protocols and network architectures need to be thought out, possibly resulting in a generic evaluation model.

3 Research Apparatus and Methods

The topics presents itself from a technical as well as a methodical point of view. For both, one has to find the right tools to tackle the problems. Possible ways are presented in the following two sections. To conclude, an example of already conducted work is given on how to apply the apparatus.

3.1 Technical Solution Spaces

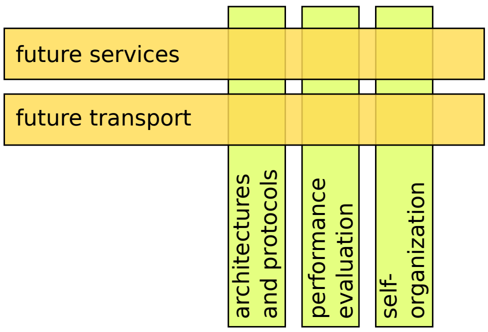


Figure 5: Technical solution spaces to the problem layers.

As discussed, the core problems are layered into services and transport. Technologically, to solve the tasks of video streaming, one can approach these threefold as displayed in

Fig. 5.

The first is to dissect the involved protocols and architectures and break them down into their functional and methodical components. This will result in an improved understanding on the manner and process of their implementation. These components serve as building blocks for generalized models that abstracts the problem space from the actual implementation. The model will be defined by a set of parameters. To explore viable parameter ranges performance evaluation methods will be facilitated.

Secondly, using performance evaluation a system is methodically tested to the outcome of determining the influence of the system's parameters on a set of performance metrics. The parameters can be categorized into system intrinsic parameters, describing behavior only relevant and observable inside the system, and external parameters. In communication networks a good example for external parameters are the network Quality of Service parameters including latency, loss, jitter, and bandwidth capacity. Identifying fitting metrics for the measurement is a challenge. They can be either subjective or objective. The former are called Quality of Experience (QoE) metrics. They can only be measured by conducting empirical user studies and questionnaires and are mapped to a Mean Opinion Score (MOS). Extensive work has already been done to define baseline references for QoE metrics. Using these, one can directly translate objectively measurable outcomes into QoE metrics. However, these mappings may need to be adjusted to be able to handle stalling as the main source of quality loss. Examples for measuring subjective quality are available in [25, 26]. Finally, one could employ methods of self-organization to try to reach improvements over conventional network setups.

3.2 Apparatus for System Analyses and Comparison

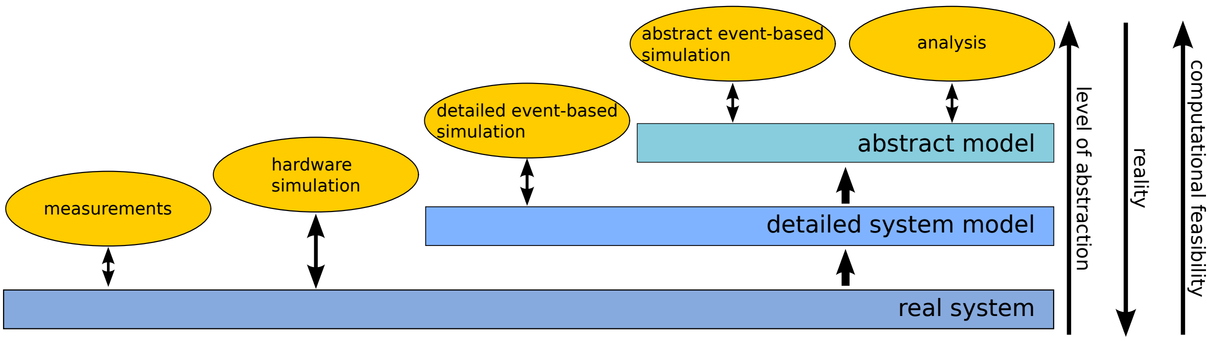


Figure 6: Methodical solution spaces and apparatus comparison.

Depicted in Figure 6 are levels of abstraction involved in system analysis and possible approaches to understand the system involved.

The upper limit for precision is achieved through actual measurements of the system itself. These give a point of reference and can be used to validate all other methods. However, this is not feasible to ascertain a larger view. The time frame or the physical size of the system is always limited.

Aside from measurements on implementations there are three further possible approaches which can widen the scope: emulation, simulation and mathematical analysis.

Emulation tries to resemble implemented functionality as closely as possible while reducing the non-important parts to a minimum. Measurements using emulations can run in a normal network environment testbed and thus can only be done on a scale equivalent to implementations. The simulative approach implements all internal and external functionality, including the physical nodes and the network, in a discrete event simulation (DES). There may be subtle functional differences between a simulation and a real implementation. Therefore, validation is required. This approach benefits from the decoupling of the simulation from physical nodes as well as real time, allowing to measure large-scale networks in a short amount of time. A mathematical analysis, for example using queuing theory and stochastic models, can then further broaden the understanding of the system.

The methods can all be used to define and explore solution spaces and are therefore important tools in understanding the problem. A fitting combination of these tools has to be found to advance the research. Our initial approach is to investigate existing streaming services, YouTube [27, 28] and,e.g., video libraries of broadcast stations for simple HTTP streaming. A suitable candidate for measuring adaptive streaming still needs to be found as some candidate services apply regional restrictions. There are several reference applications available that implement different standardization approaches. These can be used to either directly measure the performance or to setup an emulation model based on their specifications.

3.3 Modeling Example for the Case of YouTube Streaming

To give an example of the thesis' planned work an exemplary overview on the modelling process is presented. It was conducted for the buffering involved in YouTube's mode of streaming [27]. As YouTube uses a single-command pull-based approach for streaming, the influence the client has on the playback process is minimal. The player needs to maintain a tight control scheme on the playback buffer to accomplish as few buffer underruns as possible. The parameters that can be influenced by the player are:

- Initial playback start delay.
- Condition and timing for resuming playback after a buffer underrun occurs.

The constraints and metrics that must be achieved and can be observed:

- Maximum allotted buffer space at the client.
- Subjective: Perceived quality of the playback process.
- Objective: total and initial stalling times, frequency of interruptions.

External influences that were subjected on the model candidates to test their viability:

- Packet loss
- Transmission latency

During the test, four specific models were formulated. These are:

1. *Simple playback stalling.* Whenever anything can be played from the buffer do so. This results in the lowest required buffer space and also a lower limit for the total stalling time. However, it does not keep a safety buffer to handle insufficient connection quality, and is therefore prone to frequent micro-stalls.

2. *Initial playback delay.* Delay the initial start until the video can be played without buffer underruns. It also acts as a lower limit for buffer space and total stalling time but additionally has the lowest frequency of stalls – precisely one initial stall. It exhibits the best-case scenario for perceived quality disregarding the initial delay. However, it can not be implemented as an online algorithm, as it requires global knowledge.

3. *Firefox HTML5.* The algorithm is depicted in Alg. 1, its variables in Table 1. It does not differentiate between intermittent and initial conditions. The approach is in concept similar to the theoretical initial playback delay but results in very large required maximum buffer space due to conservatively chosen buffering times.

Algorithm 1 Firefox playback (re-)start decision algorithm.

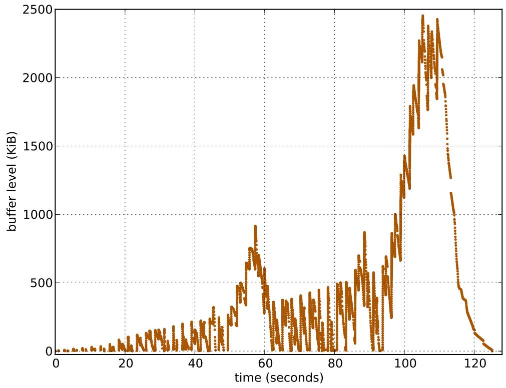
if $s_{MA} > v_{MA}$ **then**
  $c \leftarrow (b_b = 20s \vee b_T = 20s)$
else
  $c \leftarrow (b_b = 30s \vee b_T = 30s)$
end if

Variable	Explanation
s_{MA}	Moving average of the transmission speed.
v_{MA}	Moving average of the video bitrate.
c	Condition upon which to start/resume playback.
b_b	Amount of video data the buffer contains.
b_T	Amount of time spent in non-playing buffering state.

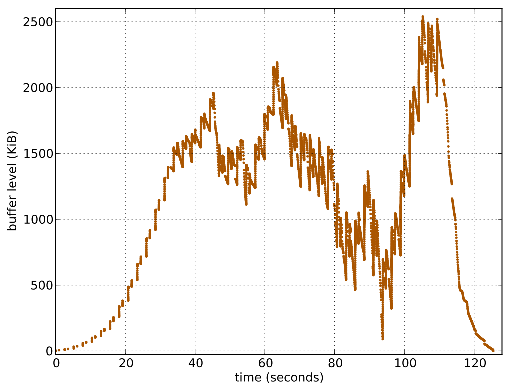
Table 1: Variables involved in buffering decisions.

4. *YouTube Flash Player.* $b_{b,initial} = 2s$, $b_{b,intermittent} = 5s$. Assumes sufficient network conditions in the beginning requiring only a short initial playback delay. If the assumption does not hold and stalling does occur, it will buffer longer to keep the occurrence frequency down.

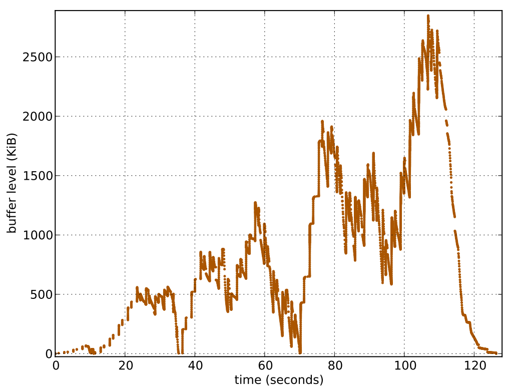
These models were compared with regards to the stalling frequency and duration in a test scenario. The resulting graphs are depicted in Figure 7. Further research is now required to gather measurement data from several videos and QoS conditions to be able to make conclusions on the viability of the models.



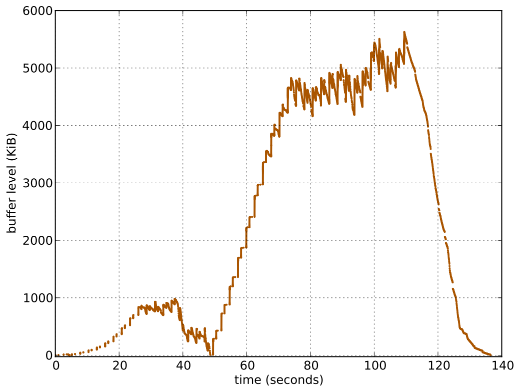
(a) Simple playback stalling method, 33s total stalling.



(b) Initial playback delay, 33s total stalling.



(c) YouTube Flash Player method, 34s total stalling.



(d) Firefox HTML5 method, 44s total stalling.

Figure 7: Modelled buffer fill level graphs and resulting total stalling times.

4 Research Schedule

Time Frame	***Work Item***
2010	Research of LTE & EPS specifications
	CDN Exploration
2011	Evaluation of simple HTTP streaming systems
	Modeling of buffering behavior
	Video streaming emulation
2012	Adaptive streaming exploration and emulation
	Evaluation of mobile core network traffic
	Cross-layer awareness investigation
	Simulation of video streaming in mobile networks
2013	Further performance evaluations and stochastic analyses
	Definition of streaming mechanisms optimized for mobile networks
	Written collection and presentation of the thesis' results
Q1 2014	Finalization

Table 2: Proposed thesis schedule.

As demonstrated, the ongoing work of this thesis is twofold, the application layer streaming and transport part and the mobile network component. Both can – to some degree – be tackled independently of each other. This is reflected in the thesis' proposed schedule in Table 2. When a sufficient understanding of both subtopics is reached one can then combine them and look at the bigger picture involved. Currently, work is being conducted on detailing the buffering behavior of simple HTTP streaming, partially already being published in [27]. Additionally, time is spent working on thematically overlapping ongoing projects.

The thesis should be concluded in the first or second quarter of 2014 with the last quarters being dedicated to the writeup and finalization work.

References

[1] P. Baran, "On distributed communications networks," *Communications Systems, IEEE Transactions on*, vol. 12, no. 1, pp. 1–9, 1964.

[2] "Sandvine: Sandvine's Spring 2011 Global Internet Phenomena Report Reveals New Internet Trends." [Online]. Available: http://www.sandvine.com/news/pr_detail.asp?ID=312

[3] "Entering the Zettabyte Era: Visual Networking Index." [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/VNI_Hyperconnectivity_WP.html

[4] L. Popa, A. Ghodsi, and I. Stoica, "HTTP as the narrow waist of the future internet," in *Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics*

in Networks, ser. Hotnets '10. New York, NY, USA: ACM, 2010, pp. 6:1–6:6. [Online]. Available: http://doi.acm.org/10.1145/1868447.1868453

[5] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550 (Standard), Internet Engineering Task Force, Jul. 2003, updated by RFCs 5506, 5761, 6051. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt

[6] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)," RFC 2326 (Proposed Standard), Internet Engineering Task Force, Apr. 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2326.txt

[7] J. Kurose, K. Ross, and B. Anand, *Computer networking: a top-down approach.* Pearson/Addison Wesley, 2008.

[8] L. Peterson and B. Davie, *Computer networks: a systems approach.* Morgan Kaufmann Pub, 2007.

[9] "3GPP TS 22.146 V10.0.0 Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service (MBMS)," 3GPP, 2011.

[10] "3GPP TS 22.246 V10.0.0 Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service (MBMS) user services; Stage 1," 3GPP, 2011.

[11] S. Alcock and R. Nelson, "Application flow control in YouTube video streams," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 2, pp. 24–30, 2011.

[12] K. Ma, R. Bartos, S. Bhatia, and R. Nair, "Mobile video delivery with HTTP," *Communications Magazine, IEEE*, vol. 49, no. 4, pp. 166–175, 2011.

[13] R. Pantos and W. May, "HTTP Live Streaming," 2011. [Online]. Available: http://tools.ietf.org/html/draft-pantos-http-live-streaming-06

[14] A. Zambelli, "IIS smooth streaming technical overview," *Microsoft Corporation*, 2009.

[15] S. Akhshabi, A. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," in *Proceedings of the second annual ACM conference on Multimedia systems.* ACM, 2011, pp. 157–168.

[16] P. Ross, "Cloud Computing's Killer App: Gaming," *Spectrum, IEEE*, vol. 46, no. 3, p. 14, march 2009.

[17] S. Wang and S. Dey, "Modeling and characterizing user experience in a cloud server based mobile gaming approach," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE.* IEEE, 2009, pp. 1–7.

[18] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "An Evaluation of QoE in Cloud Gaming Based on Subjective Tests," in *Workshop on Future Internet and Next Generation Networks (FINGNet-2011)*, Seoul, Korea, Jun. 2011.

[19] Austinat, R. and Fechteler, P. and Gieselmann, H., "Über den Wolken: Wie Cloud Gaming den Spielemarkt revolutioniert," *c't*, no. 21, pp. 76–83, 2010.

[20] E. Kohler, M. Handley, and S. Floyd, "Designing DCCP: Congestion control without reliability," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 27–38, 2006.

[21] I. Fette and A. Melnikov, "The WebSocket protocol," 2011. [Online]. Available: http://tools.ietf.org/html/draft-ietf-hybi-thewebsocketprotocol

[22] "WebSocket.org." [Online]. Available: http://websocket.org/

[23] M. Olsson, S. Sultana, S. Rommer, L. Frid, and C. Mulligan, *SAE and the Evolved Packet Core: Driving the mobile broadband revolution.* Academic Pr, 2009.

[24] "3GPP TS 23.228 V11.2.0 Technical Specification Group Services and System Aspects; IP Multimedia Subsystem (IMS); Stage 2," 3GPP, 2011.

[25] J. Gustafsson, G. Heikkila, and M. Pettersson, "Measuring multimedia quality in mobile networks with an objective parametric model," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on.* IEEE, 2008, pp. 405–408.

[26] I. Ketykó, K. De Moor, T. De Pessemier, A. Verdejo, K. Vanhecke, W. Joseph, L. Martens, and L. De Marez, "QoE measurement of mobile YouTube video streaming," in *Proceedings of the 3rd workshop on mobile video delivery.* ACM, 2010, pp. 27–32.

[27] F. Metzger, A. Rafetseder, D. Stezenbach, and K. Tutschku, "Analysis of Web-based Video Delivery," in *FITCE Congress, 50th, Proceedings*, 2011.

[28] R. Mok, E. Chan, and R. Chang, "Measuring the quality of experience of http video streaming," in *Proc. IEEE/IFIP IM (pre-conf.)*, 2011.