

Linköping Studies in Science and Technology

Dissertations. No. 1902

Efficient HTTP-based Adaptive Streaming of Linear and Interactive Videos

by

Vengatanathan Krishnamoorthi



Department of Computer and Information Science
Linköping University
SE-581 83 Linköping, Sweden

Linköping 2018

Copyright © 2018 Vengatanathan Krishnamoorthi

ISBN 978-91-7685-371-9

ISSN 0345-7524

Populärvetenskaplig sammanfattning together with Marcus Bendtsen

Printed by LiU Tryck 2018

URL: <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-143802>

Abstract

Online video streaming has gained tremendous popularity over recent years and currently constitutes the majority of Internet traffic. As large-scale on-demand streaming continues to gain popularity, several important questions and challenges remain unanswered. This thesis addresses open questions in the areas of efficient content delivery for HTTP-based Adaptive Streaming (HAS) from different perspectives (client, network and content provider) and in the design, implementation, and evaluation of interactive streaming applications over HAS.

As streaming usage scales and new streaming services emerge, continuous improvements are required to both the infrastructure and the techniques used to deliver high-quality streams. In the context of Content Delivery Network (CDN) nodes or proxies, this thesis investigates the interaction between HAS clients and proxy caches. In particular, we propose and evaluate classes of content-aware and collaborative policies that take advantage of information that is already available, or share information among elements in the delivery chain, where all involved parties can benefit. Besides from the users' playback experience, it is also important for content providers to minimize users' startup times. We have designed and evaluated different classes of client-side policies that can prefetch data from the videos that the users are most likely to watch next, without negatively affecting the currently watched video. To help network providers to monitor and ensure that their customers enjoy good playback experiences, we have proposed and evaluated techniques that can be used to estimate clients' current buffer conditions. Since several services today stream over HTTPS, our solution is adapted to predict client buffer conditions by only observing encrypted network-level traffic. Our solution allows the operator to identify clients with low-buffer conditions and implement policies that help avoid playback stalls.

The emergence of HAS as the de facto standard for delivering streaming content also opens the door to use it to deliver the next generation of streaming services, such as various forms of interactive services. This class of services is gaining popularity and is expected to be the next big thing in entertainment. For the area of interactive streaming, this thesis proposes, models, designs, and evaluates novel streaming applications such as interactive branched videos and multi-video stream bundles. For these applications, we design and evaluate careful prefetching policies that provides seamless playback (without stalls or switching delay) even when interactive branched video viewers defer their choices to the last possible moment and when users switches between alternative streams within multi-video stream bundles. Using optimization frameworks, we design and implement effective buffer management techniques for seamless playback experiences and evaluate several tradeoffs using our policies.

The work was supported by the Swedish Graduate School in Computer Science (CUGS), the Center for Industrial Information Technology (CENIIT), and the Swedish Research Council (Vetenskapsrådet).

Populärvetenskaplig sammanfattning

Internet och World Wide Web (WWW) har vuxit till viktiga hörnstenar i vårt samhälle. Vi använder dem för allt från kommunikation, bankärenden, utbildning, företagsverksamhet, arkivering till underhållning samt många andra applikationer. Vårt samhälle förändras ständigt av tillströmningen av nya applikationer som utnyttjar denna globala kommunikations- och informationsinfrastruktur. Vi har till exempel bevittnat hur e-post och program för snabbmeddelanden förvandlat mellanmänsklig kommunikation, och hur elektroniska nyhetsmedier har ersatt tryckta nyheter. På liknande sätt har videoströmning via Internet förändrat tv och underhållningsindustrin de senaste åren.

Även om flera videoströmningsprogram över Internet har existerat tidigare, så har dagens standard för strömning, HTTP-baserad Adaptiv Strömning (HAS), blivit så populär att den är ansvarig för den största andelen data som överförs via Internet. Studier har visat att över 70% av den totala nedladdade datamängden över Internet kan knytas till videoströmning. Denna andel förväntas att öka under de närmaste åren. HAS använder HTTP-protokollet för att ladda ner videodata, vilket är det protokoll som ursprungligen utvecklades för att ladda ner webbsidor. Genom att använda HTTP kan HAS utnyttja flera teknologier som utvecklades för webbtrafik, till exempel content distribution networks, cachar, serverkluster, etc. Dessutom stoppas inte heller vanlig webbtrafik som kommer från klienter av brandväggar hos nätoperatörer, eftersom de är kritiska för att deras kunder ska ha fortsatt anslutning till webben. HAS är också ett klientdrivet protokoll, vilket innebär att klienten måste begära all information som den behöver. Klientstyrd nedladdning, möjligheten att använda infrastruktur som utvecklades för traditionellt webbinnehåll samt anpassning av uppspelningskvalitet för att matcha den tillgängliga bandbredden är kanske de mest karaktäristiska funktionerna i HAS.

För att effektivt kunna skala till massiva datamängder krävs noga genomtänkta lösningar vid utformningen av nedladdningsalgoritmer på klientsidan och infrastrukturen för dataleverans. Den första delen av denna avhandling bidrar till detta forskningsområde. Först undersöker vi hur HAS klienter och cachar interagerar. Genom experiment identifierar vi både positiva och negativa aspekter av denna interaktion. Baserat på våra slutsatser föreslår vi nya HAS-medvetna strategier (där cachan försöker förutse vilken del av videoklipppet klienten kommer att begära härnäst) och samarbetsstrategier (där klienten och cachan delar information) som kan bidra till att förbättra tittarupplevelsen. Förutom en bra uppspelningsupplevelse förväntar sig tittarna också mycket låga uppstartstider (den tid det tar från att användaren valt att se en video till att den börja spela). I detta avseende har vi undersökt idén att hämta innehåll i förväg från rekommenderade videoklipp medan användaren tittar på en video. Genom att noggrant styra när dessa nedladdningar sker har vi också visat att uppspelningsupplevelsen av

den aktuella videon också kan förbättras. Nätverksoperatörer som tillhandahåller tillgång till Internet (både trådbundna och trådlösa) har i allmänhet möjligheten att finjustera sina nätverk för att hantera och optimera dem. Men eftersom de flesta strömmande tjänster idag använder den krypterad versionen av HTTP (känt som HTTPS) har operatören reducerats till en simpel kanal för strömmande data. I ett av våra bidrag visar vi hur en nätoperatör kan få information om klientens buffertförhållanden genom att samla in statistik av HTTPS-förfrågningar och svar. Genom att erhålla denna information kan nätoperatören bättre hjälpa klienter att undvika tillfälliga stopp i uppspelningen även om trafiken är krypterad.

Den andra delen av denna avhandling behandlar interaktiva strömmande tekniker med HAS. Interaktiv videoströmning är en ny applikation där tittaren får interagera med videospelaren. Interaktionen kan göras med klickbara objekt i videon, till exempel en kaffemaskin där tittaren kan bestämma vilken typ av kaffe en karaktär ska dricka med hjälp av knapparna på kaffemaskinen. Genom att tillåta tittaren att interagera vid specifika tillfällen i videon kan skaparen tillåta olika handlingar med potentiellt olika slut, baserat på tittarens smak och tycke. Vi är de första som formaliserar problemet med interaktiv förgrenad strömning med HAS och vi föreslår lösningar för optimerade nedladdningar, bufferhantering samt nedladdning av data i förväg. Detta gör att tittaren upplever en jämn uppspelning av videon. Möjligen av större vikt är att våra lösningar säkerställer att inga stopp i uppspelningen sker under den vanliga uppspelning eller under övergångar till nya grenar. För den intresserade tittaren vill vi upplysa om att Netflix släppte ett interaktivt avsnitt av "Puss in Boots" under 2017.

Denna avhandling bidrar också till att formalisera en annan interaktiv streamingteknik som kallas *multi-video stream bundles*. Här behandlar vi ett scenario där flera kameror täcker en händelse, till exempel ett sportevenemang eller en konsert. Med multi-video stream bundles får användaren möjlighet att själv välja från en mängd av strömmar. När användaren bestämmer sig för att se en ny ström erbjuds den nya vyn, men vid samma tidpunkt som för den föregående videon. Vi har utvecklat bufferhantering och lösningar för att hämta data i förväg som tar hänsyn till sannolikheten att en användare kommer byta ström, för att säkerställa att uppspelningen och övergången är fri från tillfälliga stopp.

Acknowledgments

Life as a PhD student is a fascinating journey of personal change and development. Yet, it is the people around you that have a large part in shaping this experience. Throughout my time as a PhD student, I have had the privilege of interacting with many that have made this journey possible and worthwhile.

This thesis would not have been possible without the help, support and guidance from my primary adviser Dr. Niklas Carlsson. Niklas has been a forthcoming source of vision, insight, and guidance that I could count on at all times. Throughout my time here, you have played a big part in all of my successes and have taught valuable lessons, both in research and with life. Thank you for believing in me and for always motivating me towards the next goal. You will always be a role model who I look up to.

I would also like to thank Prof. Nahid Shahmehri, my secondary adviser. Nahid has been instrumental in my development as a PhD student and has always kept a watchful eye over me. I thank you for giving me the opportunity, for showing genuine interest towards my education and my well-being, and for being an enthusiastic badminton partner.

Among our collaborators, I would first like to mention Dr. Emir Halepovic for his valuable additions to our joint works. Thank you for being a patient listener, and for the industry insights that you shared with us. I would also like to acknowledge and say thanks to Prof. Derek Eager and Dr. Anirban Mahanti for collaborating with us.

The Department of Computer and Information Science has been an excellent work place. The administrative and technical staff have been of immense help and support during my time here. I would especially like to thank Karin Baardsen, Anne Moe, and Inger Norén for help with various things over the years.

The Division of Database and Information Techniques (ADIT) has been my second home for the past five years. I would like to say thanks to all current and former staff at ADIT for having contributed positively towards the work environment. A special mention goes to Olaf Hartig, Rahul Hiran, Cyriac James, Rajaram Kaliyaperumal, Ulf Kargén, Huanyu Li, and Anna Vapen. I have had brilliant company over numerous lunches, coffee, and fika meetings in the form of Marcus Bendtsen, Zlatan Dragisic, Valentina

Ivanova, Patrick Lambrix, Jose M. Peña and Dag Sonntag. Lunch became an event to look forward to in your company, and went a long way in coping with day-to-day life.

This journey would not have been possible without the unwavering support of my family and friends. Thank you for always being there for me. My friends in Linköping have truly been a second family away from home. Last but not the least, I would like to say heartfelt thanks to my wife Dharshini for her endless love, support, and understanding. Having you beside me makes life all the more special and colorful.

Once again, I would like to thank you all and say that this has been an exciting journey, a great privilege, and an absolute pleasure! To those who part ways with me, I hope that the future shall present opportunities to collaborate and have fun together.

Vengat
January 2018
Linköping

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation and problem description | 3 |
| 1.2 | Contributions | 6 |
| 1.3 | Thesis outline | 8 |
| 2 | Background and related work | 11 |
| 2.1 | Historical perspective of video streaming | 11 |
| 2.1.1 | Video streaming over the web | 12 |
| 2.1.2 | Non-adaptive and quality adaptive streaming | 13 |
| 2.2 | HTTP-based streaming | 14 |
| 2.2.1 | Progressive downloading | 14 |
| 2.2.2 | Segmented downloading | 15 |
| 2.2.3 | HTTP-based adaptive streaming | 17 |
| 2.2.4 | Quality adaptation in HAS | 19 |
| 2.2.5 | Standardization and protocol trends | 20 |
| 2.2.6 | Live streaming | 21 |
| 2.3 | Factors affecting HAS performance | 22 |
| 2.3.1 | Access networks and characteristics | 22 |
| 2.3.2 | Quality of Service (QoS) and Quality of Experience (QoE) | 24 |
| 2.3.3 | Competing players and interaction with TCP | 25 |
| 2.4 | Large-scale content delivery | 27 |
| 2.4.1 | Peer-to-peer networks | 27 |
| 2.4.2 | Content delivery network | 27 |
| 2.4.3 | HTTP caching | 29 |
| 2.5 | Network-driven approaches to improve content delivery | 31 |
| 2.5.1 | Content provider based solutions | 32 |
| 2.5.2 | Network operator based solutions | 34 |
| 2.5.3 | Monitoring and guaranteeing QoE and QoS | 35 |
| 2.6 | The next generation of streaming services | 36 |
| 2.6.1 | Personalization on the web | 36 |
| 2.6.2 | Interactive and personalized video streaming | 37 |
| 2.6.3 | Multi-view video streaming | 39 |
| 2.6.4 | 3D, VR and AR streaming | 41 |

| | | |
|----------|--|---------------|
| 3 | Contributions and discussion | 43 |
| 3.1 | Publication list | 43 |
| 3.2 | Summary of papers | 44 |
| 3.3 | Discussion | 50 |
| 4 | Conclusions and future work | 53 |
| 4.1 | Summary and conclusion | 53 |
| 4.2 | Future work | 55 |
| | Bibliography | 57 |
| | Publications | 73 |
| | Paper I Helping Hand or Hidden Hurdle: Proxy-assisted HTTP-based Adaptive Streaming Performance | 75 |
| | Paper II Bandwidth-aware Prefetching for Proactive Multi- video Preloading and Improved HAS Performance | 103 |
| | Paper III BUFFEST: Predicting Buffer Conditions and Real- time Requirements of HTTP(S) Adaptive Streaming Clients | 131 |
| | Paper IV Empowering the Creative User: Personalized HTTP- based Adaptive Streaming of Multi-path Nonlinear Video | 163 |
| | Paper V Quality-adaptive Prefetching for Interactive Branched Video using HTTP-based Adaptive Streaming | 181 |
| | Paper VI Optimized Adaptive Streaming of Multi-video Stream Bundles | 209 |

Chapter 1

Introduction

The World Wide Web (WWW) began as a set of protocols, conventions, and software to organize and retrieve information over the Internet, based on pages containing hypertext. Today, the WWW is the most commonly used information retrieval service, which in addition to hypertexts includes graphics, audio, video, and plain text (commonly referred to as hypermedia). Transformations of the WWW has facilitated deployment of improved web-based services. Today's network applications and services over the web have come to dominate several aspects of our day-to-day lives, including education, business, banking, communication, and entertainment.

Video streaming is a technique that allows clients to start playback of a video before having downloaded the entire file. Such services were initially difficult to realize, since many networks did not have the capacity to satisfy the high bandwidth requirements of video. Deployment of such services were also hampered by a lack of efficient delivery architectures, video-encoding, compression, and distribution techniques.

By the time the first well known commercial streaming video players appeared (e.g., RealNetwork's RealPlayer¹, Microsoft's ActiveMovie² and Apple's QuickTime³ player), residential Internet speeds were for the first time sufficient to stream a low-quality video without stalls. Since then, much has happened, and today, video and audio entertainment services are being delivered to the masses over the Internet. The feasibility and common usage of these services can be attributed to the recent improvements in network bandwidths, the computational power available at end hosts, and the adoption of scalable content-delivery techniques. These improvements not only make efficient distribution of high volumes of data possible, but have also opened gateways to new and innovative services.

¹<https://en.wikipedia.org/wiki/RealPlayer>

²<https://en.wikipedia.org/wiki/ActiveMovie>

³<https://en.wikipedia.org/wiki/QuickTime>

On-demand streaming has become the largest source of traffic on the Internet. A recent study by Sandvine [1] suggests that over 70% of all downstream traffic over fixed access lines in North America can be attributed to real-time entertainment services, such as audio and video streaming. A majority of this traffic is from websites such as Netflix (35.5%) and YouTube (17.5%). Similar trends have also been observed in other continents.

Streaming services can be classified into on-demand and live streaming services. With on-demand services, the viewer can choose from a catalog of pre-recorded videos, and the video is streamed to a client only when an explicit request for the video is made. Live-streaming services stream live events over the Internet as they happen. Both on-demand and live services typically also offer functionalities such as pause, fast-forward (with live streaming, the amount of fast forwarding that is possible is limited), rewind and stop. As streaming becomes mainstream, events such as the Olympics, FIFA World Cup, other sporting events, news, and other mass media are disseminating their content over the Internet. Furthermore, several national and regional TV channels, such as BBC in the United Kingdom and SVT in Sweden make their shows available online for viewers within their respective countries^{4,5}. These online services break the barriers imposed by traditional broadcast services, where the viewer had to tune in to a channel at a particular time and could not control their current playpoint.

In the area of content delivery, service providers and network operators are constantly striving to decrease capital and operational expenses. The sheer volume of video data, the wide-spread popularity of streaming services, and the ever increasing user base of on-demand streaming [2] require current and future video streaming content delivery systems to be highly scalable and efficient. Such requirements combined with a very large reduction in storage costs, have driven the deployment of purpose built Content Distribution Networks (CDN) and edge caches in the access networks, for example, allowing popular content to be delivered from closer to the user.

The distributed nature of these techniques comes with many challenges. For example, efficient client-side, network, and server-side algorithms may benefit from information sharing between the involved parties, to help coordinate the available resources. However, there are many open problems related to how to best leverage such interaction or how to obtain the information when not all parties cooperate.

In addition to the challenges posed by traditional on-demand streaming services, which we refer to as regular/linear streaming, emerging services such as Virtual Reality (VR), Augmented Reality (AR), free viewpoint streaming, 360° streaming, etc., pose unique challenges where user satisfaction is strongly coupled to the service being interactive, with seamless viewing experiences, and without interruptions. Although residential Internet speeds have considerably increased, improved downloading and prefetching

⁴<http://www.bbc.co.uk/iplayer>

⁵<http://www.svtplay.se>

strategies are required to provide good Quality of Experience (QoE) to the viewer, while at the same time scaling to a large number of users in a cost effective manner. Besides from the technical factors mentioned above, socio-economic and human factors play important roles in the revenue model of such services. The combination of all these factors makes for a very interesting research field.

Emerging interactive on-demand streaming techniques present new opportunities for personalization and engaging user experiences. In addition to being on-demand, these services allow the viewer to interact with the video based on pre-defined or dynamic personalization options. This includes videos that provide users with interactive plot-lines, where the user chooses between alternative plot-lines, different viewing angles, or explore a scene interactively. Across such services and scenarios, user interaction takes the viewer to either a different point in time in the same video, to a new video, or present the viewer with a different view of the same event, for example. For these services, it is critical that the user does not have to wait to start playing after transitions to different story lines, new videos, or alternative views. Such seamless transitions are very important for both user engagement and satisfaction.

Interactive streaming applications have only recently began to emerge but have generated significant interest from mass media and the user communities. For example, websites such as Eko⁶, and others similar to it, allow users to create interactive storylines using a web-based editor. Here, certain objects of a base video can be annotated with clickable interfaces, allowing the viewer to click on these objects while watching the video. These and other forms of user input can then be used to personalize the plot sequence to the users liking during playback. Interactive videos based on annotations are also common in several YouTube channels, where the viewer can click on an annotation to go to a different video. However, in the case of YouTube this process is not seamless as the webpage has to reload and start playback of a new video. Similar extensions can also be derived using virtual-reality headsets such as the Google Cardboard⁷ or the Oculus Rift⁸. Here, the user may only be shown a portion of the viewable content, but the user can open up or explore new content by simply moving their head, looking at an object, or by interacting with a controller.

1.1 Motivation and problem description

Much increased residential bandwidths and cheap storage for content replication have enabled today's landscape of high-quality on-demand and live video streaming services, and further network improvements are expected to enable yet new streaming services, including various interactive services.

⁶<https://helloeko.com/stories/>, formerly <https://interlude.fm/>

⁷<https://www.google.com/get/cardboard/>

⁸<https://www.oculus.com/en-us/rift/>

While several methods have been explored to deliver video streams over the Internet, today, HTTP-based Adaptive Streaming (HAS)⁹ is by far the most popular. This family of streaming protocols utilizes the HyperText Transfer Protocol (HTTP) as the application level protocol. At a high level, HAS breaks the video into smaller chunks, each of which are encoded at multiple different bitrates. Using quality adaptive algorithms, each client then tries to determine the best possible encoding rate to download every chunk, so as to stream the video at a high quality without depleting the buffer (resulting in playback stalls) during playback. The clients also try to start playback of a video (startup delay) as soon as possible and try to avoid switching back and forth between quality levels too often. Analogous to the web, HAS is client-driven and takes advantage of components developed and deployed for the web. HAS content can easily be replicated, stored and delivered by caches and off-the-shelf HTTP servers, without the need for any specialized or proprietary software.

Today, all popular commercial streaming services use HAS. The increasingly common trend of television viewers resorting to on-demand, IP-based services for their entertainment is expected to continue changing the traffic patterns on the Internet. Predictions by Cisco [4] suggest that more and more Internet traffic will originate from CDNs and that the global IP video traffic will account for 80% of all IP traffic on the Internet. As HAS continues to consolidate its dominance as the main streaming technology, it is therefore important that both the current and the next generation of streaming services are scalable and efficient over HAS. Content caching and CDNs provide valuable tools here. However, while CDNs and content caches are easy to implement with HAS-based content, their impact on streaming performance is relatively unexplored.

The HAS content delivery chain consists of many parts (e.g., proxy caches, CDNs) that were originally developed for web traffic. While repurposing existing infrastructure and protocols helps reduce costs, HAS workloads differ significantly when compared to traditional web traffic. For example, HAS streams typically download much larger data volumes. Furthermore, the use of sequential downloading combined with quality adaptation makes HAS' download patterns substantially different than those observed for regular web users.

With many HAS-based services being delivered as over-the-top video, in which a streaming content provider offers video streaming service directly to a consumer and the video data (often encrypted) only transits the cable or Internet Service Provider (ISP) networks, the optimizations that can be done by an ISP or network operator have largely been neglected. To do such optimizations, the operator needs to have detailed information about the clients

⁹We will use the acronym HAS to refer to all HTTP-based Adaptive Streaming (HAS) solutions, including proprietary players such as YouTube, Netflix, Amazon Prime, etc., and new standards such as Dynamic Adaptive Streaming over HTTP (DASH) [3]. A detailed description of HAS is provided in Section 2.2.

and how the operator's actions can affect HAS clients and their policies. Naturally, as in any large-scale distributed system, information sharing and identification of actionable information can have a significant impact on the performance of the entire system. In the context of a network operator, by actionable information, we mean observations and measurements that can be used to take concrete actions to improve client performance. For example, if clients share their buffer sizes with network elements, or if techniques can be developed to identify characteristics of clients having low buffer sizes, this information can be used by the network to allocate additional resources or to take actions to help clients from experiencing playback stalls.

The ever changing landscape of Internet-based services has recently transitioned into yet a new phase, where personalized on-demand services are replacing the traditional broadcast-based entertainment services. As this significant change in the industry develops and matures, we will see many new services that require new delivery and download solutions to be developed. In the context of these next-generation services, there are many interesting and important research questions pertaining to problems related to content organization, optimized client-side implementations, efficient content delivery, and content caching that remain largely unexplored.

This thesis contributes towards addressing the following research questions in the areas of content delivery for (i) regular/linear HAS videos and (ii) interactive streaming services over HAS.

1. HAS has gained significant adoption by leveraging the architecture and protocols developed for the web, including content-caching and client-driven download semantics facilitating on-demand and live services. Despite these advantages, the interaction between HAS clients and network elements such as proxy caches are not well understood. Furthermore, HAS-aware caching and collaborative policies have not been explored in much detail. This thesis addresses open questions regarding how HAS clients and proxy caches interact, and designs and evaluates content-aware and collaborative caching techniques in the context of HAS.
2. Playback stalls and startup delays are important factors in determining user satisfaction with streaming services. Today's viewers are generally impatient and are known to switch between different movies or channels within a few minutes of viewing. With every new movie that is played, there is an associated startup delay. This thesis provides novel prefetching strategies for HAS clients that can significantly decrease the startup times of recommended videos without any degradation to the user satisfaction (playback quality and stalls) of the currently played video.
3. Although network operators provide the last-hop of connectivity, they do not have the tools or feedback mechanisms to understand how well HAS clients are performing within their network. Providing tools that

enable operators to do so can help in dynamic resource allocation and mitigating network-related issues that might impede viewer QoE. This thesis provides a framework that can be used to effectively estimate HAS clients' buffer conditions by only observing the encrypted network-level traffic (or traces thereof).

4. Interactive branched streaming¹⁰ allows users to select their own plot sequences through a video. Enabling the next generation of interactive services are important to realize several of the aforementioned growth projections. HAS is well suited for interactive streaming owing to the client-driven semantics, quality adaptation, and the way in which HAS videos are organized. This thesis presents the first exploration of quality adaptive interactive services over HAS and addresses some of the most important prefetching related questions and challenges in the area of interactive branched streaming.
5. Multi-video stream bundles is another class of interactive streaming application. Stream bundles offer the possibility of switching between different camera views of a stage or an event during playback. This thesis provides the first HAS-based multi-video stream bundle solution that allows clients' to switch between alternative streams seamlessly and use an optimization formulation to evaluate prefetching tradeoffs for such bundles.

This thesis addresses open questions in the aforementioned areas. In general, our research methodology relies on thorough evaluation and characterization of both existing and our proposed solutions through system implementations and real-world experiments. Whenever possible, our source codes have also been released for use by the research community.

1.2 Contributions

The main contributions of this thesis are in the area of efficient and adaptive techniques to deliver on-demand videos. We contribute towards improving the state-of-the-art of regular/linear video streaming over HAS, and interactive streaming over HAS. In general, we propose techniques to improve content delivery from the point-of-view of both the client and the network. In the following, we list the major contributions made in this thesis.

1. **Performance study of the impact that web proxy caches have on HAS clients. Propose and evaluate HAS-aware and collaborative policies for proxy caches.**
 - (a) Although HAS clients benefits from caches, the interaction between HAS clients and proxy caches is relatively unexplored and

¹⁰Traditionally, such services have been referred to as nonlinear media.

requires deeper investigation. We present a detailed evaluation under a wide range of network conditions and scenarios.

- (b) We propose content-aware proxy caches that keeps track of a HAS clients' progress through a video, and prefetches chunks at the quality that is most likely to be requested by the client.
 - (c) In addition to content-aware policies, we also design collaborative policies where a HAS client and the proxy share information, such as the current buffer occupancy and chunks available in the cache, respectively. With the help of this additional information, both the client and the proxy can make informed decisions on which chunks to download and their respective qualities.
- 2. Bandwidth-aware prefetching of recommended videos to provide instantaneous startup. Design, implementation, and evaluation of different prefetching policy classes.**
- (a) We propose client-based prefetching strategies that prefetch content from alternative videos during playback. By doing so, startup delays for alternative videos can be drastically reduced, while simultaneously helping the playback of the currently played video.
 - (b) We design and implement a prefetching framework that we use to evaluate and compare different classes of prefetching policies, that differ in how aggressive or opportunistic they are.
 - (c) By carefully controlling the time at which alternative videos are prefetched, we show that the currently played video can benefit from a larger throughput than it might have experienced without prefetching of alternative videos.
- 3. Network-based detection of client buffer conditions. Design, implementation and evaluation of a network-based buffer classifier that works with unencrypted and encrypted HAS flows.**
- (a) We propose framework called BUFFEST, that consists of an event-based buffer emulator that facilitates detailed emulation of clients' buffer conditions, an automated training module that trains online classifiers based on the emulated clients, and online classification of streaming sessions that use HTTPS using the trained models.
 - (b) The buffer emulator uses a man-in-the-middle approach to obtain access to HTTP payload information for HTTPS sessions. Although not scalable for real-time analysis, the buffer emulator can be used for detailed investigation of playback sessions and for training of more efficient packet-level classifiers. The buffer emulator is intended to be used with a subset of test streaming sessions over HTTPS.

- (c) We then propose machine learning classifiers that make use of the data obtained from the emulator to tag HTTPS packets. These classifiers can then work on real-time production traffic with access to only HTTPS packet headers and can classify client buffer conditions accurately.
4. **Design of an optimization framework for interactive branched video over HAS. Propose and evaluate classes of prefetching policies to provide stall free and optimized playback.**
- (a) We propose interactive branched video streaming over HAS and formalize clients' requirements to be able to stream branched videos without experiencing playback interruptions, regardless of how late the user makes the branch choice.
 - (b) We present the design and evaluation of an optimization framework that allows interactive branched video playback over existing HAS infrastructure with modifications only to the client-side player.
 - (c) We also present the design, implementation, and evaluation of prefetching and buffer management policies, which are developed based on our optimization framework that ensure seamless playback and transitions to new branches.
5. **Design of an optimization framework to determine optimized prefetching strategies for multi-view stream bundles over HAS. Evaluation of prefetching strategies for multi-video stream bundle through a prototype implementation.**
- (a) We propose multi-video stream bundles over HAS and formalize clients' requirements to be able to stream without experiencing playback interruptions when switching within the stream bundle.
 - (b) We present an optimization model and present both analytic and numeric insights into the characteristics of the optimized prefetching policies for alternative streams.
 - (c) Finally, we also present detailed results based on a proof-of-concept implementation of an multi-video stream bundle player where we show that our prefetching and buffer management solution can provide close to seamless playback when there is sufficient bandwidth to prefetch parallel streams.

1.3 Thesis outline

This thesis is organized around the main contributions as outlined in Chapter 1.2. To familiarize the reader, background and related works are presented in Chapter 2. Chapter 3 discusses the contributions of the different papers in this in thesis, followed by summary and conclusions in Chapter 4.

The research questions addressed in this thesis can broadly be categorized into two main areas. Papers I, II, and III make contributions in the area of content delivery for regular/linear videos. Paper I presents the evaluation and characterization of HAS clients in the presence of proxy caches and the design and evaluation of various proxy-assisted streaming techniques. Paper II presents our contributions in the area of bandwidth-aware prefetching of recommended videos to provide instantaneous startup. Paper III showcases the network-based framework for detection of client buffer conditions.

Papers IV, V, and VI make contributions in the area of interactive streaming services over HAS. In Paper IV, we present the motivation and design issues posed by interactive branched video streaming, followed by a discussion and evaluation of a simple prototype implementation of an HTTP-based interactive branched video player. Paper V presents a detailed formulation, implementation, and evaluation of the prefetching strategies and solutions required for interactive branched streaming over HAS. Finally, Paper VI presents an optimization framework based on the concept of multi-video stream bundles over HAS, followed by the proof-of-concept implementation and evaluation.

Chapter 2

Background and related work

2.1 Historical perspective of video streaming

When the first commercial streaming services emerged, the networks had comparatively larger Round-Trip Times (RTT) [5] and provided lower data rates¹ than what we see today. These limitations significantly impacted the design choices regarding the transport protocol.

Most of the video streaming services at the time were designed using User Datagram Protocol (UDP), rather than with Transmission Control Protocol (TCP). The choice to use UDP was typically motivated by the ability of the sender to transmit at the playback rate. However, UDP offers no guarantees of packet delivery, ordering, or error-correction. In contrast to UDP, TCP offers reliable connection-oriented services with congestion and flow control, but does not allow the sender to control the send rate, since the rate at which packets are delivered depend on the congestion and flow control mechanisms. While UDP can provide timely delivery of packets, the streaming services had to add functionality to overcome missing and out-of-order packets. To allow the player to recover from such situations without additional retransmissions, different error control and concealment techniques [6], forward error correction [7], and other schemes to mask imperfections were used. Such techniques were typically used with early streaming protocols such as Microsoft Media Server (MMS)², the Real Time Messaging Protocol (RTMP) [8], and the Real Time Protocol (RTP) [9] and its associated suite (Real Time Control Protocol (RTCP) and Real Time Streaming Protocol (RTSP) [10]) to mention a few.

¹<http://xahlee.info/comp/bandwidth.html>

²https://en.wikipedia.org/wiki/Microsoft_Media_Server

2.1.1 Video streaming over the web

The use of proprietary or UDP-based protocols had major limitations for streaming. Perhaps the biggest among these was that traffic over externally initiated UDP or non-standardized protocols are often blocked by firewalls and Network Address Translators (NATs), due to security concerns with connectionless services using UDP and due to risks posed by unknown protocols. Furthermore, several protocols at the time required the servers to track the client state continually, requiring dedicated infrastructure, and intelligence at the server-side as well. These limitations together with faster internet speeds (that allowed client buffers to be filled quickly), slowly started to outweigh the benefits of using UDP-like protocols.

The WWW has grown tremendously in the last decade³. Network operators and content providers developed and deployed technologies such as CDNs and caches to scale up to increasingly larger user bases. Several of the largest websites and other major companies have built CDNs, consisting of vast interconnected server networks linked via Internet eXchange Points (IXP) which can be used to deliver content globally. Also, a multitude of web caches have been deployed, which store a copy of webpages that are delivered via them. These local copies, allow for significant reduction in fetch times of future requests for the same webpages. Both CDNs and caches were developed specifically for web-based traffic.

The WWW uses the Hypertext Transfer Protocol (HTTP) as its application layer protocol and the TCP protocol to ensure that all bytes of a webpage are (eventually) delivered. The development and deployment of CDNs and proxy caches significantly reduced the cost incurred by the content provider and the network operator in delivering data to end users.

Incremental improvements to access speeds [11], [12] and round-trip times [5] over the years have made streaming over TCP a feasible alternative. Furthermore, as client-side computational power and storage capacity improved, the clients could use a larger buffer to accommodate for short-term fluctuations in the network bandwidth. The comparatively larger buffer also provides additional time for TCP's error correction and recovery protocols to recover packets in time for playback [13]. These developments, in addition to improvements in delivering web content through CDNs, large deployments of proxy caches at the network's edge, and NAT/firewalls not blocking client-driven TCP traffic lead to the adoption of streaming over HTTP [14]. In contrast to the earlier streaming protocols, HTTP and therefore streaming over HTTP is entirely client driven. This significantly reduces the complexity required at the server-side, helps the entire system to scale better; e.g., by making use of CDNs and proxy caches and even allows the possibility to download streams from multiple servers in parallel. Finally, deployment and licensing costs of HTTP servers are much lower when compared to deployment of proprietary servers.

³<http://www.internetworldstats.com/emarketing.htm>

2.1.2 Non-adaptive and quality adaptive streaming

Several techniques have been explored previously to efficiently distribute multimedia content. Many of the first scalable solutions were based on multicast and broadcast domains [15], [16], [17], [18]. These methods provide much better server-side scalability than unicast solutions, by only requiring a single multicast stream to reach many users. With IP-based multicast, the distribution trees are dynamically built using management protocols such as the Internet Group Management Protocol (IGMP), where a client or set-top-box wishing to connect to a certain broadcast connects to a router and subscribes to a distribution tree. This architecture requires the network core to maintain state, and wide-area deployment has been limited.

Scalable Video Coding (SVC), an extension to the H.264 Advanced Video Codec was envisioned as an adaptive scalable video delivery codec [19]. SVC supports adaptive screen sizes (spatial resolution), adaptive frame rates (temporal resolution), and bit rates (quality resolution). This makes it possible to adapt content based on different modalities where the spatial, temporal, or quality resolution might be adapted on the fly in real-time applications [20]. SVC requires only one video encoding for several adaptation profiles. Video streams in SVC can be split into a base layer and several adaptation layers. For example, in the multicast context, the viewer can subscribe to multiple multicast trees, based on their available bandwidth. The client can receive these layers in a best-effort manner and reconstruct content on the fly to adaptively present the video. However, the process of generating several adaptation layers involves a significant coding penalty [21], limiting their practical use.

Before today's **rate adaptive HAS protocols**, several rate adaptation mechanisms have been proposed. Some of these were client-driven and others were server-driven. The server-driven mechanisms typically relied on feedback messages (conveying application layer metrics [22] or transport-layer information [23], [9] about bandwidth, buffer occupancy and other parameters of interest [24]) from the clients. In general, server-driven rate adaptation mechanisms have the disadvantage that the control loop always lags behind the network condition by a factor of at least one RTT. The delay might increase substantially in cases where there are severe network congestion or packet losses. In addition, these mechanisms also require the server to maintain state information for every active client, thereby occupying more server resources.

Similar to the SVC example mentioned before, several client-controlled adaptive multicast protocols have been explored, in which the clients subscribe to one or more multicast groups [25]. To adapt to the current conditions, these protocols typically add or remove enhancement layers to a video based on the relation between the consumption rate and the available bandwidth [26] or just adapting the number of layers that the client is currently subscribing to [27].

Currently, the most commonly used format for video content is H.264/AVC, where AVC stands for Advanced Video Coding, that is downloaded over HTTP. H.264/AVC does not use multiple adaptation layers, but rather a single video stream that the client needs to download completely for playback. HTTP is a pull-based protocol, where the process of streaming a video over HAS is dictated by the clients' request. HAS videos are split into smaller pieces called chunks, which are quality adaptively downloaded by the clients. Hence a large significance is placed on the client-side algorithms that are used to perform quality adaptation. We describe the process of streaming over HTTP in more detail in the following section.

2.2 HTTP-based streaming

With HTTP, clients typically connect to port 80 on a server. Having established a TCP connection, clients then request or transfer data from/to the server using standard HTTP methods such as GET (used to retrieve content) and POST (to upload data to the server).

2.2.1 Progressive downloading

The first generation of HTTP-based streaming players requested the entire video using a single GET request. However, videos of even a few seconds in duration are considerably larger than an average webpage and the embedded images and animations that HTTP originally was designed to transfer. To avoid having to wait for the download to complete, clients therefore typically began playback before completing the download. This technique is referred to as progressive download.

Although progressive download works well for short video clips, and when the viewer watches from the beginning of a video to its end, there are several issues when taking into account **a typical user's behavior**. For example, studies have shown that users seldom watch videos from start to end [28]. In fact, **viewers often navigate to parts of the video which they consider interesting**. Under such use cases, a client which progressively downloads a video stream would have to wait for a long time if the viewer decides to seek towards the end of the video as soon as playback commences. Another common use case is that viewers watch the beginning of several videos before settling down to watch a video completely [29]. In cases where a viewer decides to abandon watching a particular video, there is no benefit to download data which lies a few seconds beyond the current playpoint. In both the aforementioned cases, a large portion of the downloaded data might not have been used for playback. This leads to wasted bandwidth and resources at both the client and the server side, potentially forcing the content provider to deploy additional server replicas to meet the bandwidth demands.

2.2.2 Segmented downloading

To overcome the apparent shortcomings of progressive download video streams, commercial solutions use either a chunk-based or range-request-based system. Both these systems are quite similar, but differ slightly in the way in which the video is represented and requested from the server.

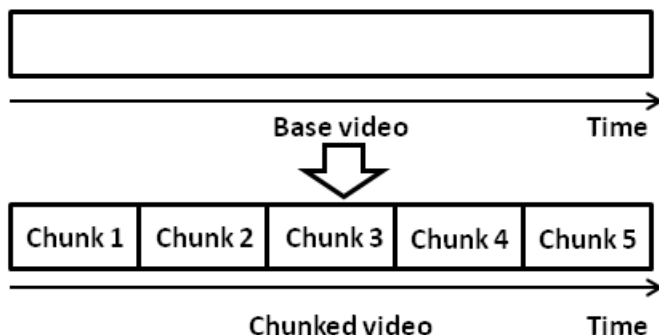


Figure 2.1. Generating chunks from a base video

A chunk-based system divides a base video into smaller pieces called chunks, as shown in Figure 2.1. Successive chunks are continuations of the original video byte stream, with chunk boundaries based on the playtime of each chunk, rather than volume. Typical chunk durations observed in the real-world are between 2-10 seconds long. These values vary between different services and even between two videos in some services. Once the chunked video is generated, each chunk is assigned a unique Uniform Resource Locator (URL). Generally, the assigned URLs are based on the URL which identifies the video, and a number is typically added as a suffix to indicate the relative chunk position within the stream.

In contrast to a chunk-based system, a range-request based system does not partition the byte stream into chunks, but instead, relies on HTTP range-requests. HTTP range-requests require that the server accepts such requests, which is published via the **accept-ranges** response header. With such requests, the client can independently request any sequence of bytes in the requested object by using a start and an end byte value.

Video chunks or range-requests are generally generated in a way such that the beginning of each chunk or range request aligns with an I-frame. Here, I-frame stands for Intra-coded frame. These frames are fully specified; i.e., the decoder can reconstruct this frame on the screen without any additional information. Since I-frames contain all information about a scene without dependencies to earlier or later frames, they are considerably larger compared to P-frames (Predicted frame) and B-frames (Bi-predictive frame) which can refer to frames before and after it.

In contrast to progressive downloads, both the chunk-based and range-request-based systems require the client to have some additional information about the video. In a chunk-based system, the client must at least know the chunk length, the number of chunks that are available, and the naming convention used for that video. Similarly, a range-request-based client must know the first and the last byte of a video stream, the mapping between play times and bytes in the video, and that the protocol stack supports HTTP range-requests, otherwise known as byte serving. With these systems, a manifest file, also called as the Media Presentation Description (MPD), is used to send bootstrap information to the client. In general, the data contained in the manifest can be used to map a certain playtime of the video to a specific chunk or a byte-range. Whenever playback of a video is initiated, the MPD file is sent to the client along with player binaries and other objects that are necessary to bootstrap the client. There might also be additional configuration information about the adaptation logic and codec related information in the file. Once the MPD has been parsed, the client is expected to act according to the conventions of the service and start downloading the video for playback.

Using chunks or range-requests overcomes the aforementioned drawbacks of progressive downloads. When a viewer decides to seek to a new playback point, the player can now request for a chunk or a range of bytes that corresponds to the play-time requested by the viewer. This eliminates the unnecessary downloading of data between the current playpoint and the new playpoint. With chunk and range-request-based players, maximum buffer sizes can now be controlled, as the request for each chunk or byte-range can be used to control how much future data is available in the buffer.

To exemplify the operation of such protocols, let us now describe a chunk-based player and its operation. First, we let the maximum buffer size (T_{max}^{buf}) be defined as the maximum number of seconds of video that the client can locally store in its buffer. Once T_{max}^{buf} is reached, the player can be instructed to not request future chunks. By using another threshold, called minimum buffer size (T_{min}^{buf}), which corresponds to the minimum value that the buffer should ever reach during normal operation, the buffer can be allowed to drop until T_{min}^{buf} is reached. Once this threshold is reached, new requests are sent again until T_{max}^{buf} is reached. Using this simple technique, buffer sizes can be regulated to always remain between these two thresholds. The size and difference between these two values must be carefully considered to allow for stall free playback, with minimal bandwidth and resource wastage.

Controlling the amount of data available in the buffer through thresholds, results in two distinct phases of operation being observed in steady-state. Whenever the buffer occupancy equals or exceeds T_{max}^{buf} , the client does not place any requests until the buffer occupancy drops to T_{min}^{buf} . Since T_{max}^{buf} and T_{min}^{buf} are actually set in terms of play-time (seconds) the player remains in this state for the duration given by $(T_{max}^{buf} - T_{min}^{buf})$. Similarly, the client would download chunks whenever the buffer has fallen below T_{min}^{buf} and has not yet

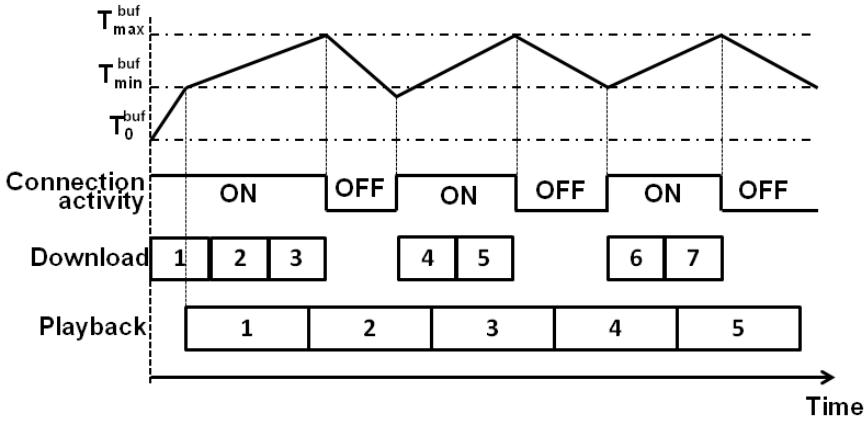


Figure 2.2. Typical download pattern of HTTP-based players

reached T_{max}^{buf} . In popular literature, these two states are often called the *off* and *on* states, respectively. Figure 2.2 shows a diagrammatic example illustrating these two states, the associated buffer occupancy, as well as the per-chunk download and playback details for a chunk-based system.

2.2.3 HTTP-based adaptive streaming

HTTP-based Adaptive Streaming (HAS) is currently the most widely used standard to deliver video streams over HTTP. HAS is very similar to the chunk and range-based streaming over HTTP. However, in addition to on/off pacing (to control the buffers) and the use of chunk/range-requests, a HAS stream has multiple encodings of the same video available on the server. Furthermore, the client adaptively switches between different qualities of the video (and audio) chunks (from the set of qualities available) as it is progressing from one chunk to the next, so as to adapt to the current bandwidth and buffer conditions. Figure 2.3 shows a representation of multiple encoding rates of a base-video organized into chunks whose boundaries align perfectly.

Some HTTP-based players allow the clients to manually chose an encoding based on their expected needs. This is not truly adaptive streaming. In true HAS videos, the chunks are synchronized across encoding-rates (e.g., as shows in Figure 2.3) **so that, the start and end time of chunks are exactly the same across all available encoding-rates the video is available at.** This allows **the client to run rate-estimation and quality adaptation algorithms to determine the best quality** at which the next chunk should be requested. The qualities are selected in such a way that the clients are expected to achieve high playback quality, while avoiding playback stalls.

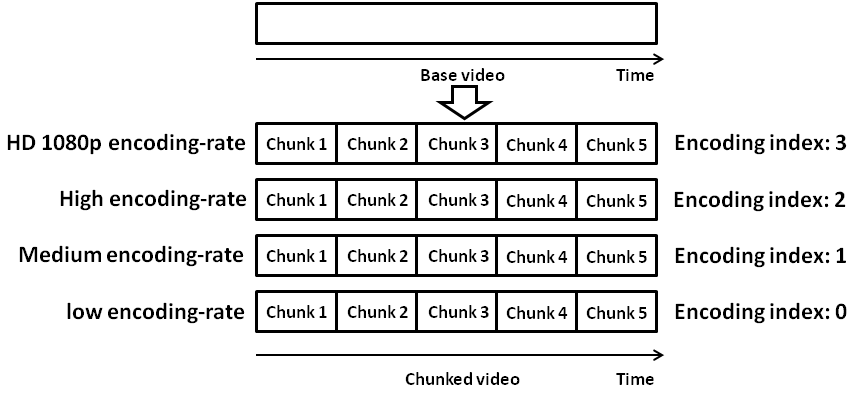


Figure 2.3. Multiple chunk encoding-rates in a HAS video

Quality Adaptation (QA) algorithms are used in HAS to determine the quality at which the next chunk is downloaded. Since chunk boundaries are aligned across all encodings of the video, the client, if necessary, can choose to playback chunks at any quality, irrespective of previous quality choices. To estimate the network throughput, HAS players typically track the time at which each request is sent to the server ($T_{request}$), the time at which the chunk was completely received ($T_{response}$) and the size (S) of the chunk or range-request. Using this information, the client can calculate the average download rate of this chunk as $S/(T_{response} - T_{request})$. Based on simple throughput-based metrics, such as the one described, QA algorithms can adapt the quality of the next chunk such that it is likely to be downloaded before its playback deadline. Whenever the observed download rate suggests that downloading chunks at the current encoding rate would lead to playback deadline violations, chunk requests can be made at lower encoding rates to avoid playback stalls. Similarly, the encoding rates can be appropriately increased in cases when the observed download rate suggests the opposite. In fact, similar to non-adaptive HTTP-based streaming, an additional buffer T_{min} is typically maintained to avoid stalls.

The rate-estimation and quality adaptation procedure discussed above is very simplistic. It is well known that network throughput during a playback session can vary drastically over both short and long time durations (e.g., due to competing traffic, TCP dynamics, and dropped packets). Throughput variations can be especially pronounced in the case of wireless access techniques such as Long-Term Evolution (LTE) and WiFi, where additional factors such as mobility, interference, and environmental factors can negatively impact the throughput. Therefore, rate-estimation based on the observed download rate of a single chunk can be an unreliable predictor of future throughput. In practice, the expected download rate is instead often extrapolated either based on a rolling window or a weighted average of sev-

eral previous chunk downloads. For example, an Exponentially Weighted Moving Average (EWMA) to calculate the estimated available bandwidth as $BW_i = (1 - \alpha) \cdot BW_{i-1} + \alpha \cdot (S / (T_{response} - T_{request}))$, where BW_{i-1} is the estimated bandwidth during the previous iteration of the weighted average calculations, and the weight α determines the weight given to the newly observed download rate. The Open-Source Media Framework's (OSMF) HAS player uses the above described EWMA to estimate the available bandwidth.

2.2.4 Quality adaptation in HAS

A good estimate of the available bandwidth is important for effective quality adaptation. Design of QA algorithms has garnered significant attention from the industry and the research community. At a high-level, the partially conflicting goals of QA algorithms are to ensure that the clients do not experience any playback stalls, start playback quickly, do not fluctuate between quality levels too often (with competing clients and varying network throughput), and the video is played at as high a quality as possible.

Most QA algorithms in practice use a bandwidth-based approach (as described above). These approaches make use of segment or chunk fetch times [30] and calculate smoothed averages over a history of chunk fetch times, which is used as a predictor of the future bandwidth. In addition to robust measurements, prior works have also proposed randomizing chunk download times [31] to remove periodicity in HAS clients, and probing the network from time-to-time to proactively identify the available bandwidth, rather than reacting to change in bandwidth [32].

As alternatives to the bandwidth-based approach, buffer-based approaches to QA have also been proposed. These algorithms [33], [34], typically do not use the measured bitrate to determine chunk qualities, but use the buffer occupancy and the change in buffer size to determine the quality of the next chunk. Whenever the buffer is sufficiently high, clients might request a large encoding rate, while with small buffer sizes, lower qualities are chosen to quickly increase the buffer. These algorithms are gaining adoption with commercial streaming players; for e.g., Netflix and dash.js both use a buffer-based algorithm to perform QA.

This thesis largely presents HAS clients in terms of chunk-based players. With chunk-based HAS, every encoding of the video is split into smaller pieces that are identifiable by a unique URL. However, range-request based HAS clients are also prevalently used. Naturally, the manifest files used by these clients contain different bootstrap information. Although chunk-based and range-request based clients use similar QA algorithms and are considered equivalent, one inherent benefit with range-request based clients is that these clients can request for different ranges of bytes in their requests (where a single request can download either multiple or parts of chunks), however, this can affect the effectiveness of caches if they cannot parse the requested URL and the response.

QA algorithms are also responsible for a HAS clients' startup behavior. Most HAS players start playback of a video only after the buffer exceeds a certain threshold. As it is preferred to keep startup delays to a minimum, the threshold to start playback is generally kept as low as possible. However, by choosing too small a value for this parameter increases the chance that the client might encounter stalls immediately after playback has began. Given that stalls during the beginning of playback sessions are much more likely to lead to abandonment, the value has to be carefully chosen such that the startup delay is as small as possible, while at the same time ensuring stall free playback.

In most commercial players, the QA and the measurement algorithms are constantly fine-tuned during run-time. Several players use a smaller T_{max}^{buf} when the measured network conditions are poor. By doing so, the player avoids downloading too many chunks at low encoding rates in low-bandwidth situations. Similarly, the OSMF player keeps track of the history of playback stalls, which conservatively factors down the chosen encoding rate for the next chunk. Other metrics that are used by commercial players include the number of dropped frames (can indicate that the CPU or GPU is unable to decode video frames), screen size, aspect ratio, state of the player (full screen vs minimized), platform (App vs browser vs TV), etc.

The most common HAS clients of today are Apple's HTTP Live Streaming (HLS)⁴, Microsoft's Smooth Streaming (MSS)⁵, Adobe's HTTP Dynamic Streaming (HDS)⁶, DASH implementation for VLC [35], the DASH industry forum's dash.js⁷ and ExoPlayer⁸. Among these Adobe HDS, VLC DASH, ExoPlayer and dash.js are open source implementations, while the others are proprietary. HAS is specified as multiple standards in the Motion Picture Experts Group- Dynamic Adaptive Streaming over HTTP (MPEG-DASH) [3] specifications and in the 3rd Generation Partnership Project (3GPP) [36]. These standards are loosely based on several commercial implementations that are detailed above.

2.2.5 Standardization and protocol trends

The Internet Engineering Task Force (IETF) has recently released the HTTP 2.0 specifications [37], largely based on SPDY [38], which is intended to provide improvements to the WWW in general. However, HTTP 2.0 has several additional features over HTTP 1.1, including stream termination, server push, response multiplexing, and other features that can be beneficial in HAS scenarios. For example, the server push feature can be used to push chunks to the client that are most likely to be requested next, without having to wait for a request from the client, while stream termination can be

⁴<https://en.wikipedia.org/wiki/QuickTime>

⁵<http://www.iis.net/downloads/microsoft/smooth-streaming>

⁶<http://www.adobe.com/products/hds-dynamic-streaming.html>

⁷<https://github.com/Dash-Industry-Forum/dash.js/wiki>

⁸<https://github.com/google/ExoPlayer>

used to close or terminate a request at a high encoding rate when the client determines that encoding rate to be unsustainable [39]. In addition to new application layer protocols, improvements to the transport layer have also been sought. For example, QUIC (Quick UDP Internet Connections) [40], is a UDP-based protocol that has been designed to provide features such as congestion control (implemented at the application layer), connection oriented semantics, multiplexing with reduced latency, and includes integrated security features and pluggable congestion control algorithms. Furthermore, both QUIC and HTTP 2.0 are designed for interoperability between one another. Finally, we note that TCP friendliness and improved loading times of QUIC have also been documented [41].

The growing trend of streaming video replacing broadcast television at households and increased mobile adoption is continually placing greater demand on the content delivery infrastructures. Although several aspects of delivery networks are over provisioned, the sheer scale and volume of traffic makes it increasingly difficult to design and improve the current state-of-the-art. Large-scale measurements have shown that more than 20% of all streaming sessions have re-buffering ratios greater than 10%, and that over 14% of the sessions have a startup delay of more than 10 seconds [42]. Evaluation of today's QA algorithms have identified potential improvements that could be implemented in the client-side as well [43].

2.2.6 Live streaming

HAS is also used to perform live streaming. Compared to traditional live or broadcast television, where the video is pushed to the client, live streaming over HAS relies on the client “pulling” the content from the server. Although HTTP 2.0 supports features such as server push, we detail the current practice of live streaming over (pull-based) HTTP.

At a high level, live streaming over HAS is very similar to on-demand streaming [44], where the client initially downloads a MPD file, parses it, and initiates playback by requesting chunks one after another. Under a live scenario, any new client must be able to decipher the ‘live’ chunk, i.e, the chunk which must be downloaded to start playing the live stream from an MPD file. This is achieved by continually updating the MPD file. Clients might also have to refresh their MPD files at times when some parameters of the stream change. Chunks are generally numbered incrementally; thus, the client can also rewind to an earlier point in time. A client side buffer is used to guard against download jitter and throughput variations. Clients are staggered with relation to the actual live stream based on their buffer sizes and the download follows an on-off download pattern. In addition to the fairness and stability issues faced by on-demand clients, it has been shown that under live streaming over HAS, multiple clients can easily become synchronized, thereby leading to resource utilization issues [45]. Solutions to tackle such problems include randomly distributing requests over time.

2.3 Factors affecting HAS performance

In this section, we discuss the most common network-related and user-related factors that might affect HAS performance.

2.3.1 Access networks and characteristics

Wired access networks: The most reliable and common way to access Internet services has long been over wired networks, with recent statistics suggesting that wired networks still contribute to the majority of the IP traffic (51%) [12]. The most common wired accesses at home and office networks is through the Ethernet standard specified by IEEE 802.3. Recent iterations of the 802.3 standard provide theoretical access speeds of 100 Gigabit/s (IEEE 802.3ba-2010) over optical and electrical cables. When compared to most other residential and commercial access technologies, Ethernet provides the most stable operating characteristics. Transmissions over such networks typically guarantee Bit Error Rates (BER) of 10^{-12} or lower [46].

Ignoring packet drops due to congestion, the guaranteed BER of less than 10^{-12} corresponds to negligible packet loss rates. In the absence of persistent congestion in such networks, the higher layers can expect consistently stable download performances. Perhaps the biggest advantage of accessing content over a closed medium like Ethernet is the absence of competition/interference from other sources that are not within the system. In the context of HAS, this helps provide stable, high-quality playback.

Wireless access networks, WiFi: Wireless Fidelity (WiFi) is a radio access technology specified in the IEEE 802.11 family of standards operating in the 2.4 GHz and the 5 GHz bands. Some of the most recent amendments to the standard have speeds ranging from 400 Mbit/s to several Gbits/s [47]. As of 2016, there were 94 million public WiFi hotspots worldwide and by 2021, it is expected to be 514.6 million public hotspots [12]. WiFi-based traffic already corresponds to 41% of the overall IP traffic globally, and wireless accesses (WiFi and mobile) are expected to grow beyond wired accesses by 2021, reaching a projected share of 63% of the global IP traffic.

The frequency band used by WiFi falls in the Industrial, Scientific and Medical (ISM) radio band, which is an internationally reserved, unlicensed radio band for civilian use. Although cost reduction from using an unlicensed frequency band has been a significant factor in increasing the adoption of WiFi, this band is becoming increasingly crowded as there are multiple standards and technologies that operate in this band. For example, Bluetooth, cordless phones, Near Field Communication (NFC), Microwave ovens and several medical applications use this band in conjunction with WiFi. In addition to interference, wireless transmissions have to cope with signal attenuation and fading due to objects that might be present in the environment. Although standards are developed to take into account the presence of interfering sources, in general, the BER in wireless accesses like

WiFi are expected to be at least an order of magnitude worse than wired networks, and in reality, they could be up to several orders of magnitude worse depending on the conditions.

Wireless access networks, LTE: In conjunction with WiFi, LTE and the future iterations of cellular/mobile standard are expected to be the most common way of accessing the Internet. LTE is specified as a standard by the 3rd Generation Partnership Project (3GPP). In contrast to WiFi, LTE uses licensed spectrums, which differs from country to country. While operating over a licensed spectrum has benefits in terms of resource allocation and interference, deploying and updating infrastructure requires heavy capital and operational expenses. Compared to WiFi, LTE operates over much larger wireless distances. Owing to the large expenses incurred by LTE operators, in general, users are limited to a certain volume of download and upload data per month. Presently, for a large majority of global subscribers, the monthly data-limits are too restrictive to stream high-quality content consistently. However, the projected increase in mobile adoption is expected to bring down the associated costs. Recently, there have also been multiple proposals for LTE to operate under the 5 GHz ISM band [48], [49].

Round-trip times: The RTT is an important factor for a HAS clients' performance. An RTT is defined as the time it takes from a packet being sent until receiving an acknowledgement for the same packet. The steady-state throughput of a TCP connection has been found to be approximately proportional to the ratio of the window size (W) and the RTT [50]. While this model ignores explicitly including the packet loss rate as a variable, TCP window size in general, depends on how congested the network and how many packets are generally lost due to other reasons.

Naturally, under most typical use cases, wired accesses would be expected to have the lowest RTT and therefore potentially the largest throughput. Interestingly, the difference between WiFi and LTE has become much smaller in the recent years, with LTE outperforming WiFi in certain cases [51] in terms of RTT and throughput. Publications by prominent 3GPP participants such as Ericsson [52] suggest that future LTE networks would be able to support extremely low latencies too.

Other factors: In addition to access technologies affecting RTT, server or replica placement can have a significant impact on HAS clients. In general, the most popular streaming services use a front-end server that might have a national domain (for example .se or .in). These servers typically index the catalog available for that region and present the content to the user. Once the user decides to watch a specific video, the service then assigns the user to one among several server replicas. The choice of how a CDN or replica is chosen varies from service to service [53]. In general, the process of identifying the right CDN node and the correct replica for every client

can be challenging. In this regard, CDN-ISP collaboration to enable better informed user-server assignment have been suggested [54], [55].

Popular services such as Netflix, who deploy their own CDNs, have started placing custom-built caches and servers within an ISP's network or at Internet eXchange Points (IXP) [56]. Clients accessing streaming content within provider networks that collaborate can now be assigned to the replica server that is placed within the ISP network. Such agreements benefit the clients, the ISP, and the streaming service.

2.3.2 Quality of Service (QoS) and Quality of Experience (QoE)

Ultimately, the end-users' subjective perception of the video that is being streamed determines the utility generated by the delivery chain. Therefore, most of the costs incurred within the network must thus (indirectly) be paid by the end-users. It is therefore important to understand the clients' Quality of Experience (QoE). QoE is a subjective metric that captures the degree of delight or annoyance of the user of an application or service⁹. QoE can change from person to person for the same video, based on the user's personality and current state of mind. Furthermore, it has been argued that identifying the right set of QoE metrics for a given service is hard and can change based on several factors including user diversity [57].

Quality of Service (QoS) corresponds to the network level service guarantees that can be given to a client, while QoE is a measure of user-related happiness/satisfaction due to network behavior and compliance with the network-level guarantees [58]. HAS players can protect clients from short-term network degradations (such as packet drops or losses) through the use of a playback buffer (typically these buffers in the order of several seconds to a few minutes in length). However, long-term outages or network throughput degradations can still have a pronounced effect on HAS clients.

Several prior works have looked at measuring the QoE of streaming sessions based on playback artifacts observed on clients; e.g., number of stalls, stall time, quality switches, and startup delay. Irrespective of the viewer device or viewer, rebuffering events or playback stalls [28], [59], [60], [61], [62] have the most significant impact on user satisfaction during streaming. Other user activity related information can also be used to measure QoE, including pausing the video, and manually reducing quality [63].

QoS is typically much easier to measure and control in a network. The most common QoS metrics include latency, jitter, throughput, and packet loss rates. These metrics are relatively more objective than QoE metrics. However, video streaming services are becoming so pervasive that standardization efforts by organizations like 3GPP, for example, are looking to collect metrics from the clients for use by network elements to direct QoS adaptation so as to help in improve QoE [64]. In addition, several streaming

⁹<http://www.qualinet.eu/>

analytics services have emerged, that use a third-party plugin service within the player to collect information from streaming clients.

There are also several non-QoS related factors that might impact the QoE. For example, due to the wide-spread adoption of mobile devices, content creators and streaming websites must also accommodate for devices having different screen sizes (ranging from 9 to 200 cm) and aspect ratios (e.g., 4:3, 3:2, 8:5, 5:3, 16:9). In cases where aspect ratios do not match, the video player is forced to display black bands at the edges of the video, which might decrease the portion of the screen where the video is being played, and therefore negatively affect the QoE.

It is becoming common that services use different players (which in turn might have different QA algorithms) based on the platform from which the service is accessed. For example, within Android platforms, services are encouraged to use native frameworks such as the ExoPlayer¹⁰ while with Apple devices, HTTP-Live Streaming (HLS)¹¹ is recommended to build clients. Similarly, several native and third-party frameworks also exist in the PC market for all major platforms (e.g., Windows, Mac, Linux). In addition, cross-platform solutions like HTML5-based playback are also gaining popularity. These variations have to be accounted for by the streaming provider to guarantee consistent performance across platforms.

Finally, differences in the billing models used by network providers can affect clients' streaming experiences. In most countries, wired Internet accesses are not metered or allow large enough volumes that the clients consistently can attempt to stream videos at the highest possible quality. However, with wireless accesses, such as LTE, users are often billed a base amount for a certain monthly bandwidth budget and are charged more if they exceed the limits. This disparity means that mobile streaming clients have to manage their quality choices in accordance with their data plans. This has prompted several streaming services to limit streaming HD content on mobile devices (that typically have smaller screens), with viewers having the option to unlock HD encoding rates if they require it¹².

2.3.3 Competing players and interaction with TCP

Several studies have suggested that bandwidth-based QA algorithms can perform poorly under challenging network conditions [31], [65], [66], [67]. At a high level, these findings are due to the interesting interactions between TCP and the QA algorithms.

Congestion control in TCP aims to avoid congestion collapse, while trying to provide the client its fair-share of network throughput based on the data path and the associated state of several network equipment along the data path [68]. These algorithms are well understood and it is known that

¹⁰<https://developer.android.com/guide/topics/media/exoplayer.html>

¹¹<https://developer.apple.com/streaming/>

¹²<https://www.netflix.com/HdToggle>

TCP does not instantaneously provide a fair-share, but provides this guarantee over a period of time. In-fact, TCP congestion control typically results in a well known saw-toothed pattern based on an internal variable called the congestion window. The congestion window specifies the maximum allowed difference between the number of in-order bytes transmitted thus far and the number of in-order bytes acknowledged thus far.

Similar to the saw-toothed throughput curve of TCP, HAS clients also have a distinctive on-off download pattern, as shown in Figure 2.2. This pattern depends on the receiver buffer parameters, namely, T_{max}^{buf} and T_{min}^{buf} and the client's QA logic. Broadly speaking, it has been shown that different download patterns might arise from HAS players [69]. The simplest among these is the case where there are no off periods. The client simply downloads chunks back-to-back until it has downloaded the entire video. The other cases can contain off periods of short or long durations. Contrary to the download patterns of HAS players, regular file downloads over HTTP are modulated mainly by the sender's TCP congestion control and flow control mechanisms. With HAS clients, the control loop at the receiver's application layer is independent of the sender's TCP congestion control mechanism. This additional control loop, together with the periodic on-off patterns and the lack of interoperability information between the sender and the receiver has been found to sometimes result in oscillations, unfairness, and inaccurate estimates [66].

To overcome issues with competing clients, and to ensure that the algorithms account for bandwidth and bitrate variability across a playback session, and, in general, to ensure that estimates are reliable and robust, different means to estimate the available throughput and schedule chunk downloads from application level measurements have been suggested. This includes randomized chunk scheduling through varying target buffer sizes, better statistical methods to estimate available bandwidths [31], control theoretic approaches to chose chunk encodings [70], [71], [72], network probing at the application layer [32], and objective-based optimization to ensure consistent playback quality [73]. Paradigms other than client-driven solutions have also been explored, including server-driven traffic shaping [74] which can be used to ensure that clients avoid unnecessary oscillations. Our contributions in Paper II leverages findings from the aforementioned works to propose a novel prefetching solution that downloads chunks from a list of alternative videos during off-periods. By prefetching during off-periods, we facilitate near instantaneous startup of alternative videos and we show that in some cases the playback quality of the currently watched video can be improved simultaneously.

2.4 Large-scale content delivery

When designing systems for large user bases, it is important to make sure that large-scale content distribution is achievable with the least overhead. The system and protocols used must ensure that the protocol stack can inter-operate with other traffic while allowing the clients to consistently claim their fair-shares. The distribution systems must also make sure that the content catalog is easily updated and provides consistent service across the globe. Although the goals of most content delivery systems remain the same, different architectures have varying characteristics, performance, and workloads that they can handle [75]. Availability and costs play significant roles in the adoption and evolution of services over the Internet. Video streaming requires dedicated and globally distributed resources to store data, and network bandwidth to disseminate content on-demand. We next discuss some of the most widely used techniques.

2.4.1 Peer-to-peer networks

Peer-to-peer networks, as exemplified by some of the early file sharing systems such as Napster and Gnutella, or the more recent BitTorrent-like [76], [77], [78] systems, have relied on utilizing the distributed power available at end-hosts to disseminate content. Such systems are typically more scalable and cost effective for the content provider than centralized client-server based systems. However, compared to the classical client-server paradigm, it is difficult to provide guarantees when delivering content over regular peer-to-peer networks. As means to address this, both cloud-assisted, peer-assisted, and virtualized content delivery techniques have been proposed [79], [80], [81], [82], [83]. In such systems, a centralized servers or CDNs cooperate with end-hosts' in the content delivery. These servers typically are used to ensure that there is at least one copy of the content available (needed for availability), and clients' upload bandwidth is used to improve scalability.

2.4.2 Content delivery network

To effectively serve a large (and often geographically disperse) user base, CDNs [84] use a network of interconnected servers, which are distributed strategically and geographically to ensure quick access to data stores from several parts of the world. Several large CDNs have a global presence, and are capable of quickly moving content form origin servers to replicas, located closer to the users. The cost of delivering content using CDNs has dramatically reduced in the recent years (by a factor of 10 between 2006 and 2010 [85]) and has thereby made CDN-based content delivery very attractive. In addition to decreasing operating costs, CDN-based content delivery offers reconfigurability, centralized management, admission control, utilization of web-caches and other advantages.

Perhaps, the key benefit of CDNs, as opposed to other distribution techniques, in general, is that the CDN provider and the content provider are different entities. This allows both the providers to focus on their core business. A content provider can purchase resources from a CDN and relinquish control of handling and moving replicas to the CDN. Such agreements are analogous to the Service Level Agreements (SLA) between autonomous systems to ensure Quality of Service (QoS). Whenever a client from a certain part of the globe attempts to access a video from the content providers' website, it is redirected to a nearby CDN server that would provide good QoS, based on the current network load, client location and other parameters.

This process is most commonly done through Domain Name System (DNS) resolution, based on the entries in the DNS and various DNS-based performance metrics [86]. Other works have shown that this assignment can be done by simply looking at the address of the resolving name server that forwards the DNS query to the authoritative name server [87]. CDNs also perform load balancing in accordance to the demand, user location, network characteristics and QoS. This is generally performed at the edge-server or by a server within the cluster, which transparently fetches data through multiple back-end CDN servers [88]. The CDN also deploys caches hierarchically at every level, with the origin server (operated by the content provider) being contacted only when no copies are available within the CDN [87]. Load balancing based on Software Defined Networking (SDN) [89] have also been proposed. In addition, there are several components within the context of CDNs that are research fields in their own right, including server placement, request routing [90], replica selection [91], pricing, caching within CDNs, and many more [92]. Finally, it has also been shown that load balancing within CDN clusters can be optimized by utilizing third-party information, such as popularity of videos on social networks [93].

The content delivery infrastructures have time varying performance, depending on load, congestion in the network, outages, and other time-varying aspects. These variations potentially contribute to client-side playback impairments such as stalls and quality fluctuations. Client-driven replica selection based on the prevailing network conditions is complicated by the need to monitor multiple replicas and potentially multiple CDNs. It would therefore be much simpler if there was a central controller directs clients to the replica that provides the best performance, performs resource reservations, provides access control, and manages global resources under flash crowd events or network outages. Proof-of-concept and feasibility of such a centralized controller have been demonstrated through previous works [94], [95]. In fact, the increased adoption of SDN suggests that future large-scale control planes might run SDN-based controllers to perform resource and user QoE oriented optimizations.

Using CDNs has now become the norm to deliver video streaming services. Given the global user-base and diverse access patterns, CDNs have been pivotal in on-demand streaming over the Internet gaining wide-spread

acceptance. Some of the most popular content providers such as YouTube, Netflix, Amazon Prime and others employ CDNs for content delivery. Some of the most popular third-party CDNs include Akamai¹³, Limelight¹⁴, Level3¹⁵, Microsoft Azure¹⁶, Amazon CloudFront¹⁷ and CloudFlare¹⁸, to mention a few. However, it has also become common for Internet Service Providers (ISP), such as AT&T and Verizon, for example, to deploy their own streaming services over their CDNs. Although these developments have blurred the lines between content providers, network operators (ISP), and CDNs, the dynamics between the different parties is one of the major forces that shapes the future of large-scale content delivery [96].

2.4.3 HTTP caching

Proxies are intermediary devices which are used as an aggregation point for all traffic within a network. This facilitates the implementation of centralized firewalls, monitoring, and NATs at proxies, while at the same time providing transparent Internet access to the end hosts. Web caching through the use of proxy caches was envisioned in the early 1990s [97]. These caches operate by storing a local copy of the content that goes through them. When a locally cached content is requested again, the cache serves the request from the local cache rather from the origin server. Web caching, as discussed before, reduces bandwidth consumption, transmission delays, and the server workload. It also allows for greater network utilization, decreased network traffic, lower operational expenses [98] and enables centralized monitoring and management of the data flowing in and out of a network.

Caches are generally organized in a hierarchical manner at each level of the network [99]. Perhaps the most natural example of such an architecture includes caches at client devices, followed by caches at subnets, followed by caches at the operator level organized at regional and national levels [99]. While distributed architectures for caching [100] have been explored, hierarchical organization is most commonly used in practice.

Proxy caches store data on a finite disk. Proxy caches can use one of several replacement algorithms to make effective use of the disk. Some of the most popular cache replacement strategies are based on the Least Recently Used (LRU) policy or the Least Frequently Used (LFU) policy or hybrid policies such as Adaptive Replacement Caches (ARC) which behave as a hybrid between LRU and LFU strategies. The LRU policy evicts an object that has been requested the least recently while the LFU scheme evicts the object that has the least access frequency. Several policies perform tradeoffs between cache hit rate and byte hit rate. Some policies might optimize cache

¹³<https://www.akamai.com/>

¹⁴<https://www.limelight.com/>

¹⁵<http://www.level3.com/en/products/content-delivery-network/>

¹⁶<https://azure.microsoft.com/en-us/services/cdn/>

¹⁷<https://aws.amazon.com/cloudfront/>

¹⁸<https://www.cloudflare.com/>

hit rate by keeping smaller objects for longer in the cache (e.g., Greedy-Dual Size Frequency [101]) while others such as the Least Frequently Used with Dynamic Aging (LFU-DA) [102] store objects irrespective of their sizes to optimize the byte hit rate. There are several tradeoffs in this area and the choice of replacement strategy has to be fine-tuned based on the use case and the factors to be optimized. Intelligent cache insertion policies have also been proposed to take into account the long tail of one-time requests [87], [103].

Since streaming traffic over HTTP is also web traffic, caches can also store and replicate chunks of HTTP streams. Progressive streaming over HTTP requires the caches to store the entire video file in the local memory. This is not necessary with chunked or range-request based video, due to the video being divided into smaller addressable pieces. With chunked or range-based HTTP streams, the cache can store any number of chunks of a video initially, and gradually discard the less popular chunks and replace them with more popular chunks of other videos or HTTP responses that are most likely to be requested in the future.

Although proxy caches are beneficial to the clients, the network operator, and the content provider [104], a part of the research presented in this thesis [105] and others, [14], [65], [66], [67] reveal some drawbacks and challenges that must be addressed in the context of HAS. Here, we highlight one such challenge. First, note that the client cannot differentiate if a chunk is being delivered from the origin server or from a proxy cache. Second, the encoding rate and the time at which requests are made are determined based on the buffer occupancy and the inferred network throughput during previous chunk downloads. As discussed earlier in this section, chunks allow proxy caches to contribute positively to a HAS session with only a few cached chunks. For example, when the client's request encounters a cache-hit, the response reaches the client much faster than it regularly would if the chunk were sent by the server. This is due to the RTT to a web server being much larger than the RTT to a proxy cache. While this is encouraging, the faster chunk downloads lead to the client over estimating the available throughput and can potentially cause clients to request a chunk at too high an encoding rate. As long as the client experiences cache hits, it will likely continue to successfully download chunks in time for playback, but when the client encounters a cache miss, the skewed estimated bandwidth might not accurately reflect the available throughput to the server. Under such circumstances, the client might experience playback stalls. This simple example illustrates the importance of understanding the interaction between HAS, existing infrastructure, and other factors such as cross-traffic.

The process of prefetching, which involves predicting and downloading content before it is actually requested has been shown to be highly beneficial to traditional web-based services [106]. Along similar lines (as shown in Paper I), prefetching can be beneficial for HAS streams too. In addition to prefetching content to caches, cache replacement strategies also play a critical role in having chunks available for future requests. The quality adap-

tive nature of HAS clients require specialized cache replacement solutions as well. Previous works have looked into areas of cache replacement [107] and prefetching strategies for caches [108] in the case of quality adaptive videos, segment-based caching and prefetching [109] and, prefetching under the context of layered quality adaptation [110] as seen with SVC. This thesis proposes prefetching strategies for caches, which are designed with HAS download patterns in mind. These strategies are designed to improve the cache hit rate, while also considering bandwidth constraints.

This thesis also presents the design and evaluation of several proxy-assisted solutions that are designed to aid in improving the viewers QoE. Cooperative client-proxy solutions are also provided, in which, the client and the proxy share information between one another so as to ensure that the decisions made by the client and the proxy are consistent and result in the best possible playback experience at the client.

Recently, several popular streaming services have started streaming over HTTPS. HTTPS is the secure version of HTTP where the payload is encrypted by Transport Layer Security (TLS) [111]. Although HTTPS is intended for security critical applications, by streaming over HTTPS, the services are preventing the network operators and other network elements from observing any application level information, such as HTTP requests and other metadata [112]. In this thesis, Paper III describes a classification framework that can be used to classify and predicting streaming clients' buffer conditions even while streaming over HTTPS. Our framework also allows for potentially tracking clients' QoE and therefore identify network-related issues that might be affecting clients' QoE.

2.5 Network-driven approaches to improve content delivery

The wide-spread adoption of streaming services has not only lead to increased data volumes within networks, but more and more of this data originates from a third-party streaming content provider. This is generally referred to as Over-The-Top (OTT) media. While OTT media has unlocked an abundance of alternatives for the end-user, OTT, and even streaming HD content within the operator network for that matter, has its own set of unique challenges. Today, it has become increasingly common for organizations ranging from network operators (AT&T¹⁹) to content producers (Disney, HBO²⁰), device manufacturers (Apple, through App Store), platform creators (Google, through PlayStore), social networking websites (Facebook), etc., to deploy their own streaming services. Such services compete amongst one another and with traditional services.

¹⁹<https://www.att.com/directv/directv-now.html>

²⁰<https://www.hbonordic.com/>

The following subsections discuss some of the steps taken to improve the content delivery process and streaming in general, from the perspective of a content provider, a network operator and finally collaborative strategies between the content provider and the network operator.

2.5.1 Content provider based solutions

The content provider, arguably, has the most at stake in terms of how users perceive a streaming service. Whenever the services are being accessed over-the-top, the player code which executes at the client-side (responsible for chunk downloads, quality adaptation, appearance, etc.) is typically downloaded from the content provider, every time a client initiates a new playback session. By implementing carefully chosen QA algorithms in their players, streaming services can ensure that their clients are resilient to the variations that might be encountered in the network throughput.

Distribution platform: In addition to the client side algorithm, the content providers can also optimize their data distribution platforms to perform intelligent resource allocation [113]. OpenFlow [114] and other SDN-based techniques are being increasingly deployed in data center and CDN networks as they offer quick reconfiguration and centralized control. Prior works have applied the SDN paradigm to guarantee end-to-end QoS support by streaming over an OpenFlow network [115], perform dynamic routing and traffic engineering to improve QoE [116]. The centralized and standardized communication to-and-from a controller in SDN facilitate information sharing over standard interfaces. Other SDN-based approaches have proposed sharing application-level feedback to SDN controllers to improve client QoE [117].

Server farms and CDNs are fairly complex architectures and are their own area of research. For example, there are several techniques that can be applied in such networks to improve the content delivery process. Prior works have shown that by taking advantage of multipath communication, playback stalls can be reduced and playback quality can be improved [118], [119]. In addition to multiple paths, CDNs also deploy a hierarchy of caches that might effectively reduce the load on the origin and the assigned servers, both during regular operation and during flash-crowd events. Studies have looked at how caching, request redirection, and dynamic provisioning can be used to improve resilience during high demand [120].

Over the past decade, the Internet has undergone tremendous changes. This includes significant changes in access patterns [121], user behavior, and the content that is distributed over the Internet. Any access over the Internet relies on host-to-host communication, in which content is retrieved from hosts associated with unique addresses. However, today's Internet is dominated by content. As a natural extension, content-based architectures [122] have been proposed. Such architectures provide a paradigm shift by using content-based addressing rather than host addresses to access content. This

has the potential to simplify several cache-related issues in today's networks as it decouples location and content. Although Content Centric Networks (CCN) have not yet gained popularity, proof-of-concept implementations to perform HAS over CCNs have been attempted [123], and it remains as one potential design path for a content-dominated Internet of the future.

Content optimization: Several popular commercial streaming services aggressively replace parts of the video that have been downloaded at a lower encoding, although this practice might lead to increased demands on the servers and the content downloaded at the lower encoding are wasted. In general, the players want to avoid playback stalls and might therefore chose to download chunks at low qualities when the bandwidth conditions are poor. Since the playback buffer is in the order of many seconds to a few minutes, the network conditions might improve after the client has recovered from low-buffer conditions. In such cases, the clients can re-download some of the chunks in buffer at a higher encoding rate to playback at a consistently higher quality [124], [125]. Although SVC-based techniques are naturally better suited for such situations, as they do not waste content that has been downloaded, this remains a standard practice with HAS clients.

Personalization through recommendations: The overarching need to personalize and customize content and advertisements on a per user basis has led to the development of several recommender and personalization systems. At a high level, these systems obtain data which represents the behavior of a user or a user demographic under specific use cases. This data is then mined using machine learning techniques to generate a list of recommendations [126]. The success of the generated recommendations is continually evaluated to allocate more weight to measures which resulted in the recommendation being used by the user. In addition to generating a list of recommendations, the system has to also rank the results to improve user retention. These requirements have led to the development of content and context aware recommendation solutions [127], [128]. Other recommender systems have used matrix propagation [129], graph propagation [130], association rules [131] and machine learning based approaches such as SVM [132] have been proposed.

Over the last few years, the mobile web has grown significantly. Mobile data traffic has grown by 18-folds in the past five years and is expected to increase sevenfold in the period between 2016 and 2021 [133]. Powered by significant improvements in mobile connectivity and capability of hand-held devices, usage patterns have shifted from desktop-only to increasing use of mobile devices. This has led to further personalization opportunities. For example, the inbuilt sensors can be used to extract user's mental state and emotions and suggest actions or recommendations based on this information [134]. Several Location Based Services (LBS) have emerged that provides high-quality, relevant, timely, and location-based services, features

and information [135]. Services such as Foursquare²¹, Google Now²², tripadvisor²³ and many more have emerged lately which provide location, user interest and user history-based recommendations. There have even been LBS which use bandwidth prediction based on the current user location to perform quality adaptation accordingly [136], [137].

Video recommendations become especially important in services with large video catalogs, as users might find the task of finding interesting videos to watch very daunting otherwise²⁴. Recommendation systems are also integrated together with the built-in social network functionalities. For example, using third-party authorization [138] through websites like Google or Facebook, a signed-in user can like, up-vote, comment or share videos to other friends. This information is used in addition to user search queries, watched videos, and viewing time of videos to generate a list of recommended videos. YouTube's recommender system [139], makes use of co-visitation counts of pairs of videos (i.e., videos that were co-watched within a specific amount of time by other users), and generate a recommendation list as the most popular co-watched videos. This information can be combined with user profile information to generate a user specific recommendation, which can be ranked according to video quality, user specificity, and diversification of viewer's taste. Others have demonstrated that a strong rich-get-richer phenomenon is observed in video popularity, with a number of factors like video age and search engine biases towards popular content contributing toward this effect [140].

In this thesis, we look at client-based prefetching strategies that can be used to ensure near-instantaneous startup of alternative/recommended videos. This is motivated by users typically watching the beginnings of multiple videos before settling down to watch a video to the end [29]. We also show that by prefetching during the right time-windows, the playback quality of the video that a viewer currently is watching can be improved.

2.5.2 Network operator based solutions

Network operators provide transit for streams from a streaming service (content provider) to end-users. Although recent developments, where network operators provide a streaming service, have muddled the lines separating content providers and network operators, for the large part, this distinction still exists today. Network operators typically run a core and an access network that users can connect to, and can be classified into different tiers based on their global reachability and presence [141]. Transit providers and operator networks provide the critical last hops of connectivity to end-users and can have a significant effect on viewer satisfaction.

²¹<https://foursquare.com/>

²²<https://www.google.com/landing/now/>

²³<http://www.tripadvisor.com/>

²⁴<http://www.telegraph.co.uk/technology/2016/02/01/youtube-is-now-more-popular-than-googlecom-on-desktop-computers/>

Network-based detection and optimization of QoS and QoE:

Typically, network operators today are proportionally carrying more and more streaming traffic due to increasing popularity and user-base [1]. Among all the streaming flows observed by medium-large sized networks, it can be argued that unless there are network-related issues, most clients will have sufficient data in their buffers to make sure that they will not stall in the near future. Asides from these clients, a small number of clients might just be beginning playback, while another small proportion of clients might be experiencing low-buffer conditions and require data to be downloaded quickly. For the few clients that are in imminent risk of experiencing a playback stall, both the content provider and the operator are in positions where they can take actions that can help avoid playback stalls. Prior to performing actions, it is actually interesting to see if client low-buffer conditions can be identified by network operators, before any remediation actions can be taken. Prior works have looked at techniques to identifying clients' QoE based on network traces [142], [143], [144] and other client-based techniques [145], [146].

Deep-Packet Inspection (DPI) has long been used as a tool to perform detailed analysis of network flows. However, their scalability is questionable [147] and a recent shift towards almost all large streaming services transferring their data over HTTPS hinders both DPI and trace-driven identification of client buffer conditions. This restricts the information that can be obtained by a purely trace-driven methodology. In this thesis, we propose a novel classification framework that can predict client buffer conditions under both HTTP and HTTPS. We leverage a trusted proxy to generate a training dataset by using a set of experimental flows. Using the collected data, we train machine-learning classifiers to identify client buffer conditions in real-time even when the data is encrypted and transferred over HTTPS.

Network operations and management: The performance of operator networks is time-varying and depends on several factors, including planned capacity, actual capacity, network management and configuration, reliability of links, queue sizes, jitter, latency, used protocols, and bandwidth. Factors such as jitter, latency, and queue sizes have a significant impact on achievable throughputs in networks. Especially with TCP, end-hosts experience sharply decreasing throughputs as latency increases. Naturally, queue sizes of switches and routers have a significant effect on RTTs. In the extreme case, deployment of large queues which remain full most of the time have been found to be the root-cause of the bufferbloat problem [148]. However, for the large part, issues due to bufferbloat have been addressed using techniques based on Random Early Detection (RED) [149] and other Active Queue Management (AQM) techniques [150], [151].

2.5.3 Monitoring and guaranteeing QoE and QoS

Residential and commercial access networks typically do not provide guaranteed bitrates or latencies. However, QoS agreements guaranteeing flow characteristics are common for backbone links involving two or more ISPs or other network operators. Since IP lacks QoS specification and monitoring, other architectures to deliver QoS over IP have been developed. Initial solutions for voice and video calling include IntServ [152], DiffServ [153] and Resource reSerVation Protocol (RSVP) [154]. These have later been extended to IP applications through RSVP-TE (Traffic Engineering) by setting up Multi-Path Label Switching (MPLS) [155]. However, the area of protocol design and specification to perform traffic engineering is an entire domain of research with lots of interesting directions. For this thesis, we focus on monitoring QoS of streaming flows and for providing improvements in clients' QoE.

Monitoring client QoS, with today's streaming clients, is generally performed by client feedback. With most popular services like YouTube, for example, there is a dedicated server which collects client statistics such as buffer occupancy, played quality, and current playtime [125]. Critically, this data is transmitted out-of-band over another connection, using custom semantics that has been developed by the streaming service for their own internal use and may be used to evaluate the user experience. However, the network and transit providers that might be part of the delivery chain are generally oblivious to this data. The recent migration towards streaming over HTTPS has made it almost impossible for operators to gather this information, even if they want to. In addition to using custom designed feedback from clients, several streaming services also use third-party analytics. These third-party sites collect data via a plugin that is integrated with the streaming client.

One of the contributions in this thesis looks at ways in which network operators could potentially leverage available information, even if the communication is over HTTPS, to monitor HAS clients' buffer conditions. Again, buffer occupancy directly affects the clients stalling and therefore, can be leveraged to improve playback experiences.

2.6 The next generation of streaming services

2.6.1 Personalization on the web

Today it is common that different (personalized) versions of a website are shown to different visitors. Most websites personalize based on at least one of the following factors [156]: behavioral, contextual, historical, and collaborative personalization. The user may implicitly or explicitly be involved in the personalization process. Content personalization and recommendations have had significant effects on advertising and marketing. Prior to

wide-spread adoption of the Web, advertisers focused on marketing ideas that would resonate with the masses. Companies could typically only make a few versions of an advertisement and these had to appeal to the general public. Now, Web advertisements make it possible to create targeted ads for individual viewers that are much more personalized than what was possible with traditional methods.

Personalization and personalized recommendations have been well studied concepts in the area of content recommendation systems. While personalization has been used by some to promote the right products, others have used it as a means to reduce clutter and provide user-driven customization. In general, personalization has been used to achieve higher user retention and satisfaction. Similar to the multitude of use cases for personalization, there are several views of personalization and its goals [157]. This includes views based on popular theories such as the information overload theory, the uses and gratification theory, and user involvement theory. The information overload theory has the premise that as the recommendation system's accuracy increases, the user satisfaction increases; i.e., by removing the clutter, the user satisfaction is improved. The uses and gratifications theory suggests that the user's motivations for information access affect user satisfaction; however, when the right information is found, the user is satisfied. Finally, the user involvement theory suggests that users prefer content recommended by a process in which they have explicit involvement. Here, the explicit feedback mechanism between the user and the system drives user satisfaction.

The explosion in the number of sensors available in today's hand-held devices has led to web pages being personalized based on information from several of these sensors. Most commonly, services actively personalize web-pages based on location, device and browser information, device orientation, screen size and aspect ratio, user history, demographics, and user preferences, for example.

2.6.2 Interactive and personalized video streaming

Contrary to Web browsing, video streaming has almost exclusively been personalized only through video recommendations (as part of browsing), while allowing basic actions such as fast forward and rewind. In most cases, the viewer is expected to watch a video from the beginning to the end. One solution to enable further video interactivity and personalization of streaming videos is interactive branched video streaming. In the following, we outline some of the key concepts and ideas behind interactivity and personalization in streaming videos.

Interactive storytelling is a concept where the reader/listener/viewer is allowed to interact with the story at predetermined decision points. At these decision points, the subject is typically given choices from which to decide the outcome of a character's decision (e.g., will the character take the road

to the left or to the right) or how the story should progress (e.g., should the character talk to the stranger). By allowing the subject to interact with the story, and offering different versions of a story that resonate with different user groups, a broad range of users may be provided the desired experience.

Interactive storytelling has been used in several novels, perhaps most notably in the popular children’s novel series called “Goosebumps”. While reading the book, the reader would simply turn to a different page based on the different options given by the author. Another very popular instance of interactive storytelling was the Hollywood movie “Clue”, which in cinemas showed one of several endings that the movie had, but in the DVD release, the viewer could chose to see one of the many endings available. Along similar lines, interactive video streaming is a concept where the viewer can interact with a video and influence the content that is about to be presented. Earliest digital applications of interactive videos appeared in DVDs where the viewer, by pushing a button could navigate to a different chapter in the DVD, thereby altering the story line according to the choice made by the user. In the following, this class of interactive videos is referred to as interactive branched streaming, or simply branched streaming.

Although interactive branched streaming is fairly simple at a concept-level, there are several challenges to overcome in a real-world scenario. Typically, users expect the process of interaction and resulting changes to the video to be seamless, in that, playback should be continuous and stall free regardless of which option is selected at a branch point and how late the decision is made. Our work on interactive branched streaming over HAS is the first to formalize the branched video streaming over HAS problem and demonstrate feasibility by implementing such a player. Furthermore, our source codes have also been released for use by the research community.

We argue and show how HAS simplifies the process of branched streaming. Since the videos are split into addressable chunks or ranges with known playback durations, potential playback paths along a video can be designated which resemble graph or tree structure. Furthermore, quality adaptation can be used to meet the deadline constraints associated with branch points. User interaction, by means of clicking a button or key strokes can be used to determine branches to traverse, with defaults assigned in case the user does not interact. Our contributions in this area are detailed in Papers IV and V.

Websites such as YouTube, which offers User Generated Content (UGC), have become very successful in the recent past. The success of UGC-based sites can be attributed to them aggregating the endless resources and creativity of regular home users. Owing to the prevalent acceptance and growth of Internet-based services, several creators are interested to share their content with users all around the globe for a small incentive. These incentives are typically advertisement-driven payments to the content uploaders. Thereby, the success of a content creator depends on the number of views or clicks the content attracts. Today, websites such as YouTube have over a billion

users²⁵ and stream over a billion hours of video every day²⁶. These websites' global popularity and mass-media status have resulted in several professional creators and organizations publishing content. However, despite the rich set of content available in these systems, not much personalization is available. In this thesis, we enhance the creative power of the content creators. In particular, our design allows for both a content creator and other third-parties to define branch points and branch options using a simple text file.

In the past, nonlinear and interactive playback have been well investigated in the broadcast and multicast domains, where the client may subscribe to one or more channels and dynamically switch between channels [158, 159, 160]. Under such scenarios, it is important to balance the server bandwidth and client path-diversity; i.e., the different paths that a viewer might choose while watching the same video. In contrast, with on-demand interactive branched streaming over HAS, the client downloads content from the server using a unicast connection, and the bandwidth usage is typically not negatively affected by high path diversity (except the additional prefetching that it may require). Streaming of nonlinear datasets, such as 3D graphics and visualizations, have also been studied under the broadcast domain [161].

Branched video streaming has also been studied using progressive download, coupled with prefetching strategies and algorithms designed to parse user input and the video's structure [162]. The main drawbacks with such systems are the lack of quality adaptation, which might lead to large stall times due to bandwidth fluctuations and waiting times when performing interactive actions. Delays in executing interactive actions can affect the user satisfaction drastically. As means to reduce such events, interactive-content-aware proxy caches (proxies that can parse the location of branch choices and attempt to always cache the most popular branches) and caching strategies specific to interactive branched streaming, similar to HAS-aware proxies that we propose in Paper I, have also been proposed [163], [164].

Prior works have also looked at developing formalisms required to generate and consume personalized videos [165], authoring tools for interactive branched videos [166], [167] and dynamic video composition systems [168]. These works look at generating and conveying information about personalized videos which can be created using GUI's or search queries.

2.6.3 Multi-view video streaming

Analogous to interactive branched streaming, with multi-view video streaming, the user interacts with the video to perform actions. However, rather than telling a story through the user traversing different branches in a video, in multi-view and 360° videos, the user navigates a scene through multiple cameras or viewpoints. For example, such scenarios might include a concert where multiple cameras are recording an event. Ideally, with a multi-view

²⁵<https://www.youtube.com/yt/about/press/statistics.html>

²⁶<https://youtube.googleblog.com/2017/02/you-know-whats-cool-billion-hours.html>

video capable system, a viewer that is currently watching a view should instantaneously be able to switch to a new view without experiencing any playback interruptions. The video will continue playing from the same playback point at which the switch was initiated, but it would have switched to the stream from a different camera. Several interesting use cases can be developed for such applications, some of which include sports events, where instead of a director publishing a single view, the viewers are allowed to freely chose the camera they wish. Other examples include concerts, surveillance and monitoring.

For multi-video streaming to allow for seamless switches, multiple views need to be available at the receiver. Related works have looked at different ways of making this content available at the viewer. Video and network coding of multi-view videos have been used in both the broadcast and unicast domain [169], [170], [171], [172], [173]. With these schemes, multiple views or streams are typically synthesized into a single stream (using redundant information for compression) and some of these techniques even facilitate generation of virtual views [174], [175] (views generated based on computation of two or more views). However, the drawback of such schemes is that the viewer effectively views only one stream at a time, and the client therefore effectively has to download a significant amount of unnecessary information that is discarded. In addition to server and cloud-based systems, P2P-based schemes [176] and distributed system design over IP networks [177], have also been proposed.

360° streaming can to some extent be considered as a special case of multi-view streaming, where rather than having a set of fixed views, 360° videos allows the users to look in all directions across a multitude of yaw, pitch, and roll angles. While multi-view stream bundles can be generated with off-the-shelf cameras, including mobile phone cameras, 360° videos require specialized hardware [178], although multiple off-the-shelf cameras are become available now. 360° video adoption is increasing primarily due to commercial headsets like the Oculus Rift²⁷ and HTC Vive²⁸ which are designed for use with PCs. Headset for mobile phones, such as Google Daydream²⁹ and Samsung Gear VR³⁰, are also driving popularity. 360° videos, comes with its own set of challenges. Some of the most important challenges in this area include user head movement tracking and prediction [179], [180], masking/synthesizing intermediate views [181], and bandwidth management [182], [183].

In contrast to the related work presented in this section, Paper VI presents a HAS-based multi-view streaming solution that can seamlessly switch between multiple views based on user interaction. To provide seamless switching, the client-side algorithm not only downloads content from

²⁷<https://www.oculus.com/rift/>

²⁸<https://www.vive.com/eu/>

²⁹<https://vr.google.com/daydream/>

³⁰<http://www.samsung.com/global/galaxy/gear-vr/>

the currently watched view, but also prefetches content belonging to other streams. To balance the volume of data downloaded for each stream, our implementation performs quality-adaptive prefetching across all available views. By allocating more bandwidth to the current view and views that the viewer is most likely to view next, than to lower-probability views, we can decrease the bandwidth requirements of multi-view streaming, while ensuring that players can seamlessly transition to other views.

2.6.4 3D, VR and AR streaming

In the context of interactive streaming, several new immersive application areas have emerged recently. Although this thesis does not look specifically at these areas, we briefly discuss some of these concepts. 3D streaming involves delivery of video data complemented with depth information. Virtual Reality (VR) places the user in a simulated environment that takes up the entire field of view. For these purposes, either one or two screens (one screen per eye) are typically used, with different projection techniques [184], [185] to convey depth information. These images are interpreted by the mind as 3D or VR video. Encoding techniques for VR videos are being standardized under the MPEG-VR initiative³¹. Ignoring the video encoding and information representation challenges, the network-related challenges in 3D streaming is similar to that of multi-view and free-view streaming. In these videos, the viewer can change their orientation at any point in time and the system has to be capable of rendering the new field-of-view almost instantaneously. To adaptively stream 3D videos, resilience to view-switching has to be built into both the video codec (mostly to cover small-scale head movements) and in the downloading of content. Similar to free-viewpoint videos, where the simplest method of download would be to download the entire 360°, significant performance improvements can be attained by quality adaptively downloading different areas of the videos at different encoding rates [182]. These methods generally take advantage of head tracking, prediction, and prefetching [186], [187], [188].

Finally, Augmented Reality (AR) is another exciting application on both regular and VR-capable devices, where the view of a real-world element is enhanced or augmented by adding virtual or computer generation information to it [189]. AR can also be applied in a VR environment, where an emulated 3D video object can be enhanced by AR information. Most importantly, AR is not restricted to binocular screens and can be achieved with simple devices like our mobile phones. Important challenges with AR involves object detection and tracking in real-time [190]. For example, if the AR application involves augmenting information to what is currently recorded by the device's camera, the tracking algorithm must be capable of identifying continually an object at different degrees of rotation. The tracking information is then passed on to the AR application to generate

³¹<https://mpeg.chiariglione.org/tags/vr>

additional information on top and then needs to be rendered and displayed on the screen. AR applications can have use cases spanning several areas, including manufacturing, measurements, information retrieval, games, and television.

Chapter 3

Contributions and discussion

This chapter presents the papers included in this thesis and summarizes their major contributions. In addition to the included papers, we also list the other published papers, thesis, and paper under review by the defendant.

3.1 Publication list

Included papers:

[191] V. Krishnamoorthi, N. Carlsson, D. Eager, A. Mahanti, and N. Shahmehri, **Helping Hand or Hidden Hurdle: Proxy-assisted HTTP-based Adaptive Streaming Performance**. In *Proceedings of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (IEEE MASCOTS)*, Aug. 2013.

[105] V. Krishnamoorthi, P. Bergström, N. Carlsson, D. Eager, A. Mahanti, and N. Shahmehri, **Empowering the Creative User: Personalized HTTP-based Adaptive Streaming of Multi-path Nonlinear Video**, In *Proceedings of the ACM SIGCOMM Workshop on Future Human-Centric Multimedia Networking (FhMN)*, Aug. 2013. This paper also appeared in *ACM SIGCOMM Computer Communication Review (CCR)*, Oct. 2013.

[192] V. Krishnamoorthi, N. Carlsson, D. Eager, A. Mahanti, and N. Shahmehri, **Quality-adaptive Prefetching for Interactive Branched Video using HTTP-based Adaptive Streaming** In *Proceedings of the ACM International Conference on Multimedia (ACM Multimedia)*, Nov. 2014.

[193] V. Krishnamoorthi, N. Carlsson, D. Eager, A. Mahanti, and N. Shah-mehri, **Bandwidth-aware Prefetching for Proactive Multi-video Preload-ing and Improved HAS Performance**. In *Proceedings of the ACM International Conference on Multimedia (ACM Multimedia)*, Oct. 2015.

[125] V. Krishnamoorthi, N. Carlsson, E. Halepovic, and E. Petajan, **BUFFEST: Predicting Buffer Conditions and Real-time Require-ments of HTTP(S) Adaptive Streaming Clients**, In *Proceedings of the ACM International Conference on Multimedia Systems (ACM MMSys)*, Jun. 2017.

[194] N. Carlsson, D. Eager, V. Krishnamoorthi, and T. Polishchuk, **Opti-mized Adaptive Streaming of Multi-video Stream Bundles**, *IEEE Transactions on Multimedia (IEEE TMM)*, Jul. 2017.

Other publications:

[195] A. Bianco, V. Krishnamoorthi, L. Nanfang, and L. Giraudo, **Open-Flow Driven Ethernet Traffic Analysis**, In *Proceedings of the IEEE International Conference on Communications (IEEE ICC)*, Jun. 2014.

[196] V. Krishnamoorthi, **Efficient and Adaptive Content Delivery of Linear and Interactive Branched Videos**, Licentiate thesis, *Linköping University Electronic Press*, Nov. 2016.

[197] J. Gustafsson, R. Hiran, V. Krishnamoorthi, and N. Carlsson, **The Hidden Mailman and His Mailbag: Routing Path Analysis from a European Perspective**, In *Proceedings of the IEEE International Con-ference on Communications (IEEE ICC)*, May 2017.

[198] V. Krishnamoorthi, N. Carlsson, and E. Halepovic, **Slow but Steady: Cap-based Network Orchestration for Improved Streaming Expe-rience**, *Under submission*, 12 pages, Dec. 2017.

3.2 Summary of papers

As discussed in the previous chapters, this thesis addresses research prob-lems in two main research areas. Papers I, II, and III focus on efficient content delivery of HAS videos. These papers consider techniques that can be used by the clients or the network to (i) improve the QoE and efficiency of HAS clients in the presence of caches, (ii) improve startup times and play-back experiences when serving a catalog of videos, and (iii) new detection techniques that can be used by a network operator to identify HAS clients with low-buffer conditions in their network.

Papers IV, V, and VI present novel solutions and techniques to enable interactive streaming over HAS. In these papers, we present the formulation, requirements, implementation, and validation of two different interactive streaming techniques: (i) interactive branched videos and (ii) multi-video stream bundles.

Paper I: Helping Hand or Hidden Hurdle: Proxy-assisted HTTP-based Adaptive Streaming Performance

Streaming over HTTP comes bundled with benefits such as firewall friendliness and content caching. This made it possible to stream data to almost any host, while taking advantage of content replication closer to the edge. As explained in Section 2.2.4, quality adaptation in HAS relies on clients adaptively downloading chunks from different copies of the currently played video, with each copy being at a different encoding rate. Since chunks regardless of the encoding are independent of each other, intermediate caches need to have a local copy of the requested chunk at the requested encoding rate to offload the server. However, previous clients that played the same video might have experienced different network conditions, and therefore the cached chunks might be at a different encoding rate. This introduces client-cache interactions where cache hits and misses might interleave each other. Furthermore, the download rates measured by the clients may cycle between high (with a sequence of hits) and low (with a sequence of misses).

In this paper, we first evaluate the impact that such interactions can have on HAS clients. We then propose cache prefetching policies for improved performance. By parsing client requests, the cache tries to pre-download chunks that the client is most likely to request next. Generally, chunks are linearly numbered or have a naming convention that makes it easy to work out the index of the next chunk if the current is known. Our simple prefetching policies prefetch the next chunk in the video stream at the same quality level as that of the previous chunk. This will result in a continuous sequence of cache-hits as long as the cache can download data ahead of the client's request and when the requested encodings match. To generate cache hits even when clients experience dynamic bandwidth conditions, we develop co-operative policies where the client and the cache share information such as the buffer occupancy of the client and the indices of the chunks in the cache, for example. By combining collaborative policies with prefetching, we show that the clients can be provided stable playback experiences. Our solutions were demonstrated using Adobe's OpenSource Media Framework (OSMF), and a Squid proxy running in a dedicated testbed. Using this testbed, we also evaluate our best-effort policies and collaborative policies under different bottlenecks, such as proxy-server link being bottlenecked or client-proxy link being bottlenecked. By capturing the different performance characteristics under different bottlenecks, we characterize HAS performance with different caching and prefetching techniques.

Paper II: Bandwidth-aware Prefetching for Proactive Multi-video Preloading and Improved HAS Performance

The best video streaming experiences are provided when the playback does not stall and the video playback is of consistently good quality. In addition to the negative impact of stalls, low playback quality, and fluctuating playback quality, it is well known that large startup delays negatively affect viewer satisfaction. Furthermore, in most on-demand services, viewers are typically impatient and frequently switch between different videos based on the content [29]. To retain users within their service for long periods of time, most websites therefore use recommendation engines that suggest videos that the viewer might be interested in viewing next. However, every time viewers start playing a new video, they endure a startup delay. This paper presents and evaluates prefetching techniques to provide users instantaneous startup when switching to view such recommended videos.

However, when prefetching chunks from alternative videos to facilitate almost instantaneous startup of these videos, it is critical that we do not negatively affect the playback experiences of the currently played video. For this purpose, we propose prefetching chunks during off-periods (defined in Section 2.2.2) associated with the download of the currently played video. By downloading chunks of alternate videos during these off-periods, we take advantage of the period during which the communication channel would otherwise be unused. Within this context, we then design and evaluate different classes of prefetching policies that suit different use cases. For example, we consider a best-effort policy which prefetches a certain number of alternate chunks (per-video) from a list recommended videos whenever off-periods are in effect. The best effort policy starts prefetching from the next video once the specified number of alternative chunks of a video has been downloaded. To provide more control over when different videos are prefetched, we propose a token-based policy, which allows prefetching from a new video only when a token is available. By controlling times at which tokens are released, a client can control the time at which different alternative videos are prefetched. Finally, we also consider a deadline-based policy, which has fixed targets for the number of chunks of the current video and of the alternative video that have to be downloaded by specified deadlines. To compute a chunk download schedule that satisfies these constraints, our implementation solves an optimization formulation which ensures the deadlines are met while also ensuring that the playback quality of the current video is maximized. We have implemented and experimentally validated the performance of our policies using the OSMF player and trace-driven experiments. Our evaluations show that in addition to improving startup times of alternative videos, continuing to download chunks during the off-period is beneficial to the video that is currently being watched. This is achieved by the prefetch requests using the same connection as the one used by the current video. By keeping the connection active during off-periods, we avoid entering slow-

start every time the client resumes downloading new chunks. We have also validated that our findings and conclusions are consistent across multiple TCP implementations.

Paper III: BUFFEST: Predicting Buffer Conditions and Real-time Requirements of HTTP(S) Adaptive Streaming Clients

Proactive network management and resource allocation is becoming increasingly important as IP-based entertainment gains popularity and wide-spread use. However, network operators typically do not have direct access to per-session QoE related information from OTT video streaming services. Recently, a large proportion of streaming services have started streaming over HTTPS, which makes it difficult for network operators to access application-level information, even if they wanted to. In this paper, we tackle the problem of identifying per-client-level buffer information from HTTPS traces. With this information, an operator can evaluate how streaming clients perform within their network and can perform short-term and long-term resource planning accordingly.

Our framework consists of a trusted proxy, an event-based buffer emulator, and machine learning classifiers. To train machine learning classifiers, a set of experimental clients are configured to trust the proxy. By doing so, we obtain a trace of application-level messages in clear-text. While these messages can show the transactions between the client and the server, they do not contain explicit information about client buffer levels. Our buffer emulation module leverages additional information such as the manifest file, video metadata, etc., to generate a detailed buffer emulation that we use to automatically label simultaneously collected HTTPS traces with client buffer conditions.

Although the use of the trusted proxy does not allow real-time classification of all clients, we use this to create training data for more scalable classifiers. The operator would direct a group of test clients via the proxy or run targeted experiments to obtain detailed buffer level information (using the emulator) and collect encrypted packet-level traces in parallel. After gathering a suitably large tagged dataset, we build machine learning models using boosted decision trees and SVM-based classifiers. Once the models are trained, they can then be used on real-world traffic or production HTTPS traces. These classifiers are designed to use statistics that can be computed quickly from observing packet-level information. We evaluate the performance of our classifiers and find that they can identify clients experiencing low-buffer conditions with good accuracy.

Paper IV: Empowering the Creative User: Personalized HTTP-based Adaptive Streaming of Multi-path Nonlinear Video

Personalized streaming, where the content of a video is adjusted to suit viewers' individual taste, offers the possibility to cater to a wide spectrum of user tastes. While personalization for a large spectrum of users is feasible, server-side and network scalability would be poor if a new copy of a video is required for every user. In this paper, we propose novel interactive streaming techniques that utilize chunking in HAS to allow effective personalization to a diverse user-base with as little as a single video.

To allow effective implementation of such a service, we present a text-based representation that allows interactive branched video sequences to be defined and stitched together into a tree-like structure. This specifies when a branch point becomes available and what options are available to the user at each branch point. For example, if certain users want to avoid violent scenes in a video, a content creator could insert branch points just ahead of such scenes linking the user instead to parts that are not as violent. By giving users the choice of which path to take at a branch point, users can also adapt their viewing based on the part that most resonates with their current state-of-mind. In addition to proposing branched video streaming over HAS, we also present proof-of-concept prefetching and buffer management strategies that make sure that the client can actively prefetch from the different available branches and makes sure that the viewer does not experience playback stalls when transitioning to a new branch option.

Paper V: Quality-adaptive Prefetching for Interactive Branched Video using HTTP-based Adaptive Streaming

Papers IV and V consider the problem of interactive branched video streaming. Compared to Paper IV, which presents a proof-of-concept implementation and demonstrates that seamless interactive streaming over HAS is possible, this paper formalizes in detail the downloading and prefetching problem in interactive branched streaming over HAS. We consider the use of multiple parallel TCP connections to download both the chunks of the video that are in the currently viewed path and the chunks that need to be prefetched to ensure that clients can be provided seamless transitions into each of the next set of possible branches. We build an optimization formulation that determines both which chunks to download over the multiple connections and the quality at which each chunk should be downloaded so to provide the best possible user experience.

Through our detailed implementation and evaluation using the OSMF player¹, we show that different tradeoffs can be obtained by maximizing

¹Available for download here: <http://www.ida.liu.se/~nikca89/papers/mm14.html>

different parameters. This includes comparison of greedy policies that maximize the overall bitrate of the downloaded chunks, and more advanced policies that compute an optimized download schedule based on different constraints. We compare the different policies against a naive client and a client that uses a single connection to download all the chunks. Through our evaluation, we show that our implementation can take advantage of the multiple TCP connections and schedule chunk qualities such that stalls are avoided during playback, the playback quality remains stable, and that transitions to new branches are seamless.

Paper VI: Optimized Adaptive Streaming of Multi-video Stream Bundles

Finally, Paper VI considers the problem of streaming multi-video stream bundles over HAS. The design challenges with multi-video stream bundles are different when compared to interactive branched streaming. With interactive branched streaming, there are isolated branch points that might only occur a few times during playback. With stream bundles, the user is interested in watching a scene from one of many perspectives. For example, scenarios suited for multi-video stream bundles include events like a concert or a basketball game, where there are multiple cameras covering the same point of focus, and the viewer can, at any point in time chose to change their view to a different one.

In this paper, we formalize this problem of downloading multi-video stream bundles over HAS, and both analytically and numerically characterize the optimal solution. Following this, we present an implementation and evaluation using an OSMF client². Our formalization carefully considers the different streams, the likelihood that the viewer might switch to a different stream, and schedules chunk download requests that reflect the user behavior. By assigning a larger portion of bandwidth to the currently watched stream and the streams that are most likely to be switched to by the viewer, we can ensure good playback quality even when clients perform a stream switch. Furthermore, we have also shown that we can keep the stream switching latency to a minimum by performing careful buffer management and by allocating a much smaller bandwidth to prefetch less likely streams in the bundle.

²Available for download here: <http://www.ida.liu.se/~nikca89/papers/tmm17.html>

3.3 Discussion

In this section we discuss our contributions in the context of other research and some developments that are closely related to our work.

Proxy-assisted delivery of HAS videos: Caching and prefetching are well investigated techniques in the context of the web. The adoption of video streaming over HTTP was partly driven by the simplicity of leveraging existing content replication methods with these techniques. However, during the period in which we performed this work, many questions remained unanswered. While previous works have investigated cache replacement policies, policies that download missing pieces in a video stream in the context of SVC [108], and the interaction between proxy caches and HAS clients' rate detection algorithms [14], we were one of the first to study content-aware and collaborative policies in the context of HAS. Paper I investigates a range of proxy-based solutions, including best-effort proxies, prefetching proxies, and collaborative proxies, and evaluate them under different bottlenecks.

While our work shows the potential benefits of collaborative caching and prefetching, such techniques are not fully exploited today in caches deployed by the network operator. This is largely due to content providers setting the `no-cache` or `no-store` cache-control headers, owing to security and privacy risks, and increasingly using HTTPS. HTTPS is encrypted at a per-connection level, therefore most ISP-owned web caches are forced to simply relay the HTTPS traffic without caching the video data. However, the techniques discussed in our paper are still highly applicable to HTTPS traffic when caches are deployed by the content provider, including through operator-provider collaboration or CDN solutions that leverage caching. Some commercial streaming services are even placing caches and replica servers within the operator networks [56]. Our contributions can be used to dynamically populate such replica servers when the requested content is unavailable locally.

Bandwidth-aware prefetching for preloading HAS videos: Providing low startup times for viewers remains as one of the top-most priorities for streaming services. Studies have reported that viewers start abandoning playback sessions when startup delays are longer than 2 seconds. With every additional second beyond that resulting in a 5.8% increase in the abandonment rate [199]. Most HAS clients of today start playback at a predefined encoding rate (e.g., the lowest) that provides small startup times. Recent protocols such as QUIC [40] have been designed to enable transmission of the payload from the first packet in a connection. While such techniques help reduce the handshake and connection setup times, several RTTs are still needed to download the large volume of data required to commence playback.

In this paper, we consider viewers switching to another video within a list of recommended videos. Our bandwidth-aware prefetching techniques proposed in Paper II, takes an orthogonal approach in that it leverages in-

formation regarding recommended alternative videos, and is able to provide instantaneous startup of these videos. We explore different policy classes which range from best-effort to guaranteed prefetching of alternative videos.

In addition to providing instantaneous playback when switching to alternative videos, our solution simultaneously addresses the unfairness problem that has been observed between competing HAS clients. While previous works have looked at competing HAS clients [65], [66] and techniques to solve the resulting unfairness [74], [70], [33] through shaping and better rate-estimations, the core of the problem is due to HAS clients not downloading any content during the off-periods. By performing prefetching at opportune moments, our prefetching solution decreases unfairness between competing clients. While this approach might not be applicable for clients streaming over metered links (as we download additional data during off-periods), the additional benefit of playing alternative videos instantaneously can be very attractive to clients in unmetered links.

Predicting buffer conditions of HAS clients: Compared to streaming services of the past, streaming videos from OTT providers has changed operator networks into best-effort delivery networks. Ideally, network operators would like to implement numerous optimizations to their networks to improve utilization, user satisfaction, etc. Without collaboration between clients and networks, it is difficult to track or estimate the impact of QoE related metrics such as stalls, and build preventative measures based on clients' buffer conditions, for example. This problem is further complicated by increasingly many video streaming services using HTTPS. To address these problems, in Paper III, we designed a framework that can perform detailed per-client buffer emulation, automated labeling and training of machine learning classifiers, and machine learning classifiers that can be used to identify clients' buffer conditions even when streaming over HTTPS. Our contribution focused on identifying clients experiencing low-buffer conditions. Our contributions open the possibility of network-assisted clients and services, which in the long-run can be used to improve network utilization and viewer satisfaction, by assisting the clients to perform the right QA choices.

Related works have looked at network-based traffic management solutions for HAS clients that allocate resources so as to reduce extremes of QoE across clients [200]. Ideally, when clients that are likely to stall are identified, techniques to quickly increase the clients' buffer before stalls are encountered can help improve viewer QoE. We are currently investigating this problem and a candidate set of actions that an operator can perform when clients with low-buffer conditions are identified.

Interactive branched streaming and multi-video stream bundles: Papers IV–VI consider two forms of interactive streaming: branched video and multi-video stream bundles. Within the area of interactive streaming, there are several other techniques, including (but not limited to) free-view point streaming (that allows the viewer to navigate to different syn-

thetically generated viewing positions in the video), tiled video, and 360° streaming. While these different techniques have their own advantages and challenges, some of the fundamental challenge of efficiently transferring the different views over a bottlenecked link also exists for these techniques. In our contributions in this area, we focus on prefetching and buffer management challenges that need to be solved in order to facilitate seamless transitions to new branches or streams. Arguably, experiencing seamless interactive video is vital to viewer satisfaction.

While interactive streaming techniques have been previously explored in the context of broadcast and multicast streaming, and even in the context of non-adaptive streaming over HAS, our works are the first to formalize and demonstrate feasibility of quality adaptive interactive streaming over HAS. Through our contributions in Paper IV, we provide the means for a content creator to define an interactive video, for a third-party enthusiast to create his/her own interactive version of a video, rules to resolve conflicts by the use of longest-path matching when multiple branch options conflict with one another, prefetching and quality adaptation strategies using multiple parallel TCP connections to ensure seamless transitions and that clients obtain their fair bandwidth shares when competing with others. In general, our solutions account for different user behavior and ensure that the transitions are seamless through quality adaptive prefetching and buffer management.

Closely related to our interactive branched streaming works are works which have proposed authoring tools [166], and download and caching strategies for interactive videos [162]. In contrast to most related work, our solutions are designed with the current state-of-the-art (HAS) in mind. In our framework, the definition of branch points and branch options are done through a text-based metafile. While we manually create these files for our experiments, we note that more advanced graphical authoring websites have emerged lately. Using these to create our text-based metafiles could provide interesting future work. Interactive branched streaming is also gaining popularity in terms of use in the advertisement and marketing industries. Furthermore, Netflix has even released an interactive branched video based story of the character Puss in Boots.

Multi-video stream bundles allow the viewer to watch the same video from different angles. In this context too, we formulate, analyze and evaluate solutions to facilitate seamless switching to streams within the bundle. While problems similar to that of multi-video stream bundles exist in the area of 360°, VR, and AR streaming, our solution based on user behavior prediction can be applied to these use cases too. In the context of both multi-video stream bundles and interactive branched videos, our solutions are completely client-driven and do not require any modifications to the server side.

Chapter 4

Conclusions and future work

4.1 Summary and conclusion

Over the recent years, on-demand streaming services over the Internet have become the main source of household entertainment. This thesis contributes towards improving the state-of-the-art of today's content delivery techniques. Main contributions include the design and evaluation of novel content-aware and collaborative proxies, prefetching techniques to decrease startup times, network-based detection techniques to estimate client buffer conditions, and two novel categories of interactive streaming over HAS called interactive branched streaming and multi-video stream bundles. We believe that improvements to the infrastructure and personalized content delivery over HAS are important steps towards realizing several forecasts that have been made regarding tomorrow's video services and the delivery thereof.

Efficient content delivery of linear videos: This thesis investigates several techniques that can be used to improve the efficiency and scalability of on-demand video streaming services. First, we evaluate the effects of proxy caches on HAS streams, and design and evaluate several content-aware prefetching policies on a proof-of-concept implementation. Our policies are shown to make the process of delivering content more efficient, and at the same time, improve the viewer's QoE. Our findings reveal that proxy caches do not always offer a helping hand to HAS streams, but that they often require careful configuration and should be made content aware, so that their benefits are maximized and that potential drawbacks with their interaction with clients are minimized. Through evaluation of our content-aware policies, using real-world representative experiments, we show several areas of improvements, including how to improve and leverage opportunistic prefetching and content-caching opportunities to significantly reduce operating expenses, increase hit rates, and provide clients with a higher average

playback quality.

Second, we propose classes of prefetching policies and buffer management techniques that help in preloading the beginnings of alternative videos. Through careful buffer management and by prefetching content ahead of time, our solutions offer instantaneous startup times for alternative videos. We consider different policy classes and demonstrate the performance of our solutions through implementation using a HAS client. Our policies leverage the otherwise unused off durations, to perform quality-adaptive prefetching of alternative videos. In addition to providing instantaneous playback of alternative videos, our solution also address the unfairness observed in HAS clients in the presence of competing traffic.

Third, we propose techniques that can be used by the network operator to detect important client-related events, such as low-buffer conditions. Such events are important precursors to playback stalls and it is well known that stalls have a significant impact on the viewers' QoE. We have proposed a classification framework called BUFFEST, which includes machine learning techniques to detect clients with low-buffer conditions. We show that detecting clients with low-buffer conditions is possible even if the client streams are encrypted using HTTPS. To generate a dataset on which the machine learning classifiers can be trained, we design an emulator module that uses a trusted proxy to capture and emulate playback sessions in detail. Using datasets collected from two real-world streaming services, we demonstrate that identifying clients with low-buffer conditions is possible even when the traffic is encrypted using HTTPS. The proposed solution is particularly important since most popular streaming services have migrated to using HTTPS today.

Efficient content delivery of interactive videos: Modern applications and services offered over the Internet are continually adapting their behavior based on user preferences and other user specific customizations. Using the current state-of-the-art for streaming (HAS), we have proposed, designed, implemented, and evaluated interactive branched video streaming and multi-video stream bundles over HTTP.

Interactive branched videos allow the viewer to interactively choose one among several alternative story lines at predefined times while watching a video. Our contributions here include an optimization framework which can be used to determine when to prefetch chunks, which parts of the video to prefetch from several potential branches that the client might chose, and the quality level to be selected for each such chunk download. Our framework and policies are designed to handle a wide range of scenarios, including fluctuating bandwidths, frequent branch switches, and other variations that one might encounter in the real world. Our client-side implementation tries to minimize playback stalls, maximize the playback qualities, and make sure that the action of switching to a different branch does not lead to interruptions that might affect the playback experience of the viewer.

With the concept of multi-video stream bundles, multiple parallel video

streams that are synchronized in time are used, each providing the video from a different camera capturing the same event or movie. Our stream bundle framework leverages quality adaptive downloading and users' probability of switching to each stream to carefully balance the prefetching of content along both the current view and alternative views. We propose an optimization framework to adaptively decide a download schedule based on the current bandwidth and stream switching probabilities. We also demonstrate through analytic, numeric, and experimental evaluation, that our optimization framework and buffer management solution are able to achieve effective prefetching and close to seamless playback switching between the streams in the bundle when there is sufficient bandwidth to prefetch the alternative streams.

4.2 Future work

There are several interesting avenues along which the work in this thesis can be extended. We believe that the increased adoption of HAS will continue to drive innovative architectural solutions that aim to simultaneously provide better user QoE and reduce the operational and capital expenses. Several proxy-cache-based optimizations can be performed to improve a subset of metrics that the operator is most interested in. Prefetching and caching policies can be tuned to focus on metrics such as byte hit rate, overall hit rate, hit rate for low-buffer clients, or hit rate for the first chunks, for example. In addition to the content-aware prefetching policies proposed in this thesis, proxies can also be used to pre-push the parts of the videos that the clients are most likely to view. For example, the HTTP 2.0 standard includes features for server push, where the server can send content without receiving requests for them. We envision such features to be used by proxy caches too. By pre-pushing parts of the video to the client when the network is not busy, the network can reduce some of its future resource usage.

Future work could also consider how to intelligently pre-push content to clients based on user location in mobile networks. If certain areas suffer from poor coverage or poor transmission characteristics caused by factors out of control of the operator, pre-pushing content can help avoid playback stalls when clients are traversing such stretches. Similar solutions can also be used to reduce startup times for the videos that a user is likely to play next. However, when pre-pushing content, the server or the proxy cache need to be aware of the bandwidth budget that is available to ensure that the pushed content does not starve out the resources that are being pulled by the client. Prior works have looked at location-based bandwidth prediction based on known bandwidth measurements [136], [201], however, from the perspective of a client that plans for future bandwidth. Offloading the prediction tasks to a server can simplify the client-side implementation.

This thesis proposed techniques to identify client buffer conditions by observing live network traces. Equipped with this information, the network

operator could decide to help clients that are in imminent danger of experiencing playback stalls. The operator could chose to help selected clients' flows within their network, by prioritizing their packets, tweak the bandwidth metering or bandwidth shaping to enable a temporary increase to the download rates, etc. Furthermore, the clients, network operator, and the content provider can collaborate together to improve the network utilization, QoS, and QoE. Investigation of how such policies might be implemented in real networks is currently ongoing.

In the area of interactive services, several user-centric measurements based on attached web cameras, mics, and other sensors can be used to deduce the viewer's interest or engagement while watching a particular video. This information can then be used to further personalize the content, determine branch-points that will be offered to that particular viewer, or even to predict the choices that the viewer is most likely to make.

Future work could also consider the application of machine-learning techniques in the context of interactive streaming. Our branched streaming solution over HAS requires either the creator or another third-party to create a metafile that indicates the different branch points and branch options in the video. However, with developments in automated captioning and speech recognition techniques, the process of generating transcripts for pre-recorded videos is simplified. Using topic models, parts of videos can be tagged to automatically generate different video segments, which can then be linked via branch points and branch options. While we have done preliminary investigation in the area of automated branched video generation using transcripts, several interesting use cases remain to be explored.

The recent proliferation of VR and AR capable devices and headsets provide opportunities to develop innovative streaming experiences. Several of these devices come with a hand-held controller that is designed to be an interface for gaming applications. However, using the same controller and headsets, interactive branched streaming and stream bundling can also be taken into the VR/AR arena. A multitude of applications and use cases are possible, which present several interesting research questions to explore.

Finally, our contributions in this thesis can also be applied to streaming of 360° videos. We are currently investigating how to best serve 360° videos over HAS. The current practice under such scenarios is that the whole 360° view is streamed to a client. This often requires at least three times more bandwidth compared to only streaming the currently viewed portion of the video. By carefully segmenting the video into different regions of interest, we can use HAS' chunking and quality adaptation to download the currently viewed area at a high quality. However, in order to protect against playback stalls when the viewers change their views, prefetching strategies are required to adapt the downloaded qualities of the other parts of the video. By carefully controlling the allocated bandwidth to different regions of the video, clients can still enjoy stall-free interaction with such videos.

Bibliography

- [1] 2016 Global Internet Phenomena: Latin America and North America. Technical report, Sandvine Incorporated ULC., 2016.
- [2] Cisco Visual Networking Index: Forecast and Methodology, 2013-2018. Technical report, Cisco systems INC., 2014.
- [3] ISO/IEC 23009-1:2012: Dynamic Adaptive Streaming over HTTP (DASH). Technical report, ISO/IEC, 2012.
- [4] Cisco Visual Networking Index: The Zettabyte Era- Trends and Analysis. Technical report, Cisco systems INC., 2015.
- [5] B. Briscoe, A. Brunstrom, A. Petlund, D. Hayes, D. Ros, I. J. Tsang, S. Gjessing, G. Fairhurst, C. Griwodz, and M. Welzl. Reducing Internet latency: A survey of techniques and their merits. *IEEE Communications Surveys Tutorials*, 18(3):2149–2196, 2016.
- [6] J. Apostolopoulos, W. Tan, and S. Wee. Video streaming: Concepts, algorithms, and systems. *HP Laboratories, report HPL-2002-260*, 2002.
- [7] J. Rosenberg and H. Schulzrinne. An RTP payload format for generic forward error correction. RFC 2733.
- [8] H. Parmar and M. Thornburgh. Real-Time Messaging Protocol (RTMP) specification. *Adobe specifications, December*, 2012.
- [9] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. 2003. RFC 3550.
- [10] H. Schulzrinne, A. Rao, and R. Lanphier. Real Time Streaming Protocol (RTSP). 1998. RFC 2326.
- [11] J. Nielsen. Nielsen’s law of Internet bandwidth. <http://www.nngroup.com/articles/law-of-bandwidth/>, 1998. Accessed: 02-Jan-2018.
- [12] Cisco Visual Networking Index: The Zettabyte Era- Trends and Analysis. Technical report, Cisco systems INC., 2017.

- [13] C. Krasic, K. Li, and J. Walpole. The case for streaming multimedia with TCP. In *Proc. IDMS workshop*, 2001.
- [14] S. Benno, J. O. Esteban, and I. Rimac. Adaptive streaming: The network HAS to help. *Bell Lab. Tech. Journal*, 16(2):101–114, Sep. 2011.
- [15] V. Venkataraman, K. Yoshida, and P. Francis. Chunkyspread: Heterogeneous unstructured tree-based peer-to-peer multicast. In *Proc. IEEE ICNP*, Nov. 2006.
- [16] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: High-bandwidth multicast in cooperative environments. *SIGOPS Oper. Syst. Rev.*, 37(5):298–313, 2003.
- [17] A. Mahanti, D. Eager, M. Vernon, and D. Sundaram-Stukel. Scalable on-demand media streaming with packet loss recovery. *ACM SIGCOMM Computer Communication Review*, 31(4):97–108, 2001.
- [18] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas. Protocol Independent Multicast - Sparse Mode (PIM-SM). 2006. RFC 4601.
- [19] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Trans. on circuits and systems for video technology*, 17(9):1103–1120, 2007.
- [20] Y. Wang, Z. Ma, and Y. Ou. Modeling rate and perceptual quality of scalable video as functions of quantization and frame rate and its application in scalable video adaptation. In *Proc. Packet Video Workshop*, May 2009.
- [21] J. Famaey, S. Latre, N. Bouten, W. Van de Meerssche, B. De Vleeschauwer, W. Van Leekwijck, and F. De Turck. On the merits of SVC-based HTTP adaptive streaming. In *Proc. IFIP/IEEE IM*, May 2013.
- [22] J. Liu, B. Li, and Y. Zhang. Adaptive video multicast over the Internet. *IEEE Multimedia*, 10(1):22–33, 2003.
- [23] L.S. Lam, J.Y.B. Lee, S.C. Liew, and W. Wang. A transparent rate adaptation algorithm for streaming video over the Internet. In *Proc. AINA*, Mar. 2004.
- [24] L. Chenghao, I. Bouazizi, and M. Gabbouj. Advanced rate adaption for unicast streaming of scalable video. In *Proc. IEEE ICC*, May 2010.
- [25] S. Cheung, M. Ammar, and X. Li. On the use of destination set grouping to improve fairness in multicast video distribution. In *Proc. IEEE INFOCOM*, Mar. 1996.

- [26] R. Rejaie, M. Handley, and D. Estrin. Quality adaptation for congestion controlled video playback over the Internet. In *Proc. ACM SIGCOMM*, Aug. 1999.
- [27] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven layered multicast. In *Proc. ACM SIGCOMM*, Aug. 1996.
- [28] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang. Understanding the impact of video quality on user engagement. In *Proc. ACM SIGCOMM*, Aug. 2011.
- [29] L. Chen, Y.P. Zhou, and D.M. Chiu. A study of user behavior in online VoD services. *Computer Communications*, 46:66–75, 2014.
- [30] C. Liu, I. Bouazizi, and M. Gabbouj. Rate adaptation for adaptive HTTP streaming. In *Proc. ACM MMSys*, Feb. 2011.
- [31] J. Jiang, V. Sekar, and H. Zhang. Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE. In *Proc. ACM CoNEXT*, Dec. 2012.
- [32] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A.C. Begen, and D. Oran. Probe and adapt: Rate adaptation for HTTP video streaming at scale. *IEEE Journal on Selected Areas in Communications*, 32(4):719–733, Apr. 2014.
- [33] T.Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson. A buffer-based approach to rate adaptation: Evidence from a large video streaming service. In *Proc. ACM SIGCOMM*, Aug. 2014.
- [34] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman. BOLA: Near-optimal bitrate adaptation for online videos. In *Proc. IEEE INFOCOM*, Apr. 2016.
- [35] C. Müller and C. Timmerer. A VLC media player plugin enabling dynamic adaptive streaming over HTTP. In *Proc. ACM Multimedia*, Nov. 2011.
- [36] 3GPP TS 26.247 version 11.1.0. Technical report, European Telecommunications Standards Institute, 2012.
- [37] M. Belshe, R. Peon, and M. Thomson (Ed.). HyperText Transfer Protocol version 2 (HTTP/2). 2015. RFC 7540.
- [38] M. Belshe and R. Peon. SPDY protocol. Technical report, 2012. IETF draft-mbelshe-httpbis-spdy-00.
- [39] R. Huysegems, J. van der Hooft, T. Bostoen, P. Alface, S. Petrangeli, T. Wauters, and F. De Turck. HTTP/2-based methods to improve the live experience of adaptive streaming. In *Proc. ACM Multimedia*, Oct. 2015.

- [40] J. Iyengar and I. Swett. QUIC: A UDP-based secure and reliable transport for HTTP/2. Technical report, 2015. IETF draft-tsvwg-quic-protocol-00.
- [41] G. Carlucci, L. De Cicco, and S. Mascolo. HTTP over UDP: An experimental investigation of QUIC. In *Proc. ACM SAC*, Apr. 2015.
- [42] X. Liu, F. Dobrian, H. Milner, J. Jiang, V. Sekar, I. Stoica, and H. Zhang. A case for a coordinated Internet video control plane. In *Proc. ACM SIGCOMM*, Aug. 2012.
- [43] C. Müller, S. Lederer, and C. Timmerer. An evaluation of dynamic adaptive streaming over HTTP in vehicular environments. In *Proc. MoVid workshop*, Feb. 2012.
- [44] T. Lohmar, T. Einarsson, P. Frojdh, F. Gabin, and M. Kampmann. Dynamic adaptive HTTP streaming of live content. In *Proc. IEEE WoWMoM*, Jun. 2011.
- [45] T. Kupka, P. Halvorsen, and C. Griwodz. Performance of on-off traffic stemming from live adaptive segmented HTTP video streaming. In *Proc. IEEE LCN*, Oct. 2012.
- [46] IEEE standard for Ethernet amendment 9: Physical layer specifications and management parameters for 1000 Mb/s operation over plastic optical fiber. *IEEE Std 802.3bv-2017*, pages 1–140, Mar. 2017.
- [47] IEEE standard for information technology– telecommunications and information exchange between systemslocal and metropolitan area networks– specific requirements–part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications–amendment 4: Enhancements for very high throughput for operation in bands below 6 ghz. *IEEE Std 802.11ac-2013*, pages 1–425, 2013.
- [48] Qualcomm research: LTE in unlicensed spectrum. Technical report, Qualcomm technologies INC., 2014.
- [49] MulteFire release 1.0 technical paper: A new way to wireless. Technical report, MulteFire Alliance, Jan. 2017.
- [50] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: A simple model and its empirical validation. *ACM SIGCOMM Computer Communication Review*, 28(4):303–314, 1998.
- [51] S. Deng, R. Netravali, A. Sivaraman, and H. Balakrishnan. WiFi, LTE, or both?: Measuring multi-homed wireless Internet performance. In *Proc. ACM IMC*, Nov. 2014.
- [52] 5G radio access. Technical report, Ericsson AB, 2016.

- [53] V. Adhikari, Y. Guo, F. Hao, V. Hilt, Z. Zhang, M. Varvello, and M. Steiner. Measurement study of Netflix, Hulu, and a tale of three CDNs. *IEEE/ACM Trans. Netw.*, 23(6):1984–1997, Dec. 2015.
- [54] B. Frank, I. Poesse, Y. Lin, G. Smaragdakis, A. Feldmann, B. Maggs, J. Rake, S. Uhlig, and R. Weber. Pushing CDN-ISP collaboration to the limit. *SIGCOMM Comput. Commun. Rev.*, 43(3):34–44, Jul. 2013.
- [55] T. Böttger, F. Cuadrado, G. Tyson, I. Castro, and S. Uhlig. Open connect everywhere: A glimpse at the Internet ecosystem through the lens of the netflix CDN. *arXiv preprint arXiv:1606.05519*, 2016.
- [56] Netflix open connect overview. Technical report, Netflix INC., 2017.
- [57] T. Hoßfeld, P. E. Heegaard, L. Skorin-Kapov, and M. Varela. No silver bullet: QoE metrics, QoE fairness, and user diversity in the context of QoE management. In *Proc. QoMEX*, May 2017.
- [58] M. Varela, L. Skorin-Kapov, and T. Ebrahimi. Quality of service versus quality of experience. In *Quality of experience*, pages 85–96. Springer, 2014.
- [59] R. Mok, E. Chan, and R. Chang. Measuring the quality of experience of HTTP video streaming. In *IFIP/IEEE IM Workshop*, May 2011.
- [60] T. De Pessemier, K. De Moor, W. Joseph, L. De Marez, and L. Martens. Quantifying the influence of rebuffering interruptions on the user’s quality of experience during mobile video watching. *IEEE Trans. on Broadcasting*, 59(1):47–61, Mar. 2013.
- [61] T. Hoßfeld, M. Seufert, C. Sieber, T. Zinner, and P. Tran-Gia. Identifying QoE optimal adaptation of HTTP adaptive streaming based on subjective studies. *Computer Networks*, 81:320–332, 2015.
- [62] H. Yu, D. Zheng, B. Zhao, and W. Zheng. Understanding user behavior in large-scale video-on-demand systems. In *Proc. ACM SIGOPS/EuroSys*, Apr. 2006.
- [63] R. Mok, E. Chan, X. Luo, and R. Chang. Inferring the QoE of HTTP video streaming from user-viewing activities. In *Proc. ACM SIGCOMM W-MUST*, Aug. 2011.
- [64] O. Oyman and S. Singh. Quality of experience for HTTP adaptive streaming services. *IEEE Communications Magazine*, 50(4):20–27, 2012.
- [65] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis. What happens when HTTP adaptive streaming players compete for bandwidth? In *Proc. ACM NOSSDAV*, Feb. 2012.

- [66] T. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari. Confused, timid, and unstable: Picking a video streaming rate is hard. In *Proc. ACM IMC*, Nov. 2012.
- [67] S. Akhshabi, A. C. Begen, and C. Dovrolis. An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP. In *Proc. ACM MMSys*, Feb. 2011.
- [68] M. Allman, V. Paxson, and E. Blanton. TCP congestion control. 2009. RFC 2581.
- [69] A. Rao, A. Legout, Y. Lim, D. Towsley, C. Barakat, and W. Dabbous. Network characteristics of video streaming traffic. In *Proc. ACM CoNEXT*, Dec. 2011.
- [70] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo. ELASTIC: A client-side controller for dynamic adaptive streaming over HTTP (DASH). In *Proc. IEEE Packet Video Workshop*, Dec. 2013.
- [71] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli. A control-theoretic approach for dynamic adaptive video streaming over HTTP. In *Proc. ACM SIGCOMM*, Aug. 2015.
- [72] L. De Cicco and S. Mascolo. An adaptive video streaming control system: Modeling, validation, and performance evaluation. *IEEE/ACM Trans. on Networking*, 22(2):526–539, 2014.
- [73] Z. Li, A. Begen, J. Gahm, Y. Shan, B. Osler, and D. Oran. Streaming video over HTTP with consistent quality. In *Proc. ACM MMSys*, Mar. 2014.
- [74] S. Akhshabi, L. Anantakrishnan, C. Dovrolis, and A. C. Begen. Server-based traffic shaping for stabilizing oscillating adaptive streaming players. In *Proc. ACM NOSSDAV*, Mar. 2013.
- [75] S. Saroiu, K. Gummadi, R. Dunn, S. Gribble, and H. Levy. An analysis of Internet content delivery systems. *SIGOPS Operating Systems Review*, 36(SI):315–327, 2002.
- [76] B. Cohen. Incentives build robustness in BitTorrent. In *Proc. Workshop on Economics of Peer-to-Peer systems*, Jun. 2003.
- [77] A. Legout, G. Urvoy-Keller, and P. Michiardi. Understanding BitTorrent: An experimental perspective. Technical report, 2005.
- [78] C. Zhang, P. Dhungel, D. Wu, and K.W. Ross. Unraveling the BitTorrent ecosystem. *IEEE Trans. on Parallel and Distributed Systems*, 22(7):1164–1177, 2011.

- [79] C. Huang, J. Li, and K. Ross. Peer-assisted VoD: Making Internet video distribution cheap. In *Proc. IPTPS*, Feb. 2007.
- [80] G. Dán and N. Carlsson. Dynamic content allocation for cloud-assisted service of periodic workloads. In *Proc. IEEE INFOCOM*, May 2014.
- [81] N. Carlsson, G. Dan, D. Eager, and A. Mahanti. Tradeoffs in cloud and peer-assisted content delivery systems. In *Proc. IEEE P2P*, Sep. 2012.
- [82] H. Yin, X. Liu, T. Zhan, V. Sekar, F. Qiu, C. Lin, H. Zhang, and B. Li. Design and deployment of a hybrid CDN-P2P system for live video streaming: Experiences with LiveSky. In *Proc. ACM Multimedia*, Oct. 2009.
- [83] J. Mukherjee, M. Wang, and D. Krishnamurthy. Performance testing web applications on the cloud. In *Proc. ICST Workshop*, Mar. 2014.
- [84] G. Peng. CDN: content distribution network. *Computing Research Repository (CoRR)*, 2004.
- [85] L. Popa, A. Ghodsi, and I. Stoica. HTTP as the narrow waist of the future Internet. In *Proc. ACM SIGCOMM HotNets Workshop*, pages 6:1–6:6, 2010.
- [86] A. Vakali and G. Pallis. Content delivery networks: Status and trends. *IEEE Internet Computing*, 7(6):68–74, 2003.
- [87] B. Maggs and R. Sitaraman. Algorithmic nuggets in content delivery. *SIGCOMM Comput. Commun. Rev.*, 45(3):52–66, Jul. 2015.
- [88] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl. Globally distributed content delivery. *IEEE Internet Computing*, 6(5):50–58, 2002.
- [89] R. Wang, D. Butnariu, and J. Rexford. OpenFlow-based server load balancing gone wild. In *Proc. USENIX Hot-ICE*, Mar. 2011.
- [90] N. Carlsson, D. Eager, A. Gopinathan, and Z. Li. Caching and optimized request routing in cloud-based content delivery systems. *Performance Evaluation*, 79:38–55, 2014.
- [91] N. Carlsson and D. Eager. Server selection in large-scale video-on-demand systems. *ACM Trans. on Multimedia Computing, Communications, and Applications*, 6(1):1, 2010.
- [92] G. Pallis and A. Vakali. Insight and perspectives for content delivery networks. *Communications of the ACM*, 49(1):101–106, Jan. 2006.

- [93] T. Hoßfeld, L. Skorin-Kapov, Y. Haddad, P. Pocta, V. Siris, A. Zgank, and H. Melvin. Can context monitoring improve QoE? A case study of video flash crowds in the Internet of services. In *Proc. IFIP/IEEE IM*, May 2015.
- [94] G. Cofano, L. De Cicco, and S. Mascolo. A control architecture for massive adaptive video streaming delivery. In *Proc. VideoNext Workshop*, Dec. 2014.
- [95] A. Ganjam, F. Siddiqui, J. Zhan, X. Liu, I. Stoica, J. Jiang, V. Sekar, and H. Zhang. C3: Internet-scale control plane for video quality optimization. In *Proc. USENIX NSDI*, May 2015.
- [96] Cisco Visual Networking Index: Forecast and Methodology, 2015-2020. Technical report, Cisco systems INC., 2015.
- [97] A. Luotonen and K. Altis. World-wide web proxies. *Computer Networks and ISDN Systems*, 27(2):147–154, 1994.
- [98] A. Araldo, D. Rossi, and F. Martignon. Cost-aware caching: Caching more (costly items) for less (ISPs operational expenditures). *IEEE Trans. on Parallel and Distributed Systems*, 27(5):1316–1330, 2016.
- [99] A. Chankhunthod, P. B. Danzig, C. Neerdaels, M. E Schwartz, and K. J. Worrel. A hierarchical Internet object cache. In *Proc. USENIX ATC*, Jan. 1996.
- [100] D. Povey and J. Harrison. A distributed Internet cache. In *Proc. Australian Computer Science Conference*, 1997.
- [101] L. Cherkasova. Improving WWW proxies performance with greedy-dual-size-frequency caching policy. Technical report, Hewlett-Packard Laboratories, 1998.
- [102] M. Arlitt, L. Cherkasova, J. Dilley, R. Friedrich, and T. Jin. Evaluating content management techniques for web proxy caches. *ACM SIGMETRICS Perf. Eval. Review*, 27(4):3–11, 2000.
- [103] N. Carlsson and D. Eager. Ephemeral content popularity at the edge and implications for on-demand caching. *IEEE Trans. on Parallel and Distributed Systems*, 28(6):1621–1634, Jun. 2017.
- [104] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. YouTube traffic characterization: A view from the edge. In *Proc. ACM IMC*, Aug. 2007.
- [105] V. Krishnamoorthi, P. Bergström, N. Carlsson, D. Eager, A. Mahanti, and N. Shahmehri. Empowering the creative user: Personalized HTTP-based adaptive streaming of multi-path nonlinear video. In *Proc. ACM SIGCOMM FhMN Workshop*, Aug. 2013.

- [106] V. Padmanabhan and J. Mogul. Using predictive prefetching to improve World Wide Web latency. *ACM Computer Communication Review*, 26(3):22–36, Jul. 1996.
- [107] A. Gouta, D. Hong, A. Kermarrec, and Y. Lelouedec. HTTP adaptive streaming in mobile networks: Characteristics and caching opportunities. In *Proc. IEEE MASCOTS*, Aug. 2013.
- [108] M. Zink, J. Schmitt, and R. Steinmetz. Layer-encoded video in scalable adaptive streaming. *IEEE Trans. on Multimedia*, 7(1):75–84, 2005.
- [109] S. Chen, H. Wang, X. Zhang, B. Shen, and S. Wee. Segment-based proxy caching for Internet streaming media delivery. *IEEE Multimedia*, 12(3):59–67, 2005.
- [110] R. Rejaie and J. Kangasharju. Mocha: A quality adaptive multimedia proxy cache for Internet streaming. In *Proc. ACM NOSSDAV*, Jun. 2001.
- [111] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) protocol version 1.2. 2008. RFC 5246.
- [112] Z. Cao, G. Xiong, Y. Zhao, and Z. Liand L. Guo. A survey on encrypted traffic classification. In *Proc. USENIX ATC*, Jun. 2014.
- [113] V. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z. Zhang. Unreeling Netflix: Understanding and improving multi-CDN movie delivery. In *Proc. IEEE INFOCOM*, pages 1620–1628, Mar. 2012.
- [114] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.
- [115] H. Egilmez, S. Dane, K. Bagci, and A. Tekalp. OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end quality of service over software-defined networks. In *Proc. Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, Dec. 2012.
- [116] H. Nam, K. Kim, J. Kim, and H. Schulzrinne. Towards QoE-aware video streaming using SDN. In *Proc. IEEE GLOBECOM*, Dec. 2014.
- [117] M. Jarschel, F. Wamser, T. Hohn, T. Zinner, and P. Tran-Gia. SDN-based application-aware networking on the example of YouTube video streaming. In *Proc. European Workshop on SDN*, Oct. 2013.

- [118] B. Wang, W. Wei, Z. Guo, and D. Towsley. Multipath live streaming via TCP: Scheme, performance and benefits. *ACM Trans. Multimedia Comput. Commun. Appl.*, 5(3):25:1–25:23, Aug. 2009.
- [119] S. Gouache, G. Bichot, A. Bsila, and C. Howson. Distributed and adaptive HTTP streaming. In *Proc. IEEE ICME*, Jul. 2011.
- [120] P. Wendell and M. Freedman. Going viral: Flash crowds in an open CDN. In *Proc. ACM IMC*, Nov. 2011.
- [121] A. Ali-Eldin, M. Kihl, J. Tordsson, and E. Elmroth. Analysis and characterization of a video-on-demand service workload. In *Proc. ACM MMSys*, Mar. 2015.
- [122] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard. Networking named content. In *Proc. ACM CoNEXT*, Dec. 2009.
- [123] Y. Liu, J. Geurts, J. C. Point, S. Lederer, B. Rainer, C. Müller, C. Timmerer, and H. Hellwagner. Dynamic adaptive streaming over CCN: A caching and overhead analysis. In *Proc. IEEE ICC*, Jun. 2013.
- [124] C. Sieber, A. Blenk, M. Hinteregger, and W. Kellerer. The cost of aggressive HTTP adaptive streaming: Quantifying YouTube’s redundant traffic. In *Proc. IFIP/IEEE IM*, May 2015.
- [125] V. Krishnamoorthi, N. Carlsson, E. Halepovic, and E. Petajan. BUFFEST: Predicting buffer conditions and real-time requirements of HTTP (S) adaptive streaming clients. In *Proc. ACM MMSys*, Jun. 2017.
- [126] R. Zhou, S. Khemmarat, and L. Gao. The impact of YouTube recommendation system on video views. In *Proc. ACM IMC*, Nov. 2010.
- [127] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The Adaptive Web: Methods and Strategies of Web Personalization (LNCS)*, volume 4321, pages 325–341. Springer Berlin Heidelberg, 2007.
- [128] X. Amatriain. Mining large streams of user data for personalized recommendations. *ACM SIGKDD Explorations*, 14:37–48, Dec. 2012.
- [129] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.
- [130] Q. Huang, B. Chen, J. Wang, and T. Mei. Personalized video recommendation through graph propagation. *ACM Trans. on Multimedia Computing, Communications, and Applications*, 10(4):32:1–32:17, Jun. 2014.

- [131] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Effective personalization based on association rule discovery from web usage data. In *Proc. WIDM Workshop*, Nov. 2001.
- [132] K. Oku, S. Nakajima, J. Miyazaki, and S. Uemura. Context-aware SVM for context-dependent information recommendation. *Proc. IEEE Mobile Data Management*, May 2006.
- [133] Cisco Visual Networking Index: Global mobile data traffic forecast update, 2016–2021. Technical report, Cisco systems INC., 2017.
- [134] H. Kim and Y. Choi. Emosens: Affective entity scoring, a novel service recommendation framework for mobile platform. In *Proc. ACM RecSys*, Dec. 2011.
- [135] B. Rao and L. Minakakis. Evolution of mobile location-based services. *Communications of the ACM*, 46(12), Dec. 2003.
- [136] H. Riiser, T. Endestad, P. Vigmostad, C. Griwodz, and P. Halvorsen. Video streaming using a location-based bandwidth-lookup service for bitrate planning. *ACM Trans. Multimedia Comput. Commun. Appl.*, 8(3):24:1–24:19, Aug. 2012.
- [137] K. Evensen, A. Petlund, H. Riiser, P. Vigmostad, D. Kaspar, C. Griwodz, and P. Halvorsen. Mobile video streaming using location-based network prediction and transparent handover. In *Proc. NOSSDAV*, Jun. 2011.
- [138] A. Vapen, N. Carlsson, A. Mahanti, and N. Shahmehri. A look at the third-party identity management landscape. *IEEE Internet Computing*, 20(2):18–25, 2016.
- [139] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath. The YouTube video recommendation system. In *Proc. ACM RecSys*, Sep. 2010.
- [140] Y. Borghol, S. Ardon, N. Carlsson, D. Eager, and A. Mahanti. The untold story of the clones: content-agnostic factors that impact YouTube video popularity. In *Proc. ACM SIGKDD*, Aug. 2012.
- [141] L. Subramanian, S. Agarwal, J. Rexford, and R. Katz. Characterizing the Internet hierarchy from multiple vantage points. Technical report, University of California at Berkeley, 2001.
- [142] I. Orsolic, D. Pevec, M. Suznjivic, and L. Skorin-Kapov. YouTube QoE estimation based on the analysis of encrypted network traffic using machine learning. In *Proc. IEEE Globecom Workshops*, Dec. 2016.

- [143] G. Dimopoulos, I. Leontiadis, P. Barlet-Ros, and K. Papagiannaki. Measuring video QoE from encrypted traffic. In *Proc. IMC*, Nov. 2016.
- [144] B. Staehle, M. Hirth, R. Pries, F. Wamser, and D. Staehle. YoMo: A YouTube application comfort monitoring tool. *New Dimensions in the Assessment and Support of Quality of Experience for Multimedia Applications*, pages 1–3, 2010.
- [145] R. Schatz, T. Hoßfeld, and P. Casas. Passive YouTube QoE monitoring for ISPs. In *Proc. IEEE IMIS*, Jul. 2012.
- [146] P. Casas, M. Seufert, and R. Schatz. YOUQMON: A system for on-line monitoring of YouTube QoE in operational 3G networks. *SIGMETRICS Perform. Eval. Rev.*, 41:44–46, 2013.
- [147] F. Yu, Z. Chen, Y. Diao, T. Lakshman, and R. Katz. Fast and memory-efficient regular expression matching for deep packet inspection. In *Proc. ACM/IEEE ANCS*, Dec. 2006.
- [148] J. Gettys and K. Nichols. Bufferbloat: Dark buffers in the Internet. *Queue*, 9(11):40:40–40:54, Nov. 2011.
- [149] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. on Networking*, 1(4):397–413, 1993.
- [150] K. Nichols and V. Jacobson. Controlling queue delay. *Queue*, 10(5):20:20–20:34, May 2012.
- [151] R. Pan, P. Natarajan, C. Piglione, M. S. Prabhu, V. Subramanian, F. Baker, and B. VerSteeg. Pie: A lightweight control scheme to address the bufferbloat problem. In *Proc. HPSR*, Jun. 2013.
- [152] R. Branden, D. Clark, and S. Shenker. Integrated services in the Internet architecture: an overview. 1994. RFC 1633.
- [153] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. 1998. RFC 2475.
- [154] R. Branden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource reSerVation Protocol (RSVP). 1997. RFC 2205.
- [155] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and S. Swallow. RSVP-TE: Extensions to RSVP for LSP tunnels. 2001. RFC 3209.
- [156] N. Thurman and S. Schifferes. The future of personalization at news websites: lessons from a longitudinal study. *Journalism Studies*, 13(5-6):775–790, 2012.

- [157] L. Ting-Peng, L. Hung-Jen, and K. Yi-Cheng. Personalized content recommendation and user satisfaction: Theoretical synthesis and empirical findings. *Journal of Management Information Systems*, 23(3):45–70, 2006.
- [158] Y. Zhao, D. L. Eager, and M. K. Vernon. Scalable on-demand streaming of nonlinear media. *IEEE/ACM Trans. on Networking*, 15(5):1149–1162, Oct. 2007.
- [159] N. Carlsson, A. Mahanti, Z. Li, and D. L. Eager. Optimized periodic broadcast of nonlinear media. *IEEE Trans. on Multimedia*, 10(5):871–884, 2008.
- [160] A. Hu. Video-on-demand broadcasting protocols: A comprehensive study. In *Proc. IEEE INFOCOM*, Aug. 2001.
- [161] D. Gotz. Scalable and adaptive streaming for non-linear media. In *Proc. ACM Multimedia*, Oct. 2006.
- [162] B. Meixner and J. Hoffmann. Intelligent download and cache management for interactive non-linear video. *Multimedia Tools and Applications*, 70(2):905–948, 2014.
- [163] C. Griwodz, F. Johnsen, S. Rekkedal, and P. Halvorsen. Caching of interactive multiple choice MPEG-4 presentations. In *Proc. IEEE IPCCC*, Apr. 2006.
- [164] M. Reisslein, F. Hartanto, and K. Ross. Interactive video streaming with proxy servers. *Information Sciences*, 140(1-2):3–31, 2002.
- [165] A. Sobe, L. Böszörményi, and M. Taschwer. Video Notation (ViNo): a formalism for describing and evaluating non-sequential multimedia access. *International Journal on Advances in Software*, 3(1/2):19–30, 2010.
- [166] B. Meixner, K. Matusik, C. Grill, and H. Kosch. Towards an easy to use authoring tool for interactive non-linear video. *Multimedia Tools and Applications*, 70(2):1251–1276, 2014.
- [167] U. Spierling, S. Weiß, and W. Müller. Towards accessible authoring tools for interactive storytelling. In *Proc. TIDSE*, pages 169–180, 2006.
- [168] D. Johansen, P. Halvorsen, H. Johansen, H. Rüser, C. Gurrin, B. Olstad, C. Griwodz, Å. Kvalnes, J. Hurley, and T. Kupka. Search-based composition, streaming and playback of video archive content. *Multimedia Tools and Applications*, 61(2):419–445, 2012.
- [169] P. Merkle, A. Smolic, K. Müller, and T. Wiegand. Multi-view video plus depth representation and coding. In *Proc. IEEE ICIP*, Apr. 2007.

- [170] P. Merkle, A. Smolic, K. Müller, and T. Wiegand. Efficient prediction structures for multiview video coding. *IEEE Trans. Circuits and Systems for Video Technology*, 17(11):1461–1473, Nov. 2007.
- [171] N. Cheung, A. Ortega, and G. Cheung. Distributed source coding techniques for interactive multiview video streaming. In *Proc. PCS*, May 2009.
- [172] G. Cheung, A. Ortega, and T. Sakamoto. Coding structure optimization for interactive multiview streaming in virtual world observation. In *Proc. IEEE MMSP*, Oct. 2008.
- [173] T. Su, A. Sobhani, A. Yassine, S. Shirmohammadi, and A. Javadtalab. A DASH-based HEVC multi-view video streaming system. *Journal of Real-Time Image Processing*, 12(2):329–342, Aug. 2016.
- [174] K. Müller, A. Smolic, K. Dix, P. Merkle, and T. Wiegand. Coding and intermediate view synthesis of multiview video plus depth. In *Proc. IEEE ICIP*, Nov. 2009.
- [175] A. Smolic, K. Müller, K. Dix, P. Merkle, P. Kauff, and T. Wiegand. Intermediate view interpolation based on multiview video plus depth for advanced 3D video systems. In *Proc. IEEE ICIP*, Oct. 2008.
- [176] E. Kurutepe and T. Sikora. Feasibility of multi-view video streaming over P2P networks. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, 2008.
- [177] J. Lou, H. Cai, and J. Li. A real-time interactive multi-view video system. In *Proc. ACM Multimedia*, Nov. 2005.
- [178] S. Nayar. Omnidirectional video camera. In *Proc. DARPA Image Understanding Workshop*, 1997.
- [179] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan. Optimizing 360 video delivery over cellular networks. In *Proc. All Things Cellular: Operations, Applications and Challenges*, Oct. 2016.
- [180] Y. Bao, H. Wu, T. Zhang, A. Ramli, and X. Liu. Shooting a moving target: Motion-prediction-based transmission for 360-degree videos. In *Proc. IEEE Big Data*, Dec. 2016.
- [181] A. Hamza and M. Hefeeda. Adaptive streaming of interactive free viewpoint videos to heterogeneous clients. In *Proc. ACM MMSys*, Jun. 2016.
- [182] M. Hosseini and V. Swaminathan. Adaptive 360 VR video streaming: Divide and conquer. In *Proc. IEEE ISM*, Dec. 2016.

- [183] D. Ochi, Y. Kunita, K. Fujii, A. Kojima, S. Iwaki, and J. Hirose. HMD viewing spherical video streaming system. In *Proc. ACM Multimedia*, Nov. 2014.
- [184] R. Toth, J. Nilsson, and T. Akenine-Möller. Comparison of projection methods for rendering virtual reality. In *Proc. HPG*, Jun. 2016.
- [185] K. Calagari, K. Templin, T. Elgamal, K. Diab, P. Didyk, W. Matusik, and M. Hefeeda. Anahita: A system for 3D video streaming with depth customization. In *Proc. ACM Multimedia*, 2014.
- [186] E. Kurutepe, M. Civanlar, and A. Tekalp. Client-driven selective streaming of multiview video for interactive 3DTV. *IEEE Trans. on Circuits and Systems for Video Technology*, 17(11):1558–1565, Nov. 2007.
- [187] A. Smolic and P. Kauff. Interactive 3-D video representation and coding technologies. *Proc. of the IEEE*, 93(1):98–110, Jan. 2005.
- [188] A. M. Tekalp, E. Kurutepe, and M. R. Civanlar. 3DTV over IP: End-to-end streaming of multiview video. *IEEE Signal Process. Mag.*, 24(6), 2007.
- [189] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, and M. Ivkovic. Augmented reality technologies, systems and applications. *Multimedia Tools and Applications*, 51(1):341–377, 2011.
- [190] R. Azuma. Tracking requirements for augmented reality. *Commun. ACM*, 36(7):50–51, Jul. 1993.
- [191] V. Krishnamoorthi, N. Carlsson, D. Eager, A. Mahanti, and N. Shahmehri. Helping hand or hidden hurdle: Proxy-assisted HTTP-based adaptive streaming performance. In *Proc. IEEE MASCOTS*, Aug. 2013.
- [192] V. Krishnamoorthi, N. Carlsson, D. Eager, A. Mahanti, and N. Shahmehri. Quality-adaptive prefetching for interactive branched video using HTTP-based adaptive streaming. In *Proc. ACM Multimedia*, Nov. 2014.
- [193] V. Krishnamoorthi, N. Carlsson, D. Eager, A. Mahanti, and N. Shahmehri. Bandwidth-aware prefetching for proactive multi-video preloading and improved has performance. In *Proc. ACM Multimedia*, Oct. 2015.
- [194] N. Carlsson, D. Eager, V. Krishnamoorthi, and T. Polishchuk. Optimized adaptive streaming of multi-video stream bundles. *IEEE Trans. Multimedia*, 19(7):1637–1653, Jul. 2017.

- [195] A. Bianco, V. Krishnamoorthi, N. Li, and L. Giraudo. OpenFlow driven Ethernet traffic analysis. In *Proc. IEEE ICC*, Jun. 2014.
- [196] V. Krishnamoorthi. Efficient and adaptive content delivery of linear and interactive branched videos. Licentiate thesis, Nov. 2016.
- [197] J. Gustafsson, R. Hiran, V. Krishnamoorthi, and N. Carlsson. The hidden mailman and his mailbag: Routing path analysis from a European perspective. In *Proc. IEEE ICC*, May 2017.
- [198] V. Krishnamoorthi, N. Carlsson, and E. Halepovic. Slow but steady: Cap-based network orchestration for improved streaming experience. Under submission.
- [199] S. Krishnan and R. Sitaraman. Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs. In *Proc. IMC*, Nov. 2012.
- [200] A. Zahran, J. Quinlan, K. Ramakrishnan, and C. Sreenan. SAP: Stall-Aware Pacing for improved DASH video experience in cellular networks. In *Proc. ACM MMSys*, Jun. 2017.
- [201] T. Linder, P. Persson, A. Forsberg, J. Danielsson, and N. Carlsson. On using crowd-sourced network measurements for performance prediction. In *Proc. IEEE/IFIP WONS*, Jan. 2016.

Papers

The papers associated with this thesis have been removed for copyright reasons. For more details about these see:

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-143802>

Dissertations

Linköping Studies in Science and Technology

Linköping Studies in Arts and Science

Linköping Studies in Statistics

Linköping Studies in Information Science

Linköping Studies in Science and Technology

- No 14 **Anders Haraldsson:** A Program Manipulation System Based on Partial Evaluation, 1977, ISBN 91-7372-144-1.
- No 17 **Bengt Magnhagen:** Probability Based Verification of Time Margins in Digital Designs, 1977, ISBN 91-7372-157-3.
- No 18 **Mats Cedwall:** Semantisk analys av processbeskrivningar i naturligt språk, 1977, ISBN 91- 7372-168-9.
- No 22 **Jaak Urmi:** A Machine Independent LISP Compiler and its Implications for Ideal Hardware, 1978, ISBN 91-7372-188-3.
- No 33 **Tore Risch:** Compilation of Multiple File Queries in a Meta-Database System, 1978, ISBN 91- 7372-232-4.
- No 51 **Erland Jungert:** Synthesizing Database Structures from a User Oriented Data Model, 1980, ISBN 91-7372-387-8.
- No 54 **Sture Hägglund:** Contributions to the Development of Methods and Tools for Interactive Design of Applications Software, 1980, ISBN 91-7372-404-1.
- No 55 **Pär Emanuelson:** Performance Enhancement in a Well-Structured Pattern Matcher through Partial Evaluation, 1980, ISBN 91-7372-403-3.
- No 58 **Bengt Johnsson, Bertil Andersson:** The Human-Computer Interface in Commercial Systems, 1981, ISBN 91-7372-414-9.
- No 69 **H. Jan Komorowski:** A Specification of an Abstract Prolog Machine and its Application to Partial Evaluation, 1981, ISBN 91-7372-479-3.
- No 71 **René Reboh:** Knowledge Engineering Techniques and Tools for Expert Systems, 1981, ISBN 91-7372-489-0.
- No 77 **Östen Oskarsson:** Mechanisms of Modifiability in large Software Systems, 1982, ISBN 91- 7372-527-7.
- No 94 **Hans Lunell:** Code Generator Writing Systems, 1983, ISBN 91-7372-652-4.
- No 97 **Andrzej Lingas:** Advances in Minimum Weight Triangulation, 1983, ISBN 91-7372-660-5.
- No 109 **Peter Fritzon:** Towards a Distributed Programming Environment based on Incremental Compilation, 1984, ISBN 91-7372-801-2.
- No 111 **Erik Tengvald:** The Design of Expert Planning Systems. An Experimental Operations Planning System for Turning, 1984, ISBN 91-7372- 805-5.
- No 155 **Christos Levcopoulos:** Heuristics for Minimum Decompositions of Polygons, 1987, ISBN 91-7870-133-3.
- No 165 **James W. Goodwin:** A Theory and System for Non-Monotonic Reasoning, 1987, ISBN 91-7870-183-X.
- No 170 **Zebo Peng:** A Formal Methodology for Automated Synthesis of VLSI Systems, 1987, ISBN 91-7870-225-9.
- No 174 **Johan Fagerström:** A Paradigm and System for Design of Distributed Systems, 1988, ISBN 91-7870-301-8.
- No 192 **Dimiter Driankov:** Towards a Many Valued Logic of Quantified Belief, 1988, ISBN 91-7870-374-3.
- No 213 **Lin Padgham:** Non-Monotonic Inheritance for an Object Oriented Knowledge Base, 1989, ISBN 91-7870-485-5.
- No 214 **Tony Larsson:** A Formal Hardware Description and Verification Method, 1989, ISBN 91-7870-517-7.
- No 221 **Michael Reinfrank:** Fundamentals and Logical Foundations of Truth Maintenance, 1989, ISBN 91-7870-546-0.
- No 239 **Jonas Löwgren:** Knowledge-Based Design Support and Discourse Management in User Interface Management Systems, 1991, ISBN 91-7870-720-X.
- No 244 **Henrik Eriksson:** Meta-Tool Support for Knowledge Acquisition, 1991, ISBN 91-7870-746-3.
- No 252 **Peter Eklund:** An Epistemic Approach to Interactive Design in Multiple Inheritance Hierarchies, 1991, ISBN 91-7870-784-6.
- No 258 **Patrick Doherty:** NML3 - A Non-Monotonic Formalism with Explicit Defaults, 1991, ISBN 91-7870-816-8.
- No 260 **Nahid Shahmehri:** Generalized Algorithmic Debugging, 1991, ISBN 91-7870-828-1.
- No 264 **Nils Dahlbäck:** Representation of Discourse-Cognitive and Computational Aspects, 1992, ISBN 91-7870-850-8.
- No 265 **Ulf Nilsson:** Abstract Interpretations and Abstract Machines: Contributions to a Methodology for the Implementation of Logic Programs, 1992, ISBN 91-7870-858-3.
- No 270 **Ralph Rönquist:** Theory and Practice of Tense-bound Object References, 1992, ISBN 91-7870-873-7.
- No 273 **Björn Fjellborg:** Pipeline Extraction for VLSI Data Path Synthesis, 1992, ISBN 91-7870-880-X.
- No 276 **Staffan Bonnier:** A Formal Basis for Horn Clause Logic with External Polymorphic Functions, 1992, ISBN 91-7870-896-6.
- No 277 **Kristian Sandahl:** Developing Knowledge Management Systems with an Active Expert Methodology, 1992, ISBN 91-7870-897-4.
- No 281 **Christer Bäckström:** Computational Complexity of Reasoning about Plans, 1992, ISBN 91-7870-979-2.
- No 292 **Mats Wirén:** Studies in Incremental Natural Language Analysis, 1992, ISBN 91-7871-027-8.
- No 297 **Mariam Kamkar:** Interprocedural Dynamic Slicing with Applications to Debugging and Testing, 1993, ISBN 91-7871-065-0.
- No 302 **Tingting Zhang:** A Study in Diagnosis Using Classification and Defaults, 1993, ISBN 91-7871-078-2.
- No 312 **Arne Jönsson:** Dialogue Management for Natural Language Interfaces - An Empirical Approach, 1993, ISBN 91-7871-110-X.
- No 338 **Simin Nadjem-Tehrani:** Reactive Systems in Physical Environments: Compositional Modelling and Framework for Verification, 1994, ISBN 91-7871-237-8.
- No 371 **Bengt Savén:** Business Models for Decision Support and Learning. A Study of Discrete-Event Manufacturing Simulation at Asea/ABB 1968-1993, 1995, ISBN 91-7871-494-X.
- No 375 **Ulf Söderman:** Conceptual Modelling of Mode Switching Physical Systems, 1995, ISBN 91-7871-516-4.
- No 383 **Andreas Kägedal:** Exploiting Groundness in Logic Programs, 1995, ISBN 91-7871-538-5.
- No 396 **George Fodor:** Ontological Control, Description, Identification and Recovery from Problematic Control Situations, 1995, ISBN 91-7871-603-9.
- No 413 **Mikael Pettersson:** Compiling Natural Semantics, 1995, ISBN 91-7871-641-1.
- No 414 **Xinli Gu:** RT Level Testability Improvement by Testability Analysis and Transformations, 1996, ISBN 91-7871-654-3.

- No 416 **Hua Shu**: Distributed Default Reasoning, 1996, ISBN 91-7871-665-9.
- No 429 **Jaime Villegas**: Simulation Supported Industrial Training from an Organisational Learning Perspective - Development and Evaluation of the SSIT Method, 1996, ISBN 91-7871-700-0.
- No 431 **Peter Jonsson**: Studies in Action Planning: Algorithms and Complexity, 1996, ISBN 91-7871-704-3.
- No 437 **Johan Boye**: Directional Types in Logic Programming, 1996, ISBN 91-7871-725-6.
- No 439 **Cecilia Sjöberg**: Activities, Voices and Arenas: Participatory Design in Practice, 1996, ISBN 91-7871-728-0.
- No 448 **Patrick Lambrix**: Part-Whole Reasoning in Description Logics, 1996, ISBN 91-7871-820-1.
- No 452 **Kjell Orsborn**: On Extensible and Object-Relational Database Technology for Finite Element Analysis Applications, 1996, ISBN 91-7871-827-9.
- No 459 **Olof Johansson**: Development Environments for Complex Product Models, 1996, ISBN 91-7871-855-4.
- No 461 **Lena Strömbäck**: User-Defined Constructions in Unification-Based Formalisms, 1997, ISBN 91-7871-857-0.
- No 462 **Lars Degerstedt**: Tabulation-based Logic Programming: A Multi-Level View of Query Answering, 1996, ISBN 91-7871-858-9.
- No 475 **Fredrik Nilsson**: Strategi och ekonomisk styrning - En studie av hur ekonomiska styrsystem utformas och används efter företagsförvärv, 1997, ISBN 91-7871-914-3.
- No 480 **Mikael Lindvall**: An Empirical Study of Requirements-Driven Impact Analysis in Object-Oriented Software Evolution, 1997, ISBN 91-7871-927-5.
- No 485 **Göran Forslund**: Opinion-Based Systems: The Cooperative Perspective on Knowledge-Based Decision Support, 1997, ISBN 91-7871-938-0.
- No 494 **Martin Sköld**: Active Database Management Systems for Monitoring and Control, 1997, ISBN 91-7219-002-7.
- No 495 **Hans Olsén**: Automatic Verification of Petri Nets in a CLP framework, 1997, ISBN 91-7219-011-6.
- No 498 **Thomas Drakengren**: Algorithms and Complexity for Temporal and Spatial Formalisms, 1997, ISBN 91-7219-019-1.
- No 502 **Jakob Axelsson**: Analysis and Synthesis of Heterogeneous Real-Time Systems, 1997, ISBN 91-7219-035-3.
- No 503 **Johan Ringström**: Compiler Generation for Data-Parallel Programming Languages from Two-Level Semantics Specifications, 1997, ISBN 91-7219-045-0.
- No 512 **Anna Moberg**: Närhet och distans - Studier av kommunikationsmönster i satellitkontor och flexibla kontor, 1997, ISBN 91-7219-119-8.
- No 520 **Mikael Ronström**: Design and Modelling of a Parallel Data Server for Telecom Applications, 1998, ISBN 91-7219-169-4.
- No 522 **Niclas Ohlsson**: Towards Effective Fault Prevention - An Empirical Study in Software Engineering, 1998, ISBN 91-7219-176-7.
- No 526 **Joachim Karlsson**: A Systematic Approach for Prioritizing Software Requirements, 1998, ISBN 91-7219-184-8.
- No 530 **Henrik Nilsson**: Declarative Debugging for Lazy Functional Languages, 1998, ISBN 91-7219-197-X.
- No 555 **Jonas Hallberg**: Timing Issues in High-Level Synthesis, 1998, ISBN 91-7219-369-7.
- No 561 **Ling Lin**: Management of I-D Sequence Data - From Discrete to Continuous, 1999, ISBN 91-7219-402-2.
- No 563 **Eva L. Ragnemalm**: Student Modelling based on Collaborative Dialogue with a Learning Companion, 1999, ISBN 91-7219-412-X.
- No 567 **Jörgen Lindström**: Does Distance matter? On geographical dispersion in organisations, 1999, ISBN 91-7219-439-1.
- No 582 **Vanja Josifovski**: Design, Implementation and Evaluation of a Distributed Mediator System for Data Integration, 1999, ISBN 91-7219-482-0.
- No 589 **Rita Kovordányi**: Modeling and Simulating Inhibitory Mechanisms in Mental Image Reinterpretation - Towards Cooperative Human-Computer Creativity, 1999, ISBN 91-7219-506-1.
- No 592 **Mikael Ericsson**: Supporting the Use of Design Knowledge - An Assessment of Commenting Agents, 1999, ISBN 91-7219-532-0.
- No 593 **Lars Karlsson**: Actions, Interactions and Narratives, 1999, ISBN 91-7219-534-7.
- No 594 **C. G. Mikael Johansson**: Social and Organizational Aspects of Requirements Engineering Methods - A practice-oriented approach, 1999, ISBN 91-7219-541-X.
- No 595 **Jörgen Hansson**: Value-Driven Multi-Class Overload Management in Real-Time Database Systems, 1999, ISBN 91-7219-542-8.
- No 596 **Niklas Hallberg**: Incorporating User Values in the Design of Information Systems and Services in the Public Sector: A Methods Approach, 1999, ISBN 91-7219-543-6.
- No 597 **Vivian Vimarlund**: An Economic Perspective on the Analysis of Impacts of Information Technology: From Case Studies in Health-Care towards General Models and Theories, 1999, ISBN 91-7219-544-4.
- No 598 **Johan Jenvald**: Methods and Tools in Computer-Supported Taskforce Training, 1999, ISBN 91-7219-547-9.
- No 607 **Magnus Merkel**: Understanding and enhancing translation by parallel text processing, 1999, ISBN 91-7219-614-9.
- No 611 **Silvia Coradeschi**: Anchoring symbols to sensory data, 1999, ISBN 91-7219-623-8.
- No 613 **Man Lin**: Analysis and Synthesis of Reactive Systems: A Generic Layered Architecture Perspective, 1999, ISBN 91-7219-630-0.
- No 618 **Jimmy Tjäder**: Systemimplementering i praktiken - En studie av logiker i fyra projekt, 1999, ISBN 91-7219-657-2.
- No 627 **Vadim Engelson**: Tools for Design, Interactive Simulation, and Visualization of Object-Oriented Models in Scientific Computing, 2000, ISBN 91-7219-709-9.
- No 637 **Esa Falkenroth**: Database Technology for Control and Simulation, 2000, ISBN 91-7219-766-8.
- No 639 **Per-Arne Persson**: Bringing Power and Knowledge Together: Information Systems Design for Autonomy and Control in Command Work, 2000, ISBN 91-7219-796-X.
- No 660 **Erik Larsson**: An Integrated System-Level Design for Testability Methodology, 2000, ISBN 91-7219-890-7.
- No 688 **Marcus Bjärelund**: Model-based Execution Monitoring, 2001, ISBN 91-7373-016-5.
- No 689 **Joakim Gustafsson**: Extending Temporal Action Logic, 2001, ISBN 91-7373-017-3.
- No 720 **Carl-Johan Petri**: Organizational Information Provision - Managing Mandatory and Discretionary Use of Information Technology, 2001, ISBN 91-7373-126-9.
- No 724 **Paul Scerri**: Designing Agents for Systems with Adjustable Autonomy, 2001, ISBN 91-7373-207-9.
- No 725 **Tim Heyer**: Semantic Inspection of Software Artifacts: From Theory to Practice, 2001, ISBN 91-7373-208-7.
- No 726 **Pär Carlshamre**: A Usability Perspective on Requirements Engineering - From Methodology to Product Development, 2001, ISBN 91-7373-212-5.
- No 732 **Juha Takkinen**: From Information Management to Task Management in Electronic Mail, 2002, ISBN 91-7373-258-3.

- No 745 **Johan Åberg:** Live Help Systems: An Approach to Intelligent Help for Web Information Systems, 2002, ISBN 91-7373-311-3.
- No 746 **Rego Granlund:** Monitoring Distributed Teamwork Training, 2002, ISBN 91-7373-312-1.
- No 757 **Henrik André-Jönsson:** Indexing Strategies for Time Series Data, 2002, ISBN 917373-346-6.
- No 747 **Anneli Hagdahl:** Development of IT-supported Interorganisational Collaboration - A Case Study in the Swedish Public Sector, 2002, ISBN 91-7373-314-8.
- No 749 **Sofie Pilemalm:** Information Technology for Non-Profit Organisations - Extended Participatory Design of an Information System for Trade Union Shop Stewards, 2002, ISBN 91-7373-318-0.
- No 765 **Stefan Holmlid:** Adapting users: Towards a theory of use quality, 2002, ISBN 91-7373-397-0.
- No 771 **Magnus Morin:** Multimedia Representations of Distributed Tactical Operations, 2002, ISBN 91-7373-421-7.
- No 772 **Pawel Pietrzak:** A Type-Based Framework for Locating Errors in Constraint Logic Programs, 2002, ISBN 91-7373-422-5.
- No 758 **Erik Berglund:** Library Communication Among Programmers Worldwide, 2002, ISBN 91-7373-349-0.
- No 774 **Choong-ho Yi:** Modelling Object-Oriented Dynamic Systems Using a Logic-Based Framework, 2002, ISBN 91-7373-424-1.
- No 779 **Mathias Broxvall:** A Study in the Computational Complexity of Temporal Reasoning, 2002, ISBN 91-7373-440-3.
- No 793 **Asmus Pandikow:** A Generic Principle for Enabling Interoperability of Structured and Object-Oriented Analysis and Design Tools, 2002, ISBN 91-7373-479-9.
- No 785 **Lars Hult:** Publika Informationstjänster. En studie av den Internetbaserade encyklopedins bruksegenskaper, 2003, ISBN 91-7373-461-6.
- No 800 **Lars Taxén:** A Framework for the Coordination of Complex Systems' Development, 2003, ISBN 91-7373-604-X.
- No 808 **Klas Gäre:** Tre perspektiv på förväntningar och förändringar i samband med införande av informationssystem, 2003, ISBN 91-7373-618-X.
- No 821 **Mikael Kindborg:** Concurrent Comics - programming of social agents by children, 2003, ISBN 91-7373-651-1.
- No 823 **Christina Ölvingson:** On Development of Information Systems with GIS Functionality in Public Health Informatics: A Requirements Engineering Approach, 2003, ISBN 91-7373-656-2.
- No 828 **Tobias Ritzau:** Memory Efficient Hard Real-Time Garbage Collection, 2003, ISBN 91-7373-666-X.
- No 833 **Paul Pop:** Analysis and Synthesis of Communication-Intensive Heterogeneous Real-Time Systems, 2003, ISBN 91-7373-683-X.
- No 852 **Johan Moe:** Observing the Dynamic Behaviour of Large Distributed Systems to Improve Development and Testing - An Empirical Study in Software Engineering, 2003, ISBN 91-7373-779-8.
- No 867 **Erik Herzog:** An Approach to Systems Engineering Tool Data Representation and Exchange, 2004, ISBN 91-7373-929-4.
- No 872 **Ascel Berglund:** Augmenting the Remote Control: Studies in Complex Information Navigation for Digital TV, 2004, ISBN 91-7373-940-5.
- No 869 **Jo Skåmedal:** Telecommuting's Implications on Travel and Travel Patterns, 2004, ISBN 91-7373-935-9.
- No 870 **Linda Askenäs:** The Roles of IT - Studies of Organising when Implementing and Using Enterprise Systems, 2004, ISBN 91-7373-936-7.
- No 874 **Annika Flycht-Eriksson:** Design and Use of Ontologies in Information-Providing Dialogue Systems, 2004, ISBN 91-7373-947-2.
- No 873 **Peter Bunus:** Debugging Techniques for Equation-Based Languages, 2004, ISBN 91-7373-941-3.
- No 876 **Jonas Mellin:** Resource-Predictable and Efficient Monitoring of Events, 2004, ISBN 91-7373-956-1.
- No 883 **Magnus Bång:** Computing at the Speed of Paper: Ubiquitous Computing Environments for Healthcare Professionals, 2004, ISBN 91-7373-971-5.
- No 882 **Robert Eklund:** Disfluency in Swedish human-human and human-machine travel booking dialogues, 2004, ISBN 91-7373-966-9.
- No 887 **Anders Lindström:** English and other Foreign Linguistic Elements in Spoken Swedish. Studies of Productive Processes and their Modelling using Finite-State Tools, 2004, ISBN 91-7373-981-2.
- No 889 **Zhiping Wang:** Capacity-Constrained Production-inventory systems - Modelling and Analysis in both a traditional and an e-business context, 2004, ISBN 91-85295-08-6.
- No 893 **Pernilla Qvarfordt:** Eyes on Multimodal Interaction, 2004, ISBN 91-85295-30-2.
- No 910 **Magnus Kald:** In the Borderland between Strategy and Management Control - Theoretical Framework and Empirical Evidence, 2004, ISBN 91-85295-82-5.
- No 918 **Jonas Lundberg:** Shaping Electronic News: Genre Perspectives on Interaction Design, 2004, ISBN 91-85297-14-3.
- No 900 **Mattias Arvola:** Shades of use: The dynamics of interaction design for sociable use, 2004, ISBN 91-85295-42-6.
- No 920 **Luis Alejandro Cortés:** Verification and Scheduling Techniques for Real-Time Embedded Systems, 2004, ISBN 91-85297-21-6.
- No 929 **Diana Szentivanyi:** Performance Studies of Fault-Tolerant Middleware, 2005, ISBN 91-85297-58-5.
- No 933 **Mikael Cäker:** Management Accounting as Constructing and Opposing Customer Focus: Three Case Studies on Management Accounting and Customer Relations, 2005, ISBN 91-85297-64-X.
- No 937 **Jonas Kvarnström:** TALplanner and Other Extensions to Temporal Action Logic, 2005, ISBN 91-85297-75-5.
- No 938 **Bourhane Kadmiry:** Fuzzy Gain-Scheduled Visual Servoing for Unmanned Helicopter, 2005, ISBN 91-85297-76-3.
- No 945 **Gert Jervan:** Hybrid Built-In Self-Test and Test Generation Techniques for Digital Systems, 2005, ISBN 91-85297-97-6.
- No 946 **Anders Arpteg:** Intelligent Semi-Structured Information Extraction, 2005, ISBN 91-85297-98-4.
- No 947 **Ola Angelsmark:** Constructing Algorithms for Constraint Satisfaction and Related Problems - Methods and Applications, 2005, ISBN 91-85297-99-2.
- No 963 **Calin Curescu:** Utility-based Optimisation of Resource Allocation for Wireless Networks, 2005, ISBN 91-85457-07-8.
- No 972 **Björn Johansson:** Joint Control in Dynamic Situations, 2005, ISBN 91-85457-31-0.
- No 974 **Dan Lawesson:** An Approach to Diagnosability Analysis for Interacting Finite State Systems, 2005, ISBN 91-85457-39-6.
- No 979 **Claudiu Duma:** Security and Trust Mechanisms for Groups in Distributed Services, 2005, ISBN 91-85457-54-X.
- No 983 **Sorin Manolache:** Analysis and Optimisation of Real-Time Systems with Stochastic Behaviour, 2005, ISBN 91-85457-60-4.
- No 986 **Yuxiao Zhao:** Standards-Based Application Integration for Business-to-Business Communications, 2005, ISBN 91-85457-66-3.
- No 1004 **Patrik Haslum:** Admissible Heuristics for Automated Planning, 2006, ISBN 91-85497-28-2.

- No 1005 **Aleksandra Tešanovic:** Developing Reusable and Reconfigurable Real-Time Software using Aspects and Components, 2006, ISBN 91-85497-29-0.
- No 1008 **David Dinka:** Role, Identity and Work: Extending the design and development agenda, 2006, ISBN 91-85497-42-8.
- No 1009 **Iakov Nakhimovskii:** Contributions to the Modeling and Simulation of Mechanical Systems with Detailed Contact Analysis, 2006, ISBN 91-85497-43-X.
- No 1013 **Wilhelm Dahllöf:** Exact Algorithms for Exact Satisfiability Problems, 2006, ISBN 91-85523-97-6.
- No 1016 **Levon Saldamli:** PDEModelica - A High-Level Language for Modeling with Partial Differential Equations, 2006, ISBN 91-85523-84-4.
- No 1017 **Daniel Karlsson:** Verification of Component-based Embedded System Designs, 2006, ISBN 91-85523-79-8.
- No 1018 **Ioan Chisalița:** Communication and Networking Techniques for Traffic Safety Systems, 2006, ISBN 91-85523-77-1.
- No 1019 **Tarja Susi:** The Puzzle of Social Activity - The Significance of Tools in Cognition and Cooperation, 2006, ISBN 91-85523-71-2.
- No 1021 **Andrzej Bednarski:** Integrated Optimal Code Generation for Digital Signal Processors, 2006, ISBN 91-85523-69-0.
- No 1022 **Peter Aronsson:** Automatic Parallelization of Equation-Based Simulation Programs, 2006, ISBN 91-85523-68-2.
- No 1030 **Robert Nilsson:** A Mutation-based Framework for Automated Testing of Timeliness, 2006, ISBN 91-85523-35-6.
- No 1034 **Jon Edvardsson:** Techniques for Automatic Generation of Tests from Programs and Specifications, 2006, ISBN 91-85523-31-3.
- No 1035 **Vaida Jakoniene:** Integration of Biological Data, 2006, ISBN 91-85523-28-3.
- No 1045 **Genevieve Gorrell:** Generalized Hebbian Algorithms for Dimensionality Reduction in Natural Language Processing, 2006, ISBN 91-85643-88-2.
- No 1051 **Yu-Hsing Huang:** Having a New Pair of Glasses - Applying Systemic Accident Models on Road Safety, 2006, ISBN 91-85643-64-5.
- No 1054 **Åsa Hedenskog:** Perceive those things which cannot be seen - A Cognitive Systems Engineering perspective on requirements management, 2006, ISBN 91-85643-57-2.
- No 1061 **Cécile Åberg:** An Evaluation Platform for Semantic Web Technology, 2007, ISBN 91-85643-31-9.
- No 1073 **Mats Grindal:** Handling Combinatorial Explosion in Software Testing, 2007, ISBN 978-91-85715-74-9.
- No 1075 **Almut Herzog:** Usable Security Policies for Runtime Environments, 2007, ISBN 978-91-85715-65-7.
- No 1079 **Magnus Wahlström:** Algorithms, measures, and upper bounds for Satisfiability and related problems, 2007, ISBN 978-91-85715-55-8.
- No 1083 **Jesper Andersson:** Dynamic Software Architectures, 2007, ISBN 978-91-85715-46-6.
- No 1086 **Ulf Johansson:** Obtaining Accurate and Comprehensible Data Mining Models - An Evolutionary Approach, 2007, ISBN 978-91-85715-34-3.
- No 1089 **Traian Pop:** Analysis and Optimisation of Distributed Embedded Systems with Heterogeneous Scheduling Policies, 2007, ISBN 978-91-85715-27-5.
- No 1091 **Gustav Nordh:** Complexity Dichotomies for CSP-related Problems, 2007, ISBN 978-91-85715-20-6.
- No 1106 **Per Ola Kristensson:** Discrete and Continuous Shape Writing for Text Entry and Control, 2007, ISBN 978-91-85831-77-7.
- No 1110 **He Tan:** Aligning Biomedical Ontologies, 2007, ISBN 978-91-85831-56-2.
- No 1112 **Jessica Lindblom:** Minding the body - Interacting socially through embodied action, 2007, ISBN 978-91-85831-48-7.
- No 1113 **Pontus Wärnestål:** Dialogue Behavior Management in Conversational Recommender Systems, 2007, ISBN 978-91-85831-47-0.
- No 1120 **Thomas Gustafsson:** Management of Real-Time Data Consistency and Transient Overloads in Embedded Systems, 2007, ISBN 978-91-85831-33-3.
- No 1127 **Alexandru Andrei:** Energy Efficient and Predictable Design of Real-time Embedded Systems, 2007, ISBN 978-91-85831-06-7.
- No 1139 **Per Wikberg:** Eliciting Knowledge from Experts in Modeling of Complex Systems: Managing Variation and Interactions, 2007, ISBN 978-91-85895-66-3.
- No 1143 **Mehdi Amirjoo:** QoS Control of Real-Time Data Services under Uncertain Workload, 2007, ISBN 978-91-85895-49-6.
- No 1150 **Sanny Syberfeldt:** Optimistic Replication with Forward Conflict Resolution in Distributed Real-Time Databases, 2007, ISBN 978-91-85895-27-4.
- No 1155 **Beatrice Alenljung:** Envisioning a Future Decision Support System for Requirements Engineering - A Holistic and Human-centred Perspective, 2008, ISBN 978-91-85895-11-3.
- No 1156 **Artur Wilk:** Types for XML with Application to Xcerpt, 2008, ISBN 978-91-85895-08-3.
- No 1183 **Adrian Pop:** Integrated Model-Driven Development Environments for Equation-Based Object-Oriented Languages, 2008, ISBN 978-91-7393-895-2.
- No 1185 **Jörgen Skågeby:** Gifting Technologies - Ethnographic Studies of End-users and Social Media Sharing, 2008, ISBN 978-91-7393-892-1.
- No 1187 **Imad-Eldin Ali Abugessaisa:** Analytical tools and information-sharing methods supporting road safety organizations, 2008, ISBN 978-91-7393-887-7.
- No 1204 **H. Joe Steinhauer:** A Representation Scheme for Description and Reconstruction of Object Configurations Based on Qualitative Relations, 2008, ISBN 978-91-7393-823-5.
- No 1222 **Anders Larsson:** Test Optimization for Core-based System-on-Chip, 2008, ISBN 978-91-7393-768-9.
- No 1238 **Andreas Borg:** Processes and Models for Capacity Requirements in Telecommunication Systems, 2009, ISBN 978-91-7393-700-9.
- No 1240 **Fredrik Heintz:** DyKnow: A Stream-Based Knowledge Processing Middleware Framework, 2009, ISBN 978-91-7393-696-5.
- No 1241 **Birgitta Lindström:** Testability of Dynamic Real-Time Systems, 2009, ISBN 978-91-7393-695-8.
- No 1244 **Eva Blomqvist:** Semi-automatic Ontology Construction based on Patterns, 2009, ISBN 978-91-7393-683-5.
- No 1249 **Rogier Woltjer:** Functional Modeling of Constraint Management in Aviation Safety and Command and Control, 2009, ISBN 978-91-7393-659-0.
- No 1260 **Gianpaolo Conte:** Vision-Based Localization and Guidance for Unmanned Aerial Vehicles, 2009, ISBN 978-91-7393-603-3.
- No 1262 **AnnMarie Ericsson:** Enabling Tool Support for Formal Analysis of ECA Rules, 2009, ISBN 978-91-7393-598-2.
- No 1266 **Jiri Trnka:** Exploring Tactical Command and Control: A Role-Playing Simulation Approach, 2009, ISBN 978-91-7393-571-5.
- No 1268 **Bahlol Rahimi:** Supporting Collaborative Work through ICT - How End-users Think of and Adopt Integrated Health Information Systems, 2009, ISBN 978-91-7393-550-0.
- No 1274 **Fredrik Kuivinen:** Algorithms and Hardness Results for Some Valued CSPs, 2009, ISBN 978-91-7393-525-8.
- No 1281 **Gunnar Mathiason:** Virtual Full Replication for Scalable Distributed Real-Time Databases, 2009, ISBN 978-91-7393-503-6.

- No 1290 **Viacheslav Izosimov**: Scheduling and Optimization of Fault-Tolerant Distributed Embedded Systems, 2009, ISBN 978-91-7393-482-4.
- No 1294 **Johan Thapper**: Aspects of a Constraint Optimisation Problem, 2010, ISBN 978-91-7393-464-0.
- No 1306 **Susanna Nilsson**: Augmentation in the Wild: User Centered Development and Evaluation of Augmented Reality Applications, 2010, ISBN 978-91-7393-416-9.
- No 1313 **Christer Thörn**: On the Quality of Feature Models, 2010, ISBN 978-91-7393-394-0.
- No 1321 **Zhiyuan He**: Temperature Aware and Defect-Probability Driven Test Scheduling for System-on-Chip, 2010, ISBN 978-91-7393-378-0.
- No 1333 **David Broman**: Meta-Languages and Semantics for Equation-Based Modeling and Simulation, 2010, ISBN 978-91-7393-335-3.
- No 1337 **Alexander Siemers**: Contributions to Modelling and Visualisation of Multibody Systems Simulations with Detailed Contact Analysis, 2010, ISBN 978-91-7393-317-9.
- No 1354 **Mikael Asplund**: Disconnected Discoveries: Availability Studies in Partitioned Networks, 2010, ISBN 978-91-7393-278-3.
- No 1359 **Jana Rambusch**: Mind Games Extended: Understanding Gameplay as Situated Activity, 2010, ISBN 978-91-7393-252-3.
- No 1373 **Sonia Sangari**: Head Movement Correlates to Focus Assignment in Swedish, 2011, ISBN 978-91-7393-154-0.
- No 1374 **Jan-Erik Källhammer**: Using False Alarms when Developing Automotive Active Safety Systems, 2011, ISBN 978-91-7393-153-3.
- No 1375 **Mattias Eriksson**: Integrated Code Generation, 2011, ISBN 978-91-7393-147-2.
- No 1381 **Ola Leifler**: Affordances and Constraints of Intelligent Decision Support for Military Command and Control – Three Case Studies of Support Systems, 2011, ISBN 978-91-7393-133-5.
- No 1386 **Soheil Samii**: Quality-Driven Synthesis and Optimization of Embedded Control Systems, 2011, ISBN 978-91-7393-102-1.
- No 1419 **Erik Kuiper**: Geographic Routing in Intermittently-connected Mobile Ad Hoc Networks: Algorithms and Performance Models, 2012, ISBN 978-91-7519-981-8.
- No 1451 **Sara Stymne**: Text Harmonization Strategies for Phrase-Based Statistical Machine Translation, 2012, ISBN 978-91-7519-887-3.
- No 1455 **Alberto Montebelli**: Modeling the Role of Energy Management in Embodied Cognition, 2012, ISBN 978-91-7519-882-8.
- No 1465 **Mohammad Saifullah**: Biologically-Based Interactive Neural Network Models for Visual Attention and Object Recognition, 2012, ISBN 978-91-7519-838-5.
- No 1490 **Tomas Bengtsson**: Testing and Logic Optimization Techniques for Systems on Chip, 2012, ISBN 978-91-7519-742-5.
- No 1481 **David Byers**: Improving Software Security by Preventing Known Vulnerabilities, 2012, ISBN 978-91-7519-784-5.
- No 1496 **Tommy Färnqvist**: Exploiting Structure in CSP-related Problems, 2013, ISBN 978-91-7519-711-1.
- No 1503 **John Wilander**: Contributions to Specification, Implementation, and Execution of Secure Software, 2013, ISBN 978-91-7519-681-7.
- No 1506 **Magnus Ingmarsson**: Creating and Enabling the Useful Service Discovery Experience, 2013, ISBN 978-91-7519-662-6.
- No 1547 **Wladimir Schamai**: Model-Based Verification of Dynamic System Behavior against Requirements: Method, Language, and Tool, 2013, ISBN 978-91-7519-505-6.
- No 1551 **Henrik Svensson**: Simulations, 2013, ISBN 978-91-7519-491-2.
- No 1559 **Sergiu Raffliu**: Stability of Adaptive Distributed Real-Time Systems with Dynamic Resource Management, 2013, ISBN 978-91-7519-471-4.
- No 1581 **Usman Dastgeer**: Performance-aware Component Composition for GPU-based Systems, 2014, ISBN 978-91-7519-383-0.
- No 1602 **Cai Li**: Reinforcement Learning of Locomotion based on Central Pattern Generators, 2014, ISBN 978-91-7519-313-7.
- No 1652 **Roland Samlaus**: An Integrated Development Environment with Enhanced Domain-Specific Interactive Model Validation, 2015, ISBN 978-91-7519-090-7.
- No 1663 **Hannes Uppman**: On Some Combinatorial Optimization Problems: Algorithms and Complexity, 2015, ISBN 978-91-7519-072-3.
- No 1664 **Martin Sjölund**: Tools and Methods for Analysis, Debugging, and Performance Improvement of Equation-Based Models, 2015, ISBN 978-91-7519-071-6.
- No 1666 **Kristian Stavåker**: Contributions to Simulation of Modelica Models on Data-Parallel Multi-Core Architectures, 2015, ISBN 978-91-7519-068-6.
- No 1680 **Adrian Lifa**: Hardware/Software Codesign of Embedded Systems with Reconfigurable and Heterogeneous Platforms, 2015, ISBN 978-91-7519-040-2.
- No 1685 **Bogdan Tanasa**: Timing Analysis of Distributed Embedded Systems with Stochastic Workload and Reliability Constraints, 2015, ISBN 978-91-7519-022-8.
- No 1691 **Håkan Warnqvist**: Troubleshooting Trucks – Automated Planning and Diagnosis, 2015, ISBN 978-91-7685-993-3.
- No 1702 **Nima Aghae**: Thermal Issues in Testing of Advanced Systems on Chip, 2015, ISBN 978-91-7685-949-0.
- No 1715 **Maria Vasilevskaya**: Security in Embedded Systems: A Model-Based Approach with Risk Metrics, 2015, ISBN 978-91-7685-917-9.
- No 1729 **Ke Jiang**: Security-Driven Design of Real-Time Embedded System, 2016, ISBN 978-91-7685-884-4.
- No 1733 **Victor Lagerkvist**: Strong Partial Clones and the Complexity of Constraint Satisfaction Problems: Limitations and Applications, 2016, ISBN 978-91-7685-856-1.
- No 1734 **Chandan Roy**: An Informed System Development Approach to Tropical Cyclone Track and Intensity Forecasting, 2016, ISBN 978-91-7685-854-7.
- No 1746 **Amir Aminifar**: Analysis, Design, and Optimization of Embedded Control Systems, 2016, ISBN 978-91-7685-826-4.
- No 1747 **Ekhiotz Vergara**: Energy Modelling and Fairness for Efficient Mobile Communication, 2016, ISBN 978-91-7685-822-6.
- No 1748 **Dag Sonntag**: Chain Graphs – Interpretations, Expressiveness and Learning Algorithms, 2016, ISBN 978-91-7685-818-9.
- No 1768 **Anna Vapen**: Web Authentication using Third-Parties in Untrusted Environments, 2016, ISBN 978-91-7685-753-3.
- No 1778 **Magnus Jandinger**: On a Need to Know Basis: A Conceptual and Methodological Framework for Modelling and Analysis of Information Demand in an Enterprise Context, 2016, ISBN 978-91-7685-713-7.
- No 1798 **Rahul Hiran**: Collaborative Network Security: Targeting Wide-area Routing and Edge-network Attacks, 2016, ISBN 978-91-7685-662-8.
- No 1813 **Nicolas Melot**: Algorithms and Framework for Energy Efficient Parallel Stream Computing on Many-Core Architectures, 2016, ISBN 978-91-7685-623-9.
- No 1823 **Amy Rankin**: Making Sense of Adaptations: Resilience in High-Risk Work, 2017, ISBN 978-91-7685-596-6.

- No 1831 **Lisa Malmberg:** Building Design Capability in the Public Sector: Expanding the Horizons of Development, 2017, ISBN 978-91-7685-585-0.
- No 1851 **Marcus Bendtsen:** Gated Bayesian Networks, 2017, ISBN 978-91-7685-525-6.
- No 1852 **Zlatan Dragisic:** Completion of Ontologies and Ontology Networks, 2017, ISBN 978-91-7685-522-5.
- No 1854 **Meysam Aghighi:** Computational Complexity of some Optimization Problems in Planning, 2017, ISBN 978-91-7685-519-5.
- No 1863 **Simon Ståhlberg:** Methods for Detecting Unsolvable Planning Instances using Variable Projection, 2017, ISBN 978-91-7685-498-3.
- No 1879 **Karl Hammar:** Content Ontology Design Patterns: Qualities, Methods, and Tools, 2017, ISBN 978-91-7685-454-9.
- No 1887 **Ivan Ukhov:** System-Level Analysis and Design under Uncertainty, 2017, ISBN 978-91-7685-426-6.
- No 1891 **Valentina Ivanova:** Fostering User Involvement in Ontology Alignment and Alignment Evaluation, 2017, ISBN 978-91-7685-403-7.
- No 1902 **Vengatanathan Krishnamoorthi:** Efficient HTTP-based Adaptive Streaming of Linear and Interactive Videos, 2018, ISBN 978-91-7685-371-9.

Linköping Studies in Arts and Science

- No 504 **Ing-Marie Jonsson:** Social and Emotional Characteristics of Speech-based In-Vehicle Information Systems: Impact on Attitude and Driving Behaviour, 2009, ISBN 978-91-7393-478-7.
- No 586 **Fabian Segelström:** Stakeholder Engagement for Service Design: How service designers identify and communicate insights, 2013, ISBN 978-91-7519-554-4.
- No 618 **Johan Blomkvist:** Representing Future Situations of Service: Prototyping in Service Design, 2014, ISBN 978-91-7519-343-4.
- No 620 **Marcus Mast:** Human-Robot Interaction for Semi-Autonomous Assistive Robots, 2014, ISBN 978-91-7519-319-9.
- No 677 **Peter Berggren:** Assessing Shared Strategic Understanding, 2016, ISBN 978-91-7685-786-1.
- No 695 **Mattias Forsblad:** Distributed cognition in home environments: The prospective memory and cognitive practices of older adults, 2016, ISBN 978-91-7685-686-4.
- No 4 **Owen Eriksson:** Kommunikationskvalitet hos informationssystem och affärsprocesser, 2000, ISBN 91-7219-811-7.
- No 5 **Mikael Lind:** Från system till process - kriterier för processbestämning vid verksamhetsanalys, 2001, ISBN 91-7373-067-X.
- No 6 **Ulf Melin:** Koordination och informationssystem i företag och nätverk, 2002, ISBN 91-7373-278-8.
- No 7 **Pär J. Ågerfalk:** Information Systems Actability - Understanding Information Technology as a Tool for Business Action and Communication, 2003, ISBN 91-7373-628-7.
- No 8 **Ulf Seigerroth:** Att förstå och förändra systemutvecklingsverksamheter - en taxonomi för metautveckling, 2003, ISBN 91-7373-736-4.
- No 9 **Karin Hedström:** Spår av datoriseringens värden - Effekter av IT i äldreomsorg, 2004, ISBN 91-7373-963-4.
- No 10 **Ewa Braf:** Knowledge Demanded for Action - Studies on Knowledge Mediation in Organisations, 2004, ISBN 91-85295-47-7.
- No 11 **Fredrik Karlsson:** Method Configuration method and computerized tool support, 2005, ISBN 91-85297-48-8.
- No 12 **Malin Nordström:** Styrbar systemförvaltning - Att organisera systemförvaltningsverksamhet med hjälp av effektiva förvaltningsobjekt, 2005, ISBN 91-85297-60-7.
- No 13 **Stefan Holgersson:** Yrke: POLIS - Yrkeskunskap, motivation, IT-system och andra förutsättningar för polisarbete, 2005, ISBN 91-85299-43-X.
- No 14 **Benneth Christiansson, Marie-Therese Christiansson:** Mötet mellan process och komponent - mot ett ramverk för en verksamhetsnära kravspecifikation vid anskaffning av komponent-baserade informationssystem, 2006, ISBN 91-85643-22-X.

Linköping Studies in Statistics

- No 9 **Davood Shahsavani:** Computer Experiments Designed to Explore and Approximate Complex Deterministic Models, 2008, ISBN 978-91-7393-976-8.
- No 10 **Karl Wahlin:** Roadmap for Trend Detection and Assessment of Data Quality, 2008, ISBN 978-91-7393-792-4.
- No 11 **Oleg Sysoev:** Monotonic regression for large multivariate datasets, 2010, ISBN 978-91-7393-412-1.
- No 13 **Agné Burauskaitė-Harju:** Characterizing Temporal Change and Inter-Site Correlations in Daily and Sub-daily Precipitation Extremes, 2011, ISBN 978-91-7393-110-6.

Linköping Studies in Information Science

- No 1 **Karin Axelsson:** Metodisk systemstrukturerig- att skapa samstämmighet mellan informationssystem-arkitektur och verksamhet, 1998. ISBN 9172-19-296-8.
- No 2 **Stefan Cronholm:** Metodverktyg och användbarhet - en studie av datorstödd metodbaserad systemutveckling, 1998, ISBN 9172-19-299-2.
- No 3 **Anders Avdic:** Användare och utvecklare - om anveckling med kalkylprogram, 1999. ISBN 91-7219-606-8.